MODIFIED ANT COLONY OPTIMIZATION ALGORITHMS FOR DETERMINISTIC AND STOCHASTIC INVENTORY ROUTING PROBLEMS

LILY WONG

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

2018

UNIVERSITY OF MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: LILY WONG

Matric No: SHB110013

Name of Degree: DOCTOR OF PHILOSOPHY

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

MODIFIED ANT COLONY OPTIMIZATION ALGORITHMS FOR DETERMINISTIC AND STOCHASTIC INVENTORY ROUTING PROBLEMS

Field of Study: APPLIED MATHEMATICS

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:

MODIFIED ANT COLONY OPTIMIZATION ALGORITHMS FOR DETERMINISTIC AND STOCHASTIC INVENTORY ROUTING PROBLEMS

ABSTRACT

Inventory routing problem (IRP) integrates two important components of supply chain management: routing and inventory management. In this study, a one-to-many IRP network comprises of a single depot (warehouse) and geographically dispersed customers in a finite planning horizon is presented. Multi products are transported from the warehouse by using a fleet of a homogeneous vehicle which located at the ware house to meet customer's demand on time. The customers are allowed to be visited more than once in a given period and the demand for each product is deterministic and time varying. The problem is formulated as a mixed integer programming problem and is solved using CPLEX to obtain the lower and upper bound (the best integer solution) for each instance considered. The classical Ant Colony Optimization (ACO) is modified by including the inventory cost in the global pheromones updating is proposed in this study. The sensitivity analysis on important parameters that influence decision policy in ACO in order to choose the appropriate parameter settings is carried out. Among the two proposed algorithms, that is, ACO and ACO2, ACO2 outperform than ACO. Both ACO and ACO2 perform better on large instances compared to the upper bound and perform equally well for small and medium instances. In order to improve the proposed algorithms, population based ACO where the ants are subdivided into subpopulations and each subpopulation represents one inventory level is proposed. In addition, a new formulation for customer's inventory pheromones is proposed and the selection of inventory updating mechanism is based on these pheromone values. The computational results show that the algorithms which implement this new formulation are able to produce better solutions. The computational results also show that the algorithms of population based ACO performs better than the algorithms of non-population based ACO. The deterministic IRP model is then extended to solve the Stochastic Inventory Routing Problem (SIRP). The demands in SIRP are modeled by some probability functions and due to the stochastic nature of customer demands, the service levels constraint where it limits the stock out probability at each customer and the probability of overfilling the stock of each customer is introduced in this study. A two phase algorithm named SIRPACO1 is proposed to solve the SIRP. Phase I solved the inventory sub problem to determine the quantity to be delivered to each customer as well as inventory level at each customer while Phase II employs the population based ACO to determine the routes for each period. The algorithm was further enhanced by incorporating the inventory updating mechanism into Phase II with the aim of obtaining a set of inventory level which will give minimum overall cost and named as SIRPACO2. The computational experiments are tested on different combinations of two important parameters that are standard deviation and service level. The computational results showed that the enhanced SIRPACO2 gave better performance compared to SIRPACO1.

Keywords: ant colony optimization, inventory routing problem

ALGORITMA PENGOPTIMUMAN KOLONI SEMUT TERUBAH SUAI UNTUK MASALAH LALUAN INVENTORI BERKETENTUAN DAN STOKASTIK

ABSTRAK

Masalah laluan inventori (*Inventori Routing Problem (IRP)*) mengintegrasikan dua komponen yang penting dalam pengurusan rantaian bekalan iaitu masalah laluan dan pengurusan inventori. Dalam kajian ini, satu masalah *IRP* rangkaian satu-ke-banyak yang terdiri daripada depot (gudang) tunggal dan pelanggan-pelanggan berserakan secara geografi dalam satu perancangan ufuk yang terhingga dikaji. Pelbagai produk akan dihantar dari gudang dengan menggunakan sejumlah kenderaan homogen yang berpusat di depot bagi memenuhi permintaan pelanggan dalam setiap tempoh masa. Pelanggan boleh dikunjungi lebih daripada satu kali dalam masa yang ditertukan dan permintaan bagi setiap produk adalah tetap dan tempoh masa adalah berbeza-beza. Masalah ini dirumuskan sebagai masalah pengaturcaraan integer campuran dan diselesaikan dengan menggunakan CPLEX bagi mendapatkan batas bawah dan atas (penyelesaian integer terbaik) bagi setiap kes yang dipertimbangkan. Kaedah klasik pengoptimuman koloni semut (*Ant Colony Optimisation* (*ACO*)) diubahsuai dengan mengambil kira kos inventori semasa mengemaskini feromon global dicadangkan dalam kajian ini. Analisis sensitiviti dijalankan untuk menentukan nilai parameter penting yang mempengaruhi ACO dalam membuat keputusan. Antara dua algoritma yang dibangunkan, ACO2 adalah lebih baik berbanding ACO. Kedua-dua algoritma ACO dan ACO2 memberi keputusan yang lebih baik jika dibandingkan dengan batas atas untuk kes besar dan keputusan yang setara bagi kes kecil dan sederhana. Penambahbaikan algoritma ACO dicadangkan iaitu ACO populasi di mana semut-semut dibahagikan kepada sub-populasi dan setiap sub-populasi mewakili satu tahap inventori.

Di samping itu, satu formula baru yang dikenali sebagai feromon inventori pelanggan telah dicadangkan dan juga pemilihan mekanisme mengemaskini inventori adalah berdasarkan kepada nilai-nilai feromon. Keputusan pengiraan menunjukkan bahawa algoritma yang menggunakan formula baru ini mampu menghasilkan keputusan yang lebih baik. Keputusan mengiraan juga menunjukkan algoritma *ACO* populasi memberi keputusan yang lebih baik jika dibandingkan dengan algoritma bukan *ACO* populasi. . Model IRP yang berketentuan kemudiannya dilanjutkan kepada masalah laluan inventori stokastik (*Stochastic Inventory Routing Problem* (*SIRP*)). Permintaan pelanggan dalam *SIRP* dimodelkan oleh beberapa fungsi kebarangkalian dan disebabkan permintaan pelanggan bersifat stokastik, kekangan aras perkhidmatan telah diperkenalkan di mana kekangan tersebut akan menghadkan kebarangkalian kehabisan stok pada setiap pelanggan dan juga kebarangkalian penambahan stok yang berlebihan bagi setiap pelanggan. Satu algoritma dua fasa yang dinamakan SIRPACO1 telah dicadangkan untuk menyelesaikan *SIRP*. Fasa I menyelesaikan sub masalah inventori untuk menentukan kuantiti penghantaran serta tahap inventori pada setiap pelanggan manakala Fasa II menggunakan *ACO* populasi untuk menentukan laluan kenderaan bagi setiap tempoh masa. Algoritma yang dicadangkan kemudiannya di tambah baik dengan memasukkan mekanisme mengemaskini inventori ke Fasa II dengan tujuan untuk mendapatkan satu set aras inventori yang akan memberikan minimum kos keseluruhan dan algoritma ini diberi nama SIRPACO2. Eksperimen komputasi dijalankan untuk kombinasi dua parameter yang penting iaitu sisihan piawai dan aras perkhidmatan. Keputusan pengiraan menunjukkan bahawa SIRPACO2 memberi keputusan yang lebih baik berbanding SIRPACO1.

Kata kunci: pengoptimuman koloni semut, masalah laluan inventori

ACKNOWLEDGEMENTS

First of all, I would like to express my infinite gratitude to Associate Professor Dr. Noor Hasnah Moin, my supervisor, for her guidance, encouragement, patience, advice and critical reviews towards completion of my study and thesis. Without her guidance and help, I could never accomplish this difficult task. I would like to take the opportunity to thank the University of Malaya and Ministry of Higher Education, Malaysia to support my tuition fees and living expenses via *Skim Biasiswazah* (Semester 1 and 2) and MyPhD (Semester 3 to Semester 8) respectively. This research is supported also by Postgraduate Research Fund (PG020-2012B). I would also like to extend my gratitude to those who give me the advice and providing related information throughout my study especially for the part of statistical. Without your help, I could have more difficulty in completion my study. In addition, I would also like to thank all my fellow friends who give me support and encouragement when I encounter problem and in the border of giving up. It is a great pleasure to acknowledge my deepest gratitude to my beloved husband for his support, encouragement and comprehensive. Thank you for being there with me every moment when I faced problem and under stress. And last but not least, my deepest appreciation goes to my dearest parents and family for their greatest encouragement and unlimited support. Thank you so much for kept me going and pushing me farther than I thought I could go.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

SYMBOLS

$t = 1, 2, \dots, T$	period index
$W = 0$	warehouse/depot
$S = 1, 2, \dots, N$	a set of customers where customer *i* demands product *i* only
C	vehicles capacity (assume to be equal for all the vehicles).
F	fixed vehicle cost per trip (assumed to be the same for all periods)
V	travel cost per unit distance
M	the number of vehicles and it is assumed to be ∞ (unlimited)
m	predefined number of ants
c_{ij}	travel distance between customer *i* and *j* where $c_{ij} = c_{ji}$ and the triangle inequality, $c_{ik} + c_{kj} \geq c_{ij}$ holds for any different *i*, *j*, and *k* with $i \neq j$, $k \neq i$ and $k \neq j$
h_i	inventory carrying cost at the customer for product *i* per unit product per unit time
d_{it}	demand of customer i in period *t*
$d_{i,(1,t)}$	$=\sum_{f=1}^{t} d_{it}$cumulative stochastic demand from period 1 to *t*
a_{it}	delivery quantity to customer i in period *t*
I_{it}	inventory level of product *i* at the customer *i* at the end of period *t*
$I_{it} = \max(0, I_{it})$	On-hand inventory of customer *i*at the end of period *t*, which excludes the stock-out ($I_{it} < 0$).
q_{ijt}	quantity transported through the directed arc (ij) in period *t*
x_{ijt}	number of times that the directed arc (ij) is visited by vehicles in period *t*
ρ	the rate of pheromone evaporation

μ_i	mean of customer i
σ_i	standard deviation of customer i
$\hat{R}_i^2$	equal to the rank total of the ith problem
$\hat{R}_j^2$	equal to the rank total of the jth algorithm
R_i	the average rankings by the Friedman test of the ith algorithms
R_0	the average rankings by the Friedman test of the average ranking of the control method.
L_{ini}	the total of distance obtained from the heuristic algorithms.
W_i	the inventory capacity for customer site.

ABBREVIATION

N_DEM / predefinedIterInvUpt	Predefined number of iterations for updating inventory mechanism
MaxiITER	Predefined number of iterations
N_GL / PredefinedIterationForGL	Predefined number of iterations for global pheromones updating
N_moveData	maximum number of moves to be allowed for each data
N_moveTime	the maximum number of move to be allowed per time
temp_move	the current number of moves
sum_move	the current accumulative moves that have been done
cur_move	number of move generated by random number which is no more than *N_moveTime* per time
L_{nn}	the total distance obtained from nearest neighbor algorithm
Predefinedprob	predefined values of selecting the forward and backward transfer.
PredefinedIterChgInvMeth	predefined number of iteration for selecting the customer deterministically or randomly for transferring their quantity delivery in inventory updating mechanism
PredefinedMove	Predefine number of transfer to be allowed per time
$invpher_{jp}$	pheromones of customer's inventory for customer *j* in period *p*
predefinedIterInvUptGL	predefined number of iteration for global pheromones updating of customer's inventory

N_prob	predefined values of selecting the customers based on attraction or using deterministic backward and forward mechanism

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

This chapter briefly gives the introduction to Inventory Routing Problem (IRP). The chapter starts by introducing the inventory routing problem (IRP), describing its importance and its relevance to the current economic situations. This is followed by a discussion on the problem statements of the study. The objectives of this study are outlined in the following section and finally, this chapter concludes by giving the organization of the thesis.

1.1 Introduction

Customers play important roles in logistics management as more companies are competing to improve their services including quality, on time delivery, warranty and repair services, pricing contracts, and remanufacturing. The timeliness and consistency of delivery are two aspects which are desired to be improved by many companies as it will offer greater values to their customers. The integration of different components of supply chain such as production, inventory and distribution yields new benefits to balance the setup, holding, inventory and delivery costs while tightly managing available resources. Moreover, the integration can have a significant impact on overall system performance.

Vendor Managed Inventory (VMI) is one of good examples of the type of integration which mentioned above. In VMI model, the supplier or manufacturer observes and controls the inventory levels of its customers or retailers. This is different from the conventional approach where the customers monitor their own inventory and decide the time and amount of products to reorder. One of the most important benefits of VMI is that it allows a more uniform utilization of transportation resources. This leads to a higher level of efficiency and a much lower distribution cost that often constitutes the largest part of the overall cost. VMI created a win–win situation for both suppliers and customers where the vendors can have the savings in terms of distribution costs by being able to coordinate the deliveries to different customers in more efficient way while customers do not have to dedicate resources to inventory management.

Inventory management and transportation are two of the important components in Supply Chain Management (SCM). The coordination of these two components is often recognized as the Inventory Routing Problems (IRP). In addition, the other components such as production, location, marketing and purchasing (see Moin and Salhi, 2007 and Coelho et al., 2013a for an overview) can also be taken into account but it depends on the model of the IRP considered.

IRP is a challenging NP-hard problem that combines the vehicle routing problem (VRP) and inventory management. The VRP, itself is NP-hard (Cordeau et. al. (2007) that determines a set of routes to visit the customers and the inventory management which concerns the amount to be delivered to the customers.There are three simultaneous decisions that have to be made by the supplier: when to serve a given set of customers, the amount to deliver to the customer when it is served and how to combine the customers into the vehicle routes. The main objective of IRP is to minimize both the total transportation that consists of fixed and variable costs and inventory cost over the planning horizon. IRP arises in many distribution systems, especially in Vendor Managed Inventory (VMI). In general, applications of IRP arise in large variety of industries including liquefied natural gas and ship routing problems, distribution of automobile components and perishable items, groceries distribution, transportation for cement, blood as well as the waste organic oil.

IRP has been the focus of many researches and since it is NP-hard the exact algorithms proposed are able to solve relatively small instances. Hence many metaheuristic methods have been developed, such as genetic algorithms and tabu search to suit the problems that lead to optimal or near optimal solutions. We propose a modified and enhanced ant colony optimization (ACO) to solve our models. ACO is initially proposed by Dorigo and coworkers (Dorigo, 1992, Dorigo and Blum, 2005, and Di Caro and Dorigo, 1999) and also the first algorithm which targeting to find for an optimal path in a graph based on the nature behavior of ants to seek for a path between the source of food and their colonies. Further discussions on ACO are presented in Chapter 2.

1.2 Problem Statements

The integration of various aspects of supply chain management is an important component for companies to remain competitive. Since the IRP is known to be NP-hard, developing metaheuristic algorithms have been crucial and these have motivated us to seek good and efficient methods / algorithms to achieve the objective of the IRP problem. Several variants of IRP have been studied in the past literature, ranging from deterministic demand cases to stochastic models. Most models considered in vendor managed resupply require accurate and timely information about the inventory status of customers and often the customers' demands are considered to be deterministic where the demand is known in advance. The model considers deterministic customer's demand where consumption rate is fixed and known in advance. The main concern of the researchers who studied deterministic demand is to find the solution of which customer to be visited in each period, how much to deliver to each customer and also the delivery routes based on the known demand. However, in real world, there are many industries /

companies where their customer's consumption rate are difficult to predict with certainty and can only be represented at best by a random variable with known probability distribution. This problem is modeled as stochastic. The natural objective of stochastic model is to minimize the total expected costs. The scope of this study covers both deterministic and stochastic IRP.

1.3 Objective of this study

The objective of this study is stated as below:

1. To modify the conventional Ant Colony Optimization (ACO) to solve our proposed model of Deterministic Inventory Routing Problem (DIRP).
    - propose a new modified model of DIRP which considers multi products in the model and also allows the customers to be served by more than one vehicle (i.e. split delivery),
    - modify conventional ACO by incorporating the inventory component in the local and global pheromone to reflect the importance of the inventory component, and
    - perform the sensitivity analysis in order to obtain the appropriate parameter which gives better solution to our proposed model.

2. To enhance the modification of ACO by proposing the population based ACO to solve our proposed model of DIRP,
    - develop the population based ACO which is different from the conventional ACO that only have one population to construct the solutions, and

- develop a new mathematical formulation which can incorporate the information of transportation into the inventory management in order to attain the best set of inventory with minimum transportation costs.

3. To extend our study to cater for Stochastic Inventory Routing Problem (SIRP).
    - extend our DIRP model to SIRP in which the demand of customer is only known in probabilistic sense,
    - embed the service level constraint which prevent the stock out cost and also overloading at the customer's warehouse, and
    - Propose a population based ACO to construct the solution.

1.4 Contribution to Scientific Knowledge

There are several scientific contributions that this thesis proposes to the literature of IRP, mainly consisting of the development of new models and algorithms. The specific contributions are outlined as follows:

- Nowadays, many companies supply not only a single product to their customers but multi products, thus, in this study, we propose a one-to-many network of IRP that supplies multi products. We also allow split delivery in which each customer is allowed to be served by more than one vehicle. The aim of implementing the split delivery in our model is to increase vehicle utilization, thus reducing the number of vehicles. The details of the model are discussed in Chapter 3.
- In literature, the conventional ACO only consists of one population to construct the solutions. In our study, we propose a new algorithm of ACO, in which we subdivide the ants into subpopulation to build the solution and this is discussed

in Chapter 4. By implementing this, we give the ants more exploration in order to obtain better results. A new mathematical formulation is introduced in the global updating scheme which carries the information on inventory in order to build a set of inventory level that can balance between the inventory and transportation cost.

- The final contribution of this thesis is that we extend our DIRP model to solve SIRP (Chapter 5). We embed the constraint of service level which avoids excessive the stock out cost and overfilling at the customer's warehouse in our model. In this study, we modify our population based ACO where each subpopulation starts with different initial solution (differentiated by level of inventory) instead of starting with the same initial solution for all the subpopulations. Different heuristics are employed to generate different initial solutions.

1.5 Thesis Outlines

The thesis consists of six chapters and the organization of the thesis is discussed below. In Chapter 2, a literature review on the variants of IRP model is discussed in details. The literature review is grouped based on the type of the demand which we consider in this study (i.e. deterministic and stochastic). In addition, a review regarding Ant Colony Optimization (ACO) for which our algorithm is based on is given in this chapter as well. The aim of this chapter is to identify the motivation that leads to the objectives behind the thesis.

Chapter 3 presents our proposed model of Deterministic Inventory Routing Problem (DIRP) where the demand of customer is known in advance. In this study, we focus on modifying ACO for solving the DIRP model and the details of the algorithms are discussed in this chapter. The related computational results and the discussion are presented in the later part of this chapter.

Chapter 4 presents the population based ACO which subdivides the population into several number of population to build the solutions. This is different from the conventional ACO which only have one population to build the solution. Apart from proposing population based ACO, another objective of this enhancement algorithm in this chapter is to integrate the information of the transportation into the inventory updating mechanism in order to construct a set of customers' inventory which can balance between the inventory and transportation cost. The corresponding results and discussion are given in the later part of this chapter.

Meanwhile, Chapter 5 presents the extension of our DIRP model to cater the problem where the customer demand is unknown in advance which is called stochastic inventory routing problem (SIRP). In this study, we modify our population based ACO such that it can be incorporated into our proposed algorithms for solving our SIRP model. The proposed algorithms are presented in details in this chapter. The computational results as well as the related discussion are explained in details in Chapter 5.

Finally, the last chapter (i.e. Chapter 6) gives the summary about the main findings of the thesis and concludes with the potential research directions for future research.

CHAPTER 2: LITERATURE REVIEW

This chapter presents the literature review of Inventory Routing Problem (IRP). The chapter starts with the introduction and followed by the discussion on the classification of different types of inventory routing problems. Then, the next two sections discuss the literature review on Deterministic Inventory Routing Problem (DIRP) and Stochastic Inventory Routing Problem (SIRP) respectively. Meanwhile, discussion of the literature review pertaining Ant Colony Optimization (ACO) is discussed in this chapter as well. Finally, the chapter concludes with a summary.

2.1 Introduction

Supply Chain Management (SCM) is the control of supply chain to manage the flow of commodity both within and among the companies. In order to remain competitive, companies are proposing innovative ways of optimizing their supply chain by integrating certain parts of the supply chain (see for example Moon et al. (2006) which coordinates the planning and scheduling of the supply chain and Yang and Liu (2013) when the coordination involves some uncertainties). Inventory Routing Problem (IRP) involves the integration and coordination of inventory management and transportation, where the customers rely on a central supplier to deliver the commodity on a repeated basis. The main objective of this problem is to minimize the corresponding costs (fixed and variable costs) under the constraint that the deliveries to customers are on time.

The IRP is relevant in the Vendor Managed Inventory (VMI) strategy. Under this strategy, the supplier or manufacturer decides when to visit their customers, the quantity of delivery and how to combine them into vehicle routes. VMI is argued to be beneficial to both the customer and the supplier although the supplier may take a longer period of adjustment and reconfiguration before the benefits of VMI can be realized (Dong and Xu, 2002). Applications include the distribution of liquefied natural gas and ship routing problems, distribution of raw material to the paper industry, and food distribution to supermarket chains, among others. The literature on IRP has received considerable attention in the last decade. Methods of solving IRP can be divided into two categories: the exact methods and heuristics or metaheuristics approaches. The classification of IRP is presented next.

2.2 Classification of Different Types of Inventory Routing Problems

In the past researches, many different varieties of IRP have been developed and solved. Bell et al. (1983) first investigated the integration between inventory management and vehicle scheduling. There are various versions of IRP that have been studied extensively. IRP indeed can be modeled in different ways depending on its characteristic. In fact, there is no standard version of the problem. In paper by Coelho et al. (2013a), the authors cited that most of the research efforts have been concentrated on 'basic versions' while the study on extended models, denoted as 'extension of the basic version' are relatively new. The basic version of IRP is shown in Table 2.1. Generally, the basic version can be classified into seven different criteria; time horizon, structure, routing, inventory policy, inventory decisions, fleet composition and fleet size.

From Table 2.1, the criteria of time horizon is divided into finite and infinite. Most of the earlier works concentrates on an infinite planning horizon (see for example Anily and Fedegruen, 1990, Anily and Bramel, 2004 and Campbell and Savelsbergh, 2004). The number of customers and suppliers may vary, thus the structure can be one-to-one when only one supplier serves one customer, and one-to-many is the most common case where one supplier serves several customers while many-to-many is when several suppliers serve several customers. However, in the recent literature, the structure of many-to-one having several suppliers serving one customer also has been proposed. Routing component can be categorized into direct when only one customer per route is allowed. Meanwhile multiple refer to the case in which several customers are on the same route, while continuous is the case without central depot, like some of the maritime applications such as ship routing and inventory management problem.

Inventory policies establish the rules on how to replenish customers. There are two common policies that are applied in most of the literature: Maximum Level (ML) and Order-up-to-level (OU). Under the policy of ML, the replenishment level is flexible but limited to the capacity available at the customer site. Under the policy of OU, the replenishment level is triggered when the inventory level on hand falls below specified minimum level and then the quantity delivered is that to fill its inventory capacity. Meanwhile, inventory decisions determine how the inventory management is modeled. If the inventory is allowed to become negative, then backordering occurs and the corresponding demand is delivered to customers at a later stage. However if backorder is not allowed, the corresponding demand will be considered as lost sales. For both cases, the penalty may be applied for the stockout. In most cases the inventory is not allowed to be negative especially for the deterministic model.

Fleet composition and fleet size are the two additional criteria considered. The composition can be divided either homogeneous or heterogeneous while the number of vehicles available can be fixed or unconstrained.

Table 2.1: Classification on IRP

Criteria	**Possible options**		
Time horizon	Finite	Infinite	
Structure	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	Lost sales	Back-order	Non-negative
Fleet composition	Homogeneous	Heterogeneous	
Fleet size	Single	Multiple	Unconstrained

Source: Adapted from Andersson et. al. (2010)

The time at which the demand is known can be classified into several categories. If the demand of customers is known at the beginning of the planning horizon, the problem is deterministic. However, if the demand is unknown and based on some probability functions, it yields the Stochastic Inventory Routing Problem (SIRP). Dynamic SIRP arises when demand is not fully known in advance, but is gradually revealed over time, as opposed to what happens in a static context. In this case, one can still exploit its statistical distribution in the solution process, then yielding a Dynamic and Stochastic Inventory Routing Problem (DSIRP). Although both demand of SIRP and DSIRP are known in probabilistic sense, however in DSIRP, the customer's demand is gradually revealed over time, for an instance, at the end of each period, one must solve the problem repeatedly when the information becomes available.

2.3 Deterministic Inventory Routing Problem

Deterministic Inventory Routing Problem (DIRP), in which the customer's demand is known in advance, is discussed. The initial studies published on IRP were mostly extended from the standard Vehicle Routing Problem (VRP) in which the heuristics developed are extended to take inventory costs into consideration. As discussed in Section 2.2, IRP can be modelled in different ways based on the assumptions that are taken into account in the model. The discussion of the literature review in this section is focused on exact and heuristic algorithm respectively.

2.3.1 Exact Algorithm

In this subsection, we present a literature review emphasizing those studies that implemented the exact algorithms to solve their proposed model. Several exact algorithms such as Branch-and Cut, Branch-and-Price and Lagrangian Relaxation algorithms have been implemented.

Archetti et al. (2007) were the first proposed branch-and-cut algorithms for a single product and single vehicle IRP. They have introduced a special formulation for maximum order policy. The instances solved are 30 customers and 6-period horizon and 50 customers with 3-period horizon. The formulation was improved by Solyali and Sural (2008, 2011) who introduced customer replenishment strategy by incorporating shortest path network and uses a heuristic approach to get initial bound to branch-and-cut algorithm. This new formulation enables the authors to solve 15 customers with 12 periods, 25 customers with 9 periods and 60 customers with 3 periods.

Recently, Coelho and Laporte (2013c) extended the formulation proposed by Archetti et al. (2007) by including the multiple vehicles known as Multi-vehicle IRP (MIRP). The authors proposed a branch-and-cut algorithm for the exact solution of several classes of IRP. Several cases have been considered in their computational experiment namely MIRP solutions under the maximum level (ML) replenishment policy, MIRP solution with a homogeneous and heterogeneous fleet of vehicles, IRP with transshipment options and MIRP with additional consistency features. The computational experiments done on the benchmark instances and the computational results confirm the success of the proposed algorithm.

Coelho and Laporte (2013b) extended their work to propose branch-and-cut algorithm for solving multi product multi vehicle IRP (MMIRP) with deterministic demand and stockout cost is not allowed. In this paper, Coelho and Laporte (2013b) have implemented a solution of improvement algorithm after branch-and-cut identifies a new best solution. The purpose of solution improvement algorithm is to approximate the cost of a new solution resulting from the vertex removal and reinsertions. In this paper, the authors considered additional of two features namely the driver partial consistency and visiting space consistency. The driver partial consistency plays the role of increasing the quality of the solution provided by the IRP both to customers and suppliers in a multi-product environment. The results show that the visiting space helps in reducing the search space while providing meaningful solution. The computational experiments to test the efficiency of the algorithm for their proposed MMIRP model and MMIRP with the additional two consistency features are presented in this paper. The authors have proposed larger instances where the number of customers has increased to 50 and up to seven time periods.

In the most recent work, Desaulniers et al. (2015) works on a single supplier who produces a single product at each period over a finite horizon to fulfill the demand of a set of customers by using a fleet of homogeneous capacitated vehicles. Each customer has their inventory capacity and initial inventory. The authors introduced an innovative formulation for the IRP and developed a state-of-the-art branch-price-and-cut algorithm for solving their proposed IRP model. The developed algorithm integrated known and new families of valid inequalities, appending an adaptation of the well-known capacity inequalities, as well as an ad hoc labeling algorithm in order to solve the column generation subproblems. The computational results showed that their algorithm outperforms existing exact algorithms for instances with more than three vehicles. The authors proved that the proposed valid inequalities, branching decisions, and other speed up strategies are effective.

Chien *et al.* (1989) is amongst the first to simulate a multiple period planning model where the model is based on a single period approach. This is achieved by passing some information from one period to the next through inter-period inventory flow. The authors have formulated their problem as a mixed integer program and developed a Lagrangian based procedure to generate both good upper bounds and heuristic solutions. Since then many researchers have focused their modeling on a finite planning horizon.

Yu et al. (2008) solved a large-scale IRP that delivers a single product with split delivery and vehicle fleet size constraint. The problem is solved by using a Lagrangian relaxation method and it combines with the surrogate subgradient method. The solution of the model obtained by the Lagrangian relaxation method is used to construct a near-optimal solution of the IRP by solving a series of assignment problems. Numerical experiments show that the proposed hybrid approach can find a high quality near-optimal solution for the IRP with up to 200 customers and 10 periods in a reasonable computation time.

Bard and Nananukul (2010) proposed a branch-and-price (B&P) algorithm for solving the production, inventory, distribution, routing problem (PIDRP), a variant of IRP. The model of this problem had included a single production facility, a set of customers with time varying demand, a finite planning horizon, and a fleet of homogeneous vehicles. The aim of this study is to construct a production plan and delivery schedule that minimizes the total cost while ensuring that each customer's demand is met over the planning horizon. In this study, a new branching rule for dealing with an unstudied form of master problem degeneracy is introduced, while reducing the effects of symmetry and obtaining feasible solutions by combining a rounding heuristics and tabu search within B&P, and the use of column generation heuristics. The computational results indicated that the PIDRP instances with up to 50 customers and 8 time periods can be solved within 1 hour. The hybrid scheme performed better than CPLEX and standard branch and price alone.

2.3.2 Heuristic Algorithms

IRP is known to be NP-hard because the VRP reduces to TSP where the time horizon is one, the inventory costs are zero, the capacity of the vehicle is infinite, and all the retailers need to be served; hence it is unlikely that a polynomial time algorithm will be developed for its optimal solution. The largest instance that can be solved by exact algorithms consist of 50 customers with 7 periods (see Coelho and Laporte (2013b) and most researchers resort to heuristic or metaheuristic algorithm to solve large instances, which represents the real world problems. There are several heuristic / metaheuristic algorithms such as Tabu Search, Genetic Algorithm, Variable Neighborhood Search, Ant Colony Optimization (ACO) et cetera that have been developed widely to solve IRP. In this subsection, we present a literature review of those algorithms that implemented the heuristic / metaheuristic algorithms to solve their proposed IRP model.

Bertazzi et al. (1997) proposed a set of decomposition heuristics for the transportation of multi-product in multi-period with constant demand. In the first of phase of the algorithm, each destination is considered independently and direct shipping is solved using the algorithm of Speranza and Ukovich (1996). The second phase aggregates customers visited at the same frequency on the same route. Meanwhile, each set is considered separately and a heuristic procedure is used to determine an estimation of the minimum transportation cost to deliver the products to all the destinations of the given set in the third phase. Inventory cost will remain unchanged but reduction of the transportation may occur in third phase of the algorithm. The authors introduce the concept of split deliveries where the quantity of a product required at a destination can be served in different shipments, possibly with different frequencies. For simplicity, most multi-product models assume that each retailer requires only one type of product. The authors tested their proposed algorithms on a set of randomly generated problem instances. The computational results indicate the algorithms perform well on the instances but required more computational time.

Bertazzi et al. (2002) considered a multi-period distribution problem in which a set of products has to be delivered from a supplier to several retailers in a given time horizon and the demand of the retailers is known in advance (deterministic). The authors adopted the order up to a level inventory policy (S, s), where each retailer determines the maximum(S) and the minimum(s) levels of inventory of each product and the products have to be replenished before the minimum level is attained. The quantity of the product delivered is the amount such that the maximum level is reached at the retailer. The authors proposed a two-step heuristic algorithm. The first stage focuses on route construction algorithms. Meanwhile, the second stage attempts to improve the existing solution iteratively by performing simple swap operators that aim to remove or insert customers at different positions on the route. The authors compared the cost of the solution generated by the heuristic algorithm with the optimal cost of two intuitive policies. The first policy, referred to the case that visit all the retailers in each discrete time instant on the basis of an optimal route while the second policy referred to the case that visit the set of retailer in each delivery time instant on the basis of an optimal route and stockout is occurred if not served at given time. The obtained results showed that the heuristic algorithm always outperforms the optimal solution of the two intuitive policies

Since split delivery is one of the important components, in our model thus, in this subsection some of the papers that include the split delivery into their DIRP model are reviewed. Dror and Trudeau (1989) first introduced the split delivery VRP (SDVRP) by relaxing the constraint of the VRP that requires every customer to be served by only one vehicle. The authors showed that the relaxation increased the flexibility of distribution and could lead to important savings, both in the total distance traveled and in the number of vehicles used. The SDVRP remains NP hard despite this relaxation (Dror and Trudeau, 1990). Several authors (see for example Mjirda et al., 2014, Moin et al., (2011) and Yu et al., 2008) have extended the concept of split delivery in the multi-period IRP.

Moin et al. (2011) proposed an efficient hybrid genetic algorithm to solve the IRP in a many-to-one network which involves multi products where each supplier supplies different products in multi period scenario. The problem is to find the minimum cost to pick up the products from a set of geographically dispersed suppliers over a finite planning horizon to the assembly plant by using a fleet of capacitated homogeneous vehicles which are housed at a depot and the split pick-ups are allowed. The proposed hybrid genetic algorithm is based on the allocation-first-route-second strategy and takes both the inventory and the transportation costs (fixed and variable) into consideration. The computational experiments have been done on the data sets which were extended from the existing data sets to show the effectiveness of the proposed approach. Small, medium and large size problems are added to the existing data sets. With the increase of problem size, GA based algorithms performed relatively much better.

Mjirda et al. (2014) improved the results obtained by Moin et al. (2011) by proposing a two-phase Variable Neighborhood Search (VNS). The first phase develops an initial solution without considering the inventory but only focuses on minimizing the transportation cost by using VNS. While in the second phase the initial solution is iteratively improved to minimize both the inventory and transportation costs using Variable Neighborhood Descent (VND) and a VNS algorithm. For the part of inventory management in the second phase, Linear Programming (LP) formulations and a heuristic method (i.e. backward method to define the amount of products to deliver by each vehicle at each period in order to satisfy the demands of the preceding periods) which consider the priority of rules on suppliers and vehicles are developed to calculate the amount of products to collect from each supplier at each period during the planning horizon. The computational results showed that the proposed methods had given better results than the existing methods from the literature for both solution quality and the running time. Both Moin et al. (2011) and Mjirda et al. (2014) considered the many-to-one network, which is equivalent to one-to-many network under certain assumption.

Heuristic / metaheuristic algorithms have been widely used to solve different variants of IRP. Abdelmaguid (2004) studied the integrated inventory distribution problem (IIDP) in which they considered an environment such that the demand of each customer is relatively small compared to the vehicle capacity, and the customers are located closely such that a consolidated shipping strategy is appropriate. In their model, they take into consideration the components of inventory holding, backorder, and transportation costs. The author proposed a construction heuristic for the IIDP, called the approximate transportation costs heuristic (ATCH). However, this strategy can give poor solutions if the order quantities of the customer are not significantly less than the vehicle capacity. To alleviate this, Abdelmaguid and Dessouky (2006) introduced a genetic algorithm (GA) approach for improving the constructed solution that allows partial deliveries. In their GA construction phase, they implemented a randomized version of their previously developed construction heuristic to generate the initial random population. Then, two random neighbourhood search mechanisms, the crossover and mutation operations are developed in their GA improvement phase. The main concern in designing the mutation operator was to develop a suitable mechanism that allows for deliveries to customers to cover part of their demand requirements, which is referred as partial deliveries. The computational results show that GA outperform the results produced in their earlier paper, Abdelmaguid (2004).

Abdelmaguid et al. (2009) improved the results of Abdelmaguid and Dessouky (2006) by introducing a constructive improvement heuristic which is based on the idea of allocating single transportation cost estimates for each customer. Two subproblems, comparing inventory holding and backlogging decisions with these transportation cost estimates, are formulated and their solution methods are incorporated in the developed heuristic. An improvement heuristic is developed to overcome some of the limitations of the constructive heuristic. This improvement heuristic is based on the idea of exchanging delivery amounts in between periods to allow for partial fulfilments of demands and exploit associated reductions in costs.

Bard and Nananukul (2009) developed reactive tabu search to solve Production Inventory Distribution Routing Problem (PIDRP) which involved the integration of both production and distribution decision. In this paper, the authors considered a single production facility, a set of customers with time varying demand, limited capacity of inventory at customer's site, a finite horizon and a fleet of homogeneous vehicles for making the deliveries. The authors developed reactive tabu search to solve their proposed model. The authors applied the allocation model in the form of mixed integer program to find good feasible solutions as the starting points of tabu search. The allocation model is modified to attain lower bounds on the optimum. Computational experiments are tested on 90 benchmark instances with up to 200 customers and 20 periods and the results showed the improvements in all cases if compared with the existing greedy randomized adaptive search procedure (GRASP). In our algorithm of Stochastic Inventory Routing Problem (SIRP), we are inspired by the concept of the allocation model which was presented in this paper and modified it in order to solve our problem.

2.4 Stochastic Inventory Routing Problem

Stochastic Inventory Routing Problem (SIRP) is the problem in which the customer's demand is only known in probabilistic sense. Since the demand is uncertain, shortages may occur and normally a penalty is imposed whenever a customer runs out of stock. The penalty is usually modelled as a proportion of the unsatisfied demand. Unsatisfied demand is typically considered to be lost, that is, there is no backlogging. The main objective of SIRP is similar to DIRP where the total of inventory and transportation cost are the main concerns to be minimized but is written to embed the stochastic and unknown future parameters: the supplier must determine a distribution policy that maximizes its expected discounted value (revenue minus costs) over the planning horizon, which can be finite or infinite. The discussion in this section is based on the studies that consider their model in infinite and finite horizon.

First, we discuss some of the literature that considers their model over an infinite planning horizon. Kleywegt et al. (2002) tackled the problem in which a supplier supplied a single product to serve a set of customers using a fleet of capacitated homogeneous vehicles. In the paper, the authors considered each vehicle route served one customer only (i.e. direct deliveries). The customer's demand is uncertain and a penalty cost is taken into account if the stock out occurred. Backlogged is not allowed. The authors formulated the SIRP as a Markov decision problem (MDP) over an infinite horizon. They proposed approximate methods based on the dynamic programming in order to find good quality solutions with a reasonable computational effort. Kleywegt et al. (2004) extended the work where each vehicle services up to three customers. However, in the paper of Adelman (2004) there is no limit on the number of customers to be served in a route but restricted by maximal route duration and vehicle capacity.

The author studied the case in which the number of customers visited in every route is unbounded and a fleet of vehicles with an unlimited number of vehicles is available.

Qu et al. (1999) addressed a periodic policy for a multi-item joint replenishment problem in a stochastic setting with simultaneous decisions made on inventory and transportation policies. The authors proposed a heuristic decomposition method which applied a property of the combined problem to divide the model into subproblems, namely inventory and vehicle routing models. Each of the subproblems is solved using the methods from existing literature. The computational experiments were done on the divided groups that are based on the problem sizes of 15–50 items and the results showed that their method performed satisfactory on solving the problem.

Based on the literatures available, there are quite a number of articles that considered SIRP model with finite time horizon and this is closely related to our study. Federgruen and Zipkin (1984) are among the first who studied IRP by modifying the Vehicle Routing Problem of Fisher and Jaikumar (1981) to accommodate inventory and shortage costs where the customers' demands are assumed to be random variables. The problem decomposes into a nonlinear inventory allocation problem which determines the inventory and shortage costs and a Travelling Salesman Problem (TSP) for each vehicle considered to represent the transportation costs.

Federgruen et al. (1986) considered the problem in which each product have a fixed lifetime (perishable items) during which it can be consumed, otherwise if the period is exceeded then the product has to be discarded. Computational experiments are done on two approaches, namely combined approach and separate approach. In the combined approach, the authors adopted the solution approach of Federgruen and Zipkin (1984) but implemented a more complex subproblem by imposing extra constraints. The separate approach solved the allocations and routing decisions separately in the conventional way. The computational results showed that the combined approach produced less total costs compared to separate approach. However, the combined approach required more computational time.

Minkoff (1993) considered a dynamic and stochastic vehicle dispatching problem called the delivery dispatching problem. The author modeled the problem as a Markov decision process and adopted a decomposition heuristic approach to solve the problem. The heuristic solves a linear program to allocate joint transportation costs to individual customers and solves a dynamic program for each customer locally. In this paper, the author described how to compute bounds on the algorithm's performance, and several examples is applied on the algorithm with good results. However, algorithm is applied for relatively small problem instances only (up to 6-customers).

Bertazzi et al. (2013) studied the IRP in which a supplier has to serve a set of retailers and the maximum inventory level is defined for each retailer. The demand of the retailer is stochastic and has to be satisfied over a given time horizon. The inventory policy of this study is order-up-to-level where the quantity delivery to each retailer is such that its inventory level reaches the maximum level whenever the retailer is served. Inventory holding cost is applied whenever the inventory level is positive while the penalty cost is imposed when the inventory level is negative. Backlogged is not allowed in their problem. The objective of the study is to minimize the expected total cost which is given by the sum of the expected total inventory both at the supplier and retailers and penalty cost at the retailers as well as the expected routing cost. The authors proposed a hybrid rollout algorithm to solve their problem. The computational results showed that the proposed rollout algorithm provided significantly better solution than the one that obtained from the benchmark algorithm. The authors also implemented a branch-and-cut algorithm to solve their proposed mixed-integer linear programming model which is the deterministic counterpart of their problem (set the future demand equal to the average demand). The proposed approach was able to determine the optimal solution in reasonable time limit (within a time limit of 7000 seconds) in the majority of the considered instances otherwise the best obtained feasible solution is used.

Recently, Yu et al. (2012) presented a stochastic IRP with split delivery (SIRPSD), which implemented the service level to satisfy each customer's demand by limiting the possibility of the stockout within a given value and also the service level to each customer's warehouse measured in its overfilling probability. This paper studied the stochastic version of the deterministic one proposed by Yu et al. (2008). The authors proposed the transformation of stochastic components of a model of the SIRPSD into deterministic ones and used the Lagrangian relaxation to decompose the model into sub problem of inventory and routing. The partial linearization approach for the subproblem of inventory, the minimum cost flow for the subproblem of routing and the local search improvement of feasible solutions of the studied SIRPSD are proposed in this paper to solve their proposed model. The computational results showed that their proposed approach can obtain high quality solutions in a reasonable computational time.

2.5 Ant Colony Optimization

Swarm Intelligence (SI) is the discipline that manages the natural and artificial systems consist of many individuals that coordinate using decentralized control and self-organization. The properties of the typical swarm intelligence system are shown below (see Dorigo and Birattari, 2007):

- Consists of many individuals;
- The individuals are relatively homogeneous (i.e., they are either all identical or they belong to a few typologies);
- The interactions among the individuals are based on simple behavioral pattern that exploit only local information that the individuals exchange directly or via the environment (stigmergy);
- The interactions between individuals and with their environment will give the overall behavior of the system, (i.e. the group behavior self-organizes).

Swarm-based algorithms have recently arisen as a family of nature-inspired, population-based algorithms which are able to produce low cost, fast, and robust solutions to several complex problems (see Panigrahi et al., 2011). SI can therefore be defined as a relatively new branch of Artificial Intelligence that is used to model the collective behavior of social swarms in nature. Examples of systems that are represented by SI are ant colonies, honey bees, bird flocking, animal herding, bacteria growth, fish schooling and microbial intelligence. These agents (insects or swarm individuals) are interacting together with certain behavioral patterns in order to carry out the necessary task for their survival. Through the study on the behavioral pattern of those agents, it leads the researchers to develop the nature-inspired metaheuristics for solving their problems. Since the computational modeling of swarms was proposed, the number of research papers reporting the successful application of SI algorithms in several optimization tasks and research problems is increasing rapidly. SI principles have been implemented successfully in a variety of problem domains including function optimization problems, vehicle routings, scheduling, structural optimization, and image and data analysis (see Lim et al., 2009).

In this study, we modifying Ant Colony Optimization (ACO) which was initially proposed by Dorigo and coworkers (Dorigo(1992), Dorigo and Blum (2005), and Di Caro and Dorigo (1999)) to solve our proposed model. ACO is a metaheuristic method which implements artificial ants to find the solutions to combinatorial optimization. ACO is based on the behavior of ants and possesses enhanced abilities such as storing the memory regarding the past actions and passing the information to other ants. In fact, ants cannot hunt for food effectively if they work individually but in a group, ants possess the ability to solve complex problems and successfully obtain the food for their colony. Ants make use of chemical substances named pheromones to share the information regarding the distance of the path which they share with other ants. When an ant passes by a location, it will deposit the pheromones on that trail in order to allow other ants to follow. Each ant moves in a random pattern, but when the ant faces the pheromone trail then it has to decide whether to follow or not. If the ant chooses to follow the trails, then the ant's own pheromone reinforces the existing trail. Therefore, it will increase the probability of the next ant to follow the same path. Consequently, the more ants use that path, the more pheromones will be deposited and that path becomes more attractive for the subsequent ants. In addition, ants which use shorter route will return to their colonies sooner before other ants reach. This indeed will influence the selection probability for the next ant leaving the nest. Over time, as more ants completed the shorter route, it increases the pheromone accumulation on shorter path but longer path will be less reinforced. The natural evaporation of the pheromones also makes less attractive routes more difficult to detect and further decreases their use. However, the continued random selection of paths by individual ants helps the colony discover alternative routes and ensure successful navigation around obstacles that interrupt a route. Trail selection by ants is a pseudo-random proportional process and is an important element of the simulation algorithm of ACO (Dorigo and Gambardella, 1997).

Metaheuristic algorithms such as genetic algorithm (Sin et al, 2013), scatter search (Huacuja et al., 2012) and variable neighborhood search (Rasheed et al., 2014) have been applied to different type of combinatorial problems. ACO algorithms have been applied to many combinatorial optimization problems, and the first ACO algorithm was called the ant system (Dorigo et al. (1996)), which aimed to solve the Travelling Salesman Problem (TSP) with the goal to search the shortest round-trip to link a series of cities. In the later literature, more researches have the interest on applying ACO in solving their TSP model (see Dorigo and Gambardella (1997), Hlaing and Khine (2011) and Brezina and Čičková (2011)). The application of ACO included solving Vehicle Routing Problem (VRP), which aimed to build the routes by using a fleet of vehicles to deliver the products to a set of customers (see Bullnheimer et al. (1999), Bell and McMullen (2004), Chen and Ting (2006) and Yu et al. (2009)). However, recently several researchers have applied ACO to solve their proposed IRP model. In fact, those papers inspired us to modify classical ACO to solve our proposed IRP model that integrates both transportation and inventory. We modify ACO in term of not only focusing on solving the routing part but also can give beneficial information to the inventory updating mechanism to build better set of inventory. We present some literatures that employed ACO to solve their IRP model.

Huang and Lin (2010) proposed a modified ACO for solving the multi-item inventory routing problems in which the demand is stochastic and by choosing a delivery policy that minimizes the total costs. The algorithm was developed for the replenishment of the vending machine and the authors modified ACO algorithm which incorporates the stock out cost in the calculation of the pheromone values, which is not included in the conventional ACO. The nodes with high stock out costs are given higher priority even though the total transportation costs are higher than the other nodes. The test instances were constructed using the Solomon's (1987) 56 benchmark problems created for the vehicle routing problem with time windows. The results show that the modified ACO algorithm achieves highly significant improvements compared to the conventional ACO.

Calvete et al. (2011) is the first to study a bilevel model in the context of hierarchical production-distribution (PD) planning. In this problem, a distribution company, which is the leader of the hierarchical process, controls the allocation of retailers to each depot and the routes which serve them. The manufacturing company, the follower of the hierarchical process will decide which manufacturing plants will produce the orders received by the depot. The authors developed ACO algorithm to solve the bilevel model in which ants are used to construct the routes of feasible solutions for the associated multi-depot vehicle routing problem (MDVRP). The computational experiment is carried out to analyze the performance of the algorithm. Since the bilevel model is first time proposed, there is no data for comparison purposes. The computational time is reasonable, taking into account the problem sizes.

In the most recent work of Tatsis et al. (2013), they developed a mixed integer mathematical model in which a fleet of capacitated homogeneous vehicle is used to deliver distinct products from multi suppliers to a retailer to meet the demand in each period over the planning horizon. However, backlogging is allowed in this study. The ant based optimization algorithm is applied to solve the corresponding vehicle routing problem. The objective of this study is to find the best compromise between the transportation, inventory and backlogging cost. Preliminary results show that the solution gaps between the algorithm and CPLEX solutions is kept reasonably low values and offered prospective for further improvement.

2.6 Summary of the chapter

This chapter has reviewed the existing works in the literature on several variants of inventory routing problem and discussed the literature review regarding ACO. In this section, the review summarized and the research's direction of this study is presented.

2.6.1 Summary of literature review

The literature reviews in this chapter are focused on the variation of IRP problems which have been studied as well as important findings and well-known methods that have been implemented to solve their proposed problem. The IRP model can be variants based on the classification which have been mentioned in Section 2.2. However, the type of the demand for the customers can be a crucial factor to decide the model of IRP which we desire to explore. As mentioned in ection 2.2, the customer's demand generally can be divided into deterministic and stochastic. In this study, we tackle both types of customer's demand. ACO is a well-known metaheuristic which has been used widely to solve variants of combinatorial problems and also those research related to vehicle routing. Initially, ACO is developed to solve TSP problem, and then later many of the researchers have the interest to apply ACO to solve for their model of VRP or the research related to vehicle routing. In the most recent work, few researchers also have interested to use ACO to solve for their IRP model. Through the body of literature reviewed above, it can be concluded that there are more researches to be done in order to find a good mechanism which can integrate both inventory and transportation well in order to attain the minimum cost.

2.6.2 Research Direction

To the best of our knowledge, majority of the IRP problems which had been studied in the literature are focused on non-split delivery. However, split delivery can be beneficial in term of transportation savings as the utilization of the vehicles can be maximized and hence the vehicle cost can be reduced. This drives us to model our IRP problem which allows for the split delivery.

Based on the literature many ACO algorithms have been applied to solve the vehicle~~s~~ routing problem and only a handful of researchers are working on IRP model. This inspires us to modify ACO algorithms which are able to balance between the inventory and transportation to solve our proposed IRP model. We enhance the algorithm further by incorporating some information regarding the level of inventory and this information is useful for the inventory updating mechanism so as to build better set of inventories.

CHAPTER 3: MODIFIED ANT COLONY OPTIMIZATION

This chapter presents the newly modified Ant Colony Optimization (ACO) to solve multi products multi periods Inventory Routing Problem (IRP) in which split delivery is allowed. The chapter begins with the introduction and followed by the mathematical formulation. Meanwhile, the description of the developed algorithm and the improved version of the developed algorithm will be discussed in this chapter as well. The chapter also discusses the characteristics of the data sets and the related computational results as well as the discussion of the obtained results. The sensitivity of the parameters α and β which control the influence of the pheromone value allocated on arc (i, j) is also presented. Finally, the chapter ends with the summary of the chapter.

3.1 Introduction

In the literature of the previous chapter, various models of IRPs have been proposed and solved by their respective developed methods. In this study, one of the objectives is to develop a new metaheuristic method to solve IRP. The main aim of inventory routing problem is to minimize the corresponding related costs by balancing between inventory and transportation costs.

The model that is considered in this study is extended from the formulation proposed by Yu et al. (2008) to incorporate multi products. We consider a network consisting of a warehouse that supplies multi products to a geographically dispersed customers and the product are transported by a fleet of homogeneous vehicles. We assumed that the customer's demand must be met on time and we allow the customers to be served by more than one vehicle (split delivery). Figure 3.1 illustrated inventory routing problem with split delivery.

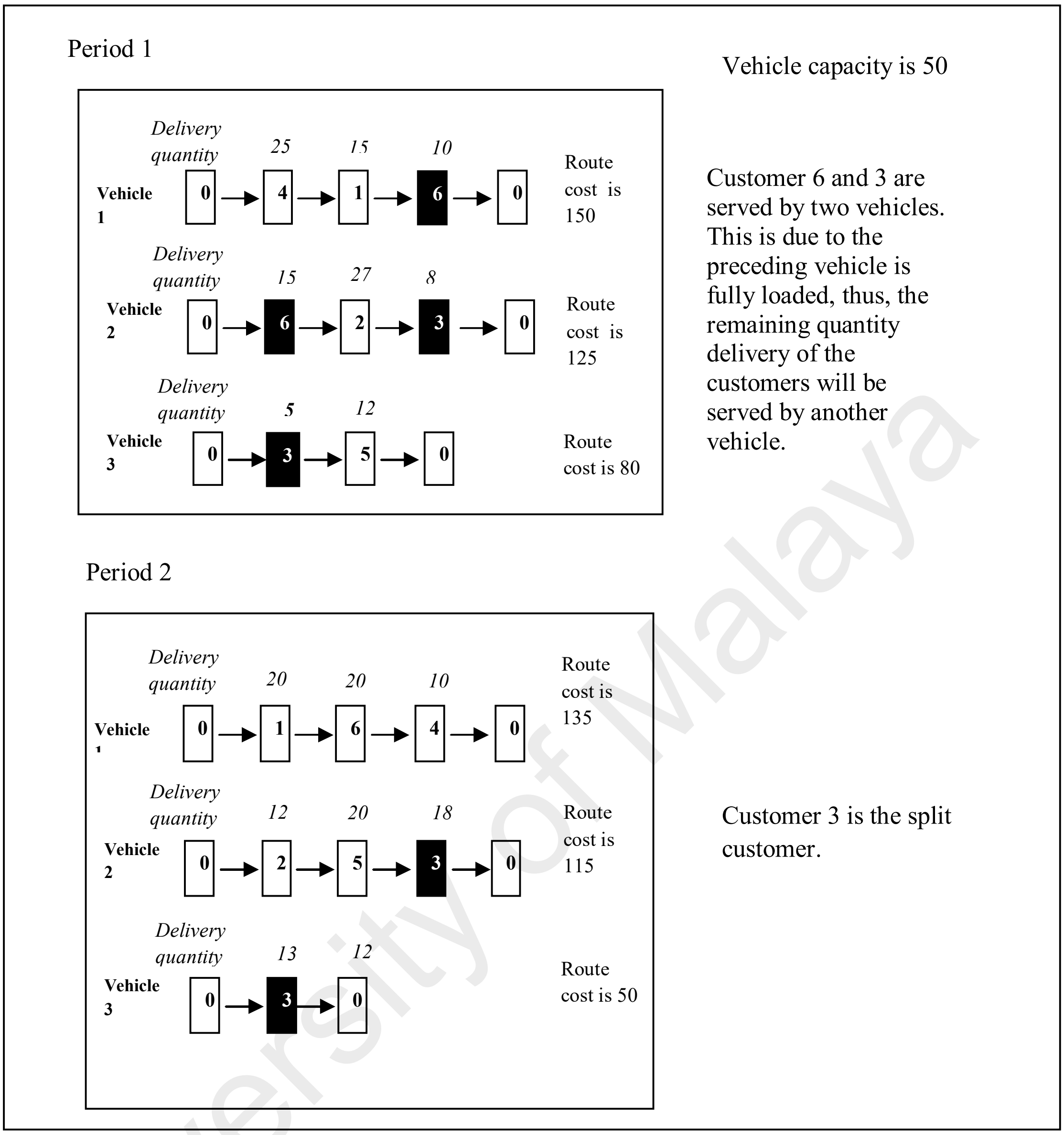

Figure 3.1: Inventory routing problem with split delivery

In this study, we focus on modifying ACO to solve our proposed model. ACO was initially proposed by Dorigo and co-workers (Dorigo, 1992, Dorigo and Blum, 2005, and Di Caro and Dorigo, 1999) and was inspired by the food-foraging behavior of ant which tends to find the minimum path between the food source and the colony by storing the information in the pheromones trails in order to lead more ants to use the minimum path.

The main contribution of this chapter can be summarized as follows:

1. We modify the algorithm by incorporating the inventory component in the global updating scheme that not only calculates the pheromone along the trail but identifies a set of feasible neighbors making use of the attractions on the nodes which differs from the classical ACO.
2. We develop a new heuristic called swap which aiming of merging the split customers in order to obtain the savings in term of transportation cost.
3. The sensitivity analysis is done on the parameters in order to choose the best combination of α and β which give better results for the proposed model. The sensitivity analysis is important as the parameter of α and β will influence the pheromone value allocated on arc (i, j) and these parameters will affect the selection of the next arc (customer).

3.2 Model Formulation

In this study, we consider a one-to-many network in which a fleet of homogeneous vehicle transports multi products from a warehouse or depot to a set of geographically dispersed customers in a finite planning horizon. The following assumptions are made in this model:

- The fleet of homogenous vehicles with limited capacity is available at the warehouse.
- Customers can be served by more than one vehicle (split delivery is allowed).
- Each customer requests a distinct product and the demand for the product is known in advance but may vary between different periods.
- The holding cost per unit item per unit time is incurred at the customer sites but not incurred at the warehouse. The holding cost does not vary throughout the planning horizon.
- The demand must be met on time and backordering or backlogging is not allowed.

The problem is modelled as a mixed integer programming problem and the following notation is used in the model:

Indices

$t = 1, 2, \dots, T$ period index

$W = 0$ warehouse/depot

$S = 1, 2, \dots, N$ a set of customers where customer *i* demands product *i* only

Parameters

C vehicles capacity (assumed to be equal for all the vehicles).

F fixed vehicle cost per trip (assumed to be the same for all periods)

V travel cost per unit distance

M the number of vehicles and it is assumed to be ∞ (unlimited)

c_{ij} travel distance between customer *i* and *j* where $c_{ij} = c_{ji}$ and the triangle inequality, $c_{ik} + c_{kj} \geq c_{ij,}$ holds for any different *i*, *j*, and *k* with $i \neq j$, $k \neq i$ and $k \neq j$

h_i inventory carrying cost at the customer for product *i* per unit product per unit time

d_{it} demand of customer i in period t

Variables

a_{it} delivery quantity to customer i in period t

I_{it} inventory level of product i at the customer i 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

The model for our inventory routing problem is given as below:

$$Z = \min \underbrace{\sum_{t=1}^{T}\sum_{i=1}^{N} h_i I_{it}}_{\mathrm{I}} + \underbrace{V\left(\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{i=0}^{N} c_{ij} x_{ijt} + \sum_{t=1}^{T}\sum_{i=1}^{N} c_{i0} x_{i0t}\right)}_{\mathrm{II}} + \underbrace{F\sum_{t=1}^{T}\sum_{i=1}^{N} x_{0it}}_{\mathrm{III}} \tag{3.1}$$

subject to

$$I_{it} = I_{i,t-1} + a_{it} - d_{it},\ i = 1, 2, \dots, N,\ t = 1, 2, \dots, T \tag{3.2}$$

$$\sum_{\substack{j=0 \\ i \neq j}}^{N} q_{ijt} + a_{it} = \sum_{\substack{j=0 \\ i \neq j}}^{N} q_{jit},\ i = 1, 2, \dots, N,\ t = 1, 2, \dots, T \tag{3.3}$$

$$\sum_{i=1}^{N} q_{0it} = \sum_{i=1}^{N} a_{it},\ t = 1, 2, \dots, T \tag{3.4}$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{N} x_{ijt} = \sum_{\substack{i=0 \\ i \neq j}}^{N} x_{jit},\ j = 0, 1, 2, \dots, N,\ t = 1, 2, \dots, T \tag{3.5}$$

$$I_{it} \geq 0,\ i = 1, 2, \dots, N,\ t = 1, 2, \dots, T \tag{3.6}$$

$$a_{it} \geq 0,\ i = 1, 2, \dots, N,\ t = 1, 2, \dots, T \tag{3.7}$$

$$q_{ijt} \geq 0,\ i = 0, 1, 2, \dots, N,\ j = 0, 1, 2, \dots, N, j \neq i,\ t = 1, 2, \dots, T \tag{3.8}$$

$$q_{ijt} \leq C x_{ijt},\ i = 0, 1, 2, \dots, N,\ j = 0, 1, 2, \dots, N, j \neq i,\ t = 1, 2, \dots, T \tag{3.9}$$

$$x_{ijt} \in \{0,1\},\ i = 1, 2, \dots, N,\ j = 1, 2, \dots, N,\ t = 1, 2, \dots, T \tag{3.10}$$

$$x_{0jt} \geq 0, \text{ and integer},\ j = 1, 2, \dots, N,\ t = 1, 2, \dots, T \tag{3.11}$$

The objective function (3.1) includes the inventory costs (I), the transportation costs (II) and vehicle fixed cost (III). Constraint (3.2) is the inventory balance equation for each product at the warehouse while constraint (3.3) is the product flow conservation equations, to ensure that the flow balance at each customer and eliminating all subtours. Constraint (3.4) assures the collection of accumulative delivery quantity at the warehouse (split delivery). Constraint (3.5) ensures that the number of vehicles leaving the warehouse equals to the number of vehicles returning to warehouse. Constraint (3.6) assures that the demand at the warehouse is completely fulfilled without backorder. Constraint (3.9) guarantees that the vehicle capacity and gives the logical relationship between q_{ijt} and x_{ijt} which allows split delivery. This formulation is used to determine the lower and upper bounds for each data set using CPLEX 12.4.

3.3 Modified Ant Colony Optimization (ACO)

Ant Colony Optimization inspired by the nature behavior of ants finding the shortest path between their colony and a source of food is modified to solve the proposed model of IRP. The information collected by ants during the searching process is stored in pheromone trails. Hence, when an ant has built a solution, the ant deposits a certain amount of pheromone proportionally (the information about the goodness of the solution) on the pheromone trails of the connection it used. The pheromone information directs search of the following ants while exploring the different path. The higher density of pheromones on an arc leads to attracting more ants to the arc. Therefore, appropriate formulation associated to the model for updating pheromones trail (equation 3.15 and 3.16) are very crucial. This is due to the reason that the greater amounts of pheromone it deposits on the arcs tend to provide a shorter path (the minimum cost).

In the conventional ACO, only the transportation cost is taken into account for the global pheromones updating. In IRP, the inventory cost as well as the transportation cost are equally important components and thus the two components are added in the global updating in order to balance the transportation and inventory cost. The details of global pheromone updating are discussed in subsection 3.3.4. The procedure for ACO can be divided into three main steps: the route construction, a local pheromone-update rule and a global pheromone-update rule. These steps are described in detail in the following subsections and Figure 3.2 outlines the algorithm.

3.3.1 Initial solution

We construct the initial solution by having all the demands met in every period. In this study we adopt a simple Nearest Neighbor algorithm (NN) and the algorithm is modified to allow split delivery. The vehicle starts at the depot and repeatedly assigns the nearest customer (in terms of distance) until the capacity of the vehicle is fully occupied. Then, a new vehicle is initiated and the process continues until all customers have been assigned or visited. The total distance obtained by NN is embedded to initialize the τ_0, the initial pheromone in the local pheromone updating.

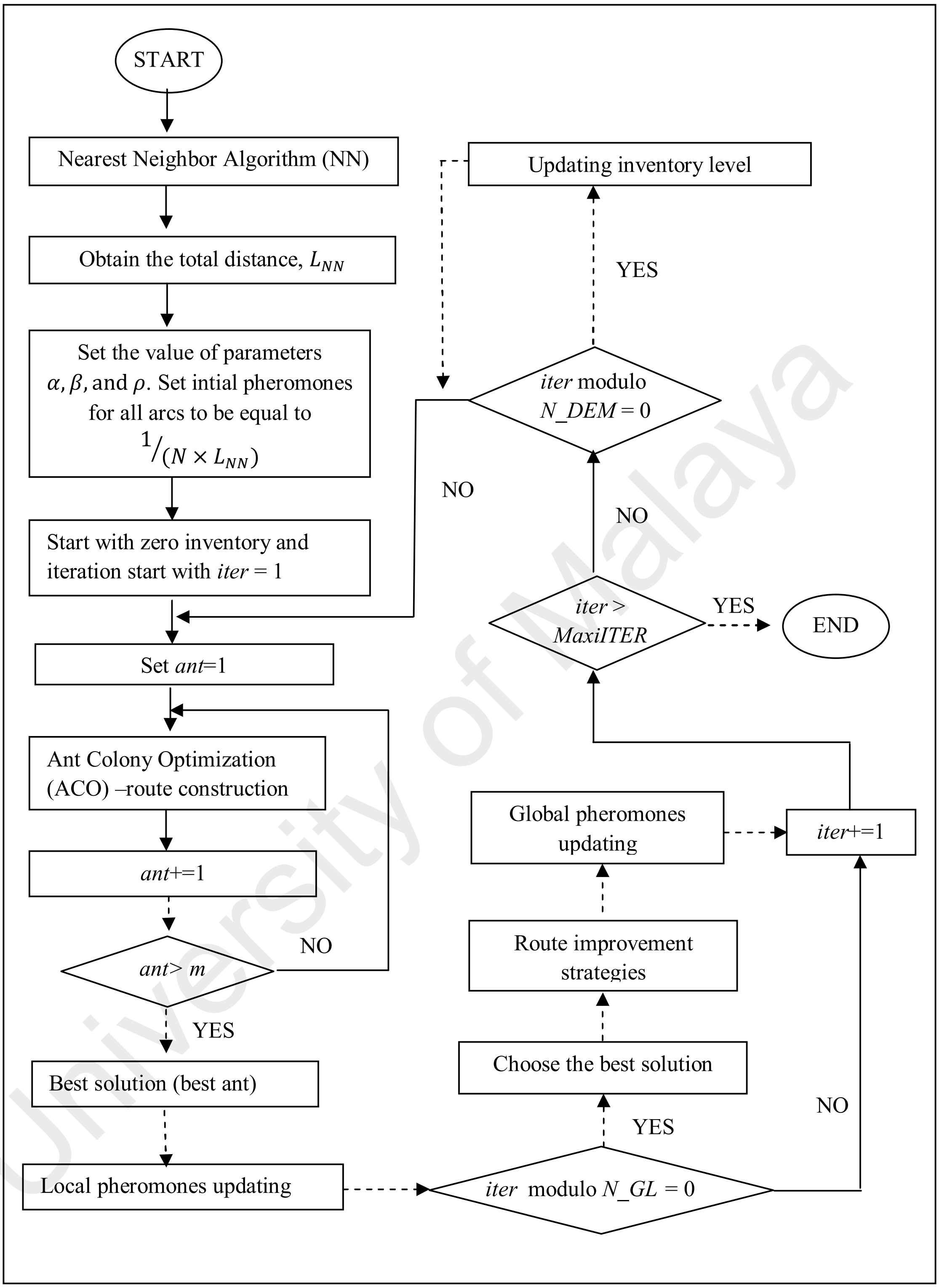

Figure 3.2: Algorithm of the developed method

3.3.2 Route construction for ACO

The route construction begins by setting the value of all the parameters α, β, τ_0, q_0 and ρ. Parameters α and β control the influence of the pheromone value allocated on arc (i, j) and the desirability of arc (i, j) respectively whilst q_0 is a predefined real number where $0 \leq q_0 \leq 1$ and ρ is the rate of pheromone evaporations. Note that the value of τ_0, the initial value of pheromones for each arc is obtained from the total distance of the initial solution. Starting from the depot (warehouse) each ant utilizes equation (3.12) to select the next customer to be visited. Ants tend to be attracted to the arc which consists of higher density of pheromones. From equation (3.12), if q is less than the predefined parameter q_0, then the next arc chosen is the arc with the highest attraction. Otherwise, the next arc is chosen using the biased Roulette Method with the state transition probability p_{ij} given by equation (3.14).

$$j = \begin{cases} \max\limits_{j \in \Omega_i} \left\{ Att_{i,j} \right\} & \text{if} \quad q \leq q_0 \\ p_{ij} & \text{otherwise} \end{cases} \tag{3.12}$$

where $Att_{ij} = \left(\tau_{ij}\right)^{\alpha} \left(\eta_{ij}\right)^{\beta}$, (3.13)

$$p_{ij} = \begin{cases} \dfrac{\left(\tau_{ij}\right)^{\alpha} \left(\eta_{ij}\right)^{\beta}}{\sum\limits_{k \in \Omega_i} \left(\tau_{ik}\right)^{\alpha} \left(\eta_{ik}\right)^{\beta}} & \forall j \in \Omega_i \\ 0 & \forall j \notin \Omega_i \end{cases}, \tag{3.14}$$

τ_{ij} is the amount of pheromone deposited on arc(i, j) and η_{ij} is inversely proportional to the length of arc (i, j), c_{ij}. The set of unvisited customers for ant i is denoted by Ω_i.

3.3.3 The local pheromone-updating rule

Local updating is used to reduce the amount of pheromone on all the visited arcs in order to simulate the natural evaporation of pheromone and it is intended to avoid a very strong arc being chosen by all ants. After a predefined number of ants, m had completed their solutions, the best among the built solutions is chosen and the pheromone on each arc is updated using

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\tau_0 \tag{3.15}$$

where ρ represents the rate of pheromone evaporation.

3.3.4 The global pheromone-updating rule

After a predefined number of iterations, the ACO updates the pheromone allocation on the arcs of the current optimum route γ^{gl}. The global pheromone-updating rule resets the ant colony's situation to a better starting point and encourages the use of shorter routes. It also increases the probability that future routes make use the arcs contained in the best solutions. In the classical ACO only the transportation cost is taken into account in the global updating. Since the IRP tries to find a balance between the transportation and inventory cost, it is natural to incorporate the inventory holding cost in the formulation. The global update rule is enhanced as follows:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \frac{\rho}{J_{\gamma^{gl}}}, \quad (i,j) \in \gamma^{gl} \tag{3.16}$$

where $J_{\gamma^{gl}}$ is the weight of the best solution found where it incorporates the inventory element as well as the variable transportation costs. The term $J_{\gamma^{gl}}$ is given by

$$J_{\gamma^{gl}} = \sum_{t=1}^{T}\sum_{i=1}^{N} h_i I_{it} + V\left(\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{i=0}^{N} c_{ij} x_{ijt} + \sum_{t=1}^{T}\sum_{i=1}^{N} c_{i0} x_{i0t}\right) \tag{3.17}$$

where the first component defines the total inventory costs whilst the second component gives the total transportation cost.

3.3.5 Route improvement strategies

The routes can be further improved by adding route improvement strategies in the route construction procedure. In this study, we implement two local searches consisting of inter route swap and intra route 2–*opt* in order to improve the solution built by ACO.

3.3.5.1 Swap for split customer

The first local search is the swap algorithm focusing on the split customers and they comprise of a transfer to the selected vehicle or a swap between different vehicles. Starting from the last vehicle, the split customer is identified and we try to merge to the current selected vehicle if the respective vehicle capacity is not violated. If this fails, then the swap with the other customers from the preceding vehicle or to the current selected vehicle that results in the least transportation cost is carried out. If none of the swap provides an improvement in the objective value than the solution built by ACO, the route remains unchanged. The process continues until all vehicles in every period have been examined. The aim of this method is to eliminate the split customers (merge as many as possible) if the merge improves the objective value. Figure 3.3 and Figure 3.4 illustrated the procedure of swap.

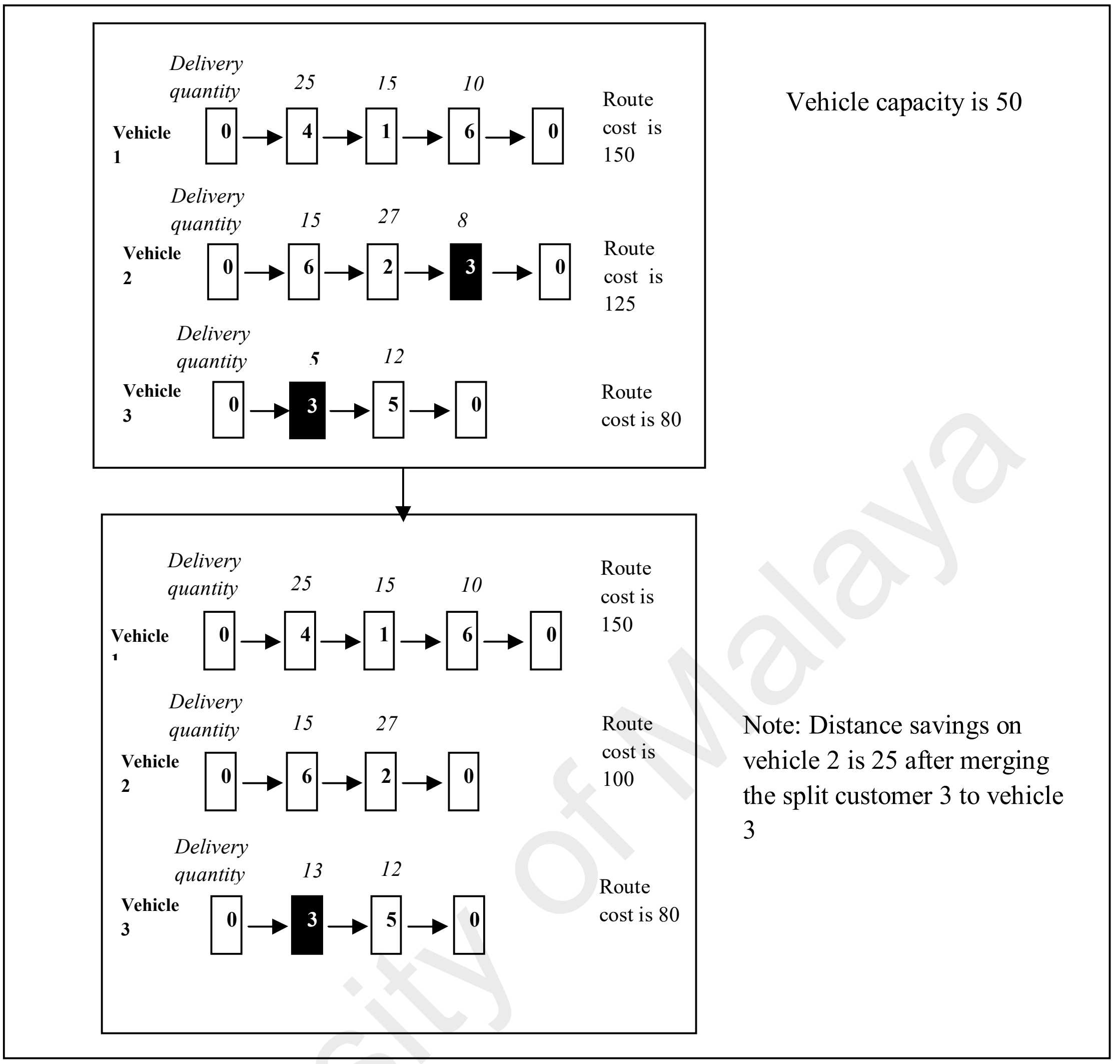

Figure 3.3: Swap procedure if the split customer is able to merge directly to the selected vehicle

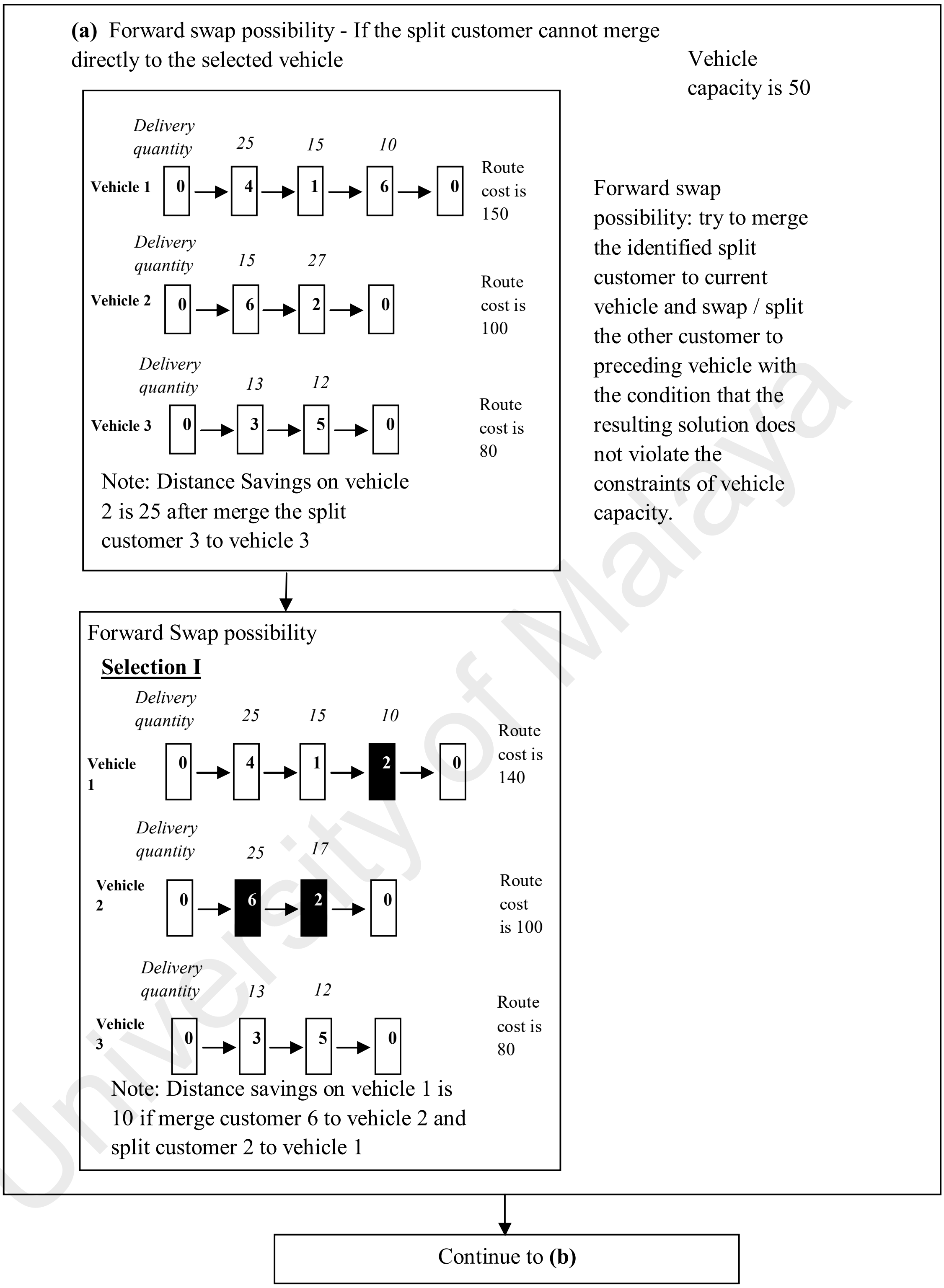

Figure 3.4: Swap procedure if the split customer cannot be merged directly: (a) Forward Possibility, (b) Backward Possibility

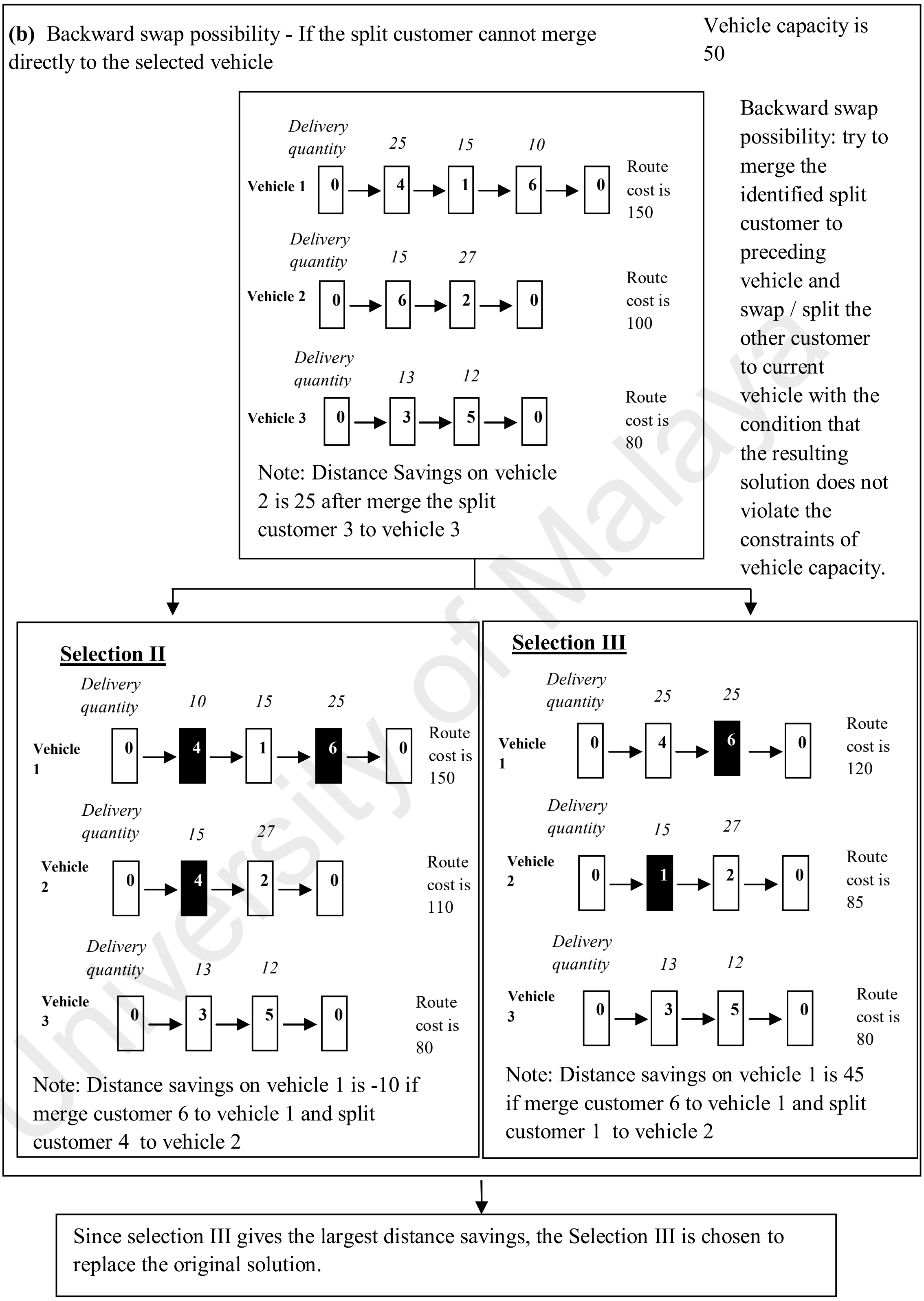

Figure 3.4: continued

3.3.5.2 2-*opt*

2-*opt* (Lin, 1965) heuristic is an intra-route optimization procedure. This heuristic is testing on all possible pairwise exchange within a vehicle instead of between vehicles to see if an overall improvement in the objective function can be obtained. The current solution is replaced if the improved solution is better.

3.3.6 Updating the inventory level

The inventory updating mechanism is applied after certain predefined number of iterations has been completed. Figure 3.5 illustrates the process of the inventory updating. First we randomly select the period to be moved. The number of customers to be moved is limited by some predefined maximum number of moves allowed, *N_moveTime*. The available customers on a selected period p are those with positive delivery quantities ($q_{ijp} > 0$) and the inventory has not been updated yet (has not received from period $p+1$). This extra constraint is to ensure that the inventory holding cost is not excessive. Additional criterion imposed is that the inventory of the preceding period ($p-1$) has not been updated in the present iteration. The customers who will be selected to update the inventory are those with the least inventory cost. We note that when updating the inventory, there is no restriction imposed except for the vehicle capacity constraint and may result in an increase in the number of vehicles. Figure 3.5 shows the algorithm of updating the inventory level for customers.

The following definitions are introduced for the procedure of updating inventory level:

N_moveData the maximum number of moves to be allowed for each data

N_moveTime the maximum number of moves to be allowed per time

temp_move the current number of moves

sum_move the current accumulative moves that have been done *cur_move* the number of moves generated by random number which is no more than *N_moveTime* per time.

Step 1: Check the availability of customers on all period. If none of the period consists of available customers, go to **Step 9**. Otherwise, go to **Step 2**.
Step 2: Randomly select a period, *p,* with the condition that there is at least one available customer. Go to **Step 3**.
Step 3: Randomly select the number of moves (cannot exceed *N_moveTime*), *cur_move*:
If (*cur_move* + *sum_move*) <= *N_moveData*
real_move = *cur_move*
else
real_move = *N_moveData* – *sum_move*
Set *temp_move* = 0.
Go to **Step 4**.
Step 4: Select an available customer from period *p*, who will give the least inventory cost.
Move all the delivery quantity on period *p* to period *p* – 1.
temp_move++.
Go to **Step 5**.
Step 5: Update the availability of the customer on period *p*.
If (*temp_move* < *real_move*)
Go to **Step 6**.
Else
Go to **Step 7**.
Step 6: Check if there is any available customer on period *p*.
If yes, go to **Step 4**. Otherwise, go to **Step 7**.
Step 7: *sum_move* += *temp_move.* Go to **Step 8**.
Step 8: Update the inventory level and inventory cost for each customer on each period.
Step 9: Select the set of inventory level that had been built for the current best solution to continue with the routing.

Figure 3.5: Algorithm of updating inventory level

3.4 Enhanced Modified Ant Colony Optimization (ACO2)

In ACO, the route improvement strategies are focused only on merging/swapping the split customer between the selected vehicles and then $2-opt$ (Lin, 1965) is applied as intra route optimization. Hence, we found out that we need to enhance the algorithm by adding $2\text{-}opt^*$ (Potvin and Rousseau, 1995) heuristic as inter route optimization procedure in the route improvement strategies. This improvement strategies; 2-opt^*(Potvin and Rousseau, 1995) is applied after the swap for split customer have been done. The purpose of this strategy is to test on all possible pairwise exchange between vehicles to see if an overall improvement in the objective function can be attained. The heuristic calculates the distances for all pairwise permutations and compared those distance with the current solution. If any of these solutions is found to improve the objective function, then it replaces the current solution. Hence, ACO2 is not only tried to improve the route within vehicle but also between vehicles.

3.5 Data sets

In this section, we will explain the experimental design used for evaluating the efficiency of the developed algorithm. The algorithm is tested on 12, 20, 50 and 100 customers, and combination with different number of periods, 5, 10, 14 and 21. The coordinates for each customer is generated randomly in the square of 100×100. The coordinates of each customer for the 20 customer instance comprises the existing 12 customer instance with additional 8 newly randomly generated coordinates. The same procedure is used to create the 50 and 100 customer instances. Figure 3.6 illustrates the distribution of the data sets. The holding cost for each customer lies between 0 and 10 while the demand for each customer is generated randomly between 0 and 50. The vehicle capacity is fixed at 100.

3.6 Results and discussion

The problem is formulated as a mixed integer programming problem and we let CPLEX 12.4 run for a limited time 9000 seconds (reached the limitation of memory) to get the lower bound and upper bound (the best integer solution) for each instance considered. All problem instances do not reach the optimal solution since the upper bound is different from the lower bound. The algorithms were written in C++ language by using Microsoft Visual studio 2008. The results of this study are compared with the upper bound (UB) which is generated from CPLEX 12.4. All the computations were performed on a 3.10 GHz processor with 8 GB of RAM.

3.6.1 Sensitivity Analysis

Since the two most important parameters in any ACO are α and β that control the decision policy in the selection of customers (see equation 3.13), we conduct the computational experiments to test on the different combination of parameters $\alpha = [1, 2, 3]$ and $\beta = [1, 2, 3, 4, 5]$ in order to determine the appropriate values of α and β. The performance of the modified ACO is measured for each data set, and averaged over 5 runs. Table 3.1 shows the mean and standard deviation over 5 runs of the different combinations of parameters α and β and parameter α is represented by alphabet A (A1, A2 and A3) while B (B1, B2, B3, B4 and B5) refers to parameter β. The results show that the combination of $(\alpha, \beta) = (1,5)$ gives the best average in the larger data set. However, the best combination of α and β for S12 and S20 are $(\alpha, \beta) = (2,1)$ and $(\alpha, \beta) = (2,3)$ respectively. Extra computations are done to compare between using existing parameters $(\alpha, \beta) = (1,5)$ and the best parameter settings for instances of S12 and S20 and the results are tabulated in Table 3.2 (S12) and Table 3.3 (S20). The improvement in the mean between $(\alpha, \beta) = (1,5)$ and $(\alpha, \beta) = (2,1)$ of S12 and $(\alpha, \beta) = (2,3)$ of S20 are small which is less than 1.7%. Therefore we set the parameter values for $(\alpha, \beta) = (1,5)$ for all data sets and the results are tabulated in Table 3.4.

3.6.2 Comparison of Modified ACO and Enhanced Modified ACO

The parameters for both versions of ACO (ACO and ACO2) are set as follows: $\alpha = 1.0,\ \beta = 5.0,\ q_0 = 0.9, \rho = 0.1,\ \tau_0 = 1/(N \times L_{nn})$, where L_{nn} is the total distance obtained from nearest neighbor algorithm. The algorithm is ran for 5000 iterations and each of the iterations consists of 25 ants to build a solution. *N_moveData* is determined by $\{\frac{1}{12} \times T \times N\}$ while *N_moveTime* is set to be equal to 3.

We performed 10 runs for each data set. Table 3.4 shows the results of ACO which only applies Swap and 2-*opt* as local search and ACO2 which includes 2-*opt** as inter route optimization procedure. Table 3.4 presents the best total costs, the number of vehicles, the CPU time, the lower bound and the upper bound (best integer solutions) which are obtained from CPLEX. From Table 3.4, we observed that the gaps which are calculated as the ratio of the difference between the lower bound and the upper bound to the lower bound, for all the solutions are greater than 10%. This ratio increases as the periods and the number of customers increase. Thus, it is hard to justify the quality of the lower bound obtained by CPLEX. This may be due to the lower bound is really loose or the upper bound is rather poor.

From the results shown in Table 3.4 for ACO as well as ACO2, we note that the total costs of the data sets with 50 and 100 customers are less than the upper bound which means the algorithm is able to obtain better results when compared with the upper bound. However, ACO gives less than 9 percent gaps for the small and medium instances with 12 customers and 20 customers. Meanwhile, ACO2 performs equally well for both small and medium instances and produced the gaps between the results and the best integer solutions with less than 5 percent. If comparing both ACO and ACO2, we found out that ACO2 perform better than ACO. Table 3.5 presents the best total costs, the distance cost, the number of vehicles and the inventory cost of the best solution between ACO and ACO2.

Table 3.6 show the average and standard deviation of the total costs and CPU running time over 10 runs for both ACO and ACO2. From Table 3.6, we note that ACO2 gives less standard deviation than ACO, which means that ACO2 gives better results in terms of solution quality when compared with ACO.

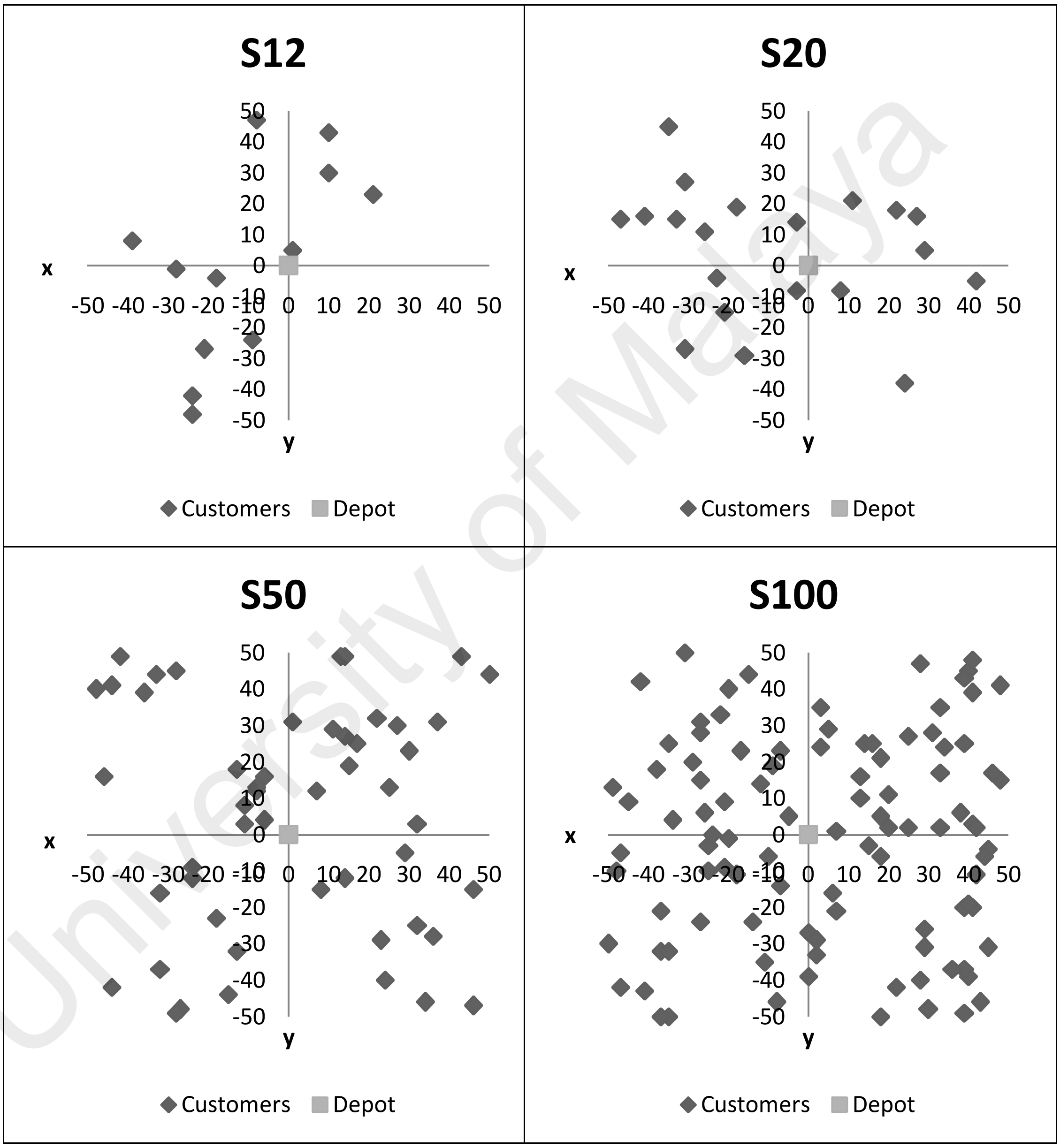

Figure 3.6: Distribution of the data sets

Table 3.1: The results of mean and standard deviation for different parameters settings over 5 runs.

Combination of parameters	MEAN				STDEV			
	S12T14	**S20T21**	**S50T21**	**S100T14**	**S12T14**	**S20T21**	**S50T21**	**S100T14**
A1B1	6501.44	14397.58	37401.46	45549.38	4.880	49.947	214.086	198.486
A1B2	6480.71	14477.42	36978.60	44780.64	23.131	38.739	76.765	81.299
A1B3	6494.63	14534.18	36825.98	44494.96	26.374	46.013	48.109	57.527
A1B4	6509.66	14545.96	36788.92	44314.14	9.268	20.261	125.782	72.611
A1B5	6502.01	14554.66	36729.76	44216.72	22.996	45.268	77.270	43.506
A2B1	6394.06	14383.48	37162.56	45341.20	20.631	85.372	225.680	341.760
A2B2	6502.29	14336.86	37038.28	45032.00	16.981	46.222	89.375	224.243
A2B3	6505.26	14309.66	36789.50	44693.04	12.281	56.123	180.622	110.700
A2B4	6503.11	14372.46	36919.64	44400.80	13.745	51.106	109.498	155.846
A2B5	6506.18	14545.30	36820.14	44282.44	5.637	70.642	60.905	100.351
A3B1	6410.164	14437.34	37369.62	45386.00	15.728	96.595	139.770	289.770
A3B2	6461.848	14349.34	37116.32	45304.10	42.698	58.585	117.479	117.729
A3B3	6489.428	14357.42	37050.30	44631.50	24.138	96.583	83.243	317.133
A3B4	6502.286	14315.22	36930.08	44531.80	9.043	26.120	106.103	186.044
A3B5	6500.422	14411.86	36842.48	44561.46	30.014	52.133	70.393	47.792

Note: Alphabet A refers to the parameter of alpha while alphabet B refers to the parameter of beta for the equation (3.13).

Table 3.2: Comparison of the results for S12 with two different parameters

Data Sets	UB (Best Integer)	A1B5				A2B1			
		Best costs	**Gaps ** (%)**	**Mean**	**STDEV**	**Best costs**	**Gaps ** (%)**	**Mean**	**STDEV**
S12T5	2231.96	2290.38	2.62	2296.98	7.682	2254.04	0.99	2272.04	9.976
S12T10	4305.33	4453.58	3.44	4512.78	30.855	4409.91	2.43	4459.50	24.550
S12T14	6196.35	6462.09	4.29	6505.47	17.702	6361.24	2.66	6403.95	17.333

Gaps** refers to the difference between the obtained results and the CPLEX Upper Bound

Table 3.3: Comparison of the results for S20 with two different parameters

Data Sets	UB (Best Integer)	A1B5				A2B3			
		Best costs	****Gaps (%)**	**Mean**	**STDEV**	**Best costs**	****Gaps (%)**	**Mean**	**STDEV**
S20T5	3394.78	3527.00	3.89	3551.21	11.566	3431.84	1.09	3456.66	16.788
S20T10	6759.71	7046.34	4.24	7114.85	36.759	6924.58	2.44	6946.81	25.406
S20T14	9368.08	9707.08	3.62	9783.33	52.582	9609.24	2.57	9678.84	39.873
S20T21	13929.21	14514.10	4.20	14598.30	82.124	14262.00	2.39	14325.30	53.532

Table 3.4: Results for both ACO and ACO2

Data	LB (Objective)	UB (Best Integer)		Gap*	ACO				ACO2			
		Costs	# veh		Best Costs	#veh	Time (secs)	Gap** (%)	Best Costs	#veh	Time (secs)	Gap** (%)
S12T5	2033.00	2231.96	19	12.08	2353.04	19	16	5.42	2290.38	19	15	2.62
S12T10	4047.64	4305.33	36	13.49	4604.56	37	30	6.95	4453.58	36	30	3.44
S12T14	5329.58	6196.35	52	14.92	6665.05	52	42	7.56	6462.09	52	41	4.29
S20T5	3208.35	3394.78	28	10.06	3617.39	28	47	6.56	3527.00	28	47	3.89
S20T10	6330.97	6759.71	56	11.00	7293.06	56	90	7.89	7046.34	56	91	4.24
S20T14	8769.73	9368.08	77	11.78	9982.36	77	126	6.56	9707.08	77	128	3.62
S20T21	12407.58	13929.21	115	14.25	15093.50	113	184	8.36	14514.10	113	188	4.20
S50T5	7614.43	8213.22	64	18.81	8176.18	59	317	-0.45	8115.38	61	324	-1.19
S50T10	13913.84	17359.20	135	22.03	17205.70	124	653	-0.88	16935.40	124	664	-2.44
S50T14	19300.45	25181.61	197	24.36	24357.10	176	942	-3.27	23969.10	178	941	-4.82
S50T21	29418.86	38626.96	311	25.01	37485.60	272	1438	-2.95	36620.40	273	1432	-5.19
S100T5	13208.54	16130.13	134	22.39	15247.60	122	1709	-5.47	15117.00	122	1734	-6.28
S100T10	25601.69	34388.15	293	26.74	31407.60	249	3527	-8.67	30963.90	249	3517	-9.96
S100T14	-	-	-	-	44610.50	355	4960	-	44155.00	355	4956	-

Gaps* refers to the gap between lower and upper bounds which obtained from CPLEX.

Gaps** refers to the difference between the obtained results and the CPLEX Upper Bound

Table 3.5: The details of the best solution between ACO and ACO2

Data	Algorithms	Results	Distance Cost	#Vec	Inventory Cost
S12T5	ACO2	2290.38	1910.38	19	0
S12T10	ACO2	4453.58	3721.58	36	12
S12T14	ACO2	6462.09	5359.09	52	63
S20T5	ACO2	3527.00	2967.00	28	0
S20T10	ACO2	7046.34	5926.34	56	0
S20T14	ACO2	9707.08	8140.08	77	27
S20T21	ACO2	14514.10	12254.10	113	0
S50T5	ACO2	8115.38	6863.38	61	32
S50T10	ACO2	16935.40	14418.40	124	37
S50T14	ACO2	23969.10	20395.10	178	14
S50T21	ACO2	36620.40	30957.40	273	203
S100T5	ACO2	15117.00	12636.00	122	41
S100T10	ACO2	30963.90	25983.90	249	0
S100T14	ACO2	44155.00	37043.00	355	12

Table 3.6: The average and standard deviation of total costs and CPU running time over 10 runs

Data	ACO				ACO2			
	Average (results)	**STDEV (results)**	**Average (Time)**	**STDEV (Time)**	**Average (results)**	**STDEV (results)**	**Average (Time)**	**STDEV (Time)**
S12T5	2398.75	47.34	15.60	0.52	2296.98	7.68	15.20	0.42
S12T10	4640.84	25.12	30.30	0.67	4512.78	30.86	29.40	0.52
S12T14	6718.46	32.53	41.50	0.53	6505.47	17.70	41.40	0.52
S20T5	3633.42	20.66	46.40	0.52	3551.21	11.57	46.80	0.42
S20T10	7320.39	19.11	90.20	0.63	7114.85	36.76	90.60	0.52
S20T14	10233.19	92.74	126.20	2.44	9783.33	52.58	127.90	4.43
S20T21	15250.71	76.43	185.80	0.92	14598.30	82.12	188.10	1.45
S50T5	8275.94	69.60	322.70	2.58	8180.07	46.66	326.90	4.23
S50T10	17290.91	65.23	657.50	4.33	16981.35	26.48	657.30	4.40
S50T14	24500.48	78.38	940.70	4.45	24034.87	42.56	936.50	6.77
S50T21	37601.92	100.29	1434.70	10.13	36804.87	99.54	1442.20	9.22
S100T5	15371.85	72.61	1703.00	17.16	15177.12	39.62	1705.50	13.91
S100T10	31523.81	87.42	3520.10	18.27	31099.62	67.01	3511.40	12.98
S100T14	44754.70	88.05	4961.80	10.53	44204.10	50.15	4956.60	10.31

3.7 Summary of the chapter

In this chapter, the explanation of the developed algorithm and the computational results are shown for both ACO and ACO2. Both of the algorithms are modified by adding the inventory cost in the global pheromones updating to solve for routing part while the procedure of updating inventory level is to determine the inventory for customers. A new transfer/swap aimed at combining split customers is also developed. This is carried out in order to obtain the improvement in term of the transportation costs. ACO embedded swap and 2-*opt* which is the intra route optimization procedure in the route improvement strategies. However, despite of swap and 2-*opt*, ACO2 also included 2-*opt** which is the inter route optimization procedure in the route improvement strategies. Thus, ACO2 is not only improving in term of transportation cost within the vehicles but also between the vehicles.

In this study, the computational experiments are done on different combination of the number of customers, 12, 20, 50 and 100 with the number of periods, 5, 10, 14 and 21. The overall results for both ACO and ACO2 showed that the algorithm performs better in larger instances if compared with small and medium instances as the obtained results for larger instances are better than the upper bound which generated from CPLEX 12.4. Meanwhile, we can also observe that ACO2 gives better results if compared with ACO for all the problem instances. From the standard deviation of the total costs, we can see that ACO2 gives better solution quality as well.

CHAPTER 4: POPULATION BASED ANT COLONY OPTIMIZATION

This chapter proposes the enhanced version of the previous modification on ACO to solve the same model of IRP which discussed in section 3.2. The chapter begins with the introduction. Next, the description of the developed algorithms is presented. There are 3 algorithms to be discussed in this chapter. The corresponding computational experiments are done and the results as well as the discussion of the obtained results will be presented. Moreover, this chapter also gives the statistical analysis on the computational results. Finally, the chapter ends with a summary.

4.1 Introduction

As mentioned in the previous chapter, the aim of IRP is to determine the schedule of deliveries, delivery amount, and how to route vehicles while minimizing the total cost that consists of inventory and transportation costs. The IRP is especially relevant in the vendor managed inventory strategy. Under this strategy, the supplier or manufacturer decides when to visit its customer, how much to deliver to each of them and how to combine them into vehicle routes. In the Chapter 3, we presented the model formulation which we intended to solve. In this chapter, we still tackle the same problem which is one-to-many network where a fleet of homogeneous vehicle transports multi products from a warehouse or depot to a set of geographically dispersed customers in a finite planning horizon and split delivery is allowed in our study. However, we will enhance the ACO with different modification with the aim to obtain more efficient results. Here, we introduce population based ACO.

In Chapter 3, we have discussed about the modification on ACO which includes the inventory cost in the global pheromones updating to solve our proposed model. However, we observed that the developed algorithm is not performing well as the algorithms only consist of one population to explore different set of inventory level in order to obtain the minimum cost. Therefore, in this chapter we will enhance the algorithm in order to improve the solution and hence give better results for our proposed model.

In the previous chapter, we can see that all the ants for both ACO and ACO2 share the same set of the inventory to build the routing part. We started all the solutions with zero inventory and inventory changes very slowly with inventory updating that we have applied. As a result, the algorithm requires many iterations in building different set of inventory to get a better solution. With the aim to improve this problem, we introduce the subpopulation of ants to build different set of inventory and then implement ACO to build for the routing part.

The contributions of this chapter are as follows:

1. The first proposed modification is done by dividing the ants into subpopulation and each subpopulation represents different set of inventory level. In addition, the inventory updating mechanism includes both forward and backward transfer.
2. The second proposed modification of algorithm is done by proposing the new formulation of pheromones values on customer's inventory which with the aim to store the information of the best inventory level for the customer. This lead to obtain the best set of inventory level which can also minimize the transportation cost as well.

3. Statistical analysis is done in order to obtain the significant difference between the proposed algorithms.

4.2 Population Based Ant Colony Optimization

In the typical ACO, only one population of ant is involved in building the solution. In this study, we propose dividing the ants into subpopulations to solve the problem instead of having one population (where the ants all have the same inventory). Each subpopulation represents one inventory level and the pheromones values will be different between the subpopulation but shares the same values within the subpopulation. Figure 4.1 outlines the algorithm of the population based ACO. In this chapter, we will implement three different approaches on updating inventory mechanism in order to improve the solution. The detail of the proposed algorithm will be presented in the following subsection.

Step1: Start the algorithm with Nearest Neighbour Algorithm and obtain the total distance and inventory cost.
Do the following steps from *i* =1 to *i*= *MaxiITER*,
Step 2: Do the route construction by using the ACO for all the ants in each subpopulation.
  Step 2.1: Local pheromones updating
    Choose the best solution among all the ants in each subpopulation to do the local pheromones updating.
  Step 2.2: Route improvement strategies
    Choose the best solution among all the subpopulations to do the route improvement strategies.
  Step 2.3: Global pheromones Updating
    IF (*i* modulo *PredefinedIterationForGL*)= 0,
      Choose the current best built solution to do the global pheromones updating. Then, go to **Step 3**.
    Else, go to **Step 3**.
Step 3: Update inventory level mechanism
IF (*i* modulo *predefinedIterInvUpt*) = 0
    Update the inventory for all subpopulations except the subpopulation containing the current best solution.
    Go to **Step 2**.
Else, go to **Step 2**.

Figure 4.1: Algorithm of the population based ACO

4.2.1 First modification of population based ACO (ACOBF)

In this subsection, we discuss the first modification on population based ACO and the method is named ACOBF. ACOBF divides the population of ants into subpopulation and each of the subpopulation will consist of different set of inventory level. The pheromones values of the ants within the subpopulation are the same.

ACOBF starts the algorithm by considering all the demands are met on time. In another word, ACOBF starts with zero inventories for all subpopulation. Although we started with the same inventory level for all the subpopulations, we observe that the inventory for each subpopulation will be different with each other after updating the inventory for a few times. Consequently the pheromone value differs between subpopulation as we select the best solution according to the subpopulation to generate the pheromone value.

In addition, ACOBF implements both forward and backward transfer mechanism in the updating inventory process for the customers. The details of ACOBF are discussed in the following sub-subsection.

4.2.1.1 Initial Solution

Similar to ACO2, Nearest Neighbor algorithm (NN) is applied in order to obtain the total distance. We construct the initial solution by having all the demand met in every period. In this study we adopt a simple Nearest Neighbour algorithm (NN) and the algorithm is modified to allow for split delivery. The vehicle starts at the depot and repeatedly visits the nearest customer (in terms of distance) until the capacity of the vehicle is fully occupied. Then, a new vehicle is initiated and the process continues until all customers have been assigned or visited. The total distance obtained by NN plus the inventory cost are embedded to initialize the τ_0, the initial pheromone in the local pheromone updating.

4.2.1.2 Route Construction

As mentioned above, ACOBF starts with the predefined number of subpopulation of ants and each subpopulation consists of predefined number of ants to build the solution. Similar to ACO2, each ant of the subpopulation in ACOBF will also implement the same mechanism to select the customer to be visited (discussed in subsection 3.3.2).

4.2.1.3 The local pheromone-updating rule

As discussed in the previous chapter, local updating is used to prevent a very strong arc being chosen by all the ants. After each of the ants in every subpopulation has built the solution, the best solution from each subpopulation will be selected. Then, the local updating will be done on each arc of the best solution from each subpopulation by using equation (3.15).

4.2.1.4 The global pheromone-updating rule

As mentioned in the previous chapter, global pheromones updating is done so that the ants will have a better starting point in searching for shorter path. After a predefined number of iterations, the current best solution γ^{gl} among all the subpopulations is selected and its routes are used as a reference for the global pheromones-updating for all subpopulation. Hence, the pheromones value for each arc of the best solution is updated by using the equation (3.16) for all the subpopulations.

4.2.1.5 Route improvement strategies

After the predefined number of iterations, the current best solution among the subpopulations will be selected and the routes will be further improved by adding route improvement strategies in the route construction procedure which is similar to ACO2. Three local searches; namely swap, 2-*opt** (Potvin and Rousseau, 1995) and 2-*opt* (Lin, 1965), are applied to improve the solution built by ACO.

4.2.1.6 Updating inventory level

We have done modification on the inventory updating mechanism for ACOBF. In ACO and ACO2 we only consider the backward transfer (where the periods to be transferred are selected randomly and inventory of the customer with the lowest inventory holding cost (subject to certain feasibility conditions which have been discussed in subsection 3.3.6) are transferred to the period $(t-1)$. However, we modified the inventory updating mechanism for ACOBF by including the forward transfer as well.

The inventory updating mechanism will be initiated after the predefined number of iterations. As mentioned above, we proposed two types of transfers, the forward and the backward transfers to update the inventory level. The selection of the forward and backward transfer is controlled by a random number and biasing towards the forward transfer. The selection of period is done randomly but we experiment two ways of selecting the customer to transfer their quantity of delivery based on the predefined number. If the current number of iteration is less than the predefined number, then the customer will be selected randomly. Otherwise, the customer who fulfilled the conditions (highest and lowest inventory holding cost for the forward and backward transfer respectively) will be selected. We observed that implementing a combination of randomly generated and deterministically generated customers allow the ants to have more explorations at the beginning of the iterations. The number of transfers for each updating mechanism after the predefined number of iterations is limited to the predefined number of transfers.

The updating mechanism will be implemented until the predefined number of iterations is reached. After the predefined number of iterations, the set of amount delivery which produced the best solution is determined and the routing part is executed continued. Figure 4.2 illustrated the algorithm of updating inventory for ACOBF.

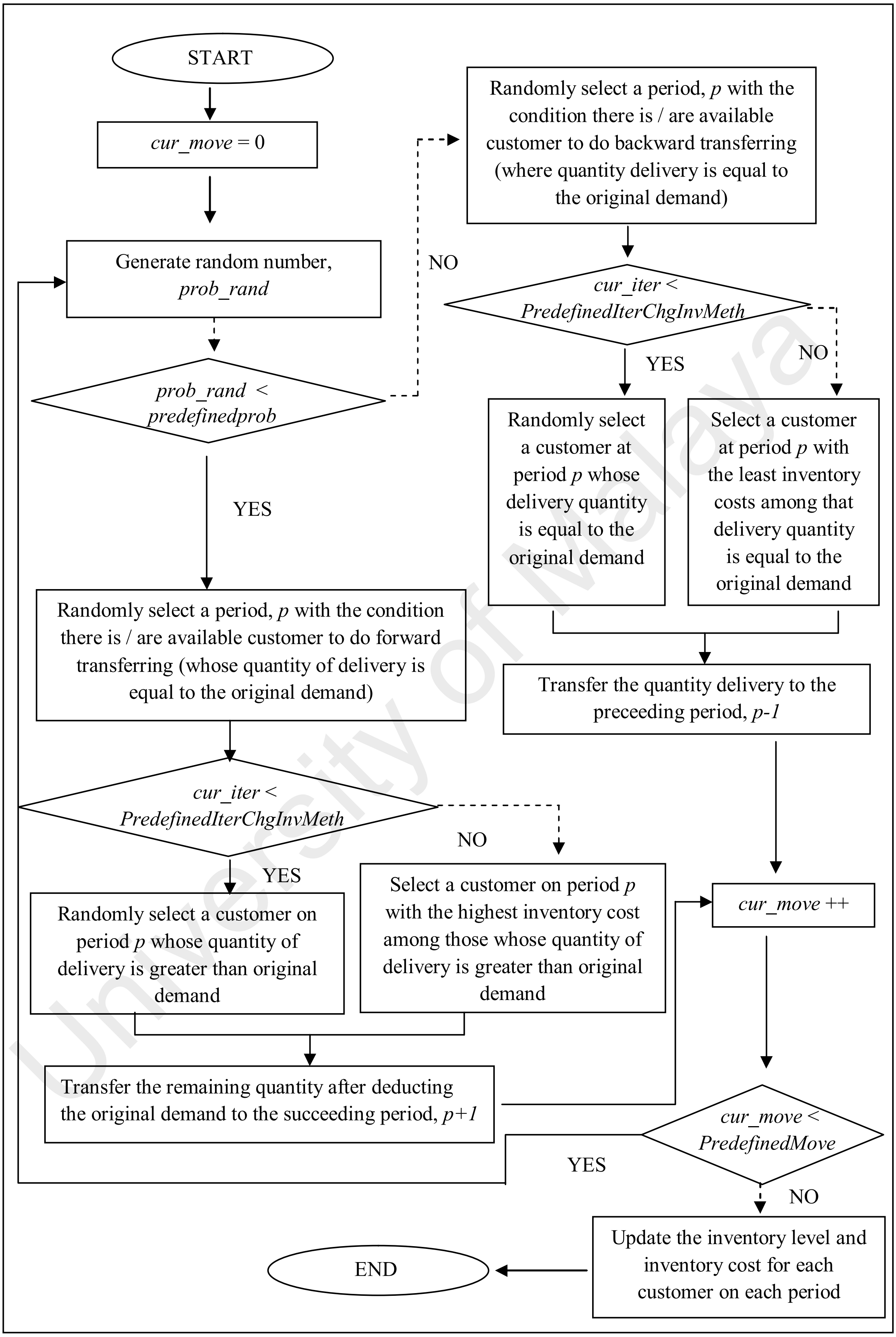

Figure 4.2: Algorithm of updating inventory for ACOBF

4.2.2 Second modification on population based ACO (ACOPher)

In the previous discussion on the three algorithms, ACO and ACO2 as well as ACOBF, we mentioned that the inventory cost is added at the global pheromones updating. However, the emphasis is still more on the routing part rather than inventory. We observed that, those proposed algorithms are more likely to give savings in term of transportation costs instead of inventory. Therefore, we will discuss an improved version of the algorithm in order to balance between inventory and transportation costs in this subsection and we called this proposed algorithm ACOPher.

We modify the algorithm such that the inventory updating is based on the pheromone value of each customer. The following equation is used to update the pheromones of customer's inventory for customer j in period p, $invpher_{jp}$:

$$invpher_{jp} = (1-\rho) * \left(invpher_{jp}\right) + \rho * \frac{a_{jp}}{c_{ij}} \quad (4.1)$$

where a_{jp} is the delivery quantity of customer j in period p while c_{ij} is the distance between the customer i and j for the routing of the current best solution and ρ represents the rate of pheromone evaporation as defined earlier. The second element of equation (4.1) specifies that if the ratio of the delivery quantity of the respective customer to its distance is higher, then higher pheromone value will be allocated, emphasizing favorably towards customer with higher ratio. The selection of c_{ij} for local and global inventory is discussed in more details in the following subsections.

4.2.2.1 Local pheromones updating of customer's inventory

For the local inventory updating, the c_{ij} value is taken from the best solution after some predefined number of iterations. Note that the predefined number of iterations is smaller than the predefined value of the overall inventory updating. The purpose of selecting the current best solution after a few iterations instead of selecting the best from all the current built solutions is to avoid the customer with strong customer's inventory pheromones being chosen all the time by all the subpopulations.

4.2.2.2 Global pheromones updating of customer's inventory

A slightly different mechanism is used in selecting the value of c_{ij} for the global inventory updating. The value of c_{ij} is selected from the current best solution among all the current built solution. The aim of this step is to lead the ants to select the best amount of delivery quantity to be carried for the customer. Figure 4.3 illustrates the algorithm for the process of inventory updating. Note that *PredefinedIterationForGL$_{p,}$* < *PredefinedIterationForGL* < p*redefinedIterInvUpt* < *predefinedIterInvUptGL* and

$$PredefinedIterationForGL_p = \frac{predefinedIterInvUpt}{PredefinedIterationForGL}$$

For *i* =1 to *i*= *MaxiITER*
Do the following steps after the route construction of ACO.
IFi modulo p*redefinedIterInvUpt* = 0, then
    IFi modulo *predefinedIterInvUptGL* = 0, then
        Choose the current best solution among all the built solutions.
        Use the routing part of the best solution to update the customer's inventory pheromones (Global pheromones updating of customer's inventory)
        Then, continue with the updating inventory mechanism.
    Else
        Choose the best solution from *PredefinedIterationForGL$_p$* solutions that have been built.
        Use the routing part of the best solution to update the customer's inventory pheromones (Local pheromones updating of customer's inventory)
        Then, continue with the updating inventory mechanism.
Else
        Continue with the route construction part.

Figure 4.3: Algorithm of updating customer's inventory pheromones for the second modified algorithm of ACO

Procedure for updating inventory level for each subpopulation

After the customer's inventory pheromones have been updated, the algorithm proceeds to select the customer to undergo the transfer. As mentioned earlier, the mechanism of selecting customer is based on the random number that has been generated but the priority is given to the customer based on the attraction value. If random value is less than certain predefined parameter, the customer with the highest attraction value of inventory pheromones is selected to undergo the transfer. The attraction of customer's inventory pheromones for customer j in period p, $AttInv_{ip}$ in each subpopulation is calculated using the equation

$$AttInv_{jp} = \left(invPher_{jp}\right)^{\mu} * \left({}^{1}/_{a_{jp}}\right)^{\omega} \tag{4.2}$$

Note that μ and ω are the parameters that control the influence of the inventory pheromone value and the desirability of delivery quantity for customer j in period p, respectively. The value of μ is set to be greater than ω as we want the pheromones values to influence more on the values of attraction. However, the value of μ should not be set to be too large as the ratio part of $\frac{a_{jp}}{c_{ij}}$ in the pheromone values will increase rapidly if the value of $a_{jp} > c_{ij}$. This is done to avoid a very strong customer to be chosen all of the time. Otherwise, the deterministic backward / forward transferring mechanism (highest and lowest inventory holding cost for the forward and backward transfer respectively) will be used to select the customer to update the inventory.

Similar to ACOBF, the updating mechanism will be implemented until the predefined number of iterations. After the predefined number of iterations, the set of delivery amount which produced the best solution will be determined and continue to let ACO to build the routing part. Figure 4.4 illustrates the algorithm of updating inventory for ACOPher.

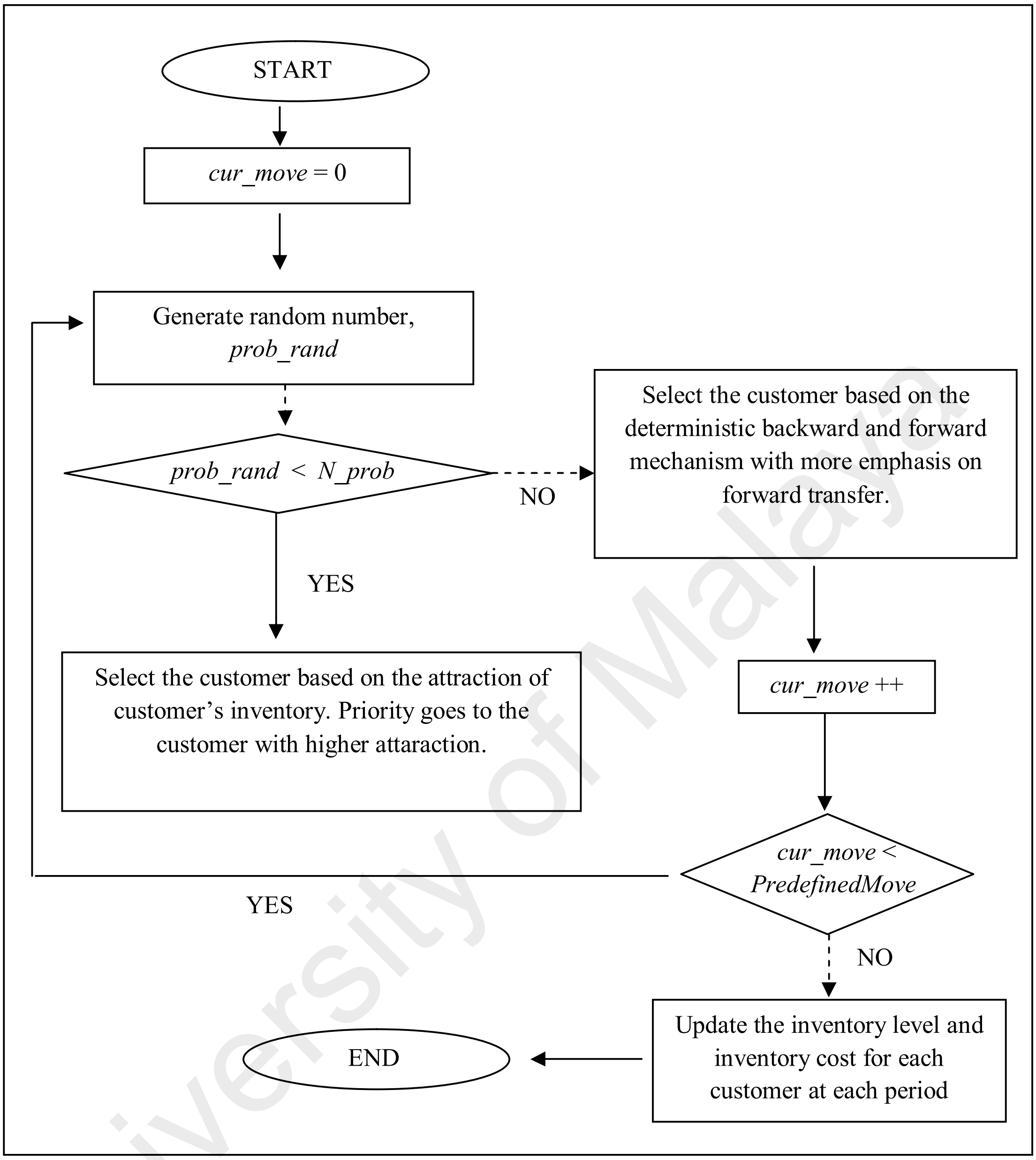

Figure 4.4: Algorithm of updating inventory for ACOPher

4.2.3 Enhanced version of second modification population based ACO (ACOPher2)

In the previous subsection, we discussed the modification on population based ACO by proposing the pheromones values on the customer's inventory. However, the algorithm is started with the same set of inventory for each subpopulation. This may cause more iterations needed to obtain the best set of inventory while balancing the transportation cost.

Therefore, in this subsection, we present the improved algorithm of the population based ACO where each subpopulation starts with the different sets of inventory level which include the set of zero inventory cost (all the demands are met on time), the set of inventory which is generated from the allocation model (the model is shown as below) and a set of randomly generated set of inventory.

The randomly generated set of inventory is obtained by randomly selecting the customer and the period to transfer the whole amount of demand to the preceding period to create different sets of inventory with the limited number of transfer is allowed for each set of inventory. With this strategy of starting with the different sets of inventory, it provides the ants to have more explorations at the beginning of the iterations and we hope it will converge to better solutions in less number of iteration.

Allocation Model:

$$Z = \min \sum_{t=1}^{T}\sum_{i=1}^{N} e_{it} a_{it} + \sum_{t=1}^{T}\sum_{i=1}^{N} h_i I_{it} \tag{4.3}$$

subject to constraint (3.2), (3.6), (3.7) and

$$I_{i0} = 0,\ i = 1, 2, \ldots, N \tag{4.4}$$

where e_{it} in the Equation (4.3) is referred to $e_{it} = \frac{2c_{0i}}{d_{it}}$.

4.3 Computational Results and discussion

The algorithms were written in C++ language by using Microsoft Visual studio 2008. The results of this study is compared with the lower bound (LB) and the upper bound (UB) generated by solving the formulation presented in Section 3 using CPLEX 12.4. All the computations were performed on 3.10 GHz processor with 8 GB of RAM.

4.3.1 Data sets

The algorithm is tested by using the same set of data sets that were discussed in the previous chapter, which consists of 12, 20, 50 and 100 customers, and combination with different number of periods, 5, 10, 14 and 21.

4.3.2 Results and Discussion

As mentioned in Section 3.6, we let CPLEX 12.4 run for a limited time 9000s (2.5 hours) in order to obtain the lower and upper bound (best integer solutions) for all the instances. The results of ACOBF which implemented the forward as well as backward transferring in the mechanism of updating inventory level and the results of ACOPher as well as ACOPher2 which implemented the customer's inventory pheromones are shown in Table 4.1. The parameters for ACOBF and ACOPher as well as ACOPher2 are set as follows: $\alpha = 1.0,\ \beta = 5.0,\ q_0 = 0.9, \rho = 0.1,\ \mu = 1.0,\ \omega = 0.5, \tau_0 = 1/(N \times L_{nn})$ where L_{nn} comprises of the total distance obtained from nearest neighbour algorithm and the inventory holding cost. The values of α and β are obtained from the sensitivity analysis which have been done in Chapter 3 while the value of q_0 is taken from Dorigo et al. (1996). The pheromones on customers' inventory in ACOPher is initially set as $1/d_{ip}$ if d_{ip} is not equal to 0, otherwise set as 0. In both ACOBF and ACOPher, there are 5 subpopulations and each subpopulation consists of 5 ants to build solution.

Table 4.1 tabulates the results of the algorithms of the population based ACO, ACOBF and ACOPher as well as ACOPher2. It presents the best total costs, the number of vehicles, lower and the upper bound (best integer solutions) obtained from CPLEX while Table 4.2 shows the CPU time.

From the results shown in Table 4.1 for all of the three algorithms of population based ACO, we note that the total costs of the data sets with 50 and 100 customers are less than the upper bound, which means that the algorithm is able to obtain better results when compared with the upper bound for 50 and 100 customer instances. However, ACOBF and ACOPher as well as ACOPher2 perform equally well for the small and medium instances and produced the gaps between the results and the best integer solutions that are less than 4.5 percent. In the same table, we can observe that ACOPher and ACOPher2 which implemented the new formulation of the customer's inventory pheromones to select the customer for updating the inventory level give better solution in most of the instances (12 out of 14 problem instances) if compared with ACOBF. We can expect the new mechanism of selecting the customer based on the customer's inventory pheromones is potential to produce the set of inventory which can balance between the inventory and transportation cost and hence give minimum of the total cost. Among the three algorithms of population based ACO, ACOPher2 perform better in 7 out of 14 instances.

Table 4.2 gives the results of non-population and population based ACO. In this table, we can see that the algorithms of population based ACO outperform than non-population based ACO in all of the problem instances. We can conjecture that dividing the ants into subpopulation in order to give more chance to the ants to explore more set of inventory level in less iteration which can balance with the transportation cost is potential to produce better solution. The details including the distance cost, number of vehicles as well as the inventory cost of the best solution among the 5 algorithms are presented in Table 4.3.

Table 4.4 gives the computational time of ACOBF, ACOPher and ACOPher2. In this table, we can observe that the computational time of the three algorithms is not much different in all instances as they required almost the same computational time to produce the results. Table 4.5 presents the average and standard deviation of the total cost and computational time over 10 runs for the population based ACO. In this Table 4.5, we can see that the algorithms including the pheromones values on the customer's inventory, ACOPher and ACOPher2 gives better in term of quality solution than ACOBF. This is because of the both algorithms; ACOPher and ACOPher2 give less standard deviation in terms of the best obtained solution if compared with ACOBF in most of the problem instances (11 out of 14 problem instances).

Table 4.1: Results of the algorithms of population based ACO

Data	LB (Objective)	UB (Best Integer)		ACOBF			ACOPher			ACOPher2		
		Costs	# veh	Best Costs	#veh	Gap* (%)	Best Costs	#veh	Gap* (%)	Best Costs	#veh	Gap* (%)
S12T5	2033	2231.96	19	2285.94	19	2.42	2279.62	19	2.14	2278.79	19	2.1
S12T10	4047.64	4305.33	36	4441.23	36	3.16	4436.01	36	3.04	4427.11	36	2.83
S12T14	5329.58	6196.35	52	6422.24	52	3.99	6421.95	52	3.98	6388.74	51	3.1
S20T5	3208.35	3394.78	28	3522.65	28	3.77	3527.8	28	3.92	3507.07	28	3.31
S20T10	6330.97	6759.71	56	7046.23	56	4.24	7037.72	56	4.11	7042.96	56	4.19
S20T14	8769.73	9368.08	77	9697.48	77	3.52	9689.11	77	3.43	9662.48	77	3.14
S20T21	12407.58	13929.21	115	14487.1	113	4.01	14481.1	113	3.96	14476.3	113	3.93
S50T5	7614.43	8213.22	64	8110.05	60	-1.26	8134.02	60	-0.96	8121.4	59	-1.12
S50T10	13913.84	17359.2	135	16871.2	124	-2.81	16852.7	125	-2.92	16862.3	125	-2.86
S50T14	19300.45	25181.61	197	23886.3	178	-5.14	23939.8	178	-4.93	23930	178	-4.97
S50T21	29418.86	38626.96	311	36715.1	273	-4.95	36572	274	-5.32	36644.1	274	-5.13
S100T5	13208.54	16130.13	134	15080.2	122	-6.51	15108.9	122	-6.33	15009.2	122	-6.95
S100T10	25601.69	34388.15	293	30960.4	249	-9.97	30857.2	249	-10.27	30897.4	249	-10.15
S100T14	-	-	-	43997.5	355		43934.6	355	-	44043.9	355	-

Gaps* refers to the difference between the obtained results and the CPLEX Upper Bound

Table 4.2: Results of the algorithms of non-population and population based ACO

Data	UB (Best Integer)	Non-population based ACO				Population based ACO					
		ACO		ACO2		ACOBF		ACOPher		ACOPher2	
	Costs	Best Costs	Gap* (%)	Best Costs	Gap* (%)	Best Costs	Gap* (%)	Best Costs	Gap* (%)	Best Costs	Gap* (%)
S12T5	2231.96	2353.04	5.42	2290.38	2.62	2285.94	2.42	2279.62	2.14	2278.79	2.1
S12T10	4305.33	4604.56	6.95	4453.58	3.44	4441.23	3.16	4436.01	3.04	4427.11	2.83
S12T14	6196.35	6665.05	7.56	6462.09	4.29	6422.24	3.99	6421.95	3.98	6388.74	3.1
S20T5	3394.78	3617.39	6.56	3527	3.89	3522.65	3.77	3527.8	3.92	3507.07	3.31
S20T10	6759.71	7293.06	7.89	7046.34	4.24	7046.23	4.24	7037.72	4.11	7042.96	4.19
S20T14	9368.08	9982.36	6.56	9707.08	3.62	9697.48	3.52	9689.11	3.43	9662.48	3.14
S20T21	13929.21	15093.5	8.36	14514.1	4.2	14487.1	4.01	14481.1	3.96	14476.3	3.93
S50T5	8213.22	8176.18	-0.45	8115.38	-1.19	8110.05	-1.26	8134.02	-0.96	8121.4	-1.12
S50T10	17359.2	17205.7	-0.88	16935.4	-2.44	16871.2	-2.81	16852.7	-2.92	16862.3	-2.86
S50T14	25181.61	24357.1	-3.27	23969.1	-4.82	23886.3	-5.14	23939.8	-4.93	23930	-4.97
S50T21	38626.96	37485.6	-2.95	36620.4	-5.19	36715.1	-4.95	36572	-5.32	36644.1	-5.13
S100T5	16130.13	15247.6	-5.47	15117	-6.28	15080.2	-6.51	15108.9	-6.33	15009.2	-6.95
S100T10	34388.15	31407.6	-8.67	30963.9	-9.96	30960.4	-9.97	30857.2	-10.27	30897.4	-10.15
S100T14	-	44610.5	-	44155	-	43997.5	-	43934.6	-	44043.9	-

Gaps* refers to the difference between the obtained results and the CPLEX Upper Bound

Table 4.3: The details of the best solution among the 5 algorithms; ACO, ACO2, ACOBF, ACOher, and ACOPher2

Data	Algorithms	Results	Distance Cost	#Vec	Inventory Cost
S12T5	ACOPher2	2278.79	1857.79	19	41
S12T10	ACOPher2	4427.11	3695.11	36	12
S12T14	ACOPher2	6388.74	5308.74	51	60
S20T5	ACOPher2	3507.07	2941.07	28	6
S20T10	ACOPher	7037.72	5917.72	56	0
S20T14	ACOPher2	9662.48	8092.48	77	30
S20T21	ACOPher2	14476.30	12189.30	113	27
S50T5	ACOBF	8110.05	6904.05	60	6
S50T10	ACOPher	16852.70	14350.70	125	2
S50T14	ACOBF	23886.30	20326.30	178	0
S50T21	ACOPher	36572.00	31084.00	274	8
S100T5	ACOPher2	15009.20	12569.20	122	0
S100T10	ACOPher	30857.20	25877.20	249	0
S100T14	ACOPher	43934.60	36833.60	355	1

Table 4.4: Computational time of the algorithms

Data	ACOBF	ACOPher	ACOPher2
	Time (secs)	Time (secs)	Time (secs)
S12T5	21	21	21
S12T10	38	38	39
S12T14	51	51	51
S20T5	54	55	55
S20T10	102	103	102
S20T14	140	140	139
S20T21	207	207	208
S50T5	348	354	347
S50T10	712	721	708
S50T14	994	1013	1004
S50T21	1509	1544	1524
S100T5	1844	1892	1873
S100T10	3698	3789	3646
S100T14	5175	5294	5082

Table 4.5: The average and standard deviation of total costs and CPU running time over 10 runs (Population Based ACO)

Data	ACOBF				ACOPher				ACOPher2			
	Average (results)	STDEV (results)	Average (Time)	STDEV (Time)	Average (results)	STDEV (results)	Average (Time)	STDEV (Time)	Average (results)	STDEV (results)	Average (Time)	STDEV (Time)
S12T5	2302.38	9.03	21.50	0.53	2295.46	9.25	21.00	0.00	2295.57	8.97	21.30	0.48
S12T10	4458.82	18.07	38.10	0.32	4453.49	11.34	38.10	0.32	4460.70	24.50	38.20	0.42
S12T14	6453.36	16.68	50.60	0.52	6444.12	24.88	51.10	0.32	6430.83	24.14	50.80	0.63
S20T5	3544.60	20.61	55.20	0.79	3557.85	18.25	55.20	0.63	3532.75	13.80	55.00	0.47
S20T10	7090.80	50.00	102.60	0.70	7074.03	16.76	102.60	0.52	7065.40	13.99	102.00	0.67
S20T14	9720.30	15.74	140.20	0.79	9704.23	11.28	140.30	0.67	9709.10	22.25	139.80	0.63
S20T21	14499.68	12.04	205.30	1.06	14499.05	9.27	207.20	1.14	14492.47	12.82	207.60	2.46
S50T5	8151.60	29.75	348.90	1.52	8156.95	15.33	355.50	1.65	8156.44	19.89	351.40	2.07
S50T10	16940.68	52.65	710.80	3.97	16936.59	39.75	722.00	3.62	16927.49	34.74	710.40	2.76
S50T14	23991.73	60.69	1001.00	4.35	24002.73	56.42	1018.70	5.06	24002.17	38.71	998.30	6.11
S50T21	36759.41	29.66	1514.40	4.88	36748.35	74.84	1543.20	4.16	36769.03	68.72	1501.50	19.20
S100T5	15136.59	31.91	1848.10	9.86	15143.22	19.92	1897.50	4.12	15116.49	45.14	1860.60	8.44
S100T10	31069.68	53.81	3693.30	11.55	31037.96	81.26	3786.60	4.67	31050.97	59.72	3645.40	17.38
S100T14	44128.40	70.62	5172.20	13.36	44075.46	88.62	5293.60	10.44	44114.38	43.89	5065.40	31.05

4.4 Statistical Analysis Results

In this study, we performed a nonparametric statistical analysis for multiple comparisons to determine whether there is a significant difference between the five algorithms which are ACO, ACO2, ACOBF, ACOPher and ACOPher2. The Friedman test, Iman and Davenport and Friedman Aligned Rank test are often employed inside the framework of experimental analysis to decide when one algorithm is considered better than another (Derrac et al, 2011).

The Iman and Davenport and Friedman Aligned Rank tests are to alleviate the weakness of the Friedman test. Iman and Davenport proposed a less conservative test to improve on the Friedman test which is conservative. The ranking scheme adopted in the Friedman test has a weakness in which it allows for intra-set comparison only. It is based on n sets of ranks, one set for each data set in this case the performances of the algorithms analysed are ranked separately for each data set. Hence the intra-set comparisons are not meaningful. When the number of algorithms is small (in this study is only five), this may pose a disadvantage. We propose Friedman Aligned Rank test, where the observation is aligned with respect to the problems (datasets) as well as with respect to the algorithms. The alignment is carried out by subtracting the mean of each dataset in each algorithm. The details of each test will be discussed in the following subsections.

4.4.1 Friedman Test

Friedman test gives the multiple comparisons test with the aim to detect significant differences between behaviors of two or more algorithms. The null hypothesis for Friedman's test states the equality of medians between the populations while the alternative hypothesis gives the negation of the null hypothesis. Firstly, each of the problem, i, in the original results is ranked from 1 (best result) to k (worst result), and each ranks is denoted by $r_i^j (1 \leq j \leq k)$. Meanwhile, the average of the ranks obtained in all problems is calculated by using the equation (4.5) for each algorithm j.

$$R_j = \frac{1}{n} \sum_i r_i^j \; . \tag{4.5}$$

The Friedman statistic F_f can be calculated as follows:

$$F_f = \frac{12n}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right]. \tag{4.6}$$

Friedman test is distributed according to Chi-Square distribution with $k - 1$ degree of freedom.

4.4.2 Iman and Davenport

Iman and Davenport is derived from the Friedman statistic and is distributed according to an F distribution with $k - 1$ and $(k - 1)(N - 1)$ degrees of freedom. The statistic of Iman and Davenport is calculated by using the equation (4.7) as follows:

$$F_{ID} = \left(\frac{(n-1)\chi_F^2}{n(k-1) - \chi_F^2} \right). \tag{4.7}$$

4.4.3 Friedman Aligned Ranks test

In the method of Friedman Aligned Ranks test, the average performance achieved by all algorithms in each problem is computed. Then, the difference between the performance obtained by an algorithm and the calculated average is obtained. These steps are repeated for each combination of algorithms and problems. The resulting differences (aligned observations) for each combination of the problems and algorithms are then ranked from 1 to $k \cdot n$. The ranks assigned to the aligned observations are called aligned ranks. The statistic values for Friedman Aligned Rank test is calculated as follows:

$$F_{AR} = \frac{(k-1)\left[\sum_{j=1}^{k} \hat{R}_j^2 - (kn^2/4)(kn+1)^2\right]}{\left\{kn(kn+1)(2kn+1)\right\}/6 - (1/k)\sum_{i=1}^{n} \hat{R}_i^2} \tag{4.8}$$

where $\hat{R}_i^2$ and $\hat{R}_j^2$ is equal to the rank total of the ith problem and jth algorithm respectively. The test statistic is then compared with Chi-Square distribution in which the degree of freedom is $k - 1$.

4.4.4 Results of non-parametric tests

The statistical value of F_f, F_{ID} and F_{AR} are compared with the critical values of the respective distribution at significance level, α equal to 0.05. From Table 4.4, we can observe that all the statistical values of F_f , F_{ID} and F_{AR} are greater than the critical values. It means that the null hypothesis is rejected and shows that there are significant differences among all the algorithms.

Table 4.6: Statistical Analysis on results of ACO, ACO2, ACOBF, ACOPher and ACOPher2

Test	**Critical values on alpha = 0.05**	**Statistical Values**
Friedman test [1,2]	9.488	39.829
Iman and Davenport test[3]	2.550	32.018
Friedman Aligned Ranks test	9.488	39.461

4.4.5 Post-hoc test

The results of the statistical analysis of Friedman, Iman-Davenport and Friedman Aligned Rank tests show that there are significant differences over the whole multiple comparisons but unable to give the proper comparisons between some of the algorithms considered. In this subsection, we apply Bonferroni-Dunn procedure (see Derrac et al., 2011) to do the comparison by considering a control method and a set of algorithms with a family of hypotheses is defined. Application of the post-hoc test can lead to give the *p*-value which is used to determine the degree of rejection of each hypothesis.

The *p*-value can be attained via the conversion of the rankings which is computed from the main nonparametric procedure by using normal approximation. In this subsection, we selected Friedman test as the main nonparametric procedure to compare the *i*th algorithm with the selected control method and then used the equation (4.9) in order to obtain the *z* value. In this study, we take ACOPher2 as the control method.

$$z_i = (R_i - R_0) / \sqrt{\frac{k(k+1)}{6n}} \tag{4.9}$$

where R_i and R_0 refer to the average rankings by the Friedman test of the *i*th algorithms and the average ranking of the control method. Then, k and n refer to the number of algorithms and number of problems, respectively.

On the basis of these z_i values, the corresponding cumulative normal distribution values, p_i can be calculated. The values of p_i are compared with θ_i that are calculated based on the selected significance level α (set to 0.05 in our study), since $\theta_i = \alpha / (k-1)$. If $p_i < \theta_i$, it means that the corresponding null hypothesis which gives the assumption that there is no significant difference between the two compared algorithms is rejected and denoted as $h=1$. Otherwise, the null hypothesis is accepted which means that there is no significant difference between the two algorithms, denoted as $h=0$.

From the ranking results based on the Bonferroni-Dunn procedure shown in Table 4.5, we can conclude that ACOPher2 is significantly different with ACO and ACO2 but not with ACOBF and ACOPher.

Table 4.7: Ranking results based on Bonferroni-Dunn Procedure for the compared algorithms.

Rank	Algorithms	z	p	h	θ
1	ACOPher2	-	-	-	0.0125
2	ACOPher	0.5976	0.5501	0	
3	ACOBF	1.4343	0.1515	0	
4	ACO2	3.2271	0.0013	1	
5	ACO	5.4981	0.0000	1	

4.5 Summary of the chapter

The integration of inventory and transportation plays an important role in supply chain management. In this study we solved the model that consists of multi-products and multi-periods IRP with split delivery being allowed by using our developed algorithms. The development of a modified ACO in which the ants are divided into subpopulation in order to solve the problem are discussed in this chapter. Therefore, we discussed on several developed algorithms on the population based ACO. We design a population based ACO by segregating each subpopulation by the inventory level. We have constructed a modified global routing updating for routes that includes some information on inventory.

In this study, we proposed several different algorithms for the population-based ACO. The first, ACOBF incorporates the random and deterministic inventory updating mechanism where the forward and backward inventory updating biasing towards the forward inventory updating. Secondly, the new inventory updating mechanism where it takes into account the customers inventory in the local and global pheromone updating is also explained in this chapter and the algorithms are called ACOPher as well as ACOPher2. The selection of customers for the transfer is based on the attraction for both algorithms, ACOPher and ACOPher2. Both ACOBF and ACOPher start with initial population of zero inventories. However in ACOPher2, each subpopulation starts with different set of inventory level. The combination comprises of zero inventory, inventory level generated using the allocation model and randomly generated inventory levels.

All the algorithms developed in this chapter are tested on the data with different combinations of the number of customers, 12, 20, 50 and 100 with the number of periods, 5, 10, 14 and 21. The computational results for the algorithms ACOBF, ACOPher and ACOPher2 are presented in this chapter. The overall results for the algorithms ACOBF, ACOPher and ACOPher2 show that the algorithms performed better in larger instances if compared with small and medium instances as the obtained results for larger instances are better than the upper bound which generated from CPLEX 12.4. Meanwhile, we can also conclude that ACOPher2 gave better results if compared with ACOBF as well as ACOPher in most of the instances. As the conclusion, we can say that the results are better in most of the instances by including the selection of customer for transferring based on the attraction. From the standard deviation of the total costs, we can observe that the algorithms including the pheromones values at customer's inventory, ACOPher as well as ACOPher2 give better performance in terms of solution quality if compared with ACOBF.

CHAPTER 5: STOCHASTIC INVENTORY ROUTING PROBLEM

This chapter focuses on Stochastic Inventory Routing Problem (SIRP) in which the demand is not known in advance but given in probabilistic sense. The chapter starts with the introduction and followed by the mathematical formulation of SIRP. A two phase algorithm is proposed to solve our model and the details on these two phases are discussed in the following subsections respectively. An enhanced version of the two phase algorithm will be discussed in the following section. The computational experiments on a set of randomly generated data set are carried out and the results as well as the discussion of the algorithms obtained are discussed. Finally, the chapter ends with a summary.

5.1 Introduction

In the previous two chapters, we are concerned in solving the problem of IRP in which the customer's demand is deterministic and time varying in each period. In this chapter, we extended our work to tackle the problem of SIRP in which the demand is stochastic and expressed as some probability functions. This chapter introduces a multi-period SIRP with split delivery (SIRPSD) where the depot / warehouse housed a fleet of homogeneous capacitated vehicles for transportation of products to customers to fulfill their demand and the demands are stochastic in each period. The total transportation and expected inventory costs are considered as the main objective of the problem to be minimized while the service level of the customers are satisfied by imposing some constraints and they can be adjusted according to practical applications.

In this study, we implemented two-phase algorithms to solve our proposed model on SIRP where Phase I solves the inventory subproblem while Phase II build the routes and obtains the transportation costs.

The contributions of this chapter are as follows:

1. The Lagrangian relaxation problem in Yu et al. (2012) can be decomposed into the inventory and routing sub problem. In phase I, we modified the inventory sub problem of Yu et al. (2012) to solve our own inventory sub problem. The distance cost in their inventory sub problem is different from our model. Inspired by the allocation model of Bard and Nananukul (2009), we modified our inventory sub problem by replacing the distance costs in Yu et al. (2012) with the approximated cost which is modified from the cost presented in Bard and Nananukul (2009). The details are presented in subsection 5.3.1**.**
2. In phase II of routing subproblem, we proposed population based ACO described in Chapter 4, but has been modified accordingly. The inventory in the local and global pheromones updating has been removed and the initial pheromone depends only on the total distance of the route. Instead of depending on different inventory levels to create the subpopulations, we employ different routing heuristics to differentiate between each subpopulation. The quantity of deliveries obtained from Phase I is used to build routes using different heuristics and the purpose of implementing this is to give the subpopulation of ants more chances to explore the solution, hence obtaining better results. The details are presented in subsection 5.3.2**.**
3. The two-phase algorithm is then enhanced by adding the mechanism of updating inventory. The updating inventory mechanism is focused on backward transfer (i.e. transfer the quantity delivery to the preceding period) and the customers are

selected based on the new formulation of savings that was proposed in this study. The customer with the highest savings is selected to undergo the backward transfer and the details are discussed in Section 5.4.

5.2 Mathematical Formulation

We consider a one-to-many network, as in Chapter 4, where a fleet of homogeneous vehicles transports multi products from a warehouse or depot to a set of geographically dispersed customers in a finite planning horizon. The following assumptions are made in this model.

- The fleet of homogenous vehicles with limited capacity is available at the warehouse.
- Customers can be served by more than one vehicle (split delivery is allowed).
- Each customer's demand is stochastic in each period, and the customers require the depot/warehouse to satisfy their demands with a certain service level by limiting the possibility of stockout within a given range. The demand in each period is stochastic and obeys a given probability distribution. In this study, we assume that the stochastic demand is subject to normal distribution which is discussed later in this subsection. The exact demand is only known after its realization.
- The holding cost per unit item per unit time is constant for each product and incurred at the customer sites but not at the warehouse. The holding cost does not vary throughout the planning horizon.
- A multi-period horizon is considered.

The SIRP model is modified from Yu et al. (2012) by replacing q_{ijt}, the quantity delivered through the arc (*ij*), with x_{ijt} the distance through directed arc (*ij*) in the objective function.

The problem is modeled as mixed integer programming and the following notations are used:

Indices

$t = 1, 2, \dots, T$ period index

$W = 0$ warehouse/depot

$S = 1, 2, \dots, N$ a set of customers where customer *i* demands product *i* only

Parameters

C vehicles capacity (assume to be equal for all the vehicles).

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)

c_{ij} travel distance between customer *i* and *j* where $c_{ij} = c_{ji}$ and the triangle inequality, $c_{ik} + c_{kj} \geq c_{ij}$ holds for any i, j, and k with $i \neq j$, $k \neq i$ and $k \neq j$

h_i inventory carrying cost at the customer for product *i* per unit product per unit time

d_{it} stochastic demand of customer *i* in period *t*

$d_{i,(1,t)}$ $=\sum_{f=1}^{t} d_{it}$ cumulative stochastic demand from period 1 to *t*

W_i the inventory capacity for customer site.

Variables

a_{it} delivery quantity to customer i in period t

I_{it} inventory level of product i at the customer i at the end of period t

$I_{it} = \max(0, I_{it})$ On-hand inventory of customer $i$at the end of period t, which excludes the stock-out ($I_{it} < 0$).

q_{ijt} quantity transported through the directed arc (ij) in period t

x_{ijt} number of times that the directed arc (ij) is visited by vehicles in period t

$E(I_{it}^{+})$ $\int_0^{I_{i,0}+\sum_{f=1}^{t} a_{if}} \left(\left(I_{i,0} + \sum_{f=1}^{t} a_{if} - x \right) \left(\frac{e^{-\frac{(x-t\mu)^2}{2\left(\sqrt{t}\sigma\right)^2}}}{\sqrt{t}\sigma\sqrt{2\pi}} \right) dx \right)$

The model for our inventory routing problem is given as below:

$$Z = \min \underbrace{\sum_{t=1}^{T}\sum_{i=1}^{N} h_i E\left(I_{it}^{+}\right)}_{\mathrm{I}} + \underbrace{V\left(\sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{i=0}^{N} c_{ij} x_{ijt} + \sum_{t=1}^{T}\sum_{i=1}^{N} c_{i0} x_{i0t}\right)}_{\mathrm{II}} + \underbrace{F\sum_{t=1}^{T}\sum_{i=1}^{N} x_{0it}}_{\mathrm{III}} \tag{5.1}$$

subject to

$$I_{it} = I_{i,0} + \sum_{k=1}^{t} a_{ik} - \sum_{f=1}^{t} d_{if},\ i = 1, 2, \dots, N, t = 1, 2, \dots, T \tag{5.2}$$

$$Prob(I_{it} \geq 0) \geq \alpha_i,\ i = 1, 2, \dots, N, t = 1, 2, \dots, T \tag{5.3}$$

$$Prob\left(I_{i,t-1} + a_{it} \leq W_i\right) \geq \beta_i,\ i = 1, 2, \dots, N, t = 1, 2, \dots, T \tag{5.4}$$

$$I_{i,0} + a_{i1} \leq W_i, \quad i = 1, 2, \dots, N \tag{5.5}$$

$$\sum_{\substack{j=0 \\ i \neq j}}^{N} q_{ijt} + a_{it} = \sum_{\substack{j=0 \\ i \neq j}}^{N} q_{jit},\ i = 1, 2, \dots, N, t = 1, 2, \dots, T \tag{5.6}$$

$$\sum_{i=1}^{N} q_{0it} = \sum_{i=1}^{N} a_{it},\ t = 1, 2, \dots, T \tag{5.7}$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{N} x_{ijt} = \sum_{\substack{i=0 \\ i \neq j}}^{N} x_{jit},\ j = 0, 1, 2, \dots, N,\ t = 1, 2, \dots, T \tag{5.8}$$

$a_{it} \geq 0,\ i = 1, 2, \dots, N, t = 1, 2, \dots, T$ (5.9)

$q_{ijt} \geq 0,\ i = 0, 1, 2, \dots, N,\ j = 0, 1, 2, \dots, N, j \neq i,\ t = 1, 2, \dots, T$ (5.10)

$q_{ijt} \leq Cx_{ijt}, i = 0, 1, 2, \dots, N,\ j = 0, 1, 2, \dots, N, j \neq i,\ t = 1, 2, \dots, T$ (5.11)

$x_{ijt} \in \{0,1\}$, $i = 1, 2, \dots, N,\ j = 1, 2, \dots, N,\ t = 1, 2, \dots, T$ (5.12)

$x_{0jt} \geq 0$, and integer, $j = 1, 2, \dots, N,\ t = 1, 2, \dots, T$ (5.13)

The objective function (5.1) includes the inventory costs (I), the transportation costs (II) and the vehicle fixed cost (III). (5.2) is the inventory balance equation for each product at the warehouse whilst (5.3) ensures that the probability for customer i's demand satisfied in period t is no less thanα_i while (5.4) describes the service levels related to the customers' warehouses and guarantee that the probability of customer i's warehouse capacity being able to accommodate its maximum inventory level is not less than β_i at period $t = 2, \dots, T$. (5.5) ensure that every customer's warehouse inventory capacity should be no less than its maximum level in period 1.

The product flow balance at each customer is ensured by the flow conservation equations (5.6) whilst eliminating all possible subtours. (5.7) assures the collection of accumulative delivery quantity at the warehouse (split delivery). (5.8) ensures that the number of vehicles leaving the warehouse equals to the number of vehicles returning to warehouse. Meanwhile, (5.11) guarantees that the vehicle capacity is respected and gives the logical relationship between q_{ijt} and x_{ijt} which allows for split delivery.

Normal distribution for stochastic demands

Supposing that the demand for customer i in period t, d_{it} is a random variable subject to a normal distribution with mean, μ_i and standard deviation, σ_i. That is

$$d_{it} \sim N\left(\mu_i, \sigma_i^2\right) \tag{5.14}$$

We define the probability density function: $\frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(d_{it}-\mu_i)^2}{2\sigma_i^2}}, -\infty \le d_{it} \le +\infty$.

The accumulative customer demand is given by $d_{i,(1,t)} = \sum_{f=1}^{t} d_{if}$ and it obeys

$$d_{i,(1,t)} \sim N\left(t\mu_i, t\sigma_i^2\right) \tag{5.15}$$

Defining $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-t^2/2}\, dt$ we have

$$F_{i,(1,t)}(r) = P\left(d_{i,(1,t)} \le r\right) = P\left(\frac{d_{i,(1,t)} - t\mu_i}{\sqrt{t}\sigma_i} \le \frac{r - t\mu_i}{\sqrt{t}\sigma_i}\right) = \Phi\left(\frac{r - t\mu_i}{\sqrt{t}\sigma_i}\right)$$

which is accumulative probability distribution function of stochastic demand d_{it}.

Transformation from stochastic constraints into deterministic ones

In Yu et al (2012), the authors mentioned that the stochastic terms in (5.1), (5.2), (5.3) and (5.4) can be transformed into a simplified deterministic model which is easier to solve. After substituted the term of (5.2) into the objective function (5.1), the objective function (5.1) can be reformulated as

$$\sum_{i=1}^{N} h_i E\left(I_{it}^{+}\right) = \sum_{i=1}^{N} h_i \int_0^{I_{i,0}+\sum_{f=1}^{t} a_{if}} \left(\left(I_{i,0} + \sum_{f=1}^{t} a_{if} - x\right)\left(\frac{e^{-\frac{(x-t\mu)^2}{2\left(\sqrt{t}\sigma\right)^2}}}{\sqrt{t}\sigma\sqrt{2\pi}}\right) dx\right).$$

By substituting (5.2) into the terms of (5.3) as well as (5.4), the authors stated that the terms of (5.3) and (5.4) can be reformulated as the term of (5.3') and (5.4') respectively.

$$\sum_{f=1}^{t} a_{if} \ge F_{i,(1,t)}^{-1}(\alpha) - I_{i,0}, \qquad i = 1, \ldots, N,\, t = 1, \ldots, T \tag{5.3'}$$

$$\sum_{f=1}^{t} a_{if} \leq W_i + F^{-1}_{i,(1,t-1)}(1-\beta) - I_{i,0}\ , \qquad i = 1, \ldots, N, t = 2, \ \ldots, T \tag{5.4'}$$

We considered our stochastic demand obeyed Normal Distribution like in Yu et al. (2012), in the later part of the paper indeed indicated that both constraints term of (5.3') and (5.4') can be reformulated again into a more practical way after take into account the normal distribution condition (i.e. equation (5.15)). Hence, the constraints term of (5.3'') and (5.4'') are obtained.

$$\sum_{f=1}^{t} a_{if} \geq t\mu_i + \sigma_i \sqrt{t}\Phi^{-1}\left(\Phi\left(-\mu_i\sqrt{t} \big/ \sigma_i\right) + \alpha_i\right) - I_{i,0} \tag{5.3''}$$

$$\sum_{f=1}^{t} a_{if} \leq W_i + (t-1)\mu_i + \sigma_i \sqrt{t-1}\phi^{-1}(1-\beta_{it}) - I_{i,0}\,. \tag{5.4''}$$

The details for the derivation of all the transformation / reformulation can be referred to the paper of Yu et al (2012).

5.3 Two-Phase Algorithm of SIRP (SIRPACO1)

We proposed two-phase algorithm to solve our proposed model of SIRP. Phase I solved the sub problem of inventory in order to get the expected inventory cost and quantity delivery for each customer in each period. Meanwhile, Phase II implements ACO to build the routes based on the quantity delivery obtained from Phase I in order to get the transportation costs. Further details of Phase I and II is discussed in subsection 5.3.1 and 5.3.2 respectively.

5.3.1 Determination of Inventory Level for SIRPCO1 (Phase I)

As the result of the transformation, the SIRP problem is decomposed into two subproblems: inventory subproblem where the expected inventory cost and the quantity to deliver to each customer are determined, and the routing subproblem which builds the routes and calculates the overall transportation cost.

As discussed in section 5.2, the transportation cost in our model depends on x_{ijt} instead of q_{ijt} , therefore we modify the distance cost (the term of $\sum_{t=1}^{T}\sum_{j=1,j\neq i}^{N}\sum_{i=0}^{N}(\lambda_{ijt}+c_{ij})q_{ijt}$ where λ_{ijt} is the Lagrange multipliers) of inventory subproblem in Yu et al. (2012). In order to modify this, we get the inspiration from the allocation model in Bard and Nananukul (2009). The variable cost term $\sum_{t=1}^{T}\sum_{j=1}^{N}\sum_{i=0}^{N}c_{ij}x_{ijt}$ is replaced by $\sum_{t=1}^{T}\sum_{i=1}^{N}r_{it}^{c}a_{it}$ where r_{it}^{c} is the approximated cost which takes into account the cost of making a delivery to customer *i* directly from the depot divided by the expected demand of customer *i* on period *t*, that is $r_{it}^{c}=\dfrac{2c_{i0}}{t\mu_i}$. We adopted this concept to our subproblem INV and the model s shown below. The model is used to determine the quantity delivery, $a_{it}$and can be formulated as follows:

$$Z(a)=\min\sum_{t=1}^{T}\sum_{i=1}^{N}h_iE(I_{it}^{+})+\sum_{t=1}^{T}\sum_{i=1}^{N}r_{it}^{c}a_{it} \tag{5.16}$$

subject to constraints (5.3), (5.4), (5.5) and (5.9).

However, inspired by Yu et al. (2012), the linearization of $Z(a)$, denoted by $\bar{Z}(a,a^k)$, is thus

$$\bar{Z}(a,a^k) = \min \underbrace{\sum_{t=1}^{T}\sum_{i=1}^{N} h_i \int_0^{I_{i,0}+\sum_{f=1}^{t} a_{if}^k} \left(\left(I_{i,0} + \sum_{f=1}^{t} a_{if}^k - x \right) \left(\frac{e^{-\frac{(x-t\mu)^2}{2\left(\sqrt{t}\sigma\right)^2}}}{\sqrt{t}\sigma\sqrt{2\pi}} \right) dx \right)}_{\mathrm{I}} +$$

$$\underbrace{\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{f=1}^{t} \left(a_{if} - a_{if}^k\right) h_i F_{i,(1,t)} \left(I_{i,0} + \sum_{f=1}^{t} a_{if}^k \right)}_{\mathrm{II}} + \underbrace{\sum_{t=1}^{T}\sum_{i=1}^{N} r_{it}^c a_{it}}_{\mathrm{III}}$$

(5.17)

We note that Part I and II of equation (5.17) are taken from linearization of Yu et al. (2012) while Part III is modified from Bard and Nananukul (2009) which I mentioned in above. The partial linearization method solves a linear programming problem and performs a line search at each iteration. For our problem, at each iteration, k, the method solves the following linear programming problem (denoted by LP^k):

LP^k :

$$\min \bar{Z}(a, a^k)$$

subject to the constraints (5.3), (5.4), (5.5) and (5.9). A line search is then performed to minimize$Z(a)$for subproblem INV,

$$\min_{\rho} \left\{ Z(a) \mid a = \rho a^k + (1-\rho)\bar{a}^k, \quad 0 \le \rho \le 1 \right\} \tag{5.18}$$

where $\bar{a}^k$ is an optimal solution of LP^k.

The model of subproblem INV is solved using Mathematica 7.0. Figure 5.1 illustrates the flow of Phase I.

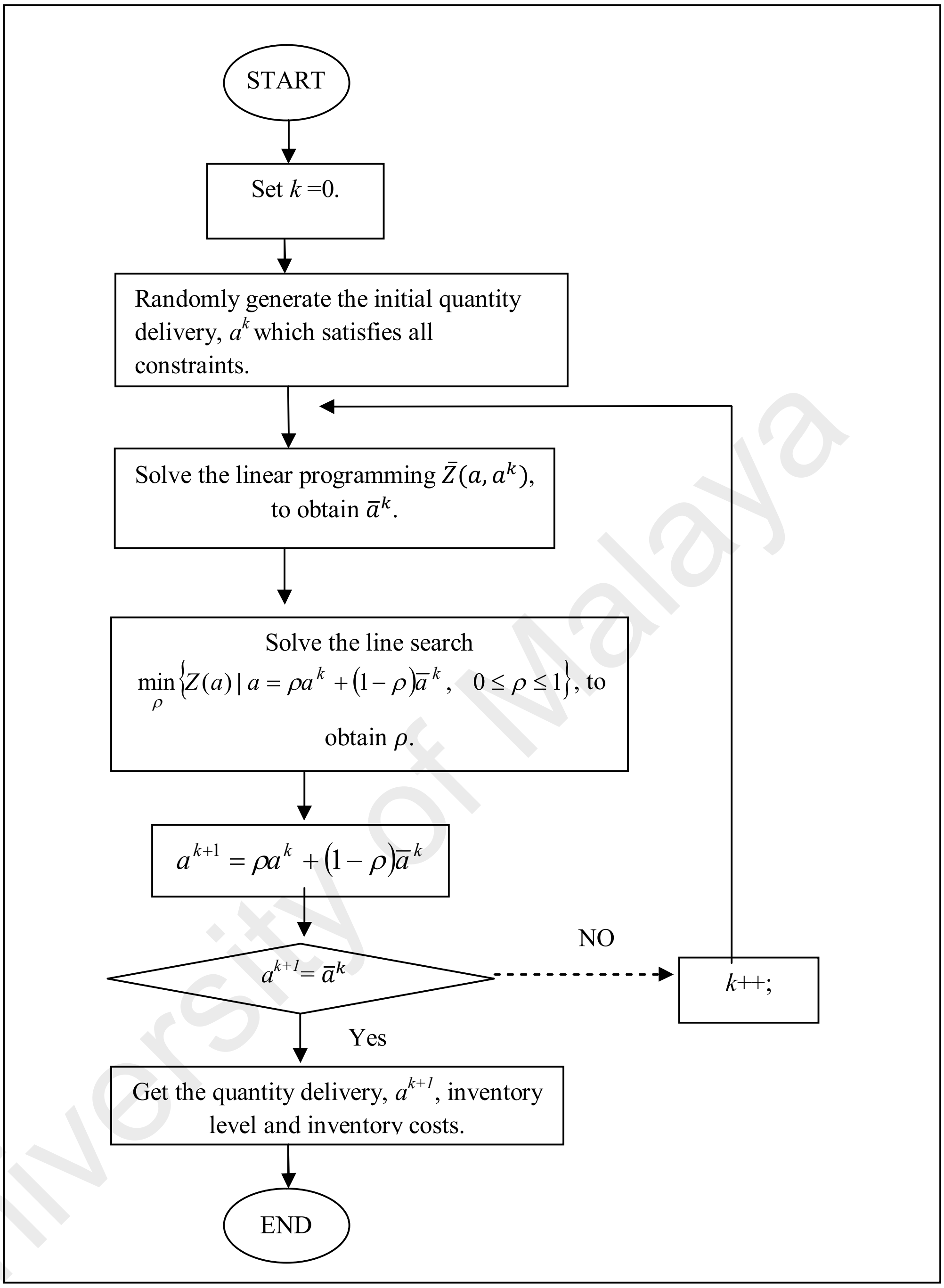

Figure 5.1: Flow chart of Phase I of SIRP algorithm

5.3.2 Determination of Transportation Cost for SIRPACO1 (Phase II)

Population Based ACO is implemented in Phase II in order to build the routes and then obtains the transportation cost. The difference between the previous population based ACO which were discussed in Chapter 4 and the population based ACO in Phase II is that the pheromone values for all arcs in each subpopulation is initialized differently based on the several routing heuristics which we apply. With different initial starting pheromone values for each subpopulation, it gives more chances for the ants to explore in order to attain better results. The detail of Phase II is discussed in the following subsection and Figure 5.2 gives the illustration of the algorithm.

5.3.2.1 Initial Solution

The following algorithms are utilized to create a subpopulation and the total distance obtained by each algorithm is embedded to initializeτ_0, the initial pheromone in the local pheromone updating for each subpopulation. We note that all the algorithms are modified to allow for split delivery.

Nearest Neighbor Algorithm (NN)

In Nearest Neighbor algorithm (NN) the vehicle starts at the depot and repeatedly visits the nearest customer (in terms of distance) until the capacity of the vehicle is fully occupied. Then, a new vehicle is initiated and the process continues until all customers have been assigned or visited.

Sweep Algorithm(Gillett and Miller (1974))

The sweep algorithm which was proposed by Gillett and Miller (1974) requires the transformation from Cartesian coordinates to polar angles in order to determine the direction of sweep, either in clockwise or anti-clockwise. In this study, both directions are considered to form two different subpopulations. The feasible routes are created by rotating a ray centered at the depot and gradually including customers in a vehicle route until the capacity constraint is attained. A new route is then initiated and the process is repeated until the entire plane has been swept (all customers have been visited).

Savings algorithm (Clark and Wright, 1964)

The Clarke and Wright (1964) heuristic is one of the best known and remains widely used in practice in these days. This algorithm is based on the notion of savings. The savings is obtained if two customers are merged on a route with the condition that it does not violate the vehicle capacity constraints. The savings is calculated according to the following equation:

$$s_{ij} = c_{0i} + c_{0j} - c_{ij} \tag{5.19}$$

The savings are arranged in descending order and starting from the top of the list, the algorithm builds a route and the route is expanded by selecting the customer with the most savings to the last node added to the route. The customer which forms a loop is omitted. This selection process is continued until the capacity of the vehicle is fully loaded. Then, a new vehicle is initiated and the process continues until all customers have been assigned or visited.

Savings algorithm (Paessen, 1988)

Paessen (1988) proposed a modification to equation (5.14) which incorporates the customer distance from the depot as follows:

$$s_{ij} = c_{0i} + c_{0j} - \theta_1 c_{ij} + \theta_2 \left| c_{0i} - c_{0j} \right| \tag{5.20}$$

where $\theta_1 \in [1, 3]$ and $\theta_2 \in [0, 1]$. This ensures that the radial distance, the distance between the customers and the depot, is also taken into account. Generally, taking $\theta_1 = 1.5$ and $\theta_2 = 0.5$ give better results. Note that if $\theta_1 = 1.0$ and $\theta_2 = 0.0$ give exactly the savings of Clarke and Wright (1964).

5.3.2.2 Route Construction

The algorithm starts with the predefined number of subpopulation of ants and each subpopulation consists of predefined number of ants to build the solution. Each ant in the subpopulation implements the same mechanism to select the customer to be visited (discussed in subsection 3.3.2). The pheromone values for all arcs in each subpopulation are initialized based on τ_0 generated from five different algorithms (each form a subpopulation).

5.3.2.3 The local pheromone-updating rule

As discussed in the previous chapter, local updating is used to prevent a very strong arc being chosen by all the ants. After each of the ants in every subpopulation has built the solution, the best built solution from each subpopulation is selected. Then, the local updating is done on each arc of the best solution from each subpopulation by using equation (3.15). The different from the previous modification is each subpopulation will have its own value of τ_0 and we set as $\tau_0 = 1/L_{ini}$ where L_{ini} is the total of distance obtained from the five algorithms.

5.3.2.4 The global pheromone-updating rule

As mentioned in the previous chapter, global pheromones updating is done so that the ants will have a better starting point in searching for shorter path. After a predefined number of iterations, the current best solution γ^{gl} among all the subpopulations is selected and its routes are used as a reference for the global pheromones-updating for all subpopulation. Hence, the pheromones value for each arc of the best solution is updated by using ~~the~~ equation (3.16) for all the subpopulations.

5.3.2.5 Route improvement strategies

After a predefined number of iterations, the current best solution among the subpopulations is selected and the routes are further improved by adding route improvement strategies in the route construction procedure which is similar to the previous modification on ACO. Three local searches; namely swap, 2 - *opt** (Potvin and Rousseau, 1995) and 2 – *opt* (Lin, 1965), are applied to improve the solution built by ACO.

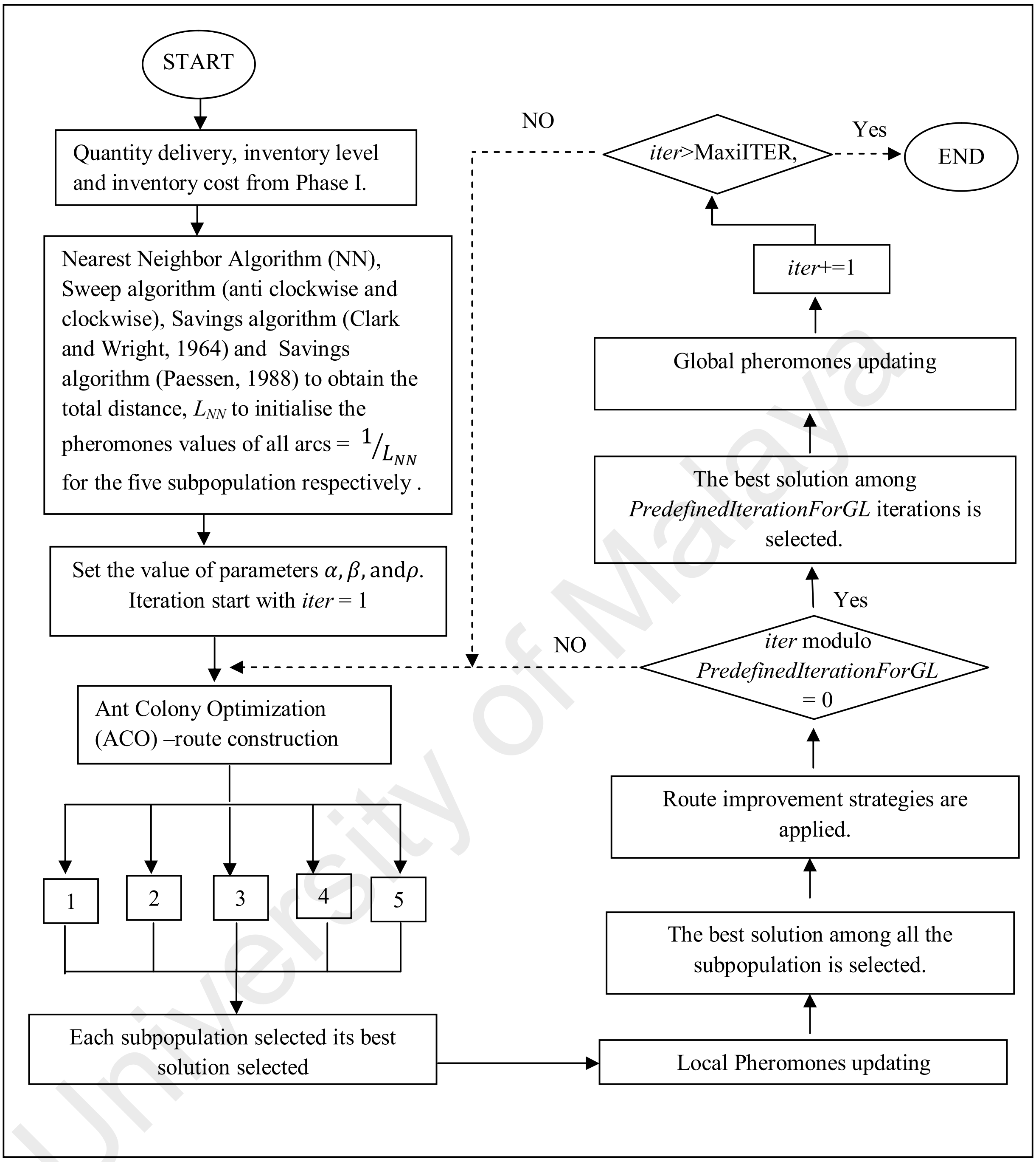

Figure 5.2: Flow chart of Phase II SIRP algorithm - Population Based ACO

5.4 Enhanced Version for the First Algorithm Of SIRP (SIRPACO2)

In our first algorithm of SIRP (i.e. SIRPACO1), Phase II is to determine the transportation cost only. There is no any inventory updating mechanism being included in the phase II. Thus, in this section, the algorithm is enhanced by adding the inventory updating mechanism after every predefined number of iterations in order to produce a set of inventory level that can also minimize the transportation cost and hence reduce the total cost.

The enhanced version of SIRPACO1 is referring to SIRPACO2. Similar to SIRPACO1, SIRPACO2 also consists of two phases where the algorithms of Phase I as well as the routing part in phase II still remain the same algorithm of SIRPACO1. However, we added the inventory updating mechanism in Phase II after every predefined number of iterations in SIRPACO2 in order to obtain the new set of inventory level. After that, the same algorithm of routing part in SIRPACO1 as mentioned above (i.e. population based ACO) is applied to build the routes.

5.4.1 Updating Inventory Mechanism of SIRPACO2

This updating mechanism is focused on backward transfer (i.e. transfer the quantity delivery from period t to $t-1$) based on the savings of the customers. In this mechanism, we proposed a new formulation to calculate the savings of the customers. The savings of the available customers are calculated based on the ratio given below:

$$g_i = \frac{2c_{i0}}{a_{it}h_i}, \quad i = 1,\ldots,N, t = 1,\ldots,T \tag{5.21}$$

The purpose of we formulate our savings of the available customer like shown in equation (5.21) is we want to obtain the largest savings in term of transportation cost without huge increasing in inventory cost if the backward transfer is performed to the customer. Therefore, we tend to choose the available customer who gives the highest savings in this mechanism. The available (feasible) customers are those who do not violet the service level constraint as well as not exceeding the storage capacity at the customers site in period $t - 1$. The updating mechanism is applied in order to produce new set of inventory level after some predefined number of iterations. The savings are sorted in descending order–and customer with the highest savings is selected and the quantity delivery of the selected customer is transferred to period $t - 1$.

5.5 Computational Results

The Phase I of the algorithm is solved using Mathematica 7.0 while Phase II was written in C++ language using Microsoft Visual studio 2008. All the computations were performed on 3.10 GHz processor with 8GB of RAM.

5.5.1 Data sets

Similar to the previous two chapters, Chapter 3 and 4, the algorithm is tested on 12, 20, 50 and 100 customers, and combination with number of periods, 5, 10, 14 and 21. The coordinates for each customer is same with the previous two chapters. The mean and holding costs of the customers are generated randomly from the interval [10, 100] and [0.02, 0.10] respectively. We set the inventory capacity of customers' warehouse is two times of the summation of mean and standard deviation of the customer. Meanwhile, the initial inventory of customer is set as 0.2 of the inventory capacity of customers' warehouse. The vehicle capacity is fixed at 250.0.

5.5.2 Results

In this study, the experiment is carried out using two different parameters for standard deviations which are $0.05\mu_i$ and $0.10\mu_i$. We also consider the service level (α_i and β_i) in different cases at 95% and 99%. As the simplification, our computational experiments are done in the following combination of parameters:

Set A: Standard deviation is $0.05\mu_i$ and service level (α_i and β_i) is 95%

Set B: Standard deviation is $0.05\mu_i$ and service level (α_i and β_i) is 99%

Set C: Standard deviation is $0.10\mu_i$ and service level (α_i and β_i) is 95%

Set D: Standard deviation is $0.10\mu_i$ and service level (α_i and β_i) is 99%

The algorithm in Phase II of SIRPACO1 and SIRPACO2 are run for 1000 iterations and we performed 10 runs for each instance. The algorithm comprises of 5 subpopulations of ants and each subpopulation consists of 5 ants to build a solution. As mentioned in the subsection 5.4.3, the values of τ_0 for each of the subpopulation is different and is set as $\tau_0 = 1/(N \times L_{ini})$. The parameters are set similar as the previous chapter which is $\alpha = 1.0,\ \beta = 5.0,\ q_0 = 0.9, \rho = 0.1$ where the values of α and β are obtained from the sensitivity analysis which have been done in Chapter 3 while the value of q_0 is taken from Dorigo et al. (1996).

Table 5.1 gives the inventory costs produced by the Phase I of the algorithms (discussed in subsection 5.3.1) which is solved using Mathematica 7.0. Table 5.2 and Table 5.3 present the comparison of the best cost under different scenario of parameters for both SIRPACO1 and SIRPACO2 respectively. Meanwhile, Table 5.4 to Table 5.6 give comparisons of the best costs, number of vehicles and CPU time of both algorithms SIRPACO1 and SIRPACO2 respectively. Figure 5.3 presented the line chart for the comparison of the best cost between SIRPACO1 and SIRPACO2.

Table 5.1: Results for the Phase I of the algorithms

Data	Set A		Set B		Set C		Set D	
	Inventory Cost	CPU Time (seconds)	Inventory Cost	CPU Time (seconds)	Inventory Cost	CPU Time (seconds)	Inventory Cost	CPU Time (seconds)
S12T5	31.47	11.65	43.98	11.59	62.81	11.59	87.91	11.58
S12T10	84.26	25.21	117.87	25.15	168.43	25.15	235.61	25.15
S12T14	137.19	37.41	191.97	37.42	274.34	37.33	383.75	37.22
S20T5	46.61	19.42	65.12	19.42	93.01	19.48	130.17	19.34
S20T10	124.81	41.96	174.53	41.95	249.38	41.82	348.86	41.87
S20T14	203.22	62.23	284.23	62.15	406.2	62.23	568.17	61.92
S20T21	367.84	104.85	514.46	102.68	735.33	102.99	1028.52	102.59
S50T5	104.05	48.89	145.51	48.34	207.93	48.27	291	48.20
S50T10	278.8	104.91	390.05	104.64	557.57	104.44	779.95	104.21
S50T14	454.09	156.27	635.19	156.13	908.14	155.98	1270.29	155.27
S50T21	821.91	258.09	1149.8	258.32	1644.05	257.38	2299.39	256.73
S100T5	208.91	96.77	292.15	96.86	417.74	96.52	584.4	96.24
S100T10	559.86	210.94	783.22	209.99	1119.92	209.41	1566.41	209.34
S100T14	912.01	313.33	1275.61	311.84	1824.06	310.35	2551.32	310.97

Table 5.2: Comparison the best cost under different scenario of parameter for SIRPACO1

Data	Set A	Set B	Set C	Set D	Comparison between different service level under same standard deviation (%)		Comparison between different standard deviation under same service level (%)	
	SIRPACO1	SIRPACO1	SIRPACO1	SIRPACO1	Set A vs Set B	Set C vs Set D	Set A vs Set C	Set B vs Set D
S12T5	2229.85	2246.89	2291.32	2387.03	0.76	4.18	2.76	6.24
S12T10	4621.42	4664.41	4765.89	4915.43	0.93	3.14	3.13	5.38
S12T14	6545.03	6622.38	6733.05	6977.28	1.18	3.63	2.87	5.36
S20T5	3033.89	3066.08	3111.72	3220.28	1.06	3.49	2.57	5.03
S20T10	6261.35	6364.12	6436.38	6637.74	1.64	3.13	2.80	4.30
S20T14	8863.66	8952.32	9153.10	9403.55	1.00	2.74	3.27	5.04
S20T21	13475.06	13606.34	13845.14	14327.33	0.97	3.48	2.75	5.30
S50T5	7025.83	7236.82	7427.21	7781.55	3.00	4.77	5.71	7.53
S50T10	14463.84	14863.14	15244.53	15875.23	2.76	4.14	5.40	6.81
S50T14	20473.01	20853.28	21585.60	22307.15	1.86	3.34	5.43	6.97
S50T21	31316.83	31719.25	32420.72	33767.77	1.28	4.15	3.52	6.46
S100T5	13311.80	13613.71	13929.96	14414.11	2.27	3.48	4.64	5.88
S100T10	27690.33	28215.16	28879.97	29744.56	1.90	2.99	4.30	5.42
S100T14	39228.36	40102.88	40968.12	42101.16	2.23	2.77	4.43	4.98

Table 5.3: Comparison the best cost under different scenario of parameter for SIRPACO2

Data	Set A	Set B	Set C	Set D	Comparison between different service level under same standard deviation (%)		Comparison between different standard deviation under same service level (%)	
	SIRPACO2	SIRPACO2	SIRPACO2	SIRPACO2	Set A vs Set B	Set C vs Set D	Set A vs Set C	Set B vs Set D
S12T5	2134.98	2107.68	2127.49	2270.81	-1.28	6.74	-0.35	7.74
S12T10	4410.72	4341.61	4463.43	4606.11	-1.57	3.20	1.19	6.09
S12T14	6200.64	6274.23	6298.93	6528.91	1.19	3.65	1.59	4.06
S20T5	3030.00	3050.20	3091.16	3216.32	0.67	4.05	2.02	5.45
S20T10	6267.12	6341.67	6409.16	6608.25	1.19	3.11	2.27	4.20
S20T14	8891.56	9005.57	9115.88	9391.38	1.28	3.02	2.52	4.28
S20T21	13493.10	13575.07	13893.74	14347.32	0.61	3.26	2.97	5.69
S50T5	7096.07	7243.89	7500.81	7693.11	2.08	2.56	5.70	6.20
S50T10	14656.73	14903.18	15312.57	15915.06	1.68	3.93	4.47	6.79
S50T14	20675.90	20853.62	21623.14	22434.00	0.86	3.75	4.58	7.58
S50T21	31400.10	31967.82	32789.14	34031.33	1.81	3.79	4.42	6.45
S100T5	13348.97	13610.98	13900.83	14427.04	1.96	3.79	4.13	6.00
S100T10	27937.11	28104.59	28918.76	29661.09	0.60	2.57	3.51	5.54
S100T14	39246.50	40072.23	40965.92	42153.10	2.10	2.90	4.38	5.19

Table 5.4: Comparison of the best costs of SIRPACO1 and SIRPACO2

Data	Set A			Set B			Set C			Set D		
	SIRPACO1	SIRPACO2	Gaps (%)	SIRPACO1	SIRPACO2	Gaps (%)	SIRPACO1	SIRPACO2	Gaps (%)	SIRPACO1	SIRPACO2	Gaps (%)
S12T5	2229.85	2134.98	-4.25	2246.89	2107.68	-6.20	2291.32	2127.49	-7.15	2387.03	2270.81	-4.87
S12T10	4621.42	4410.72	-4.56	4664.41	4341.61	-6.92	4765.89	4463.43	-6.35	4915.43	4606.11	-6.29
S12T14	6545.03	6200.64	-5.26	6622.38	6274.23	-5.26	6733.05	6298.93	-6.45	6977.28	6528.91	-6.43
S20T5	3033.89	3030.00	-0.13	3066.08	3050.20	-0.52	3111.72	3091.16	-0.66	3220.28	3216.32	-0.12
S20T10	6261.35	6267.12	0.09	6364.12	6341.67	-0.35	6436.38	6409.16	-0.42	6637.74	6608.25	-0.44
S20T14	8863.66	8891.56	0.31	8952.32	9005.57	0.59	9153.10	9115.88	-0.41	9403.55	9391.38	-0.13
S20T21	13475.06	13493.10	0.13	13606.34	13575.07	-0.23	13845.14	13893.74	0.35	14327.33	14347.32	0.14
S50T5	7025.83	7096.07	1.00	7236.82	7243.89	0.10	7427.21	7500.81	0.99	7781.55	7693.11	-1.14
S50T10	14463.84	14656.73	1.33	14863.14	14903.18	0.27	15244.53	15312.57	0.45	15875.23	15915.06	0.25
S50T14	20473.01	20675.90	0.99	20853.28	20853.62	0.00	21585.60	21623.14	0.17	22307.15	22434.00	0.57
S50T21	31316.83	31400.10	0.27	31719.25	31967.82	0.78	32420.72	32789.14	1.14	33767.77	34031.33	0.78
S100T5	13311.80	13348.97	0.28	13613.71	13610.98	-0.02	13929.96	13900.83	-0.21	14414.11	14427.04	0.09
S100T10	27690.33	27937.11	0.89	28215.16	28104.59	-0.39	28879.97	28918.76	0.13	29744.56	29661.09	-0.28
S100T14	39228.36	39246.50	0.05	40102.88	40072.23	-0.08	40968.12	40965.92	-0.01	42101.16	42153.10	0.12

Note: Gaps refers to the difference between the obtained results for SIRPACO1 and SIRPACO2

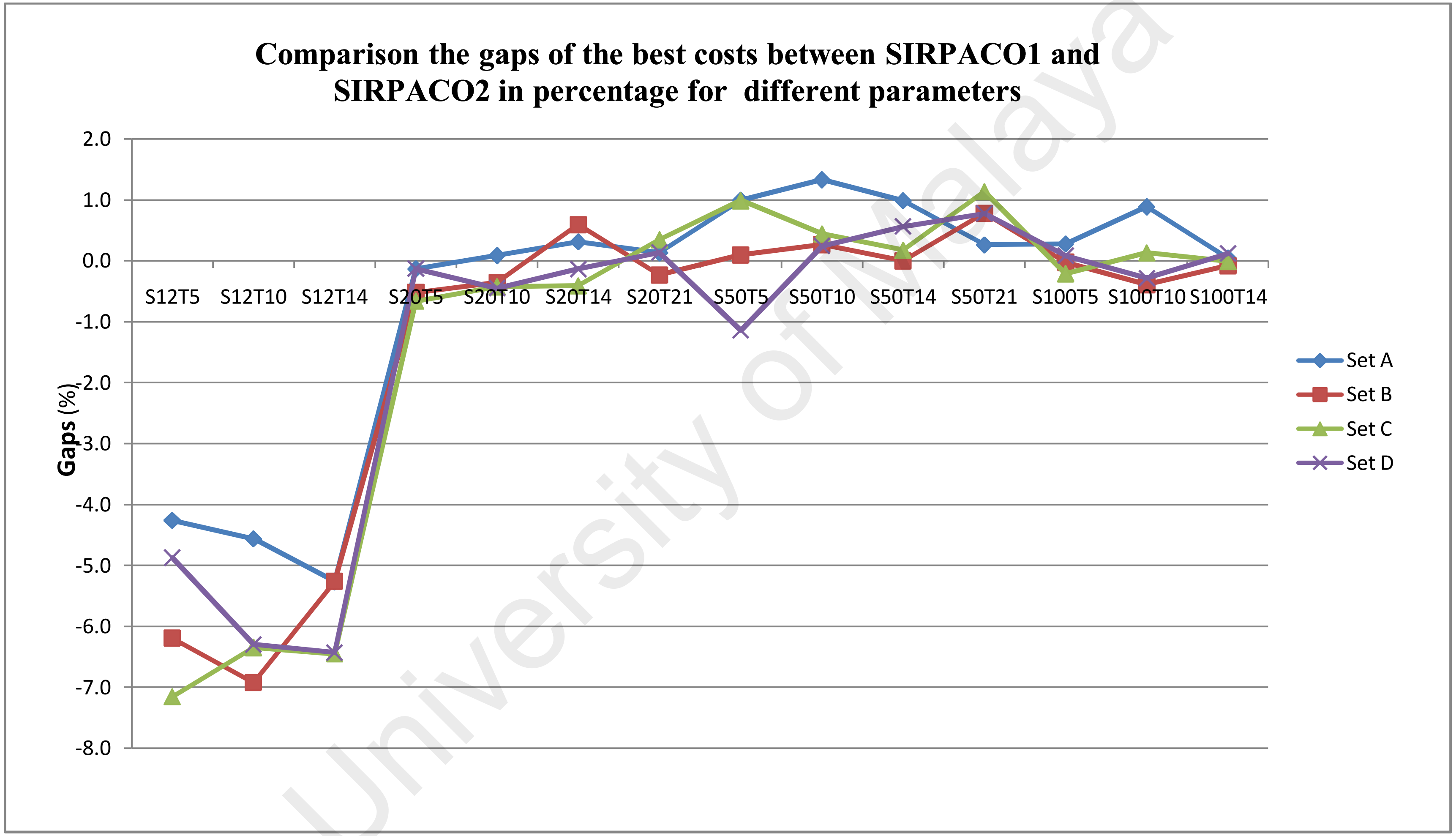

Figure 5.3: Line chart for comparison the best cost between SIRPACO1 and SIRPACO2 in percentage.

Table 5.5: Comparison of the number of vehicles of SIRPACO1 and SIRPACO2

Data	Set A		Set B		Set C		Set D	
	SIRPACO1	SIRPACO2	SIRPACO1	SIRPACO2	SIRPACO1	SIRPACO2	SIRPACO1	SIRPACO2
S12T5	18	16	18	16	18	16	19	17
S12T10	38	33	38	34	38	35	39	35
S12T14	54	47	54	48	54	48	55	49
S20T5	23	23	23	23	23	23	24	24
S20T10	48	48	48	48	48	48	49	49
S20T14	68	68	68	68	68	68	69	69
S20T21	103	103	103	103	103	103	104	104
S50T5	50	50	51	51	52	52	56	54
S50T10	105	105	106	106	107	108	111	111
S50T14	149	149	150	150	151	151	155	155
S50T21	226	226	227	227	228	228	232	232
S100T5	100	99	101	101	103	103	106	106
S100T10	210	209	211	211	213	213	216	216
S100T14	295	295	299	299	301	301	304	303

Table 5.6: Comparison of the CPU Time of SIRPACO1 and SIRPACO2

Data	Set A		Set B		Set C		Set D	
	SIRPACO1	**SIRPACO2**	**SIRPACO1**	**SIRPACO2**	**SIRPACO1**	**SIRPACO2**	**SIRPACO1**	**SIRPACO2**
S12T5	15.65	15.65	15.59	15.59	15.59	15.59	15.58	15.58
S12T10	32.21	32.21	32.15	32.15	32.15	32.15	32.15	32.15
S12T14	47.41	47.41	47.42	47.42	47.33	47.33	47.22	47.22
S20T5	29.42	29.42	29.42	29.42	29.48	29.48	29.34	29.34
S20T10	62.96	61.96	61.95	60.95	61.82	61.82	61.87	61.87
S20T14	89.23	89.23	91.15	89.15	89.23	89.23	88.92	88.92
S20T21	145.85	144.85	143.68	142.68	143.99	143.99	143.59	142.59
S50T5	125.89	122.89	125.34	124.34	124.27	123.27	125.20	124.20
S50T10	256.91	253.91	256.64	253.64	255.44	253.44	255.21	253.21
S50T14	368.27	369.27	368.13	371.13	366.98	364.98	366.27	368.27
S50T21	574.09	575.09	575.32	575.32	572.38	573.38	571.73	573.73
S100T5	472.77	467.77	475.86	466.86	473.52	466.52	473.24	466.24
S100T10	957.94	954.94	963.99	950.99	959.41	954.41	960.34	956.34
S100T14	1361.33	1355.33	1355.84	1349.84	1356.35	1348.35	1356.97	1352.97

5.5.3 Discussion

From the results shown in Table 5.1, we observe that the inventory cost increases when increasing either the standard deviation or service level (α_i and β_i).The comparison of the best cost (in percentage) under different scenario of the parameters for both SIRPACO1 and SIRPACO2 are given in Table 5.2 and Table 5.3 respectively. Table 5.2 and Table 5.3 tabulate the results for different scenarios of service level (α_i and β_i) when the standard deviation is fixed. The results indicate that as the values of α_i and β_i the total cost increases as well. Similar trend is observed when we vary the standard deviations keeping the service level constraint constant. However, the percentage increase is observed to be more when the standard deviation is increased compared to the increase of the service level constraints. Based on the results which we obtained, we can concluded that the data set with less standard deviation and service level (α_i and β_i) is 95% will give less total cost (best cost). Therefore, we can expect that the algorithms perform better if the standard deviation is small and also in order to obtain better results we have to set the service level constraints to 95%.

Table 5.4 gives the comparison of the best cost between the two methods, SIRPACO1 and SIRPACO2. It is observed that SIRPACO2 outperformed in most instances when compared with SIRPACO1 i.e. 9 out of 14 problem instances in Set B, 8 out of 14 problem instances in Set C and D. Although SIRPACO2 did not performed better in medium and larger problem instances in Set A, however the gaps between the two methods are small which are less than 1.5%. The gaps are illustrated in Figure 5.3. In addition, for sets B, C and D it is also observed that for those instances which SIRPACO1 outperformed SIRPACO2, the gaps are also comparatively small that is less than 2%. As a conclusion, the overall performance of SIRPACO2 is better than SIRPACO1.

Table 5.5 showed the number of vehicles utilized for both methods, SIRPACO1 and SIRPACO2 of all 4 different scenarios. From the results, we can notice that SIRPACO2 is able to reduce the number of vehicles for small instances, S12. Meanwhile, the remaining instances (i.e. S20, S50 and S100) utilized the same number of vehicles to produce the solution as in Table 5.5. It may due to higher quantity delivery of the customers (i.e. S20, S50 and S100) and subsequently maximum utilization of the vehicles, thus causing more challenges in collapsing the number of vehicles. Therefore, there is no improvement in term of the utilization number of vehicles for the instances; S20, S50 and S100 even though SIRPACO2 are able to produce better results (i.e. minimum cost) in most of the instances. Table 5.6 gave the CPU time for both SIRPACO1 and SIRPACO2. From the results, we can notice that both methods require relatively short time to obtain the results.

5.6 Summary of the Chapter

In this chapter, we extended our model to tackle the stochastic inventory routing problem (SIRP). Different from the previous two chapters where the customer's demand is deterministic and time varying, the demand of customer in SIRP is only known in a probabilistic sense. We consider a multi-period SIRP with split delivery (SIRPSD) where a fleet of homogeneous capacitated vehicles housed at a depot/warehouse transport the multi-product to geographically dispersed customers to fulfill their demand. The total inventory and transportation cost are the two main components to be minimized. In this study, we implement service level constraints to ensure that the stock out cost is not too excessive and also to prevent overloading at the customer's warehouse.

We developed two-phase algorithms, SIRPACO1 to solve the proposed SIRP model where phase I solved the inventory sub problem to give inventory costs and quantity delivery while phase II played the role of building routes from the input attained from phase I in order to obtain the transportation costs. In addition, we also developed enhanced version of SIRPACO1, which is referred to SIRPACO2 by incorporating the inventory element in the updating mechanism in Phase II to obtain better results. In this updating mechanism, we proposed a new formulation to calculate the savings for each feasible customer and the selection of the customer is based on the highest savings the backward transfer.

We test the algorithm using different combinations of the service level and standard deviation parameters for the computational experiments. It is observed that the results are affected by the variation in the standard deviation rather than the service level parameters. The overall results show that we can conclude that SIRPACO2 outperformed SIRPACO1 in most of the instances. Moreover, SIRPACO2 also have the potential of reducing the utilization of number of vehicles. Both SIRPACO1 and SIRPACO2 required relatively small CPU time.

CHAPTER 6: CONCLUSION AND FUTURE RESEARCH

This chapter will conclude our findings in this study. The chapter starts with the discussion regarding the conclusion of this study. This chapter will also present the related future research pertaining enhancement of the algorithms in order to get better results or the potential extension of the model in order to tackle another IRP problem of the industries.

6.1 Summary of thesis

Supply chain management (SCM) plays the role of managing the flow of goods and services that takes into consideration the movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption. Inventory management and transportation are two important components in supply chain management.

We considered one-to-many network for a finite planning horizon which consists of a manufacturer that produces multi-products to be delivered by a fleet of capacitated homogeneous vehicles, housed at a depot/warehouseto a set of geographically dispersed customers. The demand for each product is deterministic and time varying and each customer requests a distinct product. Split delivery is allowed in our model.

In this study, we modified the conventional Ant Colony Optimization (ACO)by adding the inventory cost in the global pheromones updating to build the routes. Meanwhile, the procedure of updating inventory level is done by implementing the deterministic backward transfer to determine the inventory for customers. In addition, we also developed a new transfer/swap aimed at combining split customers in order to attain the improvement in term of the transportation costs. The computational experiments are done and the overall results showed that our developed algorithms-performed better in larger instances if compared with small and medium instances. The results obtained for larger instances are better than the upper bound which was generated from CPLEX 12.4. Meanwhile, from the two algorithms that we developed namely, ACO and ACO2, we found out that ACO2 gives better results than ACO for all the problem instances. We can also observe that ACO2 gives better solution quality through the standard deviation of the total costs that we obtained. Sensitivity analyses on various parameters were also performed with the aim to attain appropriate parameter settings which can give better results.

In the conventional ACO, only one population of ants is used to build the solution. In this study, we enhanced ACO by dividing the ants into subpopulations in order to solve the problem and this is referred to as Population Based ACO. We design a population based ACO by segregating each subpopulation using the inventory level. We proposed several different algorithms for the population-based ACO. We called our first proposed algorithm of population based ACO as ACOBF which incorporates the random and deterministic inventory updating mechanism where the forward and backward inventory updating, biasing towards the forward inventory updating. ACOBF starts with initial population of zero inventories. However, the emphasis in ACOBF is still more on routing part than inventory. Thus, we proposed our second algorithms of population based ACO where new formulation called pheromones of customer's inventory is developed. The new inventory updating mechanism where it takes into account the local and global pheromone of customer's inventory updating is also proposed in this study. The selection of customers for the transfer is based on the attraction. We called this algorithm as ACOPher. Similar to ACOBF, ACOPher starts with initial population of zero inventories. We propose an enhancement to the algorithm where we let each subpopulation starts with different set of inventory level including zero inventory, inventory level generated using the allocation model and randomly generated inventory levels. We called the enhanced algorithms as ACOPher2.

Similar to ACO and ACO2, all the algorithms developed for population based ACO are tested on the data with different combinations of the number of customers, 12, 20, 50 and 100 with the number of periods, 5, 10, 14 and 21. The overall results for the algorithms ACOBF, ACOPher and ACOPher2 show that the algorithms performed better in larger instances if compared with small and medium instances. Moreover, we also can observe that ACOPher2 gave better results if compared with ACOBF as well as ACOPher in most of the instances. ACOPher as well as ACOPher2 give better in term of quality solution if compared with ACOBF based on the obtained standard deviation of the total costs. From the observation of the obtained result, we can conclude that to implement the selection of customer for transferring based on the attraction are efficient of getting minimum costs.

In this study, we also extended our deterministic model to tackle the stochastic inventory routing problem (SIRP) where the demand of customer is unknown in advance but known in a probabilistic sense. We considered a multi-period SIRP with split delivery (SIRPSD) comprising of a depot/warehouse where a fleet of homogeneous capacitated vehicles to transport the products to a set of customers is located. We implemented service level constraints to prevent stock out cost and overloading at the customer's storage. We developed two-phase algorithms, SIRPACO1 to solve our proposed SIRP model where phase I solved the inventory sub problem to produce inventory costs and quantity delivery. Meanwhile, phase II built routes for the quantity delivery obtained from phase I in order to calculate the transportation costs. The proposed SIRPACO1 was further enhanced by incorporating inventory updating mechanism into Phase II with the aim of producing a good set of inventory level which will give minimum overall cost. In the updating mechanism, a new formulation was proposed to calculate the savings for each customer and the selection of a customer is biased towards customers with higher savings in the backward transfer. In order to test the efficiency of our algorithms, combinations of different parameters were carried out for the service level and standard deviation. It was observed that increasing the standard deviation is more likely to produce higher cost when compared with increasing service level (α_i and β_i) parameters. In addition, we found out that SIRPACO2 performed better than SIRPACO1 in most of the instances and also have the potential to decrease the number of vehicles. Based on CPU time that we attained, we also note that both algorithms SIRPACO1 and SIRPACO2 required relatively short time to build the solution.

6.2 Future Research

In this study, our proposed model on deterministic IRP considered that the customers' demand can be met on time and backordering / backlogging is not allowed. However, there are the possibilities that the demand of the customers cannot be fulfilled on time. This may due either to the overall demand of the customers are beyond the production capacity of the company or it may be cheaper to serve the customers in the next period (due to the limited capacity of the vehicles). This is called backordering. Generally, the penalties fees will be charged for those demands that cannot be fulfilled on time. Normally the backordering is limited to several period only. An increasing in backlogged order of the products can show the rising in sales but also may indicate the inefficiency of the production management. Therefore, the goal of balancing between the inventory management and backordering / backlogging is the challenge problem in

IRP. In future research, we can extend our proposed model to tackle the problem to include backordering / backlogging and/or lost sales. In addition we also can add the production components into our model with the aim of balancing between the inventory and production management.

Similar to our proposed model of SIRP, we can also extend it to tackle the problem which appending backordering. We also can enhance our developed SIRP algorithm by selecting appropriate weights or incorporating more powerful algorithm into our proposed ACO for solving SIRP to balance between transportation and inventory costs. However selecting appropriate weights is very complex.

REFERENCES

Abdelmaguid, T. F. (2004), Heuristic approaches for the integrated inventory distribution problem, PhD Dissertation, University of Southern California, Los Angeles.

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.

Adelman, D. (2004). A price-directed approach to stochastic inventory/routing. *Operations Research, 52*(4), 499-514.

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.

Anily, S., & Bramel, J. (2004). A probabilistic analysis of a fixed partition policy for the inventory-routing problem. *Naval Research Logistics (NRL), 51*(7), 925-948.

Anily, S., & Federgruen, A. (1990). One warehouse multiple retailer systems with vehicle routing costs. *Management Science, 36*(1), 92-114.

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.

Bard, J. F., & Nananukul, N. (2009). The integrated production–inventory–distribution–routing problem. *Journal of Scheduling, 12*(3), 257-280.

Bard, J. F., & Nananukul, N. (2010). A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research, 37*(12), 2202-2217.

Bell, J. E., & McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics, 18*(1), 41-48.

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.

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., 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., Speranza, M. G., & Ukovich, W. (1997). Minimization of logistic costs with given frequencies. *Transportation Research Part B: Methodological, 31*(4), 327-340.

Brezina Jr, I., & Čičková, Z. (2011). Solving the Travelling Salesman Problem Using the Ant Colony Optimization. *Management Information Systems, 6*(4), 010-014.

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). Applying the ant system to the vehicle routing problem *Meta-Heuristics* (pp. 285-296): Springer.

Calvete, H. I., Galé, C., & Oliveros, M.-J. (2011). Bilevel model for production–distribution planning solved by using ant colony optimization. *Computers & Operations Research, 38*(1), 320-327.

Campbell, A. M., & Savelsbergh, M. W. (2004). A decomposition approach for the inventory-routing problem. *Transportation Science, 38*(4), 488-502.

Chen, C.-H., & Ting, C.-J. (2006). An improved ant colony system algorithm for the vehicle routing problem. *Journal of the Chinese institute of industrial engineers, 23*(2), 115-126.

Chien, T. W., Balakrishnan, A., & Wong, R. T. (1989). An integrated inventory allocation and vehicle routing problem. *Transportation Science, 23*(2), 67-76.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568-581.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2013a). Thirty years of inventory routing. *Transportation Science, 48*(1), 1-19.

Coelho, L. C., & Laporte, G. (2013b). A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research, 51*(23-24), 7156-7169.

Coelho, L. C., & Laporte, G. (2013c). The exact solution of several classes of inventory-routing problems. *Computers & Operations Research, 40*(2), 558-565.

Cordeau, J.-F., G. Laporte, M. W. P. Savelsbergh, D. Vigo. (2007). Vehicle routing. C. Barnhart, G. Laporte, eds. *Transportation, Handbooks in Operations Research and Management Science*, Vol. 14. Elsevier, Amsterdam, 367–428.

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.

Desaulniers, G., Rakke, J. G., & Coelho, L. C. (2015). A branch-price-and-cut algorithm for the inventory-routing problem. *Transportation Science, 50*(3), 1060-1076.

Di Caro, G., & Dorigo, M. (1999). *Ant colony optimization: a new meta-heuristic.* Paper presented at the Proceedings of the 1999 Congress on Evolutionary Computation.

Dong, Y., & Xu, K. (2002). A supply chain model of vendor managed inventory. *Transportation Research Part E: Logistics and Transportation Review, 38*(2), 75-95.

Dorigo, M. (1992). *Learning and natural algorithms.* (PhD dissertation), Politecnico di Milano, Milan, Italy.

Dorigo, M., & Birattari, M. (2007). Swarm intelligence. *Scholarpedia, 2*(9), 1462. doi:10.4249/scholarpedia.1462

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical computer science, 344*(2), 243-278.

Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. *BioSystems, 43*(2), 73-81.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 26*(1), 29-41.

Dror, M., & Trudeau, P. (1989). Savings by split delivery routing. *Transportation Science*, 141-145.

Dror, M., & Trudeau, P. (1990). Split delivery routing. *Naval Research Logistics (NRL), 37*(3), 383-402.

Federgruen, A., Prastacos, G., & Zipkin, P. H. (1986). An allocation and distribution model for perishable products. *Operations Research, 34*(1), 75-82.

Federgruen, A., & Zipkin, P. (1984). A combined vehicle routing and inventory allocation problem. *Operations Research, 32*(5), 1019-1037.

Fisher, M. L., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks, 11*(2), 109-124.

Gillett, B. E., & Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research, 22*(2), 340-349.

Hlaing, Z. C. S. S., & Khine, M. A. (2011). Solving traveling salesman problem by using improved ant colony optimization algorithm. *International Journal of Information and Education Technology, 1*(5), 404.

Huacuja, H. J. F., Valdez, G. C., Rangel, R. A. P., Barbosa, J. G., Reyes, L. C., Valadez, J. M. C., . . . Villanueva, D. T. (2013). Scatter search with multiple improvement methods for the linear ordering problem. *Malaysian Journal of Computer Science, 25*(2).

Huang, S.-H., & Lin, P.-C. (2010). A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review, 46*(5), 598-611.

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.

Lim, C., Jain, L., & Dehuri, S. (2009). Innovations in Swarm Intelligence: Studies in Computational Intelligence, Vol. 248: Springer.

Lin, S. (1965). Computers solutions of the traveling salesman problem. *Bell System Technical Journal,* 44, 2245–2269.

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.

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. (2011). An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. *International Journal of Production Economics, 133*(1), 334-343.

Moon, C., Seo, Y., Yun, Y., & Gen, M. (2006). Adaptive genetic algorithm for advanced planning in manufacturing supply chain. *Journal of Intelligent Manufacturing, 17*(4), 509-522.

Paessens, H. (1988). The savings algorithm for the vehicle routing problem. *European Journal of Operational Research, 34*(3), 336-344.

Panigrahi, B., Shi, Y., & Lim, M. (2011). Handbook of Swarm Intelligence. Series: Adaptation, Learning, and Optimization: Berlin: Springer-Verlag.

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.

Qu, W. W., Bookbinder, J. H., & Iyogun, P. (1999). An integrated inventory–transportation system with modified periodic policy for multiple products. *European Journal of Operational Research, 115*(2), 254-269.

Rasheed, A., Ajmal, S., & Qayyum, A. (2014). adaptive routing update approach for VANET using local neighbourhood change information. *Malaysian Journal of Computer Science, 27*(4).

Sin, O. C., Moin, N. H., & Omar, M. (2013). Multi Parents Extended Precedence Preservative Crossover For Job Shop Scheduling Problems. *Malaysian Journal of Computer Science, 26*(3).

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research, 35*(2), 254-265.

Solyali, O., & Süral, H. (2011). A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory-routing problem. *Transportation Science, 45*(3), 335-345.

Solyalı, O., & Süral, H. (2008). A single supplier–single retailer system with an order-up-to level inventory policy. *Operations Research Letters, 36*(5), 543-546.

Speranza, M., & Ukovich, W. (1996). An algorithm for optimal shipments with given frequencies. *Naval Research Logistics (NRL), 43*(5), 655-671.

Tatsis, V. A., Parsopoulos, K. E., Skouri, K., & Konstantaras, I. (2013). An ant-based optimization approach for inventory routing *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV* (pp. 107-121): Springer.

Yang, G., & Liu, Y. (2013). Designing fuzzy supply chain network problem by mean-risk optimization method. *Journal of Intelligent Manufacturing, 26*(3), 447-458.

Yu, B., Yang, Z.-Z., & Yao, B. (2009). An improved ant colony optimization for vehicle routing problem. *European Journal of Operational Research, 196*(1), 171-176.

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.

LIST OF PUBLICATIONS

Wong, L., & Moin, N. H. (2017). Ant Colony Optimization For Split Delivery Inventory Routing Problem. *Malaysian Journal of Computer Science*, *30*(4), 333-348.

Wong, L., & Moin, N. H. (2015, October). Enhanced ant colony optimization for inventory routing problem. In *THE 22ND NATIONAL SYMPOSIUM ON MATHEMATICAL SCIENCES (SKSM22): Strengthening Research and Collaboration of Mathematical Sciences in Malaysia* (Vol. 1682, p. 030007). AIP Publishing.

Wong, L., & Moin, N. H. (2014, December). Ant colony optimization for one-to-many network inventory routing problem. In *Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on* (pp. 877-881). IEEE.

Wong, L., & Moin, N. H. (2014, July). A modified ant colony optimization to solve multi products inventory routing problem. In *PROCEEDINGS OF THE 21ST NATIONAL SYMPOSIUM ON MATHEMATICAL SCIENCES (SKSM21): Germination of Mathematical Sciences Education and Research towards Global Sustainability* (Vol. 1605, pp. 1117-1122). AIP Publishing.