DETERMINISTIC AND STOCHASTIC INVENTORY ROUTING PROBLEMS WITH BACKORDERS USING ARTIFICIAL BEE COLONY

HUDA ZUHRAH BINTI AB HALIM

FACULTY OF SCIENCE
UNIVERSITY OF MALAYA
KUALA LUMPUR

2019

DETERMINISTIC AND STOCHASTIC INVENTORY ROUTING PROBLEMS WITH BACKORDERS USING ARTIFICIAL BEE COLONY

HUDA ZUHRAH BINTI AB HALIM

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

INSTITUTE OF MATHEMATICAL SCIENCES
FACULTY OF SCIENCE
UNIVERSITY OF MALAYA
KUALA LUMPUR

2019

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **HUDA ZUHRAH BINTI AB. HALIM**

Matric No: **SHB 120012**

Name of Degree: **DOCTOR OF PHILOSOPHY**

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

DETERMINISTIC AND STOCHASTIC INVENTORY ROUTING PROBLEMS WITH BACKORDERS USING ARTIFICIAL BEE COLONY

Field of Study: **OPERATIONS RESEARCH**

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation:

DETERMINISTIC AND STOCHASTIC INVENTORY ROUTING PROBLEMS WITH BACKORDERS USING ARTIFICIAL BEE COLONY

ABSTRACT

This thesis is devoted to solving the inventory routing problem (IRP) and its variants. It is well known that the IRP is an important component in the implementation of Vendor Managed Inventory. The IRP comprises the coordination of two components: inventory management and vehicle routing problem. Details of the components define the variation of the IRP. Four different variants of IRP are studied in this thesis. The first variant is a many-to-one distribution network, consisting of a depot, an assembly plant, and geographically dispersed suppliers where a capacitated homogeneous vehicle, housed at the depot delivers a distinct product from the suppliers to fulfill the deterministic demand specified by the assembly plant. The inventory holding cost is assumed to be product specific and only incurred at the assembly plant. An Artificial Bee Colony (ABC) algorithm is proposed for the problem. The performance of ABC is evaluated on existing datasets and compared with Scatter Search (SS) and Genetic Algorithm (GA). The statistical analysis carried out shows that the ABC, SS and GA are significantly different with 95% confidence level with ABC performing significantly better compared to SS and GA. An enhanced ABC is also developed which performs better in terms of quality of the solutions. IRP with backordering (IRPB) represents the second variant and it defines the condition where unsatisfied demand is delayed and fulfilled in future periods. The network of IRPB consists of a supplier and geographically scattered customers, where a set of vehicles performs the delivery to fulfill customer's demand. Two ABC algorithms are proposed which embed two inventory updating mechanisms; *random exchange* and *guided exchange*. Results of both ABCs are compared with existing literature and bounds found by CPLEX. The statistical analysis result shows that all the algorithms are significantly different with 95% confidence level. The third variant investigated is IRP with stochastic demand. The main characteristic of the demand where the demand is known in a probabilistic sense and the demand is gradually revealed at the end of each period (dynamic). The problem is known as dynamic and stochastic inventory routing problem (DSIRP) and the distribution network considered consists of a supplier and a set of retailers. An order-up-to level inventory policy is applied, and each unit of positive inventory incurs a holding cost while a penalty is incurred for each negative inventory level. The transportation of the product is handled by a third party. The DSIRP is modeled as stochastic dynamic programming and solved using a matheuristic, enhanced hybrid rollout algorithm. The enhanced algorithm embeds additional controls generated using ABC. The DSIRP is then extended to handle backorder decisions (DSIRPB) which is the fourth variant of IRP studied. A new MILP for DSIRPB is formulated and used within the algorithm. The DSIRPB is modeled as stochastic dynamic programming and solved using hybrid rollout algorithm. Both DSIRP and DSIRPB are evaluated on 60 instances. Analysis of controls, the number of visits, quantity delivery and backorders are carried out to observe the patterns.

Keywords: Inventory Routing Problem, Artificial Bee Colony, backordering, stochastic IRP.

MASALAH LALUAN INVENTORI BERKETENTUAN DAN STOKASTIK DENGAN PENANGGUHAN MENGGUNAKAN ALGORITMA KOLONI LEBAH BUATAN

ABSTRAK

Tesis ini ditumpukan untuk menyelesaikan masalah laluan inventori (*Inventory Routing Problem* (*IRP*)) dan variasinya. Seperti yang diketahui, IRP adalah komponen penting di dalam implementasi inventori terurus pembekal. IRP terdiri daripada koordinasi dua komponen: pengurusan inventori dan masalah laluan kenderaan. Perincian butiran kepada komponen-komponen ini menentukan variasi kepada IRP. Empat variasi IRP yang berbeza dikaji di dalam tesis ini. Variasi IRP yang pertama ialah rangkaian pengagihan banyak-ke-satu yang mempunyai depoh, kilang pemasangan dan pembekal-pembekal yang berserakan di mana kenderaan homogen yang bertempat di depoh, menghantar produk berbeza dari pembekal bagi memenuhi permintaan yang ditetapkan oleh kilang pemasangan sepanjang ufuk masa. Kos pemegangan inventori adalah berbeza mengikut produk dan dikenakan oleh kilang pemasangan. Algoritma koloni lebah buatan (*ABC*) di cadangkan untuk menyelesaikan masalah ini. Prestasi ABC diuji ke atas set data sedia ada dan di bandingkan dengan *Scatter Search* (*SS*) dan *Genetic Algorithm* (*GA*). Analisis statistik yang dilakukan menunjukkan ABC, SS dan GA adalah berbeza secara signifikan pada aras keyakinan 95%, di mana prestasi ABC adalah lebih baik berbanding SS dan GA. ABC juga dipertingkatkan lagi di mana prestasinya lebih baik dengan penyelesaian yang berkualiti. IRP dengan penangguhan (*IRP with backordering* (*IRPB*)) adalah variasi yang kedua di mana ia menakrifkan kondisi permintaan yang tidak dipenuhi, akan di tangguhkan dan dipenuhi pada masa akan datang. Rangkaian pengagihan IRPB termasuklah pembekal dan pelanggan yang berserakan, di mana satu set kenderaan akan membuat penghantaran bagi memenuhi permintaan pelanggan. Dua ABC diusulkan dan di masukkan dengan dua mekanisma untuk mengemas kini inventori: tukaran rawak dan

tukaran berpandu. Keputusan daripada kedua-dua algoritma ini, di bandingkan dengan kajian asal dan dengan batas-batas daripada CPLEX. Keputusan analisis statistik menunjukkan kesemua algoritma ada berbeza secara signifikan dengan 95% aras keyakinan. Variasi ketiga yang dikaji ialah IRP dengan permintaan stokastik. Ciri-ciri utama permintaan adalah permintaan diketahui dalam probabilistik dan ianya didedahkan secara beransur-ansur pada akhir setiap tempoh masa. Masalah ini di kenali sebagai masalah laluan inventori dinamik dan stokastik (*Dynamic and Stochastic Inventory Routing Problem* (*DSIRP*)) dan rangkaian pengagihan yang dipertimbangkan mengandungi satu pembekal dan set peruncit. Polisi inventori order sehingga ke tanda aras (*Order-up-to level* (*OU*)) digunakan dan setiap nilai positif inventori di kenakan kos pemegangan manakala penalti di kenakan bagi setiap negatif inventori. Pengagihan produk di kendalikan oleh pihak ketiga. DSIRP dimodelkan sebagai pengaturcaraan dinamik stokastik dan di selesaikan menggunakan kaedah matheuristik, algoritma hibrid *rollout* yang dipertingkatkan. Algoritma yang dipertingkatkan itu dimasukkan kontrol tambahan yang dijanakan oleh ABC. DSIRP kemudiannya dilanjutkan untuk menangani penangguhan pesanan (*DSIRPB*) iaitu variasi IRP keempat yang dikaji. MILP baru untuk DSIRPB di formulasi dan digunakan di dalam algoritma. DSIRPB di modelkan sebagai pengaturcaraan dinamik stokastik dan diselesaikan menggunakan algoritma hibrid *rollout*. Kedua-dua algoritma DSIRP dan DSIRPB di uji ke atas 60 set data sedia ada. Analisis kontrol, jumlah lawatan, kuantiti penghantaran dan penangguhan di lakukan untuk melihat coraknya.

Kata Kunci: Masalah laluan inventori, Algoritma koloni lebah buatan, penangguhan pesanan, stokastik IRP.

ACKNOWLEDGEMENTS

First and foremost, all praises to the Almighty God, Allah S.W.T for the granting me the strength and knowledge in finishing this thesis. Nothing could be possible without His help and blessing. I would like to thank my first supervisor, Associate Professor Dr. Noor Hasnah Moin, for the opportunity to do research and learning from her without her full support, I would not be able to push through my limit. I appreciate her guidance, advices, ideas and opinions throughout my study and in completing this thesis. I also sincerely thank her for the time spent in proofread and commenting my thesis.

My sincere thanks for my second supervisor, Professor Dr. Nor Aishah Hamzah for her advice and her help throughout the final stage of the thesis. I also appreciate her comments on the statistical analysis parts of the thesis.

Special thanks to MyPhd programme from the Ministry of Higher Education, Malaysia for funding my study and Postgraduate Research Fund (PPP) from University of Malaya. I also wanted to acknowledge the help from Dr. Ng Choung Min and Dr. Norli Anida Abdullah who helped to improved my understanding of the stochastic part of the problem.

I am grateful to my husband Mohd Zaki and my son Muhammad Ammar Zihni for their enormous patience and never ending support during my ups and down. Not forgetting my family, especially my parents Esah Hasan and Abd Halim Ismail for their unconditional support and du’a. Special thanks for my sister, Fauziah Zakiah for her care and time for Ammar when I studied.

Lastly, I would like to thanks my colleague Dr. Dharini, Dr. Najihah, Nadhirah, Raihan, Tan Siu Ching, Lily Wong and Dicky Lim for their constant encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ABC	:	Artificial Bee Colony
ABCGX	:	Artificial Bee Colony with Guided Exchange
ABCIRP	:	Artificial Bee Colony for Inventory Routing Problem
ABCRX	:	Artificial Bee Colony with Random Exchange
ACO	:	Ant Colony Optimization
ADP	:	Approximate Dynamic Programming
BQ	:	Average Backorder Quantity
CVRP	:	Capacitated Vehicle Routing Problem
DP	:	Dynamic Programming
DQ	:	Average Delivery Quantity
DSIRP	:	Dynamic Stochastic Inventory Routing Problem
DSIRPB	:	Dynamic Stochastic Inventory Routing Problem with Backordering
EABCIRP	:	Enhanced Artificial Bee Colony for Inventory Routing Problem
ETCH	:	Estimated Transportation Cost Heuristic
FAR	:	Friedman Aligned Rank test
GA	:	Genetic Algorithm
GT	:	Giant Tour
HC	:	High Inventory Cost
ID	:	Iman-Davenport test
IRP	:	Inventory Routing Problem
IRPB	:	Inventory Routing Problem with Backordering
LC	:	Low Inventory Cost
MDP	:	Markov Decision Process
MILP	:	Mixed Integer Linear Programming

ML	:	Maximum Level inventory policy
NDP	:	Neuro-Dynamic Programming
NV	:	Average Number of Visits
OU	:	Order-up-to level inventory policy
PSO	:	Particle Swarm Optimization
RL	:	Reinforcement Learning
SCM	:	Supply Chain Management
SI	:	Swarm Intelligence
SIRP	:	Stochastic Inventory Routing Problem
SS	:	Scatter Search
SUS	:	Stochastic Universal Sampling selection method
TS	:	Tabu Search
TSP	:	Traveling Salesman Problem
VMI	:	Vendor Managed Inventory
VNS	:	Variable Neighborhood Search
VRP	:	Vehicle Routing Problem

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

This chapter begins by introducing the supply chain system and the concepts of Vendor Managed Inventory (VMI). This is followed by a discussion on inventory routing problem (IRP) which is the main topic of this thesis. The different variations of IRP are presented giving inventory routing problem with backorder a special emphasis. The motivations in carrying out this research due to the importance of the IRP are discussed at length. The objectives of the thesis are outlined in section 1.4 and are divided according to each of the IRP variations studied. Following this are the contributions to the knowledge that have been achieved and the structure of this thesis is presented which summarizes the works done in each of the chapters. A summary is given to conclude the chapter.

1.1 Overview

The supply chain management (SCM) is a complex and challenging system. SCM has been extensively attracting attention and studied by many researchers in both practitioners and academics. Important components in SCM include, but not limited to; planning, manufacturing, logistics, marketing and distributions. The proper coordination of these components results in the company logistics efficiency, and this will help the company to remain competitive in the industries. There are several definitions of SCM available. The definition adopted in this thesis is given below.

"*Supply chain is define as a set of three entities (organization or individuals) directly involved in the upstream (supply) and downstream (distribution) flows of products, services, finances, and/or information from a source to a customer"* (Mentzer et al., 2001).

Vendor Managed Inventory (VMI) is a supply chain strategy where the decision-making is centralized such that the supplier is responsible for the replenishment decision based on availability of the retailer or customer information. Hence the delivery is coordinated by the supplier in an efficient manner. Traditionally, customer managed their own inventory, where customers will place their own order to the supplier whenever the inventory is low and the supplier will deliver the products based on the request. The whole concept of VMI increases the efficiency of the distribution system where suppliers benefit from the utilization of the vehicles and customers has a guaranteed delivery. This is justified by studies of the benefits of the VMI strategy in various aspects of SCM (Erengüç et al., 1999; Yao et al., 2007; Yu, Wang, et al., 2012).

1.2 Inventory Routing Problem (IRP)

Inventory Routing Problem (IRP) is a combination of two main components of the supply chain; inventory management and vehicle routing problem (VRP). The coordination of these two components is normally carried out in the VMI environment where supplier decides when to replenish based on the demand information shared by the customers. This ensures no excessive inventory at the customer's site and shortages are avoided. VMI is a profitable and cost effective strategy and it is a win-win situation as highlighted in several papers (see for example Bertazzi et al. (2005), Archetti et al. (2007) and Archetti and Speranza (2016)). The IRP and its variants are studied under the VMI strategy where it is assumed that the replenishment is coordinated by the supplier.

The complexity of the VRP has been analyzed in Lenstra and Rinnoy Kan (1981), which the authors concluded that VRP is an NP-*hard* (Nondeterministic Polynomial-time *hard*) problem. Other extensions of the VRP, for example, VRP with Time Windows and VRP with Split Deliveries also belonged in the same class (Belfiore et al., 2008). Since the IRP embeds the VRP, it is naturally NP-*hard* problem (Campbell et al., 1998). IRP is worth exploring and has a tremendous potential contribution to the society (Campbell et al., 2002; Coelho & Laporte, 2015; Peres et al., 2017).

In this thesis, four different variants of IRP are studied. The first variant is a many-to-one IRP where the network consists of a supplier and geographically dispersed customers. This is then extended to where backorder decisions are considered and this represents a second variant, known as Inventory Routing Problem with Backordering (IRPB). The third variant is the Dynamic and Stochastic Inventory Routing Problem (DSIRP) where the demands are expressed as some probability functions and the final variant considered is an extension of DSIRP where the backordering decision is considered, known as the Dynamic Stochastic Inventory Routing Problem with Backordering (DSIRPB). The distribution network studied in all variants of IRP is consolidated and centralized, where the customers/retailers are located surrounding the depot. This is the common practice in a real problem (see for example Teo et al. (2001)). Each variant of IRP is discussed in details in Chapters 3, 4, 5 and 6, respectively.

1.3 Motivation and the Importance of the Research

The IRP is simple to describe, but mathematically complex and challenging in practice. The inventory and transportation decisions are made simultaneously to find the optimal policy of the IRP which increases the efficiency of the system. However, as the difficulty of the IRP increases (for example, larger datasets and/or if consider more element, such as multi-product or backorder decisions), it is harder to solve. In traditional IRP the order is placed by the consuming facility (either retailers or customers) to a supplying facility and this normally resulted in excess build-up of inventories at the supplier site in the form of safety stock to hedge against the uncertainty related to the re-supply and the final customer demands. However, very low service levels are achieved. These factors form a vital motivation to study the complexity and inherently difficulty of the problem.

This subsection is divided into 3 parts where the first discusses the connection of the complexity of the IRP and the need of applying metaheuristics and matheuristics.

Secondly, the importance of backorder decisions that leads to customer satisfactions and finally the necessity of solving IRP with stochastic demand where in real life the demands are normally stochastic in nature.

1.3.1 Complexity of the IRP, Metaheuristic and Matheuristic

IRP has been proven to be NP-*hard* combinatorial optimization problem and finding an optimal solution for this problem is difficult, but it is not impossible. Furthermore representing the real world problem as MILP increases the complexity of the problem as more variable needs to be considered. Exact algorithm can be used to find the optimal solution for relatively small problem and for larger problems they are computationally more expensive and time-consuming (if the optimal exist).

Most researchers have opted for metaheuristic because of its ability to solve hard and complex combinatorial optimization problems within a reasonable computational time. Metaheuristics are methods that guide the exploration of the search space with the aim of finding (near) optimal solution (Osman & Kelly, 1996; Ribeiro & Hansen, 2012; Voß et al., 2012). There are several definitions of metaheuristic, and among them, the metaheuristic is defined as:

"*Metaheuristics are typically high-level strategies which guide an underlying, more problem specific heuristics, to increase their performance. The main goal is to avoid the disadvantages of iterative improvement and, in particular, multiple descent by allowing the local search to escape from local optima. This is achieved by either allowing worsening moves or generating new starting solutions for the local search in a more "intelligent" way than just providing random initial solutions. Many of the methods can be interpreted as introducing a bias such that high quality solutions are produced quickly. This bias can be of various forms and can be cast as descent bias (based on the objective function), memory bias (based on previously made decisions) or experience bias (based* *on prior performance). Many of the metaheuristic approaches rely on probabilistic decisions made during the search. But, the main difference to pure random search is that in these algorithms randomness is not used blindly but in an intelligent, biased form*" (Stützle, 1999).

Some examples of metaheuristic methods are Ant Colony Optimization (ACO), Genetic Algorithm (GA), Scatter Search (SS), Tabu Search (TS), Variable Neighborhood Search (VNS), Particle Swarm Optimization (PSO), and the recently introduced Artificial Bee Colony (ABC). ABC was first introduced for numerical optimization (Karaboga, 2005), and extended to combinatorial optimization in 2009 (see Karaboga et al. (2014) for a survey on ABC) specifically in Capacitated Vehicle Routing Problem (CVRP) (Szeto et al., 2011). However, there is no application, as far as we are aware, in inventory routing problem (IRP) when we started this research in 2013. Since ABC is a systematic and powerful technique, it motivates us to extend the applications to IRP and its other variants.

Another interesting heuristic method arises in the IRP is matheuristics. "Matheuristic make use of the mathematical programming models, typically Mixed Integer Linear Programming Problems (MILPs), inside a heuristic scheme. The computational effectiveness of commercial optimization software makes it interesting and promising the design of heuristic solution approaches that make use of the optimal solution of MILPs" (Bertazzi & Speranza, 2012).

Archetti and Speranza (2014), categorize matheuristic approaches into three, that are *decomposition approach*, *improvement heuristics* and *branch-and-price/column generation-based approach*. Decomposition approach is when the problem is decomposed into sub-problems, and some (or all) sub-problems are optimally solved using mathematical programming models. Whilst in the improvement heuristic, mathematical programming is used to improve the solutions found by different heuristic methods. In the last approach, the branch-and-price/column generation-based is used to speed up the convergence of the procedure of the mathematical programming models. In this thesis, a decomposition matheuristic is proposed for the IRP with stochastic demand, where a MILP is solved inside rollout algorithm scheme.

1.3.2 Backorder Decisions and Customer Satisfaction

Motivated by the successful implementation of the many-to-one IRP (in Chapter 3), the application of ABC is extended to IRP with backorder decisions. It is a norm, for the manufacturers to apply backorder policy to satisfy customer orders as much as possible. Manufacturers are willing to invest in upgrading their services to ensure customer satisfaction (especially in terms of quality and availability of the products).

Today's customers are smart and busy. They prefer good quality product(s) with dependable and excellent services, not only they are willing to wait for the product when it is out of stock, but they are also willing to pay more for it. For example, the Apple Inc's product, iPhone 7 and iPhone 7 plus were sold out in 2 days before the release date as the demand exceeds the supply. Yet customers are willing to wait for the availability of the product after some period of time. Even that Apple has announced only a limited stock available, but customers are smart enough to choose a good quality and updated product (Musil, 2016). Apple ensures their customer's satisfaction, and they have the ability to fulfill customers order. Apple provides over 500 stores worldwide that offers not only their products but also services (repairs, restores information of lost phones and also locks the phone to secure the data) for their customers (Fingas, 2018). They also have an iPhone Upgrade Program that offers lots of privileges for their members. Furthermore, Apple customer services can be contacted via Toll-free direct contact and 24/7 online customer care (via online chat or email) (Apple Inc, 2018).

Backorder decisions can occur due to several reasons such as strong demands and the productions cannot cope with the demands, and the limited vehicles with limited capacity. This is affecting the availability of the product and the backorders decisions are the best practices. There are companies and retailers that open orders for backorder item, especially in fashion industries, for example, Malaysian based online shops: https://www.christyng.com, https://www.sometime.asia/ and https://poplook.com/. The backorder decisions are studied with the aim to increase customer satisfaction.

1.3.3 Stochastic demand

In practical real-life problems, the demands are often stochastic by nature and many industries including industrial gases company, airlines, hotels, retails industry and manufacturing face uncertain demands from their clients. In addition, the uncertainty may also occur because of the limited amount of products due to the rate of production, storage capacities, shortages of raw materials, natural disasters and innovations of new technologies. Some factors can be avoided such as getting the right number of forecasted demand or the company may initiate an emergency production to avoid shortages. However, the later initiative may increase the production cost tremendously.

Some unknown demands are observed to follow some probability distribution functions such as Binomial, Poisson and Uniform distributions. For example, data that follows a Poisson distribution indicates a situation where the random data happens at a certain rate over a period of time (Niu, 2018). In a situation where historical data can be obtained, its corresponding probability distribution can be determined by distribution fitting or by using available commercial software such as Minitab (Frost, 2017). Furthermore, selecting an appropriate forecasting method is vital and this improves the supply chain with stochastic demand, and consequently enhances the customer satisfactions. Inspired by these, a study on the stochastic demands in IRP with backorder is carried out.

1.4 Objectives of the Thesis

The main objective of this thesis is to develop metaheuristic and matheuristic algorithms that is based on an Artificial Bee Colony (ABC) and the performance of these algorithms are evaluated on four variants of the IRP. The general objective of the four IRPs is to find an optimal policy, which determines how much quantities to deliver and the shipping strategy that is when to visit a customer/retailer; that minimizes the objective function, which includes the inventory holding cost at the supplier, the inventory holding cost at the customer/retailer, the transportation cost and the stock out or backorder cost. The specific objectives according to the different IRP are presented below.

Objective 1: To study a many-to-one IRP network distribution

- To propose an ABC algorithm for the problem.
- To embed mechanism that is able to accommodate both inventory and transportation.
- To find the balance between inventory holding cost and transportation cost.
- To perform statistical analysis to see the performance of the developed ABC and the compared algorithm, Scatter Search (SS) and Genetic Algorithm (GA).
- To enhance the proposed ABC algorithm.
- To observe the convergence of the enhanced ABC algorithm.

Objective 2: To study backorder decision in IRP

- To modify the ABC algorithm for IRP to handle backorder decisions.

- To develop two different exchange mechanisms that handle inventory and backorder updating.
- To perform statistical analysis to see the performance of the developed ABC and the compared algorithm, Estimated Transportation Cost Heuristics (ETCH).

Objective 3: To study stochastic demands in IRP

- To study an approximate dynamic programming approach to handle stochastic demand.
- To develop a hybrid rollout algorithm for solving the problem.
- To enhance the hybrid algorithm by developing an ABC algorithm as one of method to generate the controls.
- Test the algorithm for two discrete probability distributions, binomial and uniform.
- To compare the ratios of the enhanced algorithm with the literature.
- To analyze the controls that contributed to the best-expected costs found.
- To analyze the pattern of the number of visits, the number of delivery quantities and the stock out quantities.

Objective 4: To study backorder decisions in IRP with stochastic demands

- To extend the previous problem for backorder decisions.
- To propose a new MILP for the problem.
- To implement the enhanced hybrid rollout algorithm for solving the problem.
- Test the algorithm for demands that follow uniform probability distribution.

- To analyze the controls that contributed to the best-expected costs found.
- To analyze the pattern of the number of visits, the number of delivery quantities and the backorder quantities.

1.5 Contributions of the Thesis

This section highlights the contributions of this research. The main contributions are in the development of new algorithms and models. The ABC algorithm is utilized in the development of all algorithms and its performance and efficiency are evaluated and tested on four different IRP models.

The first contribution of this research is in the design of Artificial Bee Colony (ABC) for solving a many-to-one IRP (Chapter 3). The algorithm embeds an inventory updating mechanism and route improvement procedures. New forward and backward transfers are proposed in the inventory updating mechanism and their aims are to balance the inventory and transportation cost. The forward transfer emphasizes in reducing the inventory whilst the backward transfer attempts to reduce the transportation cost. The existing route improvement procedures from the literature such as 1-0 exchange and 2-opt are utilized to further improve the routes. The performance of the developed ABC is tested on the 14 datasets proposed in (Moin et al., 2011) and compared with two other well-known metaheuristics, Scatter Search (SS) and Genetic Algorithm (GA). Extensive nonparametric statistical tests are performed which shows that the ABC developed is significantly different and better when compared to SS and GA. The ABC algorithm is further enhanced by incorporating both the inventory updating and route improvement in the onlooker bee phase. In addition, a post-optimization using Dijkstra’s algorithm is applied at the end of the algorithm. An insight on parameters controlling the search process in ABC is also carried out. The performance of the new improved algorithms surpassed the performance of the previous ABC algorithm.

The ABC is further enhanced to consider IRP with backorder decision. This is the second contribution of this thesis (Chapter 4). Two new inventory updating mechanisms are proposed to take into account the backorder decision. The first is the randomly selected exchanges where the exchanges between two customers are selected at random. However, the feasibility must be maintained at all times. The second inventory updating mechanism takes into account the inventory holding cost and the backorder cost. Both inventory updating mechanisms are able to deal with partial inventory and backorder. Additional route improvement 2-opt* besides the existing 1-0 exchange and 2-opt is adopted. The performances of both ABCs are evaluated on benchmark datasets generated by Abdelmaguid et al. (2009). Nonparametric statistical tests and post hoc procedure are carried out on both the newly developed ABCs and the results given in Abdelmaguid et al. (2009). The finding shows that both ABC algorithms are found to be significantly different when compared to Abdelmaguid et al. (2009).

The third contribution of this research is in solving Dynamic Stochastic Inventory Routing Problem (DSIRP) (Chapter 5). The work of Bertazzi et al. (2015) is extended by introducing additional controls based on ABC in the rollout algorithm. The number of scenarios is also increased to capture more realization of demands. Performance of the algorithm is tested on two discrete probability distributions, binomial and uniform. The ratios between the obtained policy and the bounds are smaller using binomial probability distribution when compared to the Bertazzi et al. (2015). The analysis on the controls revealed that the ABC controls contributed the most when compared to other type of controls. The patterns of the number of visits and the delivery quantities are also examined to see the relation to the OU policy adopted. The uniform probability is added to be utilized in the following chapter.

The final contribution of this thesis is the extension of the DSIRP to accommodate backorder decisions (Chapter 6). Backorders are considered instead of stock out (loss sales) to create a win-win situation where we assumed that the customers agree to consider backorder. A new MILP for the problem is proposed where the formulation in Bertazzi et al. (2015) is modified and several new constraints relating to backorder are introduced. The DSIRP with backorders (DSIRPB) is modeled as a stochastic dynamic programming where the state and the dynamic system are able to handle backorders decision. The enhanced hybrid rollout algorithm and additional control where all the customers are visited are proposed. The algorithm is tested on the demand that follows uniform probability distribution. The efficiency of the algorithm is evaluated by means of the ratio of the policy to the bounds. Further analysis on the controls that contributed to the optimal policy and the number of visits, delivery quantities and backorder quantities are also carried out.

A comprehensive contribution is elaborated in Chapter 7.

1.6 Structure of the Thesis

This thesis is arranged into 7 chapters. Chapter 1 (this chapter) introduces the study that are carried out beginning with the background of the research studied, motivation and contributions of the thesis to new knowledge in Inventory Routing Problems and its variants.

Chapter 2 gives the literature survey that comprises of two parts: The literature on IRP and the second part discusses the literature on ABC. A literature survey on IRP identifies the research gap where similarities and differences between related works on IRP are identified. The definition of the IRP is introduced, and this is followed by the classification of the different types of IRP and it is divided into IRP with deterministic and stochastic demands. The second part of Chapter 2 is on the metaheuristic method studied in this thesis, Artificial Bee Colony (ABC). The definition of ABC and its underlying concept of ABC are presented. The general algorithm of ABC that comprises of three main bee phases: employed bee, onlooker bee and scout bee are also discussed. A diagram is provided to illustrate the steps of ABC algorithm. A general ABC algorithm for the IRP is then presented.

Chapter 3 describes the distribution network of the problem modeled for an automotive part supply chain where a many-to-one network consisting of a depot, multiple suppliers and an assembly plant. A capacitated homogeneous vehicles travel from the depot to collect the parts (demand is deterministic and set by the assembly plant, equivalent to vendor managed inventory settings) from suppliers and deliver to the assembly plant and back to the depot. No backordering is allowed due to the excessive cost incurred for not satisfying the demand. The descriptions and other assumption of the problem were given together with the formulation of the problem. The designated ABC algorithm for IRP is named *ABCIRP*. The details implementation of *ABCIRP* is described in details. To the best of author knowledge, this is the first implementation of ABC for IRP. An inventory updating mechanism is proposed where it takes into account both inventory management and vehicle routing. The inventory updating mechanism consists of two transfers, *forward transfer* and *backward transfer*. Two neighborhood operators 1-0 exchange and 2-opt are used to handle the vehicle routing part of the algorithm. The performance of *ABCIRP* is tested on 14 datasets created based on the existing 4 datasets given in Lee et al. (2003). Some analyses including statistical analysis are carried out to support the results obtained. Further enhancement of the *ABCIRP* (*EABCIRP)* are also examined.

The second variant of the IRP, which is the IRP with backordering decisions (IRPB) is presented in Chapter 4. There are two conditions of backordering cases considered, first is when it is economical to backorder than to make the delivery (transportation cost is high) and the second condition is when there is not enough capacity to do the delivery. The model formulation of IRPB is presented and the ABC algorithm is modified for IRPB (*ABCIRPB*) is presented, particularly the two mechanisms of inventory updating; *Random Exchange* (RX) and *Guided Exchange* (GX) which are able to handle the backorder decisions. *ABCIRPB* embeds RX and GX are referred to as *ABCRX* and *ABCGX* respectively. Both algorithms are tested on 135 datasets that are divided into three different scenarios. Each scenario describes different conditions of the backordering allowed. The condition for Scenario 1 is that, it is not beneficial to do backorder decisions, while condition for Scenario 2 and 3 are when backorder decision is more economical. Datasets for Scenario 1 and 2 consists of 5, 10 and 15 customers with 5 and 7 periods. Whilst for Scenario 3 the datasets consist of 20, 25 and 30 customers with 7 periods. Results obtained are analyzed and compared with Abdelmaguid et al. (2009). A statistical analysis is carried out to see the significant difference between the *ABCRX*, *ABCGX* and the method from literature.

The network considered in Chapter 4 is extended to solve problem where the demand is stochastic. Two characteristics of the demand are considered: (1) the demand is known in a probabilistic sense, specifically in this thesis is the binomial probability distribution; and (2) the demand is gradually revealed at the end of each period. Thus, the problem studied in Chapter 5 is coined as the Dynamic Stochastic Inventory Routing Problem (DSIRP) where a matheuristic hybrid rollout algorithm is proposed. A Mixed Integer Linear Programming (MILP) formulation for the deterministic version of DSIRP is presented where an order up-to-level (OU) inventory policy is adopted to control stock out. Enhancement of the hybrid rollout is carried out by proposing an ABC algorithm for generating additional controls. The enhanced hybrid rollout algorithm is known as *Policy* M^+. The performance of *Policy* M^+ is evaluated by comparing with the bounds and some quantitative analysis of the controls contributed to the algorithm, the number of visits and the delivery quantities are also investigated. *Policy* M^+ is tested on two discrete probability distributions that are binomial and uniform.

The DSIRP is extended to include the backorder decision (DSIRPB) and is presented in Chapter 6. A new MILP formulation for DSIRPB which differs from Solyali et al. (2012) is proposed and is described in details. A similar approach as in Chapter 5 is adopted where an enhanced hybrid rollout algorithm embedding the additional controls using ABC is utilized. The algorithm denoted as *Policy B* and its performance is tested on the same datasets as Chapter 5. However, the demands are generated using the uniform probability distribution only and the results are compared with the bounds found from the MILP. Similar analysis of the controls used, patterns of the delivery quantities and the number of visits are discussed. The backorder quantities and its pattern with respect to the number of visits are also examined.

Chapter 7 concludes the works and future research are outlined.

1.7 Summary

This chapter presented the backgrounds, the motivations and the importance of the research. The objectives, contribution and the structure of the thesis are presented where all the four variants of IRP are discussed.

CHAPTER 2: LITERATURE REVIEW

This chapter provides the related literature review for this study. The literature review is segregated into two parts. The first part gives the review of the problem considered; Inventory Routing Problem (IRP) and the presentation is subdivided into deterministic IRP and stochastic IRP. Whilst the second part details the review of the metaheuristic method used, Artificial Bee Colony (ABC). The essential components and the basic ABC are presented next, followed by the detailed ABC algorithm proposed for the IRP. Summary and research direction are given at the end of this chapter.

2.1 Inventory Routing Problem

2.1.1 Introduction

This section presents the literature review on the inventory routing problem (IRP). IRP incorporates two main components of the supply chain management, inventory management and vehicle routing problem (VRP). The combination of these two components is capable of improving the efficiency and performance of the system as compared to the conventional management where both elements are considered separately (see for example Raa and Aghezzaf (2009), Schmid et al. (2013), Archetti et al. (2007) and Archetti and Speranza (2016)).

IRP emerges in conventional and non-conventional situations such as vendor managed inventory (VMI). In VMI a vendor has the ability to make decisions about the timing and sizing of deliveries as well as the routing and this strategy ensures that customers will never run out of stocks. VMI is an established policy in many literatures and also in the industry (Andersson et al., 2010). The objective of the IRP is to determine a distribution strategy that minimizes the overall total costs that include both inventory cost and transportation cost such that the customers are satisfied.

Andersson et al. (2010) stated that Golden et al. (1984) is one of the first paper that uses the term inventory/routing problem to describe the combination of inventory management and VRP, which is then commonly used as the IRP. However, there is also other literature that explains the IRP using different term, for example Baita et al. (1998) defines IRP as dynamic routing and inventory (DRAI) problems which focus on the simultaneous presence of three aspects namely routing, inventory and dynamic, and dynamicity is defined as all the above issues (inventory and routing) are embedded in a dynamic framework, in the sense that repeated decisions have to be taken at different times within some time horizon, and earlier decisions influence later decisions. According to Baita et al. (1998), “DRAI problems deal with how to manage the activity of supplying (one or several) goods from (one or several) origins to (one or several) destinations during some (finite or infinite) time horizon whilst considering both routing and inventory issues.”

The detail IRP components determine the different variants of IRP. The detail components can be categorized into 7 different characteristics namely time period, types of demand, network topology, types of routing, inventory decisions, fleet compositions, and sizes. Table 2.1 exhibits the IRP’s detail components, which is adapted from Andersson et al. (2010) and Coelho et al. (2014b).

Table 2.1: Details of the IRP components.

Characteristic	**Alternative**			
Time Period	Instant	Finite	Infinite	
Demand	Stochastic	Deterministic		
Topology	One-to-one	One-to-many	Many-to-many	
Routing	Direct	Multiple	Continuous	
Inventory Policy	Maximum Level (ML)	Order-up-to level (OU)		
Inventory Decisions	Fixed	Stock out	Lost sales	Backorder
Fleet Composition	Homogeneous	Heterogeneous		
Fleet Size	Single	Multiple	Unconstrained	

From Table 2.1, time period or planning horizon refers to the length of time taken into consideration by the IRP models. It is divided into instant (one period), finite and infinite horizons. The demand for IRP models can be deterministic or stochastic. Since IRP has been modeled to solve a practical problem, it is common to have customers demand that is known in stochastic form. The topology of IRP varies based on the number of suppliers and customers (or retailers). The structure of one supplier serving one customer is one-to-one, while the most common structure in IRP is the one-to-many structure where a supplier serves multiple customers while the many-to-many structure consists of multiple suppliers and multiple customers. Additionally, there is a many-to-one structure in which this can be seen in an automotive parts supply distribution, where different parts (from multiple suppliers) are to be collected and sent to the assembly plant to produce a finished product.

Inventory policy refers to the replenishment strategy. The two most common replenishment strategies are maximum level (ML) policy and order-up-to level (OU) policy. In ML policy, the quantity delivered is only bounded by the customers' maximum level, while in OU policy the quantity delivered must reach the customer's maximum level. There is also another replenishment strategy proposed (Coelho & Laporte, 2015) referred to as optimised target-level (OTL). OTL defines that whenever a customer is visited, the quantity delivered reaches its variable target level.

There are four different types of inventory management decisions. In many cases, the inventory is not allowed to be zero; instead, the inventory level is fixed to a safety level. The second is the stock out where unsatisfied demand occurs, and it is considered lost sales if the demands cannot be fulfilled. The last type is the backorder decision where the demand is postponed and fulfilled later. The different types of demands are considered in this thesis.

There are several review papers on IRP that provide the different characteristics and categories of IRP, specific solution approaches (exact, heuristics, and metaheuristics) and reviews on recent trends in IRP (industrial aspects and benchmarks). Among the latest papers are Moin and Salhi (2007), Bertazzi et al. (2008), Andersson et al. (2010) and Coelho et al. (2014b). Moin and Salhi (2007) focused on road distribution of IRP, where in the literature, IRP is classified according to the time period; single-period, multi-period, and infinite horizon. The authors also discussed a brief of stochastic demand in IRP.

Bertazzi et al. (2008) discussed the different variants of IRP by modifying the example introduced in Bell et al. (1983). They investigated the trade-off between inventory holding costs and transportation cost by examining different settings of inventory capacities, inventory holding costs, vehicle capacities and continuous production.

Andersson et al. (2010) highlighted the industrial aspects of IRP where industrial applications, trends, and methodologies are discussed. The authors also emphasized on maritime IRP and discussed the difference between maritime and road-based IRP. It emphasizes that maritime IRP has no central facilities and has other uncertainties due to sea weather conditions and technical problems, while in road-based IRP is often concerned with traffics (rush hours). Similar to Moin and Salhi (2007), the IRP literature is classified according to the time horizon; instant time, finite time and infinite planning horizon.

Lastly, literature by Coelho et al. (2014b) complemented Andersson et al. (2010) where they focused on the methodological aspects of the IRP. The authors divided the literature into two parts, where the first part discussed on the publications of the deterministic IRP, the solution approach (exact and heuristic algorithms), and also the extensions (production-routing IRP, multi-period IRP, direct deliveries and transshipment). The stochastic IRP (SIRP) is focused in the second part of the paper, where different approaches are needed in solving the SIRP. The approaches: heuristics method, robust optimization approach and dynamic programming approach are discussed in details. Besides deterministic and stochastic IRP, the authors also discussed the recent trends in IRP where the demand is stochastic and dynamic such that it will be gradually revealed over time where the problem is known as Dynamic and Stochastic Inventory Routing Problem (DSIRP).

As mentioned in Chapter 1, there are four different variants of IRP considered in this thesis. The variants include different topology (many-to-one and one-to-many), different types of demands (both deterministic and stochastic), different inventory decisions (fixed, stock out and backorder) and also different inventory policies (maximum level and order-up-to level). The subsequent discussions of the IRP literature will follow the same classifications as in Coelho et al. (2014b) where it is categorized into two, that is, deterministic IRP and stochastic IRP.

2.1.2 Deterministic IRP

Deterministic IRP refers to the IRP with deterministic demand where the demand is known beforehand. The advantage of known demand information allows these decisions to be made: when to serve a customer, how much to deliver, and which customers are served in which routes. This subsection discusses on planning horizon, network topologies, and solution methodology (decomposition, heuristics, metaheuristic) of the deterministic IRP. It will also focus on multi-period planning horizon, different delivery strategy and extension of IRP: backorder decisions and production.

Chien et al. (1989) investigated a multiple period planning model IRP based on a single period approach. An inter-period inventory flow is proposed as a method to link the single period problems where in each period, the inventory information is passed to the following period. The problem is formulated as a mixed integer programming and a

Lagrangian relaxation-based heuristics is applied, where the problem is decomposed into two subproblems; inventory allocation problem and vehicle utilization. The customers demand is deterministic, and any unfulfilled demand (stock out) is penalized.

Lee et al. (2003) studied a multi-period, many-to-one IRP distribution network in an automotive supply chain industry. The network consists of multiple suppliers and a single assembly plant where each supplier provides a distinct automotive part. A fleet of capacitated vehicles is always available for deliveries to fulfill the demand specified by the assembly plant. They do not allow backorder decisions due to the expensive cost penalized by the assembly plant, and the inventory holding cost is charged for every unit of inventory. A simulated annealing approach is proposed for the problem where the problem is decomposed into two subproblems; vehicle routing and inventory control. The first variation of IRP considered in this thesis is similar to Lee et al. (2003). Their datasets are adopted and modified to suit the problem. We used a different assembly plant coordinate as the coordinate given fails to route and the route length constraints are also removed as the constructed routes could not be verified. Consequently the results from our algorithms cannot be compared with.

Bertazzi et al. (2002) also studied a multi-period IRP with a one-to-many distribution network that consists of a supplier and multiple retailers. Retailers determine its minimum and maximum inventory level for each product, and a vehicle must visit a retailer before it reaches its minimum level and the products are to be filled to its maximum level (order-up-to level (OU) policy). The authors proposed two-step heuristics algorithms where a feasible solution is constructed in the first step and then improved in the second step. They then investigated three variations in the objective functions to access the impact of each cost component; transportation costs only, inventory costs (at supplier and retailers) only and a combination of both, transportation cost and inventory cost. The objective costs obtained are not very different from each other and the combinations of both transportation and inventory costs give a slight improvement in the minimum objective value.

Delivery strategy is an important factor in achieving the reduction in total cost. Delivery strategies include direct deliveries, non-split delivery (a customer can only be served by one vehicle) and also split deliveries. Moin et al. (2011) extended the idea in Lee et al. (2003) by imposing the constraint that allows a supplier to be visited by more than one vehicle (split delivery). The idea of split delivery is to utilize a vehicle fully and attain saving in terms of transportation. The authors proposed a hybrid GA using two different representations that are binary and real representations and the algorithms are then modified to improve the underutilized vehicles. The hybrid GAs are tested on a modified set of 14 datasets provided in Lee et al. (2003). GAs with binary representations produced better solutions compared to the real representations and the hybrid GA algorithms clearly performed better in large datasets (98 suppliers) compared to CPLEX with less than 3.5% gap.

Mjirda, Jarboui, Macedo, et al. (2014) proposed a two-phase Variable Neighborhood Search (VNS) heuristic for the same problem in Moin et al. (2011). The first phase focuses on routing decisions with a zero initial inventory, and in the second phase considered the inventory updating. Seven different types of neighborhood structures are proposed including shifting, exchange, insertion, remove and replace of a supplier(s) within the same period (and also inter periods). A priority scheme is considered to estimate the cost if one unit of product is not delivered and management of inventory procedure is proposed. This VNS approach yields a lower total cost compared to Moin et al. (2011) in general but uses an extra number of vehicles.

Mjirda, Jarboui, Mladenović, et al. (2014) then further explored the applications of VNS on the same problem by proposing a general VNS (GVNS). GVNS makes use of Variable Neighborhood Descent (VND), a deterministic variant of VNS, as a local search inside the VNS scheme. Six different neighborhood structures that concern with insertion and exchange moves of routes (one or two routes) and periods (intra and inter periods) are described. GVNS produced better results than the previous literature (Mjirda, Jarboui, Macedo, et al., 2014; Moin et al., 2011) in 10 out of 14 datasets, in terms of average total cost, computational time and it also attained small standard deviation (average of 0.35).

Wong and Moin (2017) tackled a multi-product multi-period, one-to-many distribution network where split deliveries are allowed. They proposed a new swap heuristic for split customers in their two modified Ant Colony Optimization (ACO) algorithms, named ACO and ACO2 where ACO2 added 2-opt* in the route improvement. Both ACOs are tested on 14 instances with a maximum of 100 customers with 21 periods. The results showed that ACO2 is better than ACO in terms of total costs, where in both ACOs the costs are dominated by the inventory costs. A sensitivity analysis of the ACO's parameters that influences the decision policy was carried out to appropriately set the values.

Another extension of IRP is coordinated production and distribution decisions. The integrated production decisions within the IRP is known as production, inventory, and distribution routing problem (PIDRP). Lei et al. (2006) were the first to tackle PIDRP by formulating the problem as a mixed-integer program (MIP). They proposed a two-phase method for the problem, where in the first phase the MIP is solved using direct deliveries and the second phase solved the delivery consolidation problem which is to tackle the inefficiency of the direct delivery imposed in the first phase. Boudia et al. (2007) and Boudia and Prins (2009) proposed a memetic algorithm with population management and a reactive greedy randomized adaptive search procedure (GRASP) with path relinking respectively for the PIDRP.

Bard and Nananukul (2009) outlined the full model MIP formulation for the PIDRP. The distribution network of the PIDRP model consists of a single production plant and multiple customers in a finite planning horizon. The objective of the problem is to minimize the total production, inventory, and transportation costs without incurring any stock out at the customers' sites. The problem addressed is based on a production plant, multiple customers with time-varying demand, a finite planning horizon, and a fleet of homogenous vehicles. The aim of their paper is to propose a solution methodology that provides a high-quality solution in an acceptable runtime. To achieve this, the authors proposed a method centered on Tabu Search (TS) algorithm with path relinking for improving the results. High quality initial feasible solutions are generated using an allocation model in the form of a mixed integer program. The algorithm is tested on benchmark instances with a maximum of 20 customers and 20 periods. The gap between the best solution found and the bounds generated are within 20.5%. Readers can refer to Adulyasak et al. (2015) for the review of the PIDRP that focused on the formulations and solution algorithms.

Inventory decisions such as the order must be satisfied on time that is there is no backlogging or backordering allowed, however in practical situations some of the demand is allowed to be unfulfilled (stock out) or some backordering/backlogging is allowed with some penalty for the delay play an important role in the development of the IRP model as highlighted in Table 2.1. Abdelmaguid and Dessouky (2006) took into account backorder decisions and the backorder decisions are considered in two situations when there is not enough vehicle capacity to perform delivery, and the second situation is when the saving in transportation cost is higher than the backorder penalty costs by a customer.

The network distribution consists of a depot with infinite supply and a set of customers with deterministic demands. The deterministic demands considered are relatively small compared to the vehicle capacity, and the customers are located closely to each other such that a consolidated shipping strategy is appropriate. They proposed a GA in which the crossover operator and mutation operators are designed to handle partial deliveries. The GA is tested on existing benchmark instances with a maximum data set consisting of 15 customers and 7 periods. Generally, the GA produced better results compared to the CPLEX.

Abdelmaguid et al. (2009) expanded the idea in Abdelmaguid and Dessouky (2006) by introducing a constructive heuristic, Estimated Transportation Cost Heuristics (ETCH). This method is used to estimate transportation cost value for each customer in each period. The problem is formulated and decomposed into two subproblems, where the two subproblems inventory and backordering decisions are compared with the estimated transportation cost. The algorithm for estimating the transportation cost is embedded in the ETCH. An improvement heuristic is introduced to overcome some of the limitations of ETCH, where delivery amounts between periods are exchanged to allow for partial fulfillment of the demands. Larger instances with 20, 25 and 30 customers, and with a maximum of 7 periods are generated in Abdelmaguid et al. (2009) in addition to the benchmark instances presented in Abdelmaguid and Dessouky (2006). The second variation of the IRP studied in this thesis is similar to this paper. We offer a different approach where an Artificial Bee Colony (ABC) algorithm is proposed where the algorithm embeds inventory updating mechanism that is able to deal with backorder decisions as well as partial backorders.

2.1.3 Stochastic IRP

Most of the IRP problems are designed for real world applications and they are stochastic by nature. The stochasticity of IRP occurs due to the uncertainty in the demands where there is a possibility of shortages where the demand cannot be satisfied on time. Note that the unsatisfied demands can either be delayed and satisfied in the following period or it can be treated as lost sales. Hence, the solution is measured by the expected cost (also known as the approximate value function) since the demand is uncertain and it is only known in a probabilistic sense. Consequently, solving the stochastic IRP (SIRP) consist of proposing a solution policy (solution strategy or distribution policy) as opposed to the deterministic solution in the deterministic or static IRP (Coelho et al., 2014a).

The discussion on the literature of the SIRP is focused on the solution methodology (heuristics, metaheuristics, Markov decision process (MDP) and approximate dynamic programming) and the inventory decisions, either stock out or backorder.

Federgruen and Zipkin (1984) are among the earliest to study the SIRP. They modified the vehicle routing problem proposed by Fisher and Jaikumar (1981). The objective is to determine the replenishment strategy for each customer such that the transportation, inventory and shortage costs are minimized. A non-linear mixed integer programming model for the problem is proposed and then solved using generalized Benders' decomposition, which appropriately coordinates the allocation and routing decisions. For any assignment of customers to routes, the problem decomposes into a non-linear inventory allocation problem which determines the inventory and shortage costs, and a Traveling Salesman Problem (TSP) for each vehicle considered which produces the transportation costs.

Barnes-Schuster and Bassok (1997) tackled SIRP with direct shipping. The distribution network consists of a single depot, multiple retailers and the vehicle is assumed to be adequate with limited capacity. The depot coordinates the replenishment quantities, and they assumed no inventory is held at the depot. Backorder decisions are permitted but it must be satisfied eventually. Their aim is to find cost-effective policy using direct shipping strategy. Simulation results show that vehicle sizes that are close to the mean of demand produced a good strategy for the direct shipping. The authors proposed the model with infinite horizon.

Reiman et al. (1999) also studied SIRP with backordering decisions. The distribution network considered consists of a central warehouse and a set of geographically dispersed retailers where sufficient products are available to be distributed by a single capacitated vehicle using direct deliveries and predefined routes. The problem is modeled as a queueing controls problem and solved using a heavy traffic analysis and Monte Carlo simulations. The objective of the problem is to minimize the transportation costs, and inventory and backordering costs at retailer's site.

Minkoff (1993) proposed a decomposition approach named Future Value Decomposition (FVD) to solve the SIRP. The SIRP is modeled as a Markov Decision Process (MDP), where the state of the system represents the customers' inventory levels in every period, and the routing decisions navigate from one state to another. FVD involves two subproblems that are transportation allocation and delivery quantity, where the delivery quantity subproblem finds the quantity to deliver and also estimates the future state value. The authors tested the FVD algorithm for instances of up to 10 customers.

Kleywegt et al. (2002) work is motivated by their collaboration with an air producer company where the customer's demands are stochastic. They formulated the SIRP as a discrete time MDP and proposed an approximate dynamic programming approach for the problem. The network distribution consists of a single supplier, multiple customers, single product distributed using direct deliveries (one vehicle only serves one customer) and split deliveries are not allowed. Unsatisfied demand is considered as lost sales, and no backordering is allowed. The objective is to choose a policy that maximizes the expected total discounted value over an infinite time horizon. They studied the impact of number of customers, number of vehicles, and customer demand coefficient on the performance of their policies.

Kleywegt et al. (2004) extended their previous work (Kleywegt et al., 2002) by loosening the direct deliveries constraint by allowing a maximum of 3 customers served on a route. They presented the solution approach that uses decomposition and optimization to approximate the value function where the overall problem is decomposed into subproblems and solution of each subproblem is then combined to maximize the expected discount value. The policies obtained are very close to the optimal value in small instances, and they also tested their algorithm for realistic cases with approximately 20 customers.

Adelman (2004) proposed a price directed approach for SIRP, where the future costs of the current actions are obtained using optimal dual prices from two linear programming relaxations. The problem is formulated as an MDP, where the vehicle capacity constraint is removed, and unfulfilled demand is considered as lost sales. The number of customers served in a route is unbounded and unlimited number of vehicles are available to perform the routing. The price directed policy is stable as the value function is approximated using mathematical programming.

Yu, Chu, et al. (2012) studied SIRP with split deliveries where one customer can be served by more than one vehicle. The problem is known as SIRPSD. This paper is an extension of the deterministic version presented in Yu et al. (2008). SIRPSD is modeled as an approximate stochastic model, which is then simplified into a deterministic model where a Lagrangian relaxation approach is proposed for the problem. The Lagrangian relaxation approach decomposes the problem into two subproblems: inventory and vehicle routing. The inventory subproblem is solved by partial linearization whilst the minimum cost flow is used for the routing subproblem. The authors introduced the service level constraint at both the supplier and the retailers in the model in order to control the amount of stock out. The optimal value obtained from the Lagrangian relaxation approach provides the lower bound which is used to assess the feasible solution found. The algorithm is tested on 10 instances with 100 and 200 customers and 5 to 10 periods. The parameter for service level constraints is varied from 95% to 99%, where the results showed that if the level is increased, the average total cost is also increased.

Solyali et al. (2012) incorporated robustness inside the IRP, where the problem is referred to as Robust IRP (RIRP). The term robust is coined because the number of routes (after the realization of demand) must less than or equal to some threshold values. They have adapted robust optimization that ensures the feasibility of the solution for any realization of uncertain demands. Robust optimization is suitable to deal with uncertainty where no information is known regarding the parameter probability distribution. The network distribution considered consists of a supplier where a single vehicle is available to transport a single product to multiple customers over a finite horizon. Backordering decision is implicitly considered where the backorder is penalized when the total amount replenished is not enough to satisfy the demand, else inventory is recorded. They use facility location reformulation which defines the convex hull of feasible solutions of the inventory replenishment problem of each customer and a two-index vehicle flow formulation for the routing decisions. They developed a branch and cut algorithm for the RIRP, and computational experiments were performed for 450 new generated instances with the largest dataset consisting of 30 customers and 7 periods. The authors extended the formulation of Pochet and Wolsey (1988) and incorporated the robust constraint. In this thesis, a slightly different formulation combining Bertazzi et al.'s (2015) with Abdelmaguid et al.'s (2009) and also by adding some constraints.

Bertazzi et al. (2013) solved SIRP with stock out. Coelho et al. (2014b) classified the problem presented in Bertazzi et al. (2013) as dynamic and stochastic inventory routing problem (DSIRP) where dynamic is defined as the decisions made without full knowledge of future events and the knowledge is gradually unfolded over time (normally at the end of each period). The authors proposed a dynamic programming formulation where a single supplier served a set of retailers and the retailers' demand is known in a probabilistic sense. In this model, each retailer defines its maximum inventory level and a single vehicle with limited capacity is available to perform the distribution of the product over a given time horizon and stock out may occur due to the limited vehicle capacity. Order-up-to level (OU) inventory policy is adopted which ensures that whenever a retailer is visited, the quantity delivered to the retailer reaches its maximum inventory level. The objective of the problem is to determine the delivery strategy that minimizes the expected total cost that consists of the sum of the expected total inventory and stock out cost at the retailer's site, and the expected routing cost. A hybrid rollout algorithm is proposed where the algorithm utilizes the MILP for deterministic counterpart inside the rollout scheme. The demand generated follows a normal probability distribution. The algorithm is tested on instances with a maximum of 50 customers and concluded that the hybrid rollout algorithm performs better in instances with higher inventory cost.

Coelho et al. (2014a) proposed an adaptive large neighborhood search (ALNS) with reactive and proactive policies to solve DSIRP. The reactive policy observed the state of the system for making the next decision while proactive policy anticipates the future by forecasting the demand. Both policies considered emergency lateral transshipment as an effort to avoid stock out whenever customer inventory level becomes negative. The reactive policy is designed to make use of OU inventory policy while the proactive policy is able to handle both ML and OU policies. The demand generated is based on normal distribution, and historical demands are used for forecasting. ALNS is tested on 450 instances with up to a maximum of 200 customers and analyzed with different service levels. Higher service levels reduced lost demand and ensured no emergency lateral transshipment. Results showed that the use of longer rolling horizon step does not improve the solutions. The policy adopted in this thesis is different from the authors as we do not consider emergency transshipments in our policy. Moreover our approach does not make use of the forecasted data sets.

Recently, Bertazzi et al. (2015) worked on a similar distribution network as in Bertazzi et al. (2013) except that the transportation is handled by a third party (through transportation procurement). Transportation procurement indicates that whenever there is delivery, a transportation capacity is purchased. A modified MILP by eliminating the transportation constraints is proposed and used inside the hybrid rollout algorithm. The objective of their study is to minimize the total expected cost over a planning horizon which is given by the sum of inventory cost at the suppliers, the inventory cost and stock out penalty cost at the retailers and the cost of procurement of the transportation. Computational results showed that taking into account the probability demand distribution yield a better policy compared to considering the average demand only. A managerial insight of different probability distribution: Poisson, uniform and binomial distributions are also provided. We investigated a similar model in Chapter 5, where the hybrid rollout algorithm is enhanced by proposing an ABC algorithm for generating the controls as opposed to the simple heuristic in Bertazzi et al. (2015) and we also used an additional number of scenarios to improve the accuracy. The model is extended to consider the backorder decision in Chapter 6.

2.2 Artificial Bee Colony

Artificial Bee Colony (ABC) algorithm is a technique inspired by Swarm Intelligence. "Swarm Intelligence (SI) which is an Artificial Intelligence (AI) discipline, is concerned with the design of intelligent multi-agent systems by taking inspiration from the collective behavior of social insects" (Blum & Li, 2008). The term swarm refers to the collection of animals performing a collective behavior, for example, fishes, birds, and insects, so SI is defined as the collective behavior of decentralized and self-organized swarms.

Self-organization and division of labor features are the two important features in SI (Bonabeau et al., 1999). Self-organization is the interactions among individuals that exhibit simple behaviors. Bonabeau et al. (1999) detailed the self-organization using four criteria. A brief explanation of the criteria is given below.

1) Positive Feedback: simple behavior that promotes the creation of structures, for example recruitment and reinforcement.
2) Negative Feedback: counterbalance the positive feedback and helps to stabilize the collective pattern.
3) Fluctuations: random walks and random task switching. Randomness is significant for new structures which enables the discovery of a new solution.
4) Multiple Interactions: agents (individual) uses information coming from other agents so that the information spreads throughout the network.

These self-organization criteria can be seen in the foraging behavior of honey bees in the ABC algorithm. This behavior is described in Karaboga et al. (2014) where onlooker bees are recruited to explore high nectar food source (*positive feedback*), upon encountering poor food source, the exploitation process is stopped (*negative feedback*). Random search surrounding the nest is explored to find a new food source, performed by scout bee (*fluctuation*) and information exchanges regarding a food source happens between employed bees and onlooker bees (*multiple interactions*). The behavior above is further explained in subsection 2.2.3.

The second important feature of SI is the division of labor which describes the different tasks in a swarm executed simultaneously by specialized individuals.

Other examples of SI algorithms include particle swarm optimization (PSO) which is based on bird flocks (Kennedy & Eberhart, 1995) and also ant colony optimization (ACO) that is inspired by the behavior of the colonies of ants in hunting foods and their abilities to memorize and pass information to other ants (Dorigo & Blum, 2005).

The simplicity and effectiveness of the ABC algorithm have been proven. It is shown that since its proposal for numerical optimization in 2005, there are more than 300 papers published (Karaboga et al., 2014) in the same field as well as in other fields. Readers are referred to Karaboga et al. (2014) for a comprehensive survey of the ABC algorithm. Karaboga et al. (2014) focused on the different versions of the ABC (modification and hybridization) and the application of the ABC in different fields, including neural networks, engineering (industrial, mechanical, electronics and software), image processing, data mining and sensor network.

2.2.1 Literature

ABC algorithm was introduced by Karaboga (2005) for numerical optimization. This metaheuristic is inspired by the intelligent behavior of honey bees in searching for a food source (also known as nectar). The intelligent behavior refers to the bees' ability to share information of the food source found (through waggle dance) with other bees in the nest.

There are many papers that developed ABC algorithms published in numerical optimization field; it has been successfully tested on numerical benchmark test functions and compared with other metaheuristics such as GA, PSO and ACO. These have been documented in Karaboga and Basturk (2007), Karaboga and Basturk (2008), Karaboga and Akay (2009), Akay and Karaboga (2012) and D. Zhang et al. (2011). ABC is also studied for constrained numerical optimization (see for examples Mezura-Montes et al. (2010) and Karaboga and Akay (2011)), control parameters effect (Akay & Karaboga, 2009), different strategies such as parallel ABC (Subotic et al., 2010; Tsai et al., 2009), subpopulation of bees in local search (Banharnsakun et al., 2010b), multi-objective problems (Akbari et al., 2012), and finally hybridized with other metaheuristics such as GA, PSO and ACO (Duan et al., 2010; Kang et al., 2010).

Even though the ABC algorithm was initially introduced for numerical optimization problems, its implementation has been extended to combinatorial optimization problems such as capacitated vehicle routing problem (CVRP) (Szeto et al., 2011), scheduling problem (Li, Xie, et al., 2011; Pan et al., 2011; Ziarati et al., 2011), distribution systems problem (Abu-Mouti & El-Hawary, 2009; Pal et al., 2011) and knapsack problem (Sundar & Singh, 2010; Sundar et al., 2010). To the best of our knowledge, when we first started this research in 2013, there is no paper published in the application of IRP. This thesis adopts the concept proposed by Szeto et al. (2011) in VRP, which is one of the components in IRP.

Szeto et al. (2011) proposed an enhanced ABC algorithm for CVRP. CVRP is categorized as an NP-hard problem as only small instances can be optimally solved by using an exact algorithm (see Toth and Vigo (2002) and Baldacci et al. (2010)). Solutions are presented as a sequence of the customer visited by its corresponding vehicles. The authors proposed seven different neighborhood operators to improve the solutions which are random swaps, random swaps of subsequences, random insertion, random insertions of subsequences, reversing a subsequence, random swaps of reversed subsequence and random insertions of reversed subsequences. The enhancement of ABC involves two phases: onlooker bee phase and scout bee phase. A solution is replaced by the global optimum solution found (among the onlooker bees) if the solution is not improved (poor solution) for the largest number of explorations in the onlooker bee phase and in scout bee phase, the poor solution is replaced with its neighbor instead of a random solution. The performance of enhanced ABC is evaluated using benchmark instances ranging from a minimum of 50 up to 480 customers. They analyzed the convergence and *limit* parameter which balances the intensification and diversification of the search process. Results of enhanced ABC are better than Simulated Annealing (SA) and GA.

Gomez et al. (2013) proposed a different strategy of ABC for CVRP, where instead of a vector of the sequence of customers visited by vehicles (separated by a delimiter, 0), a solution is represented as the number of routes used. This is done by relaxing the number of vehicles as opposed to Szeto et al. (2011) where the best number of vehicles must be known beforehand. The ABC algorithms started with a good solution (generated using sweep algorithm) and supported by five different neighborhood operators that are 2-opt, random swaps, random multi swaps (multiple customers swaps), insertions and random swaps of different length in the employed bee phase whilst similar operators but with more intensive search are applied in the onlooker bee phase. The results obtained are worse compared to Szeto et al. (2011) for most of the instances and the ABC took longer time to converge (finding the best number of vehicles) because of the relaxed constraint on number of vehicles.

S. Zhang et al. (2014) proposed a hybrid ABC for the environmental VRP (EVRP) where the aim is to find the optimal solution that reduces the emission of carbon dioxide (CO_2). The emission of CO_2 is determined directly by fuel consumptions (traveling distance, truck load and traveling speed). The authors used the same representation as

Szeto et al. (2011) but the employed and onlooker bee phases included GA crossover operator as an addition to swap, reverse, insertion and exchange of 2 or 3 elements. Similar to Szeto et al. (2011), roulette wheel selection (RWS) method is used in the onlooker bee phase. They analyzed the *limit* and number of bees parameters and the results illustrated that the hybrid ABC outperformed both original ABC and GA in terms of cost effectiveness. It is found that the two main factors that influence CO_2 emissions are truck load and vehicle traveling distance.

Pal et al. (2011) studied an integrated procurement, production and shipment planning of a supply chain problem. This problem considered the situation where a manufacturer received raw materials (non-added value products) from suppliers, processed the materials to produce finished products (added value products), and the finished products are shipped to satisfy demand specified by the retailers. The objective of the problem is to minimize the total operations costs which comprise of the production cost of the factory, purchase cost and inventory holding cost of the manufacturer and retail centers. The authors proposed two metaheuristics PSO and ABC. The ABC algorithm adopted the ranking scheme selection for the onlooker bees to choose high quality employed bees. Both PSO and ABC only considered feasible solutions, so their algorithms constantly checked on the feasibility of the solution produced. The ABC proposed comprised of 60 total bees and the algorithm is run for 500 iterations and the number of scout bees is fixed at 10% of the total bees. The algorithms are tested on the benchmark instances with a maximum of 8 different products, 6 suppliers and 12 periods. The problem, however, differs from ours, mainly in the network distribution where they considered production and value-added process.

There are a few papers published in combinatorial optimization and none in the IRP when this research started in 2013. A comprehensive review of the ABC algorithm and its applications can be found in Karaboga et al. (2014).

2.2.2 The ABC Algorithm

This section presents the essential components of the ABC algorithm, followed by basic ABC algorithm and illustration to ease the process of understanding of this algorithm. The detailed algorithm of ABC for a general IRP is given in the next subsection 2.2.3.

2.2.2.1 The Essential Components

There are three essential components of the ABC algorithm listed in Karaboga (2005): the employed and unemployed bees and food sources.

(a) ***Food Source***

Food source represents the nectar in the bee system. The value of food source is measured by many factors such as the closeness to the beehive and the volume and concentration of the nectar. In the ABC algorithm, the food source is the solution in the solution space and the solution can be measured by the fitness function of the optimization problem.

(b) ***Employed Bees***

Each employed bee is assigned to a food source. Then the employed bee begins exploiting the food source and storing information regarding the location and quality of the food source. There are several choices of actions for the employed bee to take:

i. Employed bee abandons the food source if the food source is no longer profitable or the nectar amount has exhausted.

ii. Employed bee continues to forage at the same food source without sharing its information with onlooker bee if the nectar amount is sufficient.

iii. Employed bee goes to the dancing area and performs dancing (waggle dance) to share the information about the food source found with the onlooker bees waiting in the hive.

(c) ***Unemployed Bee***

Unemployed bee is the bee that constantly searches for a food source to exploit. There are two types of unemployed bee: onlooker bee and scout bee. The first type, the onlooker bee is the bee that awaits in the dancing area and watches the *waggle dance* of the employed bees before making the decision to choose a promising food source. The communications (information exchange) between bees related to the food sources happen here (in bee hive). The second type of unemployed bee is the scout bee. Scout bee is the bee that performs a random search at the area surrounding the hive.

2.2.2.2 General ABC Algorithm

As mentioned in the introduction/background the essential components of ABC are powered by three unique type of bees which are employed bees, onlooker bees and scout bees and each type is assigned with different important tasks. The steps of the ABC algorithm described in Karaboga (2005) is given in Figure 2.1 below.

Initiate the scout bees onto the initial food sources

REPEAT

> Randomly assign the employed bees to the food sources and determine their nectar value.
>
> Calculate the probability value of the sources with which they are preferred by the onlooker bees.
>
> Assign the onlooker bees to the food sources and determine their nectar amounts.
>
> Stop the exploitation process of the sources exhausted by the bees.
>
> Randomly assign the scout bees into the search area to discover new food sources, randomly.
>
> Memorize the best food source found so far.

UNTIL (requirements are met)

Figure 2.1: Steps of ABC.

2.2.3 An Illustration Example of the ABC Algorithm

Figure 2.2 illustrates the process of foraging for food sources and describes the movement of the bees in the process.

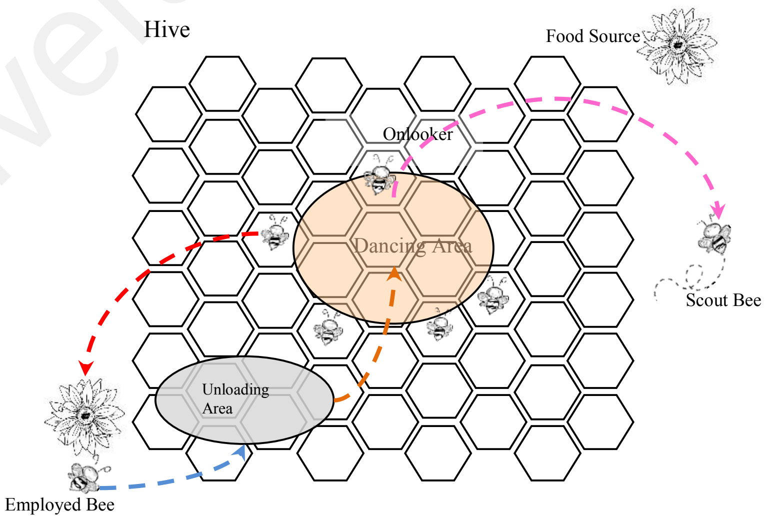


Figure 2.2: The illustration of the bee foraging behaviour.

The illustration in Baykasoğlu et al. (2007) is enhanced to better explain the algorithm where the interaction between different phases of the algorithm is clearly described through the illustration. Note that the important components in the hive are the food source (flower), employed bee, onlooker bee, scout bee and dancing area (shaded in orange). The ABC algorithm starts with the employed bee with its associated food sources. The employed bee goes to the unloading area to unload the nectar (represented by the blue arrow) it then proceeds to the dancing area and shares the nectar's information with the onlooker bees waiting in the area (represented by the orange arrow). The information sharing is done by performing the waggle dance. The onlooker bee(s) recruited begins to forage at the neighboring area of the selected food source (represented by the red arrow). Once the employed bee reaches the exploration limit, it abandons the food source and becomes a scout bee (represented by the pink arrow). The search is again repeated with the scout bee searching for a new food source. Note that the number of employed bees and onlooker bees depends on a pre-determined number.

2.2.4 ABC Algorithm for IRP

This subsection explains the detailed of the algorithm designed for IRP. Starting with the initialization phase where random solutions are generated as food sources and its fitness value is calculated. Each food source is assigned to an employed bee. The first iteration starts with the employed bee phase and each employed bee tries to improve its solution by exploring the neighborhood of the current food source. This is achieved through a neighborhood operator. The fitness value of new found food source (solution) is evaluated and if the fitness value of the new food source is better than the old food source new food source replaces the old food source.

The employed bee shares this information with the onlooker bees waiting in the hive. This is known as the onlooker phase. The information is shared through *waggle dance* performed by the employed bee. Each onlooker bee makes a decision either to follow the selected employed bee or not, based on the selection method. Generally, the roulette wheel selection method is adopted (see for example Szeto et al. (2011) and S. Zhang et al. (2014)) but in this study, the Stochastic Universal Sampling (SUS) (Baker, 1987) is chosen because it is known that SUS is not biased with minimum spread. The selected employed bee is the one with a promising food source. Note that one employed bee may be followed by more than one onlooker bee or may not be followed at all. The onlooker bee then explores the neighborhood of the selected food source using the neighborhood operator. The best food source among all the new food sources found near the old source is determined. If the fitness value of the best found food source is better than the old food source, the new food source replaces the old food source.

The last phase is the scout bee phase. This phase controls the exploitation and exploration process. If the value of the food source has not been improved after a limited number of successive iterations, the employed bee assigned to the food source will abandon the food source. The employed bee will become a scout bee and begin searching for a new food source using a neighborhood operator. After the scout bee found a new food source, it will become an employed bee again. The whole process is repeated until the stopping condition set is met.

The details of each phase of the ABC algorithm can be further explained as below.

STEP 1 Initialize the population of randomly generated solution (food sources) as x_i for $i = 1, \dots, N$. Calculate the fitness value of the population, $f(x_i)$. Assign each food source with an employed bee.

STEP 2 Set the maximum value for exploration LIMIT parameter and set l_i as counter for the parameter.

STEP 3 Set the $iteration = 1$. Repeat all phases below until its stopping condition is reached (maximum number of iteration or maximum CPU time).

3.1 Employed Bee Phase

i. Employed bee exploits the food source, x_i by applying a neighborhood operator. Denote the new food source as x_i'.

ii. Evaluate the fitness value of the new food source, (x_i') .

iii. If the new food source is better, replace the old food source, $x_i \leftarrow x_i'$. Reset counter i, l_i =0. Otherwise, increase the limit counter $l_i = l_i + 1$.

3.2 Onlooker Bee Phase

i. Open empty space $G_i = \{\}$, for set of neighbors of the food source, x_i.

ii. Assign a food source x_i to each onlooker bee using a selection method.

iii. Each onlooker bee exploits the selected food source x_i by applying a neighborhood operator. Denote the new produced food source as $\hat{x}_i$. Evaluate $\hat{x}_i$, $f(\hat{x}_i)$.

iv. Keep all the new produce food source in $G_i = G_i \cup \hat{x}_i$.

v. For each non-empty G_i, find the best fitness value. Denote the best of $f(\hat{x}_i)$ from set G_i as $f(x_i")$ such that $x_i"$ is the corresponding food source.

vi. If $f(x_i")$ is better than $f(x_i)$, replace food source x_i with $x_i"$. Reset counter i, l_i =0. Otherwise, increase the limit counter $l_i = l_i + 1$.

3.3 Scout Bee Phase

i. For each of the food source, x_i, check the limit counter l_i.

ii. If the limit counter l_i reaches the LIMIT, employed bee abandons the food source x_i and becomes a scout bee.

iii. Scout bee starts searching for new food source by using neighborhood operator. The new found food source is denoted as x_i.

iv. The scout bee turns into an employed bee again.

i. Keep the best food source found so far.

ii. $iteration = iteration + 1$.

STEP 4 Output: The best food source found so far.

One of the interesting features of ABC is that it offers the flexibility to adapt to different type of problems. It allows to manipulate and find a balance between the process of intensification and diversification. Intensification is the exploitation process where a solution is further improved by accumulated search, while diversification is the exploration process of a solution space and identifying high quality solutions areas. The employed bee and onlooker bee phases also embed neighborhood operators that exploit solutions (food sources) until they are exhausted. In addition the scout bee introduces a new solution after a solution is exhausted. Furthermore, the algorithm starts with a number of bees, which are scattered in the solution space that ensures to cover all the promising regions.

2.3 Research Gaps

In the deterministic problem, the IRP and the IRP with backorder decision are studied in this thesis. In IRP, the ABC algorithm is developed for many-to-one network and modified from the basic ABC to address the different subproblems: inventory control and vehicle routing problem.

The second addresses the IRP with backorder decision where we consider the backorder occurs in two situations: the backorder is more economical in term of cost and the restriction on limited number vehicles and its capacity (as in Abdelmaguid et al. (2009)). We proposed different methodology in our algorithm which embeds giant tour and two inventory updating mechanism.

We extended the work in Bertazzi et al. (2015) by enhancing the rollout algorithm where the ABC algorithm is proposed to generate a set of control instead of heuristic proposed in their paper. The idea in Chapter 5 is extended by incorporating backorder decision in Chapter 6. We proposed a new dynamic programming formulation where several new constraints are introduced. The formulation differs from the formulation proposed in Solyali et al. (2012) because of the different methodology adopted. Additional control in addition to ABC control is also considered. We also tested on a larger number of scenarios compared to Bertazzi et al. (2015), to emphasis on the strength of our algorithm.

2.4 Summary of the Chapter and Research Direction

This chapter provides the literature review for the problem considered, Inventory Routing Problem (IRP) and its variants, and the metaheuristic method proposed, Artificial Bee Colony (ABC) which falls into the Swarm Intelligence category. As mentioned in subsection 2.1.2 and 2.1.3, the customer's demand can be divided into two, deterministic and stochastic. Both types of demands are studied in this thesis with the variety of the network topologies, inventory decisions (stock out and backorder) and inventory policies (maximum level (ML) and order-up-to level (OU). ABC algorithm is a recent metaheuristic where it was originally proposed for numerical optimization (Karaboga, 2005). The flexibility and robustness of ABC are proven due to its success in various applications of numerical optimization (Karaboga et al., 2014). This has attracted many researchers in the combinatorial optimization field where a notable contribution is in the capacitated vehicle routing problem (CVRP) by Szeto et al. (2011). However, to the best of our knowledge, there is no paper published in the IRP thus far. This inspired and motivated us to study and proposed ABC for IRP and its variants.

CHAPTER 3: INVENTORY ROUTING PROBLEM

This chapter presents the first main contribution of the research, where Artificial Bee Colony (ABC) is designed to solve a many-to-one inventory routing problem (IRP). This chapter starts with the description of the problem, followed by its assumptions and formulations. The design of ABC for solving IRP named *ABCIRP* is discussed next; where the details of the phases of ABC that are proposed to incorporate both inventory and transportation are explained with examples. *ABCIRP* is implemented to an existing benchmark dataset, and the results obtained are compared with other metaheuristics. A proper comparison between the metaheuristics is checked by conducting several statistical tests. Finally, an enhancement for *ABCIRP* is proposed. The enhanced algorithm is denoted as *EABCIRP*. *EABCIRP* proven to be a better strategy for solving IRP.

3.1 Problem Description

The inventory routing problem considered in this research is defined on a finite planning horizon, τ where the inbound $N \times 1$ distribution network consists of a depot, 0, an assembly plant, P and a set of suppliers, $S = \{1,2, \dots, N\}$. Each supplier provides a specific product to the assembly plant, where the assembly plant determines the demand quantity for each product. A fleet of capacitated vehicles is available, housed at the depot to transport the products from the suppliers to the assembly plant, and then return to the depot. No backordering/backlogging allowed due to the high penalty cost for not fulfilling the order. However, if product's quantity collected is more than the demand in that period, then inventory was carried forward, subject to product specific holding cost incurred at the assembly plant. Inventory cost at the suppliers was not considered and the product was assumed are ready to be pickup when the vehicle arrives. Note that split delivery is not allowed, so the pickup quantity should be less or equal to the vehicle capacity. The objective is to find an optimal solution that minimizes both inventory and transportation costs over the finite horizon.

Figure 3.1 illustrates an example of the problem with the case of $N = 5$ suppliers for $\tau = 2$ periods, with vehicle capacity $C = 10$. Here, assume that the locations of depot, assembly plant and suppliers are as in the figure. Given that the supplier's demand in period 1 and 2 are $\langle 2,2,3,3,4 \rangle$ and $\langle 2,2,4,4,2 \rangle$ respectively. The routes are from depot (0) to supplier 1, assembly plant (P) and then back to the depot. Best routes in period 1 are $0 \to 1 \to 3 \to 4 \to P \to 0$ (vehicle 1) and $0 \to 2 \to 5 \to P \to 0$ (vehicle 2). The capacity for vehicle 1 is $4 + 3 + 3 = 10$ and $4 + 6 = 10$. Note that collection in period 1 for supplier 1, 2 and 5 includes the demand in period 2. Therefore, remaining only suppliers 3 and 4 need to be visited in period 2. Savings are in the transportation cost (distance and capacity).

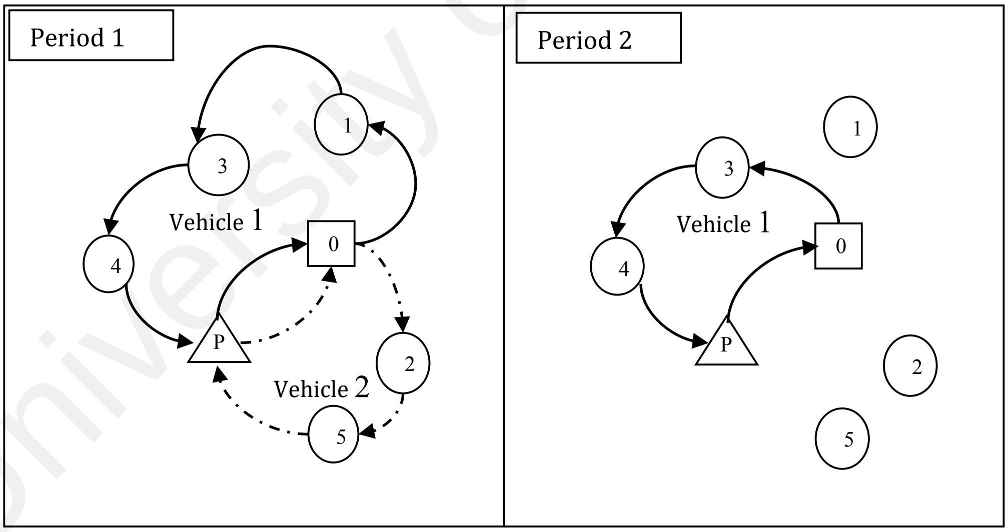


Figure 3.1: An example of $N = 5 \times 1$ distribution network for $\tau = 2$ periods.

3.2 Assumptions

There are a few assumptions made in the inventory model. The assumptions are presented as below.

a. Backordering is not allowed.

b. Split delivery is not allowed. Supplier cannot be visited by more than one vehicle.
c. Quantity collected must not exceed the vehicle's maximum capacity.
d. No holding cost penalized at the supplier site.
e. Holding costs for each part type are product specific incurred at the assembly plant and parts are assumed to be ready when the vehicle arrives.
f. Unlimited number of capacitated vehicles available at the depot. However, the algorithm can easily be modified if maximum number of vehicles is imposed.
g. Vehicle travels from the depot to the suppliers then to the assembly plant and then return back to the depot.
h. No maximum route length constraints are considered.
i. The locations of the depot, assembly plant, suppliers are given.
j. Planning horizon is finite.

3.3 Problem Formulation

Notations

The notations used in the formulation are introduced as below.

Indices

$S = \{1,2,\ldots,N\}$	A set of suppliers where supplier i $(i \in S)$ supplies product i only.
$D = \{0\}$	Depot
$P = \{N+1\}$	Assembly plant
$\tau = \{1,2,\ldots,T\}$	Period index

Parameters

C	Vehicles capacity

F	Fixed vehicle cost per trip (assumed to be the same for all periods)
V	Travel cost per unit distance
M	Size of the vehicle fleet and it is assumed to be ∞ (unlimited)
d_{it}	Demand for product from supplier i (at the Assembly Plant) in period t
c_{ij}	Travel distance between supplier i and j where $c_{ij} = c_{ji}$ and the triangle inequality, $c_{ik} + c_{kj} \geq c_{ij}$, holds for any i, j, k with $i \neq j, k \neq i$ and $k \neq j$
h_i	Inventory carrying cost at the Assembly Plant for product from supplier i per unit product per unit time
I_{i0}	Initial inventory level of product from supplier i (at the Assembly Plant) at the beginning of period 1

Variables

a_{it}	Amount of picked-up at supplier i in period t
I_{it}	Inventory level of product from supplier i at the assembly plant at the end of period t
q_{ijt}	Quantity transported through the directed arc (i, j) in period t
x_{ijt}	Number of times that the directed arc (i, j) is visited by vehicles in period t

Formulation

$\min Z$

$$= \underbrace{\sum_{i \in S} h_i \left(\sum_{t \in \tau} I_{it} \right)}_{(A)} + V \underbrace{\left(\sum_{\substack{j \in S \\ j \neq i}} \sum_{i \in S \cup D} c_{ij} \left(\sum_{t \in \tau} x_{ijt} \right) + \sum_{i \epsilon S} c_{i,N+1} \left(\sum_{t \in \tau} x_{i,N+1,t} \right) \right)}_{(B)}$$

$$+ \underbrace{\left(F + c_{N+1,0}\right) \sum_{t \in \tau} \sum_{i \in S} x_{0it}}_{(C)} \tag{3.1}$$

Subject to:

$$I_{it} = I_{i,t-1} + a_{it} - d_{it}, \quad \forall i \in S, \forall t \in \tau \tag{3.2}$$

$$\sum_{\substack{i \in S \cup D \\ i \neq j}} q_{ijt} + a_{jt} = \sum_{\substack{i \in S \cup P \\ i \neq j}} q_{ijt}, \quad \forall j \in S, \forall t \in \tau \tag{3.3}$$

$$\sum_{i \in S} q_{i,N+1,t} = \sum_{i \in S} a_{it}, \quad \forall t \in \tau \tag{3.4}$$

$$\sum_{\substack{i \in S \cup D \\ i \neq j}} x_{ijt} = \sum_{\substack{i \in S \cup P \\ i \neq j}} x_{jit}, \quad \forall j \in S, \forall t \in \tau \tag{3.5}$$

$$\sum_{j \in S} x_{ijt} = \sum_{j \in S} x_{jkt}, \quad \forall i \in D, k \in P, \forall t \in \tau \tag{3.6}$$

$$q_{ijt} \leq C, \quad \forall i \in S, \forall j \in S \cup P, i \neq j, \forall t \in \tau \tag{3.7}$$

$$I_{it} \geq 0, \quad \forall i \in S, \forall t \in \tau \tag{3.8}$$

$$a_{it} \geq 0, \quad \forall i \in S, \forall t \in \tau \tag{3.9}$$

$$x_{ijt} \in \{0,1\}, \quad \forall i, j \in S, \forall t \in \tau \tag{3.10}$$

$$x_{0jt} \geq 0 \text{ and integer}, \forall j \in S, \forall t \in \tau \tag{3.11}$$

$$x_{i,N+1,t} \geq 0 \text{ and integer}, \forall i \in S, \forall t \in \tau \tag{3.12}$$

$$x_{ijt} = 0, \quad i \in D, j \cup P, \forall t \in \tau \tag{3.13}$$

$$x_{ijt} = 0, \quad i \in S, j \cup D, \forall t \in \tau \tag{3.14}$$

$$x_{ijt} = 0, \quad i \in P, j \cup S, \forall t \in \tau \tag{3.15}$$

$$q_{ijt} \geq 0, \quad \forall i \in S, \forall j \in S \cup P, \forall t \in \tau \tag{3.16}$$

$$q_{0it} = 0, \quad \forall i \in S, \forall t \in \tau \tag{3.17}$$

The objective function (3.1) makes up both the inventory costs (A) and the transportation costs (variable travel costs (B) and vehicle fixed cost (C)). Observed that the fixed transportation cost consists of the fixed cost incurred per trip and the constant cost of vehicles returning to the depot from the assembly plant. The number of trips in period t is $\sum_{i \in S} x_{0it.}$ Constraint (3.2) is the inventory balance equation for each product at the assembly plant. The product flow conservation equations given by constraint (3.3) to ensure the flow balance at each supplier and eliminating all subtours. The accumulative picked-up quantities at the assembly plant is constrained in (3.4). Constraints (3.5) and (3.6) are to make sure that the number of vehicles leaving a supplier, assembly plant or the depot equals to the number of its arrival vehicles. Constraint (3.6) is to guarantee that vehicle travel to the plant before turning back to the depot. Constraint (3.7) indicates that the total pick-up quantity at time $t \in \tau$ does not exceed vehicle's capacity, C. Constraint (3.8) guarantees that the demand is satisfied without backorder. Constraints (3.13) – (3.15) are to assure no direct link from the depot to the assembly plant, from the supplier to the depot and from assembly plant to the suppliers, respectively. The remaining constraints are the non-negativity constraints. Initial inventories for all products are considered to be zero (i.e.$I_{i0} = 0, \forall i$). However, the algorithm can be easily modified to adapt non-zero initial inventories. The formulation is modified from Moin et al. (2011) by removing the constraints relating to the split vehicles.

3.4 Artificial Bee Colony Algorithm (ABC)

As mentioned in the previous chapter, Szeto et al. (2011) was the first to adapt ABC in combinatorial optimization specifically for capacitated VRP (CVRP), where the sequences of which customers to visit is determined such that the routing cost is minimized.

The ABC algorithm proposed in this study is quite different from Szeto et al. (2011) as IRP involves both vehicle routing and inventory management. The differences include the solution representation, neighborhood operators used and the selection method. The solution representation determines the pickup quantity together with the inventory units and we proposed neighborhood operators for each component (routing and inventory management). The selection method employed is a Stochastic Universal Sampling (SUS) method (Baker, 1987) instead of Roulette Wheel Selection (RWS) as SUS has zero bias with minimum spread.

The important phases of the ABC algorithm are the initialization phase, employed bee phase, onlooker bee phase and the scout bee phase. These basic steps are elaborated in the previous chapter in section 2.2.4. Next section discusses in details the development of ABC for this many-to-one IRP.

3.4.1 Artificial Bee Colony Algorithm for IRP (*ABCIRP)*

This subsection explains the detail steps in each of the phases that contribute to the ABC model for IRP, *ABCIRP*. Phases of *ABCIRP* presented in subsection 3.4.2 and 3.4.3 provide the details steps which highlighted the connection between different phases.

The initial food sources (initial solution) generated containing both routing and also inventory. A Giant Tour Procedure is executed to obtain the routing of the initial solution and they are improved using a simple pre-optimization procedure. The initialization phase is given in **STEP 1** whilst the declaration of all the parameters involved in the intensification and diversification procedures are presented in **STEP 2**. **STEP 3** is dedicated to the 'bee' phases: where **STEP 3.1** present the employed bee phase. Here, an inventory updating mechanism is introduced, a backward and forward transfers and their examples are illustrated in subsection 3.4.3. Once the employed bee performed the waggle dance, the onlooker bees select the best food source using the Stochastic Universal Sampling (SUS) method (Baker, 1987) which is known to have zero bias and minimum spread, this is given in **STEP 3.2**. Once the selection is made, the route is further improved using a combination of 1-0 exchange and 2-opt (Lin, 1965). Once the limit that controls the exploitation of the food source is reached, the current food sources are abandoned and replaced by a randomly generated food source. This is the scout bee phase, documented as **STEP 3.3**. The *ABCIRP* algorithm is given as follows:

STEP 1 Initialization Phase

1.1 Generate randomly n number of solutions (food sources). Each solutions indicate to visit or not visit supplier for each of the period, which from these the pickup quantity is obtained for supplier i at period t; q_{it}. Then, do preprocessing by eliminating the pickup quantity that exceeds vehicle capacity ($q_{it} > C$), as split shipment is not allowed.

1.2 Denote each food source as y_i, $i = 1, \dots, n$. Construct the initial tour for each food source, y_i by implementing the Giant Tour Procedure (Imran et al., 2009), in each period. Evaluate the fitness value for each food source; $f(y_i), i = 1, \dots, n$.

1.3 Do pre-optimization with 2-opt* (Potvin & Rousseau, 1995) and 2-opt (Lin, 1965) for each of the food sources. Assign each employed bee to a food source.

STEP 2 Set $iteration = 0$ and $l_1 = l_2 = \cdots = l_n = 0$ Declare the value of $LIMIT$ (control of exploiting a food source) and $MAXITER$, the maximum number of iterations.

STEP 3 Repeat the following until the stopping condition, $MAXITER$ is met.

3.1 Employed Bee Phase (Inventory Updating Mechanism)

a. For each food source, y_i. Select randomly a visited supplier from a random period t. Apply either forward or backward transfer based on these conditions:

i. *If* the same supplier is not visited at period $t + 1$; do forward transfer up to 3 succeeding periods.

Elseif the same supplier is not visited or visited at period $t - 1$; do backward transfer up to 2 preceding periods.

ii. Assign the new food source found, as $\bar{y}_i$.

b. *If* $f(\bar{y}_i) < f(y_i)$; replace the old food source with a new food source, $y_i \leftarrow \bar{y}_i$ and set $l_i = 0$. *Else* set $l_i = l_i + 1$.

3.2 Onlooker Bee Phase (Route Improvement Mechanism)

a. Set $G_i = \emptyset$, $i = 1, \ldots, n$, where G_i is the set of neighbor solutions of food source i.

b. For each onlookers.

i. Select a food source, y_i, using a stochastic universal sampling (SUS) selection method (Baker, 1987).

ii. Apply a neighborhood operator, $1 - 0$ exchange and $2 - opt$ on selected y_i; resulting $\tilde{y}_i$.

iii. $G_i = G_i \cup \tilde{y}_i$.

c. For each food source y_i and $G_i \neq 0$.

i. Set $\hat{y}_i \in argmin_{\tilde{y} \epsilon G_i} f(\tilde{y})$.

ii. *If* $f(\hat{y}_i) < f(y_i)$; replace the old food source with the new one; $y_i \leftarrow \hat{y}_i$ and set $l_i = 0$. *Else* set $l_i = l_i + 1$.

3.3 Scout Bee Phase

For each food source, y_i. *If* $l_i = LIMIT$, replace y_i with a randomly generated solution.

$iteration = iteration + 1$.

STEP 4 Output is the best food source found so far.

All the steps are summarized as a flow chart in Figure 3.2.

3.4.2 Initialization Phase (STEP 1)

3.4.2.1 Solution Representation

The problem is represented in a matrix form where the rows represent the number of suppliers and the columns represent the number of periods. We note that the amount to be collected depends on whether there will be a collection in the subsequent period or not. Since backordering is not allowed, the total collection from supplier i in period t is the sum of all the demands in period $t, t+1, \cdots, k-1$ where the next collection will be made in period k.

As an example we consider the problem with 5 suppliers and 5 periods where the demand matrix is given in Figure 3.3 (a) and Figure 3.3 (b) shows the representation where 1 represents that the supplier is visited in that period, 0 otherwise. Figure 3.3 (c) shows the corresponding collection matrix. Figure 3.3 (d) shows the inventory matrix and note that supplier 1 and 3 are not visited in period 2 and 3 respectively, hence the demands for them will be collected in earlier periods and this resulted in inventories.

We note that the initial inventory I_{i0} for $i = 1,2,\cdots,N$ is assumed to be zero, then the values in the first column consist of all ones. However, the algorithm can be adjusted accordingly if the initial inventory for part i is given or known in advance. If all the initial inventories exceed the first period demands for all the suppliers, then the first column in the matrix can be generated randomly. From Figure 3.3, solution representation in period 1 for suppliers $\{1,2,3,4,5\}$ and their corresponding collection quantities are $\{2,1,4,3,3\}$. Note that the cost includes the 2 units holding cost for supplier 1 and supplier 3.

3.4.2.2 Giant Tour Procedure (STEP 1.2)

Once the delivery quantity for each customer is determined, the Giant tour procedure is executed to obtain the routing of the suppliers. The procedure of giant tour entails constructing a cost network considering the location of the suppliers, vehicle capacity constraint and vehicle unit variable and fixed costs. Since the fixed cost does not vary from period to period, it is ignored from the calculation. The giant tour starts at the depot and ends at the assembly plant. Hence the cost network given in Imran et al. (2009) is modified to accommodate this situation. The travel from the assembly plant to the depot is fixed and it is included in the fixed cost. We note that the fixed cost comprises of vehicle fixed cost (every time the vehicle is initiated) and the distance from the assembly plant to the depot.

The cost network connects the nearest supplier to the depot and then find the next closest supplier to continue the connection. This process continues until all suppliers are connected to form a giant tour. The cost network is constructed as follows:

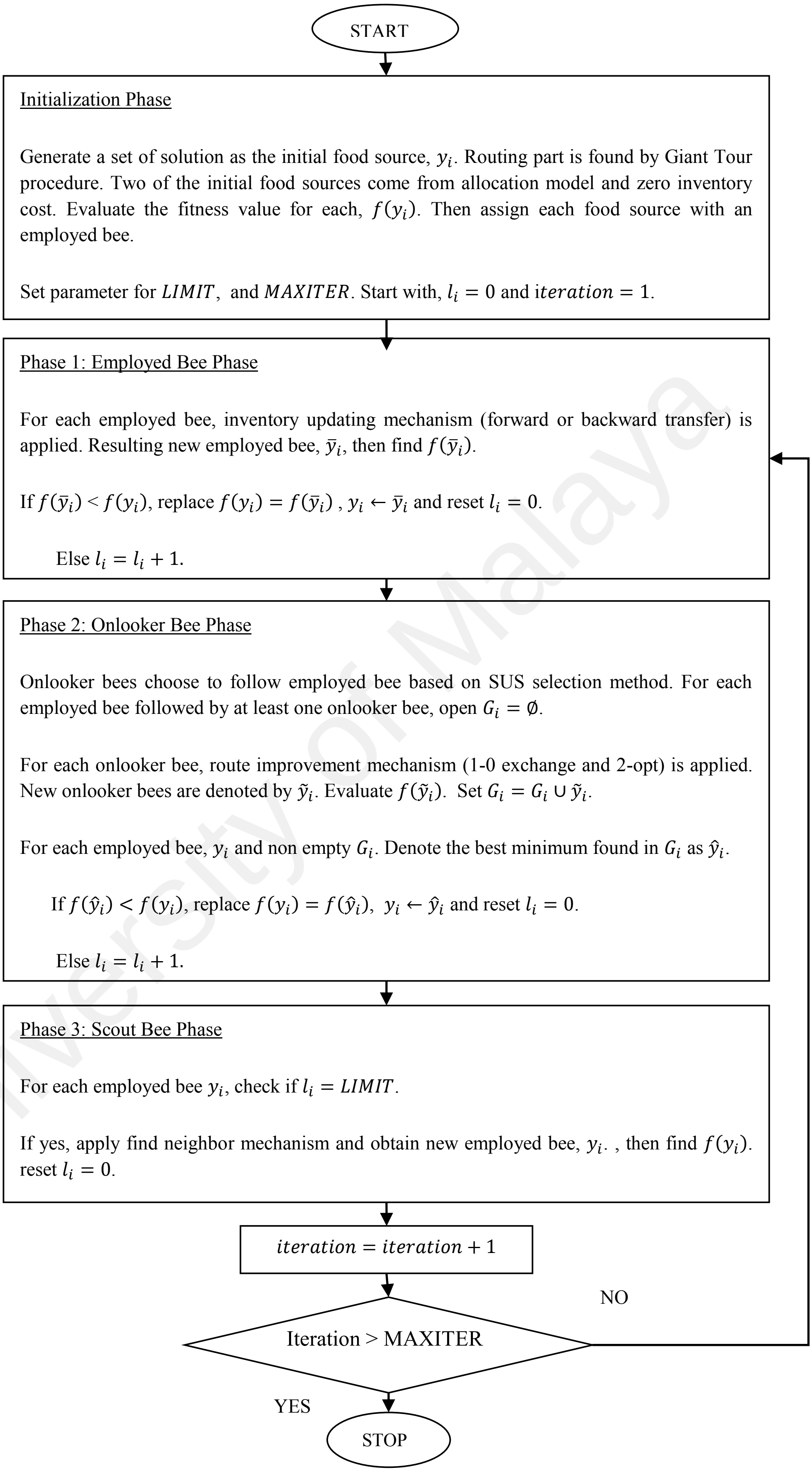


Figure 3.2: Flow Chart of ABCIRP.

	Period			
Supplier	**1**	**2**	**3**	**4**
1	2	2	2	2
2	1	1	1	1
3	2	2	2	2
4	3	1	1	1
5	3	3	2	2

Figure 3.3 (a): Demand Matrix.

	Period			
Supplier	**1**	**2**	**3**	**4**
1	1	1	0	1
2	1	1	1	1
3	1	0	1	1
4	1	1	1	1
5	1	1	1	1

Figure 3.3 (b): Binary Matrix.

	Period			
Supplier	**1**	**2**	**3**	**4**
1	2	4	0	2
2	1	1	1	1
3	4	0	2	2
4	3	1	1	1
5	3	3	2	2

Figure 3.3 (c): Collection Matrix.

	Period			
Supplier	**1**	**2**	**3**	**4**
1	0	2	0	0
2	0	0	0	0
3	2	0	0	0
4	0	0	0	0
5	0	0	0	0

Figure 3.3 (d): Inventory Matrix.

Figure 3.3: Solution Representation of *ABCIRP*.

- Each node corresponds to the last supplier of a feasible route.
- Each arc denotes a feasible route that serves all suppliers situated between the depot, the first and the last city of this feasible route, and then to the assembly plant.
- The distance on the arc is the least cost (fixed and running cost) of serving that feasible route. The running cost includes the cumulative mileage cost of going from the depot to the first supplier of this route, through all suppliers of this route and then to the assembly plant. As mentioned above the fixed cost can be removed from the calculation.
- Then, Dijkstra's algorithm (Dijkstra, 1959) is applied to optimally partition this directed and acyclic cost network to yield the shortest path. The path with the shortest total cost (length of the path) is selected. The cost network is illustrated in Figure 3.4.

(a) ***An Illustration Example on How to Construct the Cost Network.***

We consider 5 suppliers with the following giant tour sequence $S = \{1,2,3,4,5\}$ and the corresponding pickup amount, $q = \{2,1,4,3,3\}$. Let c_{ij} be the distance between supplier i and supplier j, c_{0i} be the distance between the depot and supplier i and c_{ip} is the distance between the supplier i and the assembly plant. Here, assume that the maximum vehicle capacity is 10 units. Figure 3.4 illustrates the formation of the cost network.

Start by finding the cost of arc $0 - 1$ (represented by solid line); where the cost from depot to supplier 1 is calculated, and from this supplier to the assembly plant, and finally from assembly plant back to the depot. The total cost on the arc is $AC_{01} = V(c_{01} + c_{1P})$ where V is the travel cost per unit distance. Note that the cost of assembly plant to depot, c_{P0} is omitted as the cost is fixed for all routes.

Then, it is followed by the construction of arc $0 - 2$ as the total pickup of supplier 1 and supplier 2 does not violate the vehicle capacity. The current vehicle load is 3 units. The cost of this arc is given by $AC_{02} = V(c_{01} + c_{12} + c_{2P})$. Then, continue to construct the cost network until the capacity constraint is violated and initiate a new vehicle. In this example, the cost network construction stops until arc $0 - 4$ with current vehicle load is 10 units. The cost of this arc is given by $AC_{04} = V(c_{01} + c_{12} + c_{23} + c_{34} + c_{4P})$.

The construction of the first route is complete, thus we initiate the second route by considering the arc $\mathbf{1 - 2}$ (represented by dashed lines) which denotes the cost from the depot to supplier 2, and from this supplier to the assembly plant, expressed as $\boldsymbol{AC_{12}} = \boldsymbol{V(c_{02} + c_{2P})}$. In this example, this route can be extended until arc $\mathbf{1 - 4}$ with a total vehicle load of 8 units since the inclusion of arc $\mathbf{1 - 5}$ violates the capacity constraint. Then start a new route construction again with arc $\mathbf{2 - 3}$ (represented by dotted lines);

which denotes the cost of depot to supplier **3**, and from this supplier to the assembly plant. The process is continued until there is no more arcs connecting the last supplier in the giant tour. In general the cost of arc $\boldsymbol{ij}$ can be defined as $\boldsymbol{AC_{ij} = V\left(c_{0,i+1} + \sum_{k=i+1}^{j-1} c_{k,k+1} + c_{j,P}\right)}$. After generating this cost network whose origin is the depot **0** and the destination is the last node in the giant tour, Dijkstra's algorithm (Dijkstra, 1959) is applied to determine the least cost path from the origin to the destination. Note that Dynamic Programming can also be applied to this directed acyclic network.

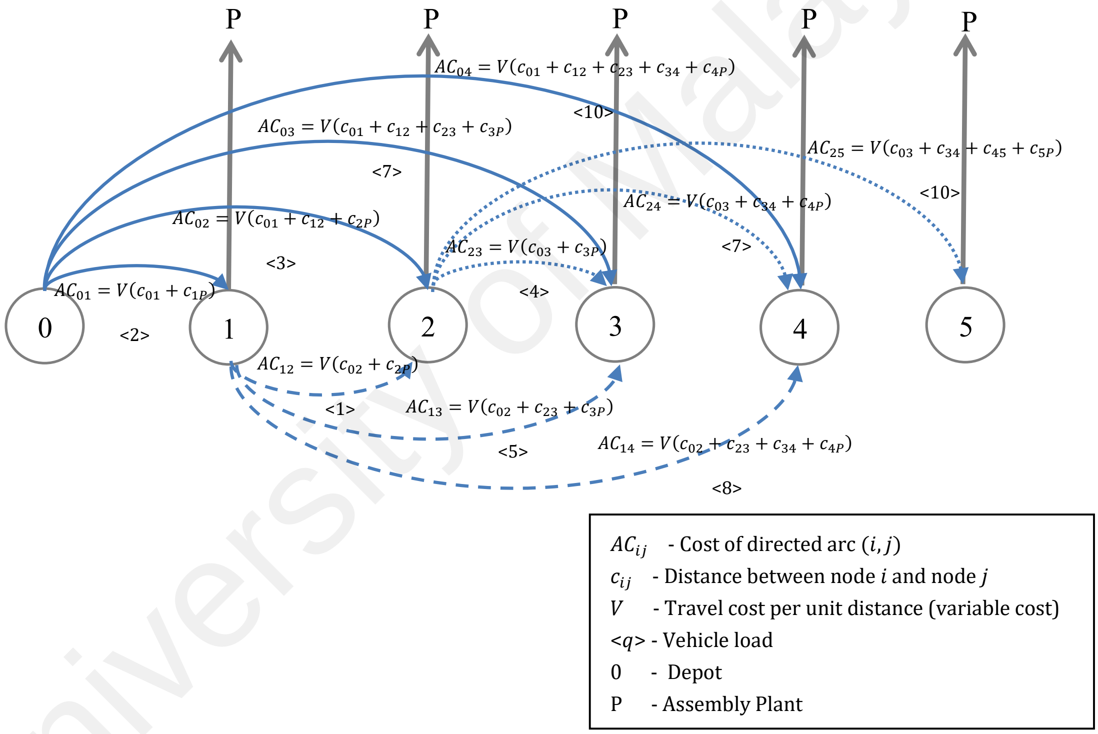


Figure 3.4: Directed Cost Network.

3.4.2.3 Allocation Model (STEP 1.1)

In addition to the randomly generated solution, an allocation model was also used to obtain a good feasible initial solution. The used of the allocation model is to ensure a good starting and hopefully this improves the overall results (Bard & Nananukul, 2009; Golden et al., 1984). Allocation model is the simplified version (relaxed) of the full model where the routing component is completely removed and substituted with an approximated cost.

The mixed integer linear programming of the allocation model proposed in Bard and Nananukul (2009) for the production inventory distribution problem was modified and adopted in developing the allocation to suit the inventory routing problem studied. The full model is modified by removing the routing variables (x_{ijt}) and constraints that contribute to the routing ((3.5),(3.6),(3.10)-(3.15)). An aggregated vehicle capacity constraints were introduced to the allocation model. The results of the allocation model provide the number of items picked up from each supplier in each period in the planning horizon. The following additional parameter and variables are introduced.

f_{it}^{C}	:	Fixed cost of making collection to supplier i in period t
e_{it}^{C}	:	Variable cost of collecting one item to supplier i in period t
z_{it}^{C}	:	1 if collection is made to supplier i in period t; 0 otherwise

Since the routing constraints are deleted in the allocation model, an alternative representation for the cost term is used to determine the actual cost which is needed to make a collection to supplier i in period t. All the problems considered in this study are large instances with $N^2T > 500$ (Bard & Nananukul, 2009). The variables are set $z_{it}^{C} = f_{it}^{C} = 0$. The variable cost term, $\sum_{i\in S}\sum_{t\in\tau} e_{it}a_{it}$ where e_{it} is approximated by the travel cost from depot to supplier i divided by the demand of supplier i in period t: $e_{it} = 2c_{0i}/d_{it}$, the logistic ratio. Thus, the new objective function, includes two components as in constraint (3.18).

The modified version of the allocation model is given as follows.

$$\min Z' = \sum_{i\in S}\sum_{t\in\tau} e_{it}a_{it} + \sum_{i\in S} h_i\left(\sum_{t\in\tau} I_{it}\right) \tag{3.18}$$

Subject to:

Constraints (3.2), (3.8) and (3.9).

3.4.3 Inventory Updating Mechanism (STEP 3.1)

The inventory updating mechanism is used in the employed bee phase. There are two different mechanisms for inventory updating, the first one is forward transfer and the second one is backward transfer.

The main objective of a forward transfer is to reduce the inventory where the transfer is done from earlier period to the succeeding period, that is from period t to period $t+1$. A backward transfer focuses on reducing the transportation cost at the expense of a slight increase of the inventory holding cost. The transfer is done from the current period to the preceding period, that is from period t to $t-1$.

The forward transfer allows up to 3 succeeding periods, whilst the backward transfer only allows is up to 2 preceding periods. This is to ensure that the tradeoff between the savings in traveling distance and the reduction (forward transfer) or increase (backward transfer) of inventory cost was not excessive. We carried out limited experiments to determine the parameters.

3.4.3.1 Illustrations of Forward and Backward Transfer

We consider 5 suppliers with 5 periods with the inventory holding cost and distance matrices given in Figure 3.5 and Figure 3.6, respectively. Note that we assume $c_{ij} = c_{ji}$.

Supplier i	h_i
1	12
2	9
3	6
4	3
5	6

Figure 3.5: The inventory holding cost, h_i for supplier i.

	D	1	2	3	4	5	AP
D							
1	2.0000						
2	1.4142	3.1623					
3	4.4721	6.3246	3.1623				
4	5.8310	7.0711	4.4721	3.1623			
5	2.2361	3.6056	3.0000	5.0000	7.2801		
AP	2.0000	4.0000	1.4142	2.8284	5.0990	2.2361	

Figure 3.6: Travel Distance, c_{ij}.

(a) ***Forward Transfer***

The selection of the supplier for the forward transfer is favorable towards the supplier with high inventory holding cost. Figure 3.7 illustrates the forward transfer. The illustration shows the routes and inventory cost before and after the forward transfer. The period considered in this example is period 1 to 3.

Given that there are 3 routes in period 1; $\{\{2,5\}, \{1\}, \{3,4\}\}$. Routes are given in blue shade. Each with corresponding pickup quantities $\{\{2,3\}, \{10\}, \{2,5\}\}$ and inventory quantities as $\{\{0,1\}, \{6\}, \{0,1\}\}$. The cost of the routes is calculated starting from depot (D) to supplier(s) and then to the assembly plant (P). Transportation cost from assembly plant to the depot is omitted as the cost is fixed. The first route in period 1 is $D - 2 - 5 - P$ with route cost is $1.4142 + 3.0000 + 2.2361 = 6.6503$. Second route is $D - 1 - P$ with route cost 6.0000. The third route is $D - 3 - 4 - P$ with route cost 12.7334. The total route costs in period 1 is 25.3837 as given in Figure 3.7. The inventory cost is calculated as $1 \times 6 + 6 \times 12 + 1 \times 3 = 81$. For all 3 periods considered, the total route cost is 50.2322 and the total inventory cost is 201 which resulting in the overall total cost is 251.2322.

The selected supplier must have an inventory as the aim of the forward transfer is to reduce the inventory holding cost and finding the balance in the routing cost. Supplier 1 in period 1 (orange shade) is selected since it has the highest holding cost compared to supplier 4 and 5 (suppliers with inventory) with 6 inventory quantities for the transfer (see Figure 3.5). All 6 units are transferred to succeeding period 2 and 3, leaving only the exact demand, 4 for supplier 1 in period 1. 2 and 4 units are transferred to period 2 and 3 accordingly.

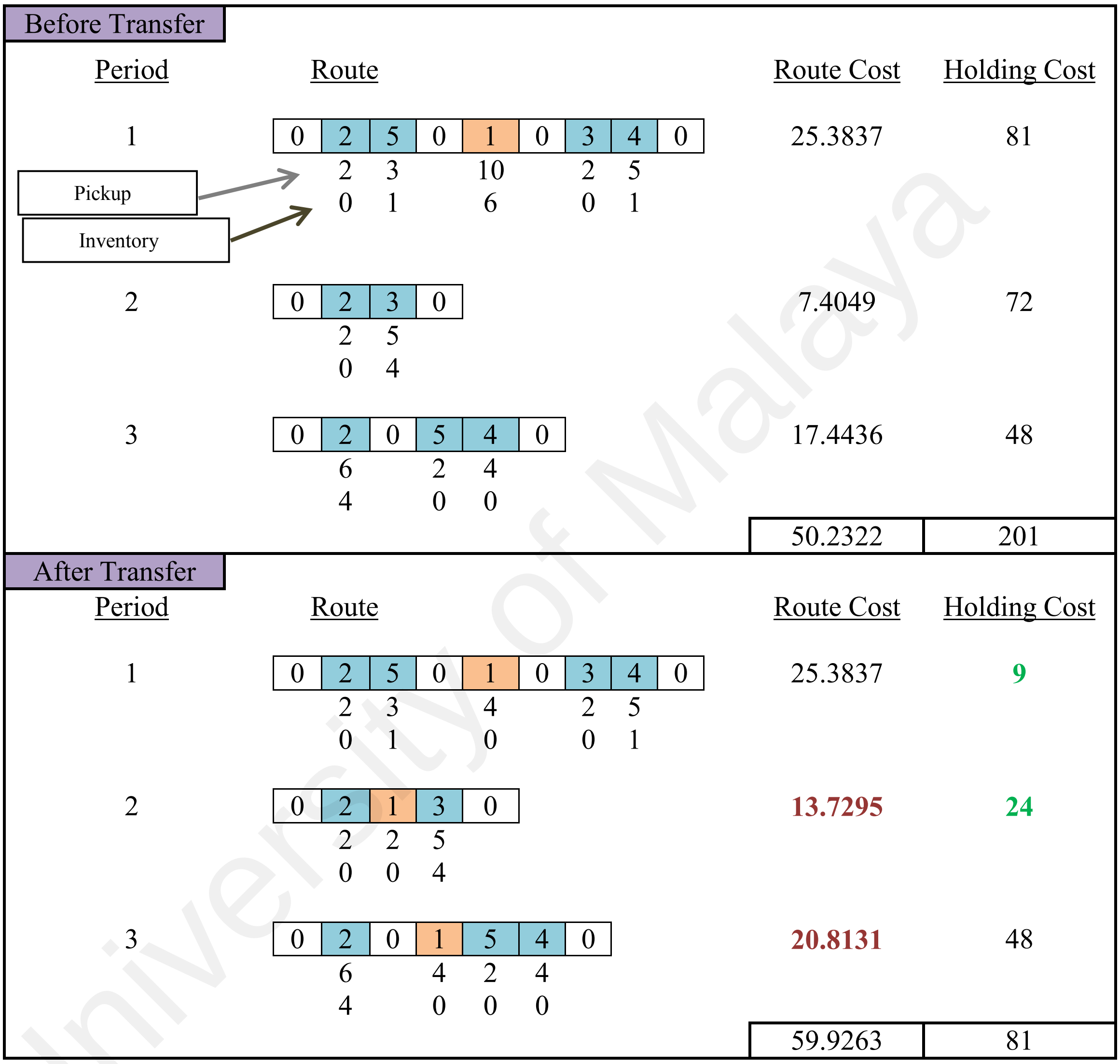


Figure 3.7: Example of Forward Transfer.

Note that transferring the whole quantities (6 units) to period 2 is not possible as the vehicle capacity must be respected. The transfer reduces the inventory cost for period 1 and 2 (green bold), at the same time, increases the transportation cost for period 2 and 3 (in bold maroon). However, the overall saving for the transfer is more that is $251.2322 - 140.9263 = 110.3059$. Best insertion method is used to insert supplier 1 in the succeeding periods. Note that forward transfer can only be done if the supplier is not visited in the subsequent period (inventory is positive) and it is selected from periods $\tau = 1, 2, \dots, T-1$.

(b) ***Backward Transfer***

The objective of the backward transfer is to reduce the transportation cost by combining the inventory (if it is feasible). This is carried out at the expense of a slight increase in the holding cost. The selection of the supplier to be transferred is biased towards supplier with a lower holding cost.

The suppliers are selected from periods $\tau = 2, \dots, T$ as no backordering is allowed. Supplier 4 (orange shade) is selected (having the lowest holding cost) where all 2 units from period 5 is transferred to period 4. Consequently, there is an increase in the holding cost to 24 (red bold). However, there is a reduction in the transportation cost (green bold). The total savings as illustrated in Figure 3.8 is $61.1355 - 56.2056 = 4.9300$.

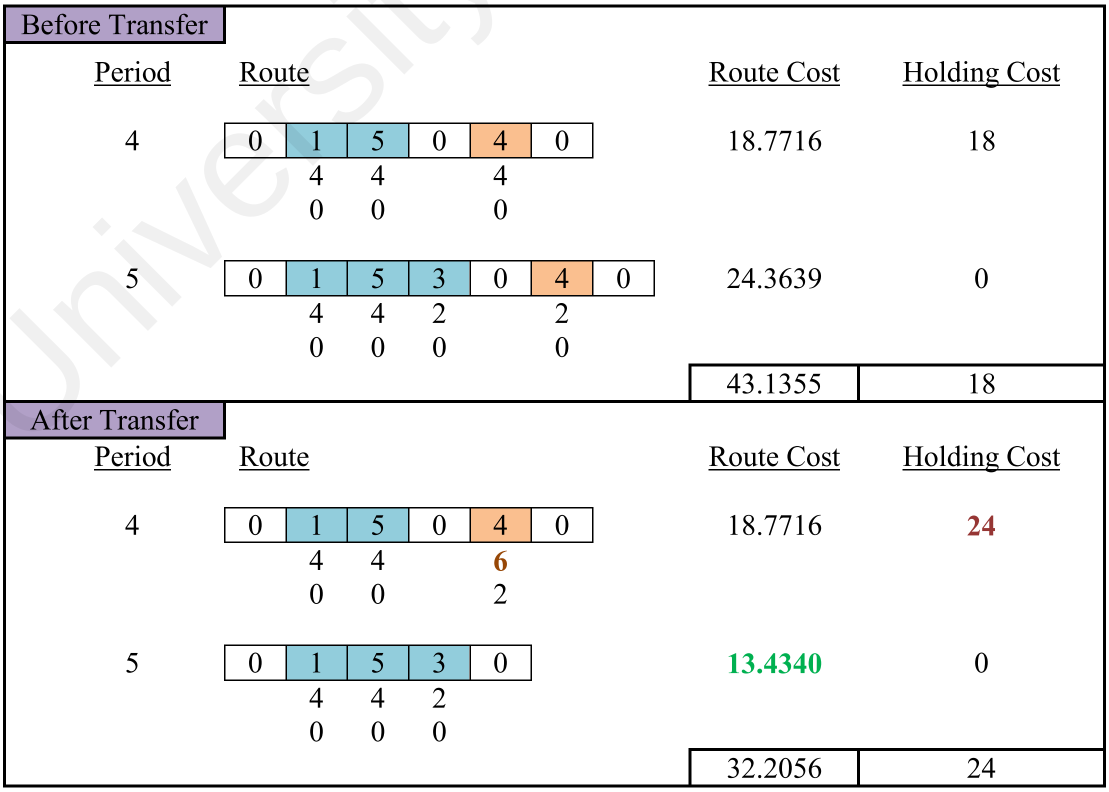


Figure 3.8: Example of Backward Transfer.

3.5 Results and Discussions

3.5.1 Datasets

The performance of the ABC algorithm, *ABCIRP* developed were tested on 14 datasets (Moin et al., 2011). The datasets were created from 4 original datasets provided in Lee et al. (2003). The original datasets are S12T14 (12 suppliers, 14 periods), S20T21 (20 suppliers, 21 periods), S50T21 (50 periods, 21 periods) and S98T14 (98 suppliers, 14 periods). Another 10 datasets were created from the original four datasets to segregate the datasets into small, medium and large datasets. The 10 datasets vary in the number of periods into 5, 10 and 14 periods; S12T5, S12T10, S20T5, S20T10, S20T14, S50T5, S50T10, S50T14, S98T5 and S98T10.

Each dataset is characterized by fixed cost of using vehicle, travel cost per unit distance, vehicle capacity and coordinates of the depot, suppliers and assembly plant. Table 3.1 summarizes the criteria of the datasets. Note that the coordinate of depot is (0,0) for all datasets and the coordinate of assembly plant for dataset S98 is different from Lee et al. (2003).

Table 3.1: Datasets Criteria.

Dataset	S12	S20	S50	S98
Fixed Vehicle Cost	20	20	20	200
Travel Cost per Unit Distance	1	1	1	50
Vehicle Capacity	10	10	10	400
Range of Holding Costs	[3,27]	[3,27]	[1,9]	[1,44]
Range of Demand	[1,4]	[1,4]	[0,9]	[0.0400, 393.3300]
Coordinate of Assembly Plant	(10,20)	(10,20)	(10,20)	(42.31,- 83.17)

3.5.2 Computational Results

ABCIRP algorithm was written in Matlab 7.7 and performed on computer with 3.1 GHz processor with 8GB of RAM. The performance of the *ABCIRP* is tested by comparing with two other metaheuristics; Scatter Search (SS) and Genetic Algorithm

(GA). Results of SS are taken from Moin et al. (2014). GA is given in Appendix A.1. Allocation model for generating one of the initial solutions were run on CPLEX 12.6.

In *ABCIRP*, we employ a total of 50 bees (representing the solutions) and divided into 25 employed bees and 25 onlooker bees. It is common to take an equal number of onlooker bees and employed bees (Karaboga & Basturk, 2007; Szeto et al., 2011). The 25 employed bees used comprises 23 randomly generated bees, 1 bee from the allocation model and 1 bee with zero holding cost (all suppliers are visited in all periods). The maximum number of iteration ($MAXITER$) is set to 250. The exploitation parameter, $LIMIT$ value varies for each dataset. $LIMIT$ is set to 25 multiply the maximum number of suppliers (MAX_SUPP); $LIMIT = 25 \times MAX\_SUPP$ (Szeto *et al*., 2011).

We carried out 10 independent runs for each dataset. Details results are given in Appendix A.2. Table 3.2 presents the best results found for *ABCIRP,* SS and GA. The best among the three algorithms are given in bold. It is observed that *ABCIRP* achieved 10 best results out of the 14 datasets. *ABCIRP* are superior for the small (S12), medium (S20) and the large dataset (S98) except in the dataset for 50 suppliers (S50) where the best is found using SS. When compare *ABCIRP* with GA specifically in dataset S50, *ABCIRP* is better in all except in S50T5.

It is observed that the S50 cases results of *ABCIRP* have low inventory costs, as the algorithm focused more on reducing the inventories while the routing parts are handled with only 1 Giant Tour in the initialization phase and 2-opt and 1-0 in the onlooker bee phase. This is contrary to the SS and GA where both algorithms emphasized more on the routing part. SS embeds the *References Set* which keeps the high quality solution in term of routing and GA adopts Double Sweep Algorithm for initial routes and carried out the 2-opt as post-optimization.

Table 3.2: Best Solution of *ABCIRP*, SS and GA.

Dataset	***ABCIRP***		**SS**		**GA**	
	Objective	**#veh**	**Objective**	**#veh**	**Objective**	**#veh**
S12T5	**1994.04**	14	2062.20	14	2041.69	14
S12T10	**4035.18**	29	4285.11	30	4181.33	30
S12T14	**5668.10**	41	6108.74	44	5935.97	43
S20T5	**3132.72**	24	3191.62	22	3139.67	22
S20T10	**6402.09**	49	6580.71	45	6428.70	44
S20T14	**9017.58**	69	9294.17	64	9053.71	62
S20T21	**13594.69**	104	14238.94	97	13696.46	94
S50T5	5515.29	47	**5392.40**	47	5489.40	47
S50T10	11610.06	104	**11420.58**	101	11631.19	101
S50T14	16530.82	147	**16340.18**	142	16828.54	143
S50T21	25391.11	228	**25210.38**	224	25930.73	221
S98T5	**586970.30**	60	606985.76	64	610264.61	62
S98T10	**1174087.80**	120	1227995.17	127	1215113.34	123
S98T14	**1643901.12**	168	1723650.99	185	1718060.57	173

Consequently, the vehicle utilization in ABC for all cases except S12 and S98 are not maximum, as the number of vehicles differs up to 10 when compared to the other two algorithms, GA and SS. Utilization of GA and SS for dataset S20 and S50 are better especially in dataset S20T21. We also conjecture that the fixed transportation cost is too low for it to have any significant impact on the number of vehicles.

Table 3.3 presents the average computational time in seconds and *ABCIRP* required the least computation time, especially for larger datasets. On average SS has the least computation time compared to *ABCIRP* and GA except for large datasets, S98 where the computational time increases exponentially.

Table 3.3: Average CPU times in Seconds.

Dataset	*ABCIRP*	SS	GA
S12T5	67.580	10.720	53.228
S12T10	112.200	22.610	569.076
S12T14	166.970	35.610	2314.821
S20T5	123.522	58.580	501.340
S20T10	241.084	71.160	799.661
S20T14	307.618	86.350	2262.748
S20T21	458.440	101.950	4314.520
S50T5	289.866	138.450	356.150
S50T10	606.938	157.450	2859.233
S50T14	792.298	192.900	4414.126
S50T21	1261.767	253.500	8956.829
S98T5	1988.607	4574.290	4619.172
S98T10	3720.421	8858.730	12101.559
S98T14	5089.860	13061.160	17043.405

Table 3.4 tabulates the best solution, standard deviation (STDEV) for each of the datasets for each algorithm and also the percentage difference between all the algorithms. GA is better compared to SS (all except S50) with the percentage difference up to 3.961%. However, when comparing GA with *ABCIRP*, *ABCIRP* are able to find better solutions with up to 4.513% except for case S50T5 with a slightly higher deviation at 0.472%. *ABCIRP* is better than SS (all except S50) with percentage difference up to 7.396%. The average deviation shows the performance of *ABCIRP* is better when comparing SS and GA with an average deviation of 2.599% and 1.942% respectively. The standard deviation results of *ABCIRP* are smaller compared to both SS and GA, which shows that *ABCIRP* developed is more consistent in finding the best results.

3.5.3 Statistical Analysis

With the emerging of the new search algorithm, it is necessary to carry out a statistical analysis to find out if the new proposed algorithm provides a significant improvement when compared to the existing algorithm in the field studied. A nonparametric statistical

Table 3.4: The best solution, standard deviation and percentage difference between the algorithms.

Dataset	Criteria	*ABCIRP*	SS	GA	PERCENTAGE DIFFERENT (%Δ)		
					ABCIRP & SS	SS & GA	*ABCIRP* & GA
S12T5	BEST	**1994.04**	2065.26	2041.69	-3.449	1.154	-2.334
	STDEV	0.00	40.76	36.71			
S12T10	BEST	**4035.18**	4295.48	4181.33	-6.060	2.730	-3.495
	STDEV	0.00	48.20	68.04			
S12T14	BEST	**5668.10**	6120.80	5935.97	-7.396	3.114	-4.513
	STDEV	0.00	36.90	78.47			
S20T5	BEST	**3132.72**	3205.18	3139.67	-2.261	2.087	-0.221
	STDEV	8.94	39.48	18.96			
S20T10	BEST	**6402.09**	6615.42	6428.70	-3.225	2.904	-0.414
	STDEV	10.86	48.28	35.34			
S20T14	BEST	**9017.58**	9294.17	9053.71	-2.976	2.656	-0.399
	STDEV	6.90	50.83	42.45			
S20T21	BEST	**13594.69**	14238.94	13696.46	-4.525	3.961	-0.743
	STDEV	10.14	74.54	62.21			
S50T5	BEST	5515.29	**5392.40**	5489.40	2.279	-1.767	0.472
	STDEV	14.05	45.57	86.83			
S50T10	BEST	11610.06	**11420.58**	11631.19	1.659	-1.811	-0.182
	STDEV	18.99	57.02	86.61			
S50T14	BEST	16530.82	**16340.83**	16828.54	1.163	-2.898	-1.769
	STDEV	8.54	61.02	107.61			
S50T21	BEST	25391.11	**25210.38**	25930.73	0.717	-2.778	-2.081
	STDEV	5.98	74.40	81.60			
S98T5	BEST	**586970.30**	606985.76	610264.61	-3.298	-0.537	-3.817
	STDEV	42.41	4644.06	7236.06			
S98T10	BEST	**1174087.80**	1227995.17	1215113.34	-4.390	1.060	-3.376
	STDEV	56.21	8949.68	9808.35			
S98T14	BEST	**1643901.12**	1723650.99	1718060.57	-4.627	0.325	-4.316
	STDEV	1394.46	9426.38	9607.30			
Average (%Δ)					-2.599	0.729	-1.942

analysis was performed for multiple comparisons to find a significant difference between the behavior of algorithms *ABCIRP*, SS and GA (Derrac et al., 2011).

3.5.3.1 The Friedman, Iman Davenport and Friedman Aligned Rank

The Friedman test, Iman and Davenport, and Friedman Aligned Rank test are carried out to determine the significant difference between the algorithms. The choice of different tests is to alleviate the weakness of the Friedman test.

(a) ***Friedman Test***

The Friedman test computes the average rankings obtained by k algorithms over n datasets. The steps are as follows (Derrac et al., 2011):

1. Rank from $l = 1 \dots k$ for best costs from each dataset m. The rank, r_m^l indicates that 1 gives the best result and k gives the worst result.
2. Get the final rank for each algorithm l; $R_l = \frac{1}{n}\sum_{m \in n} r_m^l$.
3. Then, calculate the Friedman test statistic value, $\chi_F^2 = \frac{12n}{k(k+1)}\left[\sum_l R_l^2 - \frac{k(k+1)^2}{4}\right]$, where χ_F^2 follows a χ^2 distribution with $k-1$ degrees of freedom.

(b) ***Iman-Davenport (ID) Test***

In view that the Friedman test is conservative, Iman and Davenport proposed a less conservative test. The ID 's Friedman value (F_{ID}) is derived from the Friedman test, where F_{ID} follows a F distribution with $k-1$ and $(k-1)(n-1)$ degrees of freedom. The statistic is given by $F_{ID} = \frac{(n-1)\chi_F^2}{n(k-1)-\chi_F^2}$ (Derrac et al., 2011).

(c) ***Friedman Aligned Rank (FAR) Test***

The ranking scheme adopted in the Friedman test has a weakness in which it allows for intra-set comparison only. A FAR test is proposed, where the observation is aligned with respect to the datasets as well as with respect to the algorithms. The steps are (Derrac et al., 2011) as follows:

1. Denote the best cost for each dataset m found by algorithm l as y_{lm}, then find the average of the best costs for each dataset m; $avg_m = \frac{\sum_{l \in k} y_{lm}}{k}$.
2. Form each dataset m, calculate the residual by $res_{lm} = y_{lm} - avg_m$ for $l \in k$.
3. Do ranking from 1 until kn; where 1 is assigned to the smallest res_{lm}, 2 for the next smallest res_{lm} and so on until all res_{lm}s are ranked.
4. Calculate the rank total of each l algorithm $\hat{R}_l$ and rank total of each m dataset $\hat{R}_m$.
5. Then, calculate the Friedman Aligned Rank test statistic value,

$$F_{AR} = \frac{(k-1)\left[\sum_{l=1}^{k} \hat{R}_l^2 - \left(\frac{kn^2}{4}\right)(kn+1)^2\right]}{\left\{\frac{[kn(kn+1)(2kn+1)]}{6}\right\} - \left(\frac{1}{k}\right)\sum_{m=1}^{n} \hat{R}_m^2}$$

The F_{AR} follows a χ^2 distribution with $k-1$ degrees of freedom.

3.5.3.2 Application of Statistical Tests

In this problem, the null hypothesis (H_0) and the alternative hypothesis (H_1) at significance level $\alpha = 0.05$ are given below.

H_0: There are no significant differences between the performances of *ABCIRP*, SS and GA.

H_1: At least one of the *ABCIRP*, SS and GA differs in performance.

The first test applied is Friedman test with $k = 3$ algorithms over $n = 14$ datasets. Table 3.5 presents the ranking for each dataset based on the best costs given in Table 3.2. The final rankings, R_l were also tabulated in Table 3.5. The computed value of χ_F^2 which is 8.7143 is then compared with the significance level $\alpha = 0.05$. Since the $\chi_F^2 > \chi_{0.05,2}^2$,

H_0 is rejected and it can be concluded that at least one of the *ABCIRP*, SS and GA differ in performance.

Table 3.5: The ranking and the final ranking of Friedman test statistic.

Dataset	***ABCIRP***	**SS**	**GA**
S12T5	1	3	2
S12T10	1	3	2
S12T14	1	3	2
S20T5	1	3	2
S20T10	1	3	2
S20T14	1	3	2
S20T21	1	3	2
S50T5	3	1	2
S50T10	2	1	3
S50T14	2	1	3
S50T21	2	1	3
S98T5	1	2	3
S98T10	1	3	2
S98T14	1	3	2
R_l	1.357	2.357	2.286

The second test is the ID test, where the computed value $F_{ID} = 5.8741 > F_{0.05,2,26}$, so H_0 is rejected at level $\alpha = 0.05$. The final test is the FAR statistic test where the computed statistic value, $F_{AR} = 8.6119 > \chi^2_{0.05,2}$. Thus H_0 is rejected. The rankings and rank totals for the FAR test statistic are given in Table 3.6.

Table 3.6: The rankings of Friedman Aligned Rank test statistic.

Dataset	***ABCIRP***	**SS**	**GA**	$\hat{R}_m$
S12T5	17	24	21	62
S12T10	10	31	22	63
S12T14	6	33	25	64
S20T5	19	26	20	65
S20T10	14	30	16	60
S20T14	12	32	15	59
S20T21	5	35	8	48
S50T5	27	13	23	63
S50T10	28	9	29	66
S50T14	18	7	34	59
S50T21	11	4	36	51
S98T5	3	37	38	78
S98T10	2	40	39	81
S98T14	1	42	41	84
$\hat{R}_l$	367	363	173	

Summary of the test statistics and their critical values are exhibited in Table 3.7. For all the tests carried out, with 95% confidence, the performance of algorithms *ABCIRP*, SS and GA are significantly different.

Table 3.7: Results of the Friedman, Iman and Davenport and Friedman Aligned Rank tests with significance level, $\alpha = 0.05$.

Tests	Friedman	Iman-Davenport	Friedman Aligned Rank
Statistic	$\chi_F^2 = 8.7143$	$F_{ID} = 5.8741$	$F_{AR} = 8.6119$
Critical Value	5.9915	3.3690	5.9915
Decision	Reject H_0	Reject H_0	Reject H_0

Footnote: $\chi^2_{0.05,2} = 5.9915$; $F_{0.05,2,26} = 3.3690$

3.5.3.3 Post-hoc Procedures

The Friedman, ID and FAR test can only detect the significant differences between *ABCIRP*, SS and GA. A statistical test is further extended to accomplish multiple comparisons by performing a post-hoc procedure to evaluate the performance of the best algorithm (control algorithm) compared to the rest of the algorithms. Here, a Bonferroni-Dunn procedure is used (see Derrac et al. (2011)) based on the Friedman statistical results.

The *ABCIRP* algorithm is used as the control method as it has the best average ranking (see Table 3.5). The p-value can be obtained through the conversion of the rankings computed by using normal approximation. The test statistic z depends on Friedman test for comparing the control algorithm (denote as j-th algorithm) and SS and GA (denoted as the i-th algorithm) is given as $z_i = \left(R_i - R_j\right)/\sqrt{k(k+1)/6n}$. Then, denote each p_i-value for each corresponding z_i.

The p_i-value is not suitable for multiple pairwise comparisons as it disregards the family-wise error rate (FWER). Therefore, the Bonferonni procedure proposed a correction to deal with the problem by computing the adjusted p_i-value (APV_i)

specifically $APV_i = (k-1) \times p_i$. APV_i is compared with the level of confidence, $\alpha = 0.05$.

If $APV_i < \alpha$, the null hypothesis that there is no significant difference between the *ABCIRP* and the other algorithms are rejected. Results are presented in Table 3.8. Thus, it can be concluded that the control algorithm, *ABCIRP* is significantly different from the SS and GA at 0.05 significance level.

Table 3.8: Comparison of the *ABCIRP* algorithm with SS and GA algorithm.

Algorithm	z - value	p – value	Adjusted p– value (APV)	Decision
GA	2.4568	0.0140	0.0280	Rejected
SS	2.6458	0.0082	0.0164	Rejected

3.5.4 Enhancement on *ABCIRP*

Despite that *ABCIRP* performing better than the other algorithms, SS and GA, there are areas that the algorithm can be further improved. The areas that warrant some improvements are vehicle utilization specifically for datasets S98.

The *ABCIRP* algorithm is modified and enhanced in several areas. The enhancement is in **STEP 1.2**, where 4 possible Giant tours are considered instead of just 1. Then the best routing among all four is selected. The four Giant tours are constructed based on the following criteria:

- Closest supplier to the depot.
- Farthest supplier to the depot
- The biggest gap between two suppliers where we consider both clockwise and counterclockwise directions of the sweep.

We carry out some parameters testing to determine the best settings. It is observed for parameter for *LIMIT* the improvement always occurs when *LIMIT* value is less than 100.

However, sometimes, but very rarely, the improvement is observed with larger values. Thus, the parameter *LIMIT* is set to $min\{MAX\_SUPP \times MAX\_PERIOD, 200\}$.

Based on limited experiments, the value for *MAXITER* is set at 500 and this new value is set to balance the new *LIMIT* introduced. However, we observed that in large datasets there is an improvement in every iteration. Hence the iteration is very slow to converge and time consuming. An additional termination condition is set where the algorithm is terminated in 3600 seconds (1 hour) or *MAXITER*, whichever comes first.

In *ABCIRP*, the task of employed bees and onlooker bees are distinct where the task of the employed bees is to update the inventory management whilst routing improvement is carried out by the onlooker bees. This limits the exploitation work of the onlooker bees. Hence we modify **STEP 3.2b(ii)** and instead of just exploiting the food source by improving the routing part, the onlooker bees are also assigned the inventory updating mechanism. The backward and forward transfers are embedded together with the existing neighborhood operators in order to improve the search process.

And we add the post-optimization phase, where refinement in the routing and similar to the approach in Imran et al. (2009). Here, the entire vehicle in a period is collapsed to form a giant tour again while maintaining the current sequence, partitioned and obtained the new routing by applying the Dijkstra's algorithm (Dijkstra, 1959). This is done for each period. The implementation of post-optimization is to improve the vehicle utilization, which is the weakness of *ABCIRP*.

The enhanced version of *ABCIRP* is denoted as *EABCIRP*. Table 3.9 depicts the differences of parameter settings and additional enhancements between *ABCIRP* and *EABCIRP*.

Table 3.9: Difference in *ABCIRP* and *EABCIRP*.

Criteria	*ABCIRP*	*EABCIRP*
Number of Giant Tour used	1	4
MAX_ITER	250	500
$LIMIT$	$25 \times MAX\_SUPP$	$min\ (MAX\_SUPP \times MAX\_PERIOD, 200)$
Onlooker Exploitation (**STEP 3.2b(ii)**)	Swap $1-0$ and $2-opt$	Swap $1-0$, $2-opt$ and Inventory Updating
Post Optimization Procedure	-	Dijkstra's Algorithm
Termination Condition	*MAXITER*	*MAXITER* and 3600 seconds

Table 3.10 tabulates the results of *EABCIRP* and *ABCIRP*. The percentage difference between *EABCIRP* and *ABCIRP* is also provided. All results *EABCIRP* are better than *ABCIRP* including S50, which is better than SS (previously were worse than from SS). However, *EABCIRP* requires longer computation time than *ABCIRP* because of the much more intensive and more focused inventory updating mechanism with the exception of S98T10 and S98T14. All the best solutions are given in bold. The range of improvement is from 0.11% to 5.96% (S50T14) and on average *EABCIRP* find better results with 2.25% percentage difference.

Table 3.10: Results of *EABCIRP*.

Dataset	*ABCIRP*	#veh	*EABCIRP*	#veh	TIME	STDEV	(%Δ)
S12T5	1994.04	14	**1961.71**	14	242.95	4.56	1.65
S12T10	4035.18	29	**4012.65**	29	432.53	8.86	0.56
S12T14	5668.1	41	**5645.57**	41	590.77	8.47	0.40
S20T5	3132.72	24	**2987.73**	22	1466.77	20.78	4.85
S20T10	6402.09	49	**6221.23**	46	1399.13	18.59	2.91
S20T14	9017.58	69	**8751.05**	65	1906.02	44.41	3.05
S20T21	13594.69	104	**13233.06**	97	2849.31	56.37	2.73
S50T5	5515.29	47	**5355.01**	47	3600.00	18.99	2.99
S50T10	11610.06	104	**11392.59**	101	3600.00	21.57	1.91
S50T14	16530.82	147	**15600.78**	137	3600.00	313.98	5.96
S50T21	25391.11	228	**24535.92**	219	3600.00	178.40	3.49
S98T5	586970.3	60	**585854.63**	60	3600.00	139.25	0.19
S98T10	1174087.8	120	**1165316.11**	119	3600.00	2787.84	0.75
S98T14	1643901.12	168	**1642123.06**	168	3600.00	309.36	0.11
Average							2.25

The vehicle utilization of *EABCIRP* is also improving, as the number of vehicles used is reduced up 10 vehicles (S50T14) when compared to *ABCIRP*. The utilization is also improved when compared to GA except for 3 datasets S20T10, S20T14 and S20T21. Note that smaller datasets (S12 and S20) terminate in maximum number of iterations while bigger datasets (S50 and S98) terminate within 3600 seconds.

The convergence plots of *EABCIRP* for all datasets (any 1 run) were given in Figure 3.9. The graphs plot the best total cost found versus the number of iterations.

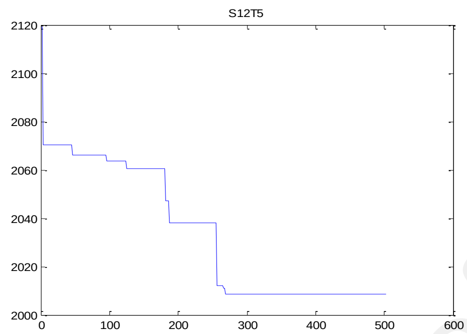


Figure 3.9 (a): Dataset S12T5.

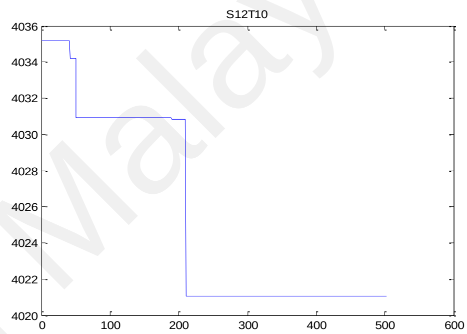


Figure 3.9 (b): Dataset S12T10.

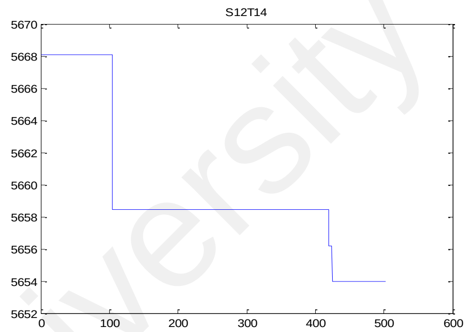


Figure 3.9 (c): Dataset S12T10.

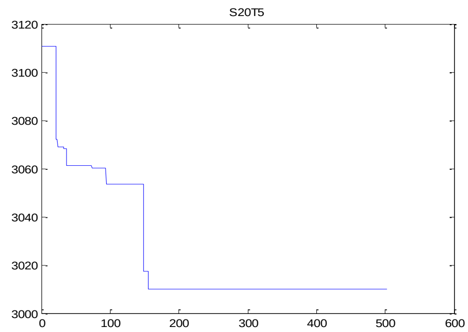


Figure 3.9 (d): Dataset S20T5.

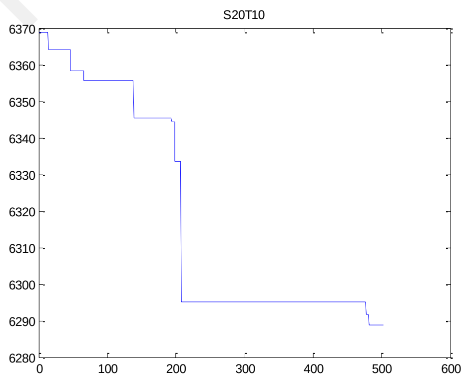


Figure 3.9 (e): Dataset S20T10.

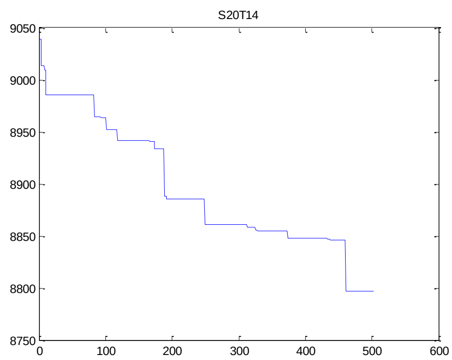
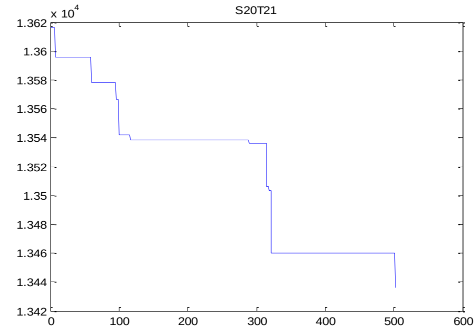


Figure 3.9 (f): Dataset S20T14.

Figure 3.9: Convergence plot for all dataset.



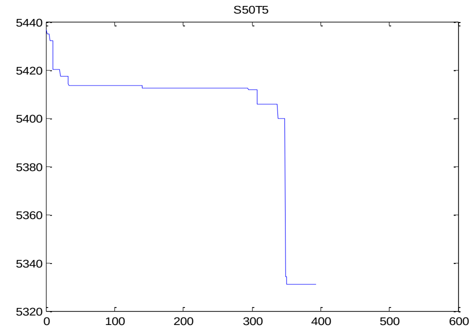
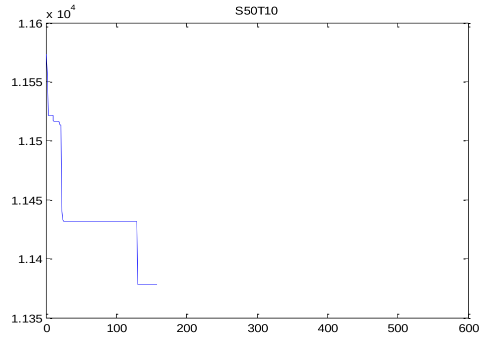


Figure 3.9 (g): Dataset S20T21. **Figure 3.9 (h):** Dataset S50T5.



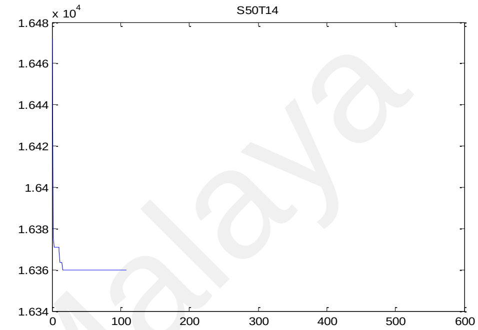
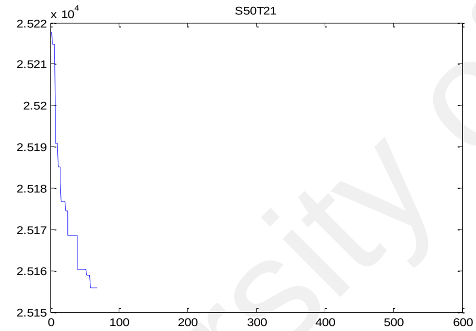


Figure 3.9 (i): Dataset S50T10. **Figure 3.9 (j):** Dataset S50T14.



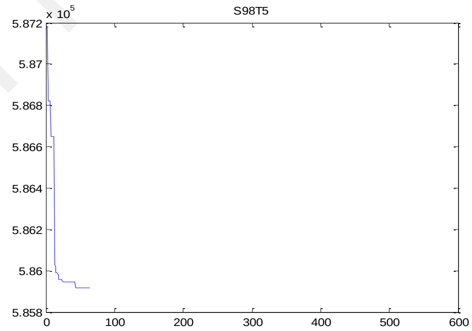
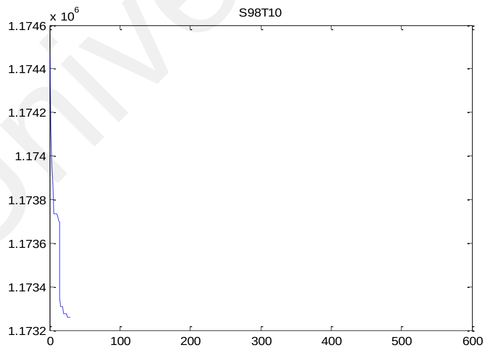


Figure 3.9 (k): Dataset S50T21. **Figure 3.9 (l):** Dataset S98T5.



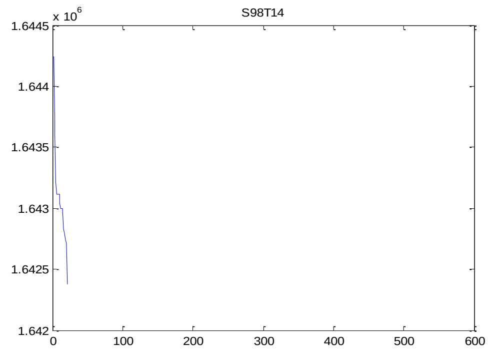


Figure 3.9 (m): Dataset S98T10. **Figure 3.9 (n):** Dataset S98T14.

Figure 3.9, Continued.

It is observed that the datasets S50 and S98 terminate prematurely after 3600 seconds and could have benefitted if the algorithm is allowed to run for a longer period of time. We let the algorithms to run for 14400 seconds (4 hours) and the best results (out of 10 runs) are tabulated in Table 3.11. The results for S50 and S98 clearly show that the algorithm benefited from longer running time, except for S98T10.

Table 3.11: Results of *EABCIRP* for 1 hour and 4 hours in datasets S50 and S98.

Dataset	1 hour	#veh	4 hours	#veh	TIME	STDEV	(%Δ)
S50T5	5355.01	47	**5342.42**	46	3997.58	22.66	0.24
S50T10	11392.59	101	**10844.13**	97	9801.48	205.81	5.06
S50T14	15600.78	137	**15502.75**	136	14401.86	365.91	0.63
S50T21	24535.92	219	**24068.04**	215	14410.50	340.59	1.94
S98T5	585854.63	60	**577723.35**	59	14425.91	2600.06	1.41
S98T10	**1165316.11**	119	1172366.21	120	14407.06	99.17	-0.60
S98T14	1642123.06	168	**1641613.40**	168	14458.69	277.93	0.03
Average							1.24

Results show that all of S50 and S98 datasets were significantly improved when running time is extended to 4 hours except for dataset S98T10. The improvements are, on average, 1.24%. The number of vehicles used is also reduced up to 4 vehicles. It is noted that all the datasets considered terminate in 4 hours except for 2 datasets (S50T5 and S50T10) which is earlier.

3.6 Summary

In this section, the ABC algorithm (*ABCIRP*) is successfully developed to solve inventory routing problem (IRP). *ABCIRP* is embedded with inventory updating mechanism that are forward and backward transfers in the employed bee phase whereas in the onlooker bee phase 2-opt and 1-0 exchange are adopted. These two combination are able to find the balance between inventory and transportation. A comparison between *ABCIRP*, SS and GA is done where *ABCIRP* obtained 10 better results out of 14 datasets. A statistical analysis is performed which shows that there is a significant difference between *ABCIRP*, SS and GA. *ABCIRP* is further improved, *EABCIRP*, which gives the best results compared to the others. Running time for *EABCIRP* in large datasets S98 is extended which shows significant improvement in the results.

CHAPTER 4: INVENTORY ROUTING PROBLEM WITH BACKORDERING

This chapter presents the second contribution of the thesis. The chapter discusses the implementation of the ABC algorithm for Inventory Routing Problem with Backordering (IRPB). The ABC for IRPB is denoted as *ABCIRPB*. This chapter starts with the description of the IRPB problem, and the two specific cases of backorder decision studied and the Mixed Integer Linear Programming (MILP) formulation of the problem is presented. The algorithm designed for IRPB, *ABCIRPB* is explained which includes the modified Giant Tour procedure to handle a fixed number of vehicles. *ABCIRPB* embeds two different inventory updating mechanisms: *random exchange, ABCRX* and *guided exchange*, *ABCGX*. The results from both algorithms are compared with the LB, UB and the results from Abdelmaguid et al. (2009). Results of the statistical analysis performed are then presented in Section 4.6 and the conclusion is drawn at the end of the chapter.

4.1 Problem Description

The distribution network considered comprises of a single supplier, 0 and a fleet of homogeneous vehicles delivering a single product to a set of N customers over a given planning horizon T. Each vehicle performs one full route beginning at the depot, delivering to customers and then return to the depot. Note that the same vehicle will not be assigned for another trip in the same period. Outsourcing the vehicles to a third party company are not considered. The demand, d_{it} for each customer i in period t is deterministic. The customers' positions are geographically scattered surrounding the depot and they are located close to the depot.

It is assumed that sufficient amount of products is available at the supplier to fulfill customers demand throughout the planning horizon. There are two different situations considered that allows backorder decisions in this model. The situations are:

(i) Vehicle capacity is not enough to satisfy all the customers demand in a period. Given that outsourcing from other transportation services is not allowed due to the higher cost imposed.

(ii) If the saving in the transportation cost is higher when compared to the backorder cost imposed by a customer.

The aim of the study is to determine the best optimal routing by minimizing the overall transportation, inventory carrying and backordering costs.

4.1.1 Illustration Example of the IRPB

An illustrative example to describe the IRPB is given in this subsection. Two examples will explain the two different situations where backorder decision can happen. Figure 4.1, 4.2 and 4.3 illustrate the examples of the problem with the case of $N = 5$ suppliers for $T = 2$ periods, $V = 2$ number of vehicles. Assume that the locations of the supplier and customers are given as in the figures. The routes are from supplier (0) to customers and then back to the supplier. The customer demands in period 1 and 2 are $\langle 2,2,3,3,1 \rangle$ and $\langle 2,2,1,3,1 \rangle$ respectively.

Given that each vehicle has capacity, $q = 5$ and Figure 4.1 illustrates the first backorder situation where vehicle capacity is exceeded for deliveries. Best routes found in period 1 are $0 \to 1 \to 3 \to 0$ (vehicle 1) and $0 \to 2 \to 4 \to 0$ (vehicle 2). The capacity for both vehicles 1 and 2 is $2 + 3 = 5$, respectivey. Note that customer 5 is not visited in period 1 as not enough vehicle capacity so the demand is backordered (fulfill in period 2). The routes and the capacity for vehicles in period 2 are $0 \to 1 \to 3 \to 5 \to 0$ (vehicle 1) with capacity $2 + 1 + 1 + 1 = 5$ (including 1 unit demand of customer 5 in period 1) and $0 \to 2 \to 4 \to 0$ (vehicle 2) with $2 + 3 = 5$.

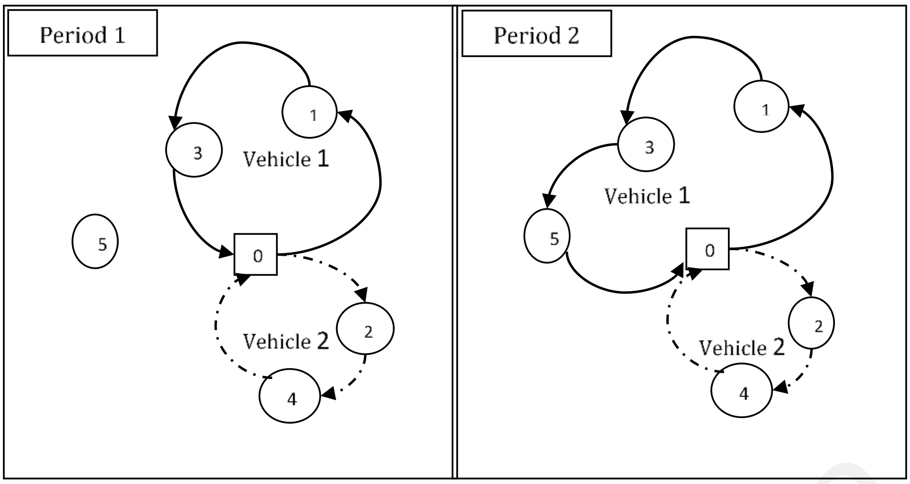


Figure 4.1: An example of the first backorder situation.

Figure 4.2 and 4.3 presents the second situation where the savings in the transportation is higher than the backorder cost. The demand for this example are as previous, but assume that each vehicle has a capacity $q = 6$. Let say that the optimal routes for both periods are given as in Figure 4.2, where all customers are visited in both periods without any inventory and backorder. Assume that the travel cost per unit distance is 1.The optimal routes are $0 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 0$ and $0 \rightarrow 2 \rightarrow 5 \rightarrow 0$ with total distance $5 + 3 + 1 + 1.5 = 10.5$ unit and $1 + 0.5 + 1.5 = 3$ unit, respectively. The corresponding delivery quantities are 6 and 5 for period 1; and 4 and 5 for period 2. The optimal transportation cost for period 1 and period 2 is $2(10.5 + 3) = 27.0$. To explain the situation, let the penalty cost of delayed order (backorder cost) for each customer is given by <0.90, 1.20, 1.80, 1.30, 20.10>.

Let say that the demand for customer 1 in period 1 is served in period 2. The new optimal solution is given as in Figure 4.3 with a distance of depot to customer 3 is 1 unit. The new distance for vehicle 1 in period 1 are $0 \rightarrow 3 \rightarrow 5 \rightarrow 0$ with $1 + 1 + 1.5 = 3.5$. The corresponding capacity is 4. Since customer 1 is not visited in period 1, the backorder cost of 0.90×2=1.8 is incurred.

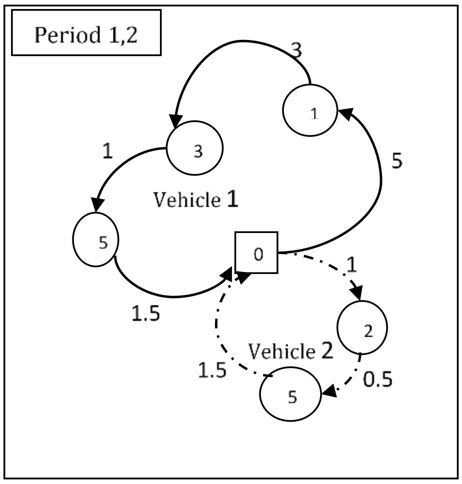


Figure 4.2: Optimal Solution without any inventory and backorder for the second scenario.

In period 2, the sequence of customer visited for vehicle 1 does not change, however the capacity increases that is $4 + 1 + 1 = 6$ because of the inclusion of the demand for customer 1 of 2 units. The new optimal cost (transportation and backorder) is 21.8 , that is the distance cost in period 1, $3.5 + 3 = 6.5$ plus the backorder cost 1.8; plus the distance cost in period 2, $10.5 + 3 = 13.5$. The saving is $27.0 - 21.8 = 5.2$. It illustrates that it is more beneficial to backorder customer 1 demand in period 1 as the overall savings is higher although there is enough capacity to fulfill the quantity in period 1.

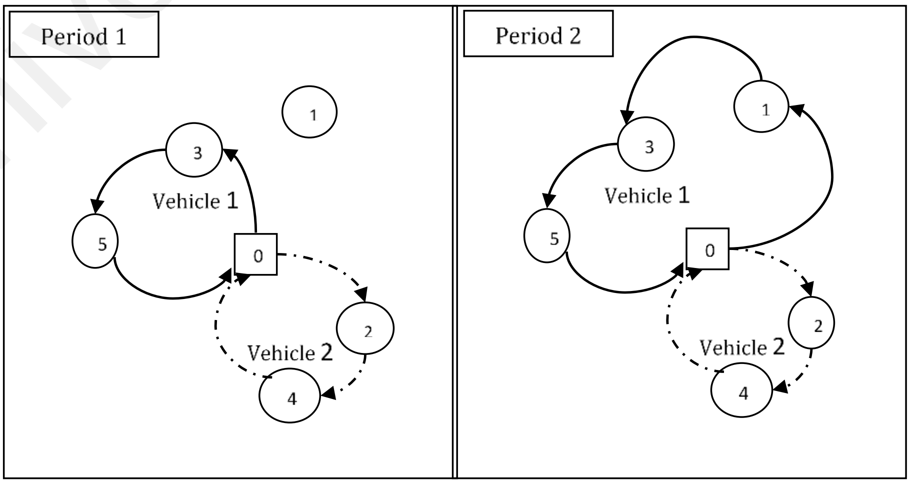


Figure 4.3: An example of the second backorder situation.

4.2 Assumptions

There are a few assumptions made in the IRPB model. The assumptions are presented as below.

a. Backordering is allowed.
b. Split delivery is not allowed. Customer cannot be visited by more than one vehicle.
c. Quantity transported must not exceed the vehicle maximum capacity.
d. Limited number of capacitated heterogeneous vehicles available for delivering the product in each period.
e. No option to schedule an emergency delivery and outsourcing to other transportation service is not allowed (as this will lead to a higher cost).
f. Supplier and customer have an agreement that products can be delayed with certain penalty charges.
g. Rolling planning horizon is adopted.

4.3 Problem Formulation

The standard formulation is as given in Abdelmaguid et al. (2009) for IRPB. The notations used throughout this chapter are as follows.

Indices

i, j	Indices for customers. Supplier is denoted by 0
t	Index for periods
v	Index for vehicle
N	Set of customers
T	Set of periods in the planning horizon
V	Numbers of capacitated heterogeneous vehicle

Parameters

C_i	Maximum inventory capacity of each customer i
c_{ij}	Distance between customer i and j, which satisfies the triangle inequality
d_{it}	Demand of customer i in period t
f_{vt}	Fixed cost for using vehicle v at period t
h_i	Holding cost for customer i
π_i	Backorder penalty cost for customer i
q_v	Vehicle capacity for vehicle v

Variables

x_{ijt}^v	Binary decision, 1 if vehicle v travels from customer i to customer j in period t; 0 otherwise
y_{ijt}^v	Amount transported on vehicle v in period t, correspond to its x_{ijt}^v
I_{it}	Inventory level at customer i at the end of period t
B_{it}	Backorder level at customer i at the end of period t

The formulation:

$$\min Z = \underbrace{\sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{v=1}^{V} f_{vt}x_{0jt}^v}_{(A)} + \underbrace{\sum_{t=1}^{T}\sum_{i=0}^{N}\sum_{\substack{j=0 \\ j\neq i}}^{N}\sum_{v=0}^{V} c_{ij}x_{ijt}^v}_{(B)} + \underbrace{\sum_{t=1}^{T}\sum_{i=1}^{N} h_i I_{it}}_{(C)} + \underbrace{\sum_{t=1}^{T}\sum_{i=1}^{N} \pi_i B_{it}}_{(D)} \tag{4.1}$$

subject to:

$$\sum_{\substack{j=0 \\ j\neq i}}^{N} x_{ijt}^v \leq 1 \qquad i = 0, \dots, N; t = 1, \dots, T; v = 1, \dots, V \tag{4.2}$$

$$\sum_{\substack{k=0 \\ k\neq i}}^{N} x_{ikt}^v - \sum_{\substack{l=0 \\ l\neq i}}^{N} x_{lit}^v = 0 \qquad i = 0, \dots, N; t = 1, \dots, T; v = 1, \dots, V \tag{4.3}$$

$$y_{ijt}^{v} - q_v x_{ijt}^{v} \leq 0 \qquad i,j = 0,\dots,N; i \neq j;\ t = 1,\dots,T; v = 1,\dots,V \tag{4.4}$$

$$\sum_{\substack{l=0 \\ l\neq i}}^{N} y_{lit}^{v} - \sum_{\substack{k=0 \\ k\neq i}}^{N} y_{ikt}^{v} \geq 0 \qquad i = 1,\dots,N; t = 1,\dots,T; v = 1,\dots,V \tag{4.5}$$

$$I_{it} - B_{it} = I_{it-1} - B_{it-1} + \sum_{v=1}^{V}\left(\sum_{\substack{l=0 \\ l\neq i}}^{N} y_{lit}^{v} - \sum_{\substack{k=0 \\ k\neq i}}^{N} y_{ikt}^{v}\right) - d_{it}$$

$$i = 1,\dots,N; t = 1,\dots,T \tag{4.6}$$

$$I_{it} \leq C_i \qquad i = 1,\dots,N; t = 1,\dots,T \tag{4.7}$$

$$I_{it} \geq 0 \qquad i = 1,\dots,N;\ t = 1,\dots,T \tag{4.8}$$

$$B_{it} \geq 0 \qquad i = 1,\dots,N;\ t = 1,\dots,T \tag{4.9}$$

$$y_{ijt}^{v} \geq 0 \qquad i,j = 0,\dots,N; i \neq j;\ t = 1,\dots,T; v = 1,\dots,V \tag{4.10}$$

$$x_{ijt}^{v} = \{0,1\} \qquad i,j = 0,\dots,N; i \neq j;\ t = 1,\dots,T; v = 1,\dots,V \tag{4.11}$$

The objective function (4.1) comprises of transportation cost (vehicle fixed costs (A) and variable travel costs (B)), the inventory carrying cost (C) and backorder costs (D). Constraint (4.2) is to make sure that a customer is not visited by a vehicle more than once in a period. Constraint (4.3) is to ensure route continuity. Constraint (4.4) is to make sure that the amount transported in a vehicle must not exceed the vehicle capacity if there is a trip from customer i to customer j. Constraint (4.5) is the subtour elimination constraint. Constraint (4.6) is the inventory balance equation. Constraint (4.7) is to make sure that the inventory level of customers does not exceed the maximum level. Constraints (4.8) – (4.10) are the non-negative constraints for the inventory level, backorder level and amount transported. Constraint (4.11) is the binary decision variable.

4.4 Artificial Bee Colony for IRPB (*ABCIRPB*)

The ABC algorithm used for solving IRPB is modified from the previously developed algorithm. The *ABCIRPB*, the ABC for IRPB differs from *ABCIRP* mainly in the inventory management aspect where *ABCIRPB* is able to handle backorders decision as an addition to the inventory decision and transportation. This changes the structure and inner process of the *ABCIRPB*

The main differences are in the solution representation, inventory updating mechanism and the neighborhood operator used. The initial solution (initial food sources) generated containing routing, inventory and backorder decisions. A modified Giant Tour (GT) procedure is performed to attain the routing part of the initial solution. GT procedure is proven to be powerful to find routes with minimum distance, however GT procedure does not have the ability to control the number of vehicles used. Despite that, it is suitable for *ABCIRPB* as the modified GT procedure is adjusted to cope with fixed number of vehicles, and eventually decide on the customers to backorder. The detail of the modified GT is explained and illustrated in subsection 4.4.1.1. The solution is then improved using a pre-optimization procedure. The initialization steps are covered in **STEP 1**. The parameters used to control the intensification and diversification were declared in **STEP 2**.

The phases of employed, onlooker and scout bees are explained in detail in **STEP 3** and **STEP 4.** The employed bee phase is described in details in **STEP 3.1**. An inventory updating mechanism to handle both inventory and backorder decisions is introduced; named *random exchanges* and *guided exchanges*. Both exchanges are explained with illustrations in subsection 4.4.2.

The onlooker bee will then select the best food source (solution) using a non-bias selection method Stochastic Universal Sampling (SUS) (Baker, 1987), as in Chapter 3. The onlooker bee phase is covered in **STEP 3.2** and **3.3**, where the routing of the selected food source is improved by using swap $1-0$, 2-opt (Lin, 1965) and 2-opt* (Potvin & Rousseau, 1995). **STEP 4** is the scout bee phase where the current food source is replaced by a randomly generated food source when the algorithm reaches the maximum exploitation limit (that is when the exploitation of the food source is exhausted).

The *ABCIRPB* algorithm is given as follows:

STEP 1 Initialization Phase

1.1 Generate n number of solutions (food sources). Each solution indicates to visit or not to visit a customer for each period, which from these the delivery quantity is obtained. Preprocessing is carried out to eliminate the delivery quantity that exceeds vehicle capacity as split shipment is not allowed.

1.2 Denote each food source as z_i, $i = 1, \dots, n$. Using the delivery quantity, construct the initial tour for each food source, z_i by modifying the Giant Tour Procedure (Imran et al., 2009) to suit the limited number of vehicles, in each period. Evaluate the fitness value for each food source; $f(z_i), i = 1, \dots, n$.

1.3 Do pre-optimization with 2-opt (Lin, 1965) and 2-opt* (Potvin & Rousseau, 1995) for each of the food sources. Assign each employed bee to a food source.

STEP 2 Set $iteration = 0$. Declare the value of $LIMIT$ (controls of exploitation a food source) and $MAXITER$, the maximum number of iterations. Set the indicator associate with $LIMIT$ as $l_1 = l_2 = \cdots = l_n = 0$.

STEP 3 Repeat the following until the stopping condition, $MAXITER$ is met.

3.1 Employed Bee Phase (Inventory/Backorder Updating Mechanism)

a. For each food source, z_i. Select two consecutive period t and $t+1$.

b. Apply the inventory/backorder updating mechanism, either *random exchanges* or *guided exchanges*.

c. Assign the new food source found, as $\bar{z}_i$.

d. *If* $f(\bar{z}_i) < f(z_i)$; replace the old food source with a new food source, $z_i \leftarrow \bar{z}_i$ and set $l_i = 0$. *Else* set $l_i = l_i + 1$.

3.2 Onlooker Bee Phase (Route Improvement Mechanism)

a. Set $G_i = \emptyset$, $i = 1, \dots, n$, where G_i is the set of neighbor solutions of food source i.

b. For each onlooker bee.

   i. Select a food source, z_i, using a stochastic universal sampling (SUS) selection method (Baker, 1987).

   ii. Apply a neighborhood operator, swap $1-0$, 2-opt (Lin, 1965) and 2-opt* (Potvin & Rousseau, 1995) on selected z_i; resulting $\tilde{z}_i$.

   iii. $G_i = G_i \cup \tilde{z}_i$.

c. For each food source z_i and $G_i \neq 0$.

   i. Set $\hat{z}_i \in argmin_{\tilde{z} \epsilon G_i} f(\tilde{z})$.

   ii. *If* $f(\hat{z}_i) < f(z_i)$; replace the old food source with the new one; $z_i \leftarrow \hat{z}_i$ and set $l_i = 0$. *Else* set $l_i = l_i + 1$.

3.3 Scout Bee Phase

For each food source, z_i. *If* $l_i = LIMIT$, replace z_i with a randomly generated solution.

$iteration = iteration + 1.$

STEP 4 Output is the best food source found so far.

All the steps are also summarized as a flow chart in Figure 4.4.

4.4.1 Solution Representation and Initialization Phase (STEP 1)

The solution representation used in this problem is in matrix form similar to the previous chapter. However, the decoding from the binary matrix results in the initial inventory and backorder decisions.

An example of 5 customers and 5 periods is considered as shown in Figure 4.5 where Figure 4.5 (a) illustrates the demand matrix whilst Figure 4.5 (b) shows the binary matrix where the columns indicate the periods and the rows indicate the customers. Referring to the example below, customer 1 in period 4, customer 3 in period 2 and period 5, customer 4 in period 3 and customer 5 in period 1 is not visited. Figure 4.5 (c) shows the delivery matrix. Figure 4.5 (d) and 4.5 (e) are the resulting inventory and backorder matrices.

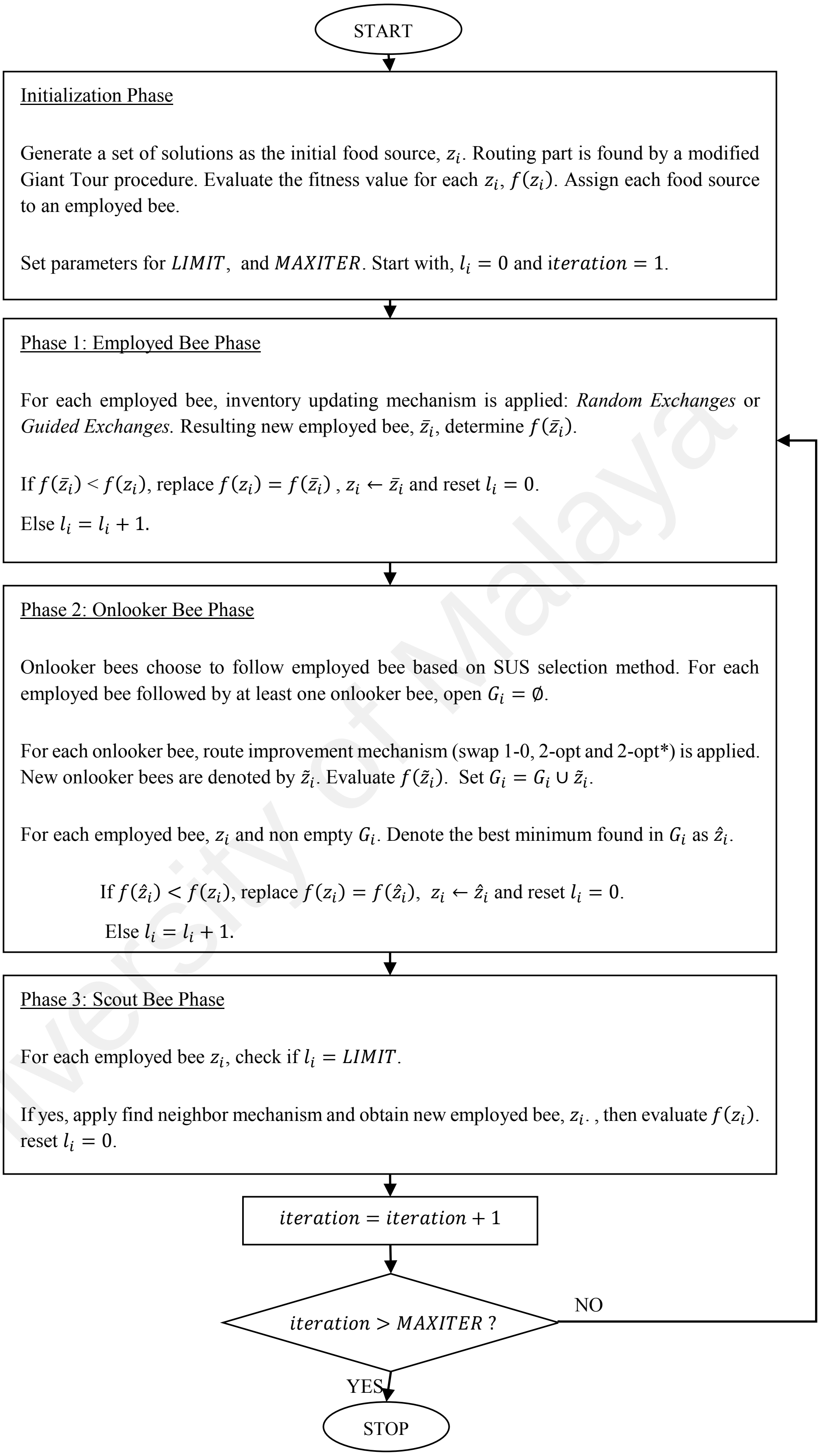


Figure 4.4: Flows of the *ABCIRPB* algorithm.

		Period				
		1	2	3	4	5
Customer	1	34	14	25	41	6
	2	17	34	23	12	18
	3	16	24	12	12	19
	4	37	39	20	27	28
	5	7	39	24	23	38

Figure 4.5 (a): Demand Matrix.

		Period				
		1	2	3	4	5
Customer	1	1	1	1	0	1
	2	1	1	1	1	1
	3	1	0	1	1	0
	4	1	1	0	1	1
	5	0	1	1	1	1

Figure 4.5 (b): Binary Matrix.

		Period				
		1	2	3	4	5
Customer	1	34	14	66	0	6
	2	17	34	23	12	18
	3	40	0	12	31	0
	4	37	59	0	27	28
	5	0	46	24	23	38

Figure 4.5 (c): Delivery Matrix.

		Period				
		1	2	3	4	5
Customer	1	0	2	41	0	0
	2	0	0	0	0	0
	3	24	0	0	19	0
	4	0	20	0	0	0
	5	0	0	0	0	0

Figure 4.5 (d): Inventory Matrix.

		Period				
		1	2	3	4	5
Customer	1	0	0	0	0	0
	2	0	0	0	0	0
	3	0	0	0	0	0
	4	0	0	0	0	0
	5	7	0	0	0	0

Figure 4.5 (e): Backorder Matrix.

Figure 4.5: Solution Representation for *ABCIRPB*.

4.4.1.1 Modified Giant Tour Procedure

The Giant Tour (GT) procedure is to determine the route for the delivery quantities obtained and it is modified as *IRPB* considers a limited number of vehicle for each period. Because of the model allows for backorder, there might be a customer(s) that cannot be served in the current period. Customer(s) that cannot be served in period t will be serve in the subsequent period, $t+1$.

The algorithm for modified Giant Tour Procedure is given as below. Denote z_t as the list of customers visited in period t and q_z_t as its corresponding delivery quantities.

1. Set period $t = 1$. While $t \leq T$ do the followings.
   (a) Construct a giant tour for z_t. Start by connecting the nearest customer i in z_t to the supplier, and set customer i as the current node. Next connects the nearest customer j in z_t (nearest to customer i) to the current node, $j \neq i$. Set customer j as the current node. Repeat until all customers in z_t are connected.
   (b) Find the corresponding cost network for the giant tour found in (a) and partitioned the network using Dijkstra's algorithm (Dijkstra, 1959) to get m vehicles with the best routing; m is a positive integer.

      If $m > V$, where V is the number of vehicles available in period t.

         Proceed to step (c).

      Else

         Accept the best routing found.
   (c) Denote $list = \{\}$. Select and keep the vehicles, V with the most vehicle utilization (denoted as $vbest_V$ with vehicle load, $load_V$) and collapse the remaining vehicle(s) and assign their customers to the $list = \{k\}$; $k = 1..K$. Find the excess capacity in each of $vbest_V$, rem_V is calculated as maximum vehicle capacity, q_v minus $load_V$.
   (d) If $list \neq \{\}$. Repeat the following until $rem_V = 0$ or $list = \{\}$.

      For all customers in $list$, get its corresponding delivery, y_k.

      For all vehicles available 1 until V. Set $increase_{V=1,2} = 0$

      (i) If $rem_V > y_k$. Do reinsertion.

         Do best insertion. Update the increase in distance of $increase_V$.

      (ii) Else $rem_V < y_k$. Do partial reinsertion.

         (A) Do best insertion. The quantity delivery of customer k inserted is exactly rem_V.

(B) The remaining amount of customer k is backordered, $(y_k - rem_V) \times \pi_k$.

(C) Update $increase_V$ as the total increase in the distance and backordered amount in (B).

Choose the minimum $increase_V$ to be inserted in $vbest_V$.

(e) If $list \neq \{\}$. If *list* is still not empty.

Bring $list$ to period $t+1$. Update z_{t+1} and q_z_{t+1}

Accept $vbest_V$ found at time t.

Increase period $t = t + 1$.

2. If $t > T$ and $list \neq \{\}$. Do

(a) Best reinsertion (or Partial insertion) backward from $t = T \ldots 1$.

Note that as it is rolling planning horizon, backorders in the last period are transferred to the demand for the first period in the next planning horizon. The modified GT procedure produces an initial solution containing the delivery quantities and the routing for all customers in all periods, explicitly inventory and backorder decisions.

(a) ***An Illustration Example of the Modified Giant Tour Procedure***

Consider the previous example given in Subsection 4.4.1 comprising 5 customers and 5 periods in order to explain the procedure. The delivery matrix, demand matrix, inventory and backorder matrices used are as presented in Figure 4.5. Additional information needed is that the number of vehicles available is 2, each with capacity 75. Starting with period $t = 1$, all the customers are visited in period t except for customer 5 (refer to Figure 4.5 (b)), hence $z_t = \{1,2,3,4\}$ and the corresponding delivery quantities $q\_z_t = \{34,17,40,37\}$. Let say that the giant tour found in period 1 is $0 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow$

1. Then the network is partitioned based on the cost network of the GT using Dijkstra's algorithm. The partition produces $m = 3$ routes with minimum routing, it is natural the capacities of vehicles are not full.

The 3 initial routing in period 1 is given as in Figure 4.6. As only 2 vehicles are available in period 1, the minimum load is chosen to collapse (labeled X) and the customer on the selected vehicle (labeled X) is put in the $list = \{1\}$ with its corresponding delivery quantity $y_1 = 34$ (step 1(c)). The 2 highest utilization of vehicles are maintained and considered as $vbest_1$ and $vbest_2$. The vehicles with $rem_V > 0$ is considered for reinsertion of the customer in the $list$. Both vehicles are examined using the best reinsertion to minimize the increase in the distance. For partial reinsertion, the increase is estimated as the product of backorder amount and the increase in the distance.

From Figure 4.6, the $rem_1 = 38$ for $vbest_1$ whilst $vbest_2$, the $rem_2 = 18$. This indicates that the available quantity for $V = 1$ exceeds the delivery quantity of the collapsed vehicle; $rem_1 > y_1$. Hence step 1(d)(i) is applied where reinsertion of customer does not violate vehicle capacity. However, for when $V = 2$; $rem_2 < y_1$, step 1(d)(ii) is applied where only partial demand can be fulfilled and the remaining, $(y_1 - rem_2) = (34 - 18) = 16$ is considered to be backordered. It is obvious that partial reinsertion cost, $increase_2$ is always higher than the full reinsertion $increase_1$ as $increase_2$ includes the approximation of if 16 units of customer 1 is backordered amount multiply by the increase in the distance of $V = 2$. The final routings for period 1 are given by $0 \rightarrow 1 \rightarrow 4 \rightarrow 0$ and $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ with vehicle load of 71 and 57 respectively.

The initial routing for period 2 is given as in Figure 4.6 where vehicle 2 is collapsed and the corresponding customer 2 is put in the list, $list = \{2\}$. Note that the delivery quantity of customer 2 can partially reinserted in either vehicle. Let say that after calculation of the best insertion, $vbest_2$ is chosen to serve customer 2, with partial fulfillment of 16 units of the delivery quantity. Because $list = \{2\}$ is still not empty with the excess delivery quantity of $y_2 = 18$, we proceed to step 1(e), where the unfulfilled demand of customer 2 is served in the subsequent period, $t + 1 = 3$.

Note that the quantity delivery for customer 2 in period 3 includes the existing delivery plus an additional 18 units from period 2, resulting in 41 units (red bold). This is illustrated in Figure 4.6 (period 3). The same applies in period 4 where the delivery for customer 5 in period 3 is backordered and delivered in period 4. Once the initial solution for all periods is attained, pre-optimization is done using 2-opt* and 2-opt.

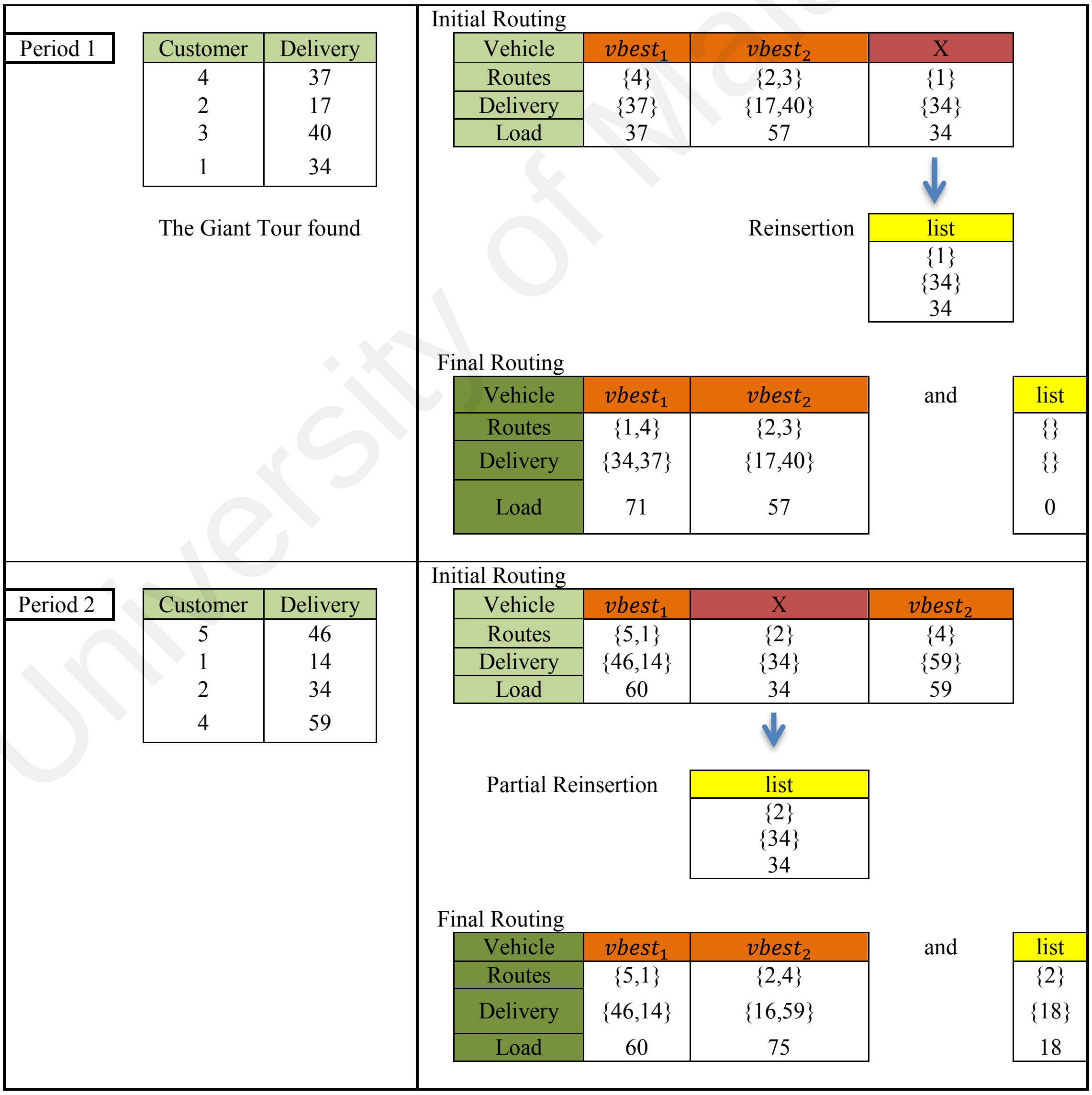


Figure 4.6: Illustration of the Modified Giant Tour Procedure.

Period 3

Customer	Delivery
5	24
1	66
2	**41**
3	12

Initial Routing

Vehicle	X	$vbest_1$	$vbest_2$
Routes	{5}	{1}	{2,3}
Delivery	{24}	{66}	{41,12}
Load	24	66	53

list
{5}
{24}
24

Partial Reinsertion

Final Routing

Vehicle	$vbest_1$	$vbest_2$
Routes	{1}	{2,3,5}
Delivery	{66}	{41,12,22}
Load	66	75

and

list
{5}
{2}
2

Period 4

Customer	Delivery
5	**25**
2	12
3	31
4	27

Final Routing

Vehicle	1	$vbest_2$
Routes	{5}	{2,3,4}
Delivery	{25}	{12,31,27}
Load	25	70

and

list
{}
{}
0

Period 5

Customer	Delivery
5	38
1	6
2	18
4	28

Final Routing

Vehicle	$vbest_1$	$vbest_2$
Routes	{5,1}	{2,4}
Delivery	{38,6}	{18,28}
Load	44	46

and

list
{}
{}
0

Figure 4.6, Continued.

4.4.2 Inventory/Backorder Updating Mechanism (STEP 3.1)

The employed bee phase embeds two different inventory updating, *random exchange* and *guided exchange* and both exchanges involve exchanging the amounts (delivery quantities) of customers between two consecutive periods. The exchanges are backward transfer and forward transfer, where backward transfers the amount from period t to period $t-1$, while the forward transfers the amount from period t to period $t+1$. Note that each backward transfer must be followed by a forward transfer to ensure the feasibility of the solution, because of the limited number of vehicles available in each period.

ABCIRPB that embeds *random exchange* is denoted as *ABCRX.* While ABC with *guided exchange* is denoted as *ABCGX.*

4.4.2.1 Random Exchange

Random *exchange* selects any possible amount of exchange between any two random consecutive periods. However, as one customer cannot be served by different vehicles in the period t (split delivery is not allowed), a swap procedure between vehicles in that period is performed if the exchange results in splitting the deliveries. The procedure merges the same customer's delivery and find other possible customer to swap. The steps for *random exchange* updating mechanism are given below.

1. Randomly choose 2 consecutive periods, let say period A and period B.
2. Randomly choose 1 vehicle from each period selected, $veh_A$and veh_B. Denote the non selected vehicle from each period as, $vehx_A$ and $vehx_B$.
3. Carry out the exchange of customers between the two vehicles $veh_A$and veh_B. Feasibility of the exchange is assured by the vehicle's capacity. Let say that PX is the number of all possible exchange between customers in veh_A and veh_B. Keep and denote each vehicle changed as $vehc_A$ and $vehc_B$.
4. Set $EX = \{\}$.
5. For all exchanges from step 3, start with 1 until PX
   (a) If the same customer, $cust_e$ exists in another vehicle for the same period. (i.e. $cust_e$ exists in $vehc_A$ and $vehx_A$).
      i. Do *merge and swap*: try merging $cust_e$ delivery quantity from $vehc_A$into $vehx_A$. If the merger resulted in violation of vehicle capacity of $vehx_A$, remove a possible (feasible) customer from $vehx_A$ and reinserted using the best insertion to $vehc_A$. Repeat the same procedure for period B.

ii. If the results from step (i) are feasible for both periods A and B, keep the exchange and place $vehc_A$, $vehc_B$, $vehx_A$ and $vehx_B$ in EX. Evaluate the change in the distance. If it is not feasible, discard the exchange.

(b) Else

i. Accept the exchange, $vehc_A$ and $vehc_B$ are included in EX. Evaluate the distance.

6. The output is the best minimum distance from EX, then update the vehicles change.

The steps of *random exchange* algorithm are illustrated in Figure 4.7, 4.8 and 4.9. Consider 5 customers with 2 available vehicles, each with a capacity, $q = 75$. Period A and period B are the two randomly selected periods, where each period is visited by 2 vehicles. Let say that the randomly selected vehicle from each period is given by; veh_A(shaded in blue) and veh_B (shaded in orange) and the non-selected vehicle; $vehx_A$ and $vehx_B$ are as in Figure 4.7.

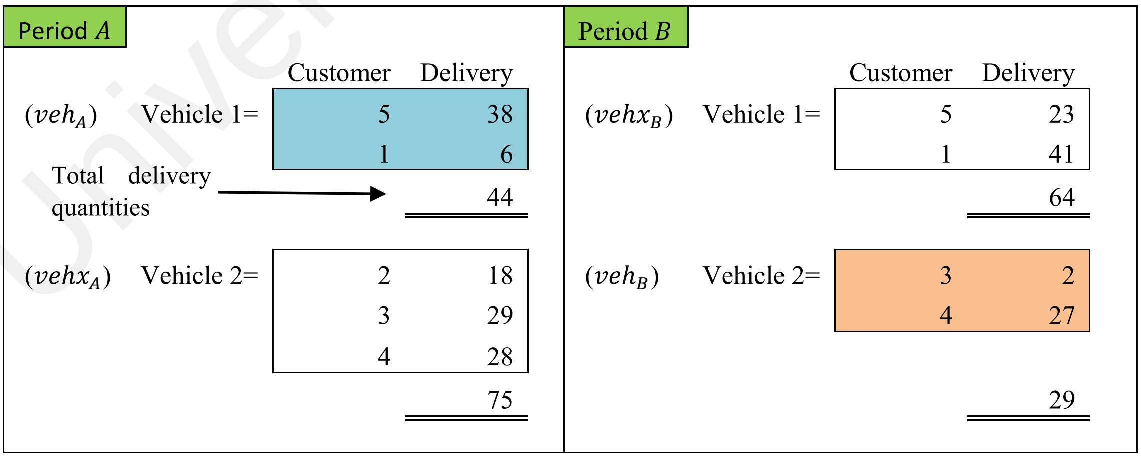


Figure 4.7: Step 1 and Step 2 from Random Exchange.

The list of all possible exchanges, PX between veh_A and veh_B (step 3) is given as in Figure 4.8. In this example $PX = 2$.

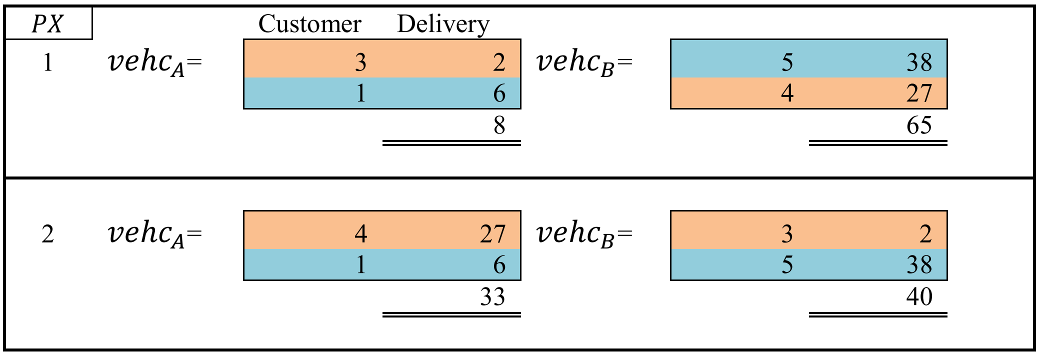


Figure 4.8: Step3, All possible exchange from veh_A and veh_B.

The same step (step 5 (a)) is applied in period B and the whole process is repeated for all other possible changes. See Figure 4.9. The best exchange (in term of distance) among all possible exchange is selected. Note that this procedure may result in a decrease or increase in the inventory or backorder.

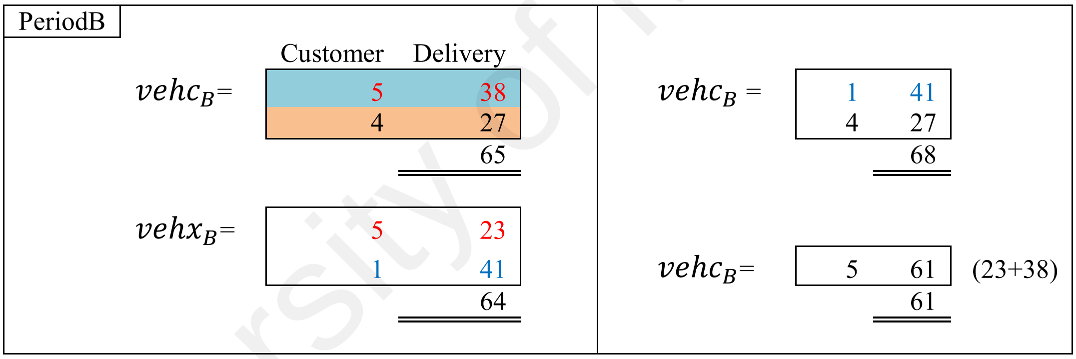


Figure 4.9: Final of *random exchange* after swap and merge for period B.

4.4.2.2 Guided Exchange

Guided exchange is similar to the *random exchange* except that the exchange is guided with the aim of reducing backorders. This exchange comprises of a backward transfer to reduce the backorder and followed by a forward transfer to reduce the inventory. This results in a total reduction of both backorder and inventory. Detail steps of the *guided exchange* are given below.

1. For all period $t = 1 \ldots T$. Start with $t = 1$.

2. Find all customers with backorder in the current period t. Denote each customer as $cust_{bit}$, its vehicle veh_{vt} and the corresponding backorder amount as B_{it}. Set $EX = \{\}$.
3. For each $cust_{bit}$ check whether it is Partial Delay or Full Delay (If $cust_{bit}$ is visited in period t, it is Partial Delay, else Full Delay).
   a. If Full Delay (not visited in period t)

      Identify the customer $cust_{bit}$ served in the succeeding period, denote as $cust_{bit_s}$ in which the delivery quatity includes the backorder B_{it} and the corresponding vehicle as veh_{bt_s}.

      For each veh_{vt}, identify customers with inventory and denotes as $cust_{Iit}$, and its corresponding inventory quantity as I_{it}. Insert B_{it} in vehicle veh_{vt} and I_{it} in vehicle veh_{bt_s}.

      If exchange is possible (does not violate vehicle capacity)

         Do the exchange with best insertion method. Do *merge and swap* to avoid split delivery.

         Keep the solution in EX.

      Else find the smallest between backorder value and inventory value, $amountX = \min\{I_{it}, B_{it}\}$.

         Apply the exchange with $amountX$ as the transfer quantity. Do *merge and swap* to avoid split delivery. Keep the solution in EX.

   b. Else Partial Delay

      Denote the vehicle served $cust_{bi}$ as veh_{vt}.

Denote $cust_{bi}$ served in the next succeeding period as $cust_{bit_s}$, and the serving vehicle as veh_{bt_s}.

Find customer(s) with inventory quantity served by veh_{vt} denote as $cust_{Iit}$, and its corresponding inventory quantity as I_{it}. Do exchange between B_{it} and I_{it} (between $cust_{Iit}$ and $cust_{bit_s}$).

If exchange is possible

Do best insertion of $cust_{Iit}$ to the veh_{bt_s} and embed the amount B_{it} from $cust_{bit_s}$ to $cust_{bi}$ in veh_{vt}. Avoid split delivery by applying *merge and swap*.

Else find $amountX = \min\{I_{it}, B_{it}\}$.

Apply the exchange with $amountX$ as the transfer quantity. Do *merge and swap* to avoid split delivery. Keep the solution in EX.

4. Output is the best from EX, then update the vehicles change.

Figure 4.10, 4.11 and 4.12 illustrate the *guided exchange* process. An example using 5 customers and 2 vehicles available with 75 capacities each. Let say the current period is $t = 1$. Given that in period 1 the customer i that have backorders ($cust_{bi}$) are $i = 1, 3, 4$ each with the corresponding quantities, $B_{it} = B_{i1} = 29, 15, 37$ respectively. And customers with inventories ($cust_{Iit}$) are $i = 2, 5$ with inventory quantities (I_{it}) are $I_{21} = 57$ and $I_{51} = 63$. Given also is the demand for customers in period 1 and 2 are $\langle 34,17,16,37,7 \rangle$ and $\langle 14,34,24,39,39 \rangle$ respectively.

Start with $cust_{b1}$ with $B_{11} = 29$ is served by $veh_{vt} = veh_{11}$. The next succeeding period is $t_s = 2$, where customer $cust_{b1}$ is served in $veh_{bt_s} = veh_{b2}$. As $cust_{b1}$ is visited in the period, it is a partial delay. Then find customer in veh_{11} with inventories to be transferred forward, while backward transfer is from customer $cust_{b1}$ from veh_{b2}. Select $cust_{I51}$ with $I_{51} = 63$. These are given as in Figure 4.10.

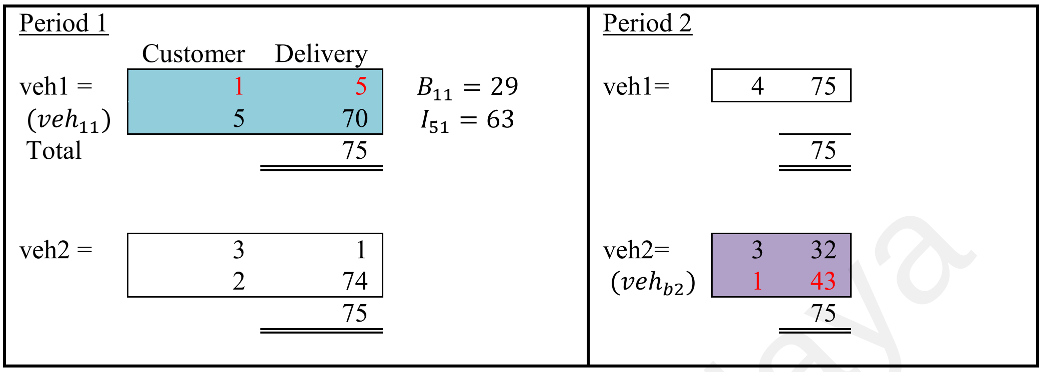


Figure 4.10: Routes and quantities for current period $t = 1$ and succeeding period, $t_s = 2$.

Do the exchange with backward transfer from $cust_{b1}$ in veh_{b2} to fulfill the backorder in the current period, with the exchange of forward transfer $I_{51} = 63$ in veh_{11}, as in Figure 4.11. This will reduce the backorder and inventory cost simultaneously. However in this case, the exchange is not feasible as the exchange exceeds the vehicle capacity in veh_{b2}.

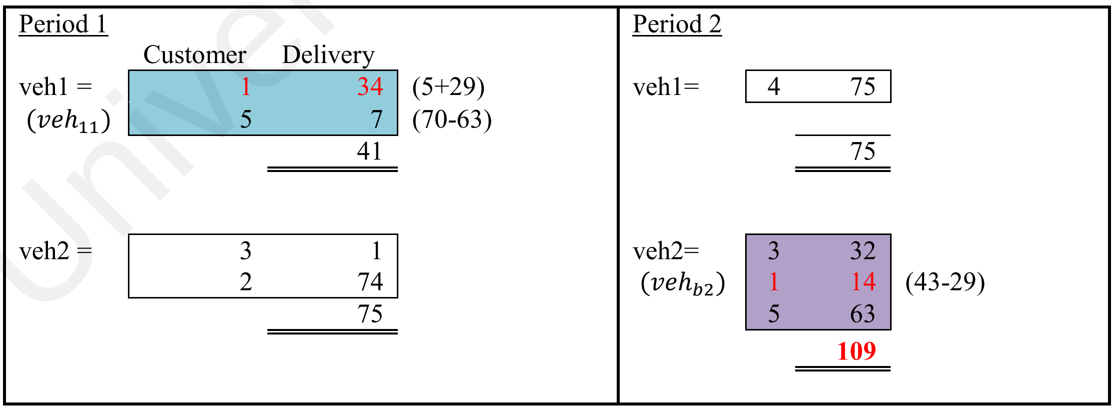


Figure 4.11: Direct exchanges to fulfill the backorder amount of customer exceeds the vehicle capacity.

Thus, the amount of the exchange is based on the minimum between the corresponding selected inventory and backorder, $amountX = \min\{I_{it}, B_{it}\} = \min\{I_{51}, B_{11}\} = \min\{63, 29\} = 29$. The exchange is shown as in Figure 4.12 and the solution is kept in

EX. Note that if the value of inventory is selected as the agreed amount, the backorder is partially fulfilled. The process is continued for other customers with backorder and for each customer the best exchange is chosen and included in EX.

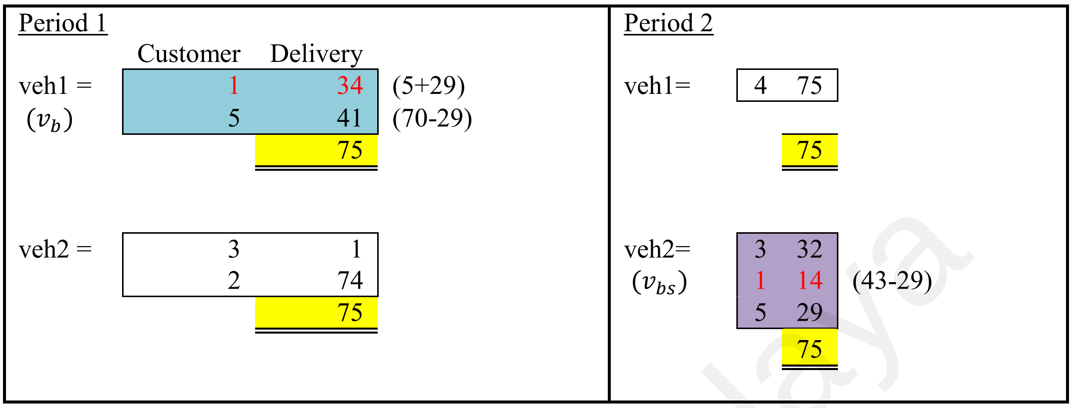


Figure 4.12: Guided exchange with agreeable value.

4.4.3 Neighborhood Operator (STEP 3.3b)

In the neighborhood operator, swap, 2-opt* (Potvin & Rousseau, 1995) and 2-opt (Lin, 1965) heuristics aiming to improve the routes were embedded.

4.5 Computational Results and Discussions

4.5.1 Datasets

ABCRX and *ABCGX* are tested on 135 datasets obtained from Abdelmaguid et al. (2009). The datasets are to test the two different situations explained earlier, where 60 datasets are designed for the Scenario 1 such that it is not beneficial to do backorder decisions, and the other 75 datasets are designed for Scenario 2 and 3 where it provides condition where backorder decision is more economical.

The datasets simulate the real situations in the industries faced by the manufacturing companies. The datasets mimic the network where customers' location is assumed to be in different major cities. The dataset consists of 5, 10, 15, 20, 25 and 30 customers with 5 and 7 periods.

The dataset is segregated according to the number of customers and the number of periods. The dataset is labeled as the scenario number S, the number of customers N, the number of periods T and the number of vehicles V. Each dataset with combination of N, T and V has 5 different replicates. The designated name for the dataset is read as S-$NNTV$-$\#$. For example, dataset 1-1572-3 referred to scenario 1 with 15 customers, 7 periods, 2 vehicles and replicate number 3.

The following parameters are fixed for each scenario. The depot is located in the middle, surrounds by customers with locations within a square of 20 x 20 distance units. Fixed vehicle cost is set at 10. Scenario 1 fixed the travel cost per unit distance to 1 and the customer demands are set from 25 to 50 per day. In scenario 2, the travel cost per unit distance is set to 2 and customers demand are set from 5 to 50 per day. While in Scenario 3, the travel cost per unit distance is set to 1 and the customers demand are set from 0 to 25. Summary of the criteria for all of the scenarios is presented in Table 4.1.

Table 4.1: Criteria of the Dataset according to Scenario.

Criteria	**Scenario 1**	**Scenario 2**	**Scenario 3**
Location	Customers within 20x20 distance units. Depot in the middle		
Maximum Inventory Level at Customer	120	120	120
Fixed Vehicle Cost	10	10	10
Travel Cost per Unit Distance	1	2	1
Demand Range	[25,50]	[5,50]	[0,25]
Vehicle Capacity	500,1000,1500	150,300,450	300,350,400
Customer Number	5,10,15	5,10,15	20,15,30
Number of Vehicle	1,2	1,2	2
Planning Horizon	5,7	5,7	7
Number of Datasets	60	60	15

4.5.2 Results and Discussions

All algorithms are coded using MATLAB 8.1 and run in 8GB RAM computer with processor 3.1 GHz. Each dataset was run 10 independent times. Detail results of each of the 10 runs are given in Appendix B. The number of bees in the ABC comprises of 50 bees with 25 employed bees and 25 onlooker bees. The 25 employed bees consist of 20 randomly generated bees, 1 bee with zeroes holding cost (all customers are visited in all periods, before the implementation of the modified GT procedure) and 4 bees from the planned delivery heuristic (PLNDLV) developed by Abdelmaguid et al. (2009). The maximum number of iteration, MAX_ITER is set to 300.

$ABCRX$ and $ABCGX$ are tested on all 3 different scenarios. Abdelmaguid et al. (2009) proposed a heuristic, Estimated Transportation Costs Heuristic (ETCH) where the IRPB is decomposed by employing dynamic programming and breadth-first search into backorder and inventory decisions.

Results for all scenarios are presented in Table 4.2, 4.3 and 4.4 respectively. The best results for both $ABCRX$ and $ABCGX$ are compared to the best results of 4 variants of ETCH given in Abdelmaguid et al. (2009), together with the upper bound (UB) and lower bound (LB) found using AMPL-CPLEX given in Abdelmaguid et al. (2009). The bests of all 4 ETCH variants are combined and denoted as ETCH. Bounds (UB and LB) with the asterisk symbol (*) denote the optimal solution where $UB = LB$ found. The percentage different $(\%\Delta)$ between $ABCRX$, $ABCGX$ and the ETCH from Abdelmaguid et al. (2009) is also provided.

Scenario 1 presented in Table 4.2 shows that $ABCRX$ and $ABCGX$ embedded in ABC provides 7 better results out of 60 instances when compared to ETCH. Best results are highlighted in bold and result with light shades are the comparison between $ABCRX$ and $ABCGX$. It can be observed that $ABCRX$ performs better than $ABCGX$ with 47 instances obtaining better results as compared to 12 better results produced by $ABCGX$ and with 1 identical results. On average, the percentage different indicate that $ABCRX$ and $ABCGX$ is 91.46% and 89.36% is better or comparable to ETCH, respectively.

Table 4.2: Results for Scenario 1 comparing UB, LB, ETCH, *ABCRX* and *ABCGX*.

Dataset	UB	LB	ETCH	ABCRX	%Δ	ABCGX	%Δ
1-0551-1	205.84	205.84*	**205.84**	216.05	4.73	216.18	4.78
1-0551-2	150.74	150.74*	**150.74**	159.41	5.44	163.11	7.58
1-0551-3	186.6	186.6*	**186.6**	209.68	11.01	200.60	6.98
1-0551-4	200.8	200.8*	**204.3**	218.04	6.30	225.96	9.59
1-0551-5	184.8	184.8*	**185.35**	196.66	5.75	201.83	8.17
1-0571-1	278.96	278.96*	**281.81**	313.72	10.17	330.29	14.68
1-0571-2	268.68	268.68*	**272.98**	317.22	13.95	331.34	17.61
1-0571-3	273.07	273.07*	**273.07**	309.92	11.89	329.52	17.13
1-0571-4	312.25	312.25*	**349.49**	372.67	6.22	364.68	4.17
1-0571-5	310.98	310.98*	**314.04**	365.30	14.03	375.00	16.26
1-0552-1	212.41	205.11	**221.69**	227.27	2.46	236.25	6.16
1-0552-2	254.28	254.28*	**254.28**	272.14	6.56	285.45	10.92
1-0552-3	220.86	220.86*	**223.98**	229.9	2.58	251.66	11.00
1-0552-4	250.35	250.35*	254.83	269.27	5.36	***253.05***	-0.70
1-0552-5	235.09	233.33	245.92	**244.66**	-0.52	255.97	3.93
1-0572-1	319.22	302.88	336.38	**336.19**	-0.06	373.20	9.87
1-0572-2	289.15	274.02	**290.33**	301.95	3.85	318.38	8.81
1-0572-3	270.66	253.78	**271.71**	292.46	7.09	312.81	13.14
1-0572-4	278.68	258.79	**286.79**	286.86	0.02	313.42	8.50
1-0572-5	292.03	271.68	307.91	**297.96**	-3.34	331.11	7.01
1-1051-1	327.09	306.82	**326.97**	386.75	15.46	391.57	16.50
1-1051-2	286.17	251.17	**276.41**	317.32	12.89	330.21	16.29
1-1051-3	300.69	295.9	**300.69**	347.80	13.55	346.29	13.17
1-1051-4	291.13	260.2	**280.13**	334.73	16.31	326.11	14.10
1-1051-5	269.47	218.9	**249.63**	297.58	16.11	281.71	11.39
1-1071-1	451.45	413.73	**451.84**	560.94	19.45	549.19	17.73
1-1071-2	454.86	374.32	**420.2**	532.79	21.13	533.70	21.27
1-1071-3	495.2	410.98	**467.65**	566.68	17.48	575.76	18.78
1-1071-4	489.67	428.21	**461.4**	561.13	17.77	537.79	14.20
1-1071-5	399.07	370.35	**397.96**	505.67	21.30	481.33	17.32
1-1052-1	325.57	268.63	**322.56**	329.4	2.08	338.64	4.75
1-1052-2	376.66	296.12	**335.05**	362.92	7.68	376.30	10.96
1-1052-3	326.41	268.99	**310.27**	341.68	9.19	357.59	13.23
1-1052-4	367.04	295.17	**346.05**	374.36	7.56	379.46	8.80
1-1052-5	342.21	264.07	**308.73**	330.91	6.70	334.37	7.67
1-1072-1	637.37	401.1	**463.28**	512.29	9.57	512.34	9.58
1-1072-2	690.6	466.94	**529.22**	581.6	9.01	612.23	13.56
1-1072-3	508.91	367.88	**431.14**	445.58	3.24	468.70	8.01
1-1072-4	551.38	413.52	**491.02**	530	7.35	559.16	12.19
1-1072-5	531.64	392.88	**454.84**	501.06	9.22	509.94	10.81
1-1551-1	458.73	337.92	**402.54**	462.38	12.94	459.13	12.33
1-1551-2	414.62	294.14	**349.76**	391.64	10.69	388.99	10.09
1-1551-3	430.06	319.32	**384.4**	437.03	12.04	437.06	12.05
1-1551-4	420.33	314.08	**367.88**	427.74	13.99	427.74	13.99
1-1551-5	425.91	315.85	**369.22**	434.91	15.10	436.22	15.36
1-1571-1	733.36	452.81	**523.57**	631.65	17.11	634.96	17.54
1-1571-2	654.56	454.07	**526.99**	613.73	14.13	616.43	14.51
1-1571-3	553.61	405.71	**483.02**	524.82	7.96	525.00	8.00
1-1571-4	649.16	469.47	**541.69**	635.60	14.78	643.30	15.80
1-1571-5	666.62	440.91	**512.48**	610.60	16.07	610.44	16.05
1-1552-1	667.69	357.51	**428**	434.72	1.55	450.67	5.03
1-1552-2	797.32	369.08	**443.07**	460.58	3.80	477.25	7.16
1-1552-3	435.36	374.53	**421.83**	441.78	4.52	454.04	7.09
1-1552-4	492.81	358.67	422.84	**405.08**	-4.38	427.21	1.02
1-1552-5	474.11	343.76	401.7	**399.34**	-0.59	400.00	-0.42
1-1572-1	751.64	488.79	**555.95**	564.21	1.46	572.50	2.89
1-1572-2	1038.4	497.06	601.67	**600.91**	-0.13	646.12	6.88
1-1572-3	933.26	520.94	**619.4**	671.48	7.76	691.83	10.47
1-1572-4	869.07	472.66	**580.5**	615.73	5.72	642.36	9.63
1-1572-5	969.49	469.56	**601.3**	610.06	1.44	648.29	7.25
Average					8.54		10.64

ABCGX obtains better results in 13 instances compared to ETCH (given in bold) for Scenario 2 (see Table 4.3) and gives better performances in 55 instances when compared to *ABCRX*. The percentage different, on average shows that the *ABCGX* performs better or comparable to ETCH whilst performs poorly at 11.40% when compared to ETCH.

For Scenario 3, ETCH gives the best overall results in all 15 instances. When comparing the two algorithms *ABCRX* and *ABCGX,* it can be observed that the performance *ABCGX* is better than *ABCRX* with 13 best out of 15 datasets. The overall percentage difference shows that the *ABCRX, ABCGX* are worse than ETCH in an average 14.00 % and 10.24% respectively. This is shown in Table 4.4. It is noted that ETCH is a combination of 4 variants of ETCH.

Table 4.3: Results for Scenario 2 comparing UB, LB, ETCH, *ABCRX* and *ABCGX.*

Dataset	UB	LB	ETCH	*ABCRX*	%Δ	*ABCGX*	%Δ
2-0551-1	649.8	649.8*	700.28	699.30	-0.14	**694.76**	-0.79
2-0551-2	468	468*	**499.86**	554.26	9.81	502.44	0.51
2-0551-3	400	400*	**435.44**	439.79	0.99	441.01	1.26
2-0551-4	475.29	475.29*	**475.95**	508.94	6.48	507.52	6.22
2-0551-5	426.01	426.01*	**450.01**	506.99	11.24	472.66	4.79
2-0571-1	522.97	522.97*	**635.55**	730.15	12.96	646.03	1.62
2-0571-2	557.89	557.89*	619.82	647.54	4.28	**597.56**	-3.73
2-0571-3	434.86	434.86*	498.38	547.68	9.00	**473.79**	-5.19
2-0571-4	536.42	536.42*	632.25	711.88	11.19	**623.78**	-1.36
2-0571-5	498.08	498.08*	582.22	650.09	10.44	**549.81**	-5.89
2-0552-1	522.82	509	564.53	587.60	3.93	**557.73**	-1.22
2-0552-2	940.47	933.76	**1030.14**	1062.77	3.07	1031.47	0.13
2-0552-3	512.44	497.98	610.05	599.35	-1.79	**573.48**	-6.38
2-0552-4	537.37	519.91	643.42	590.71	-8.92	**587.88**	-9.45
2-0552-5	553.2	536.52	**619.70**	706.29	12.26	626.42	1.07
2-0572-1	828.6	789.04	**908.20**	1089.93	16.67	1036.06	12.34
2-0572-2	988.31	943.43	**1026.72**	1084.01	5.29	1035.82	0.88
2-0572-3	864.23	793.38	897.01	953.42	5.92	**891.93**	-0.57
2-0572-4	786.53	738.55	936.96	983.45	4.73	**918.79**	-1.98
2-0572-5	771.35	728.76	**916.89**	932.62	1.69	955.24	4.01
2-1051-1	528.69	509.59	**567.51**	628.85	9.75	585.95	3.15
2-1051-2	487.7	423.78	**495.26**	602.44	17.79	565.76	12.46
2-1051-3	724.13	660.23	750.05	769.05	2.47	**748.78**	-0.17
2-1051-4	456	445.86	**463.22**	539.43	14.13	497.26	6.85
2-1051-5	591.03	546.62	**611.11**	694.10	11.96	656.38	6.90
2-1071-1	784.36	728.48	**825.64**	988.31	16.46	873.68	5.50
2-1071-2	842.4	730.1	**819.35**	972.18	15.72	964.74	15.07
2-1071-3	748.65	668.8	**743.39**	966.32	23.07	894.58	16.90
2-1071-4	897.24	799.72	**952.75**	1038.56	8.26	1020.40	6.63
2-1071-5	763.69	712.32	856.24	942.60	9.16	**840.98**	-1.81

Table 4.3, Continued.

Dataset	UB	LB	ETCH	*ABCRX*	%Δ	*ABCGX*	%Δ
2-1052-1	829.24	758.39	**854.27**	953.85	10.44	956.31	10.67
2-1052-2	676.22	566.94	**674.14**	755.42	10.76	773.93	12.89
2-1052-3	759.4	659.22	**795.74**	878.90	9.46	859.65	7.43
2-1052-4	630.89	509.46	681.28	697.17	2.28	**658.33**	-3.49
2-1052-5	799.24	718.07	**827.11**	955.67	13.45	927.94	10.87
2-1072-1	955.63	808.46	**966.18**	1153.34	16.23	1112.76	13.17
2-1072-2	1266.9	1029.21	**1270.66**	1411.94	10.01	1372.05	7.39
2-1072-3	1037.6	857.06	**1047.41**	1255.39	16.57	1223.08	14.36
2-1072-4	1135.9	896.78	**1102.54**	1274.53	13.49	1202.11	8.28
2-1072-5	938.11	750.97	**918.74**	1174.86	21.80	1058.63	13.21
2-1551-1	823.1	736.42	**801.96**	867.23	7.53	835.26	3.99
2-1551-2	781.1	725.42	**775.02**	911.86	15.01	825.84	6.15
2-1551-3	800.63	666.94	**743.83**	882.11	15.68	815.80	8.82
2-1551-4	739.67	608.49	**711.51**	811.43	12.31	785.94	9.47
2-1551-5	1012.9	971.66	**1039.61**	1143.07	9.05	1062.19	2.13
2-1571-1	1095.1	747.3	**870.14**	1170.72	25.67	1096.47	20.64
2-1571-2	1097.7	660.8	**867.32**	1069.80	18.93	1006.15	13.80
2-1571-3	1217.2	800.45	**1007.49**	1288.20	21.79	1153.11	12.63
2-1571-4	1095.3	803.99	**1008.01**	1204.21	16.29	1150.30	12.37
2-1571-5	1383.8	1130.8	**1278.61**	1446.89	11.63	1404.19	8.94
2-1552-1	924.1	620.96	**802.75**	914.48	12.22	885.34	9.33
2-1552-2	818.36	595.9	**710.57**	812.32	12.53	800.04	11.18
2-1552-3	1103.7		**867.99**	1028.56	15.61	1000.75	13.27
2-1552-4	1086.4	923.82	**1049.75**	1223.68	14.21	1184.96	11.41
2-1552-5	1125.5	729.65	**909.47**	997.02	8.78	970.99	6.34
2-1572-1	1375.1	881.63	**1126.83**	1232.93	8.61	1244.46	9.45
2-1572-2	1415.2	972.09	**1175.02**	1550.30	24.21	1352.78	13.14
2-1572-3	1768.9	1042.43	**1261.41**	1605.13	21.41	1508.12	16.36
2-1572-4	1328.7	920.04	**1115.14**	1531.41	27.18	1450.60	23.13
2-1572-5	1575.2	1117.42	**1287.29**	1475.90	12.78	1468.17	12.32
Average					11.40		6.46

Table 4.4: Results for Scenario 3 comparing UB, LB, ETCH, *ABCRX* and *ABCGX.*

Dataset	UB	LB	ETCH	*ABCRX*	%Δ	*ABCGX*	%Δ
3-2072-1	892.42	510.34	**671.86**	756.08	11.14	754.88	11.00
3-2072-2	811.23	467.85	**624.83**	737.43	15.27	712.00	12.24
3-2072-3	802.6	495.58	**648.62**	764.95	15.21	749.76	13.49
3-2072-4	890.79	473.97	**612.49**	761.47	19.56	732.92	16.43
3-2072-5	1175.38	647.95	**771.46**	1028.87	25.02	841.03	8.27
3-2572-1	1265.5	570.43	**719.28**	833.45	13.70	805.79	10.74
3-2572-2	1295.92	613.47	**795.56**	921.01	13.62	886.77	10.29
3-2572-3	1347.05	608.41	**797.21**	900.38	11.46	886.71	10.09
3-2572-4	1411.5	566.68	**771.87**	857.14	9.95	848.77	9.06
3-2572-5	1280.35	560.61	**747.33**	870.59	14.16	848.80	11.95
3-3072-1	1823	570.69	**807.77**	882.40	8.46	884.98	8.72
3-3072-2	1739.72	596.14	**788.95**	908.17	13.13	872.79	9.61
3-3072-3	1981.65	653.8	**893.61**	967.63	7.65	974.16	8.27
3-3072-4	1794.65	653.37	**857.48**	972.74	11.85	908.91	5.66
3-3072-5	2138.36	678.46	**885.75**	1105.25	19.86	959.74	7.71
Average					14.00		10.24

Table 4.5, 4.6 and 4.7 display the details results (inventory holding cost (HOLD), backorder cost (BACK) and transportation cost (TRANSP)) for ETCH and the best of *ABCIRPB* (*ABCRX* and *ABCGX*) for each scenario. *ABCIRPB* have lower inventory decisions in Scenario 1 and lower backorder decisions in Scenario 2 in overall. This indicates that the *random exchange* and *guided exchange* developed give a significant contribution in finding the optimal solution at the expense of slightly higher transportation costs.

Table 4.8, 4.9 and 4.10 tabulate the detail components of the total cost for Scenario 1, 2 and 3 respectively for both *ABCRX* and *ABCGX*. The average and the standard deviation (ST DEV) for 10 runs of each dataset were also given in the tables.

It is shown that for all of the datasets in Scenario 1 (Table 4.8), the backorder cost is equal to zero, as this scenario does not beneficial to do backorder. The standard deviation for *ABCRX* and *ABCGX* is given by 9.18 and 6.12 on average.

It is shown in Table 4.9 and 4.10 that backorder costs exist in Scenario 2 and 3 as it is more economical to do backorder in both of these scenarios. It is calculated that the transportation cost contributes at least 51.96% to the total cost compared to the inventory and backorder costs in most of the datasets in Scenario 2 and 79% for Scenario 3. The standard deviation of total cost for *ABCRX* and *ABCGX* in scenario 2 and 3 are 25.53 and 15.39; and 17.55 and 10.67. It is noted that *ABCGX* gives lower backorder cost in 62 datasets when compared to *ABCRX*.

Table 4.5: Details component of ETCH and *ABCIRPB* for Scenario 1.

	ETCH				Best of *ABCIRPB*			
Dataset	**TOTAL**	**HOLD**	**BACK**	**TRANSP**	**TOTAL**	**HOLD**	**BACK**	**TRANSP**
1-0551-1	**205.84**	71.84	0	134	216.05	70.05	0	146
1-0551-2	**150.74**	45.74	0	105	159.41	36.41	0	123
1-0551-3	**186.6**	48.6	0	138	200.6	42.6	0	158
1-0551-4	**204.3**	58.3	0	146	218.04	67.04	0	151
1-0551-5	**185.35**	49.35	0	136	196.66	74.66	0	122
1-0571-1	**281.81**	70.81	0	211	313.72	104.72	0	209
1-0571-2	**272.98**	70.98	0	202	317.22	57.22	0	260
1-0571-3	**273.07**	75.07	0	198	309.92	116.92	0	193
1-0571-4	**349.49**	80.49	0	269	364.68	124.68	0	240
1-0571-5	**314.04**	86.04	0	228	365.3	129.3	0	236
1-0552-1	**221.69**	43.69	0	178	227.27	35.27	0	192
1-0552-2	**254.28**	69.28	0	185	272.14	58.14	0	214
1-0552-3	**223.98**	59.98	0	164	229.9	56.9	0	173
1-0552-4	254.83	72.83	0	182	**253.05**	69.05	0	184
1-0552-5	245.92	59.92	0	186	**244.66**	58.66	0	186
1-0572-1	336.38	87.38	0	249	**336.19**	93.19	0	243
1-0572-2	**290.33**	85.33	0	205	301.95	69.95	0	232
1-0572-3	**271.71**	59.71	0	212	292.46	55.46	0	237
1-0572-4	**286.79**	72.79	0	214	286.86	74.86	0	212
1-0572-5	307.91	68.91	0	239	**297.96**	58.96	0	239
1-1051-1	**326.97**	108.97	0	218	386.75	44.75	0	342
1-1051-2	**276.41**	93.41	0	183	317.32	65.32	0	252
1-1051-3	**300.69**	90.69	0	210	346.29	64.29	0	282
1-1051-4	**280.13**	88.13	0	192	326.11	52.11	0	274
1-1051-5	**249.63**	69.63	0	180	281.71	45.71	0	236
1-1071-1	**451.84**	142.84	0	309	549.19	89.19	0	460
1-1071-2	**420.2**	132.2	0	288	532.79	61.79	0	471
1-1071-3	**467.65**	130.65	0	337	566.68	69.68	0	497
1-1071-4	**461.4**	142.4	0	319	537.79	62.79	0	475
1-1071-5	**397.96**	130.96	0	267	481.33	65.33	0	416
1-1052-1	**322.56**	67.56	0	255	329.4	35.4	0	294
1-1052-2	**335.05**	94.05	0	241	362.92	31.92	0	331
1-1052-3	**310.27**	105.27	0	205	341.68	16.68	0	325
1-1052-4	**346.05**	84.05	0	262	374.36	77.36	0	297
1-1052-5	**308.73**	125.73	0	183	330.91	37.91	0	293
1-1072-1	**463.28**	135.28	0	328	512.29	45.29	0	467
1-1072-2	**529.22**	178.22	0	351	581.6	48.6	0	533
1-1072-3	**431.14**	137.14	0	294	445.58	27.58	0	418
1-1072-4	**491.02**	125.02	0	366	530	24	0	506
1-1072-5	**454.84**	137.84	0	317	501.06	25.06	0	476
1-1551-1	**402.54**	153.54	0	249	459.13	10.13	0	449
1-1551-2	**349.76**	131.76	0	218	388.99	6.99	0	382
1-1551-3	**384.4**	130.4	0	254	437.03	42.03	0	395
1-1551-4	**367.88**	109.88	0	258	427.74	3.74	0	424
1-1551-5	**369.22**	121.22	0	248	434.91	52.91	0	382
1-1571-1	**523.57**	180.57	0	343	631.65	58.65	0	573
1-1571-2	**526.99**	194.99	0	332	613.73	11.73	0	602
1-1571-3	**483.02**	179.02	0	304	524.82	1.82	0	523
1-1571-4	**541.69**	196.69	0	345	635.6	47.6	0	588
1-1571-5	**512.48**	176.48	0	336	610.44	6.44	0	604
1-1552-1	**428**	112	0	316	434.72	21.72	0	413
1-1552-2	**443.07**	138.07	0	305	460.58	14.58	0	446
1-1552-3	**421.83**	48.83	0	373	441.78	34.78	0	407
1-1552-4	422.84	53.84	0	369	**405.08**	17.08	0	388
1-1552-5	401.7	66.7	0	335	**399.34**	6.34	0	393
1-1572-1	**555.95**	148.95	0	407	564.21	27.21	0	537
1-1572-2	601.67	124.67	0	477	**600.91**	46.91	0	554
1-1572-3	**619.4**	193.4	0	426	671.48	32.48	0	639
1-1572-4	**580.5**	61.5	0	519	615.73	31.73	0	584
1-1572-5	**601.3**	155.3	0	446	610.06	35.06	0	575

Table 4.6: Details component of ETCH and *ABCIRPB* for Scenario 2.

	ETCH				Best of *ABCIRPB*			
Dataset	TOTAL	HOLD	BACK	TRANSP	TOTAL	HOLD	BACK	TRANSP
2-0551-1	700.28	12.66	300.62	387	**694.76**	5.3	135.46	554
2-0551-2	**499.86**	10.86	0	489	502.44	12.44	0	490
2-0551-3	**435.44**	4.74	42.7	388	439.79	17.51	56.28	366
2-0551-4	**475.95**	5.11	19.84	451	507.52	5.68	19.84	482
2-0551-5	**450.01**	8.35	33.66	408	472.66	8.66	0	464
2-0571-1	**635.55**	20.55	0	615	646.03	54.03	0	592
2-0571-2	619.82	8.63	169.19	442	**597.56**	20.77	136.79	440
2-0571-3	498.38	17.32	37.06	444	**473.79**	33.93	41.86	398
2-0571-4	632.25	24	41.25	567	**623.78**	51.78	0	572
2-0571-5	582.22	12.62	33.6	536	**549.81**	40.74	11.07	498
2-0552-1	564.53	11.53	0	553	**557.73**	14.12	17.61	526
2-0552-2	**1030.14**	0.2	533.94	496	1031.47	0.2	495.27	536
2-0552-3	610.05	21.53	14.52	574	**573.48**	22.64	6.84	544
2-0552-4	643.42	7.02	135.4	501	**587.88**	5.96	55.92	526
2-0552-5	**619.7**	5.7	0	614	626.42	8.88	9.54	608
2-0572-1	**908.2**	6.2	0	902	1036.06	4.73	107.33	924
2-0572-2	**1026.72**	7.88	69.84	949	1035.82	26.57	107.25	902
2-0572-3	897.01	6.51	7.5	883	**891.93**	6.79	47.14	838
2-0572-4	936.96	11.96	119	806	**918.79**	14.26	94.53	810
2-0572-5	**916.89**	11.09	61.8	844	932.62	28.52	50.1	854
2-1051-1	**567.51**	26.51	0	541	585.95	29.95	0	556
2-1051-2	**495.26**	47.26	0	448	565.76	49.76	0	516
2-1051-3	750.05	3.55	196.5	550	**748.78**	3.6	159.18	586
2-1051-4	**463.22**	21.22	0	442	497.26	29.26	0	468
2-1051-5	**611.11**	27.72	39.39	544	656.38	32.91	41.47	582
2-1071-1	**825.64**	32.39	64.25	729	873.68	41.88	37.8	794
2-1071-2	**819.35**	37.67	37.68	744	964.74	33.57	39.17	892
2-1071-3	**743.39**	52.39	0	691	894.58	82.58	0	812
2-1071-4	**952.75**	49.52	45.23	858	1020.4	70.12	28.28	922
2-1071-5	856.24	29.57	72.67	754	**840.98**	30.1	32.88	778
2-1052-1	**854.27**	6.62	117.65	730	953.85	0	107.85	846
2-1052-2	**674.14**	33.14	0	641	755.42	29.97	17.45	708
2-1052-3	**795.74**	21.14	47.6	727	859.65	17.07	32.58	810
2-1052-4	681.28	23.28	0	658	**658.33**	10.33	0	648
2-1052-5	**827.11**	4.56	81.55	741	927.94	2.1	75.84	850
2-1072-1	**966.18**	35.94	37.24	893	1112.76	53.52	77.24	982
2-1072-2	**1270.66**	31.74	131.92	1107	1372.05	28.39	101.66	1242
2-1072-3	**1047.41**	51.35	48.06	948	1223.08	23.08	0	1200
2-1072-4	**1102.54**	42.62	97.92	962	1202.11	25.23	72.88	1104
2-1072-5	**918.74**	30.66	24.08	864	1058.63	19.31	49.32	990
2-1551-1	**801.96**	63.86	22.1	716	835.26	21.96	47.3	766
2-1551-2	**775.02**	12.02	0	763	825.84	15.84	0	810
2-1551-3	**743.83**	20.74	95.09	628	815.8	18.86	82.94	714
2-1551-4	**711.51**	39.51	0	672	785.94	43.94	0	742
2-1551-5	**1039.61**	6.52	223.09	810	1062.19	5.93	166.26	890
2-1571-1	**870.14**	114.14	0	756	1096.47	126.47	0	970
2-1571-2	**867.32**	91.32	0	776	1006.15	142.15	0	864
2-1571-3	**1007.49**	99.49	0	908	1153.11	131.11	0	1022
2-1571-4	**1008.01**	80.01	0	928	1150.3	64.3	0	1086
2-1571-5	**1278.61**	27.92	260.69	990	1404.19	32.53	233.66	1138
2-1552-1	**802.75**	41.15	14.6	747	885.34	57.34	0	828
2-1552-2	**710.57**	41.57	0	669	800.04	40.04	0	760
2-1552-3	**867.99**	50.99	0	817	1000.75	32.75	0	968
2-1552-4	**1049.75**	14.05	104.7	931	1184.96	15.03	129.93	1040
2-1552-5	**909.47**	29.47	0	880	970.99	12.99	0	958
2-1572-1	**1126.83**	52.83	0	1074	1232.93	40.93	0	1192
2-1572-2	**1175.02**	12.02	28	1135	1352.78	13.18	53.6	1286
2-1572-3	**1261.41**	65.91	40.5	1155	1508.12	26.74	57.38	1424
2-1572-4	**1115.14**	106.14	0	1009	1450.6	104.6	0	1346
2-1572-5	**1287.29**	19.12	130.17	1138	1468.17	17.78	134.39	1316

Table 4.7: Details component of ETCH and *ABCIRPB* for Scenario 3.

	ETCH				Best of ABCIRPB			
Dataset	TOTAL	HOLD	BACK	TRANSP	TOTAL	HOLD	BACK	TRANSP
3-2072-1	**671.86**	61.52	5.34	605	754.88	30.37	2.51	722
3-2072-2	**624.83**	66.73	6.1	552	712	42	0	670
3-2072-3	**648.62**	55.62	0	593	749.76	123.76	0	626
3-2072-4	**612.49**	78.41	3.08	531	732.92	121.92	0	611
3-2072-5	**771.46**	13.51	2.95	755	841.03	16.29	5.74	819
3-2572-1	**719.28**	51.28	0	668	805.79	50.79	0	755
3-2572-2	**795.56**	77.42	17.14	701	886.77	40.98	7.79	838
3-2572-3	**797.21**	52.97	6.24	738	886.71	38.97	21.74	826
3-2572-4	**771.87**	69.19	7.68	695	848.77	45.77	0	803
3-2572-5	**747.33**	73.33	0	674	848.8	40.8	0	808
3-3072-1	**807.77**	48.52	16.25	743	882.4	0	14.4	868
3-3072-2	**788.95**	53.77	6.18	729	872.79	34.79	0	838
3-3072-3	**893.61**	42.61	0	851	967.63	3.34	4.29	960
3-3072-4	**857.48**	43.48	0	814	908.91	34.69	12.22	862
3-3072-5	**885.75**	29.75	0	856	959.74	28.74	0	931

Average time (in seconds) used is tabulated in Table 4.11, 4.12 and 4.13. From the scheme of ABC, it is known that the running time (CPU time) for each run may vary as, if the search converges faster then the CPU time will be shorter. As the *ABCGX* is embedded with the exhaustive guided exchange it is expected that the CPU time is larger for most of the datasets. The running time overall shows that larger dataset needs more time to find the optimal solution. Note that ETCH needs at most 90 seconds on average (Abdelmaguid et al., 2009). We conjecture that as ETCH make used of the CPLEX for solving the mathematical programming parts that contributed to the shorter computational times for all data sets.

Overall *ABCGX* performs better in term of total cost with 79 better results compared to 55 from *ABCRX* (and 1 equal results). However, when compared both *ABCRX* and *ABCGX* to ETCH, only 20 better results were obtained. It is anticipated that *ABCRX* and *ABCGX* perform better for *IRPB* as the problem is treated as a whole which tackled the vehicle routing, inventory and backorder decision concurrently. However, ETCH provided better results. It is conjectured that ETCH is a hybridization of the estimation transportation cost heuristic and mathematical programming (backorder and inventory decisions subproblems). The mathematical programming parts are solved using breadth-first search and dynamic programming.

Table 4.8: Scenario 1 details component of the total cost, average and standard deviation for *ABCRX* and *ABCGX*.

	ABCRX				AVERAGE	ST DEV	*ABCGX*				AVERAGE	ST DEV
Dataset	**TOTAL**	**HOLD**	**BACK**	**TRANSP**			**TOTAL**	**HOLD**	**BACK**	**TRANSP**		
1-0551-1	216.05	70.05	0	146	242.88	15.30	216.18	42.18	0	174	237.26	10.25
1-0551-2	159.41	36.41	0	123	175.30	10.58	163.11	40.11	0	123	175.05	5.84
1-0551-3	209.68	66.68	0	143	227.16	10.76	200.6	42.6	0	158	226.02	10.94
1-0551-4	218.04	67.04	0	151	242.76	11.75	225.96	74.96	0	151	239.21	10.14
1-0551-5	196.66	74.66	0	122	210.11	7.12	201.83	36.83	0	165	212.67	6.35
1-0571-1	313.72	104.72	0	209	347.13	20.72	330.29	134.29	0	196	346.12	10.88
1-0571-2	317.22	57.22	0	260	335.18	8.99	331.34	38.34	0	293	339.36	5.76
1-0571-3	309.92	116.92	0	193	340.25	16.97	329.52	120.52	0	209	347.07	10.26
1-0571-4	372.67	74.67	0	298	407.57	18.23	364.68	124.68	0	240	395.92	17.91
1-0571-5	365.3	129.3	0	236	394.88	20.25	375	147	0	228	396.57	11.95
1-0552-1	227.27	35.27	0	192	242.73	8.60	236.25	8.25	0	228	245.95	3.94
1-0552-2	272.14	58.14	0	214	289.62	14.20	285.45	79.45	0	206	302.28	10.84
1-0552-3	229.9	56.9	0	173	261.17	12.76	251.66	74.66	0	177	263.16	7.16
1-0552-4	269.27	52.27	0	217	291.29	12.72	***253.05***	69.05	0	184	297.38	17.99
1-0552-5	**244.66**	58.66	0	186	282.00	17.97	255.97	71.97	0	184	278.77	11.74
1-0572-1	336.19	93.19	0	243	386.02	21.17	373.2	131.2	0	242	386.80	11.30
1-0572-2	301.95	69.95	0	232	339.16	15.51	318.38	39.38	0	279	339.58	9.96
1-0572-3	292.46	55.46	0	237	327.08	12.97	312.81	27.81	0	285	326.61	8.42
1-0572-4	286.86	74.86	0	212	324.10	13.53	313.42	48.42	0	265	327.71	7.51
1-0572-5	**297.96**	58.96	0	239	329.65	11.93	331.11	36.11	0	295	334.21	1.77
1-1051-1	386.75	44.75	0	342	404.89	8.39	391.57	21.57	0	370	401.94	9.21
1-1051-2	317.32	65.32	0	252	334.76	7.52	330.21	35.21	0	295	339.08	4.35
1-1051-3	347.8	7.8	0	340	349.64	0.68	346.29	64.29	0	282	348.89	1.23
1-1051-4	334.73	65.73	0	269	348.07	9.82	326.11	52.11	0	274	343.08	8.85
1-1051-5	297.58	7.58	0	290	299.76	0.77	281.71	45.71	0	236	298.17	5.78
1-1071-1	560.94	83.94	0	477	571.06	4.08	549.19	89.19	0	460	563.39	9.86
1-1071-2	532.79	61.79	0	471	547.91	9.38	533.7	79.7	0	454	552.60	10.35
1-1071-3	566.68	69.68	0	497	589.46	11.48	575.76	77.76	0	498	590.27	10.32
1-1071-4	561.13	8.13	0	553	565.83	2.00	537.79	62.79	0	475	560.98	8.54
1-1071-5	505.67	69.67	0	436	515.06	4.29	481.33	65.33	0	416	505.24	9.35

Table 4.8, Continued.

Dataset	*ABCRX*				AVERAGE	ST DEV	*ABCGX*				AVERAGE	ST DEV
	TOTAL	HOLD	BACK	TRANSP			TOTAL	HOLD	BACK	TRANSP		
1-1052-1	329.4	35.4	0	294	349.86	11.58	338.64	19.64	0	319	346.03	4.21
1-1052-2	362.92	31.92	0	331	387.54	10.29	376.3	22.3	0	354	390.87	6.03
1-1052-3	341.68	16.68	0	325	360.57	7.30	357.59	10.59	0	347	362.34	2.93
1-1052-4	374.36	77.36	0	297	392.61	8.14	379.46	25.46	0	354	390.59	6.33
1-1052-5	330.91	37.91	0	293	348.44	9.20	334.37	32.37	0	302	345.02	6.04
1-1072-1	512.29	45.29	0	467	533.81	8.83	512.34	44.34	0	468	527.07	6.47
1-1072-2	581.6	48.6	0	533	622.85	16.01	612.23	61.23	0	551	628.86	7.18
1-1072-3	445.58	27.58	0	418	470.94	9.28	468.7	40.7	0	428	475.57	4.94
1-1072-4	530	24	0	506	570.02	14.77	559.16	46.16	0	513	571.34	6.39
1-1072-5	501.06	25.06	0	476	514.02	5.39	509.94	6.94	0	503	513.76	2.65
1-1551-1	462.38	40.38	0	422	463.79	1.13	459.13	10.13	0	449	463.14	2.04
1-1551-2	391.64	6.64	0	385	393.46	1.28	388.99	6.99	0	382	392.73	1.75
1-1551-3	437.03	42.03	0	395	449.63	4.86	437.06	51.06	0	386	446.59	4.69
1-1551-4	427.74	3.74	0	424	429.61	0.72	427.74	3.74	0	424	429.25	1.05
1-1551-5	434.91	52.91	0	382	443.32	5.38	436.22	8.22	0	428	442.70	4.20
1-1571-1	631.65	58.65	0	573	637.81	4.41	634.96	2.96	0	632	641.08	3.81
1-1571-2	613.73	11.73	0	602	619.28	3.50	616.43	66.43	0	550	621.33	2.40
1-1571-3	524.82	1.82	0	523	524.98	0.06	525	0	0	525	525.00	0.00
1-1571-4	635.6	47.6	0	588	641.69	2.81	643.3	3.3	0	640	643.05	1.67
1-1571-5	610.6	3.6	0	607	613.74	2.44	610.44	6.44	0	604	615.43	1.75
1-1552-1	434.72	21.72	0	413	452.84	7.22	450.67	22.67	0	428	454.52	2.56
1-1552-2	460.58	14.58	0	446	477.35	5.99	477.25	2.25	0	475	479.25	1.22
1-1552-3	441.78	34.78	0	407	464.98	9.07	454.04	75.04	0	379	466.71	6.17
1-1552-4	**405.08**	17.08	0	388	426.46	8.20	427.21	9.21	0	418	429.59	0.93
1-1552-5	**399.34**	6.34	0	393	399.93	0.21	***400***	0	0	400	400.00	0.00
1-1572-1	564.21	27.21	0	537	572.58	3.02	572.5	2.5	0	570	573.41	0.76
1-1572-2	**600.91**	46.91	0	554	654.36	19.21	646.12	51.12	0	595	658.22	5.71
1-1572-3	671.48	32.48	0	639	697.15	9.02	691.83	46.83	0	645	698.41	2.83
1-1572-4	615.73	31.73	0	584	641.01	8.90	642.36	3.36	0	639	643.81	0.52
1-1572-5	610.06	35.06	0	575	643.87	12.14	648.29	55.29	0	593	650.52	1.03
Average						9.18						6.12

Table 4.9: Scenario 2 details component of the total cost, average and standard deviation for *ABCRX* and *ABCGX*.

	ABCRX				AVERAGE	ST DEV	*ABCGX*				AVERAGE	ST DEV
Dataset	TOTAL	HOLD	BACK	TRANSP			TOTAL	HOLD	BACK	TRANSP		
2-0551-1	699.3	6.66	198.64	494	754.95	29.50	***694.76***	5.3	135.46	554	713.58	12.14
2-0551-2	554.26	10.38	13.88	530	602.52	29.80	502.44	12.44	0	490	521.81	7.94
2-0551-3	439.79	17.51	56.28	366	492.26	20.55	441.01	16.95	44.06	380	448.94	3.34
2-0551-4	508.94	5.76	25.18	478	540.29	23.19	507.52	5.68	19.84	482	516.50	6.36
2-0551-5	506.99	4.55	34.44	468	565.93	41.87	472.66	8.66	0	464	507.96	13.08
2-0571-1	730.15	66.72	159.43	504	778.35	30.32	646.03	54.03	0	592	664.06	13.46
2-0571-2	647.54	7.99	203.55	436	670.90	8.81	***597.56***	20.77	136.79	440	613.06	11.96
2-0571-3	547.68	5.98	61.7	480	581.15	21.84	***473.79***	33.93	41.86	398	481.60	6.12
2-0571-4	711.88	23.88	0	688	750.96	26.87	***623.78***	51.78	0	572	645.80	10.73
2-0571-5	650.09	23.02	11.07	616	682.03	19.11	***549.81***	40.74	11.07	498	579.41	15.80
2-0552-1	587.6	6.49	25.11	556	601.25	9.32	***557.73***	14.12	17.61	526	576.57	11.22
2-0552-2	1062.77	0	494.77	568	1062.77	0.00	1031.47	0.2	495.27	536	1050.17	14.43
2-0552-3	599.35	13.75	11.6	574	651.17	26.63	***573.48***	22.64	6.84	544	601.30	14.86
2-0552-4	590.71	4.64	68.07	518	614.95	14.42	***587.88***	5.96	55.92	526	601.56	7.63
2-0552-5	706.29	4.74	101.55	600	750.97	25.46	626.42	8.88	9.54	608	655.55	14.61
2-0572-1	1089.93	23.06	206.87	860	1142.45	32.83	1036.06	4.73	107.33	924	1061.82	14.81
2-0572-2	1084.01	8.84	159.17	916	1143.66	26.54	1035.82	26.57	107.25	902	1078.67	22.59
2-0572-3	953.42	9.5	121.92	822	993.51	31.67	***891.93***	6.79	47.14	838	924.92	25.96
2-0572-4	983.45	16.72	140.73	826	1007.86	12.86	***918.79***	14.26	94.53	810	955.97	19.28
2-0572-5	932.62	28.52	50.1	854	1009.47	43.83	955.24	26.88	58.36	870	984.79	25.74
2-1051-1	628.85	20.85	0	608	678.39	23.19	585.95	29.95	0	556	603.61	10.18
2-1051-2	602.44	22.44	0	580	626.88	11.14	565.76	49.76	0	516	590.18	17.74
2-1051-3	769.05	0	169.05	600	769.05	0.00	***748.78***	3.6	159.18	586	762.32	6.34
2-1051-4	539.43	15.43	0	524	577.36	26.70	497.26	29.26	0	468	516.99	10.93
2-1051-5	694.1	21.1	39	634	739.31	23.95	656.38	32.91	41.47	582	672.72	8.41
2-1071-1	988.31	16.56	73.75	898	1002.32	5.36	873.68	41.88	37.8	794	889.19	13.89
2-1071-2	972.18	4.5	37.68	930	976.72	1.82	964.74	33.57	39.17	892	974.54	4.75
2-1071-3	966.32	26.32	0	940	1029.23	59.60	894.58	82.58	0	812	908.05	11.56
2-1071-4	1038.56	48.05	20.51	970	1141.06	53.75	1020.4	70.12	28.28	922	1054.75	17.12
2-1071-5	942.6	0	32.6	910	942.60	0.00	***840.98***	30.1	32.88	778	875.54	15.13

Table 4.9, Continued.

	ABCRX				AVERAGE	ST DEV	*ABCGX*				AVERAGE	ST DEV
Dataset	**TOTAL**	**HOLD**	**BACK**	**TRANSP**			**TOTAL**	**HOLD**	**BACK**	**TRANSP**		
2-1052-1	953.85	0	107.85	846	989.22	17.60	956.31	1.21	107.1	848	981.31	17.36
2-1052-2	755.42	29.97	17.45	708	813.69	30.31	773.93	35.93	0	738	796.74	13.71
2-1052-3	878.9	0	44.9	834	888.33	4.58	859.65	17.07	32.58	810	882.88	10.28
2-1052-4	697.17	15.07	8.1	674	762.46	51.15	***658.33***	10.33	0	648	691.32	17.24
2-1052-5	955.67	0	93.67	862	973.93	6.42	927.94	2.1	75.84	850	941.54	12.00
2-1072-1	1153.34	24.09	81.25	1048	1227.58	53.56	1112.76	53.52	77.24	982	1169.87	37.74
2-1072-2	1411.94	19.97	139.97	1252	1478.61	26.65	1372.05	28.39	101.66	1242	1415.49	29.28
2-1072-3	1255.39	3.36	24.03	1228	1261.46	3.36	1223.08	23.08	0	1200	1246.25	11.62
2-1072-4	1274.53	12.11	110.42	1152	1326.96	41.81	1202.11	25.23	72.88	1104	1256.75	26.43
2-1072-5	1174.86	24.24	162.62	988	1289.82	74.29	1058.63	19.31	49.32	990	1114.77	32.00
2-1551-1	867.23	4.08	45.15	818	883.56	8.81	835.26	21.96	47.3	766	860.72	11.08
2-1551-2	911.86	1.5	12.36	898	933.68	11.50	825.84	15.84	0	810	843.02	8.23
2-1551-3	882.11	3.33	82.78	796	909.48	14.97	815.8	18.86	82.94	714	838.30	12.67
2-1551-4	811.43	17.43	0	794	821.76	5.69	785.94	43.94	0	742	799.06	7.35
2-1551-5	1143.07	5.76	233.31	904	1228.37	57.33	1062.19	5.93	166.26	890	1090.07	16.13
2-1571-1	1170.72	93.9	26.82	1050	1288.68	94.02	1096.47	126.47	0	970	1126.38	14.81
2-1571-2	1069.8	93.8	0	976	1117.71	37.55	1006.15	142.15	0	864	1031.84	12.72
2-1571-3	1288.2	62.29	1.91	1224	1312.61	18.61	1153.11	131.11	0	1022	1182.56	17.69
2-1571-4	1204.21	53.75	2.46	1148	1272.44	47.90	1150.3	64.3	0	1086	1191.60	19.56
2-1571-5	1446.89	2.1	236.79	1208	1463.04	9.07	1404.19	32.53	233.66	1138	1416.27	10.89
2-1552-1	914.48	40.48	0	874	972.20	52.43	885.34	57.34	0	828	912.64	17.26
2-1552-2	812.32	27.04	19.28	766	832.75	16.99	800.04	40.04	0	760	811.56	10.19
2-1552-3	1028.56	30.56	0	998	1040.54	7.35	1000.75	32.75	0	968	1018.49	11.21
2-1552-4	1223.68	0	191.68	1032	1238.92	5.49	1184.96	15.03	129.93	1040	1206.97	20.00
2-1552-5	997.02	23.02	0	974	1074.91	34.07	970.99	12.99	0	958	995.12	14.59
2-1572-1	1232.93	40.93	0	1192	1289.19	28.98	1244.46	25.22	9.24	1210	1270.63	20.93
2-1572-2	1550.3	2.1	274.2	1274	1550.30	0.00	1352.78	13.18	53.6	1286	1387.73	31.24
2-1572-3	1605.13	54.01	121.12	1430	1677.80	49.96	1508.12	26.74	57.38	1424	1542.11	16.79
2-1572-4	1531.41	70.11	31.3	1430	1586.50	27.67	1450.6	104.6	0	1346	1500.76	32.38
2-1572-5	1475.9	0	227.9	1248	1512.75	12.95	1468.17	17.78	134.39	1316	1498.48	21.78
Average						25.53						15.39

Table 4.10: Scenario 3 details component of the total cost, average and standard deviation for *ABCRX* and *ABCGX*.

	ABCRX				AVERAGE	ST DEV	*ABCGX*				AVERAGE	ST DEV
Dataset	TOTAL	HOLD	BACK	TRANSP			TOTAL	HOLD	BACK	TRANSP		
3-2072-1	756.08	31.08	0	725	788.29	21.01	754.88	30.37	2.51	722	769.45	9.84
3-2072-2	737.43	44.43	0	693	754.79	13.99	712	42	0	670	732.51	10.46
3-2072-3	764.95	23.95	0	741	776.98	13.03	749.76	123.76	0	626	765.63	8.74
3-2072-4	761.47	35.47	0	726	775.80	12.20	732.92	121.92	0	611	745.49	10.17
3-2072-5	1028.87	0	207.87	821	1028.87	0.00	841.03	16.29	5.74	819	871.10	19.98
3-2572-1	833.45	40.45	0	793	862.91	14.27	805.79	50.79	0	755	816.45	7.55
3-2572-2	921.01	30.93	24.08	866	947.43	16.40	886.77	40.98	7.79	838	903.66	12.76
3-2572-3	900.38	6.33	19.05	875	907.67	5.02	886.71	38.97	21.74	826	907.80	8.63
3-2572-4	857.14	54.14	0	803	887.67	20.49	848.77	45.77	0	803	863.63	9.74
3-2572-5	870.59	39.59	0	831	887.00	6.34	848.8	40.8	0	808	869.12	11.81
3-3072-1	882.4	0	14.4	868	887.85	3.69	884.98	0	16.98	868	885.88	0.32
3-3072-2	908.17	54.17	0	854	1019.14	65.72	872.79	34.79	0	838	891.66	11.69
3-3072-3	967.63	3.34	4.29	960	986.40	7.16	974.16	27.01	17.15	930	991.89	11.92
3-3072-4	972.74	9.3	31.44	932	1016.21	50.02	908.91	34.69	12.22	862	935.66	13.86
3-3072-5	1105.25	24.99	107.26	973	1137.40	13.98	959.74	28.74	0	931	979.96	12.65
Average						17.55						10.67

Table 4.11: Average Time (in seconds) for Scenario 1.

Dataset	*ABCRX*	*ABCGX*
1-0551-1	33.57	47.78
1-0551-2	63.89	48.79
1-0551-3	65.89	48.27
1-0551-4	33.99	48.33
1-0551-5	34.30	48.20
1-0571-1	41.94	62.15
1-0571-2	41.21	60.78
1-0571-3	42.66	62.69
1-0571-4	42.33	60.28
1-0571-5	42.78	58.99
1-0552-1	37.76	48.18
1-0552-2	38.13	48.48
1-0552-3	37.43	46.75
1-0552-4	37.86	47.40
1-0552-5	37.36	46.17
1-0572-1	47.60	57.72
1-0572-2	47.65	57.71
1-0572-3	46.52	57.34
1-0572-4	47.71	57.87
1-0572-5	48.36	58.12
1-1051-1	77.80	131.54
1-1051-2	77.48	154.70
1-1051-3	76.97	158.41
1-1051-4	75.86	76.46
1-1051-5	78.55	76.40
1-1071-1	94.60	97.57
1-1071-2	97.31	98.83
1-1071-3	94.54	99.52
1-1071-4	92.73	97.93
1-1071-5	91.88	97.19
1-1052-1	82.97	75.98
1-1052-2	83.97	75.68
1-1052-3	80.49	75.15
1-1052-4	83.81	81.52
1-1052-5	83.91	80.27
1-1072-1	99.74	195.55
1-1072-2	104.40	191.10
1-1072-3	99.44	205.11
1-1072-4	100.79	208.51
1-1072-5	100.22	191.76
1-1551-1	151.44	132.37
1-1551-2	145.80	132.25
1-1551-3	150.02	132.14
1-1551-4	151.82	131.76
1-1551-5	149.16	130.32
1-1571-1	177.82	325.13
1-1571-2	174.80	325.34
1-1571-3	174.61	339.82
1-1571-4	179.16	323.48
1-1571-5	176.14	327.76
1-1552-1	155.21	137.31
1-1552-2	159.26	140.34
1-1552-3	155.13	144.47
1-1552-4	159.97	140.34
1-1552-5	160.61	142.81
1-1572-1	182.85	167.99
1-1572-2	184.43	170.70
1-1572-3	190.94	168.19
1-1572-4	187.59	169.79
1-1572-5	191.15	169.84

Table 4.12: Average Time (in seconds) for Scenario 2.

Dataset	*ABCRX*	*ABCGX*
2-0551-1	32.06	40.52
2-0551-2	31.71	40.44
2-0551-3	32.82	39.28
2-0551-4	32.14	40.96
2-0551-5	33.34	39.57
2-0571-1	42.44	49.50
2-0571-2	40.25	49.15
2-0571-3	40.96	47.87
2-0571-4	42.58	48.68
2-0571-5	41.57	49.60
2-0552-1	44.96	50.36
2-0552-2	42.95	50.18
2-0552-3	49.79	52.75
2-0552-4	47.30	52.45
2-0552-5	46.00	56.39
2-0572-1	58.89	66.14
2-0572-2	59.64	68.04
2-0572-3	60.28	68.92
2-0572-4	61.95	69.87
2-0572-5	58.76	68.46
2-1051-1	64.78	130.78
2-1051-2	66.33	107.69
2-1051-3	57.03	148.47
2-1051-4	62.88	134.48
2-1051-5	63.32	134.08
2-1071-1	76.64	86.08
2-1071-2	77.35	85.06
2-1071-3	81.90	83.45
2-1071-4	79.93	84.98
2-1071-5	78.88	88.33
2-1052-1	67.36	135.03
2-1052-2	84.33	163.31
2-1052-3	73.75	185.30
2-1052-4	69.38	169.80
2-1052-5	65.32	97.69
2-1072-1	201.92	84.54
2-1072-2	181.73	84.68
2-1072-3	93.40	113.93
2-1072-4	194.55	121.03
2-1072-5	189.73	112.78
2-1551-1	101.75	113.74
2-1551-2	103.66	118.31
2-1551-3	104.73	114.52
2-1551-4	111.59	105.38
2-1551-5	96.67	123.37
2-1571-1	141.99	144.59
2-1571-2	146.96	140.77
2-1571-3	152.41	140.34
2-1571-4	140.87	150.45
2-1571-5	127.29	158.23
2-1552-1	257.21	159.68
2-1552-2	322.51	181.86
2-1552-3	283.86	171.32
2-1552-4	222.38	148.11
2-1552-5	287.54	164.18
2-1572-1	344.21	198.99
2-1572-2	294.62	184.60
2-1572-3	320.75	192.75
2-1572-4	430.44	222.21
2-1572-5	274.05	178.12

Table 4.13: Average Time (in seconds) for Scenario 3.

Dataset	*ABCRX*	*ABCGX*
3-2072-1	277.55	318.02
3-2072-2	356.18	357.73
3-2072-3	336.79	350.44
3-2072-4	325.60	338.66
3-2072-5	181.37	265.36
3-2572-1	345.60	465.79
3-2572-2	412.50	520.54
3-2572-3	427.55	527.79
3-2572-4	382.04	469.22
3-2572-5	469.31	542.13
3-3072-1	613.90	804.68
3-3072-2	340.03	626.06
3-3072-3	511.29	689.00
3-3072-4	451.30	614.47
3-3072-5	382.74	583.51

4.6 Statistical Analysis

Statistical analysis is carried out to determine if the ETCH, *ABCRX* and *ABCGX* are significantly different. The same statistical tests as the previous chapter were chosen such as the Friedman, Iman-Davenport (ID) and Friedman Aligned Rank (FAR) tests. The reason for conducting three different tests is because ID test is less conservative compared to the Friedman test. The FAR test offers a ranking procedure based on the datasets as well as the problems which is to overcome the disadvantages of the ranking procedure in the Friedman test. The null hypothesis (H_0) and the alternative hypothesis (H_1) at significance level $\alpha = 0.05$ are given below.

H_0: There are no significant differences between the performances of the ETCH, *ABCRX* and *ABCGX* algorithms.

H_1: At least one of ETCH, *ABCRX* and *ABCGX* is different in performance.

The chi-square approximation of the Friedman test statistic (χ_F^2), the Iman and Davenport's statistic (F_{ID}) and Friedman Aligned Rank statistic (F_{AR}) are calculated and checked for significance level at $\alpha = 0.05$. The chi-square critical value, $\chi^2_{0.05,2}$ is to compare the χ_F^2 and F_{AR} while F_{ID} is compared to $F_{0.05,2,118}$. The test is carried out for all

3 scenarios. Details on how to perform the test were given in the previous chapter. The results are tabulated in Table 4.14.

Table 4.14: Results of the Friedman, Iman-Davenport and Friedman Aligned Rank tests for Scenario 1, 2 and 3.

	Criteria	Friedman	Iman-Davenport	Friedman Aligned Rank
Scenario 1	**Statistic**	80.2750	119.2253	77.5065
	Critical Value	5.9915	3.0731	5.9915
	Decision	Reject H_0	Reject H_0	Reject H_0
Scenario 2	**Statistic**	78.4000	111.1923	68.8640
	Critical Value	5.9915	3.0731	5.9915
	Decision	Reject H_0	Reject H_0	Reject H_0
Scenario 3	**Statistic**	26.5333	107.1538	24.0651
	Critical Value	5.9915	3.3404	5.9915
	Decision	Reject H_0	Reject H_0	Reject H_0

Based on the results obtained in Table 4.14, it can be concluded that the performances of the ETCH, *ABCRX* and *ABCGX* algorithms are significantly different.

Since H_0 is rejected, the statistical test is further extended by performing a post-hoc procedure, particularly the Bonferroni procedure to detect the proper comparison. This procedure is done based on the average ranking of the Friedman test. For all three scenarios, the control algorithm (ETCH) is compared to the *ABCRX* and *ABCGX* algorithms to see if there is significant difference in the behavior of the algorithm. The H_0 shows that there is no significant difference in the behavior of the compared algorithm. The results of the Bonferroni procedure are the adjusted p-values (APV) calculated from the p -value of z-score (Derrac et al., 2011). Table 4.15 shows the results.

Table 4.15: Results of the Post Hoc Bonferonni procedure for Scenario 1, 2 and 3.

	Methods Compared	Adjusted p-value	Decision
Scenario 1	**ETCH vs *ABCRX***	< 0.00002	Reject H_0
	ETCH vs *ABCGX*	< 0.00002	Reject H_0
Scenario 2	**ETCH vs *ABCRX***	< 0.00002	Reject H_0
	ETCH vs *ABCGX*	0.00203	Reject H_0
Scenario 3	**ETCH vs *ABCRX***	< 0.00001	Reject H_0
	ETCH vs *ABCGX*	0.00191	Reject H_0

All the adjusted p-values found are significant, so H_0 is rejected. This shows that ETCH, $ABCRX$ and $ABCGX$ are significantly different from each other. Thus the developed algorithm $ABCRX$ and $ABCGX$ are alternative algorithms for solving IRPB.

4.7 Summary

In this chapter, ABC algorithm named *ABCIRPB* was successfully developed to solve inventory routing problem with backordering (*IRPB*). Inventory and backorder decisions are handle in the inventory updating mechanism, where two different exchange heuristics; *random exchange* and *guided exchange* are proposed. The exchanges exploit solutions intensively by balancing the inventory, backorder and transportation decisions. The *ABCIRPB* embedded with *random exchange* and *guided exchange* are denoted as *ABCRX* and *ABCGX* respectively. The performance of *ABCRX* and *ABCGX* were tested on 135 datasets and compared with the Estimated Transportation Cost Heuristic (ETCH) from the original literature Abdelmaguid et al. (2009). Results show that *ABCRX* and *ABCGX* obtained 20 best results compared to ETCH. The solution approach of ABC for *IRPB* is different from ETCH where in *ABCRX* and *ABCGX* tackled IRPB as a whole, while ETCH decomposed the problem into backorder and inventory subproblem. A statistical analysis was also carried out which shows that ETCH, *ABCRX* and *ABCGX* are significantly different using Friedman test, and the further test, Bonferroni shows that both *ABCRX* and *ABCGX* are significantly different from ETCH with a significance level 0.05. It can be concluded that *ABCRX* and *ABCGX* are another potential approach in solving IRPB.

CHAPTER 5: DYNAMIC STOCHASTIC INVENTORY ROUTING PROBLEM

This chapter presents the third main contribution to the knowledge, where an Inventory Routing Problem (IRP) with stochastic demand is studied. The demand is known in a probabilistic sense and it is dynamic as it is revealed or updated at the end of each period. Hence, the problem is known as Dynamic Stochastic Inventory Routing Problem (DSIRP). This chapter starts with the description of DSIRP and followed by the Mixed Integer Linear Programming (MILP) formulation. The stochastic dynamic programming formulation of the problem is presented and the solution methodology, the hybrid rollout algorithm is presented next. The hybrid algorithm is enhanced by proposing an ABC algorithm to generate controls, as an addition to the MILP controls. The performance of the enhanced hybrid rollout algorithm is tested on benchmark datasets by Archetti et al. (2007), where the probability distribution of the demand uses a binomial distribution. The expected costs found are compared with bounds found by CPLEX, and an analysis of the controls is carried out. In addition, the enhanced hybrid rollout algorithm is tested on demand that follows a uniform distribution. The patterns in the delivery quantities and number of visits are observed and analyzed for both probability distributions.

5.1 Problem Description

The stochastic inventory routing problem (SIRP) is similar to the deterministic inventory routing problem except that the demand is known in a probabilistic sense and it is considered dynamic in which the demand is gradually revealed at the end of each period (Coelho et al., 2014b). Hence, the problem is known as dynamic and stochastic inventory routing problem (DSIRP). Solving the DSIRP is to find a *solution policy* that specifies which actions must be performed at the end of each period (Berbeglia et al., 2010; Coelho et al., 2014a).

The distribution network adopted consists of a supplier, 0 and a set of retailers, $M = \{1, \dots, n\}$. A single product is distributed from supplier to retailers over planning horizon $T = \{1, \dots, H\}$ using a third-party transportation service. The inventory level for each retailer i has an initial I_{i0} which is an integer and the maximum inventory level of U_i, with $I_{i0} \leq U_i$. Each retailer i faces demand r_{it} at time t, where the demand is defined based on a discrete probability distribution (i.e binomial distribution) with mean q_i.

An order-up-to level (OU) inventory policy is adopted, which indicate that if retailer i is visited, the quantity delivered must reach its maximum level, U_i that is $U_i - I_{it}$. The transportation services is procured and the company charges a fixed cost, f for a capacity C at each time t whenever there is a delivery.

Denote that z_{it} is a binary variable which equals to 1 if retailer i is visited at time t, and 0 otherwise. The retailer's inventory level I_{it} is defined as $I_{it} = \max\{0, (U_i - I_{it-1})\, z_{it-1} - r_{it-1}\}$ where $z_{i0} = r_{i0} = 0$ and I_{i0} is given. As no backlogging is allowed, penalty cost d_i is charged for negative inventory level (stock out) and inventory holding cost h_i is incurred for positive inventory level, with $d_i > h_i$.

Inventory level at supplier, I_{00} is given and deterministic quantity p_t is produced at time t. The inventory level at supplier is given by $I_{0t} = I_{0t-1} + p_{t-1} - \sum_{i \in M} (U_i - I_{it-1})\, z_{it-1}$. Each positive inventory level will be charged inventory holding cost h_0. Note that it is assumed that the supplier has sufficient product to fulfil the demand which indicate that supplier's initial inventory level cannot be negative.

Figure 5.1 is a pictorial example of the problem. It shows a supplier with 4 retailers and a single product to be distributed by buying a transportation capacity. The retailers face customers' demand which is known in a probabilistic sense.

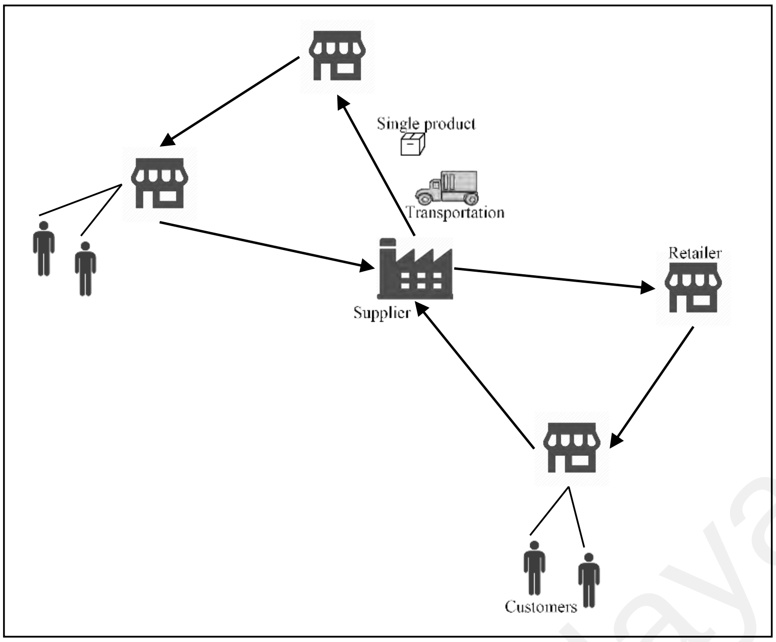


Figure 5.1: A pictorial example of the problem.

The objective of the problem is to find a policy such that the sum of the expected inventory cost at the supplier, inventory cost and stock out cost at the retailers and also transportation cost is minimized over the planning horizon.

5.2 Problem Formulation

The mixed integer linear programming (MILP) formulation is a simpler version of the DSIRP where the demand is deterministic. It is adopted from Bertazzi et al. (2015). The notations and decision variables are given below.

Notations

M	Retailers $\{1,2,\dots,n\}$
T	Periods, $\{1,2,\dots,H\}$
U_i	Maximum inventory level for retailer i
i	Index for retailer
t	Index for period
I_{it}	Inventory level of retailer i at time t
r_{it}	Demand of retailer i at time t

h_i	Inventory cost for retailer i
d_i	Penalty cost for retailer i
p_t	Quantity produce at time t
C	Transportation capacity
f	Fixed transportation cost

Decision Variable

z_{it}	Binary variable, 1 if retailer i is visited, otherwise 0
y_t	Binary variable, 1 if retailer i is served, otherwise 0
δ_{it}	Binary variable, 1 if $\beta_{it} > 0$ and 0 otherwise
γ_{it}	Binary variable, 1 if $\alpha_{it} > 0$ and 0 otherwise
s_{it}	Quantity sent to retailer i at time t
α_{it}	Inventory level of retailer i at time t
β_{it}	Stock out level of retailer i at time t
I_{0t}	Inventory level of supplier at time t

The MILP model formulation is given as below.

$$\min \sum_{t\in T'} h_0 I_{0t} + \sum_{i\in M}\sum_{t\in T'} h_i \alpha_{it} + \sum_{i\in M}\sum_{t\in T'} d_i \beta_{it} + \sum_{t\in T} f y_t \tag{5.1}$$

Subject to:

$$I_{00} = \bar{I}_{00} \tag{5.2}$$

$$I_{0t} = I_{0t-1} + p_{t-1} - \textstyle\sum_{i\in M} s_{it-1} \qquad t \in T' \tag{5.3}$$

$$\alpha_{i0} = \bar{I}_{i0} \qquad i \in M \tag{5.4}$$

$$\alpha_{it} - \beta_{it} = \alpha_{it-1} + s_{it-1} - r_{it-1} \qquad i\epsilon M, t \in T' \tag{5.5}$$

$$\alpha_{it} \leq U_i \gamma_{it} \qquad i\epsilon M, t \in T' \tag{5.6}$$

$$\beta_{it} \leq U_i \delta_{it} \qquad i\epsilon M, t \in T' \tag{5.7}$$

$$\gamma_{it} + \delta_{it} \leq 1 \qquad i\epsilon M, t \in T' \qquad (5.8)$$

$$s_{it} \geq U_i z_{it} - \alpha_{it} \qquad i\epsilon M, t \in T \qquad (5.9)$$

$$s_{it} \leq U_i - \alpha_{it} \qquad i\epsilon M, t \in T \qquad (5.10)$$

$$s_{it} \leq U_i z_{it} \qquad i\epsilon M, t \in T \qquad (5.11)$$

$$\sum_{i\epsilon M} s_{it} \leq C y_t \qquad t \epsilon T \qquad (5.12)$$

$$\sum_{i\epsilon M} s_{it} \leq I_{0t} \qquad t \epsilon T \qquad (5.13)$$

$$I_{0t} \geq 0 \qquad t \epsilon T' \qquad (5.14)$$

$$\alpha_{it} \geq 0 \qquad i\epsilon M, t \in T' \qquad (5.15)$$

$$\beta_{it} \geq 0 \qquad i\epsilon M, t \in T' \qquad (5.16)$$

$$s_{it} \geq 0, \text{integer} \qquad i \epsilon M, t \epsilon T \qquad (5.17)$$

$$\gamma_{it} \epsilon \{0,1\} \qquad i \epsilon M, t \epsilon T' \qquad (5.18)$$

$$\delta_{it} \epsilon \{0,1\} \qquad i \epsilon M, t \epsilon T' \qquad (5.19)$$

$$z_{it} \epsilon \{0,1\} \qquad i \epsilon M, t \epsilon T \qquad (5.20)$$

$$y_t \epsilon \{0,1\} \qquad t \epsilon T \qquad (5.21)$$

The objective function (5.1) comprises four components, the supplier's inventory cost, the retailer's inventory cost, the stock out penalty cost at retailers site and the transportation cost. Constraints (5.2) are the assignment of initial inventory. Constraints (5.3) are the supplier inventory balance equation that determines the inventory level at the supplier, given the initial production, $p_0 = 0$ and the quantity delivered at time $t = 0$, $s_{i0} = 0$ for retailer i, $i\epsilon M$. Constraints (5.4) and (5.5) assess the inventory level and stock out level for retailer i, $i\epsilon M$ at time t, given that $s_{i0} = r_{i0} = 0$. Constraints (5.6) and (5.7) ensure that the inventory level is positive and the stock out level is not more than U_i. Constraints (5.8) indicate that either inventory or stock out can exist at time t. The OU inventory policy adopted is defined by constraints (5.9) – (5.11), which indicate that if retailer i is served, the quantity s_{it} delivered must reach $\max\{U_i - \alpha_{it}, 0\}$. Constraints

(5.12) define the transportation capacity constraint whilst constraints (5.13) ensure that the total quantity sent to the retailers must not exceed the quantity available at the supplier site. (5.14) – (5.21) consist of the non-negative constraints and the decisions variables. This MILP formulation is used several times in the hybrid rollout algorithm in which the MILP is optimally solved to obtain one of the controls where the demand is set to be equal to the average demand, $r_{it} = q_i$. The MILP is also used to compute the approximate cost-to-go and in obtaining the bounds for comparison. The formulation is known as problem *Det* throughout this chapter. This deterministic model is proven to be NP-hard and eventually, the DSRIP considered is NP-hard (Bertazzi et al., 2015).

Solving the dynamic stochastic IRP (DSIRP) consist of proposing a solution policy (solution strategy or distribution policy) as opposed to the deterministic solution in the deterministic or static IRP (Coelho et al., 2014a). There are several policies that can be proposed and one such policy is to optimize the deterministic version of DSIRP once new information becomes available. However, this policy is very time-consuming due to the large number of instances that must be considered. The other policy, the most common one is to solve the deterministic version of DSIRP once and proceed using heuristics whenever new information becomes available. The third policy encompasses the combination of the first two policies and in this thesis, we have adopted the third policy.

5.3 Solution Methodology

The solution methodology consists of two parts. The first part is the mathematical framework; stochastic dynamic programming formulation and the second part is the hybrid rollout algorithm, based on the stochastic dynamic programming formulation. They are presented in subsections 5.3.1 and 5.3.2 respectively.

5.3.1 Stochastic Dynamic Programming

There are several approaches to solve Stochastic IRP from the literature such as heuristics algorithms, dynamic programming and robust optimization. See Coelho et al. (2014b) for details. In this section, the SIRP is formulated as stochastic dynamic programming (SDP). The SDP is extensively discussed in Bertsekas (1995) and Powell (2011).

The essential components of the SDP formulation are: state of the system, controls constraint, discrete-time dynamic system, the immediate cost and the optimization problem. Definitions of the various components are given in the following subsections.

5.3.1.1 States

The state variable is denoted as x_t, where x_t is an n-dimensional integer vector representing the inventory level of the supplier and each retailer i, $i \in M$ at time $t = 0, 1, \dots, H+1$. Hence, the state of the system is given by $x_t = (x_{0t}, x_{1t}, x_{2t}, \dots, x_{nt})$ and $x_0 = (x_{00}, x_{10}, x_{20}, \dots, x_{n0}) = (\bar{I}_{00}, \bar{I}_{10}, \dots, \bar{I}_{n0})$, given that the inventory level at both the supplier and retailer at time $t = 0$ is known.

The state of the supplier at time $t \in T$, x_{0t} is an integer that obeys the constraint $\bar{I}_{00} + \sum_{k=1}^{t-1} p_k - (t-1)C \leq x_{0t} \leq \bar{I}_{00} + \sum_{k=1}^{t-1} p_k$ and at time $t = H+1$ is $\bar{I}_{00} + \sum_{k=1}^{t-1} p_k - HC \leq x_{0H+1} \leq \bar{I}_{00} + \sum_{k=1}^{t-1} p_k$ whilst the state of the retailer i at time $t \in T, x_{it}$ is a positive integer that is constrained by $0 \leq x_{it} \leq U_i$ and at time $t = H+1$ is $-U_i \leq x_{iH+1} \leq U_i$.

5.3.1.2 Controls

Controls are a set of actions (decisions) available when the system is in state x_t. The control $u_t(x_t)$ at time $t \in T$ is defined as

$$u_t(x_t) = (z_{1t}, z_{2t}, \dots, z_{nt})$$

where z_{it}, $i \in M$, is a binary variable that determines whether a retailer is visited at time $t \epsilon T$, for each state x_t.

As an OU inventory policy is adopted, once z_{it} values are determined then the delivery quantities can be calculated as $(U_i - x_{it})z_{it}$. At time t and state x_t is the controls $u_t(x_t) \in \mathcal{U}_t(x_t)$ where $\mathcal{U}_t(x_t)$ the set of feasible controls and the feasibility of the controls is bounded by:

(1) the capacity constraints

$$\sum_{i \in M}(U_i - x_{it})z_{it} \le C \tag{5.22}$$

(2) the availability of the product at supplier site at time t given by

$$\sum_{i \in M}(U_i - x_{it})z_{it} \le x_{0t} \tag{5.23}$$

5.3.1.3 Dynamic System

The dynamic system describes the progress of the system from one state to another, specifically in this study, how to obtain the next inventory level (at state $t + 1$) from state t. The dynamic of the system is represented by a function that describes how the state progresses as new information arrives (demands) and the decisions (controls) are made.

The state of the system at time $t + 1$ is determined by the state of the system x_t at time t, the control u_t applied at time t, and demand r_t is a vector of demands r_{it} that arrives during time interval t. It is given by:

$$x_{t+1} = (\hat{x}_{0t}, max\{0, \hat{x}_{1t}\}, max\{0, \hat{x}_{2t}\}, \dots, max\{0, \hat{x}_{nt}\})$$

where for each retailer i, $i \in M$, $\hat{x}_{it} = x_{it} + (U_i - x_{it})z_{it} - r_{it}$ and supplier $\hat{x}_{0t} = x_{0t} + p_t - \sum_{i \in M}(U_i - x_{it})z_{it}$. The terminal state is given by $x_{H+1} = (\hat{x}_{0H}, \hat{x}_{0H}, \dots, \hat{x}_{0H})$.

5.3.1.4 Costs

As a result of applying control u_t in state x_t and having the realization of demand r_t, the immediate cost, g is given by the total of the inventory, stock out penalty and transportation costs.

$$g(x_t, u_t, r_t) = \sum_{i \in M} h_i \ max\{0, \hat{x}_{it}\} + \sum_{i \in M} d_i \ max\{0, -\hat{x}_{it}\} + \ \mathrm{f}\emptyset\left(\sum_{\mathrm{i \in M}} \mathrm{z_{it}}\right)$$

where h_i is the inventory cost, d_i is the penalty for stock out, f fixed transportation cost, and $\emptyset(w) = 1$ if $w > 0$ and 0 otherwise. w indicates that at least one of $\mathrm{z_{it}}$ is positive for $\emptyset(w) = 1$.

5.3.1.5 Optimization Problem

The optimization problem is to determine the best policy among all feasible policies that minimize the expected total cost. It is emphasized that policy is a *function* that returns a decision (controls in this case) (Powell, 2011).

A policy is defined as:

Definition 5.1 *A policy is a rule (or function) that determines a decision given the available information in state* x_t. (Powell, 2011).

Consider the set Π of feasible policies and each of these policies consists of a sequence of functions π where $\pi = \{\mu_1, \mu_{21}, \dots, \mu_H\}$. Each μ_t maps each state x_t into a control $u_t = \mu_t(x_t)$ such that $\mu_t(x_t) \ \in \mathcal{U}_t(x_t)$ for all states $x_t, t \in T$.

Starting from the given initial state x_0, the total expected cost of π is calculated as:

$$J_\pi(x_0) = E\left\{\sum_{t=1}^{H} g(x_t, \mu_t(x_t), r_t) + \ g_{H+1}(x_{H+1})\right\}$$

The best policy is found by choosing the best π from a family of Π, $\pi \in \Pi$ such that the total expected cost is minimized over the planning horizon. Hence, the objective is to seek an optimal policy π^*, such that :

$$J_{\pi^*}(x_0) = min_{\pi \in \Pi}\, J_\pi(x_0)$$

5.3.2 Hybrid Rollout Algorithm

An exact dynamic programming algorithm using the formulation presented in subsection 5.3.1 is introduced in Bertazzi et al. (2015) where exact solution has been determined for a dataset with up to 2 retailers and 3 periods. However, the difficulty occurs when solving realistic size dataset (larger number of retailers and periods) where the exact dynamic programming is unable to solve as the problem suffers the three curses of dimensionality. The three curses of dimensionality are the state space, the outcome space and the control space.

State Space: As the inventory level of retailer i is in the range of $(0, \dots, U_i)$, and this generated (cardinality) a state space of dimension $(\overline{U} + 1)^n$ for each of the retailer $i = 1, \dots, n$ at each time t where $\overline{U} = max_{i \in M} U_i$. Further more, this value has to be multiplied by the number of possible inventory level at supplier.

Outcome Space: The value of the demand for each retailer $i \in M$ at time $t \in T$ is defined as $U_i + 1$, so the cardinality of the set of possible outcomes is approximately $(\overline{U} + 1)^n$.

Control Space: The control vector $u_t(x_t) = (z_{1t}, z_{2t}, \dots, z_{nt})$ has n dimensions and since z_{it} can take on 2 possible outcomes for each retailer $i \in M$), then the cardinility of the set of possible controls is approximately 2^n.

In this chapter a hybrid rollout algorithm is proposed instead of using exact dynamic programming in order to solve realistic size datasets. The hybrid heuristics algorithm incorporates the Matheuristic approach where heuristic/metaheuristic is integrated with exact methods based on mathematical techniques (Caserta & Voß, 2009). Surveys and classification of Matheuristics for combinatorial optimization can be found in Puchinger and Raidl (2005) and Archetti and Speranza (2014).

Due to the advancement and efficiency of the optimization software, finding optimal (near-optimal) solution for mathematical programming particularly MILP models can be done easily. Thus, proposing a hybrid rollout algorithm that makes use of the MILP inside the scheme seems promising. In this algorithm the approximate cost-to-go, $\tilde{J}_{t+1}$ and the approximate set of controls are obtained by solving the problem *Det* presented in subsection 5.2.

5.3.2.1 Rollout Algorithm

Rollout algorithm is adopted when the number of actions (controls) per state is relatively large. Rollout algorithm is a one-step look-ahead policy. The look-ahead policies are "the policies that make decisions now, where in this policy they explicitly optimize over some horizon by combining an approximation of future actions", as given in Powell (2011).

Rollout algorithm is a method that calculates a rough estimate of the future for each of the actions (controls) taken in the current state. Specifically, the algorithm uses a *base policy* (normally either analytically or by simulation) to evaluate the future trajectory. The trajectory value is then used to estimate the cost of being in the current state. This process is repeated for each of the controls, and the best control in the current state is chosen based on the future trajectories evaluated.

"Rollout algorithms have been originally proposed in the context of Neuro-Dynamic Programming/Reinforcement Learning" (Bertazzi et al., 2015). The algorithms have been widely implemented in different fields such as scheduling (Bertsekas & Castanon, 1999;

Meloni et al., 2004), vehicle routing problem (Secomandi, 2001, 2003), and multidimensional knapsack problems (Bertsimas & Demir, 2002; Mastin & Jaillet, 2015). Rollout algorithm is simple and easy to implement and it always guarantees to produce better results as compared to the base policy alone (Bertsekas, 2013; Bertsekas & Castanon, 1999).

Note that the future trajectories are required in the calculation of approximate cost-to-go, $\tilde{J}_{t+1}$. The reader is referred to the details explanation and examples of rollout algorithm in Bertsekas et al. (1997) and Powell (2011).

5.3.2.2 Approximate Rollout Control

There are many stochastic problems arise in different fields that are solved using the dynamic programming framework, such as in the field of engineering, economy, operations research and artificial intelligence. Different communities used the same concepts and algorithms but with their own vocabularies and notations. For example, the control theory community refers the methodology as Neuro-Dynamic Programming (NDP), the artificial intelligence community used Reinforcement Learning (RL) and Approximate Dynamic Programming (ADP) in the operations research community Powell (2011). The difference in the notation style to define the state, control and policy is captured in the book "Approximate Dynamic Programming" by Powell (2011). In addition to the book, we refer readers to the Bertsekas and Tsitsiklis (1996) and Sutton and Barto (1998) for the theoretical work and applications.

The representations and formulations used in this thesis follow the common notations represented in NDP. The key idea in NDP formulation proposed is to select the decision that minimizes the expected cost in the current state (state t) in addition to the approximation of future costs (future trajectories) that starts from state t+1 to H.

Following the cost approximation formulation derived in Bertsekas and Tsitsiklis (1995), the approximate rollout control $\tilde{\mu}_t$ is known as *Policy M*. It is define as:

$$\tilde{\mu}_t(x_t) = \arg\ \min_{u_t(x_t)\in\tilde{\mathcal{U}}_t(x_t)} \tilde{Q}_t\big(x_t, u_t(x_t)\big) \tag{5.24}$$

for all $t = 1, \dots, H$, where $\tilde{Q}_t$ is the approximate Q factor that consists of two key components that are immediate cost and approximate future cost discussed previously. The $\tilde{Q}_t\big(x_t, u_t(x_t)\big) = E\{g(x_t, u_t, r_t) + \tilde{J}_{t+1}(x_{t+1})\}$ where $\tilde{J}_{t+1}(x_{t+1})$ is the approximate cost-to-go and $\tilde{\mathcal{U}}_t(x_t)$ is the approximate set of controls.

(a) ***Scenarios, $S_t(x_t, u_t)$***

The set of scenarios $S_t(x_t, u_t)$ must first be generated in order to compute the approximate Q factor, the approximate cost-to-go and the approximate set of controls. It is common in dynamic programming to use the term *scenario* to describe a set of random outcomes and in this chapter the scenario $s \in S_t(x_t, u_t)$ represents the realization of demand, r_t for each retailer at time t.

Scenarios are the randomly generated demand, r_{it} for each retailer $i \in M$ that follows a certain probability distribution, and in this thesis the binomial probability distribution is selected. Alternatively Normal or Poisson distribution can be used (Bertazzi et al., 2015). Additional three scenarios are considered. These scenarios are to capture the extreme and the most likely demand to happen for each retailer, where r_{it} is set to 0 (minimum value), U_i (maximum value) and q_i (average value).

(b) ***The Approximate Cost-to-go, $\tilde{J}_{t+1}(x_{t+1})$***

Once the scenarios are defined, the exact immediate costs $g(x_t, u_t, r_t)$ is calculated and the corresponding inventory level of the retailers $\bar{x}_{it+1}$ for $i \in M$ including the inventory level of the supplier $\bar{x}_{0t+1}$ at time $t+1$ are computed. These are the required input in solving problem *Det* in order to determine the approximate cost-to-go (period $t + 1$ to H).

The computational of the approximate cost-to-go, $\tilde{J}_{t+1}(x_{t+1})$ is as follows. The demand for time $t + 1$ is set as $r_{it+1} = 0$, U_i, q_i whilst the demands for periods $t + 2$ until H are to set to be equal to the average demand, $r_{it+2}, \ldots, r_{iH} = q_i$. Hence, the $\tilde{J}_{t+1}(x_{t+1})$ is calculated as the average of the optimal costs found by solving problem *Det* for the three different setting of demand at time $t + 1$. Note that the value of $\tilde{J}_{t+1}(x_{t+1})$ needs to be calculated for all scenarios.

(c) ***The Approximate Q factor,*** $\tilde{\boldsymbol{Q}}_t(\boldsymbol{x}_t, \boldsymbol{u}_t(\boldsymbol{x}_t))$

The values of immediate cost $g(x_t, u_t, r_t)$ and the approximate cost-to-go, $\tilde{J}_{t+1}(x_{t+1})$ have to be obtained first before the approximate Q factor, $\tilde{Q}_t(x_t, u_t(x_t))$ can be computed, $\tilde{Q}_t(x_t, u_t(x_t)) = E\{g(x_t, u_t, r_t) + \tilde{J}_{t+1}(x_{t+1})\}$. Note that each scenario can be generated several times the actual approximate Q factor, $\tilde{Q}_t(x_t, u_t(x_t))$ represents as the average value of the costs of the different scenarios where the weight of each scenario is the relative frequency of the scenario. In other words, if a scenario occurs for several times (let say K number of times), the $\tilde{Q}_t(x_t, u_t(x_t))$ is calculated as the average value with frequency K.

(d) ***The Approximate Set of Controls,*** $\tilde{\mathcal{U}}_t(\boldsymbol{x}_t)$

The approximate set of controls, $\tilde{\mathcal{U}}_t(x_t)$ associated to each state x_t at time t is composed of four types of controls, including controls generated from MILP formulations. The controls are obtained by considering:

i. No retailer is served.

ii. Solving exactly problem *Det* where the demand is set to be equal to the average demand (q_i) from time t to H.

iii. Solving exactly four different MILP models that capture the inventory level and stock out level at time $t+1$. The models comprise of a combination of 2 different objective functions and 2 different scenarios. The two objectives of the MILP considered are to maximize the number of retailers visited z_{it}, $(\max \sum_{i \in M} z_{it})$ and minimizing both the inventory level and the stock out level at time $t+1$ $(\min\{\sum_{i \in M} h_i \alpha_{it+1} + \sum_{i \in M} \beta_i d_{it+1}\})$ and the two different scenarios are $r_{it} = 0$ and $r_{it} = U_i$ for $i \in M$. Given that the inventory levels at the supplier $(\bar{I}_{0t})$ and the retailers $(\bar{I}_{it}, i \in M)$ are known, the MILP models are solved subject to the following constraints:

$$I_{0t+1} = \bar{I}_{0t} + p_t - \sum_{i \in M} s_{it} \tag{5.25}$$

$$\alpha_{it+1} - \beta_{it+1} = \bar{I}_{it} + s_{it} - r_{it} \qquad i \epsilon M \tag{5.26}$$

$$\alpha_{it+1} \leq U_i \gamma_{it+1} \qquad i \epsilon M \tag{5.27}$$

$$\beta_{it+1} \leq U_i \delta_{it+1} \qquad i \epsilon M \tag{5.28}$$

$$\gamma_{it+1} + \delta_{it+1} \leq 1 \qquad i \epsilon M \tag{5.29}$$

$$s_{it} \geq U_i z_{it} - \bar{I}_{it} \qquad i \epsilon M \tag{5.30}$$

$$s_{it} \leq U_i - \bar{I}_{it} \qquad i \epsilon M \tag{5.31}$$

$$s_{it} \leq U_i z_{it} \qquad i \epsilon M \tag{5.32}$$

$$z_{it} \leq s_{it} \qquad i \epsilon M \tag{5.33}$$

$$\sum_{i \epsilon M} s_{it} \leq C \tag{5.34}$$

$$\sum_{i \epsilon M} s_{it} \leq I_{0t} \tag{5.35}$$

$$\alpha_{it+1} \geq 0 \qquad i \epsilon M \tag{5.36}$$

$$\beta_{it+1} \geq 0 \qquad i \epsilon M \tag{5.37}$$

$$s_{it} \geq 0, \text{integer} \qquad i \epsilon M \tag{5.38}$$

$$\gamma_{it+1} \epsilon \{0,1\} \qquad i \epsilon M \tag{5.39}$$

$$\delta_{it+1} \epsilon \{0,1\} \qquad i \epsilon M \tag{5.40}$$

$$z_{it} \epsilon \{0,1\} \qquad i \epsilon M \tag{5.41}$$

The model is adopted from Bertazzi et al. (2015) where (5.26) is the inventory balance equation, (5.30)-(5.32) are the OU policy, (5.33) is to ensure that $z_{i0} = 0$ whenever $s_{i0} = 0$, and the rest are the non-negative and binary constraints.

iv. Metaheuristic Algorithm which is described in Section 5.4.

5.4 Enhanced Hybrid Rollout Algorithm

This section presents the enhancement of the hybrid rollout algorithm ($Policy\ M$). *Policy M* is described as equation (5.24) with the details of the approximate cost-to-go, approximate Q factor and approximate set of controls given in the whole subsection 5.3.2.2. The enhancement on *Policy M* is done by proposing an Artificial Bee Colony (ABC) algorithm to generate an additional set of controls. The new policy is denoted as $Policy\ M^{+}$.

$Policy\ M$ is also embedded with an improvement heuristic, *Heur_Ctrl* (Bertazzi et al., 2015). However, it tackles one control at a time. While, $Policy\ M^{+}$ with ABC algorithm employ a set of controls. This is to capture more possible decisions in each state (to explore the space to find good actions).

The flexibility and robustness of the ABC algorithm allow it to be designed for generating promising controls. The proposed ABC algorithm (referred to as *ABC_Control*) generate controls that give the near optimal solution by modifying the decision variable z_{it} indicating which retailer to visit (or not to visit) in the state, t considered. *ABC_Control* considers the impact of the decision taken by updating the fitness value comprise of the sum of inventory and stock out at retailer, and transportation cost. *ABC_Control* as mentioned in the previous chapters consists of three main phases, employed bee phase, onlooker bee phase and scout bee phase. The details of the algorithm is explained below.

The initial food source (initial solution) is represented as a string of binary numbers. The whole solution $a_j = (z_{1t}, z_{2t}, \dots, z_{Mt})$, $j = 1, \dots, n$ represents a food source. The initialization phase is given in **STEP 1** whilst the declaration of parameters l, $LIMIT$, *iteration* and *MAXITER* are presented in **STEP 2**. **STEP 3** is dedicated to the 'bee' phases: where **STEP 3.1** present the employed bee phase. A swap of the binary decision variables between two randomly selected retailers is carried out in order to improve the solution. The onlooker bees select the best food source by observing the employed bees performing the waggle dance and this is done by the using the tournament selection method (Goldberg & Deb, 1991), given in **STEP 3.2**. Once the selection is made, an insertion method (an exchange from 0 to 1 and *vice-versa*) is carried out where the unvisited retailer (0) is change to visited (1) and *vice-versa*. Once l reaches $LIMIT$ the current food sources are abandoned and replaced by randomly generated food sources. This represents the scout bee phase given as **STEP 3.3**. The *ABC_Control* algorithm is given as follows:

INPUT: inventory level, $\bar{I}_{it}$ and demand r_{it} for $i \in M$ in period t.

STEP 1 Initialization Phase

1.1 Generate randomly n number of solutions (food sources). Each solution represented by binary numbers, z_{it} with number of retailers, $i \in M$; where 1 indicates to visit and 0 otherwise.

1.2 Denote each solution as $a_j, j = 1, \dots, n$ where $a_j = (z_{1t}, z_{2t}, \dots, z_{Mt})$. The notation t is drop as only 1 period considered at a time. Hence, $a_j = (z_1, z_2, \dots, z_M)$. Evaluate the fitness value for each food source (using cost function g in subsection 5.3.1.4). Assign an employed bee to each food source.

STEP 2 Set $iteration = 0$ and $l_1 = l_2 = \cdots = l_n = 0$. Declare the value of $LIMIT$ and the maximum number of iterations, $MAXITER$.

STEP 3 Repeat the following until the stopping condition, $MAXITER$ is met.

3.1 Employed Bee Phase

a. For each food source, a_j :

i. Find all the visiting retailers ($z_i = 1$), and then calculate the product of penalty cost and the demand ($d_i \times r_{it}$). Determine retailer i (food source) that gives the minimum value and denote as sc_1.

ii. Find all the non-visiting retailers ($z_i = 0$), and then calculate $e_i = d_i(\bar{I}_{it} - r_{it})$. Denote retailer i that gives the minimum e_i value as sc_2. Note that e_i can be negative which show that demand in retailer i is not fulfill.

iii. Do swap between z_{sc1} and z_{sc2} of retailer sc_1 and sc_2. Assign the new solution found as $\tilde{a}_j$. Evaluate the fitness value for the new $\tilde{a}_j$.

b. *If* $g(\tilde{a}_j) < g(a_j)$; replace the old food source with a new food source, $a_j \leftarrow \tilde{a}_j$ and set $l_j = 0$. *Else* set $l_j = l_j + 1$.

3.2 Onlooker Bee Phase

a. Set $G_j = \emptyset, j = 1, \dots, n$, where G_j is the set of neighboring solutions of food source j.

b. For each onlooker bee,

i. Select a food source, a_j, using tournament selection method (Goldberg & Deb, 1991).

ii. Apply a neighborhood operator, insertion method on selected a_j. Find all the non-visiting retailers ($z_i = 0$), and then calculate $e_i = d_i(\bar{I}_{it} - r_{it})$. Denote retailer i that gives the minimum e_i value as ic. Change the z_{ic} to 1. Denote the new found solution as $\tilde{a}_j$.

iii. $G_j = G_j \cup \tilde{a}_j$.

c. For each food source a_j and $G_j \neq 0$.

i. Set $\hat{a}_j \in argmin_{\tilde{a} \epsilon G_j} g(\tilde{a})$.

ii. *If* $f(\hat{a}_j) < f(a_j)$; replace the old food source with the new one; $a_j \leftarrow \hat{a}_j$ and set $l_j = 0$. *Else* set $l_j = l_j + 1$.

3.3 Scout Bee Phase

For each food source, a_j, if $l_j = LIMIT$, replace a_j with a randomly generated solution.

$iteration = iteration + 1$.

STEP 4 Output: The best food source found so far.

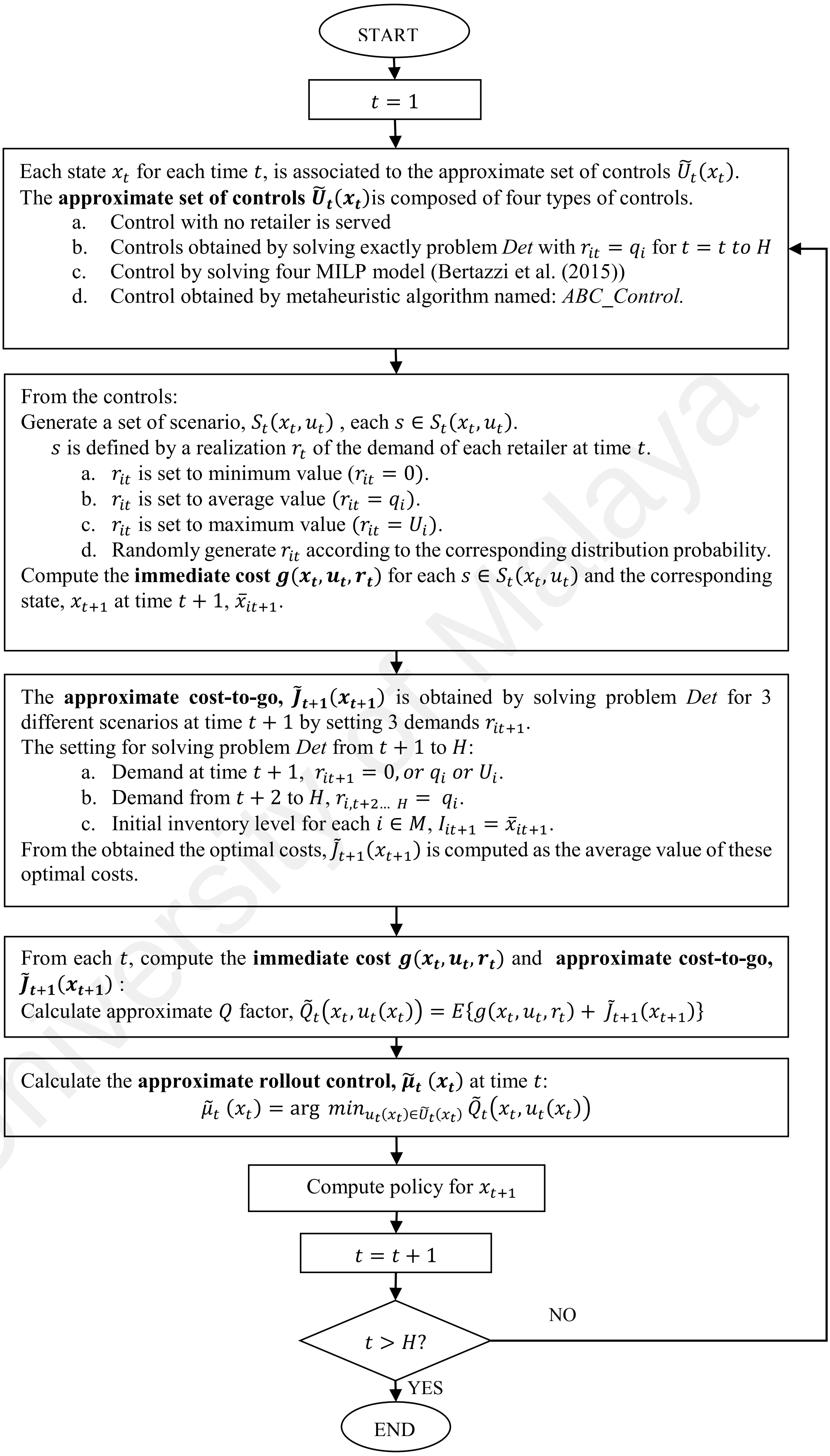


Figure 5.2: Flow chart of *Policy* M^+.

5.5 Computational Results and Discussions

All computations are performed on a 16GB RAM computer with a 3.1GHz processor. The enhanced hybrid rollout algorithm (*Policy* M^{+}) was coded and run using MATLAB 8.1. The MILP (problem *Det*) to obtain the approximate cost-to-go values, $\tilde{J}_{t+1}(x_{t+1})$ and the controls are also coded in MATLAB and run using CPLEX 12.6 connector.

5.5.1 Dataset

The performance of *Policy* M^{+} is tested on the same dataset as Bertazzi et al. (2015), where the authors fixed the dataset in Archetti et al. (2007) to suit for stochastic demand. The 60 datasets consist of 5, 10 and 20 retailers with 3 and 6 periods. The datasets are also divided into two variations of inventory holding cost: Low inventory cost (LC) and high inventory cost (HC). See http://www.leandro-coelho.com/instances to access the instances.

The dataset is segregated according to the number of retailers where each has 5 different replicates. The designated name of the dataset maintains the original naming. For example, abs1n5 is the first replicate for 5 retailers and abs4n10 is the fourth replicate for 10 retailers.

The criteria of data are set to follow the settings given in Bertazzi et al. (2015) except for the number of additional scenarios used. The number of scenarios generated is 123 inclusive of 120 randomly generated demands instead of 100 (Bertazzi et al., 2015), that follows a probability distribution and additional 3 demands considered are: all zeros demand, average demand, q_i and maximum level, U_i.

The reason to increase the number of scenarios is to capture as many combinations of demand that are likely to happen. The number of additional scenarios is determined by carrying out a simple experiment. *Policy* M^{+} is tested for four different sizes of datasets, that are 50, 100, 120 and 150. The experiment is carried out on the smallest and the largest dataset, abs1n5 (with $H = 3$ and LC) and abs5n20 (with $H = 6$ and HC) where each dataset is run for 10 times. Results of the different number of additional scenarios are tabulated in Table 5.1.

The difference in the average computational time between the two datasets is quite significant (549.77, 1049.16, 1377.34 and 1566.48 seconds for 50, 100, 120 and 150 number of additional scenarios respectively) which indicate that the larger the dataset the longer time needed to compute the results. When compared between 120 and 150, the difference in average time of abs1n5 H3LC is only 101.15 seconds, however in abs5n20 H6HC the difference increases to 290.29 seconds. As real size datasets are considered in this thesis, the number of additional scenarios chosen is 120 instead of 150 because of the time taken to obtain the results. Hence a trade-off between computational time and the solution quality.

Table 5.1: Analysis of Different Number of Additional Scenarios.

Dataset	Number of Additional Scenarios	Average Expected Cost	Average Time	Average Delivery Quantity	Average Number of visits	Standard Deviation
abs1n5 H3LC	50	104.38	166.65	406.90	5.20	2.16
	100	105.27	320.60	405.20	5.90	2.14
	120	104.14	382.72	406.90	5.00	3.16
	150	104.12	483.87	395.60	5.30	2.45
abs5n20 H6HC	50	8817.68	716.42	5837.20	63.10	474.61
	100	8766.67	1369.76	5832.00	60.50	557.47
	120	8614.32	1760.06	5951.40	58.40	470.96
	150	8611.07	2050.35	5902.90	61.30	456.60

Table 5.2 presents the summary for all datasets. The number of bees in the *ABC_Control* comprises of 30 bees employed bees and 15 onlooker bees. The maximum number of iteration, MAX_ITER is set to 10. The exploitation parameter $LIMIT$ is set at 5.

Table 5.2: Characteristics of the datasets.

Criteria	**Data Setting**
Planning Horizon, H	3,6
Number of Retailers, n	5,10 ,20
Inventory cost at Retailers, h_i	LC: [0.01,0.05] HC: [0.1,0.5]
Inventory cost at Supplier, h_0	LC: 0.03 HC: 0.3
Transportation Capacity, C	$3/2 \sum_{i \in M} q_i$
Maximum Inventory Level at Retailer, U_i	$q_i g_i$ where g_i is generate random from {2,3}
Fixed Transportation Cost, f	10
Average demand, q_i	Randomly generated integer from [10,100]
Demand of retailer $i \in M$ follow Distribution Probability, D_i	Binomial Distribution: $B\left(U_i, \frac{q_i}{U_i+1}\right)$ Uniform Distribution: $U(1, U_i)$
Initial Inventory level at Supplier, I_{i0}	$U_i - q_i$
Initial Inventory Level at Retailer, I_{00}	$\sum_{i \in M} U_i$
Quantity produced by the supplier at each time t, p_t	$\sum_{i \in M} q_i$
Stock Out penalty cost at the retailers, d_i	$1.5f + h_i U_i$
Number $S_t(x_t, u_t)$ of additional scenarios generated at time t	120

Policy M^+ is tested for demands that follow two different discrete probability distributions, binomial and uniform. Demands for binomial probability distribution is similar to the one given in Bertazzi et al. (2015). Whilst for uniform probability distribution the demand is assumed to be equally distributed between 1 and U_i.

5.5.2 Results and Discussions

The performance of the algorithm developed *Policy* M^+ is run for 10 times and tested on the 60 datasets discussed earlier. Results of *Policy* M^+ are the average of the expected costs found for the 10 runs and they are compared with the bounds obtained from CPLEX. The bounds represent the average of expected costs obtained by optimally solving problem *Det* for 10 different trajectories of demand and the 10 trajectories follow the probability distribution considered. The model is solved to optimality with a tolerance gap of $\leq 0.05\%$. Detail results of each of the 10 runs are given in Appendix C.

5.5.2.1 Binomial Distribution

Table 5.3 and 5.4 present the results for binomial distribution for case $H = 3$ and $H = 6$. The first column shows the datasets, the other four columns are the ratio between the average expected cost of the policies and the bounds. The second and fourth columns (labelled as $Policy\ M$) are the results of policy obtained from Bertazzi et al. (2015). The third and fifth columns show the results obtained for $Policy\ M^+$. Small ratios indicate the closeness of the average expected cost found to the bounds.

Table 5.3 shows that the $Policy\ M^+$ performs significantly better with the average of 0.85 (LC) and 0.81 (HC) when compared to *Policy M* with higher averages of 4.26 (LC) and 2.85 (HC) respectively. Table 5.4 presents for the case $H = 6$, where $Policy\ M^+$ performs significantly better with 1.03 (LC) and 0.97 (HC) when compared to *Policy M* with 3.37 (LC) and 2.58 (HC) respectively.

Table 5.3: Comparison of Bertazzi et al. (2015) and $Policy\ M^+$ for $H = 3$.

	LC		HC	
Dataset	**M**	M^+	**M**	M^+
abs1n5	2.78	0.89	0.56	0.79
abs2n5	10.75	0.92	1.51	0.83
abs3n5	7.17	0.88	5.87	0.85
abs4n5	3.98	0.98	0.82	0.97
abs5n5	11.88	0.88	6.56	0.77
abs1n10	2.21	0.83	2.89	0.80
abs2n10	3.03	0.80	2.31	0.80
abs3n10	3.98	0.84	3.39	0.78
abs4n10	1.53	0.84	1.54	0.83
abs5n10	4.8	0.83	4.97	0.81
abs1n20	1.98	0.81	2.45	0.79
abs2n20	2.13	0.79	1.91	0.78
abs3n20	2.84	0.79	2.98	0.79
abs4n20	1.43	0.83	1.52	0.79
abs5n20	3.35	0.78	3.5	0.78
Average	4.26	0.85	2.85	0.81

Table 5.4: Comparison of Bertazzi et al. (2015) and *Policy* M^+ for $H = 6$.

	LC		HC	
Dataset	**M**	$\boldsymbol{M^+}$	**M**	$\boldsymbol{M^+}$
abs1n5	1.68	1.08	0.47	0.92
abs2n5	6.45	1.06	1.45	1.00
abs3n5	5.54	1.06	4	0.93
abs4n5	1.67	1.17	1.33	0.99
abs5n5	8.91	1.08	5.59	0.95
abs1n10	2.63	1.12	2.94	1.10
abs2n10	3.26	1.04	2.25	1.04
abs3n10	4.06	1.04	3.61	1.01
abs4n10	1.78	1.14	1.72	1.07
abs5n10	4.66	1.04	4.75	1.03
abs1n20	1.69	0.96	2.22	0.92
abs2n20	1.79	0.92	1.62	0.88
abs3n20	2.17	0.94	2.28	0.97
abs4n20	1.19	0.90	1.25	0.91
abs5n20	3.07	0.91	3.17	0.88
Average	3.37	1.03	2.58	0.97

The details of the bounds and the average expected costs of *Policy* M^+ contributed to the ratios (in Table 5.3 and 5.4) are presented in Table 5.5 and 5.6 for $H = 3$ and $H = 6$ respectively. It is noted that bound values are approximation values, hence the results of *Policy* M^+ can be lower than the respective bounds.

Table 5.5: Bound and Average Expected Cost of $H = 3$.

	LC		HC	
Dataset	**Bound**	**Policy** $\boldsymbol{M^+}$	**Bound**	**Policy** $\boldsymbol{M^+}$
abs1n5	116.587	104.141	973.032	769.539
abs2n5	107.854	99.647	876.047	729.577
abs3n5	170.95	150.92	1423.119	1215.459
abs4n5	91.651	90.274	658.496	637.882
abs5n5	156.581	138.438	1362.43	1046.084
abs1n10	332.293	276.019	3097.498	2473.194
abs2n10	284.999	226.672	2558.475	2056.487
abs3n10	271.058	226.422	2444.741	1906.691
abs4n10	268.549	225.591	2386.778	1969.496
abs5n10	345.057	285.943	3186.77	2567.474
abs1n20	524.164	423.362	5011.089	3944.349
abs2n20	542.259	427.626	5065.119	3957.786
abs3n20	565.072	448.249	5319.794	4210.263
abs4n20	446.184	369.191	4172.24	3298.767
abs5n20	592.784	465.027	5654.347	4410.13

Table 5.6: Bound and Average Expected Cost of $H = 6$.

	LC		HC	
Dataset	**Bound**	**Policy M^+**	**Bound**	**Policy M^+**
abs1n5	330.745	357.91	2885.835	2659.593
abs2n5	302.656	321.316	2555.001	2547.267
abs3n5	301.96	320.054	2523.556	2350.48
abs4n5	264.506	308.849	2124.719	2096.109
abs5n5	290.561	314.216	2448.966	2338.715
abs1n10	528.249	591.787	4835.01	5324.851
abs2n10	424.569	442.566	3733.421	3883.41
abs3n10	480.73	500.084	4318.682	4348.636
abs4n10	472.217	539.712	4079.924	4354.727
abs5n10	612.391	635.867	5561.207	5732.845
abs1n20	955.634	919.55	9053.961	8360.014
abs2n20	1008.054	932.061	9423.271	8319.174
abs3n20	909.658	856.094	8426.327	8164.137
abs4n20	830.321	750.644	7765.206	7027.998
abs5n20	1029.571	939.534	9738.457	8614.324

Table 5.7 displays the standard deviation of the expected cost for all cases (LC and HC) for $H = 3$ and $H = 6$. For small dataset $H = 3$ and low inventory cost (LC) the results indicate that the variation (standard deviation) is relatively small. However, the same dataset with HC the standard deviation is relatively larger when compare to LC. The same pattern is observed for the case with $H = 6$ and this may be due to the larger computational time needed to obtain the results for HC.

Table 5.7: Standard Deviation of the Expected Cost.

Dataset	**$H = 3$**		**$H = 6$**	
	LC	**HC**	**LC**	**HC**
abs1n5	3.1607	32.7188	27.2126	224.6598
abs2n5	5.2421	41.6401	24.1900	279.3559
abs3n5	2.5987	58.8626	20.3655	249.5872
abs4n5	3.2655	32.8462	20.9057	229.0479
abs5n5	4.4177	54.3720	26.1413	232.6137
abs1n10	1.4490	60.8067	51.9248	700.8903
abs2n10	6.1095	98.3450	35.5782	486.6817
abs3n10	7.3339	51.8631	51.2368	460.8116
abs4n10	13.0933	135.6753	25.5585	362.0439
abs5n10	9.5817	160.5237	40.3718	635.7126
abs1n20	16.9701	155.6746	74.7839	523.9594
abs2n20	12.8410	133.9491	70.7314	75.7653
abs3n20	13.8250	88.6653	82.0499	1110.3675
abs4n20	29.3820	109.6524	20.9948	263.7426
abs5n20	17.0944	247.2227	60.8557	470.9574

(a) ***Analysis of Controls***

An in-depth analysis of the approximate set of controls $\tilde{\mathcal{U}}$ used in the algorithm which contributed to the small $g(x_t, u_t, r_t)$ cost and eventually to the approximate $\tilde{Q}$ factor value at state t. These results in small ratios when compared to the bounds observed. The aim of the analysis is to determine the benefit of using $ABC\_Control$ as compared to the controls proposed in Bertazzi et al. (2015). The controls proposed in Bertazzi et al. (2015) is labelled as $CTRL$ ((i), (ii) and (iii) in subsection 5.3.2.2) while the details of the $ABC\_Control$ (labelled as $CTRLabc$) is discussed in subsection 5.4.

The analysis of the controls is executed for all 10 runs, for example for $H = 3$, the frequency of selecting either $CTRL$ or $CTRLabc$ that contributed to the calculation of the approximate $\tilde{Q}$ factor is recorded and the percentage is determined by dividing by 30 (number of periods (H) multiply by the number of runs). Similar method is applied for $H = 6$. It is observed that $CTRLabc$ contributes on average 93.78% and 86.22% in LC and HC for $H = 3$, and 94.67% and 91.33% in LC and HC for $H = 6$.

Minimum percentage contributions of $CTRLabc$ for $H = 3$ and $H = 6$ for LC are 76.67% (abs2n10) and 83.33% (abs5n5) respectively. However, the percentage contribution of $CTRLabc$ is slightly less in cases HC when $H = 3$ and $H = 6$ (63.33% (abs3n20 and abs2n10) and 75% (abs5n10) respectively). The analyses of the controls are as tabulated in Figure 5.8 and 5.9.

Table 5.8: The percentage of controls chosen for case $H = 3$.

	LC		HC	
Dataset	$CTRLabc$	$CTRL$	$CTRLabc$	$CTRL$
abs1n5	100	0	100	0
abs2n5	90	10	90	10
abs3n5	96.67	3.33	93.33	6.67
abs4n5	96.67	3.33	76.67	23.33
abs5n5	96.67	3.33	93.33	6.67
abs1n10	100	0	83.33	16.67
abs2n10	76.67	23.33	63.33	36.67
abs3n10	96.67	3.33	80	20
abs4n10	90	10	100	0
abs5n10	80	20	70	30
abs1n20	93.33	6.67	90	10
abs2n20	100	0	96.67	3.33
abs3n20	93.33	6.67	63.33	36.67
abs4n20	100	0	100	0
abs5n20	96.67	3.33	93.33	6.67
Average	93.78	6.22	86.22	13.78

Table 5.9: The percentage of controls chosen for case $H = 6$.

	LC		HC	
Dataset	$CTRLabc$	$CTRL$	$CTRLabc$	$CTRL$
abs1n5	98.33	1.67	96.67	3.33
abs2n5	91.67	8.33	88.33	11.67
abs3n5	88.33	11.67	93.33	6.67
abs4n5	96.67	3.33	98.33	1.67
abs5n5	83.33	16.67	83.33	16.67
abs1n10	98.33	1.67	96.67	3.33
abs2n10	100	0	83.33	16.67
abs3n10	96.67	3.33	96.67	3.33
abs4n10	90	10	100	0
abs5n10	95	5	75	25
abs1n20	96.67	3.33	95	5
abs2n20	100	0	100	0
abs3n20	93.33	6.67	86.67	13.33
abs4n20	98.33	1.67	100	0
abs5n20	93.33	6.67	76.67	23.33
Average	94.67	5.33	91.33	8.67

(b) ***Analysis of Number of Visits and Delivery Quantity***

The result of the DSIRP is a policy (the control) and the best expected cost is used as a guideline to measure the quality of the policy, thus it is essential to see the pattern of the number of visits and the delivery quantity. The number of visits shows the frequency of the retailers visited, hence it influences the delivery quantity.

An analysis of the average number of visits and the average delivery quantity out of 10 runs are carried out to see the effect of the low (LC) and high (HC) inventory holding cost as the OU policy is adopted. The average delivery quantity (DQ) and the average number of visits (NV) are presented as bar plot and marker line respectively in Figure 5.3 and Figure 5.4.

It is predicted that with high inventory cost (HC), the number of visits is frequent but with lower delivery quantity. However, with the OU policy adopted (to minimize the stock-out) Figure 5.3 illustrates that delivery quantities of HC are higher than LC in all dataset cases (except for abs2n5, abs2n10 and abs3n20). This behavior is the results of the OU policy adopted (to minimize the stock-out). The average number of visits given in Figure 5.3 justifies the relation of the number of average delivery quantity and the OU policy.

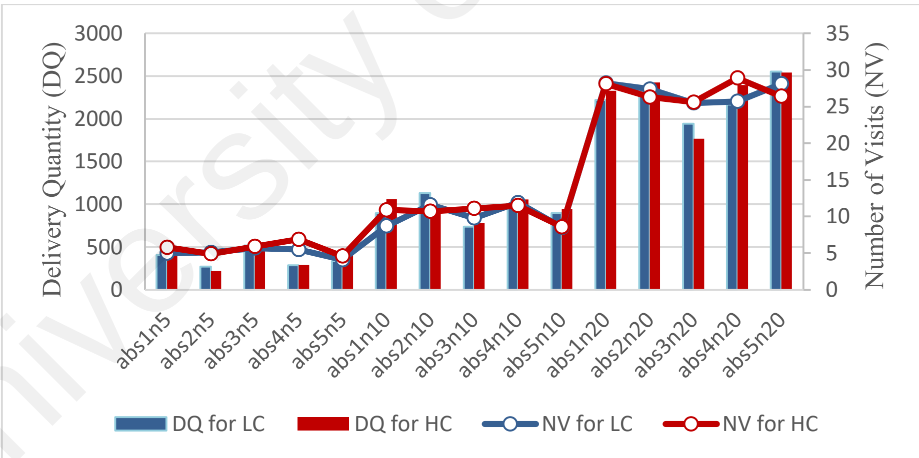


Figure 5.3: Comparison of LC and HC for Delivery Quantity and Number of Visits in $H = 3$ cases.

Figure 5.4 illustrates the results of average delivery quantity and the average number of visits for $H = 6$ and the results affirm that the OU policy adopted leads to a higher number of average delivery quantities and number of visits, resulting in relatively small amount of stock out quantities.

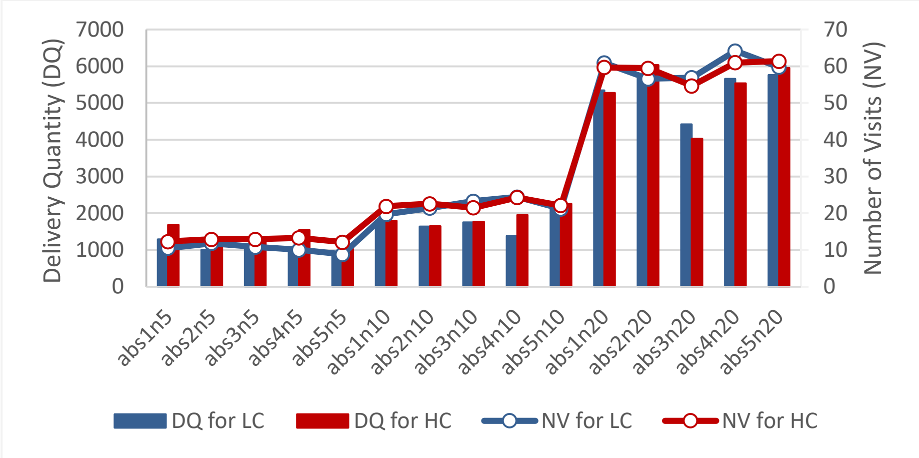


Figure 5.4: Comparison of LC and HC for Delivery Quantity and Number of Visits in $H = 6$ cases.

The average computational times taken over 10 runs for the cases $H = 3$ and $H = 6$ for both LC and HC are presented in Table 5.10. Datasets of 5 retailers with 3 periods in LC cases run for not more than 414.26 seconds (around 6.9 minutes). Cases $H = 6$ require more time to obtain the results, as seen in the table where the longest time is 2102.48 (around 35.04 minutes) for dataset abs5n20 (LC). It is expected as the running time is proportional to the size of the dataset.

Table 5.10: Average time taken for case $H = 3$ and $H = 6$ for both LC and HC.

	$H = 3$		$H = 6$	
Dataset	**LC**	**HC**	**LC**	**HC**
abs1n5	382.72	382.51	967.29	963.59
abs2n5	399.03	382.25	961.83	957.75
abs3n5	414.26	413.75	1023.42	1118.77
abs4n5	407.15	408.38	993.83	976.38
abs5n5	401.36	389.99	1026.99	1119.98
abs1n10	442.43	444.22	1240.54	1229.03
abs2n10	450.42	449.08	1266.56	1306.83
abs3n10	436.46	433.01	1272.98	1262.18
abs4n10	445.05	437.95	1178.38	1132.77
abs5n10	452.76	441.95	1320.88	1293.67
abs1n20	510.16	509.02	1918.95	1674.35
abs2n20	518.63	511.80	1902.48	1676.42
abs3n20	518.76	511.88	2002.20	1710.55
abs4n20	508.48	513.87	1846.77	1702.21
abs5n20	518.68	512.79	2102.48	1760.06

5.5.2.2 Uniform Distribution

Bertazzi et al. (2015) carry out a managerial insight and shows that stock out are completely avoided for the demand that follows binomial distribution. However, some stock out are observed for large datasets in the binomial distribution implemented in this thesis. It was pointed out in Bertazzi et al. (2015) that there exists some stock out if the demand follows uniform distribution. Motivated by this, the performance of *Policy* M^{+} is then tested for uniform distribution. No comparison can be done as only limited results are provided in the literature.

Table 5.11 and 5.12 show the bounds, expected costs from *Policy* M^{+} and the ratios for cases $H = 3$ and $H = 6$ respectively. The small ratios that are 0.634 and 0.602 for both LC and HC for $H = 3$; 0.730 and 0.706 for both LC and HC in $H = 6$ confirm the reliability and significance of *Policy* M^{+}. An optimality gap $\leq 0.05\%$ is used in the calculation of the bounds.

Table 5.11: Bounds, Policy M^{+} and Ratios of $H = 3$.

	LC			HC		
Dataset	**Bound**	**Policy M^{+}**	**Ratio**	**Bound**	**Policy M^{+}**	**Ratio**
abs1n5	414.408	96.009	0.232	1021.843	611.575	0.599
abs2n5	178.215	94.682	0.531	1555.542	592.814	0.381
abs3n5	163.447	132.529	0.811	1733.905	833.561	0.481
abs4n5	92.758	91.712	0.989	691.442	533.637	0.772
abs5n5	510.306	131.241	0.257	3928.679	798.278	0.203
abs1n10	336.209	200.514	0.596	2842.931	1717.879	0.604
abs2n10	264.081	187.696	0.711	2424.093	1497.964	0.618
abs3n10	327.194	164.842	0.504	2234.536	1433.43	0.641
abs4n10	255.98	176.326	0.689	2264.42	1449.028	0.640
abs5n10	317.455	218.24	0.687	2928.689	1749.574	0.597
abs1n20	474.649	343.031	0.723	4541.849	3095.669	0.682
abs2n20	482.873	344.725	0.714	4589.681	3442.798	0.750
abs3n20	509.415	347.33	0.682	4714.131	3342.299	0.709
abs4n20	419.981	291.77	0.695	3956.808	2727.044	0.689
abs5n20	541.379	376.151	0.695	5170.776	3452.568	0.668
Average			0.634			0.602

Table 5.12: Bounds, Policy M^{+} and Ratios of $H = 6$.

	LC			HC		
Dataset	**Bound**	**Policy M^{+}**	**Ratio**	**Bound**	**Policy M^{+}**	**Ratio**
abs1n5	591.874	262.617	0.444	2846.65	1809.374	0.636
abs2n5	298.297	253.276	0.849	3283.882	1576.625	0.480
abs3n5	639.95	289.17	0.452	2346.58	1750.506	0.746
abs4n5	389.407	273.067	0.701	2367.164	1379.911	0.583
abs5n5	472.714	246.55	0.522	2892.641	1691.6	0.585
abs1n10	450.915	412.137	0.914	3940.873	2995.954	0.760
abs2n10	363.744	275.189	0.757	3092.475	2275.377	0.736
abs3n10	420.611	323.748	0.770	3521.457	2721.962	0.773
abs4n10	406.577	352.606	0.867	3486.832	2580.356	0.740
abs5n10	531.216	403.907	0.760	4749.874	3040.35	0.640
abs1n20	790.79	602.599	0.762	7132.769	6258.454	0.877
abs2n20	822.368	680.739	0.828	7673.661	5994.199	0.781
abs3n20	723.95	573.22	0.792	6685.161	5347.55	0.800
abs4n20	717.947	552.764	0.770	6857.054	5033.556	0.734
abs5n20	841.554	644.803	0.766	8100.847	5845.664	0.722
Average			0.730			0.706

Table 5.13 gives the standard deviation of the 10 runs expected costs found using *Policy* M^{+}. Small sizes datasets (5 retailers and 3 periods) with LC has a small standard deviation, while as the problem size increases, the standard deviation as expected is larger.

Table 5.13: Standard Deviation of the Expected Cost.

Dataset	$H = 3$		$H = 6$	
	LC	**HC**	**LC**	**HC**
abs1n5	2.9230	49.0090	23.3139	234.9485
abs2n5	9.7173	75.8771	36.4404	314.2647
abs3n5	14.9759	94.9102	30.5856	352.5862
abs4n5	4.9424	71.1423	49.1477	168.1798
abs5n5	9.5049	75.3798	26.6811	265.0979
abs1n10	16.0551	164.8535	87.1616	399.4785
abs2n10	22.4713	65.3810	41.2445	236.6543
abs3n10	9.3678	155.2782	34.9456	268.2076
abs4n10	16.8540	118.7040	57.2744	403.0706
abs5n10	20.1721	154.1111	58.4284	399.2638
abs1n20	25.3292	301.8258	29.0771	766.5922
abs2n20	17.1987	366.5808	81.0422	531.3821
abs3n20	30.4150	269.2325	48.6632	892.1711
abs4n20	15.1040	258.0544	27.7818	182.7531
abs5n20	38.6945	184.4743	62.4919	217.9113

(a) ***Analysis of Controls***

An analysis of controls that contribute to the policy is also carried out to see the performance of the ABC control proposed. $CTRLabc$ contributes on average 98.44% and 96.44% in the of case $H = 3$; 96.22% and 95.78% in the of case $H = 6$ for LC and HC respectively. Table 5.14 and 5.15 illustrate the results. $CTRLabc$ performs very well for LC, however in HC, $CTRLabc$ performs slightly poor. This may be due to the ability of $CTRL$ to solve MILP optimally.

Table 5.14: The percentage of controls chosen for case $H = 3$.

	LC		HC	
Dataset	$CTRLabc$	$CTRL$	$CTRLabc$	$CTRL$
abs1n5	100	0	96.67	3.33
abs2n5	96.67	3.33	96.67	3.33
abs3n5	96.67	3.33	93.33	6.67
abs4n5	96.67	3.33	93.33	6.67
abs5n5	100	0	96.67	3.33
abs1n10	100	0	100	0
abs2n10	100	0	93.33	6.67
abs3n10	93.33	6.67	100	0
abs4n10	96.67	3.33	93.33	6.67
abs5n10	100	0	93.33	6.67
abs1n20	100	0	93.33	6.67
abs2n20	100	0	96.67	3.33
abs3n20	100	0	100	0
abs4n20	96.67	3.33	100	0
abs5n20	100	0	100	0
Average	98.44	1.56	96.44	3.56

Table 5.15: The percentage of controls chosen for case $H = 6$.

	LC		HC	
Dataset	$CTRLabc$	$CTRL$	$CTRLabc$	$CTRL$
abs1n5	95	5	88.33	11.67
abs2n5	93.33	6.67	91.67	8.33
abs3n5	95	5	91.67	8.33
abs4n5	90	10	93.33	6.67
abs5n5	81.67	18.33	93.33	6.67
abs1n10	98.33	1.67	93.33	6.67
abs2n10	100	0	96.67	3.33
abs3n10	96.67	3.33	98.33	1.67
abs4n10	98.33	1.67	96.67	3.33
abs5n10	96.67	3.33	98.33	1.67
abs1n20	100	0	95	5
abs2n20	100	0	100	0
abs3n20	98.33	1.67	100	0
abs4n20	100	0	100	0
abs5n20	100	0	100	0
Average	96.22	3.78	95.78	4.22

(b) ***Analysis of Number of Visits and Delivery Quantity***

An analysis of the average delivery quantity and the average number of visits is carried out. It is expected the results follow the previous analysis (binomial distribution); that the delivery quantity for HC is higher than LC for both cases $H = 3$ in all datasets (except for abs3n10, abs4n10, abs2n20 and abs3n20). This is portrayed in Figure 5.5. The higher number of visits influences the delivery quantity, as the OU policy ensures that quantity delivered must reach its maximum inventory level. Hence the higher the number of visits results in a higher amount of average delivery quantities. It is noted that for dataset abs4n10 even though the number of visits of HC are higher but it is observed that the delivery quantities in HC are lower than LC. Figure 5.6 shows the average number of visits for $H = 6$ for LC and HC. All HC datasets have higher delivery quantities than LC, except for abs3n10, abs1n20, abs3n20 and abs4n20. These are influenced by its corresponding the number of visits.

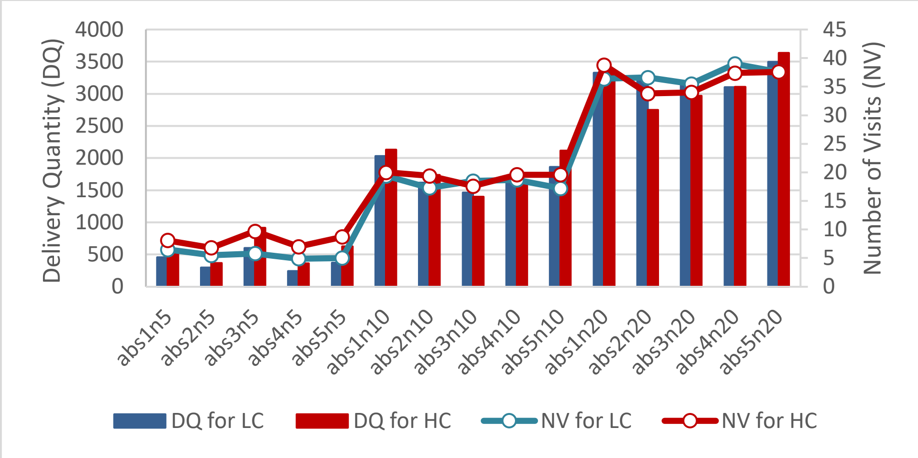


Figure 5.5: Comparison of LC and HC for Delivery Quantity and Number of Visits in $H = 3$ cases.

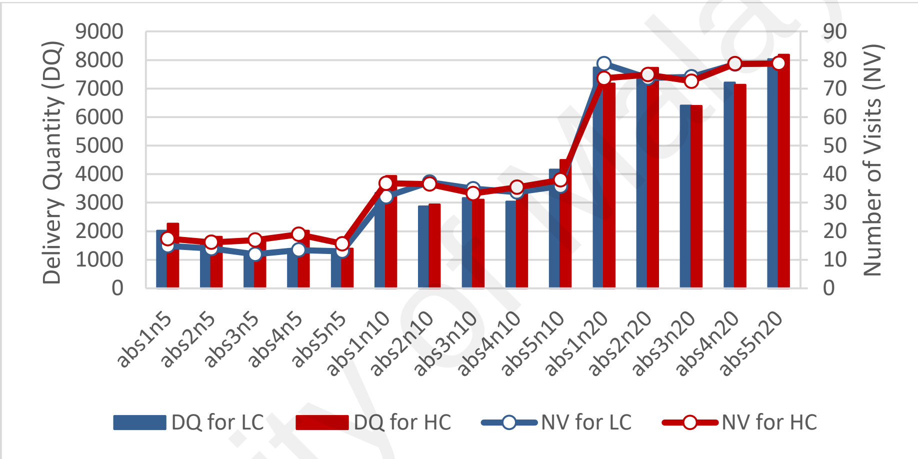


Figure 5.6: Comparison of LC and HC for Delivery Quantity and Number of Visits in $H = 6$ cases.

5.5.3 Analysis between the Binomial and Uniform Probability Distribution

The analysis between having demands that follows binomial and the uniform probability distribution is carried out. The analysis is based on the following two criteria that is the average number of visits and the average stock out quantity. It is observed from Figure 5.7 the average number of visits for demands that follow uniform distribution are higher for LC in all datasets except in abs4n5 for all the case $H = 3$. The delivery quantities in dataset abs4n5 for the binomial distribution are higher than the uniform distribution. Similar pattern shown in Figure 5.8 for case $H = 3$ for HC; uniform distribution have a higher number of visits than binomial distribution in all datasets.

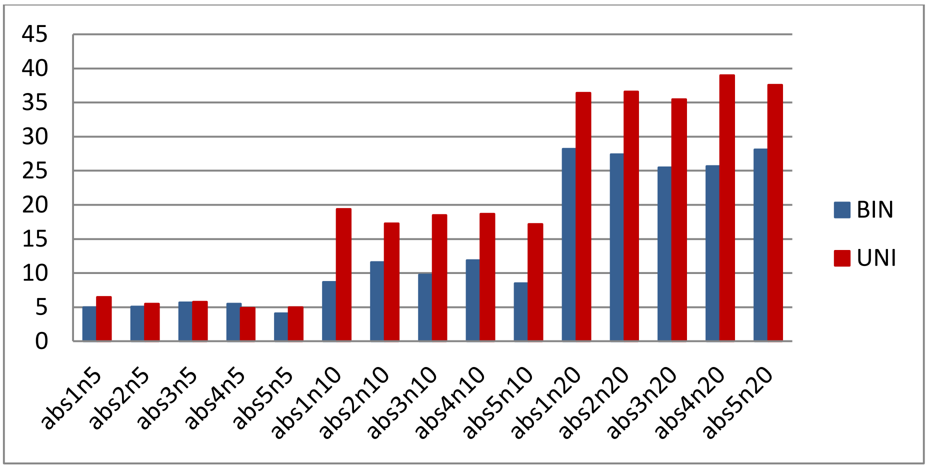


Figure 5.7: Average number of visits for case $H = 3$, LC.

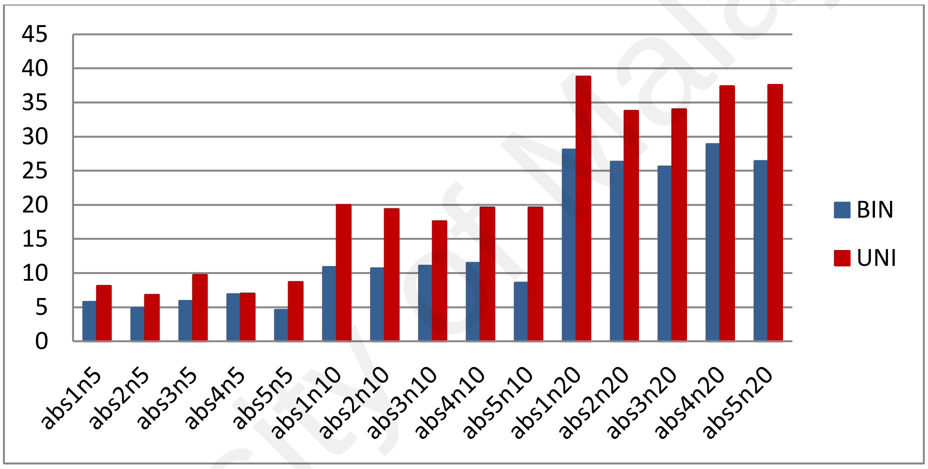


Figure 5.8: Average number of visits for case $H = 3$, HC.

The analysis is extended to $H = 6$ datasets for both LC and HC. Results show that expected cost found for demands that follow uniform distribution have a higher number of visits when compared to the binomial distribution. However, the gap between both distributions are not more than 19 visits in LC cases and 17.9 visits in HC cases. Results are shown in Figure 5.9 and 5.10.

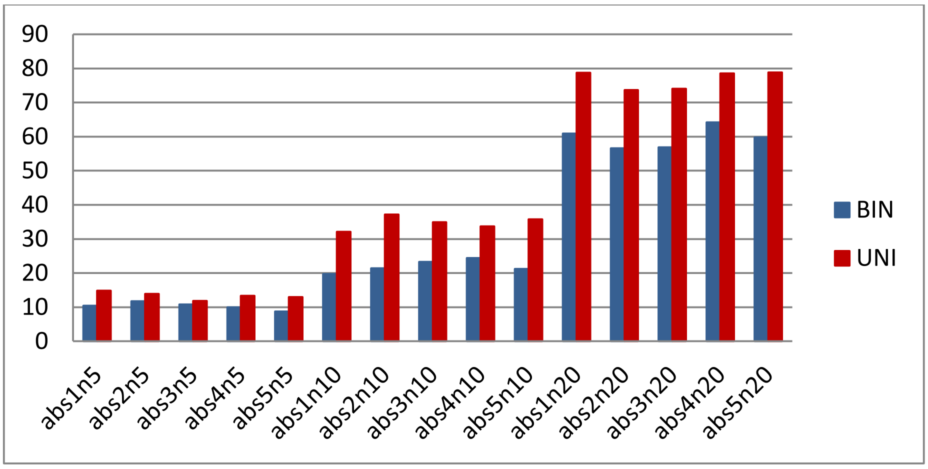


Figure 5.9: Average number of visits for case $H = 6$, LC.

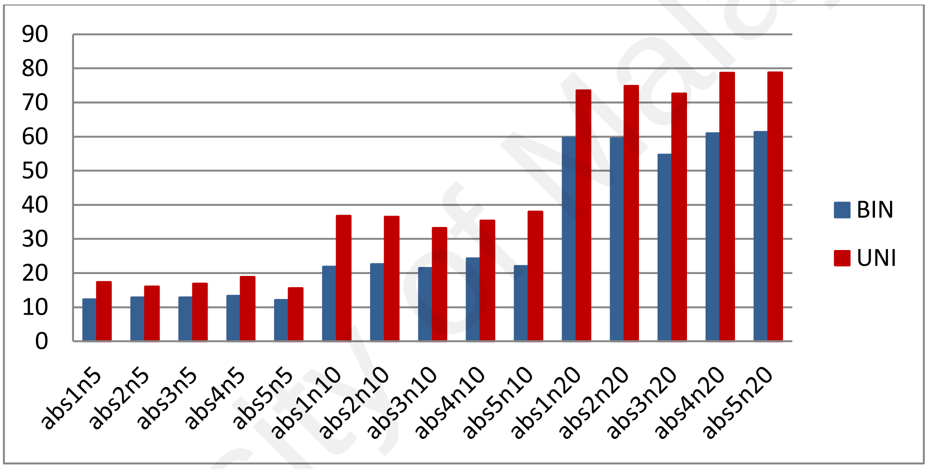


Figure 5.10: Average number of visits for case $H = 6$, HC.

The average stock out quantity for 10 runs is given in Table 5.16. Results show that the stock out quantity is quite independent of the probability distribution as opposed to the observation in Bertazzi et al. (2015). However, it is observed that the stock out quantity from uniform distribution are relatively higher than the binomial distribution. The stock out quantities are completely avoided for all LC datasets of $H = 3$ and $H = 6$ in both probability distributions. Similarly in HC datasets with small number of retailers (5 retailers) the stock out quantity is completely avoided (except for dataset abs1n5-H6HC), however as the number of retailers increases, a small amount of stock out (not more than 1.7 units) exists. It is conjectured that for al cases the OU policy is effective in controlling the stock out. For cases with stock out, frequent visits are a better strategy to avoid stock out. It is however at the expense of high inventory costs.

Table 5.16: Average Stock Out Quantity for cases $H = 3, 6$ and LC, HC.

	LC, $H = 3$		**HC, $H = 3$**		**LC, $H = 6$**		**HC, $H = 6$**	
Dataset	**BIN**	**UNI**	**BIN**	**UNI**	**BIN**	**UNI**	**BIN**	**UNI**
abs1n5	0	0	0	0	0	0	0.1	0
abs2n5	0	0	0	0	0	0	0	0
abs3n5	0	0	0	0	0	0	0	0
abs4n5	0	0	0	0	0	0	0	0
abs5n5	0	0	0	0	0	0	0	0
abs1n10	0	0	0	0	0	0	0	0.1
abs2n10	0	0	0	0	0	0	0	0
abs3n10	0	0	0	0	0	0	0	0
abs4n10	0	0	0	0.1	0	0	0	0.1
abs5n10	0	0	0	0	0	0	0	0.1
abs1n20	0	0	0	0.4	0	0	0.7	1.3
abs2n20	0	0	0.5	0.2	0	0	0.2	1.7
abs3n20	0	0	0.1	0.1	0	0	0.2	1.1
abs4n20	0	0	0.1	0.8	0	0	0.4	0.6
abs5n20	0	0	0.1	0.2	0	0	0	0.4

The computational time for demands that follow uniform distribution is given in Table 5.17.

Table 5.17: Time taken for case $H = 3$ and $H = 6$ for both LC and HC.

	$H = 3$		**$H = 6$**	
DATASET	**LC**	**HC**	**LC**	**HC**
abs1n5	370.62	370.92	892.63	872.50
abs2n5	379.51	352.10	916.48	825.03
abs3n5	405.33	412.44	1004.12	1029.00
abs4n5	427.33	412.20	964.14	897.50
abs5n5	395.96	378.57	1016.89	1045.59
abs1n10	434.78	436.92	1243.74	1178.39
abs2n10	447.63	447.77	1278.94	1231.85
abs3n10	416.90	426.21	1286.84	1174.32
abs4n10	436.29	435.54	1123.20	1043.01
abs5n10	445.49	446.21	1352.40	1231.30
abs1n20	503.61	507.09	1897.84	1580.57
abs2n20	508.05	508.45	1930.41	1591.73
abs3n20	507.04	504.28	2060.28	1630.78
abs4n20	501.80	511.56	1821.44	1518.63
abs5n20	508.84	513.43	2226.42	1670.56

5.6 Summary

This chapter discusses a dynamic stochastic inventory routing problem (DSIRP). An enhanced hybrid rollout algorithm referred to as $Policy\ M^+$ is developed and successfully implemented for the DSIRP. $Policy\ M^+$ is an enhanced version of $Policy\ M$ proposed in Bertazzi et al. (2015). $Policy\ M^+$ embeds additional controls attained using $ABC\_Control$ and the number of scenarios is increased to 120. It is observed the additional number of scenarios and the application of $ABC\_Control$ significantly contribute to the better ratios of $Policy\ M^+$.

Two discrete probability distribution functions are considered, that are binomial distribution and uniform distribution. Performance of $Policy\ M^+$ is compared with $Policy\ M$ where $Policy\ M^+$ found the optimal policies with smaller ratios, which indicate the closeness of the expected cost found to the bound. The comparison is done for demands that follow a binomial probability distribution with the results given in Bertazzi et al. (2015).

$Policy\ M^+$ is further investigate by carry out an analysis of controls between $CTRL$ and $CTRLabc$. $CTRL$ is the controls proposed in Bertazzi et al. (2015) while $CTRLabc$ is the controls generated by Artificial Bee Colony algorithm. Results show that $CTRLabc$ contributes on average 93.78% and 86.22% in LC and HC for $H = 3$, and 94.67% and 91.33% in LC and HC for $H = 6$ to obtain the near optimal policies. Details explanation of the delivery quantity and the number of visits are also discussed. $Policy\ M^+$ is then tested for uniform distribution where the small ratios and the standard deviation confirm the strength of the policy. Controls analysis are also provided which shows that on average $ABC\_Control$ contributes 98.44% and 96.44% in case $H = 3$; 96.22% and 95.78% in case $H = 6$ for LC and HC respectively.

An insight on the number of visits and stock out quantity between binomial and uniform distribution are also presented. It is shown that uniform distribution has a higher number of visits compared to binomial distribution. *Policy* M^+completely avoided the stock out quantities for all datasets with LC in both distributions for both $H = 3$ and $H = 6$. However, HC cases tabulated stock out quantities not more than 1.7 units.

CHAPTER 6: DYNAMIC STOCHASTIC INVENTORY ROUTING PROBLEM WITH BACKORDERING

This chapter presents the fourth main contribution to the knowledge, where a Dynamic Stochastic Inventory Routing Problem (DSIRP) is embedded with backorder decision instead of stock-out as in Chapter 5. In backorder, the model assumes that the customers are willing to wait until the product arrived, resulting in a win-win situation. In this model, as in the previous model, the demand is stochastic where it follows a probability distribution where a uniform distribution is considered. The problem is known as dynamic and stochastic inventory routing problem with backordering, DSIRPB. This chapter starts with the descriptions of DSIRPB. This is followed by several assumptions made in the model and a new mixed integer linear programming (MILP) formulation specifically for DSIRPB is proposed. An enhanced hybrid rollout algorithm is proposed where the algorithms is modified to suit DSIRPB. As in previous chapter, an Artificial Bee Colony (ABC) algorithm is proposed in generating the controls to enhance the performance of the hybrid algorithm. The MILP proposed is optimally solved to obtain bounds for comparison. The results of the hybrid rollout algorithm are tabulated, and analysis of the number of visits and the delivery quantities and the backorder quantities are carried out. The findings are summarized at the end of this chapter.

6.1 Problem Description and Assumptions

The DSIRPB is similar to the DSIRP considered in the previous chapter, except for instead of considering stock out, a backordering decision is taken into account. The DSIRPB is solved to find a solution strategy that provides the actions that must be performed at the end of each period.

As in Chapter 5, the distribution network consists of a single supplier, 0 and a set of retailers, $M = \{1,2,\dots,N\}$. Single product is distributed using third-party transportation service over planning horizon $T = \{1, \ldots, H\}$. Each retailer $i \in M$ faces demand r_{it} at time t, where the demand is defined based on a discrete uniform probability distribution having mean q_i. The integer initial inventory level at time $t = 0$ for each retailer i, I_{i0} and the maximum inventory level U_i such that $I_{i0} \leq U_i$ are given.

An inventory policy, order-up-to level (OU) is adopted. This policy indicates that whenever retailer i is visited, the quantity delivered must reach its maximum level U_i, it complies $U_i - I_{it}$. A transportation capacity C is bought at each time t whenever there is delivery. A fixed cost, f is paid for partly of fully use of the capacity.

It is assumed that the sufficient deterministic quantity of products, p_t is produced at the supplier site to fulfill the demands throughout the planning horizon. The positive inventory level, I_{it} and backorder level B_{it} are calculated for each retailer i at time t. An inventory cost, h_i is charged for each positive inventory unit, and a backorder cost $\theta_i > h_i$ is penalized for each positive backorder unit.

The objective of the problem is to find a policy such that the sum of expected inventory cost at the supplier, inventory cost and backorder cost at the retailers, and also transportation cost are minimized over the planning horizon.

6.2 Assumptions

There are a few assumptions made in the inventory model. The assumptions are presented below.

a. Each retailer $i \in M$ has initial inventory level, $\bar{I}_{i0}$.
b. Supplier has initial inventory level, $\bar{I}_{00}$.
c. Each retailer $i \in M$ can maintain inventory up to a maximum level U_i .
d. The OU inventory policy applied express that if retailer i is visited at time t, then the quantity delivered, s_{it} must reach the maximum level U_i.

e. The inventory level of the supplier i in period t, I_{0t} is given by the inventory level in the previous period $t-1$ plus the production at period $t-1$ minus the quantity delivery, s_{it}.

$$I_{0t} = I_{0t-1} + p_{t-1} - \sum_{i \in M} s_{it-1}$$

The supplier inventory level is calculated at the beginning of the period. Each positive inventory is charged h_0. Note that I_{0t} cannot be negative.

f. The inventory level for retailer i in period t, I_{it} and the backorder level for retailer i in period t, B_{it} is given by inventory level of retailer i in the previous period $(t-1)$ minus the backorder level of retailer i in the previous period plus with quantity delivered, s_{it} minus the demand r_{it}.

$$I_{it} - B_{it} = I_{it-1} - B_{it-1} + s_{it} - r_{it}$$

Each positive value I_{it} will be charge inventory cost h_i. For each positive backorder, B_{it} a penalty θ_i is charged.

g. The calculation of the inventory/backorder level in (f) is in this sequence of activities, which are delivery, consumption and updating inventory level. The inventory and the backorder level are calculated at the end of the period. This implies that before consumption, the amount of product may exceed the maximum level (Archetti et al., 2014).

h. The amount of backorder is assumed to be pick up by the customer as soon as it is delivered. This means that the amount delivered may exceed the maximum inventory level as the amount carried includes the backorder decision of the previous period. It is, however, does not violate the maximum inventory level at the retailer's site.

i. No initial backorder quantity, $\bar{B}_{i0}$= 0.

j. The demands follow a uniform probability distribution which indicates that the demand may be larger than the delivery quantities and initial inventory. The initial inventory level can be lesser than the retailer's demand at period $t = 1$.

k. The objective of the problem is to find the best policy that minimizes the sum of the expected inventory at the supplier, inventory cost at the retailer, backorder cost at the retailer and the transportation cost.

6.3 Problem Formulation

This section presents the formulation of the problem. A new mixed integer linear programming (MILP) is proposed for the DSIRPB. The MILP is modified from Chapter 5 to take into account the backorder decision where additional six constraints are introduced. Most variables use the same notations as in the previous chapter and the additional backorder variables are introduced. Readers are referred to Archetti et al. (2014) for alternative formulations of IRP. The notations used throughout this chapter are as follows.

Indices

0	Supplier ($i = 0$)
$M = \{1,2, \dots, N\}$	Retailer index
$T = \{1,2, \dots, H\}$	Period index
$T' = \{1,2, \dots, H + 1\}$	Period index
i	Index for supplier $i = 0$ and retailer $i \in M$
t	Index for time

Parameters

U_i	Maximum inventory level for retailer i

r_{it}	Stochastic demand face by retailer i at time t. It is defined based on a discrete uniform distribution, with mean q_i.
C	Transportation Capacity
f	Fixed Cost for each C capacity used.
h_i	Inventory holding Cost
θ_i	Backorder penalty cost

Decision Variables

s_{it}	Quantity delivery of retailer i demand at time period t.
z_{it}	Binary decision variable, $z_{it} = 1$ if retailer i is visited at time t, zero otherwise.
I_{0t}	Inventory level of supplier at time t
I_{it} , α_{it}	Inventory level of each retailer i at time t
B_{it}	Backorder level of each retailer i at time t
γ_{it}	Binary decision variable for inventory level, $\gamma_{it} = 1$ if inventory is positive.
σ_{it}	Binary decision variable for backorder level, $\sigma_{it} = 1$ if backorder is positive.
y_t	Binary decision variable for transportation, $y_t = 1$ if at least one of any retailer i is visited

Initial Value

$\bar{I}_{00}$	Inventory level of supplier at time $t = 0$
$\bar{I}_{i0}$	Inventory level of each retailer i at time $t = 0$
B_{i0}	Backorder level of each retailer i at time $t = 0$

$\bar{B}_{i0}=0$	Initial Backorder level of each retailer i at time $t=0$
$r_{i0}=0$	Initial value for demand $i \in M$
$s_{i0}=0$	Initial value for quantity delivery $i \in M$
$p_0=0$	Initial production at time $t=0$

Formulation of the problem:

$$\min \sum_{t\in T'} h_0 I_{0t} + \sum_{i\in M}\sum_{t\in T'} h_i \alpha_{it} + \sum_{i\in M}\sum_{t\in T'} \theta_i B_{it} + \sum_{t\in T} f y_t \tag{6.1}$$

Subject to:

$$I_{00} = \bar{I}_{00} \tag{6.2}$$

$$I_{0t} = I_{0t-1} + p_{t-1} - \sum_{i\in M} s_{it-1} \qquad t \in T' \tag{6.3}$$

$$\alpha_{i0} = \bar{I}_{i0} \qquad i \in M \tag{6.4}$$

$$B_{i0} = \bar{B}_{i0} \qquad i \in M \tag{6.5}$$

$$\alpha_{it} - B_{it} = \alpha_{it-1} - B_{it-1} + s_{it} - r_{it} \qquad i \in M, t \in T' \tag{6.6}$$

$$\alpha_{it} \le U_i \gamma_{it} \qquad i \in M, t \in T' \tag{6.7}$$

$$B_{it} \le U_i \sigma_{it} \qquad i \in M, t \in T' \tag{6.8}$$

$$\gamma_{it} + \sigma_{it} \le 1 \qquad i \in M, t \in T' \tag{6.9}$$

$$s_{it} \ge U_i z_{it} - \alpha_{it-1} + B_{it-1} \qquad i \in M, t \in T \tag{6.10}$$

$$s_{it} \le U_i - \alpha_{it-1} + B_{it-1} \qquad i \in M, t \in T \tag{6.11}$$

$$s_{it} \le U_i z_{it} + B_{it-1} \qquad i \in M, t \in T \tag{6.12}$$

$$\sum_{i\in M} s_{it} \le C y_t \qquad t \in T \tag{6.13}$$

$$\sum_{i\in M} s_{it} \le I_{0t} \qquad t \in T \tag{6.14}$$

$$I_{0t} \ge 0 \qquad t \in T' \tag{6.15}$$

$\alpha_{it} \geq 0 \quad i \in M, t \in T' \quad (6.16)$

$B_{it} \geq 0 \quad i \in M, t \in T' \quad (6.17)$

$s_{it} \geq 0\ , integer \quad i \in M, t \in T \quad (6.18)$

$\gamma_{it} \in \{0,1\}, binary \quad i \in M, t \in T' \quad (6.19)$

$\sigma_{it} \in \{0,1\}, binary \quad i \in M, t \in T' \quad (6.20)$

$z_{it} \in \{0,1\}, binary \quad i \in M, t \in T \quad (6.21)$

$y_t \in \{0,1\}, binary \quad t \in T \quad (6.22)$

The objective function (6.1) comprises of inventory cost at supplier site, inventory cost at retailers, backorder penalty cost at retailers and transportation cost. (6.2) is the initial inventory value for the supplier at time $t = 0$. The inventory level at the supplier at time t is given by constraint (6.3). Constraints (6.4) and (6.5) are the initial inventory level and the initial backorder level at retailers at time $t = 0$. The inventory balance equation (6.6) is adopted from Abdelmaguid et al. (2009) where the inventory level or the backorder level at time t is equal to the inventory at previous period minus the backorder level at previous period plus with quantity delivery at time t minus the demand at time t. Constraints (6.7) and (6.8) are to ensure that each positive inventory level α_{it} for each retailer $i \in M$ is not more than the maximum level U_i or, the backorder level B_{it} permitted is not more than the maximum level U_i. Constraint (6.9) is the binary decision to ensure that if positive inventory level exists, there is no backorder level, and vice versa. Constraints (6.10) until (6.12) serve as the Order Up to level (OU) inventory policy, which ensures that if retailer i is visited at time t then quantity delivery must reach the maximum level U_i including the backorder quantity. Constraint (6.13) and (6.14) describe that the summation of the quantity delivery to all retailers at time t must not exceeds the transportation capacity and the products are readily available at the respective supplier. The non-negativity constraints are given in (6.15) until (6.18). The decision variables are given by constraints (6.19) until (6.22).

This MILP formulation, designated problem *DetB* is incorporated in the solution methodology scheme and it is optimally solved for demand set as an average value, $r_{it} = q_i$ to generate the control. As in the Chapter 5, this formulation is used to compute the approximate cost-to-go (future approximations) and consequently use to obtain the bounds. This work is an extension of DSRIP and DSRIP is proven to be NP-hard (Bertazzi et al., 2015) thus the DSIRPB considered is also NP-hard.

6.4 Solution Methodology

This section presents the method for solving the DSIRPB. As the demand is known in a probabilistic sense, the problem is formulated as dynamic programming where the dynamic system can take into account the stochastic elements in the decision process. A hybrid rollout algorithm is proposed to solve the problem. The dynamic programming formulation and the hybrid rollout algorithm are presented in subsection 6.4.1 and 6.4.2 respectively.

6.4.1 Stochastic Dynamic Programming

This subsection presents the stochastic dynamic programming (SDP) formulation for the DSIRPB. As discussed in Chapter 5, the DP has 5 essential components. The components are the state of the system, the controls constraint, discrete-time dynamic system, the immediate cost and the optimization problem. The details of each component are given below.

6.4.1.1 States

The state is defined as x_t, where x_t represent both of the inventory level and the backorder level at supplier and retailer $i \in M$. The state of the system at time $t = 0,1, \dots, H+1$ is defined as $x_t = (x_{0t}, x_{1t}, x_{2t}, \dots, x_{nt})$. The inventory level at retailer i is represented by a positive x_{it} whilst the backorder level at retailer i is represented by a negative x_{it}, the additional factor in DSIRPB. The inventory level at supplier, x_{0t}, is always positive for all t, as this ensures that items are always available to fulfil the retailers demand.

The initial value $x_0 = (\bar{I}_{00},\ \bar{I}_{10}, \dots, \bar{I}_{n0})$ is given. The state of the supplier at time $t \in T$, x_{0t} is an integer number that obeys the constraint $\bar{I}_{00} + \sum_{k=1}^{t-1} p_k - (t-1)C \le x_{0t} \le \bar{I}_{00} + \sum_{k=1}^{t-1} p_k$ and at time $t = H + 1$ is $\bar{I}_{00} + \sum_{k=1}^{t-1} p_k - HC \le x_{0H+1} \le \bar{I}_{00} + \sum_{k=1}^{t-1} p_k$. The state of the retailer i at time $t \in T'$, x_{it} is an integer number that obeys the constraint $-U_i \le x_{it} \le U_i$.

6.4.1.2 Controls

Control is the set of decisions available at each state x_t at time $t \in T$. The control $u_t(x_t)$ is defined as binary variables z_{it}, $i \in M$ that is :

$$u_t(x_t) = (z_{1t}, z_{2t}, \dots, z_{nt})$$

Once the binary variables z_{it} is revealed, the corresponding quantity deliveries are also determined as an OU policy is adopted. The feasibility of the control is ensured by two constraints; the transportation constraint and the availability of the product at supplier site. The constraints are (6.23) and (6.24) given below.

$$\sum_{i \in M} (U_i - x_{it}) z_{it} \le C \qquad (6.23)$$

$$\sum_{i \in M} (U_i - x_{it}) z_{it} \le x_{0t} \qquad (6.24)$$

6.4.1.3 Dynamic System

A dynamic system is a function that describes the progress of the system from one state to another state. The discrete dynamic system defines the inventory/backorder level of the system at time $t + 1$ using the information of the previous inventory/backorder level and the decision obtained from the control on which retailer to be visited.

Given the state of the system at time t, x_t and the control enforces at time t, $u_t(x_t)$ together with r_t is the vector of the demand r_{it} is the demand at time t, $i \in M$, the state of the system at time $t + 1$ is given by:

$$x_{t+1} = (\hat{x}_{0t}, \hat{x}_{1t}, \hat{x}_{2t}, \dots, \hat{x}_{nt})$$

where inventory level of the supplier is defined as $\hat{x}_{0t} = x_{0t} + p_t - \sum_{i \in M}(U_i - x_{it})z_{it}$ and $\hat{x}_{it} = x_{it} + (U_i - x_{it})z_{it} - r_{it}$ is the inventory/backorder level at a retailer. Note that the state of the system is allowed to take non-positive values since the backorders are considered.

6.4.1.4 Costs

The cost of being in state x_t, and enforced control u_t and having demand r_t is calculated as the sum of the inventory, backorder penalty cost and transportation cost. The cost is given as:

$$g(x_t, u_t, r_t) = \sum_{i \in M} h_i \ max\{0, \hat{x}_{it}\} + \sum_{i \in M} \theta_i \ max\{0, -\hat{x}_{it}\} + f\emptyset\left(\sum_{i \in M} z_{it}\right)$$

where h_i is the inventory cost, θ_i is the penalty for backordering, f fixed transportation cost, and $\emptyset(w) = 1$ if $w > 0$ and 0 otherwise.

6.4.1.5 Optimization Problem

The optimization problem is to find a policy that minimizes the expected total cost. Consider the set Π of feasible policies consists of π where π is a sequence of functions with $\pi = \{\mu_1, \mu_2, \dots, \mu_H\}$. μ_t maps each state x_t into a control $u_t = \mu_t(x_t)$, such that $\mu_t(x_t) \in \mathcal{U}_t(x_t)$.

Starting from the given initial state x_0, the total expected cost of π is :

$$J_{\pi}(x_0) = E\left\{\sum_{t=1}^{H} g(x_t, \mu_t(x_t), r_t) + \, g_{H+1}(x_{H+1})\right\}$$

The best policy is found by choosing the best π from family Π, $\pi \in \Pi$ that minimizes the total expected cost over the horizon. Thus the aim is to find a policy π^*, that minimizes the total expected cost such that:

$$J_{\pi^*}(x_0) = min_{\pi \in \Pi} \, J_{\pi}(x_0)$$

6.4.2 Enhanced Hybrid Rollout Algorithm

A hybrid rollout algorithm is proposed to solve the DP formulation presented in subsection 6.4.1. The algorithm is proposed instead of an exact DP method because the problem suffers curses of dimensionality. This is due to the realistic problem size. The explanation of the three curses of dimensionality; the state space, the outcome space, and the control space were given in the previous chapter subsection 5.3.2.

The procedure of hybrid rollout algorithm in this chapter is similar to the previous chapter with several modifications to suit DSIRPB due the backorder factor. A detailed discussion on the components of the algorithm such as number of scenarios, approximate cost-to-go and approximate Q factor. However, a brief explanation of the components is given to ease the reader. The details of the approximate set of controls, specific for DSIRPB are also presented.

Rollout algorithm, as previously discussed, is the one-step lookahead policy, where future trajectories are evaluated to estimate the cost of being in one state. The explanation of how the rollout algorithm works is given in the previous chapter, subsection 5.3.2.1. The rollout algorithm proposed is hybrid where a combination of a mathematical programming problem *DetB* and heuristic or metaheuristic is integrated inside the scheme to determine future trajectories.

The policy given by the algorithm is known as *Policy B*. The key idea to the cost approximation derived is to capture cost at the current state and to estimate the future costs. The approximate rollout control, $\tilde{\mu}_t$ at time $t = 1,2, \dots, H$ corresponding to state x_t is define by equation (6.25).

$$\tilde{\mu}_t(x_t) = \arg\ \min_{u_t(x_t) \in \tilde{\mathcal{U}}_t(x_t)} \tilde{Q}_t\big(x_t, u_t(x_t)\big) \tag{6.25}$$

where the approximate Q factor $\tilde{Q}_t\big(x_t, u_t(x_t)\big)$ is given by

$$\tilde{Q}_t\big(x_t, u_t(x_t)\big) = E\{g(x_t, u_t, r_t) + \tilde{J}_{t+1}(x_{t+1})\}$$

the expected sum of the immediate cost, $g(x_t, u_t, r_t)$ and the approximate cost-to-go, $\tilde{J}_{t+1}(x_{t+1})$. The descriptions on $\tilde{J}_{t+1}(x_{t+1})$, $\tilde{\mathcal{U}}_t(x_t)$ are further elaborated below after the definition of the set of scenarios, $S_t(x_t, u_t)$.

6.4.2.1 Scenarios, $S_t(x_t, u_t)$

The scenario is a set of random outcomes where $s \in S_t(x_t, u_t)$ is referred to the realization of demand, r_t for each retailer in time t. The demand, r_{it} for each retailer $i \in M$ consists of 4 different demand settings, that are:

- Randomly generated a number of demands that follow the uniform probability distribution.
- Demand is set to 0 to capture the minimum value.
- Demand is set to U_i to capture the maximum value.
- Demand is set to q_i to capture the most likely demand to happen.

6.4.2.2 The Approximate Cost-to-go, $\tilde{J}_{t+1}(x_{t+1})$

The approximate cost-to-go, $\tilde{J}_{t+1}(x_{t+1})$ is obtained by solving problem *DetB* in section 6.3. It uses the inventory level obtained, $\bar{x}_{it+1}$ (for each retailer $i \in M$) as initial value. Problem *DetB* is solved for time $t+1$ to H with three different demand settings at time $t+1$. The three different demand values at time $t+1$ are:

- $r_{it+1} = 0$
- $r_{it+1} = U_i$
- $r_{it+1} = q_i$

The demand at time $t+2$ until H are set to be equal to the average value, $r_{it+2}, \dots, r_{iH} = q_i$. Thus, the $\tilde{J}_{t+1}(x_{t+1})$ is computed as the average of three optimal costs found.

6.4.2.3 The Approximate Set of Controls, $\tilde{\mathcal{U}}_t(x_t)$

The approximate set of controls, $\tilde{\mathcal{U}}_t(x_t)$ associated with each state x_t at time t. The details of generated controls are:

i. No retailer is served.
ii. Exactly solve problem *DetB* where the demand is set to be equal to the average value (q_i) for time t to H.
iii. All retailers are served.
iv. Controls are generated by Artificial Bee Colony (ABC) algorithm by finding which retailer(s) to serve and minimizes the cost. The output of the ABC is a population of feasible controls.

The control (i), (ii) and (iv) are the same as in the previous chapter, except for (iii). The control (iii) where all retailers are served ($z_{it} = 1$ for all $i \in M$) is added and the control is feasible. A simple experiment is carried out that shows that the control is selected a few times especially in period 1 as initial inventory is not zero (in the datasets considered). It is also introduced to adjust to the size of the demands (uniform probability distribution in general produces large size of demand) to increase number of visits in order to reduce the amount of backorders.

6.4.3 ABC Algorithm as Controls

This subsection presents the details on how the ABC algorithms generate the number of controls. Bees phases perturb the solution to get better control. In this algorithm, the inventory level, $\bar{I}_{it}$ and the realization of demand r_{it} for $i \in M$ in period t, are used as the input. This algorithm is similar to the *ABC_Control* in previous chapter, except that this algorithm considers backorder penalty cost. The details steps of the algorithm describe the bee phases are given below.

STEP 1 Initialization Phase

1.1 Generate randomly n number of solutions (food sources). Each solution is represented by binary numbers, z_{it} with number of retailers, $i \in M$; where 1 indicates to visit and 0 otherwise.

1.2 Denote each solution as $a_j, j = 1, \dots, n$ where $a_j = (z_{1t}, z_{2t}, \dots, z_{Mt})$. The notation t is dropped as only 1 period is considered at a time. Evaluate, $a_j = (z_1, z_2, \dots, z_M)$. Evaluate the fitness value for each food source (using cost function g in subsection 6.4.1.4). Assign each employed bee to a food source.

STEP 2 Set $iteration = 0$ and $l_1 = l_2 = \cdots = l_n = 0$. Declare the value of $LIMIT$ and the maximum number of iterations, $MAXITER$.

STEP 3 Repeat the following until the stopping condition, $MAXITER$ is met.

3.1 Employed Bee Phase

a. For each food source, a_j :

i. Find all the visiting retailers ($z_i = 1$), and then calculate the backorder penalty cost multiply the demand ($\theta_i \times r_{it}$). Denote retailer i that gives the minimum value calculated as sc_1.

ii. Find all the non-visiting retailers ($z_i = 0$), and then calculate $e_i = \theta_i(\bar{I}_{it} - r_{it})$. Denote retailer i that gives the minimum e_i value as sc_2. Note that e_i can be negative, which show that demand in retailer i is not fulfill.

iii. Do swap between z_{sc1} and z_{sc2} of retailer sc_1 and sc_2. Assign the new solution found as $\tilde{a}_j$. Evaluate the fitness value for the new $\tilde{a}_j$.

b. *If* $g(\tilde{a}_j) < g(a_j)$; replace the old food source with a new food source, $a_j \leftarrow \tilde{a}_j$ and set $l_j = 0$. *Else* set $l_j = l_j + 1$.

3.2 Onlooker Bee Phase

a. Set $G_j = \emptyset, j = 1, \ldots, n$, where G_j is the set of neighbor solutions of food source j.

b. For each onlooker bees.

i. Select a food source, a_j, using the tournament selection method (Goldberg & Deb, 1991).

ii. Apply a neighborhood operator, insertion method on selected a_j. Find all the non-visiting retailers ($z_i = 0$), and then calculate $e_i = \theta_i(\bar{I}_{it} - r_{it})$. Denote retailer i that gives the minimum e_i

value as ic. Change the z_{ic} to 1. Denote the new found solution as $\tilde{a}_j$.

iii. $G_j = G_j \cup \tilde{a}_j$.

c. For each food source a_j and $G_j \neq 0$.

i. Set $\hat{a}_j \in argmin_{\tilde{a} \epsilon G_j} g(\tilde{a})$.

ii. *If* $f(\hat{a}_j) < f(a_j)$; replace the old food source with the new one; $a_j \leftarrow \hat{a}_j$ and set $l_j = 0$. *Else* set $l_j = l_j + 1$.

3.3 Scout Bee Phase

For each food source, a_j. *If* $l_j = LIMIT$, replace a_j with a randomly generated solution.

$iteration = iteration + 1$.

STEP 4 Output is the best food source found so far.

Figure 6.1 illustrates the steps of *Policy B* in solving DSIRPB which is similar to Figure 5.2 in Chapter 5 except some modification to suit DSIRPB as explained in the algorithm.

6.5 Computational Results and Discussions

The hybrid rollout algorithm with *Policy B* developed is coded and run using MATLAB 8.1. The MILP is solved using CPLEX 12.6 connector inside MATLAB platform. All the calculations are computed on a 16GB RAM computer with a 3.1GHz processor. The probability distribution is computed from a Statistics and Machine Learning Toolbox.

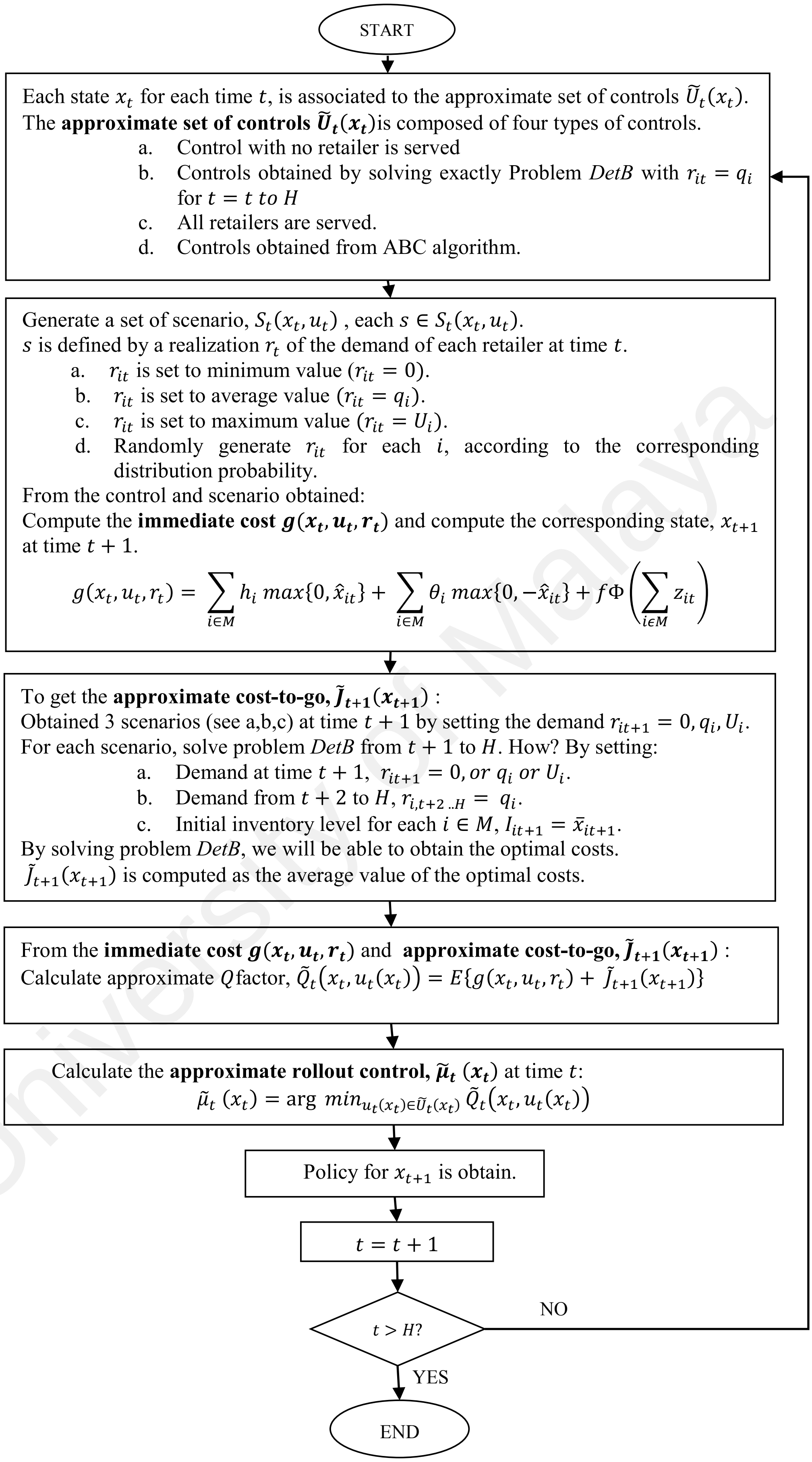


Figure 6.1: Flow chart for *Policy B*.

6.5.1 Dataset

The performance of *Policy B* is tested on 60 datasets, similar as in Chapter 5. However, only demands that follow uniform probability distribution is considered. This is to effectively see the backorder decision as large demand value has equal chances to happen. The new parameter for backorder decision is given as in Table 6.1 to substitute the stock out parameter. Note that the penalty cost varies for both low inventory cost (LC) and high inventory cost (HC).

Table 6.1: Criterion and data setting for the datasets.

Criteria	**Data Setting**
Backorder cost at the retailers, θ_i	$0.125\ (f/2 + h_i U_i)$

The ABC algorithm follow parameters setting in Chapter 5.

6.5.2 Results and Discussions

This subsection discusses the results for *Policy B*. The expected costs and the bounds are presented and the ratios are given. The controls that contributed to the best expected cost found, the number of visits, the delivery quantities and the backorder quantities are examined and analyzed.

6.5.2.1 Bounds and Expected Cost

Performance of *Policy B* is assessed by comparing with the bounds obtained by optimally solved problem *DetB* for 10 different trajectories of demand that follow the uniform probability distribution and with optimality gap of $\leq 0.05\%$. The results *Policy B* are the average expected costs found for 10 different runs. Table 6.2 and 6.3 present the bounds, results of *Policy B* and the ratios for periods, $H = 3$ and $H = 6$ respectively. Small ratios show the significance and the closeness of *Policy B* and the bounds. The average performance of the *Policy B* is calculated; 0.269 and 0.451 for low inventory cost (LC) and high inventory cost (HC) with 3 periods ($H = 3$) respectively. While for LC and HC with $H = 6$, the averages are 0.362 and 0.438 respectively.

Table 6.2: Bounds, Average Expected Cost and Ratio of $H = 3$.

	LC			HC		
Dataset	**Bound**	**Policy B**	**Ratio**	**Bound**	**Policy B**	**Ratio**
abs1n5	614.569	82.324	0.134	1018.859	550.928	0.541
abs2n5	505.412	77.811	0.154	1013.218	490.519	0.484
abs3n5	540.874	122.174	0.226	1540.402	805.552	0.523
abs4n5	380.759	70.137	0.184	810.869	416.622	0.514
abs5n5	210.9618	105.683	0.501	1374.445	750.869	0.546
abs1n10	568.3672	208.148	0.366	3811.559	1811.286	0.475
abs2n10	688.4269	194.601	0.283	3244.956	1530.975	0.472
abs3n10	530.1454	166.317	0.314	2876.574	1353.239	0.470
abs4n10	564.2262	173.012	0.307	3136.442	1424.784	0.454
abs5n10	549.9661	223.67	0.407	4107.767	1858.76	0.452
abs1n20	1365.9486	340.553	0.249	7820.408	3042.075	0.389
abs2n20	1539.949	347.525	0.226	8162.15	2973.81	0.364
abs3n20	1628.5811	342.922	0.211	8258.941	3106.735	0.376
abs4n20	1346.411	300.505	0.223	7360.714	2517.769	0.342
abs5n20	1509.106	382.881	0.254	9361.631	3351.069	0.358
Average			0.269			0.451

Table 6.3: Bounds, Average Expected Cost and Ratio of $H = 6$.

	LC			HC		
Dataset	**Bound**	**Policy B**	**Ratio**	**Bound**	**Policy B**	**Ratio**
abs1n5	368.022	201.411	0.547	3166.755	1487.17	0.470
abs2n5	386.226	190.503	0.493	2447.998	1332.84	0.544
abs3n5	374	183.974	0.492	2555.746	1310.669	0.513
abs4n5	325.731	165.712	0.509	2525.938	1177.779	0.466
abs5n5	321.852	181.021	0.562	2606.362	1272.566	0.488
abs1n10	1013.975	306.01	0.302	6885.966	2584.869	0.375
abs2n10	726.494	247.815	0.341	4337.775	2007.242	0.463
abs3n10	783.787	278.013	0.355	4832.745	2253.434	0.466
abs4n10	743.131	276.374	0.372	5061.399	2128.183	0.420
abs5n10	1276.563	355.134	0.278	7517.855	3005.08	0.400
abs1n20	2182.178	582.258	0.267	13496.279	5253.827	0.389
abs2n20	2504.893	604.66	0.241	14255.48	5750.754	0.403
abs3n20	2076.95	539.853	0.260	11599.98	5011.495	0.432
abs4n20	2386.462	497.515	0.208	12343.454	4436.544	0.359
abs5n20	3006.698	627.447	0.209	14594.671	5648.394	0.387
Average			0.362			0.438

Table 6.4 tabulates the standard deviation of the expected cost for all datasets. It is observed that the standard deviations are small for the LC cases, especially for 5 retailers'

datasets. As the complexity of the datasets increases (HC cases), the standard deviations increases, especially for datasets with 20 retailers and 6 periods.

Table 6.4: Standard Deviations of the Expected Cost for LC and HC cases with $H = 3$ and $H = 6$.

	LC		**HC**	
Dataset	$\boldsymbol{H = 3}$	$\boldsymbol{H = 6}$	$\boldsymbol{H = 3}$	$\boldsymbol{H = 6}$
abs1n5	2.785	6.699	14.953	60.778
abs2n5	4.268	12.661	22.194	31.155
abs3n5	12.695	8.620	66.546	67.491
abs4n5	2.562	3.337	12.306	35.821
abs5n5	3.634	11.420	42.280	99.128
abs1n10	5.739	14.394	87.177	107.784
abs2n10	15.138	5.831	43.070	139.271
abs3n10	2.273	8.801	48.230	140.097
abs4n10	6.396	17.074	44.591	101.802
abs5n10	14.549	10.921	24.672	251.792
abs1n20	16.534	29.365	208.505	351.387
abs2n20	16.811	32.277	272.108	494.097
abs3n20	6.383	25.967	158.951	477.521
abs4n20	12.088	21.260	171.800	217.674
abs5n20	6.592	38.504	207.896	328.507

6.5.2.2 Analysis of Controls

An analysis of the approximate set of controls that contribute to the small ratios are examined. Controls, discussed in subsection 6.4.2 comprise of $CTRL$ and $CTRLabc$, where controls (i), (ii) and (iii) are labelled as $CTRL$ while control (iv) is referred to $CTRLabc$. Table 6.5 and 6.6 tabulate the percentage of the controls chosen by the policy. The percentage is calculated for all 10 individual runs for each dataset. For example, in the case of $H = 3$, the total number of $CTRL$ selected is divided by 30 (3×10 runs) multiplies with 100.

$CTRLabc$ contributes to the best expected cost in the case of $H = 3$ for LC not less than 40% (abs5n20) and on average is more than half which is 57.1111%. While for HC, it is not less than 63.3333% and on average is 66.4444%. Higher percentage of $CTRLabc$ is observed in the case of $H = 6$ than $H = 3$; where on average $CTRLabc$ are selected 82.8889% and 80.6667% for LC and HC respectively. However, dataset with 20 retailers have a lower percentage than the datasets with 5 retailers. As the number of retailers increase, the $CTRLabc$ contribute lesser. *Policy B* tends to choose $CTRL$ for the first and last period.

It is observed that for $CTRL$ with control (iii) where all retailers are served is chosen most of times especially in period 1. This is because that the initial backorders are zero, so having to visit all the retailers is the best strategy so as to avoid accumulation of backorders, given that the future trajectories cost is calculated. Further, the amounts carried do not violate the transportation capacity and the products (inventory) are available at the production site.

Table 6.5: The percentage of controls chosen for case $H = 3$.

	LC		**HC**	
Dataset	$\boldsymbol{CTRLabc}$	$\boldsymbol{CTRL}$	$\boldsymbol{CTRLabc}$	$\boldsymbol{CTRL}$
abs1n5	60.0000	40.0000	66.6667	33.3333
abs2n5	66.6667	33.3333	66.6667	33.3333
abs3n5	66.6667	33.3333	66.6667	33.3333
abs4n5	66.6667	33.3333	66.6667	33.3333
abs5n5	66.6667	33.3333	66.6667	33.3333
abs1n10	56.6667	43.3333	66.6667	33.3333
abs2n10	60.0000	40.0000	66.6667	33.3333
abs3n10	50.0000	50.0000	66.6667	33.3333
abs4n10	63.3333	36.6667	66.6667	33.3333
abs5n10	56.6667	43.3333	66.6667	33.3333
abs1n20	50.0000	50.0000	66.6667	33.3333
abs2n20	50.0000	50.0000	63.3333	36.6667
abs3n20	53.3333	46.6667	66.6667	33.3333
abs4n20	50.0000	50.0000	66.6667	33.3333
abs5n20	40.0000	60.0000	66.6667	33.3333
Average	57.1111	42.8889	66.4444	33.5556

Table 6.6: The percentage of controls chosen for case $H = 6$.

	LC		HC	
Dataset	***CTRLabc***	***CTRL***	***CTRLabc***	***CTRL***
abs1n5	83.3333	16.6667	83.3333	16.6667
abs2n5	83.3333	16.6667	83.3333	16.6667
abs3n5	81.6667	18.3333	83.3333	16.6667
abs4n5	83.3333	16.6667	83.3333	16.6667
abs5n5	83.3333	16.6667	83.3333	16.6667
abs1n10	83.3333	16.6667	83.3333	16.6667
abs2n10	83.3333	16.6667	81.6667	18.3333
abs3n10	83.3333	16.6667	81.6667	18.3333
abs4n10	83.3333	16.6667	81.6667	18.3333
abs5n10	83.3333	16.6667	83.3333	16.6667
abs1n20	80.0000	20.0000	75.0000	25.0000
abs2n20	81.6667	18.3333	76.6667	23.3333
abs3n20	83.3333	16.6667	78.3333	21.6667
abs4n20	83.3333	16.6667	75.0000	25.0000
abs5n20	83.3333	16.6667	76.6667	23.3333
Average	82.8889	17.1111	80.6667	19.3333

6.5.2.3 Analysis of Number of Visits and Delivery Quantity

Analyses of the average delivery quantity and the average number of visits are carried out to compare the effect on different inventory cost (LC and HC), as well as the effect on the inventory policy: OU. It is anticipated that the delivery quantity in LC cases is higher as it is much more beneficial to keep inventories because of lower inventory cost. However, the analysis in Figure 6.2 shows that the quantity delivered in HC cases is higher than LC when $H = 3$ except for abs5n5 and abs2n10. This may be due to the shorter periods $H = 3$.

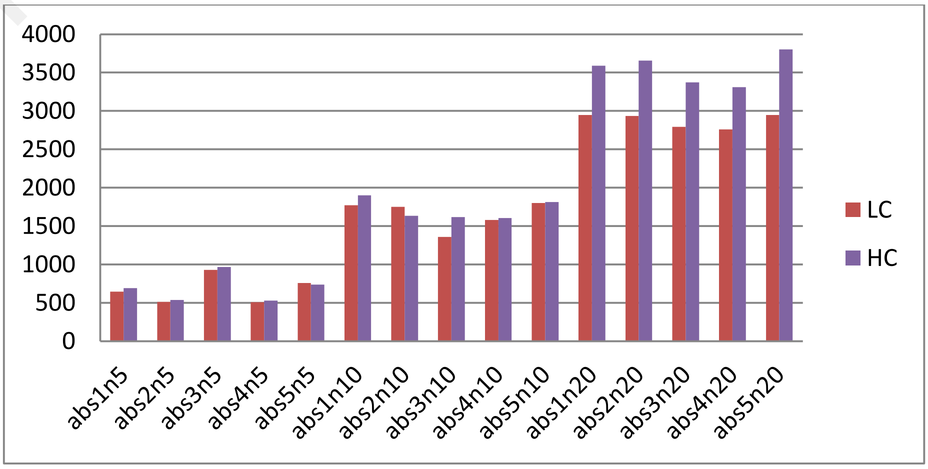


Figure 6.2: Average Delivery Quantity for case $H = 3$.

While in the case $H = 6$, datasets with 5 and 10 number of retailers have almost equals the average quantity delivered in both cases, LC and HC as shown in Figure 6.3. The significant difference where HC is lower than LC is observed in all the 20 retailers datasets.

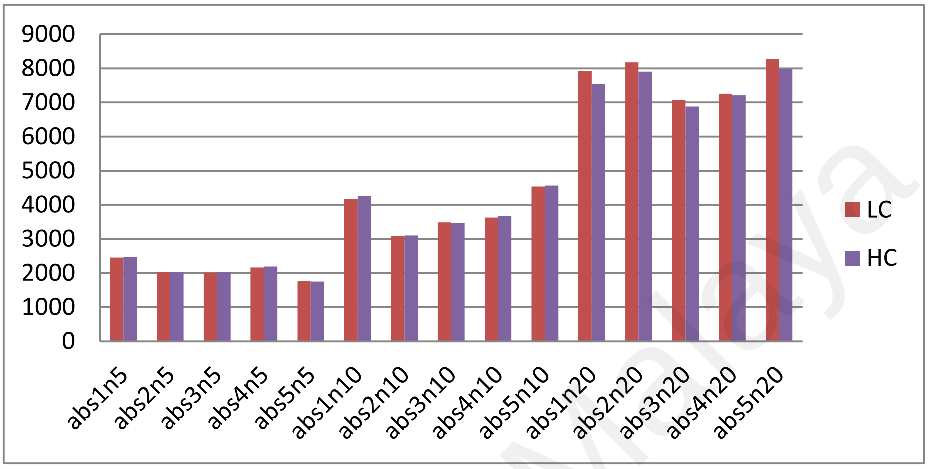


Figure 6.3: Average Delivery Quantity for case $H = 6$.

The analyses of the average number of visits is illustrated in Figure 6.4 and 6.5 for $H = 3$ and $H = 6$ and for the two inventory levels LC and HC, respectively. Results in $H = 3$ show that HC has a higher average number of visits compared to LC for all of the datasets except in 3 datasets (abs4n5, abs5n5 and abs4n10) where the number of visits are almost equal. It is noted that HC has larger quantity delivered and is justified by the larger number of visits and it also due the OU policy adopted. In contrast the results of $H = 6$ shows that the average number of visits for datasets with LC are higher than HC except for abs2n5 and abs4n5. This is coherent with the analysis done in Figure 6.3 where the quantity delivered for LC is higher than HC.

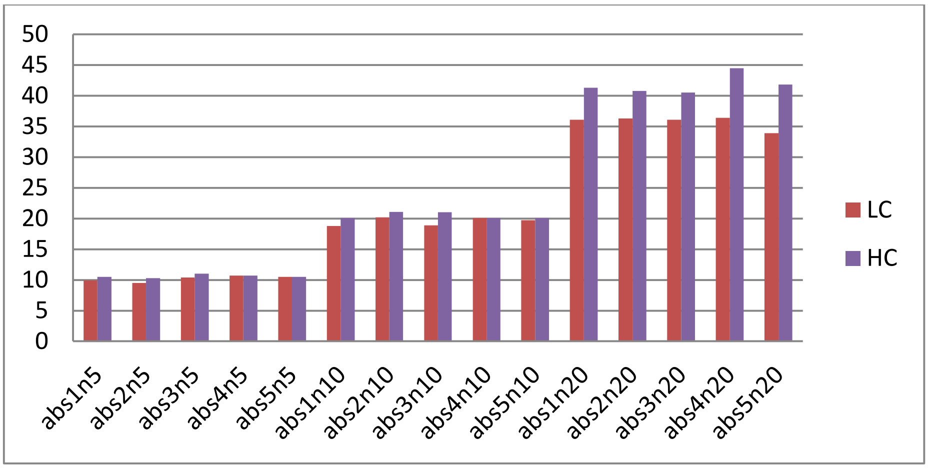


Figure 6.4: Average Number of Visits for case $H = 3$.

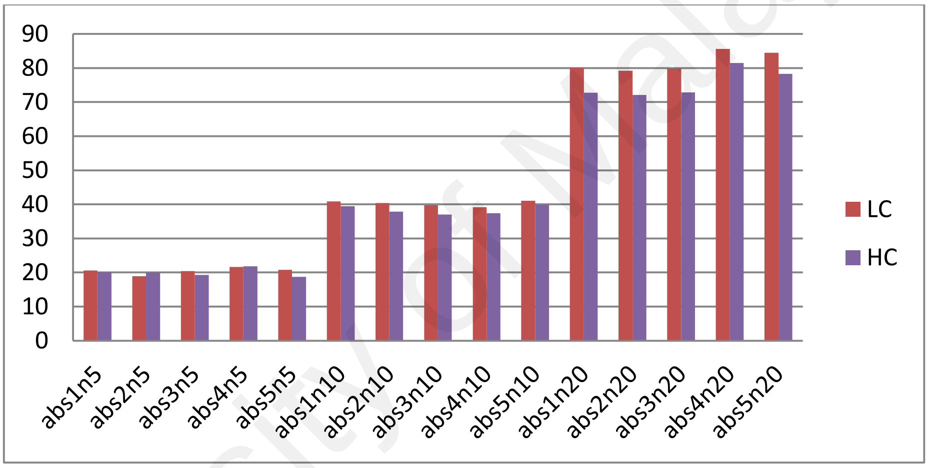


Figure 6.5: Average Number of Visits for case $H = 6$.

The average computing time (in seconds) to obtain the results of *Policy B* are given in Table 6.7. Datasets with 5 retailers and $H = 3$ in LC cases run with minimum 338.023 seconds (around 5.6 minutes) in abs3n5. As the number of retailers and the number of period increase, *Policy B* took a longer time to find the optimal policy. The maximum time observed is in abs4n20-LC, where the expected costs are found on average 1760.323 seconds (29.339 minutes).

Table 6.7: Time taken for LC and HC cases with $H = 3$ and $H = 6$.

	$H = 3$		$H = 6$	
Dataset	**LC**	**HC**	**LC**	**HC**
abs1n5	347.19	358.535	838.04	850.919
abs2n5	340.515	353.144	877.531	834.864
abs3n5	338.023	392.497	816.023	850.765
abs4n5	368.931	393.833	883.718	933.802
abs5n5	353.185	348.738	802.702	809.65
abs1n10	415.703	449.657	1040.098	1046.179
abs2n10	410.985	450.322	1047.86	1012.71
abs3n10	404.573	440.634	1102.722	994.318
abs4n10	421.4	466.888	1126.063	1000.844
abs5n10	416.915	449.904	1133.39	1063.936
abs1n20	515.852	555.672	1658.788	1450.37
abs2n20	501.528	531.838	1650.398	1337.016
abs3n20	510.42	549.486	1649.542	1270.846
abs4n20	546.48	605.591	1760.323	1471.198
abs5n20	508.665	557.352	1653.388	1402.074

6.5.2.4 Analysis of Backorder Decisions

The backorder decisions are the essential part of DSIRPB. It is interesting to examine the backorder quantity on various parameter that influences the expected cost. Figure 6.6 provides the details on the average number of visits and the backorder quantities for the case of $H = 3$ and for both LC and HC. It is observed that all datasets have backorder quantities, except for abs3n5-HC.

The number of visits in LC and HC as indicated in Figure 6.6 does not vary significantly for 5 and 10 number of retailers and the difference is very significant for the 20 retailers problems. Similar pattern is observed for the backorder quantities for both LC and HC except for datasets abs3n5 and abs2n10 where the backorder quantities are higher in LC in comparison to HC. In contrast the large datasets the difference in backorder quantities are very significant between LC and HC. In HC case the number of visits to the retailers is high for the large datasets. However, the backorder quantities are high as well. Similar patterns are found in $H = 6$ cases (see Figure 6.7), where large datasets showed that the average backorder quantities between LC and HC are significant.

Furthermore, it is observed that in large datasets for HC with the exception of abs4n20 there are several runs that give quite a high backorder quantity (the difference in hundreds). This is illustrated by the standard deviation of the backorder quantities for the 10 runs tabulated in Table 6.8. However, a more consistent backorder quantity for dataset abs4n20 is observed. Standard deviations of the number of visits are also given in the same table.

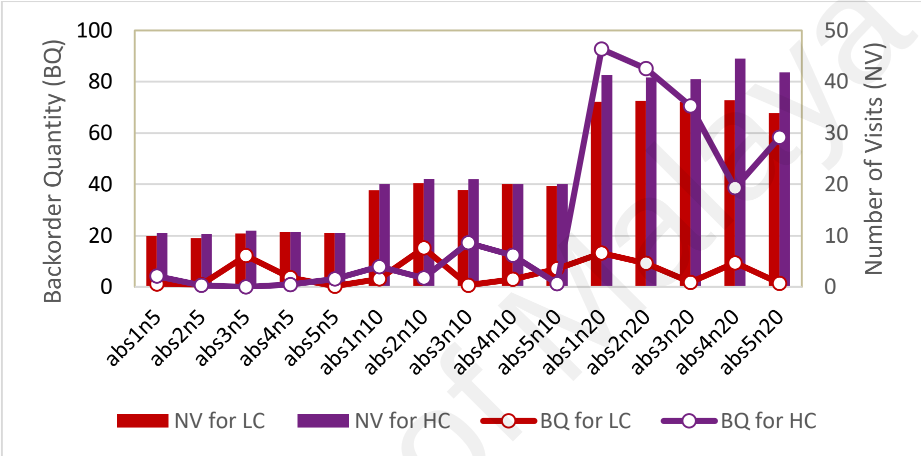


Figure 6.6: Comparison of LC and HC in Average Backorder Quantity and Average Number of Visits for case $H = 3$.

A thorough examination (the break down) of the results, indicates that the number of visits is optimal where the capacity of the vehicles are fulfilled to its maximum. It is also observed that most of the demands are large (this is a possibility using uniform distribution) and split delivery is not allowed. Further *Policy B* emphasis on retailers with a higher backorder penalty cost and it is observed that retailers with small backorder penalty cost are not visited in not more than 2 consecutive periods in both LC and HC.

The variation in backorder quantities especially for 10 and 20 number of retailers is quite significant for $H = 6$ as presented in Table 6.8. It is observed that the backorders for LC and HC occur at most 2 consecutive periods throughout the 10 runs. However, the backorders occasionally occur for 5 periods for HC with 10 and 20 number of retailers. Detail results of each of the 10 runs are given in Appendix D.

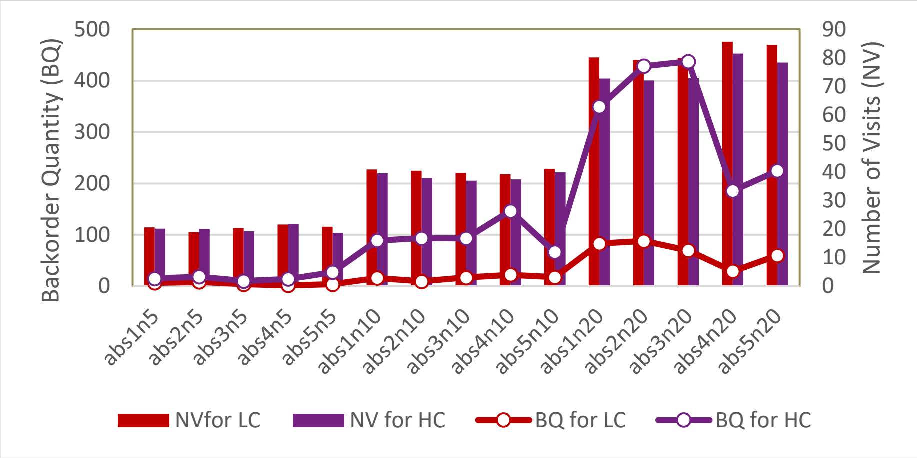


Figure 6.7: Comparison of LC and HC in Average Backorder Quantity and Average Number of Visits for case $H = 6$.

Table 6.8: Standard Deviations of the Number of Visits and the Backorder Quantities for LC and HC cases with $H = 3$ and $H = 6$.

	Number of Visits				**Backorder Quantity**			
	LC		**HC**		**LC**		**HC**	
Dataset	$\boldsymbol{H = 3}$	$\boldsymbol{H = 6}$	$\boldsymbol{H = 3}$	$\boldsymbol{H = 6}$	$\boldsymbol{H = 3}$	$\boldsymbol{H = 6}$	$\boldsymbol{H = 3}$	$\boldsymbol{H = 6}$
abs1n5	1.792	1.174	0.707	1.663	2.150	5.603	4.264	10.570
abs2n5	1.269	0.994	0.823	1.633	1.663	8.149	1.350	17.968
abs3n5	1.174	1.776	1.054	1.636	9.908	4.502	0.000	9.369
abs4n5	0.823	0.699	1.252	0.632	2.119	2.413	1.287	10.703
abs5n5	1.080	1.687	1.080	1.947	0.675	6.056	7.125	21.445
abs1n10	2.300	1.663	1.912	1.780	9.803	14.072	15.975	43.591
abs2n10	2.348	2.066	2.132	2.885	20.482	6.979	8.708	52.985
abs3n10	3.604	1.889	1.054	3.018	2.214	11.063	22.817	42.637
abs4n10	1.792	2.658	2.283	3.062	4.055	18.726	22.505	71.553
abs5n10	2.452	2.514	1.287	2.378	14.317	11.017	2.584	59.434
abs1n20	4.606	6.244	3.917	5.095	33.179	38.581	118.335	147.071
abs2n20	5.498	3.795	5.073	5.953	22.695	43.351	94.207	179.855
abs3n20	4.581	2.961	3.598	5.820	3.327	36.189	110.720	228.324
abs4n20	5.103	3.718	2.991	5.191	18.452	28.484	32.087	102.399
abs5n20	4.332	2.877	3.824	4.448	4.427	50.851	92.637	128.381

6.6 Summary

This chapter describes the implementation hybrid rollout algorithm referred to as *Policy B* for solving a dynamic stochastic inventory routing problem with backordering (DSIRPB). The DSIRPB is an extension of the previous problem (DSIRP) described in Chapter 5 where a new mixed integer linear programming formulation (MILP) is proposed for the problem. The MILP is used to optimally solved to get the bounds and they are compared with results of *Policy B*. The *Policy B* is embedded with additional controls incorporating the *ABC* and the number of scenarios is fixed at 123, as in the previous chapter. The closeness of *Policy B* and the bounds is reflected by the small ratios obtained and it shows the strength of the method adopted.

Further analysis of the controls to determine which controls contribute the most to the best expected cost found by *Policy B* is carried out. Additional control is introduced where all retailers are visited (all ones) as an addition in *CTRL* because it is assumed that the consumption takes place after the delivery. Results show that *CTRLabc* performs better by contributing on average 57.11% and 66.44% in H=3 datasets for LC and HC respectively. In addition, *CTRLabc* contributed 82.89% and 80.67% for LC and HC respectively in H=6 datasets. The percentage of *CTRLabc* is slightly lower compared to the previous chapter and further examination reveals that the new additional control (all ones in *CTRL*) is selected as the control in period one.

An analysis of the number of visits and the delivery quantities are also carried out to see its pattern. The delivery quantities in the cases of $H = 3$ for HC is generally higher than LC and the number of visits is proportional to the delivery quantities. While in cases of $H = 6$, the best policy found by *Policy B* has delivery quantities of the HC lower than the LC and it is proportional to the number of visits.

As for the backorder quantities, the smaller number of visits the higher is the backorder quantities in all cases. However, due to large demands (because of uniform distribution adopted) specifically for HC, the backorder quantities are unavoidable as the vehicle capacity has achieved maximum level and the split delivery is not allowed.

CHAPTER 7: CONCLUSION AND FUTURE RESEARCH

This chapter gives the summary of the work done in previous chapters. The main contribution of the thesis is the proposal of the metaheuristic method, Artificial Bee Colony (ABC) for solving the Inventory Routing Problem (IRP) and its variants. Four different variants of the IRP are considered. The contribution of knowledge is highlighted in this chapter, and the research objectives are revisited. The results and findings, as well as the suggestions for future research are outlined.

7.1 Summary of Thesis and Contributions

This thesis was divided into seven chapters, where the main contributions based on the problem considered were presented in Chapters 3, 4, 5 and 6. In this thesis, a metaheuristic Artificial Bee Colony (ABC) algorithm was proposed to solve a combinatorial optimization problem the Inventory Routing Problem (IRP). The ABC algorithm was first proposed by Karaboga (2005) to solve numerical optimization, and it was recently proposed for combinatorial optimization. When we first started this research in 2013, there was only a few publications in the combinatorial optimization field. A notable paper is by Szeto et al. (2011) where the ABC was proposed for capacitated Vehicle Routing Problem. This was the main reason and in addition, the flexibility of designing ABC for different problems that motivated us to propose an ABC for the IRP and its variants. The flexibility to manipulate the interactions between the bees offers an ability to produce high quality solutions.

Chapter 1 listed the motivations of our work and the research objectives. Chapter 2 covered the related literature reviews and the chapter was divided into two parts: The literature of the IRP and the literature on ABC where a general ABC algorithm was formally outlined. The general outline of our ABC algorithm for solving the IRP and its variants were also presented in this chapter.

Chapter 3 presented the first contribution to the knowledge, where the IRP that considers a many-to-one distribution network is studied. The network represents an automotive part supply chain that consisted of a depot, assembly plant and multiple suppliers. The demand is deterministic and backordering is not allowed. The designated ABC algorithm (referred to as *ABCIRP*) included an inventory updating mechanism and routes improvement methods in the bee phases. The initial bee population is randomly generated and the routes were constructed using a powerful heuristic, the Giant Tour procedure (Imran et al., 2009). An allocation model was also used to generate one of the initial solutions to diversify the population with a good initial. Forward and backward transfers that involved delivery quantities were developed as the inventory updating mechanism. The inventory updating mechanism aimed to find a balance between the inventory holding cost and transportation cost. The route improvement heuristic implemented was the 1-0 exchange and 2-opt (Lin, 1965). Both inventory updating mechanism and routing improvement were embedded in the employed bee phase and onlooker bee phase, respectively. The performance of *ABCIRP* was tested on an existing benchmark data set where the results showed that *ABCIRP* performed better compared to the Scatter Search (SS) and Genetic Algorithm (GA) with 10 better results out of 14 data sets. A statistical analysis was conducted to determine the significant different between the *ABCIRP*, SS and GA. It was concluded that the *ABCIRP*, SS and GA were significantly different with 95% confidence level and the results obtained using *ABCIRP* are the best among the three metaheuristics. *ABCIRP* was enhanced, denoted as *EABCIRP* and it concerned with the onlooker bee phase and the parameters that guided the intensification and diversification of the search. The onlooker bee phase included the inventory updating mechanism in addition to the route improvements and a new setting for parameters *LIMIT* and *MAXITER* were determined. The solutions were then further improved by applying post optimization using the Dijkstra algorithm to improve the routes. *EABCIRP* gave the best results compared to the *ABCIRP*, SS and GA. The computational time of *EABCIRP* in large datasets S98 was extended to improve the convergence.

The second variant of IRP considered in this thesis was presented in Chapter 4. The IRP takes into account backorder decisions as well as inventory decisions. The problem was known as the inventory routing problem with backordering (IRPB). The one-to-many network distribution consisted of a depot, multiple customers and a fixed number of vehicles that were available to perform the deliveries. Two backordering situations were studied; first, when the vehicle capacity was not enough to satisfy all customers' demand in a period, and second when the saving in transportation cost was higher when compared to the backorder cost imposed by the customer. The ABC for IRPB (referred to as *ABCIRPB*) was a modification from the previous *ABCIRP*. *ABCIRPB* was embedded with an inventory updating mechanism that was capable of handling backorder decisions in addition to inventory decision and routing. The initial bee population was randomly generated where a modified Giant Tour (GT) procedure was proposed to obtain the initial routings. The GT procedure was modified to suit the fixed number of vehicles where eventually the initial backorders were decided. *ABCIRPB* was embedded with a pre-optimization procedure 2-opt and 2-opt* (Potvin & Rousseau, 1995). Two different inventory updating mechanisms proposed were *random exchange*, and *guided exchange*. The *ABCIRPB* embedded with random exchange was referred to as *ABCRX* while guided exchange was referred to as *ABCGX*.

The performances of *ABCRX* and *ABCGX* were tested on 135 benchmark instances and compared with the Estimated Transportation Cost Heuristic (ETCH) from the original literature by Abdelmaguid et al. (2009). The results showed that *ABCRX* and *ABCGX* obtained 20 best results compared to ETCH. The statistical analysis performed showed that the behavior of algorithms, ETCH, *ABCRX* and *ABCGX* were significantly different, which led to the next post hoc test, Bonferroni. The results also showed that both *ABCRX* and *ABCGX* were significantly different from ETCH with a 95% confidence level. The solution to *ABCIRPB* was different from ETCH where the *ABCRX* and *ABCGX* tackled IRPB as a whole, while ETCH decomposed the problem into backorder and inventory subproblems.

The distribution networks considered in both chapters 3 and 4 were many-to-one and one-to-many respectively. The many-to-one is equivalent one-to-many under certain assumptions. The implementation of the ABC algorithm proposed was quite straightforward as the demand considered was deterministic, but it was also challenging to solve as the aim was to balance the transportation, inventory and backorder decisions (only for IRPB). However, solving IRP with stochastic demand required a methodology that was capable of handling the uncertainty component. Thus, a different approach was adopted and the ABC algorithm was embedded in the rollout algorithm and not as a main programme. Chapters 5 and 6 presented the IRP in which the demand was stochastic and dynamic. It was dynamic as the demand was revealed or updated at the end of each period. Hence, the problem was known as Dynamic Stochastic Inventory Routing Problem (DSIRP).

The DSIRP considered in Chapter 5 presented the third contribution of the research. The one-to-many distribution network consisted of a depot (supplier), multiple retailers and the transportations was handled by a third party. The problem considered was similar to Bertazzi et al. (2015) where the Mixed Integer Linear Programming (MILP) formulation with order-up-to level (OU) inventory policy was adopted. An enhanced hybrid rollout algorithm referred to as Policy M^{+} was developed and successfully implemented for the problem. Policy M^{+} was an enhanced version of the Policy M proposed by Bertazzi et al. (2015) where the enhanced policy incorporated with additional controls generated using ABC algorithm (the algorithm is referred to as *ABC_Control*). The *ABC_Control* obtained controls by aiming to fulfill the unsatisfied demand to reduce stock out cost. Another enhancement done was that the number of additional scenarios was set to 120 to capture more possible demands. The performance of Policy M^{+} was tested on 60 datasets by Archetti et al. (2007). A comparison was done using ratios, where the ratios were the average of the expected costs found for the 10 runs of Policy M^{+}divided by the average bound obtained by optimally solved MILP using CPLEX. Small ratios indicated the closeness between the expected cost found and the bound. Ratios of Policy M^{+} were smaller when compared to the Policy M ratios given in Bertazzi et al. (2015) in obtaining the optimal policies where the demands were considered to follow a binomial probability distribution. It was observed that the additional number of scenarios and the $CTRLabc$ (controls obtained by *ABC_Control*) significantly contributed to better ratios of *Policy* M^{+}. This can be seen from the analyses carried out between $CTRLabc$ and $CTRL$ (controls generated similar to Bertazzi et al. (2015)). On average, the $CTRLabc$ contributed 93.78% and 86.22% in LC and HC for $H = 3$, and 94.67% and 91.33% in LC and HC for $H = 6$ to obtain the near optimal policies. The results of *Policy* M^{+} were measured by the expected cost (as the problem was stochastic). It is also interesting to analyse the patterns of the quantity delivered and the number of visits. The number of visits matched the delivery quantity for all datasets for both low inventory cost (LC) and high inventory cost (HC) cases. As an OU inventory policy was adopted, high numbers of visits led to high numbers of delivery quantities.

The capabilities of *Policy* M^{+} were also tested and confirmed on demands that followed a uniform distribution with the small ratios and standard deviation obtained. The analysis of controls were also inspected in which $CTRLabc$ contributed 98.44% and 96.44% in case $H = 3$; 96.22% and 95.78% in case $H = 6$ for LC and HC respectively. A comparison on the patterns of number of visits and the stock out quantity for demands that followed the binomial and uniform probability distribution were carried out. This was to see if there was a different behavior between both distributions. The results showed that the uniform distribution had a higher number of visits compared to the binomial distribution. *Policy* M^+ completely avoided the stock out quantities for all datasets with LC in both distributions. Small stock out quantity and not more than 1.7 units on average were found in HC case datasets for both $H = 3$ and $H = 6$. Thus, it can be concluded that frequent visits was a better strategy to avoid stock out, but to the expense of higher inventory cost as an OU policy is applied.

Chapter 6 is an extension of the DSIRP problem with backorder decision instead of stock out. As the demand is uncertain, stock out is unavoidable, so it is beneficial if the stock out decision can be considered as backorder decision given that the customers are willing to wait. The problem is known as Dynamic Stochastic Inventory Routing Problem with Backordering (DSIRPB). A new formulation of MILP based on dynamic programming model was proposed for the problem and several assumptions were made. The states and the dynamic system were incorporated with backorder decision. The hybrid rollout algorithm developed was referred to as *Policy B*. This was embedded with two additional controls obtained by the proposed ABC algorithm (in which the backorder decision was included) and where all the customers are visited (which was not considered in Chapter 5). The number of scenarios was fixed at 123, as in Chapter 5.

Performance of *Policy B* was tested for demands with uniform probability distribution on the data set by Archetti et al. (2007). The discussion of the results was presented in a similar manner as in Chapter 5, where the ratios were computed to see the closeness of the expected cost found to the bound. The control analysis showed that *CTRLabc* (controls obtained by ABC algorithm) performed better by contributing on average 57.11% and 66.44% in H=3 datasets for LC and HC respectively. In addition, *CTRLabc* contributed 82.89% and 80.67% for LC and HC respectively in H=6 datasets. It was observed that the percentage of *CTRLabc* was slightly lower than the percentages in Chapter 5, and further investigation revealed that the new additional control (all ones in *CTRL*) was selected occasionally, especially in period one.

An analysis of the number of visits, the delivery quantities and the backorder quantities was also carried out. Generally, the delivery quantities in HC cases were higher compared to LC in $H = 3$, and the number of visits was proportional to the delivery quantities. However, the patterns were different in $H = 6$ cases, where the delivery quantities of the HC were lower than the LC and was proportional to the number of visits. For the backorder quantities analyses, it was expected that frequent visits would result in lower backorder quantities. Alternatively, a smaller number of visits resulted in a higher number of backorder quantities. However, due to large demands (because of uniform distribution adopted) specifically for HC, the backorder quantities were unavoidable as the vehicle capacity had achieved maximum level and also because of split delivery is not allowed in this study.

7.2 Suggestions for Future Research

This section presents several suggestions on potential improvement and extension of this research. In the first variant of the IRP, the *ABCIRP* and *EABCIRP* can be embedded with multiple simple improvement procedures, but implemented in systematic (repetitive or combination) ways. This can be seen in Szeto et al. (2011), Mjirda, Jarboui, Macedo, et al. (2014) and Mjirda, Jarboui, Mladenović, et al. (2014). Their implementation of at least six different neighborhood operators (consisting of random insertions, swaps, inversions, remove and replace) diversified the solution (able to avoid trapping in local optimum) and led to a new promising solution space. This may improve the convergence problem suffered by the *EABCIRP*.

Another alternative that can be considered for improving the *ABCIRP* and *EABCIRP* algorithms is to hybridize the ABC with another powerful metaheuristic, such as the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Hybridization with other metaheuristics has proven to be successful, for example by Banharnsakun et al. (2010a) where they combined greedy subtour crossover with ABC. Li, Pan, et al. (2011) proposed Tabu Search (TS) concepts in the employed bee phase to avoid searching the same solution space.

The results obtained by the *ABCRX* and *ABCGX* developed in Chapter 4 seemed to be trapped in a local optimum. It was observed that the inventory updating mechanism (random and guided exchanges) proposed in the employed bee phase needs a longer time to improve the solution (because of the restricted number of vehicles), thus affecting the convergence of the ABCs. However, there is room for improvement. For example, a combination of both random and guided exchanges can be proposed to ensure that the solution is not too random or too rigid. Alternatively, hybridization with different neighborhood operators can be explored to improve the convergence.

So far, the discussion focused on possible improvements for the algorithms itself. There is also another prospect in terms of applying the ABC to another variant of the IRP. A split delivery decision on the IRP model can be studied. Split delivery allows a customer to be served by more than one vehicle, which means that a customer's demand can be split among vehicles. The split delivery concept was first introduced by Dror and Trudeau (1990) for the vehicle routing problem (VRP). Only a few papers exist that considered split delivery within the IRP, such as by Yu et al. (2008), Moin et al. (2011), Mjirda, Jarboui, Macedo, et al. (2014) and recently, Wong and Moin (2017).

ABC algorithm can also be applied to IRP with transshipment. Transshipment is a policy that allows products to be shipped either directly from the supplier or from another customer, to customers with shortages. Coelho et al. (2014a) and Coelho et al. (2014b) introduced the concept of transshipment in their IRP model. The concept of transshipment can also be enforced inside the DSIRP and DSIRPB scheme to avoid stock out or to minimize backorder. The transshipment can be performed after the realization of demand and it is a cost-effective strategy that will lead to customer satisfaction as the demand is uncertain.

The next direction in the IRP is the green IRP (GIRP). One of the global warming factors is greenhouse gas (GHG) emissions. GIRP takes into account the energy usage such as the emission of carbon dioxide (CO_2) from the transportation operations (Soysal et al., 2015, 2018). ABC algorithm can be designed to solve the GIRP with the aim to minimize the CO_2 emissions as well as inventory and transportation costs.

REFERENCES

Abdelmaguid, T. F., & Dessouky, M. M. (2006). A genetic algorithm approach to the integrated inventory-distribution problem. *International Journal of Production Research, 44*(21), 4445-4464.

Abdelmaguid, T. F., Dessouky, M. M., & Ordóñez, F. (2009). Heuristic approaches for the inventory-routing problem with backlogging. *Computers & Industrial Engineering, 56*(4), 1519-1534.

Abu-Mouti, F., & El-Hawary, M. (2009). Modified artificial bee colony algorithm for optimal distributed generation sizing and allocation in distribution systems. *Electrical Power & Energy Conference (EPEC)*, 1-9.

Adelman, D. (2004). A price-directed approach to stochastic inventory/routing. *Operations Research, 52*(4), 499-514.

Adulyasak, Y., Cordeau, J.-F., & Jans, R. (2015). The production routing problem: A review of formulations and solution algorithms. *Computers & Operations Research, 55*, 141-152.

Akay, B., & Karaboga, D. (2009). Parameter tuning for the artificial bee colony algorithm. *International Conference on Computational Collective Intelligence*, 608-619.

Akay, B., & Karaboga, D. (2012). A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences, 192*, 120-142.

Akbari, R., Hedayatzadeh, R., Ziarati, K., & Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. *Swarm and Evolutionary Computation, 2*, 39-52.

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., & Løkketangen, A. (2010). Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research, 37*(9), 1515-1536.

Apple Inc. (2018). Contact Apple for support and service. Retrieved on 13 September, 2018, from https://support.apple.com/en-my/HT201232

Archetti, C., Bertazzi, L., Laporte, G., & Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science, 41*(3), 382-391.

Archetti, C., Bianchessi, N., Irnich, S., & Speranza, M. G. (2014). Formulations for an inventory routing problem. *International Transactions in Operational Research, 21*(3), 353-374.

Archetti, C., & Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization, 2*(4), 223-246.

Archetti, C., & Speranza, M. G. (2016). The inventory routing problem: The value of integration. *International Transactions in Operational Research, 23*(3), 393-407.

Baita, F., Ukovich, W., Pesenti, R., & Favaretto, D. (1998). Dynamic routing-and-inventory problems: a review. *Transportation Research Part A: Policy and Practice, 32*(8), 585-598.

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. *Proceedings of the Second International Conference on Genetic Algorithms*, *206*, 14-21.

Baldacci, R., Toth, P., & Vigo, D. (2010). Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research, 175*(1), 213-245.

Banharnsakun, A., Achalakul, T., & Sirinaovakul, B. (2010a). ABC-GSX: A hybrid method for solving the Traveling Salesman Problem. *Second World Congress on Nature and Biologically Inspired Computing (NaBIC)*, 7-12.

Banharnsakun, A., Achalakul, T., & Sirinaovakul, B. (2010b). Artificial bee colony algorithm on distributed environments. *Second World Congress on Nature and Biologically Inspired Computing (NaBIC)*, 13-18.

Bard, J. F., & Nananukul, N. (2009). The integrated production–inventory–distribution–routing problem. *Journal of Scheduling, 12*(3), 257-280.

Barnes-Schuster, D., & Bassok, Y. (1997). Direct shipping and the dynamic single-depot/multi-retailer inventory system. *European Journal of Operational Research, 101*(3), 509-518.

Baykasoğlu, A., Özbakır, L., & Tapkan, P. (2007). Artificial bee colony algorithm and its application to generalized assignment problem. In F. Chan & M. Tiwari (Eds.), *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization* (pp. 113-144). I-Tech Education and Publishing.

Belfiore, P., Tsugunobu, H., & Yoshizaki, Y. (2008). Scatter search for vehicle routing problem with time windows and split deliveries. In C. Tonci & G. Hrvoje (Eds.), *Vehicle Routing Problem* (pp. 1-14). Croatia: In-Teh.

Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., . . . Prutzman, P. J. (1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces, 13*(6), 4-23.

Berbeglia, G., Cordeau, J.-F., & Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research, 202*(1), 8-15.

Bertazzi, L., Bosco, A., Guerriero, F., & Lagana, D. (2013). A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies, 27*, 89-107.

Bertazzi, L., Bosco, A., & Laganà, D. (2015). Managing stochastic demand in an Inventory Routing Problem with transportation procurement. *Omega*.

Bertazzi, L., Paletta, G., & Speranza, M. G. (2002). Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science, 36*(1), 119-132.

Bertazzi, L., Paletta, G., & Speranza, M. G. (2005). Minimizing the total cost in an integrated vendor—Managed inventory system. *Journal of Heuristics, 11*(5-6), 393-419.

Bertazzi, L., Savelsbergh, M., & Speranza, M. G. (2008). Inventory Routing. In B. Golden, S. Raghavan, & E. Wasil (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges* (pp. 49-72). New York: Springer Science & Business Media, LLC.

Bertazzi, L., & Speranza, M. G. (2012). Matheuristics for Inventory Routing Problems. In J. R. Montoya-Torres, A. A. Juan, L. Huaccho Huatuco, J. Faulin, & G. L. Rodriguez-Verjan (Eds.), *Hybrid Algorithms for Service, Computing and Manufacturing Systems: Routing and Scheduling Solutions* (pp. 1-14). Pennsylvania, USA: IGI Global.

Bertsekas, D. P. (1995). *Dynamic Programming and Optimal Control* (Vol. 1). Belmont, MA: Athena scientific.

Bertsekas, D. P. (2013). Rollout Algorithms for Discrete Optimization: A Survey. In *Handbook of Combinatorial Optimization* (pp. 2989-3013). New York, NY: Springer.

Bertsekas, D. P., & Castanon, D. A. (1999). Rollout algorithms for stochastic scheduling problems. *Journal of Heuristics, 5*(1), 89-108.

Bertsekas, D. P., & Tsitsiklis, J. N. (1995). Neuro-dynamic programming: an overview. *Proceedings of the 34th IEEE Conference on Decision and Control*, *1*, 560-564.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Bertsekas, D. P., Tsitsiklis, J. N., & Wu, C. (1997). Rollout algorithms for combinatorial optimization. *Journal of Heuristics, 3*(3), 245-262.

Bertsimas, D., & Demir, R. (2002). An approximate dynamic programming approach to multidimensional knapsack problems. *Management Science, 48*(4), 550-565.

Blum, C., & Li, X. (2008). Swarm Intelligence in Optimization. In C. Blum & D. Merkle (Eds.), *Swarm Intelligence: Introduction and Applications* (pp. 43-85). Springer Berlin, Heidelberg.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: from Natural to Artificial Systems*. New York: Oxford University Press.

Boudia, M., Louly, M. A. O., & Prins, C. (2007). A reactive GRASP and path relinking for a combined production–distribution problem. *Computers & Operations Research, 34*(11), 3402-3419.

Boudia, M., & Prins, C. (2009). A memetic algorithm with dynamic population management for an integrated production–distribution problem. *European Journal of Operational Research, 195*(3), 703-715.

Campbell, A., Clarke, L., Kleywegt, A., & Savelsbergh, M. (1998). The Inventory Routing Problem. In T. G. Crainic & G. Laporte (Eds.), *Fleet Management and Logistics* (pp. 95-113). New York: Springer Science & Business Media.

Campbell, A., Clarke, L., & Savelsbergh, M. (2002). Inventory routing in practice, The Vehicle Routing Problem, eds P. Toth, D. Vigo. *SIAM Monographs on Discrete Mathematics and Applications*.

Caserta, M., & Voß, S. (2009). Metaheuristics: Intelligent Problem Solving. In V. Maniezzo, T. Stützle, & S. Voß (Eds.), *Matheuristics Hybridizing Metaheuristics and Mathematical Programming* (Vol. 10, pp. 1-38). Heidelberg, London: Springer.

Chien, T. W., Balakrishnan, A., & Wong, R. T. (1989). An Integrated Inventory Allocation and Vehicle Routing Problem. *Transportation Science, 23*(2), 67-76.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2014a). Heuristics for dynamic and stochastic inventory-routing. *Computers & Operations Research, 52*, 55-67.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2014b). Thirty years of inventory routing. *Transportation Science, 48*(1), 1-19.

Coelho, L. C., & Laporte, G. (2015). An optimised target-level inventory replenishment policy for vendor-managed inventory systems. *International Journal of Production Research, 53*(12), 3651-3660.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*(1), 3-18.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik, 1*(1), 269-271.

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science, 344*(2-3), 243-278.

Dror, M., & Trudeau, P. (1990). Split delivery routing. *Naval Research Logistics (NRL), 37*(3), 383-402.

Duan, H.-b., Xu, C.-f., & Xing, Z.-H. (2010). A hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems. *International Journal of Neural Systems, 20*(01), 39-50.

Erengüç, Ş. S., Simpson, N. C., & Vakharia, A. J. (1999). Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research, 115*(2), 219-236.

Federgruen, A., & Zipkin, P. (1984). A Combined Vehicle Routing and Inventory Allocation Problem. *Operations Research, 32*(5), 1019-1037.

Fingas, R. (2018). Here are all of the Apple retail store openings, moves, and remodels in 2018. Retrieved on 13 September, 2018, from https://appleinsider.com/articles/

18/07/14/here-are-all-of-the-apple-retail-store-openings-moves-and-remodels-in-2018

Fisher, M. L., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks, 11*(2), 109-124.

Frost, J. (2017). How to Identify the Distribution of Your Data. Retrieved on 13 September, 2018, from http://statisticsbyjim.com/hypothesis-testing/identify-distribution-data/

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms* (Vol. 1, pp. 69-93). Elsevier.

Golden, B., Assad, A., & Dahl, R. (1984). Analysis of a large scale routing problem with an inventory component. *Large Scale Systems*.

Gomez, A., Imran, A., & Salhi, S. (2013). Solution of classical transport problems with bee algorithms. *International Journal of Logistics Systems and Management, 15*(2-3), 160-170.

Imran, A., Salhi, S., & Wassan, N. A. (2009). A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research, 197*(2), 509-518.

Kang, F., Li, J., Li, H., Ma, Z., & Xu, Q. (2010). An improved artificial bee colony algorithm. *2nd International Workshop on Intelligent Systems and Applications (ISA)*, 1-4.

Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Technical report-tr06). Retrieved from Erciyes university, engineering faculty, computer engineering department: https://pdfs.semanticscholar.org/015d/f4d97ed1f541752842c49d12e429a785460b.pdf

Karaboga, D., & Akay, B. (2009). A comparative study of Artificial Bee Colony. *Applied Mathematics and Computation, 214*(1), 108-132.

Karaboga, D., & Akay, B. (2011). A modified artificial bee colony (ABC) algorithm for constrained optimization problems. *Applied Soft Computing, 11*(3), 3021-3031.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459-471.

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing, 8*(1), 687-697.

Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review, 42*(1), 21-57.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks IV, pages*, *1000*.

Kleywegt, A. J., Nori, V. S., & Savelsbergh, M. W. (2002). The stochastic inventory routing problem with direct deliveries. *Transportation Science, 36*(1), 94-118.

Kleywegt, A. J., Nori, V. S., & Savelsbergh, M. W. (2004). Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science, 38*(1), 42-70.

Lee, C.-G., Bozer, Y. A., & White III, C. (2003). *A heuristic approach and properties of optimal solutions to the dynamic inventory routing problem*. Working Paper. Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada.

Lei, L., Liu, S., Ruszczynski, A., & Park, S. (2006). On the integrated production, inventory, and distribution routing problem. *IIE Transactions, 38*(11), 955-970.

Lenstra, J. K., & Rinnoy Kan, A. (1981). Complexity of vehicle routing and scheduling problems. *Networks, 11*(2), 221-227.

Li, J., Pan, Q., & Xie, S. (2011). Flexible job shop scheduling problems by a hybrid artificial bee colony algorithm. *IEEE Congress on Evolutionary Computation (CEC)*, 78-83.

Li, J., Xie, S.-x., Pan, Q.-k., & Wang, S. (2011). A hybrid artificial bee colony algorithm for flexible job shop scheduling problems. *International Journal of Computers Communications & Control, 6*(2), 286-296.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal, 44*(10), 2245-2269.

Mastin, A., & Jaillet, P. (2015). Average-case performance of rollout algorithms for knapsack problems. *Journal of Optimization Theory and Applications, 165*(3), 964-984.

Meloni, C., Pacciarelli, D., & Pranzo, M. (2004). A rollout metaheuristic for job shop scheduling problems. *Annals of Operations Research, 131*(1-4), 215-235.

Mentzer, J. T., DeWitt, W., Keebler, J. S., Min, S., Nix, N. W., Smith, C. D., & Zacharia, Z. G. (2001). Defining supply chain management. *Journal of Business Logistics, 22*(2), 1-25.

Mezura-Montes, E., Damián-Araoz, M., & Cetina-Domingez, O. (2010). Smart flight and dynamic tolerances in the artificial bee colony for constrained optimization. *IEEE Congress on Evolutionary Computation (CEC)*, 1-8.

Minkoff, A. S. (1993). A Markov decision model and decomposition heuristic for dynamic vehicle dispatching. *Operations Research, 41*(1), 77-90.

Mjirda, A., Jarboui, B., Macedo, R., Hanafi, S., & Mladenović, N. (2014). A two phase variable neighborhood search for the multi-product inventory routing problem. *Computers & Operations Research, 52*, 291-299.

Mjirda, A., Jarboui, B., Mladenović, J., Wilbaut, C., & Hanafi, S. (2014). A general variable neighbourhood search for the multi-product inventory routing problem. *IMA Journal of Management Mathematics, 27*(1), 39-54.

Moin, N. H., Halim, H. Z. A., & Yuliana, T. (2014). Metaheuristics for multi products inventory routing problem with time varying demand. *AIP Conference Proceedings*, *1605*(1), 3-9.

Moin, N. H., & Salhi, S. (2007). Inventory routing problems: a logistical overview. *Journal of the Operational Research Society, 58*(9), 1185-1194.

Moin, N. H., Salhi, S., & Aziz, N. A. B. (2011). An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. *International Journal of Production Economics, 133*(1), 334-343.

Musil, S. (2016). Apple says it has sold out of initial quantities of iPhone 7 Plus. Retrieved on 13 September, 2018, from https://www.cnet.com/news/apple-says-it-has-sold-out-of-initial-quantities-of-iphone-7-plus/

Niu, S.-C. (2018). Important Probability Distribution. Retrieved on 13 September, 2018 from http://utdallas.edu/~scniu/OPRE6301/documents/Important_Probability_Distributions.pdf

Osman, I. H., & Kelly, J. P. (1996). Meta-heuristics: an overview. In I. H. Osman & J. P. Kelly (Eds.), *Meta-heuristics: Theory and Applications* (pp. 1-21). Massachusetts: Kluwer Academic Publishers.

Pal, A., Chan, F., Mahanty, B., & Tiwari, M. (2011). Aggregate procurement, production, and shipment planning decision problem for a three-echelon supply chain using swarm-based heuristics. *International Journal of Production Research, 49*(10), 2873-2905.

Pan, Q.-K., Tasgetiren, M. F., Suganthan, P. N., & Chua, T. J. (2011). A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences, 181*(12), 2455-2468.

Peres, I. T., Repolho, H. M., Martinelli, R., & Monteiro, N. J. (2017). Optimization in inventory-routing problem with planned transshipment: A case study in the retail industry. *International Journal of Production Economics, 193*, 748-756.

Pochet, Y., & Wolsey, L. A. (1988). Lot-size models with backlogging: Strong reformulations and cutting planes. *Mathematical Programming, 40*(1-3), 317-335.

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

Powell, W. B. (2011). *Approximate Dynamic Programming: Solving the Curses of Dimensionality* (2nd ed.). United States: John Wiley & Sons.

Puchinger, J., & Raidl, G. R. (2005). Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. *International Work-Conference on the Interplay Between Natural and Artificial Computation*, 41-53.

Raa, B., & Aghezzaf, E.-H. (2009). A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research, 192*(2), 429-441.

Reiman, M. I., Rubio, R., & Wein, L. M. (1999). Heavy traffic analysis of the dynamic stochastic inventory-routing problem. *Transportation Science, 33*(4), 361-380.

Ribeiro, C. C., & Hansen, P. (Eds.). (2012). *Essays and Surveys in Metaheuristics* (Vol. 15). Springer Science & Business Media LLC.

Schmid, V., Doerner, K. F., & Laporte, G. (2013). Rich routing problems arising in supply chain management. *European Journal of Operational Research, 224*(3), 435-448.

Secomandi, N. (2001). A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research, 49*(5), 796-802.

Secomandi, N. (2003). Analysis of a rollout approach to sequencing problems with stochastic routing applications. *Journal of Heuristics, 9*(4), 321-352.

Solyali, O., Cordeau, J.-F., & Laporte, G. (2012). Robust inventory routing under demand uncertainty. *Transportation Science, 46*(3), 327-340.

Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., & van der Vorst, J. G. (2015). Modeling an Inventory Routing Problem for perishable products with environmental considerations and demand uncertainty. *International Journal of Production Economics, 164*, 118-133.

Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., & van der Vorst, J. G. (2018). Modeling a green inventory routing problem for perishable products with horizontal collaboration. *Computers & Operations Research, 89*, 168-182.

Stützle, T. (1999). *Local search algorithms for combinatorial problems*. (PhD Thesis), Darmstadt University of Technology.

Subotic, M., Tuba, M., & Stanarevic, N. (2010). Parallelization of the artificial bee colony (ABC) algorithm. *Proceedings of the 11th WSEAS International Conference on Neural Networks and 11th WSEAS International Conference on Evolutionary Computing and 11th WSEAS International Conference on Fuzzy Systems, World Scientific and Engineering Academy and Society (WSEAS). 10*, 191-196.

Sundar, S., & Singh, A. (2010). A swarm intelligence approach to the quadratic multiple knapsack problem. *International Conference on Neural Information Processing*, 626-633.

Sundar, S., Singh, A., & Rossi, A. (2010). An artificial bee colony algorithm for the 0–1 multidimensional knapsack problem. *International Conference on Contemporary Computing*, 141-151.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction* (Vol. 1). MIT press Cambridge.

Szeto, W., Wu, Y., & Ho, S. C. (2011). An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research, 215*(1), 126-135.

Teo, C. P., Ou, J., & Goh, M. (2001). Impact on inventory costs with consolidation of distribution centers. *IIE Transactions, 33*(2), 99-110.

Toth, P., & Vigo, D. (2002). *The Vehicle Routing Problem*. Philadelphia: SIAM Society for Industrial and Applied Mathematics.

Tsai, P.-W., Pan, J.-S., Liao, B.-Y., & Chu, S.-C. (2009). Enhanced artificial bee colony optimization. *International Journal of Innovative Computing, Information and Control, 5*(12), 5081-5092.

Voß, S., Martello, S., Osman, I. H., & Roucairol, C. (2012). *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Springer Science & Business Media.

Wong, L., & Moin, N. H. (2017). Ant Colony Optimization For Split Delivery Inventory Routing Problem. *Malaysian Journal of Computer Science, 30*(4), 333-348.

Yao, Y., Evers, P. T., & Dresner, M. E. (2007). Supply chain integration in vendor-managed inventory. *Decision Support Systems, 43*(2), 663-674.

Yu, Y., Chen, H., & Chu, F. (2008). A new model and hybrid approach for large scale inventory routing problems. *European Journal of Operational Research, 189*(3), 1022-1040.

Yu, Y., Chu, C., Chen, H., & Chu, F. (2012). Large scale stochastic inventory routing problems with split delivery and service level constraints. *Annals of Operations Research, 197*(1), 135-158.

Yu, Y., Wang, Z., & Liang, L. (2012). A vendor managed inventory supply chain with deteriorating raw materials and products. *International Journal of Production Economics, 136*(2), 266-274.

Zhang, D., Guan, X., Tang, Y., & Tang, Y. (2011). Modified artificial bee colony algorithms for numerical optimization. *3rd International Workshop on Intelligent Systems and Applications (ISA)*, 1-4.

Zhang, S., Lee, C., Choy, K., Ho, W., & Ip, W. (2014). Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem. *Transportation Research Part D: Transport and Environment, 31*, 85-99.

Ziarati, K., Akbari, R., & Zeighami, V. (2011). On the performance of bee algorithms for resource-constrained project scheduling problem. *Applied Soft Computing, 11*(4), 3720-3733.

LIST OF PUBLICATIONS AND PAPERS PRESENTED

PUBLICATIONS

Halim, H. Z. A., & Moin, N. H. (2014). Solving inventory routing problem with backordering using Artificial Bee Colony. *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 913-917.

Moin, N. H., **Halim, H. Z. A.**, & Yuliana, T. (2014). Metaheuristics for multi products inventory routing problem with time varying demand. *AIP Conference Proceedings*, *1605*(1), 3-9.

PAPER PRESENTED

Halim, H. Z. A. & Moin, N.H. (2015, 24 November). Modified Artificial Bee Colony for The Integrated Inventory Routing Problem with Backordering. Paper presented at the 23rd National Symposium on Mathematical Science, Universiti Teknologi Malaysia, Johor.

Halim, H. Z. A. & Moin, N. H. (2014, 10 December). Solving inventory routing problem with backordering using Artificial Bee Colony. Paper presented at the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Selangor, Malaysia.

Halim, H. Z. A. & Moin, N.H. (2014, 25 November). Artificial Bee Colony for Inventory Routing Problem with Backordering. Paper presented at the 22nd National Symposium on Mathematical Science, Shah Alam, Selangor.