AN INTEGRATED INVENTORY CONTROL AND VEHICLE ROUTING PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OĞUZ SOLYALI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

AUGUST 2005

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Çağlar Güven
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Haldun Süral
Supervisor

Examining Committee Members

Prof. Dr. Ömer Kırca	(METU,IE)	
Asst. Prof. Dr. Haldun Süral	(METU,IE)	
Assoc. Prof. Dr. Yasemin Serin	(METU,IE)	
Dr. Ayten Türkcan	(METU,IE)	
Mehmet Kayhan	(ASELSAN)	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Oğuz SOLYALI

Signature :

ABSTRACT

AN INTEGRATED INVENTORY CONTROL AND VEHICLE ROUTING PROBLEM

Solyalı, Oğuz
M.Sc., Department of Industrial Engineering
Supervisor: Asst. Prof. Dr. Haldun Süral

August 2005, 98 pages

In this study, we consider a logistics system, in which a single supplier delivers a product to multiple retailers over a finite time horizon. Supplier decides on the amount to order in each period and services retailers facing deterministic dynamic demand via a fleet of vehicles having limited capacity. Each retailer has specific minimum and maximum levels of inventory in an order-up-to level inventory policy setting. The problem is to simultaneously determine the quantity of product to order to the supplier, retailers to be visited, the quantity of product to be delivered to retailers and routes of vehicles in each period so as to minimize system-wide costs. We present a mathematical formulation for the problem, for which we develop several Lagrangian relaxation based solution procedures providing both upper and lower bounds to the problem. We implement these solution procedures on test instances and present the results. Computational study shows that our solution procedures generate good feasible solutions in reasonable time.

Keywords: Inventory Routing Problem, Order-up-to Level Inventory Policy, Lagrangian Relaxation

ÖZ

BÜTÜNLEŞİK BİR ENVANTER KONTROL VE ARAÇ ROTALAMA PROBLEMİ

Solyalı, Oğuz

Yüksek Lisans, Endüstri Mühendisliği Bölümü

Tez Yöneticisi: Y. Doç. Dr. Haldun Süral

Ağustos 2005, 98 sayfa

Bu çalışmada bir tedarikçinin tek ürünü perakendecilere bir planlama ufku boyunca dağıttığı bir lojistik sistem ele alınmıştır. Tedarikçi her dönem ne kadar mal sipariş edeceğine karar vermekte ve deterministik dinamik taleple karşılaşan perakendecileri belirli bir kapasiteye sahip bir araç filosu ile ziyaret etmektedir. Her perakendeci belirli bir seviyeye kadar ısmarlamalı envanter politikası ile çalışmaktadır ve bu bağlamda herbir perakendecinin önceden belirlenmiş bir taban ve tavan envanter seviyesi vardır. Problem, tedarikçinin hangi zamanda ne kadar mal sipariş vereceğine ve hangi perakendecilerin, hangi zamanda ve hangi sırayla ziyaret edileceğine sistemdeki maliyetleri enazlayarak eş zamanlı olarak karar verilmesidir. Bu problem için matematiksel bir formülasyon önerilmiş ve bu formülasyon için Lagrange gevşetme yaklaşımına dayalı, problem için hem üst sınır hem de alt sınır sağlayan çözüm yordamları geliştirilmiştir. Çözüm yordamları literatürden alınan problemlerle test edilmiş, elde edilen sonuçlar çalışmada verilmiştir. Sayısal deneyler çözüm yordamlarımızın makul sürelerde iyi çözümler verdiğini göstermiştir.

Anahtar Kelimeler: Envanter-Rotalama Problemi, Belirli Bir Seviyeye Kadar Ismarlamalı Envanter Politikası, Lagrange Gevşetimi

To my family and my sweetheart

ACKNOWLEDGMENTS

I would like to extend my deepest gratitude to my supervisor Asst. Prof. Dr. Haldun Süral for his guidance, suggestions and support. Working with him was not only challenging, but also a fun for me. Whenever I got stuck in this study, he always had an answer or an idea in his mind that enabled me to proceed. Completion of this thesis would not be possible without his guidance.

A number of individuals have contributed in some way to this study. I am sincerely grateful to Erdem Demir, Bora Kat, Fatma Kılınç, Ayşe Koçlar, Halil Özbey, Melih Özlen, Selin Özpeynirci and Banu Soylu. In addition, although their names do not appear here, I would like to thank my professors and colleagues at the department for the stimulating environment they have created.

I am greatly indebted to my family for their endless support throughout my study. Last but not least, I would like to extend my gratitude to Dilek Çelik for her patience, support and presence. Your existence always inspires me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

This study is concerned with the integrated inventory management and vehicle routing problem, commonly referred to as the Inventory Routing Problem (IRP) in the literature. There exist a single supplier and multiple retailers. Retailers are supplied by the supplier via a fleet of vehicles departing from the supplier and visiting retailers on the route to make the deliveries. This problem, which arises frequently in distribution of industrial gases, soft drinks, replenishment of vending machines, etc., is to make inventory control and distribution routing decisions simultaneously so as to exploit benefits of coordination such as reduced costs. Distinguishing features of this study are the deterministic order-up-to level inventory policy employed at the retailers and the consideration of the supplier's order policy. Thus, the problem addressed in this study, referred to as the Inventory Routing Problem with Order-up-to level policy (IRO), is to decide on replenishment amounts both at the supplier and retailers and routes of vehicles over a multi-period time horizon subject to deterministic dynamic demand as well as order-up-to level policy at the retailers.

Our aim in this study is to address the IRO and develop a solution method, which yields good quality solutions in reasonable time.

1.1 Motivation

Over the past few years, vendor managed inventory (VMI) systems have become a popular trend in logistics management (Campbell et al., 1998). In VMI systems, customers (e.g., retailers) do not place orders to the vendor (e.g., supplier), instead the vendor decides on when to replenish customers, thus, manages inventories of its customers. This provides some gains to both customers and the vendor in that customers do not have to deal with inventory management, while the vendor is able to better plan deliveries which enables it to efficiently use its resources.

VMI systems entail coordination among different companies as well as coordination among different echelons of the same company. Coordination is the key issue to achieve reduced operational costs. Several studies report that substantial savings have been realized by companies that coordinated their operations (e.g., King and Love (1980), Blumenfeld et al. (1987), Martin et al. (1993)). Besides, several researchers constructed mathematical models on which they have empirically shown the savings that can be obtained by a coordinated approach over a decoupled (sequential) approach (e.g., Chandra (1993), Chandra and Fisher (1994), Fumero and Vercellis (1999)).

In order to implement a VMI system, one has to establish an integrated optimization model in which the related coordination decisions are considered simultaneously. In that respect, IRP is at the core of VMI systems and should be addressed effectively. In this study, we consider IRP with deterministic order-up-to level inventory policy employed at the retailers, i.e., IRO. A variant of IRO is first introduced by Bertazzi et al. (2002). Employing deterministic order-up-to level policy at the retailers is a reasonable strategy in several industries. Consider, for instance, the replenishment of vending machines or the replenishment of tanks of industrial gas companies, where vending machines or tanks are filled up to their capacity whenever replenished. On the other hand, assuming deterministic demand may not be correct since in real-life demand is mostly stochastic. However, minimum inventory level that must be maintained at each retailer due to the deterministic order-up-to level policy provides a hedging against the variability of demand. In addition to this, accurate demand forecasts justify usage of such a policy.

Another aspect of our study is the supplier's order policy. Instead of assuming a limitless inventory available at the supplier or a given amount made available in each period as in Bertazzi et al. (2002), we consider a more realistic approach, in which the supplier determines when to order and how much to order.

Our main motivation in this study is to develop a mathematical programming based approach to the IRO. Bertazzi et al. (2002) develop a heuristic solution procedure for a variant of the IRO, but no lower bounding procedure has been proposed. We aim at constructing a general model that can handle several cost structures and proposing efficient upper and lower bounding procedures for the general model.

1.2 Problem definition

The IRO entails joint consideration of inventory management of both supplier and the retailers and distribution planning among them. In this study, we consider the following specific IRO.

We are given a logistics system, in which a single supplier delivers a single product to multiple retailers over a finite time horizon. Retailers facing deterministic dynamic demand are serviced via a homogeneous fleet of vehicles having limited capacity and located at the supplier. Each retailer has specific minimum and maximum levels of inventory, and it has to be replenished before its inventory level drops below the minimum level and in each replenishment its inventory is brought up to the maximum level. This is called deterministic order-up-to level inventory policy in Bertazzi et al. (2002). In addition, the supplier has to order from a higher echelon (or manufacture) in order to be able to service retailers.

The problem is to simultaneously determine (i) retailers to be visited, (ii) the quantity of product to be delivered to retailers, (iii) routes of vehicles, (iv) the quantity of product to order to the supplier, in each time period so that the sum of fixed order cost and inventory holding cost at the supplier, and fixed vehicle dispatching cost, vehicle routing cost and inventory holding costs at the retailers are minimized.

Note that even if the decisions concerning (i), (ii) and (iv) are given a priori, determining only routes of vehicles is still a difficult problem. Thus, the problem we consider involves integrated complex decisions and is extremely difficult to solve.

1.3 Outline of the study

The following are the brief outline of the thesis.

In chapter 2, we present a brief review of the literature concerning the integration of different functions such as production planning, inventory management and distribution planning into a single optimization model. We confine our review on studies explicitly considering vehicle routing issues and also those with deterministic settings. The review is given with respect to the planning horizon used and decisions to be taken at different echelons.

In chapter 3, we make the connection between our problem and closely related studies by making a comparison and contrast among them, followed by the assumptions made and then the mathematical formulation. The mathematical formulation presented is adapted from the literature for our problem. It is a general model which can accommodate several cost structures with only slight changes. In this chapter, we also present a Lagrangian relaxation (LR) based solution approach proposed for the model and explain it in detail.

Chapter 4 addresses a special case of our problem, which is a single supplier-single retailer problem. In this chapter, we propose an efficient method for solving the special case of the problem. Moreover, we also propose an efficient method for the single retailer problem (ignoring the supplier), rapid solution of which is important since our LR based solution approach involves many single retailer subproblems that must be solved repeatedly.

Chapter 5 is devoted to the computational study on test instances. In this chapter, we test several LR based solution procedures on small instances to determine parameters of these procedures for the best implementation. We implement some of these solution procedures under their best settings on large instances and discuss the results.

In chapter 6, we conclude the study, briefly mention about our contributions and give future research directions.

CHAPTER 2

LITERATURE REVIEW

Over the past years, several researchers have investigated integration of different functions (e.g. inventory management, production planning, distribution planning, etc.) either within the echelons of the same company or among different companies on a single optimization model. A review of such models can be found in Thomas and Griffin (1996), Sarmiento and Nagi (1999) and Baita et al. (1998). Unlike Thomas and Griffin (1996), Sarmiento and Nagi (1999) and Baita et al. (1998) explicitly consider transportation system but the latter with routing aspects. Both of the first two papers review the work done at the strategical level and tactical level. In this study, however, we consider the research on the integrated tactical and operational level decisions such as production and distribution planning, inventory and distribution planning, and production, inventory and distribution planning, assuming that the characteristics of the underlying distribution system are provided. Moreover, we also consider research that explicitly addresses routing issues and deterministic demand at retailers (or customers). In the literature, there exist studies with stochastic demand case, for a brief review on which, one can refer to Baita et al. (1998) and Kleywegt et al. (2004).

We next present a review of the research done with respect to planning horizon (finite or infinite horizon) and decisions to be taken at different echelons (production, inventory and distribution). We consider warehouse / depot / supplier / plant as the first echelon and retailer / customer as the second echelon.

2.1 Distribution-Inventory

This part addresses distribution systems, in which a depot (warehouse) replenishes several customers (retailers) by allocating product(s) available at the depot. The decisons of interest are delivery shipment sizes to each customer and routes of vehicles.

One important point is that the distribution-inventory problem is often confused by reseachers with the so called inventory/routing problem. According to Sarmiento and Nagi (1999), an important difference exist between two problems in that while inventory-routing problem looks for optimization of depot's operation but not those of customers, the distribution-inventory problem considers optimization of all the echelons and shares the benefit among parties involved. On the other hand, several studies such as Bertazzi et al. (2002), Kleywegt et al. (2002) and Federgruen et al. (1995) do not distinguish such a difference and consider work classified as distribution-inventory problem (e.g., Federgruen and Zipkin (1994), Anily and Federgruen (1990), etc.) in Sarmiento and Nagi (1999) as inventory/routing problems. Obviously, in the literature, there are no consensus among researchers regarding how to make the classification. Therefore, we name the problems with regard to decisions to be taken at the echelons and thus, we use the terms "distribution-inventory" and "inventory/routing" interchangeable in this section.

2.1.1 Finite horizon models

Federgruen and Zipkin (1984) appears to be the first to consider the integrated inventory allocation and vehicle routing problem, in which a limited amount of single product available at a central depot is delivered to several customers facing stochastic demand by a set of vehicles with limited capacity. Inventory allocation among customers (all customers may not receive delivery) and vehicle schedules are determined simultaneously so as to minimize expected inventory holding and shortage costs at the customers as well as vehicle routing costs proportional to the distance travelled. A single period, “myopic”, mathematical programming model (nonlinear mixed integer) is constructed for the integrated problem, for the solution of which both a heuristic method (a modified interchange heuristic) and an exact method (generalized Benders’ decomposition) are proposed. Computational results, which are given only for the heuristic method, suggest that the method is fast enough for practical use. Moreover, a comparison of integrated versus sequential approach is made and it is shown that significant cost savings (6-7%) can be achieved with the integrated approach over the sequential approach. Note that this work considers stochastic demand at customers, but we include it into the review. This is because, it is one of the seminal studies in the area of integrated inventory management and vehicle routing.

Bell et al. (1983) developed a decision support system for Air Products and Chemicals, Inc., which significantly improves decisions regarding integrated inventory management of industrial gases at customers and vehicle scheduling. The problem handled by researchers was a quite complex problem. Inventory of products (two products; liquid oxygen and liquid nitrogen) must be kept above a specied safety-stock level for each customer. Also, customers have delivey time windows, outside of which delivery is not allowed. Moreover, company has a heterogenous fleet in terms of capacity and operating costs (limited size). Lastly, every truck is not eligible for serving every customer. Costs considered include driver pay (may be based on time spent on the trip or distance traveled), tolls and vehicle related costs including fuel, maintenance and depreciation. Besides costs directly related to distance traveled, there are costs that depend on the time spent by drivers for loading, unloading and performing several setup functions. The decisions of interest are to select best routes among a set of possible routes generated in advance and to allocate limited amount of products to customers such that profit (value of products delivered less costs incurred) is maximized. A large-scale mixed integer programming formulation is constructed for the problem and solved using a Lagrangian relaxation algorithm close to optimality.

Chien et al. (1989) address the integrated problem of inventory allocation and vehicle routing problem, where a limited amount of product available at the central depot is allocated among customers and distributed via a fleet of vehicles with limited capacity in order to maximize profit (revenues less delivery costs). Different than the work of Federgruen and Zipkin (1984), each customer experiences deterministic demand, which equals to the customer specific storage capacity less beginning inventory level (maximum demand of a customer). However, amount of inventory available at the depot is not sufficient to meet all customers' maximum demand. A customer specific unit revenue is gained for each unit delivered to the customer, whereas for each unit of unmet demand a customer specific unit penalty cost is charged. A single period, multi-commodity flow based mixed integer programming model is formulated for the problem, which has a maximization objective composed of total revenues less total cost consisting of penalty costs, fixed and variable routing costs. A Lagrangian relaxation approach is developed, which generates good quality solutions with small gaps between lower and upper bounds for several problem instances with a variety of parameter settings.

Bertazzi et al. (2002) combined a deterministic order-up-to level inventory policy with inventory routing problem, in which multiple products are shipped from a supplier to retailers via a vehicle with limited capacity over a finite time horizon. Shipments are to be performed before each retailer's inventory reaches its specific minimum level of inventory of each product. Every time a retailer is visited, level of each product delivered is brought up to its specific maximum level. The quantity of each product shipped to the supplier in each period is known. Also, each retailer experiences deterministic dynamic demand. The objective is to determine amount of each product to be delivered to each retailer and vehicle route in each period such that total of inventory costs both at the supplier and at the retailers as well as distance based vehicle routing costs is minimized. One aim of the paper is to solve the above defined problem. Another aim is to study the effect of various objective functions (cost structures) corresponding to different decision policies. They present a heuristic to solve the original problem and three variants obtained by considering various objective functions. To compare the quality of solutions obtained using the proposed heuristic, two intuitive policies are proposed. According to the computational results, the proposed heuristic outperforms both of the two intuitive policies in all of the randomly generated problem instances.

2.1.2 Infinite horizon models

Blumenfeld et al. (1985) and Burns et al. (1985) are the first to consider explicitly the integrated problem of inventory and vehicle routing in an infinite horizon model. Their model, however, exploits spatial density of customers rather than their exact locations. In addition, they do not consider vehicle routes as decisions variables.

Anily and Federgruen (1990) address distribution systems consisting of a depot and several geographically scattered retailers, each facing constant (retailer-specific) deterministic demand rate over infinite time horizon. Unlike Blumenfeld et al. (1995) and Burns et al.(1985), Anily and Federgruen use exact locations of retailers. The depot acts as a transshipment point, i.e. no inventory is held at the depot. Retailers can keep inventory and a variable inventory carrying cost, which is identical for all retailers, is incurred at a constant rate per unit time per unit stored. The objective is to determine routing patterns as well as retailer replenishment strategies such that long-run average vehicle routing costs and inventory carrying costs at the retailers are minimized. Routing-related costs include fixed costs per route driven and variable costs proportional to distance traveled. A class of replenishment strategies is considered in which retailers are partitioned into regions and each time a retailer in a given region is served by a vehicle, that vehicle visits all the retailers in the given region as well. A retailer may belong to several regions. They propose a heuristic procedure for the problem defined, which is asymptotically optimal within the given the class of replenishment strategies considered.

Anily (1994) extends the work of Anily and Federgruen (1990) by allowing nonidentical inventory holding cost rates and provides a heuristic procedure, which is again asymptotically optimal within the class of strategies considered.

Gallego and Simchi-Levi (1990) consider the same distribution system as in Anily and Federgruen (1990), but they analyze direct shipping strategy rather than a routing strategy. They show that direct shipping is at least 94% effective when the minimal economic lot size over all retailers is at least 71% of the vehicle capacity. Effectiveness is the ratio of the infimum of long-run average cost over all strategies to long-run average cost of strategy considered. The effectiveness deteriorates as the economic lot size (percentage of vehicle capacity) decreases.

Viswanathan and Mathur (1997) extend the work of Anily and Federgruen (1990) by generalizing it to multiple products. Authors develop a heuristic which generates a stationary nested joint replenishment policy (SNJRP) for the problem. Although the heuristic is able to handle problems with multiple products, computational results are derived for the case of single product so as to compare results with that of Anily and Federgruen (1990). Results show that SNJRP policy outperforms the replenishment strategy of Anily and Federgruen (1990) in terms of cost in most of the cases.

Chan et al. (1998) address the same model defined by Anily and Federgruen (1990), but consider different class of strategies for the solution of inventory/routing problem. They consider Fixed Partition Policies, in which retailers are partitioned into a set of regions such that each region is served separately (by a vehicle). A retailer cannot belong to multiple regions and each time a retailer in a region is visited by a vehicle, all the retailers in that region are also visited by the same vehicle. Moreover, authors consider Zero Inventory Ordering Policies, in which retailers are replenished only when their inventory levels become zero. An algorithm based on formulating inventory/routing problem as a Capacitated Concentrator Location Problem (CCLP) is presented. Application of algorithm provides solutions that are asymptotically optimal within the fixed partition policies and computational results suggest that the algorithm is very effective on several randomly generated problem instances.

2.2 Inventory-Distribution-Inventory

2.2.1 Finite horizon models

Chandra (1993) addresses the problem of integrating inventory management both at a single warehouse and at customers as well as vehicle routing over finite time horizon. Multiple products are delivered from the warehouse to customers facing deterministic dynamic demand via homogeneous vehicles (assuming enough vehicles) located at the warehouse. The decisions to be taken include replenishment quantities for each product in each period at the warehouse, quantities of each product delivered to each customer in each period and vehicle routes for the delivery of products. Costs related with inventory are fixed ordering costs and variable inventory holding costs both at the warehouse and at customers. Transportation related costs include fixed vehicle dispatching costs as well as variable routing costs proportional to distance traveled. A heuristic solution scheme is proposed by the author, which is used to assess the impact of coordinating inventory management and vehicle routing on total cost. Heuristic first solves warehouse ordering problem and then the resulting distribution routing problem sequentially. Subsequently, solution scheme attempts to change distribution routing problem solution by consolidating deliveries of different periods (i.e. delivering some items earlier) such that total of inventory holding and vehicle routing costs decreases. Computational results show that as the number of products and customers as well as the length of planning horizon increases, benefits of coordination in terms of reduced costs increases as well.

2.2.2 Infinite horizon models

Anily and Federgruen (1993) extend their previous work (Anily and Federgruen, 1990) to allow for the depot to hold inventory. The problem considered is to jointly determine replenishment of the depot as well as the replenishment of several retailers along with the delivery routes. Authors provide a lower bound on the total cost as well as a heuristic procedure to give good upper bounds. They show that their heuristic procedure generates solutions almost within 6% of optimality within the class of strategies considered.

2.3 Production-Inventory-Distribution-Inventory

2.3.1 Finite horizon models

Chandra and Fisher (1994) consider a problem similar to that of Chandra (1993). However, instead of ordering from a higher echelon to the warehouse, here production planning decisions are considered at a single plant. Multiple products are produced in the plant over time and distributed with a fleet of vehicles (unlimited size) to various geographically dispersed retailers with deterministic dynamic demand for each product. The aim is to simultaneously determine production quantities of each product in each period, delivery shipment sizes of each product in each period to each retailer and a set of routes such that total of fixed production setup costs, inventory holding costs both at the plant and the retailers, fixed vehicle dispatching cost as well as routing costs proportional to distance traveled is minimized. An integrated optimization model, a mixed integer programming model, is constructed and a heuristic procedure is proposed for the solution of the integrated optimization model. First, a capacitated lot sizing problem at the plant is solved, after which the distribution routing model is solved. Once these problems are solved separately, the heuristic procedure tries to consolidate deliveries to given customer in separate periods. Consolidation is performed if it reduces cost. A sequential approach, in which capacitated lot sizing and distribution routing problems are solved sequentially, is also proposed to evaluate the effect of coordination on total cost. Computational results show that cost reductions due to coordination increase as the number of products and retailers as well as the length of planning horizon increases, as the production and vehicle capacity get larger and as the distribution costs increase relative to production costs.

Fumero and Vercellis (1999) address the same problem considered by Chandra and Fisher (1994), but they formulated the model as a multi-commodity flow model different than Chandra and Fisher (1994). They proposed a Lagrangian relaxation based solution procedure, which decomposes production and distribution decisions, but still maintains a global optimization perspective via Lagrangian dual problem. There are some other differences between work of Fumero and Vercellis (1999) and Chandra and Fisher (1994). A limited fleet size is considered in the former, while the latter assumes unlimited fleet size. Also, instead of variable routing cost proportional only to distance traveled, routing cost proportional to both unit carried and distance traveled is considered in the former. In addition, solution obtained from an alternative decoupled approach, in which production lot sizing is solved first and the resulting distribution routing problem is then solved, is compared with the feasible solution obtained using Lagrangian relaxation based solution procedure (integrated approach). Computational results show that significant savings can be realized by integrated approach over decoupled approach.

CHAPTER 3

MATHEMATICAL FORMULATION AND THE SOLUTION APPROACH

In this chapter, we first make a comparison and contrast of IRO with respect to the closely related literature and then provide the assumptions made. Subsequently, we present the mathematical formulation of IRO, followed by a Lagrangian relaxation based solution approach, in which we describe how to compute feasible solution (upper bound) and lower bound to the problem.

The IRO integrates inventory management of both supplier and retailers and distribution planning in a single model, in which these decisions are simultaneously made so as to minimize the system-wide costs consisting of inventory holding costs both at the supplier and retailers, distance based transportation cost as well as fixed order cost of supplier and fixed vehicle dispatching costs of serving retailers.

With respect to the cost structure, our study is the same as that of Chandra (1993) and Chandra and Fisher (1994). On the other hand, cost structure of our problem is slightly different than Fumero and Vercellis (1999) and Bertazzi et al. (2002). The difference between the former and our study is that the former studies situations in which transportation costs are proportional to amount shipped while our setting is based on purely distance based transportation costs. On the other hand, the latter involves no fixed order or vehicle dispatching cost unlike ours.

Considering decisions made in IRO, it is obvious that our problem is an Inventory-Distribution-Inventory problem according to our classification scheme, as discussed in Chapter 2. In this respect, our work is very similar to that of Chandra (1993). However, it substantially differs from Chandra (1993) in that in our work, a deterministic order-up-to-level inventory policy is employed at retailers, which is the case in Bertazzi et al. (2002) as well. Thus, we can state that our study is an extension of the work of Bertazzi et al. (2002), since we extend their original problem to include order policy of the supplier.

Before presenting the mathematical model of the IRO, we state the following assumptions that we made for our formulation.

- In a given period, the amount of product shipped to a retailer which brings its inventory level up to maximum level can be transported by more than one vehicle (partial servicing is allowed).
- Each vehicle can perform at most one trip in a period.
- Aggregate capacity of vehicle fleet is sufficient to replenish retailers if all of them require replenishment in the same period.
- There is no limit on the amount of product that can be ordered to the supplier.
- There is no lead time for deliveries to both supplier[*] and retailers.
- Stockout at the supplier and an inventory level lower than minimum inventory level at the retailers are not allowed.

3.1 Mathematical model of the IRO

In this section, we present the mathematical model of the IRO and describe its properties.

Indeed, one can propose several different formulations with different structures for the IRO. In particular, for this problem, it is possible to develop either a standard subtour elimination model or a multi-commodity flow model for routing decisions. A model with standard subtour elimination constraints (Miller, Tucker and Zemlin constraints) is firstly developed by Pınar and Süral (2004) for the deterministic order-up-to level inventory routing problem as defined by Bertazzi et al. (2002). We, on the other hand, resort to a multi-commodity flow model, which lends itself

[*] In Bertazzi et al. (2002), there is a one period lead time for deliveries to the supplier.

to several decompositions and relaxations. Our model is mainly adopted from Fumero and Vercellis (1999), but with slight differences. First, they consider production policy of the supplier (or plant), while we have order policy at the supplier. Thus, we excluded production capacity constraints from the model. Besides, we have incorporated constraints ensuring deterministic order-up-to level policy at retailers into the model. Last but not least, distance based transportation cost is concerned in our model whereas transportation cost proportional to amount shipped in theirs.

The following are the indices, parameters and decision variables along with the mathematical model of the IRO. Note that last element of indices defined also denotes the set.

Indices:

t : Time periods, *1*, ..., *T*

i, j : Multiple retailers, *1*, ..., *N*, (Note that $i = 0$ denotes supplier) where N denotes the set of retailers, while $\overline{N} = N \cup \{0\}$

v : Multiple vehicles, *1*,..., *V*

Parameters:

d_{it} : Demand at retailer i in period t.

K : Capacity of a vehicle.

s_i : Minimum level of inventory at retailer i at the end of period t.

S'_i : Maximum level of inventory at retailer i at the end of period t.

c_{ij} : Cost of transportation from retailer i to j.

h_{it} : Holding cost at retailer i incurred for each unit kept at the end of period t.

f_t : Fixed vehicle dispatching cost in period t.

f_{0t} : Fixed order cost incurred at the supplier in period t.

M_0, M_i : Big numbers.

I_{00}, I'_{i0} : Beginning inventory levels at the supplier and retailers, respectively.

Variables:

I_{0t} : Inventory level of supplier at the end of period *t*.

I'_{it} : Inventory level of retailer *i* at the end of period *t*.

P_t : Amount of product ordered to the supplier from a higher echelon in period *t*.

X_{ijvt} : Amount of product transported from retailer *i* to *j* by vehicle *v* in period *t*.

Q_{ivt} : Amount of product delivered to retailer *i* by vehicle *v* in period *t*.

W_{ijvt} : 1 if retailer *j* is visited immediately after retailer *i* by vehicle *v* in period *t*. 0 otherwise.

z_{it} : 1 if a delivery is made to retailer *i* in period *t*, which brings inventory level to the maximum level. 0 otherwise.

y_t : 1 if an order for the supplier is given in period *t*. 0 otherwise.

IRO:

$$\text{Min } \sum_{t=1}^{T} f_{0t} y_t + \sum_{t=1}^{T} h_{0t} I_{0t} + \sum_{i\in N}\sum_{t=1}^{T} h_{it} I'_{it} + \sum_{j\in N}\sum_{v=1}^{V}\sum_{t=1}^{T} f_t W_{0jvt}$$

$$+ \sum_{\substack{i\in \overline{N} \\ i\neq j}} \sum_{j\in \overline{N}} \sum_{v=1}^{V}\sum_{t=1}^{T} c_{ij} W_{ijvt} \qquad (1)$$

Subject to

$$I_{0t} = I_{0,t-1} + P_t - \sum_{i\in N}\sum_{v=1}^{V} Q_{ivt} \qquad \forall t \qquad (2)$$

$$I'_{it} = I'_{i,t-1} + \sum_{v=1}^{V} Q_{ivt} - d_{it} \qquad \forall i \in N, \forall t \qquad (3)$$

$$P_t \leq M_0 y_t \qquad \forall t \qquad (4)$$

$$s_i \leq I'_{it} \qquad \forall i \in N, \forall t \qquad (5)$$

$$\sum_{\substack{j \in \overline{N} \\ j \neq i}} X_{jivt} - \sum_{\substack{j \in \overline{N} \\ i \neq j}} X_{ijvt} = Q_{ivt} \qquad \forall i \in N, \forall v, t \qquad (6)$$

$$\sum_{j \in N} X_{0jvt} - \sum_{j \in N} X_{j0vt} = \sum_{i \in N} Q_{ivt} \qquad \forall v, t \qquad (7)$$

$$\sum_{j \in N} W_{0jvt} \leq 1 \qquad \forall v, t \qquad (8)$$

$$\sum_{\substack{j \in \overline{N} \\ i \neq j}} W_{ijvt} = \sum_{\substack{j \in \overline{N} \\ j \neq i}} W_{jivt} \qquad \forall i \in \overline{N}, \forall v, t \qquad (9)$$

$$X_{ijvt} \leq K W_{ijvt} \qquad \forall i, j \in \overline{N}, i \neq j, \forall v, t \qquad (10)$$

$$\sum_{v=1}^{V} Q_{ivt} + I'_{i,t-1} \leq S'_i \qquad \forall i \in N, \forall t \qquad (11)$$

$$\sum_{v=1}^{V} Q_{ivt} \leq M_i z_{it} \qquad \forall i \in N, \forall t \qquad (12)$$

$$\left(S'_i - I'_{i.t-1}\right) - \sum_{v=1}^{V} Q_{ivt} \leq M_i \left(1 - z_{it}\right) \qquad \forall i \in N, \forall t \qquad (13)$$

$$I_{0t} \geq 0, \forall t;\ Q_{ivt} \geq 0, \forall i \in N, \forall v, t;\ X_{ijvt} \geq 0, \forall i, j \in \overline{N}, \forall v, t;\ P_t \geq 0, \forall t;$$

$$W_{ijvt} \in \{0,1\}, \forall i, j \in \overline{N}, \forall v, t;\ y_t \in \{0,1\}, \forall t;\ z_{it} \in \{0,1\}, \forall i \in N, \forall t \qquad (14)$$

The objective function (1) of the model consists of fixed order and inventory holding costs at the supplier, and inventory holding, fixed vehicle dispatching and transportation costs at the retailer. Constraints (2) and (3) are the inventory balance equations for the supplier and retailers, respectively. Constraint (4) ensures that a fixed order cost is incurred if supplier places an order in a period. Constraint (5) guarantees that inventory level of retailer never falls below the minimum level. Constraints (6) and (7) are the flow conservation equations at the retailers and supplier, respectively. Constraint (8) states that each vehicle can perform at most one trip in a period. Constraint (9) and (10) together with (6) and (7) assure the integrity and feasibility of tours. Constraint (11) stipulates that the total amount delivered to retailer plus inventory carried from previous period cannot exceed the maximum level. Constraint (13) together with (11) and (12) are the order-up-to level constraints. They ensure that the inventory level at the retailer is brought up to the maximum level if a delivery is made and no amount is shipped unless a delivery is made. Constraints (14) are for nonnegativity and integrality of variables.

The model provided above can be simplified as follows.

Let $M_i = S_i' - s_i$, $S_i = S_i' - s_i$, $I_{it}' - s_i = I_{it} \geq 0$

Then constraint (5) drops and objective function (1) as well as constraints (3), (11), (13) and (14) transform into equations in terms of I variables.

Modified IRO:

$$\text{Min } \sum_{t=1}^{T} f_{0t} y_t + \sum_{t=1}^{T} h_{0t} I_{0t} + \sum_{i \in N} \sum_{t=1}^{T} h_{it} (I_{it} + s_i) + \sum_{j \in N} \sum_{v=1}^{V} \sum_{t=1}^{T} f_t W_{0jvt} + \sum_{\substack{i \in \bar{N} \\ i \neq j}} \sum_{j \in \bar{N}} \sum_{v=1}^{V} \sum_{t=1}^{T} c_{ij} W_{ijvt} \qquad (1')$$

Subject to

$$I_{0t} = I_{0,t-1} + P_t - \sum_{i \in N} \sum_{v=1}^{V} Q_{ivt} \qquad \forall t \qquad (2)$$

$$I_{it} = I_{i,t-1} + \sum_{v=1}^{V} Q_{ivt} - d_{it} \qquad \forall i \in N, \forall t \qquad (3')$$

$$P_t \leq M_0 y_t \qquad \forall t \qquad (4)$$

$$\sum_{\substack{j \in \bar{N} \\ j \neq i}} X_{jivt} - \sum_{\substack{j \in \bar{N} \\ i \neq j}} X_{ijvt} = Q_{ivt} \qquad \forall i \in N, \forall v, t \qquad (6)$$

$$\sum_{j \in N} X_{0jvt} = \sum_{i \in N} Q_{ivt} \qquad \forall v, t \qquad (7a)$$

$$\sum_{j \in N} X_{jovt} = 0 \qquad \forall v, t \qquad (7b)$$

$$\sum_{j \in N} W_{0jvt} \leq 1 \qquad \forall v, t \qquad (8)$$

$$\sum_{\substack{j\in\overline{N}\\ i\neq j}} W_{ijvt} = \sum_{\substack{j\in\overline{N}\\ j\neq i}} W_{jivt} \qquad \forall i\in\overline{N}, \forall v,t \qquad (9)$$

$$X_{ijvt} \leq KW_{ijvt} \qquad \forall i\in\overline{N}, j\in N, i\neq j, \forall v,t \qquad (10)$$

$$\sum_{v=1}^{V} Q_{ivt} + I_{i,t-1} \leq S_i \qquad \forall i\in N, \forall t \qquad (11')$$

$$\sum_{v=1}^{V} Q_{ivt} \leq S_i z_{it} \qquad \forall i\in N, \forall t \qquad (12')$$

$$(S_i - I_{i,t-1}) - \sum_{v=1}^{V} Q_{ivt} \leq S_i(1-z_{it}) \qquad \forall i\in N, \forall t \qquad (13')$$

$$I_{it}\geq 0, \forall i\in\overline{N}, \forall t;\ Q_{ivt}\geq 0, \forall i\in N, \forall v,t;\ X_{ijvt}\geq 0, \forall i,j\in\overline{N}, \forall v,t;\ P_t\geq 0, \forall t$$

$$W_{ijvt}\in\{0,1\}, \forall i,j\in\overline{N}, \forall v,t;\ y_t\in\{0,1\}, \forall t;\ z_{it}\in\{0,1\}, \forall i\in N, \forall t \qquad (14')$$

Note that we have also deduced the minimum inventory level of each retailer (s_i) from its beginning inventory level, i.e., $I'_{i0} - s_i = I_{i0}$. As of now, we do not allow stockout (falling below zero) at the retailers instead of falling below minimum inventory level (s_i), since we have already allocated the minimum inventory level of each retailer. Also, we separated constraint (7) into two pieces: constraints (7a) and (7b), since we know that total amount of product sent from the supplier should be equal to the total amount to be left to the retailers (to be visited) and no amount of product should be transported back to the supplier (7b). In the rest of the paper, when we refer to IRO we mean the above simplified model.

As we mentioned in chapter 1, one of our goals is to develop a general model that can accommodate several cost structures. The model has this flexibility in that for example, we can easily modify our objective function to model a problem with transportation cost proportional to the amount shipped like in Fumero and Vercellis (1999) by inserting a cost term composed of X variables instead of W's into the objective function without any need for changing constraints or defining new variables. In addition, if we removed fixed order and fixed vehicle dispatching costs from the objective function, let the P_t be a parameter and added a constraint accounting for the one period lead time for deliveries to the supplier, we would come up with a model for the original problem of Bertazzi et al. (2002).

It is obvious that IRO is a large-scale mixed integer programming model (MIP) involving $2N^2VT + 3NVT + 2NT + 3T$ many variables $N^2VT + NVT + NT + T$ of which are binary variables and the rest being continuous variables. Besides huge number of variables, the formulation has $N^2VT + 2NVT + 4NT + 4VT + 2T$ many constraints. In order to see how large the formulation is, for instance consider a problem with 50 retailers, time horizon of 30 periods and a single vehicle, which is the largest problem size we will experiment with. Such a problem has over 150,000 variables made up of about 75,000 binary as well as continuous variables, in addition to approximately 84,000 constraints. As a result, it is very difficult to solve such a large scale MIP model to optimality.

3.2 Lagrangian relaxation based solution approach

As emphasized in the preceding section, we have a very large scale optimization model at hand to solve. Because solving such a large model to optimality by using even powerful commercial solvers such as CPLEX is hopeless, we decided to resort to a Lagrangian relaxation based approach, which provides not only good feasible solutions but also lower bounds on the problem (for a minimization problem).

In this section, we first present our Lagrangian relaxation approach. In particular, we show how to compute lower bounds and good feasible solutions (upper bound) for the problem. Finally we describe our multiplier updating method which is a standard subgradient optimization algorithm.

Lagrangian relaxation, a powerful technique for solving integer programming problems, is to relax hard constraints of an integer program and to insert them into the objective function with associated multipliers punishing violations regarding those constraints (Fisher, 1985). The method provides not only upper bounds but also lower bounds for the problem. It iteratively updates multipliers so that the gap between upper bound and lower bound vanishes and hopefully a good feasible solution (perhaps optimal solution) as well as a tight lower bound is obtained. There are many successful applications of Lagrangian relaxation to several difficult problems in the literature (see e.g., Fisher 1981, 1985).

There are three important questions that must be addressed for a successful application of Lagrangian relaxation (Fisher, 1985): (i) which constraint(s) to relax, (ii) how to find a good feasible solution (iii) how to obtain good multipliers. We shall answer all these questions in the sequel.

Lagrangian relaxation used in this study aims to relax a few constraints such that the replenishment problems and the distribution planning problem are separated from each other. We thus disaggregate the problem of multiple retailers into many single retailer problems by relaxation. In order to achieve these goals, we relax constraints (2), (6) and (7a) in the model. Multipliers m_t, a_{ivt} and b_{vt} are defined for constraints (2), (6) and (7a) respectively and incorporated into the objective function. The resulting relaxed problem called PR is as follows.

PR:

$$\text{Min (1')} + \sum_{t=1}^{T} m_t \left(I_{0t} - I_{0,t-1} - P_t + \sum_{i\in N} \sum_{v=1}^{V} Q_{ivt} \right)$$

$$+ \sum_{i\in N} \sum_{v=1}^{V} \sum_{t=1}^{T} a_{ivt} \left(- \sum_{\substack{j\in \overline{N} \\ j\neq i}} X_{jivt} + \sum_{\substack{j\in \overline{N} \\ i\neq j}} X_{ijvt} + Q_{ivt} \right) + \sum_{v=1}^{V} \sum_{t=1}^{T} b_{vt} \left(- \sum_{j\in N} X_{0jvt} + \sum_{i\in N} Q_{ivt} \right)$$

Subject to (3'), (4), (7b), (8), (9), (10), (11'), (12'), (13'), and (14').

Arranging the terms of the above objective function with respect to variables gives the following model:

PR:

Min

$$\sum_{t=1}^{T} f_{0t} y_t + \sum_{t=1}^{T} (h_{0t} + m_t) I_{0t} - m_1 I_{00} - \sum_{t=2}^{T} m_t I_{0,t-1} - \sum_{t=1}^{T} m_t P_t + \sum_{i \in N} \sum_{t=1}^{T} h_{it} I_{it} + \sum_{i \in N} \sum_{t=1}^{T} h_{it} s_i$$

$$+ \sum_{i \in N} \sum_{v=1}^{V} \sum_{t=1}^{T} (m_t + a_{ivt} + b_{vt}) Q_{ivt} + \sum_{j \in N} \sum_{v=1}^{V} \sum_{t=1}^{T} f_t W_{0jvt} + \sum_{\substack{i \in N \\ i \neq j}} \sum_{j \in N} \sum_{v=1}^{V} \sum_{t=1}^{T} c_{ij} W_{ijvt}$$

$$+ \sum_{\substack{j \in N \\ j \neq i}} \sum_{i \in N} \sum_{v=1}^{V} \sum_{t=1}^{T} a_{ivt} (-X_{jivt} + X_{ijvt}) \quad \sum_{j \in N} \sum_{v=1}^{V} \sum_{t=1}^{T} b_{vt} (-X_{0jvt})$$

Subject to (3’), (4), (7b), (8), (9), (10), (11’), (12’), (13’), and (14’). Note that the third and seventh terms are constant.

The problem *PR* decomposes into four subproblems, which are as follows:

- Supplier’s Order Problem
- Supplier’s Inventory Problem
- Distribution Problem
- Retailers’ Replenishment Problem

and the corresponding Lagrangian dual problem, referred to as *LDP*, is $\underset{m,a,b}{\text{Max}}\, PR$.

The solution of those four subproblems yields a lower bound to the problem, denoted by LB. Also, using the information obtained from solving these subproblems, we can easily construct feasible solutions (upper bounds) for the problem. The solution of the Lagrangian dual problem is obtained using standard subgradient optimization algorithm. These are all discussed in the following sections.

3.2.1 Computation of lower bound

Supplier's Order Problem (ORDER)

It involves lot-sizing and setup variables of the supplier. The subproblem is as follows:

ORDER: Min $\sum_{t=1}^{T}(f_{0t}y_t - m_t P_t)$

s.t.

$$P_t \le M_0 y_t \qquad \forall t \qquad (4)$$

$$y_t \in \{0,1\}; P_t \ge 0 \qquad \forall t$$

The *ORDER* problem decomposes with respect to time periods, each of which can be easily solved using simple decision rules. If for any period *t*, $f_{0t}y_t - m_t P_t$ is less than zero, then we let $P_t = M_0$ and $y_t = 1$, otherwise we let both P_t and y_t be zero. However, to make *ORDER* tighter, we propose two valid inequalities given below.

§ $$\sum_{t=1}^{T} P_t \le \sum_{i\in N}(S_i - I_{i0} + \sum_{r=1}^{T-1} d_{ir}) - I_{00} \qquad (15)$$

§ $$\sum_{r=1}^{t} P_r \ge \sum_{i\in N}\left(\sum_{r=1}^{t} d_{it} - I_{i0} \,\middle|\, \sum_{r=1}^{t} d_{it} > I_{i0}\right) - I_{00} \qquad \forall t \qquad (16)$$

The first inequality ensures that total amount of product that can be ordered to the supplier cannot exceed total of maximum requirements of customers over the horizon less the beginning inventory level at the supplier. It is easy to show that this inequality is valid. Note that $S_i - I_{i0} + \sum_{r=1}^{T-1} d_{ir}$ is the maximum amount of product that can be delivered to retailer *i* over the horizon. $S_i - I_{i0}$ is the replenishment amount to retailer *i* in the first period while $d_{i,t-1}$ is replenishment amount to retailer

i in period t ($2 \le t \le T$) if a delivery occurs to retailer i in that period. Thus, right hand side of the first inequality is obtained assuming that the replenishment of all the retailers occurs in every period.

Second inequality stipulates that total amount of product to be ordered to supplier up to period *t* should be at least the total of minimal requirements of retailers less initial amount at the supplier. Here, minimal requirements of retailers up to period *t* is the amount demanded from retailer *i* up to period *t* less initial inventory available at retailer *i*. If the total of minimal requirements over all retailers up to period *t* exceeds the initial inventory level of the supplier, then an amount at least equal to this difference must be ordered to the supplier up to period *t*.

Unfortunately, these inequalities make problem tighter but at the expense of disrupting the decomposition feature with respect to time periods and converting it into a mixed integer programming problem. Nevertheless, we are able to solve the new *ORDER* problem with commercial solver CPLEX to optimality in a very short time.

Supplier's Inventory Problem (SINV)

It consists of only inventory level variables of supplier and there are no constraints other than nonnegativity constraints. In other words, this is an unconstrained linear minimization problem, solution of which is trivial. The subproblem is as follows:

SINV: Min $\sum_{t=1}^{T-1} (h_{0t} + m_t - m_{t+1}) I_{0t} + (h_{0T} + m_T) I_{0T} - m_1 I_{00}$

s.t.

$I_{0t} \ge 0 \qquad \forall t$

This problem further decomposes with respect to time. Solution is trivial and as follows, given that $m_t \geq 0$.

From t = 1 to T-1: If $(h_{0t} + m_t - m_{t+1}) \geq 0$, then $I_{0t} = 0$ else $I_{0t} = \infty$

For $t = T$: $I_{0T} = 0$ and the last term is constant

Therefore, we let $h_{0t} + m_t \geq m_{t+1}$ where $I_{0t} = 0 \;\; \forall t$ in the optimal solution.

Distribution Problem (DIST)

It involves variables regarding vehicle routes. The following is the distribution subproblem.

$$DIST\text{: Min } \sum_{j\in N}\sum_{v=1}^{V}\sum_{t=1}^{T}(f_t + c_{0j})W_{0jvt} + \sum_{\substack{i\in N \\ i\neq j}}\sum_{j\in \overline{N}}\sum_{v=1}^{V}\sum_{t=1}^{T} c_{ij}W_{ijvt}$$

$$+\sum_{i\in N}\sum_{v=1}^{V}\sum_{t=1}^{T}(-a_{ivt} - b_{vt})X_{0ivt} + \sum_{\substack{i\in N \\ i\neq j}}\sum_{j\in N}\sum_{v=1}^{V}\sum_{t=1}^{T}(-a_{jvt} + a_{ivt})X_{ijvt}$$

s.t.

$$\sum_{j\in N} W_{0jvt} \leq 1 \qquad \forall v,t \qquad (8)$$

$$\sum_{\substack{j\in \overline{N} \\ i\neq j}} W_{ijvt} = \sum_{\substack{j\in \overline{N} \\ j\neq i}} W_{jivt} \qquad \forall i\in \overline{N}, \forall v,t \qquad (9)$$

$$X_{ijvt} \leq KW_{ijvt} \qquad \forall i\in \overline{N}, j\in N, i\neq j, \forall v,t \qquad (10)$$

$$X_{i0vt} = 0 \qquad \forall i\in N, \forall v,t \qquad (7b)$$

$$X_{ijvt} \geq 0 \qquad \forall i\in \overline{N}, j\in N, i\neq j, \forall v,t$$

$$W_{ijvt} \in \{0,1\} \qquad \forall i,j\in \overline{N}, i\neq j, \forall v,t$$

The *DIST* subproblem is a mixed integer program, which decomposes with respect to each vehicle and time period. However, decomposed subproblems are still not easy to solve, especially for large instances. In order to make the model tighter, we add the following valid inequalities.

- $\sum_{j \in \bar{N}} W_{ijvt} \leq 1 \qquad \forall i \in N, \forall v, t \qquad (17)$
- $\sum_{j \in N} W_{ijvt} \leq \sum_{j \in N} W_{0jvt} \qquad \forall i \in N, \forall v, t \qquad (18)$
- $X_{0jvt} \leq KW_{0jvt} \qquad \forall j \in N, \forall v, t \qquad (19)$
- $X_{ijvt} \leq (K - S_i + I_{i0})W_{ijvt} \qquad \forall i, j \in N, \forall v \quad for\ t = 1 \qquad (20)$
- $X_{ijvt} \leq (K - d_{i,t-1})W_{ijvt} \qquad \forall i, j \in N, v, t \in \{2,...,T\} \qquad (21)$
- $X_{ijvt} - S_j \leq \sum_{\substack{k \in N \\ k \neq i \\ k \neq j}} X_{jkvt} \qquad \forall i \in \bar{N}, j \in N, v, t \qquad (22)$

Inequality (17) ensures that only one retailer *j* (or supplier) can be visited from retailer *i* via vehicle *v* in period *t*. Constraint (18) states that a visit from retailer *i* to a retailer *j* (or supplier) can only be done if the vehicle *v* departs from the supplier in period *t*. Constraints (19), (20) and (21) are used in place of (10). Constraint (19) guarantees that if a visit from the supplier to retailer *j* occurs, distance based transportation is incurred. Constraints (20) and (21) have the same purpose as (19), but they are tighter since we deduce the minimum amount that can be delivered to retailer *i* in period *t*, which is $S_i - I_{i0}$ for $t = 1$ and $d_{i,t-1}$ for $t \in \{2,...,T\}$. We deduce these minimum amounts since we know that if vehicle *v* visits retailer *i* before *j*, from retailer *i* to *j*, it should carry an amount at most capacity of the vehicle less minimum amount delivered to retailer *i*. Last inequality (22) is the most effective among them, which ensures that for each vehicle-period pair (*v*, *t*), amount transported from retailer *j* to *k* is at least the amount transported from retailer *i* to *j* less maximum amount that can be delivered to retailer *j* (S_j).

After adding the valid inequalities, the problem needs more effort to find out the optimal solution. For instance, our preliminary experiments showed that a problem with size *N*=8 and *T*=5 takes 55 seconds per iteration to solve *DIST* to optimality, on average. Given that the problem will be solved in each iteration of the algorithm, we develop different approaches to solve *DIST* problem on the basis of time versus solution quality trade-off. We propose two approaches. First one is to solve the entire model to optimality, while the second one is to solve LP relaxation of the model to obtain a lower bound to the subproblem.

Retailers' Replenishment Problem (RETAILER)

It comprises variables concerning replenishment amounts to retailers as well as inventory levels of retailers and is given below.

RETAILER: Min $\sum_{i\in N}\sum_{v=1}^{V}\sum_{t=1}^{T}(m_t + a_{ivt} + b_{vt})Q_{ivt} + \sum_{i\in N}\sum_{t=1}^{T} h_{it} I_{it}$

s.t.

$$I_{it} = I_{i,t-1} + \sum_{v=1}^{V} Q_{ivt} - d_{it} \qquad \forall i \in N, \forall t \qquad (3')$$

$$\sum_{v=1}^{V} Q_{ivt} + I_{i,t-1} \le S_i \qquad \forall i \in N, \forall t \qquad (11')$$

$$\sum_{v=1}^{V} Q_{ivt} \le S_i z_{it} \qquad \forall i \in N, \forall t \qquad (12')$$

$$(S_i - I_{i,t-1}) - \sum_{v=1}^{V} Q_{ivt} \le S_i (1 - z_{it}) \qquad \forall i \in N, \forall t \qquad (13')$$

$$I_{it} \ge 0, \forall i \in \overline{N}, \forall t;\ Q_{ivt} \ge 0, \forall i \in N, \forall v, t;\ z_{it} \in \{0,1\}, \forall i \in N, \forall t$$

RETAILER subproblem may seem hard to solve to optimality at first sight because it is a mixed integer programming problem. However, it decomposes with respect to retailers (one of our goals at the beginning of relaxation). For each retailer, we transform the problem into a shortest path problem for which an exact polynomial algorithm based on standard dynamic programming (DP) is developed. The algorithm is described in detail in section 4.2.1.

3.2.2 Computation of upper bound

To construct a feasible solution to the problem IRO, we use the solution obtained from the *RETAILER* subproblem. Note that the solution of *RETAILER* gives the set of retailers to be visited as well as amounts to be shipped for each vehicle-period pair (Q_{ivt}). The importance of Q_{ivt} values is that we are able to construct feasible tours for each vehicle-period pair and determine amounts ordered to supplier in each period using these Q_{ivt} values. Note that the latter issue is resolved by defining a Wagner-Whitin type inventory problem for the supplier. Next, we describe the procedure used to obtain a feasible solution to IRO using Q_{ivt} values in detail.

Total of the Q_{ivt} values over *i* may be infeasible due to the capacity of vehicles. To cope with such an infeasibility and obtain feasible Q_{ivt} values, a linear programming model (LP) is developed, which reallocates Q_{it} values already computed (ignoring vehicle to which the amount assigned) to vehicles taking into account capacities of vehicles. Given $Q_{it} = \sum_{v=1}^{V} Q_{ivt}$ values for each *i*, *v*, *t*, the proposed LP is as follows:

$$\text{Min} \sum_{i \in N} \sum_{v=1}^{V} \sum_{t=1}^{T} (m_t + a_{ivt} + b_{vt}) Q_{it} B_{ivt}$$

s.t.

$$\sum_{i \in N} Q_{it} B_{ivt} \leq K \qquad \forall v, t$$

$$\sum_{v=1}^{V} B_{ivt} = 1 \qquad \forall i \in N, t$$

$B_{ivt} \geq 0 \qquad \forall i \in N, v, t,$

where B_{ivt} is the fraction of replenishment size shipped to retailer ***i*** via vehicle ***v*** in period ***t***.

The above LP model can also be cast into a transportation model as follows:

$$\text{Min} \sum_{i \in N} \sum_{v=1}^{V} \sum_{t=1}^{T} (m_t + a_{ivt} + b_{vt}) U_{ivt}$$

s.t.

$$\sum_{i \in N} U_{ivt} \leq K \qquad \forall v, t$$

$$\sum_{v=1}^{V} U_{ivt} = Q_{it} \qquad \forall i \in N, t$$

$$U_{ivt} \geq 0 \qquad \forall i \in N, v, t,$$

where U_{ivt} is the amount shipped to retailer ***i*** via vehicle ***v*** in period ***t***.

Another way to handle infeasibility with respect to vehicle capacities and obtain feasible set of retailers to be visited for each vehicle-period pair is to apply Generalized Assignment Heuristic (GAH) of Fisher and Jaikumar (1981).

Since the set of retailers to be visited for each vehicle-period pair are known (U_{ivt} values) and feasible with respect to vehicle's capacities now, the problem of finding feasible tours transforms into solving ***V*T*** many Traveling Salesman Problems[†] (TSPs). We solve a TSP for each vehicle and period pair using the CONCORDE to find feasible tours. Last action to reach a feasible solution is related to the order policy of the supplier in that inventory balance at the supplier may not hold due to a possible imbalance between ordered amounts and distributed amounts. Given that

[†] TSP is a well-studied combinatorial optimization problem (see e.g., Lawler et al., 1985), which can be stated as: Given *n* nodes and costs of traveling between these nodes, it seeks the least cost tour subject to visiting all of nodes and returning to the starting base. Despite TSP is known to be an NP-hard problem, there exists an incredibly efficient solver called CONCORDE (http://*www.tsp.gatech.edu/concorde*) that solves moderate size problems very quickly.

$\sum_{i \in N} \sum_{v=1}^{V} Q_{ivt}$, for each t is a demand from supplier, a Wagner-Whitin type inventory problem (uncapacitated lot-sizing) is defined and solved to decide on how much to order to the supplier in each period. This problem is solved optimally by using the Wagner-Whitin algorithm. After performing these steps, we obtain a feasible solution that is an upper bound to the problem IRO.

3.2.3 Solution of the Lagrangian dual problem

In this section, we describe the subgradient optimization algorithm used to solve *LDP*. The following is the step length computation, which together with the gradient and current values of multipliers determines the new values of multipliers.

$$r^k = \frac{a^k (UB^* - LB^k)}{denom_1 + denom_2 + denom_3}$$

where, $r^k =$ Step length in k^{th} iteration

$a^k =$ Scalar in k^{th} iteration where $0 \le a^k \le 2$

$UB^* =$ Best known upper bound obtained

$LB^k =$ Lagrangian lower bound obtained in k^{th} iteration

$$denom_1 = \sum_{t=1}^{T} (I_{0t}{}^k - I_{0,t-1}{}^k - P_t^k + \sum_{i \in N} \sum_{v=1}^{V} Q_{ivt}{}^k)^2$$

$$denom_2 = \sum_{i \in N} \sum_{v=1}^{V} \sum_{t=1}^{T} (-\sum_{\substack{j \in \bar{N} \\ j \neq i}} X_{jivt}{}^k + \sum_{\substack{j \in \bar{N} \\ i \neq j}} X_{ijvt}{}^k + Q_{ivt}{}^k)^2$$

$$denom_3 = \sum_{v=1}^{V} \sum_{t=1}^{T} (-\sum_{j \in N} X_{0jvt}{}^k + \sum_{i \in N} Q_{ivt}{}^k)^2$$

The new values of multiplies are computed as follows:

$$m_t^{k+1} = m_t^k + r^k (I_{0t}^k - I_{0,t-1}^k - P_t^k + \sum_{i \in N} \sum_{v=1}^{V} Q_{ivt}^k) \qquad \forall t$$

$$a_{ivt}^{k+1} = a_{ivt}^k + r^k (-\sum_{\substack{j \in N \\ j \neq i}} X_{jivt}^k + \sum_{\substack{j \in N \\ i \neq j}} X_{ijvt}^k + Q_{ivt}^k) \qquad \forall i \in N, \forall v, t$$

$$b_{vt}^{k+1} = b_{vt}^k + r^k (-\sum_{j \in N} X_{0jvt}^k + \sum_{i \in N} Q_{ivt}^k) \qquad \forall v, t$$

Note that we check if the condition $h_{0t} + m_t \geq m_{t+1}$ holds for newly computed m_t values. If that condition does not hold for any m_t, we modify that m_t to satisfy the condition.

3.2.4 An overview of the Lagrangian relaxation based solution approach

In this section, we present the generic LR based solution approach and mention about variants of LR based solution approach that will be used in computational study.

The generic LR based solution approach is depicted in the Figure 3.1.

Variants of LR based approach result from different solution procedures applied to *DIST*. We propose two ways to solve *DIST*. First one is to solve *DIST* to optimality using CPLEX, what we refer to as *LR1* algorithm. The second one is to solve LP relaxation of *DIST* to get a lower bound for the *DIST* using CPLEX and we call this as *LR2* algorithm. Thus, *LR1* and *LR2* only differ from each other on the solution of *DIST*, on the other hand, all other subproblems are solved to optimality as discussed in this chapter.

In addition, we propose another LR based solution approach, which is to use *LR1* or *LR2* within an enumeration tree constructed by forcing each vehicle either to depart or stay at the supplier in each period, referred to as the enhanced Lagrangian relaxation (ELR). This procedure will be discussed in detail in section 5.2.3.

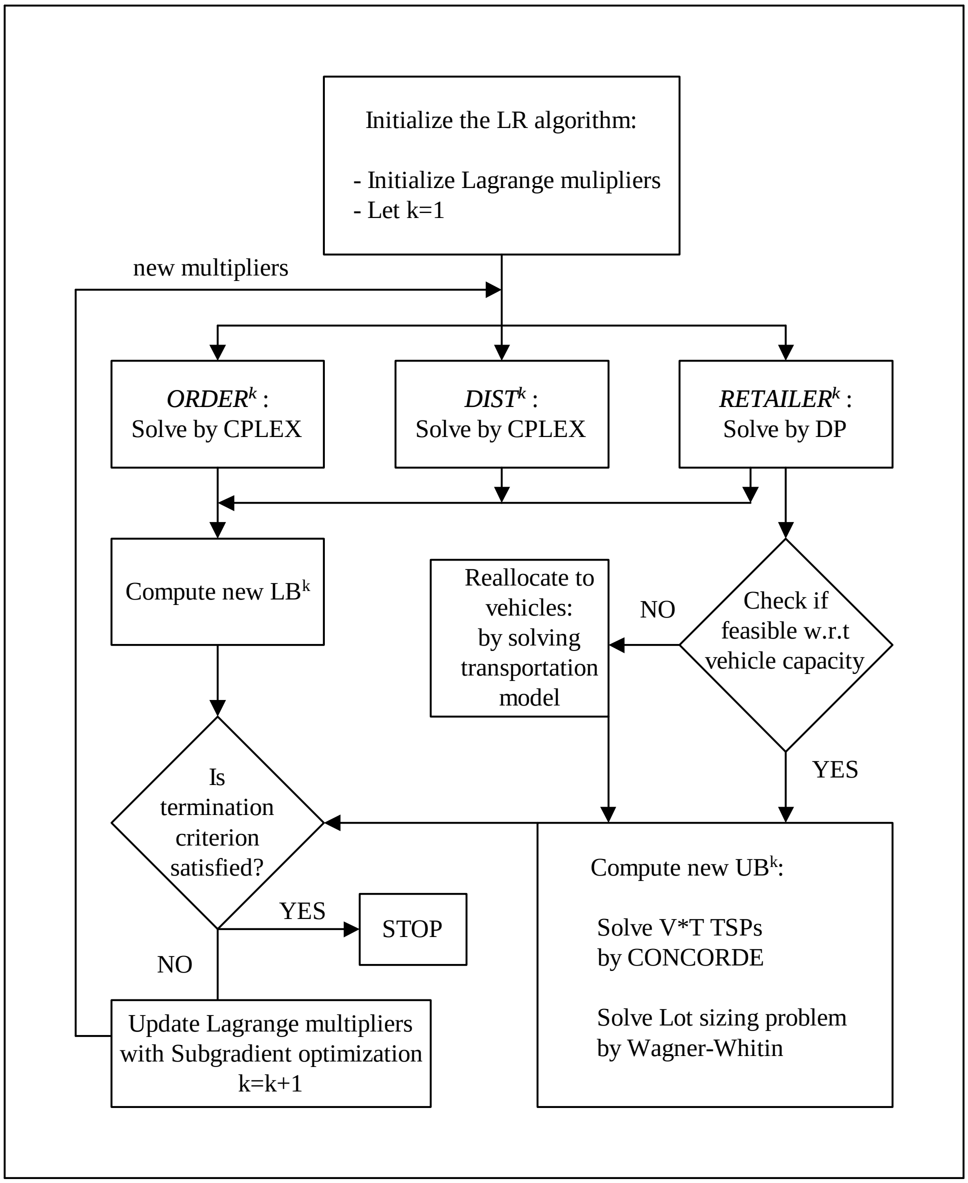

Figure 3.1 Flowchart of the generic LR based solution approach

CHAPTER 4

SINGLE SUPPLIER-SINGLE RETAILER PROBLEM

In this chapter, we address a special case of the problem IRO. A single supplier-single retailer problem is considered instead of general case of multiple retailers. Thus, the special problem is to simultaneously determine when to visit the retailer and the quantity of shipment besides the quantity of product to be ordered to the supplier in each time period so that sum of fixed order (setup) and inventory holding costs at the supplier and retailer as well as the transportation cost are minimized.

This problem is important for its own sake and its general (multiple retailer problem with order-up-to level policy) case. One reason is that vendor managed resupply is an emerging trend in logistics, as stated in Campbell et al. (1998) and the other is that a relaxation based solution approach to the multiple retailer problem results in subproblems composed of independent single retailer problems that need to be solved many times. Therefore, the scope of this chapter is twofold. One is to study the integrated problem with single supplier and single retailer and the other is to develop an efficient method for solving the single retailer problem.

In this chapter, we first give a mathematical programming formulation of the problem and then describe the solution approach proposed. Finally, we conclude the chapter.

4.1 Formulation of the model

We consider a logistic system in which a single product is delivered from supplier, denoted by 0, to retailer, denoted by 1, over a set $t = \{1, 2, ..., T\}$ of discrete time periods. Retailer faces deterministic demand d_t in each time period $t \in t$ and has a minimum (maximum) inventory level s (S') of the product. In each time period *t*, supplier may place an order of size P_t of the product. A fixed cost f_{0t} (f_{1t}) is incurred every time *t*, supplier (retailer) places an order. An inventory holding cost h_{it} is charged for each unit of product kept at the end of period *t* at location i (i=0,1). Shipments from supplier to retailer are made by a vehicle with the capacity $K \geq S' - s$. There are transportation costs from supplier (retailer) to retailer (supplier) $c_{01}(c_{10})$. The problem is to find the optimal replenishment policy that minimizes the total of fixed, inventory holding and transportation costs. It integrates lot-sizing decisions for supplier and order up-to level replenishment decisions for retailer and is called integrated lot sizing and order-up-to level problem (ILSOP). Let $y_t = 1$ if an order for supplier is given in period $t \in t$ and 0 otherwise. Let I'_{1t} and I_{0t} be the inventory levels of retailer and supplier at the end of time *t*, respectively. The retailer has to be replenished before its inventory level I'_{1t} drops below the minimum level *s*. If retailer receives a shipment Q_t from supplier in a period *t*, its inventory level I'_{1t} is brought up to maximum level S', i.e., $Q_t = S' - I'_{1,t-1}$. Let $z_t = 1$ if a delivery is made to retailer in period $t \in t$ and 0 otherwise. The beginning inventory levels both at the supplier I_{00} and at the retailer I'_{10} are known. Below, we present a mixed integer linear programming formulation adapted from the model of the general case in section 3.1. The model is given in its simplified form as in section 3.1 and is as follows.

ILSOP

Minimize $\sum_{t=1}^{T} (f_{0t} y_t + h_{0t} I_{0t} + h_{1t} I_{1t} + (f_{1t} + c_{01} + c_{10}) z_t + h_{1t} s)$ (1')

Subject to

$I_{0t} = I_{0,t-1} + P_t - Q_t$ $\quad \forall t$ (2)

$I_{1t} = I_{1,t-1} + Q_t - d_t$ $\quad \forall t$ (3')

$$P_t \leq M_0 y_t \qquad \forall t \qquad (4)$$

$$Q_t + I_{1,t-1} \leq S \qquad \forall t \qquad (11')$$

$$Q_t \leq S z_t \qquad \forall t \qquad (12')$$

$$(S - I_{1,t-1}) - Q_t \leq S(1 - z_t) \qquad \forall t \qquad (13')$$

$$I_{0t}, I_{1t}, P_t, Q_t \geq 0;\; y_t, z_t \in \{0,1\} \qquad \forall t \qquad (14')$$

Note that we simplified the model as in section 3.1 by making the transformations: $I_{1t} = I'_{1t} - s$ and $S = S' - s$. Again note that we have also deduced the minimum inventory level of the retailer (s) from its beginning inventory level, i.e., $I'_{10} - s = I_{10}$. Thus, as of now, we do not allow stockout (falling below zero) at the retailer instead of falling below minimum inventory level (s), since we have already allocated the minimum inventory level of the retailer.

We do not explain the objective function and constraints, since they correspond to the same expressions of general model and have already been explained in section 3.1.

The model may seem hard to solve to optimality at first sight. However, in the next section, we develop a polynomial time algorithm for the exact solution of the model.

4.2 Solution approach

While working on the general case of the problem (multiple retailers), Bertazzi et al. (2002) propose an improvement type heuristic in which, they solve single retailer problems repeatedly. They construct an acyclic network for a single retailer problem and solve it to obtain delivery times as well as quantities. We also transform single retailer problem into a network problem but our network (presented in section 4.2.1) is different than that of Bertazzi et al. (2002). In addition to this, we extend the single retailer problem to include supplier as well. Below is the solution approach we propose.

The method proposed for solving *ILSOP* consists of two stages for a network representation of the problem. In the first stage, a network representing only the replenishment of retailer is constructed. In this network, nodes denote inventory level of retailer at the end of a period and arcs denote amount of product delivered to the retailer at the beginning of a period. In the second stage, the network is extended to account for the coordinated replenishment of both supplier and retailer. In the extended network, nodes denote inventory levels of both supplier and retailer and arcs denote amounts of product ordered to both supplier and retailer. Thus, finding shortest path over the extended network yields the optimal solution to *ILSOP*.

4.2.1 Replenishment of retailer

As already mentioned, in the first stage only the retailer's replenishment problem with order up-to-level inventory policy is considered, ignoring the presence of supplier. In other words, the amount of product ordered to (is made available at) the supplier is taken as a priori assuming that there is sufficient amount of inventory at the supplier whenever needed. Since it is known that in every period retailer either places an order that brings its inventory level to S due to the order up-to level policy or not, it is easy to construct a network to represent the decisions to be made by the retailer. We refer to this network as the initial network.

In the initial network, we disseminate two arcs from each node where nodes denote the inventory level of retailer at the end of a period and arcs denote the amount of product supplied to retailer at the beginning of a period. We let the very first node represent the initial inventory level of retailer and generate two arcs from that origin node. The first arc emanating from the origin node indicates that an order corresponding to the order up-to amount less the value of origin node is placed and is connected to the node representing the inventory level at the end of the first period, which is equal to the order up-to amount less the amount of demand in the first period. The second arc indicates that no order is placed and enters the other node where the inventory level equals the initial inventory level less the amount of demand in the first period. Having two nodes for the first period, we again generate two arcs from each and connect them to the nodes denoting possible inventory levels of retailer at the end of second period. This procedure is repeated until examining all periods of time horizon, so that all possible nodes are generated.

One might expect to have 2^T many nodes in the initial network since always two decisions, to place an order or not, are given at each node. However, this is not the case because the network constructed has a special feature. First, nodes having negative inventory levels are not generated since backlogging is not allowed. Next, arcs, which are disseminated from different origins (nodes) at the same period and corresponding to order placement, arrive at the same destination (node) at the next period leading to a network in which the number of nodes of a period is at most one more than that of previous period. We assume that total demand ($d_1 + d_2 + ... + d_T$) is greater than the initial inventory level of retailer to eliminate the trivial case (placing no order for the retailer is the optimal policy). Due to that assumption the network, in the worst case in terms of size, will have all the possible nodes generated by the above described procedure except the node resulting from deducing the last period's demand d_T from $I_{10} - (d_1 + d_2 + ... + d_{T-1})$, as shown in dashed lines in Figure 4.1.

In the sequel, we define nodes and arcs formally. Let N_t be the set of nodes and A_t be the set of arcs in period *t*. Then, the set of nodes and arcs can be defined as follows.

$$N_0 = I_{10},\ N_T = \left\{ S - \sum_{m=1}^{x} d_{T-m+1} \ \middle|\ 1 \le x \le T \right\} \text{ and}$$

$$N_t = \left\{ S - \sum_{m=1}^{x} d_{t-m+1} \;\middle|\; 1 \le x \le t \right\} \mathbf{U} \left\{ I_{10} - \sum_{m=1}^{t} d_m \right\} \; \forall t = 1, 2, ..., T-1 \tag{23}$$

$$A_t = \left\{ \{0\} \mathbf{U} \{ S - I_{1,t-1} \} \;\middle|\; I_{1,t\text{-}1} \in N_{t-1} \right\} \forall t = 1, 2, 3, ..., T \tag{24}$$

Note that by above set of nodes N_t, we mean only nonnegative nodes but we do not specify this situation for the sake of easiness of notation. In other words, despite some of nodes defined by N_t may take negative values, they are not valid nodes and are not included in the set N_t. Also, A_t defines arcs emanating from nodes of period *t*-1 in period *t*, but it does not designate to which nodes these arcs are linked. Taking the difference between nodes of N_t and N_{t-1} and adding the demand at that period, one can find out to which nodes the arcs are linked.

The number of nodes equals (1 + 2 + 3 + … + *T*+1) – 1 = (*T*+1)(*T*+2)/2 – 1 and the number of arcs equals (2 + 4 + 6 + … + 2*T*) – 1 = *T*(*T*+1) – 1. As a result, the initial network has $O(T^2)$ nodes and arcs. An example network for a four-period time horizon problem using (23) and (24) is presented in Figure 4.1.

All feasible nodes and arcs are enumerated in the initial network. There is no other possible node or arc other than those defined by N_t and A_t, because such a case would violate the order-up-to level policy. Thus, it suffices to consider only the inventory levels of retailer defined by N_t for an optimal solution to the single retailer problem (initial network).

Finding the minimum cost path on the initial network using standard dynamic programming approach solves the single retailer problem with order up-to-level inventory policy in $O(T^2)$ time as follows. Let $F_t(I)$ be the minimum cost of having an inventory I at the end of period *t*, where $I \in N_t$ for $t = 1, 2, ..., T$, and let

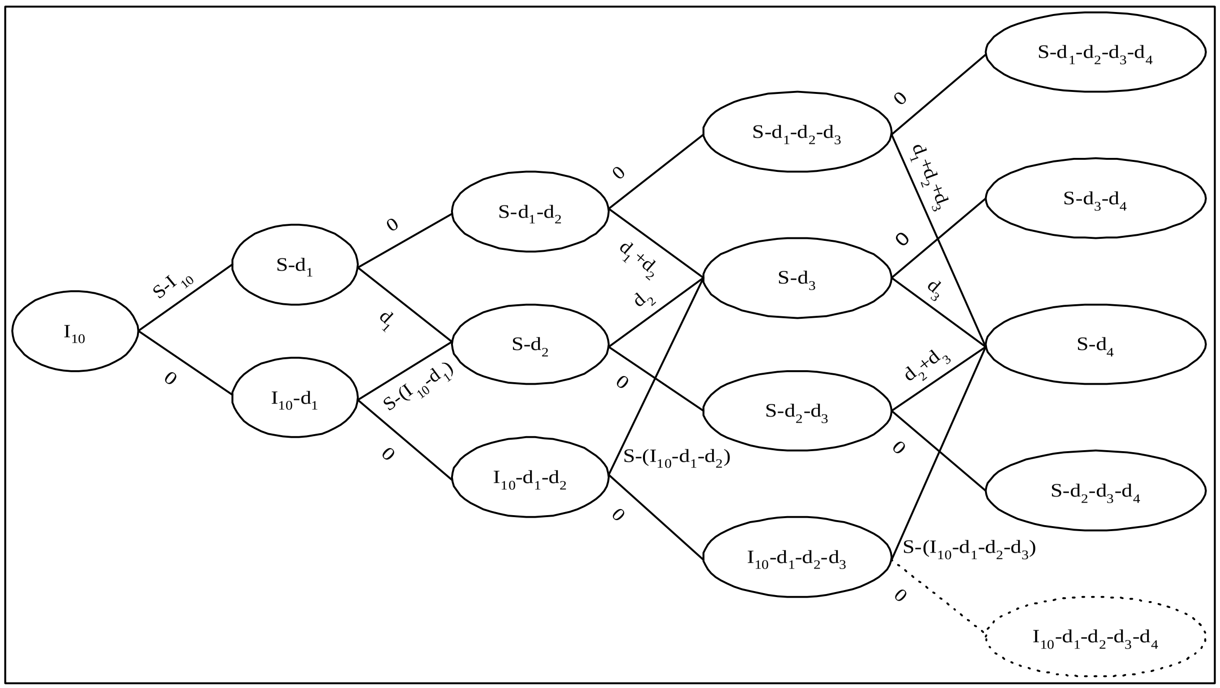

Figure 4.1 Initial network for $T = 4$

$g_t(Q_t)$ be the cost of placing an order of size Q_t. Note that $g_t(Q_t)$ $=(f_{1t}+c_{01}+c_{10})$ if $Q_t>0$ and 0 otherwise. Let $F_0(I_{10})=0$, and for each $t=1,2,...,T$, we have

$$F_t(I)=h_{1t}I+\min\left\{F_{t-1}(\bar{I})+g_t(Q_t)\ \middle|\ \bar{I}\in N_{t-1};\ Q_t\in A_t\right\},\ \forall I\in N_t \tag{25}$$

Then, the optimal solution is computed from $\min\left\{F_T(\bar{I})\ \middle|\ \bar{I}\in N_T\right\}$.

4.2.2 Coordinated supplier and retailer replenishments

In the second stage, we consider the replenishments of supplier and retailer together, and thus incorporate decisions given at the supplier into the initial network. The resulting network is called the extended network. In the extended network, a node is defined by two attributes. The first attribute denotes the ending inventory level of retailer in the initial network. The second one represents the inventory level of supplier at the end of a period. It is easy to see that the characteristic of optimal replenishment policy at the supplier is the same as the well-known Wagner-Whitin property (Wagner and Whitin, 1958) regardless how the retailer is replenished because the supplier replenishment problem is equivalent to the uncapacitated lot sizing problem in Wagner and Whitin (1958). In the Wagner-Whitin property, the amount of product ordered in a period is the sum of demands at a set of future periods. Similarly, given the demand pattern of retailer over the planning horizon, which is designated by the arcs of the initial network, the supplier in a period orders the total amount of retailer's demands in a set of next periods. Thus, the second attributes of nodes in the extended network can be constructed using the Wagner-Whitin property. However, in our case, there is no single realizable demand in a period, but a set of demands owing to the order up-to level policy employed by the retailer. Nevertheless, by following paths of the initial network, we can construct the second attribute of the extended network. We assume $I_{00}=0$ for simplicity, and

$I_{0T} = 0$ without loss of generality. The procedure regarding the extension of the initial network is as follows. In the first period, we generate T nodes for each node of initial network, thus having $2T$ many nodes in total. The first T nodes (the next T nodes) are constructed by considering an order (no order) is placed by retailer in period 1. We have T nodes for each node of initial network, because supplier may place an order size of only first period's demand, summation of first and second period's demand, …, and summation of all period's demand of retailer, as stated by Wagner-Whitin. To illustrate, the amount ordered by the supplier in period 1 can be $S - I_{10}$, S- $(I_{10} - d_1)$,..., or S-$(I_{10} - d_1 - ... - d_{T-1})$ if an order is placed by retailer in period 1 and 0, S - $(I_{10} - d_1)$, …, or S - $(I_{10} - d_1 - ... - d_{T-1})$ if no order is placed by retailer in period 1. In the second period, we generate $T-1$ nodes for each node of initial network, since supplier may place an order size considering the second period's demand and the remaining periods' demands of retailer using Wagner-Whitin property. This leads to $3(T-1)$ many nodes in the second period of extended network. This procedure is repeated for all the remaining periods of time horizon and the nodes of the extended network are constructed accordingly. On the other hand, all nodes of the extended network are not connected to each other (i.e., it is not a complete network). The nodes in period t with second attribute equal to 0 are connected to $2(T-t+1)$ nodes of period $t+1$, since it is possible for the supplier to order the sum of demands of a set of future periods (multiple 2 is due to retailer's policy of placing order or not). The nodes with second attribute corresponding to the inventory carried only for the next period's demand of retailer are connected to only one node. Finally, the nodes with second attribute corresponding to the inventory carried for more than the next period's demand of retailer are connected to two nodes, because of the retailer's policy of placing an order or not in a given period. In the sequel, we define states and arcs formally. Let $N_t^e = (I_{1t}, I_{0t})$ be the set of all nodes of the extended network in period *t* and A_t^e be the set of all arcs of the extended network in period *t*. Then, the set of states and arcs can be defined as follows.

$$N_0^e = (I_{10}, 0),\ N_T^e = \left\{ S - \sum_{m=1}^{x} d_{T-m+1},\ 0 \;\middle|\; 1 \le x \le T \right\} \text{ and}$$

$$N_t^e = \left\{ S - \sum_{m=1}^{x} d_{t-m+1},\ \{0\} \;\mathbf{U}\; \{ \sum_{n=t-x+1}^{y} d_n \mid t \le y \le T-1\} \;\middle|\; 1 \le x \le t \right\} \mathbf{U}$$

$$\left\{ I_{10} - \sum_{m=1}^{t} d_m ,\ \{0\} \;\mathbf{U}\; \{S - (I_{10} - \sum_{m=1}^{z} d_m) \mid t \le z \le T-1\} \right\} \forall t = 1, 2, \ldots, T-1 \qquad (26)$$

$$A_1^e = \left\{ S - I_{10} ,\ \{S - (I_{10} - \sum_{m=1}^{x} d_m) \mid 0 \le x \le T-1\} \right\}$$

$$\mathbf{U} \left\{ 0,\ \{0\} \;\mathbf{U}\; \{S - (I_{10} - \sum_{m=1}^{x} d_m) \mid 1 \le x \le T-1\} \right\} \text{ and}$$

$$A_t^e = \left\{ \sum_{m=1}^{t-1} d_m ,\ \{\sum_{m=1}^{x} d_m \mid 1 \le x \le T-1\} \right\} \mathbf{U} \left\{ 0,\ \{0\} \;\mathbf{U}\; \{\sum_{m=1}^{x} d_m \mid t \le x \le T-1\} \right\}$$

$$\mathbf{U} \left\{ S - (I_{10} - \sum_{m=1}^{t-1} d_m),\ \{S - (I_{10} - \sum_{m=1}^{x} d_m) \mid 1 \le x \le T-1\} \right\}$$

$$\mathbf{U} \left\{ 0,\ \{0\} \;\mathbf{U}\; \{S - (I_{10} - \sum_{m=1}^{x} d_m) \mid t \le x \le T-1\} \right\} \text{ if } I_{0t} = 0,$$

$$\left\{ \sum_{m=1}^{t-1} d_m ,\ 0 \right\} \mathbf{U} \left\{ \{S - (I_{10} - \sum_{m=1}^{x} d_m) \mid 1 \le x \le T-1\},\ 0 \right\}$$

$$\mathbf{U} \left\{ 0,\ 0 \mid I_{0t} + I_{1t} \ne S \right\} \text{ if } I_{0t} > 0,\ \forall \mathrm{t} = 2, 3, \ldots, \mathrm{T} \qquad (27)$$

Again note that by the above set of nodes N_t^e, we mean only nonnegative nodes. Here, A_t^e defines just the set of all arcs in period *t*. Backtracking between nodes of N_t^e and N_{t-1}^e, one can find out to which nodes the arcs are linked.

For the extended network, the number of nodes equals $1 + \sum_{t=1}^{T} [(t+1)(T-t+1)] - 1$ $= \frac{T^3}{6} + T^2 + \frac{5T}{6}$ and the number of arcs is equal to $2T$

$$+\sum_{t=2}^{T} t\left[2(T-t+1)+2(T-t)+1\right]-1=\frac{2T^3}{3}+\frac{3T^2}{2}-\frac{7T}{6}$$. Hence, the extended network has $O(T^3)$ nodes and arcs in total. Using (26) and (27), an example network for the four-period time horizon problem is given in Figure 4.2.

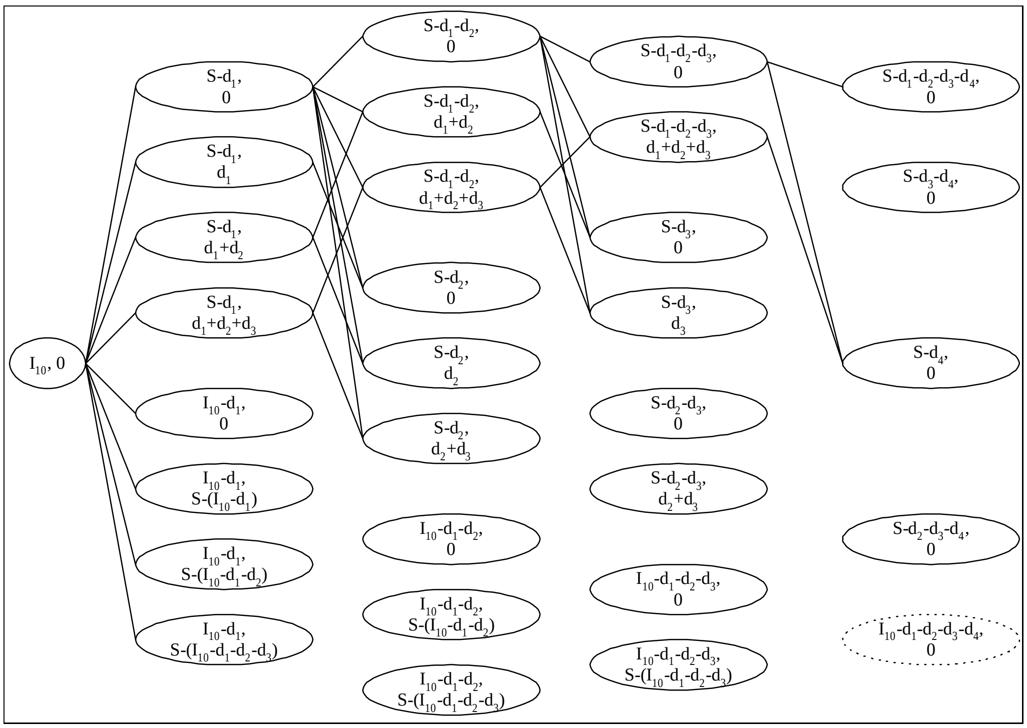

Figure 4.2 Extended network for $T = 4$

Theorem. For an optimum solution to the integrated lot sizing and order up-to-level problem for a single supplier-single retailer, it suffices to consider only the inventory levels of both supplier and retailer specified by N_t^e.

Proof. It will be shown that a state not belonging to N_t^e cannot be in the optimal solution. We have already shown that inventory levels specified by N_t constitute all possible states for the retailer. Thus, considering only inventory levels of supplier given that of retailer is adequate to make the proof.

Let $X_1 = (S - d_1, d_1 + 1)$ be a state in period 1, which is constructed by ordering an amount only one unit greater than the sum of demands of retailer in periods 1 and 2. That is, X_1 does not satisfy Wagner-Whitin property. Then there are two cases to consider:

Case1: Suppose that an order for supplier is placed in period 2. Assuming that each state in N_2^e has a known minimum cost path over the network defined by N_t^e ($2 \le t \le T$), a path from X_1 to any state $\in N_2^e$ is more costly than a path from at least one state $\in N_1^e$ to the same state in N_2^e as from X_1. To illustrate, Cost ($N_0^e \to X_1 \to (S - d_2, d_2)$) is higher than Cost ($N_0^e \to (S - d_1, 0) \to (S - d_2, d_2)$). If we generalize, ordering just the amount demanded in a period is always less costly than ordering more than the sum of a set of future demands in this case.

Case 2: Suppose that no order for supplier is placed in period 2. Let $X_2 = (S - d_2, 1)$ be a state $\notin N_2^e$ and assume that each state in N_2^e has a known minimum cost path over the network defined by N_t^e ($2 \le t \le T$). Then, it will be sufficient to show that a path from X_1 to X_2 is more costly than a path from at least one state $\in N_1^e$ to a state $\in N_2^e$, assuming minimum cost path from X_2 to the end of network is known. For instance, Cost ($N_0^e \to X_1 \to X_2$) is higher than Cost ($N_0^e \to (S - d_1, d_1) \to (S - d_2, 0)$), because it is shown above in case 1 that there is always a path from a state $\in N_2^e$ cheaper than the path from a state (like X_2) not conforming to Wagner-Whitin property.

The same idea is also valid for other periods. Hence, this completes our proof.

It follows that the minimum cost path on the extended network can be found using standard dynamic programming approach in $O(T^3)$ time as follows. Let $F_0(I_{10}, 0) = 0$. For each $t = 1, 2, ..., T$, we have $(I_1, I_0) \in N_t^e$ and

$$F_t(I_1, I_0) = h_{1t} I_1 + h_{0t} I_0 + \min\left\{F_{t-1}(\bar{I}_1, \bar{I}_0) + g_t(Q_t, P_t) \;\middle|\; (\bar{I}_1, \bar{I}_0) \in N_{t-1}^e;\ (Q_t, P_t) \in A_t^e\right\} \qquad (28)$$

where $F_t(I_1, I_0)$ is the minimum cost of having an inventory I_1 at the retailer and I_0 at the supplier at the end of period *t*, and $g_t(Q_t, P_t)$ is the cost of placing an order of size Q_t for retailer and P_t for supplier. If $Q_t > 0$, g_t involves a cost term $(f_{1t} + c_{01} + c_{10})$, 0 otherwise. If $P_t > 0$, g_t also involves a cost term f_{0t}, 0 otherwise.

Then the optimal solution value to *ILSOP* is $\min\left\{F_T(\bar{I}_1, \bar{I}_0) \;\middle|\; (\bar{I}_1, \bar{I}_0) \in N_T^e\right\}$.

4.3 Conclusion

In this chapter, we considered a logistics system, consisting of a supplier and a retailer employing order up-to level inventory policy over multiple periods. We developed a mixed integer linear programming model and proposed a polynomial time solution method both for the single-retailer problem and for the single supplier-single retailer problem.

The initial network represents not only a first step towards constructing the full network to solve *ILSOP*, but also the problem arises as a subproblem in a relaxation based solution of the multiple retailer problem. Thus, solving initial network in $O(T^2)$ is important since it will be encountered in every iteration of the relaxation algorithm. Furthermore, as stated by Campbell et al. (1998), single retailer analysis is also valid for the case of multiple retailers provided that a vehicle serves only one retailer on a trip (direct delivery) and there is enough number of vehicles to serve all the retailers in a given period.

CHAPTER 5

COMPUTATIONAL RESULTS

In this chapter, we present our computational study performed by the Lagrangian relaxation (LR) based solution approach on test instances taken from literature. We first describe our computational framework. Then, we provide some preliminary results on small instances in three phases. Lastly, we present further experiments on large instances and discuss the results.

5.1 Computational framework

There are four phases of our experimentation, first three of which are based on different implementations of LR based solution approach and the last one being the benchmarking phase on large instances.

These phases are as follows:

1. Solving all the subproblems to optimality in the LR based solution approach. This implementation is called *LR1*.
2. Instead of solving the *DIST* subproblem to optimality, generating a lower bound by using the LP relaxation solution of the *DIST*. Other subproblems are solved to optimality. This implementation is called *LR2*.
3. Implementing an enhanced LR algorithm under the best settings as obtained from phases one and two (ELR).
4. Benchmarking on large instances using all the LR based approaches in their best settings.

In phase 1, we conduct some preliminary experiments in order to examine the behavior and determine the parameters of the LR based approach. For this purpose, we run the algorithm *LR1* for 100 iterations and implement three settings for the scalar of the subgradient optimization algorithm, which are to divide scalar ***a*** (i) by 1.005 per iteration, (ii) by 2 per 30 consecutive non-improving iterations and (iii) by 2 per 10 consecutive non-improving iterations. Thus, we determine the best setting for scalar ***a*** as well as convergence point (no. of iterations) of *LR1* in phase1.

Phase 2 is very similar to phase 1 in that we perform the experiments under the same settings with the same purposes as in phase 1 except that we do not solve *DIST* subproblem to optimality but generate a lower bound on it. Solving the *DIST* subproblem to optimality would be very time consuming in large instances, and thus impractical. For this reason, we implement a variant of *LR1*, in which we solve LP relaxation of *DIST* subproblem to generate a lower bound in a short time. We refer to this LR variant as *LR2*.

In the third phase, we implement enhanced versions of *LR1* and *LR2* under best settings obtained in the first two phases, what we refer to as *ELR1* and *ELR2*, respectively. Our aim in applying *ELR1* and *ELR2* is to get both better upper bounds and tighter lower bounds. The enhanced LR approach will be described in detail in section 5.2.3. In this phase, we conduct the same experiments as in phase 1 and 2.

In the last phase, for benchmarking, we compare our LR based algorithms with the exact solution of IRO using CPLEX and two intuitive heuristics on large instances. These two intuitive heuristics, called *Every* and *Latest*, are proposed by Bertazzi et al. (2002). As the names imply, *Every* heuristic replenishes retailers in every period, while *Latest* heuristic replenishes each retailer at the last possible period, that is one period after which retailer stock outs. In addition, for *Every*, one TSP is solved (using CONCORDE) to find the optimal sequence of retailers, which is the same for each period, whereas T many TSPs are solved (using CONCORDE) to find optimal sequence of retailers to be visited in each period in *Latest*. However, since our decision context includes order policy of the supplier as well, we modified these heuristics such that for both of them an uncapacitated lot-sizing problem is solved by means of Wagner-Whitin algorithm considering replenishment amounts to retailers as demands of supplier, in addition to their original form. We use modified ones when conducting experiments on our problem. These two heuristics are used in comparison, because even finding a feasible solution is very difficult for large instances let alone finding the optimal solution despite using a powerful solver, CPLEX. Thus, we aim at constructing a comparison base by using these heuristics.

For the easiness of presentation, we use the notation '***Name(a/b, n)***' to represent the algorithms. Here, '***Name***' denotes the name of the algorithm used. The expressions in brackets denote the setting used for subgradient optimization. For instance, ***LR1(a/2, 30)*** means we use ***LR1*** implementation of the LR based approach, in which scalar ***a*** is halved every 30 consecutive non-improving iterations.

The initial value of subgradient optimization scalar (*a*) is taken as two in all of the experiments. Also, we let all the Lagrange multipliers be nonnegative although they are originally unrestricted in sign due to corresponding equality expressions.

All the algorithms are coded in C. However, for the solution of LP problems (or LP relaxations) as well as MIP problems, Callable Library of CPLEX 8.1 is embedded into the C code. Furthermore, CONCORDE is called from the C code for solving TSPs. All the experiments are conducted on Pentium IV 1.6 GHz CPU PCs with 256 MB RAM.

Basic Test Instances

All test instances used in this study are taken from the literature, which are developed by Bertazzi et al. (2002).

The test instances generated by Bertazzi et al. (2002) have the following characteristics.

- Number of retailers (N): 50
- Time horizon (T): 30
- Number of vehicles: single
- Amount of product demanded from retailer i at time t (d_{it}): Constant over time (i.e., $d_{it} = d_i, \forall t$). Randomly generated as an integer from the interval $[10, 100]$.
- Amount of product shipped to supplier at time t (r_{0t}): $r_{0t} = \sum_{i \in N} d_i$
- Minimum inventory level at retailer i (s_i): Randomly generated as an integer from the interval $[50, 150]$.
- Maximum inventory level at retailer i (S_i'): $s_i + d_i g_i$. g_i is randomly selected from the set $\{2, 3, 5, 6, 10, 15, 30\}$, which represents the number of time units required to use up the amount $S_i' - s_i$.
- Beginning inventory level at the supplier (I_{00}): $\sum_{i \in N} d_i$
- Beginning inventory level at retailer (I_{i0}): $S_i' - d_i$
- Inventory holding cost at the supplier (h_0): 0.3, 0.8
- Inventory holding cost at retailer i (h_i): Randomly generated from the intervals $[0.1, 0.5]$ and $[0.6, 1]$.
- Transportation cost (c_{ij}): $\left\lfloor \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right\rfloor$, where coordinates x_i, x_j, y_i and y_j are randomly generated as an integer from the interval $[0, 500]$ and from the interval $[0, 1000]$.

Note that although there are several vehicle capacity levels in the original instances of Bertazzi et al. (2002), we removed them since we take the vehicle capacity sufficiently large ($K = \sum_{i \in N} S_i$), i.e., uncapacitated, in our setting. As Bertazzi et al. (2002), we assume single vehicle.

Bertazzi et al. (2002) have randomly generated 10 test instances for each combination of the three parameters h_0, h_i and c_{ij} (which makes eight combinations in total). Thus, there are totally 80 instances. These instances are numbered according to their characteristics as follows.

Characteristics of the basic test instances:

c_{ij}	h_i	h_0	Instance No.
[0, 500]	[0.6, 1]	0.3	1 - 10
		0.8	11 - 20
	[0.1, 0.5]	0.3	21 - 30
		0.8	31 - 40
[0, 1000]	[0.6, 1]	0.3	41 - 50
		0.8	51 - 60
	[0.1, 0.5]	0.3	61 - 70
		0.8	71 - 80

Instances generated by Bertazzi et al. (2002) are taken almost as they are. We eliminated the vehicle capacity data, but added fixed order as well as fixed vehicle dispatching cost data to the instances to make them appropriate for use in our context. In rest of the chapter, we will explain modifications we perform to the instances generated by Bertazzi et al. (2002) whenever necessary.

5.2 Preliminary experiments

This section includes first three phases of computational experiments, respectively. Next, test instances used in these phases are described.

Test instances for the first three phases

From the basic test instances described in section 5.1, we have derived 18 test problems with eight retailers and five periods for experimentation. Eight retailers selected among others (50 retailers) are the first eight retailers that require at least one replenishment during the 5-period time horizon. The other modifications are as follows.

For the 18 instances, the following data are added.

- Fixed order cost (f_{0t}): $\sum_{i \in N} h_0 d_i$
- Fixed vehicle dispatching cost (f_t):
    - $\sum_{i \in N} h_i d_i$ (for the first six instances)
    - $0.8 * \sum_{i \in N} h_i d_i$ (for the second six instances)
    - $0.2 * \sum_{i \in N} h_i d_i$ (for the last six instances)

First six instances are derived from basic test instances 1, 2, 19, 28, 31 and 32, while the second six instances are derived from basic test instances 3, 8, 15, 24, 36 and 37, but with less dispatching cost compared to first six instances. On the other hand, the last six instances are the same as the second six instances but with substantially less dispatching cost relative to the second six instances. By convention, we name the first 12 instances using a prefix "pre" and the last six instances using "npre" to distinguish between the second six instances and the last six instances.

In the following, we give the set of measures that we use for determining the behavior and performance of algorithms in three phases of our computations.

Performance Measures

- **%UL:** The percentage gap between the best feasible solution (UB) and the best lower bound (LB), i.e. %(UB-LB)/LB.
- **%UO:** The percentage gap between the best feasible solution (UB) and the optimal solution value (Opt), i.e. %(UB-Opt)/Opt.
- **%LO:** The percentage gap between the best lower bound (LB) and the optimal solution value (Opt), i.e. %(Opt-LB)/Opt.
- **%IGAP:** The percentage gap between the optimal solution value (Opt) and the solution value of the LP relaxation of the IRO (LPR), i.e. %(Opt-LPR)/Opt.
- **%Latest:** The percentage gap between the solution value found by Latest heuristic (*Latest*) and the best feasible solution (UB), i.e. %(*Latest*-UB)/UB.
- **%Every:** The percentage gap between the solution value found by Every heuristic (*Every*) and the best feasible solution (UB), i.e. %(*Every*-UB)/UB.
- **CPU *x*:** CPU time in seconds to solve *x*, where *x* may be *LR1*, *LR2*, *ELR1*, *ELR2*, LPR (LP relaxation of IRO) as well as Opt (Optimal solution of IRO).

5.2.1 Phase 1

In this phase, *LR1* algorithm is run for 100 iterations. For terminating the algorithm, we selected only one criterion (i.e., 100 iterations) but not any other criterion. This is because we just wanted to see when the algorithm converges. As already discussed three settings are tried for subgradient optimization scalar ***a***: ***a***/1.005; ***a***/2, 30; ***a***/2, 10.

The results for these settings as well as per iteration CPU times in seconds are provided in Table 5.1. According to the results, halving scalar after 30 consecutive non-improving iterations yields the best results in all of the measures considered. The gap between the best upper bound and best lower bound (%UL) is not satisfactory but as the %UO column indicates our *LR1*(***a***/2, 30) algorithm was able to find very good upper bounds. On average, percent deviation between best upper bound and the optimal solution value is %0.84 and the algorithm could find the optimal solution in 10 instances out of 16, which are indeed very good results. It is also observed from Table 5.1 that lower bounds found by the algorithm are not tight enough and lead to high %ULs. Nevertheless, lower bounds obtained are far better than LP relaxation values of the integrated original model as indicated by %LO and %IGAP columns. Moreover, results indicate that total CPU times are intolerably high for this algorithm to run. Although we could let the algorithm run for a less number of iterations (so less CPU time) since it converges long before reaching 100 iterations, CPU times are still very high even for such small instances. See Figure 5.1 for a typical convergence graph using ***LR1***(***a***/2, 30) for pre3 and Figure 5.2 for a typical graph depicting relationship between CPU time and number of iterations. For ***LR1***(***a***/2, 30) (best implementation of ***LR1***), the figures regarding convergence for all instances are given in Appendix B.

Table 5.1 Results for ***LR1*** under different settings

			***LR1* (*a* /1.005)**				***LR1*(*a* /2, 30)**				***LR1* (*a* /2, 10)**			
	%IGAP	**CPU**	**%LO**	**%UO**	**%UL**	**CPU**	**%LO**	**%UO**	**%UL**	**CPU**	**%LO**	**%UO**	**%UL**	**CPU**
pre1	46.57	1263	22.95	2.11	32.53	41.28	18.91	2.11	25.92	63.84	21.86	2.11	30.67	86.18
pre2	47.36	78	21.77	6.08	35.59	59.98	17.78	2.82	25.06	67	21.76	6.08	35.58	51.14
pre19	45.33	973	22.57	0.00	29.14	74.86	19.79	0.00	24.67	81.52	21.31	0.00	27.08	83.4
pre28	60.91	1838	29.25	0.03	41.38	64.63	27.13	0.03	37.28	58.81	33.20	0.03	49.75	46.66
pre31	57.64	1084	35.95	3.06	60.91	22.15	30.50	0.79	45.02	28.05	32.01	3.06	51.57	41.94
pre32	63.16	63	30.03	0.00	42.93	169.72	25.19	0.00	33.68	175.84	32.59	1.39	50.41	129.99
Average	**53.49**	**883**	**27.09**	**1.88**	**40.41**	**72.10**	**23.22**	**0.96**	**31.94**	**79.18**	**27.12**	**2.11**	**40.84**	**73.22**
pre3	40.86	23	14.36	0.00	16.76	103.53	14.08	0.00	16.38	112.7	16.32	0.00	19.51	73.62
pre8	48.71	468	21.44	0.00	27.29	77.23	18.77	0.00	23.10	72.29	21.37	0.00	27.18	72.03
pre15	46.41	1421	27.21	0.67	38.30	29.21	25.67	0.67	35.43	35.88	26.42	0.67	36.81	30.05
pre24	54.80	159	35.61	0.00	55.31	21.46	28.36	0.00	39.58	29.6	29.53	0.00	41.90	22.37
pre36	60.39	74550	37.17	1.19	61.04	32.27	35.16	0.00	54.22	38.48	34.30	0.00	52.22	28.41
pre37	53.51	357	33.11	4.15	55.71	22.49	32.42	3.89	53.73	20.19	32.14	3.89	53.08	22.46
Average	**50.78**	**12830**	**28.15**	**1.00**	**42.40**	**47.70**	**25.74**	**0.76**	**37.07**	**51.52**	**26.68**	**0.76**	**38.45**	**41.49**
npre3	38.40	13	10.87	0.00	12.20	64.65	10.89	0.00	12.22	66.54	12.50	0.00	14.28	36.65
npre8	47.54	680	18.99	0.00	23.44	66.59	17.22	0.00	20.80	58.40	19.41	0.00	24.09	49.56
npre15	43.84	22889	27.52	0.72	38.96	24.36	23.26	0.72	31.25	29.14	24.87	0.72	34.06	30.19
npre24	55.00	625	32.98	0.00	49.22	22.22	27.99	0.00	38.88	21.33	29.16	0.00	41.16	23.07
npre36	59.70	14753	38.14	0.00	61.66	29.72	37.10	0.00	58.98	34.84	33.73	0.00	50.89	28.28
npre37	51.97	342	30.33	4.35	49.78	20.11	30.15	4.08	49.00	21.57	31.15	4.35	51.56	21.80
Average	**49.41**	**6550**	**26.47**	**0.84**	**39.21**	**37.94**	**24.44**	**0.80**	**35.19**	**38.64**	**25.14**	**0.84**	**36.01**	**31.59**
Overall	**51.23**	**6754**	**27.24**	**1.24**	**40.68**	**52.58**	**24.47**	**0.84**	**34.73**	**56.45**	**26.31**	**1.24**	**38.43**	**48.77**

From Figure 5.2, it can be said that one iteration of *LR1* takes approximately 110 seconds and for these test instances, the algorithm can halt after 40 iterations.

As a result, we decided to apply *LR2* algorithm in which we solve LP relaxation of *DIST* subproblem to obtain a lower bound on that subproblem and all the other subproblems to optimality. In the following section, we present *LR2* algorithm.

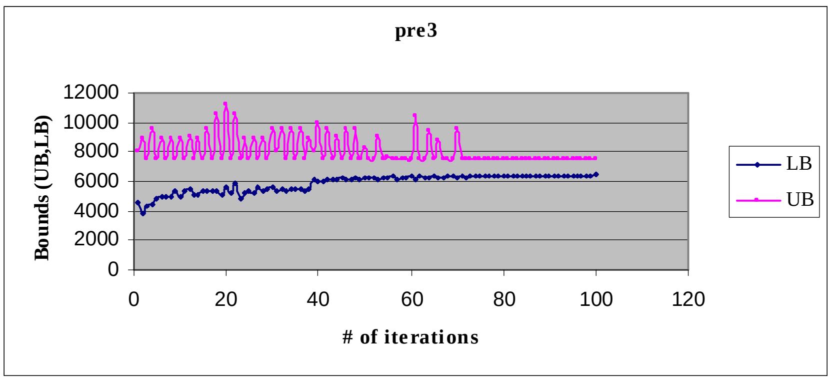

Figure 5.1 Bounds versus number of iterations graph for *LR1*(*a*/2, 30)

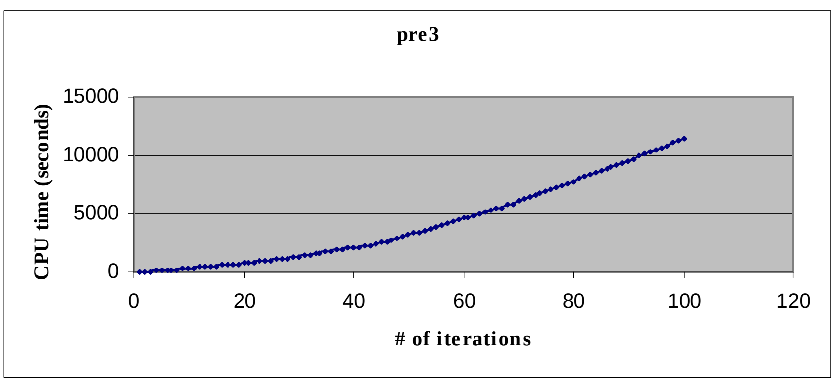

Figure 5.2 CPU time versus number of iterations graph for *LR1*(*a*/2, 30)

5.2.2 Phase 2

LR2 algorithm is also run for 100 iterations for the same reasons stated above. The same settings are tried for subgradient optimization constant ***a***: ***a***/1.005; ***a***/2, 30; ***a***/2, 10.

The results for these settings and total CPU times in seconds are provided in Table 5.2. Based on the results, we can easily say that halving scalar after 10 consecutive non-improving iterations gives the best results in all of the measures, which is a different result than ***LR1*** algorithm. In addition, this time CPU times are reasonably small. On the other hand, results in all of the measures deteriorated compared to ***LR1*** as expected. For some instances, like pre24 and npre24 percent deviations of the upper bound from optimum are relatively high, %17.28 and %16.5, respectively, for the case (***a***/2, 10). Nonetheless, upper bounds obtained by ***LR2***(***a***/2, 10) are still good; about %4 deviation from optimum on the average. We plot upper and lower bounds obtained in each iteration versus number of iterations for our instances to investigate convergence of *LR2*. Here, we only provide one typical figure (Figure 5.3) obtained on an instance (pre3). The figures for other instances can be seen in Appendix C. Figure 5.3 indicates that much less than 100 iterations are sufficient for the algorithm to converge.

These experiments indicate that there is a strong tradeoff between better bounds (upper or lower) and the effort to find out them. Keeping in mind this result, we consider how to improve both ***LR1*** and ***LR2*** to obtain better bounds in the next section.

Table 5.2 Results for *LR2* under different settings

			LR2 (*a* /1.005)				*LR2*(*a* /2, 30)				*LR2* (*a* /2, 10)			
	%IGAP	CPU	%LO	%UO	%UL	CPU	%LO	%UO	%UL	CPU	%LO	%UO	%UL	CPU
pre1	46.57	1263	42.37	2.11	77.18	76	38.66	2.11	66.48	81	37.11	2.11	62.35	72
pre2	47.36	78	44.07	7.80	92.72	68	40.62	7.80	81.54	67	38.49	7.80	75.24	72
pre19	45.33	973	42.57	0.00	74.12	76	40.59	0.00	68.32	80	37.33	0.00	59.57	80
pre28	60.91	1838	64.84	1.43	188.48	74	61.72	0.03	161.34	76	56.99	0.03	132.55	74
pre31	57.64	1084	58.86	7.60	161.51	71	54.64	3.65	128.51	78	52.45	3.65	117.97	69
pre32	63.16	63	68.02	12.04	250.37	70	61.65	12.01	192.10	67	58.97	12.01	172.99	77
Average	**53.49**	**883**	**53.45**	**5.16**	**140.73**	**73**	**49.65**	**4.27**	**116.38**	**75**	**46.89**	**4.27**	**103.45**	**74**
pre3	40.86	23	35.00	0.00	53.84	36	33.04	0.00	49.35	30	30.17	0.00	43.21	39
pre8	48.71	468	46.46	0.00	86.78	74	42.09	0.00	72.68	64	40.54	0.00	68.17	70
pre15	46.41	1421	43.99	0.67	79.75	53	40.72	0.67	69.82	47	39.11	0.67	65.33	52
pre24	54.80	159	65.52	19.78	247.40	72	56.99	19.06	176.80	76	50.38	17.28	136.34	75
pre36	60.39	74550	57.96	1.19	140.70	69	56.40	1.19	132.08	65	54.14	1.19	120.66	74
pre37	53.51	357	50.82	4.81	113.11	53	48.23	4.81	102.46	48	46.95	4.81	97.59	53
Average	**50.78**	**12830**	**49.96**	**4.41**	**120.26**	**60**	**46.24**	**4.29**	**100.53**	**55**	**43.55**	**3.99**	**88.55**	**61**
npre3	38.40	13	31.01	0.00	44.95	63	27.30	0.00	37.54	53	26.82	0.00	36.64	47
npre8	47.54	680	42.89	0.99	76.83	125	40.17	0.00	67.14	121	38.82	1.53	65.94	102
npre15	43.84	22889	40.00	0.72	67.87	103	37.95	0.72	62.32	86	36.02	0.72	57.43	73
npre24	55.00	625	64.70	17.58	233.06	145	50.47	16.50	135.22	123	50.47	16.50	135.22	119
npre36	59.70	14753	60.04	1.24	153.36	126	55.47	1.24	127.34	125	53.20	1.24	116.32	104
npre37	51.97	342	49.77	4.56	108.16	97	46.26	4.57	94.58	95	45.09	4.57	90.46	81
Average	**49.41**	**6550**	**48.07**	**4.18**	**114.04**	**110**	**42.94**	**3.84**	**87.36**	**101**	**41.74**	**4.09**	**83.67**	**88**
Overall	**51.23**	**6754**	**50.49**	**4.58**	**125.01**	**81**	**46.28**	**4.13**	**101.42**	**77**	**44.06**	**4.12**	**91.89**	**74**

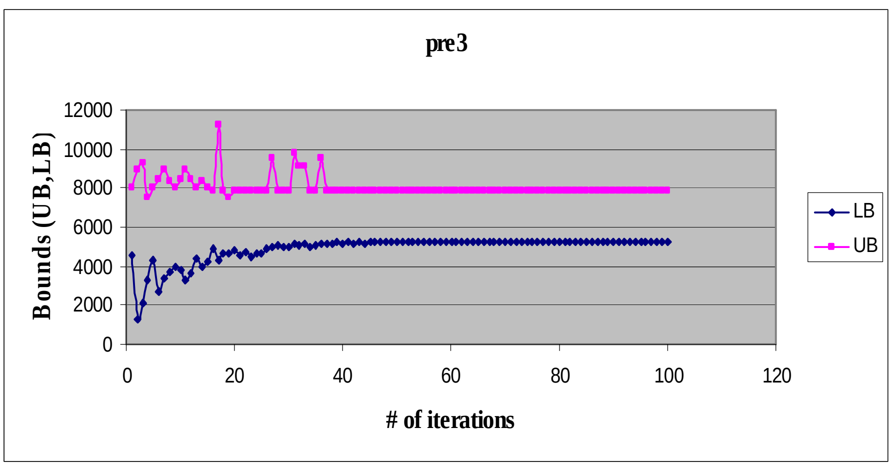

Figure 5.3 Bounds versus number of iterations graph for *LR2*(*a*/2, 10)

5.2.3 Phase 3

In this section, we shall first describe our enhanced LR based solution approach in detail. Our enhancement is based on the fact that there is no enforcement for the vehicle to depart from the supplier in the *DIST* subproblem. Consequently, we construct an enumeration tree, at which we force the vehicle either to depart or stay at the supplier in each period to obtain stronger results. In other words, we disseminate two arcs from each node of the tree corresponding to departure and stay of the vehicle in a given period, respectively. Disseminating two arcs from each node leads to an enumeration tree with at most $2^{T+1}-1$ many nodes. To force the vehicle either to depart or stay at the supplier in each period, we make the constraint related to vehicle departure in that period in *DIST* subproblem equal to either one or zero, respectively. If we let the vehicle to stay at the supplier in a given period, we do not allow replenishment of any retailer in the *RETAILER* subproblem in that period, which may cause stockout at some of the retailers and thus, pruning of some of the nodes in the tree due to infeasibility. Moreover, by running the *LR* algorithm on a given node of the tree, one obtains both an upper bound and a lower bound, which can then be used to fathom other nodes if the lower bounds on the nodes are higher than the best known upper bound like in the well-known Branch-and Bound technique, which is widely used for discrete optimization problems. This approach, however, does not work very well here since the LR hardly generates tight lower bounds. For this reason, we let the algorithm only work on each of the feasible nodes of the last level of the tree and take the smallest upper and lower bounds generated on these nodes.

The enumeration tree for an instance with time horizon of three periods is given in Figure 5.4. In this figure, two nodes are designated as infeasible. For instance, the node designated as infeasible in the second level as well as nodes originated from that node are fathomed because at least one retailer stocks out if no replenishment occurs in the first two periods in the instance concerned. As a result, our LR based approach is only implemented on five feasible nodes (leaves) of the last level of the example tree.

As already mentioned, we implement *LR1* and *LR2* algorithms on the enumeration tree, what we refer to as *ELR1* and *ELR2*, respectively. We let the *ELR* algorithms to perform five iterations on each feasible node of the last level of the tree. We make five iterations because more would cause excessively high computation times, particularly in large instances. Also, we do not change the value of subgradient optimization constant.

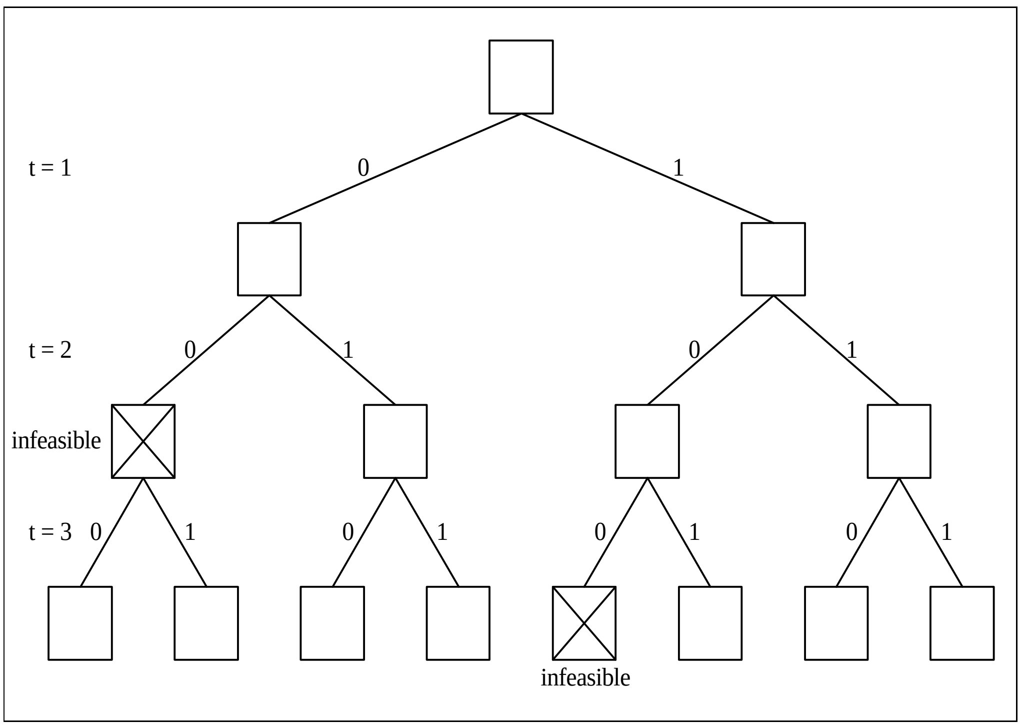

Figure 5.4 An example enumeration tree for an instance with $T = 3$

The results as well as CPU times in seconds for *ELR1* and *ELR2* algorithms are given in Table 5.3. Results indicate that our enhanced versions, *ELR1* and *ELR2*, generates better upper bounds (shown by %UO column) than their standard versions, ***LR1*** and ***LR2***, on average. It seems that enhanced versions improved upper bounds more than they did for the lower bounds. In other words, ***LR1*** is still better than ***ELR1*** with regard to %LO and %UL measures while ***ELR2*** does better than ***LR2*** in all of measures considered. One reason for ***ELR*** algorithms to improve more upper bounds than lower bounds could be related with the few iterations performed in each node. Table 5.3 also shows that performance measures are almost the same for ***ELR1*** and ***ELR2***; i.e., ***ELR1*** and ***ELR2*** are almost indifferent for these 18 test instances. Therefore, we are going to use only ***ELR2*** in large test instances, which is the faster one.

Table 5.3 Results for ***ELR*** algorithms under different settings

			ELR1				***ELR2***			
	%IGAP	**CPU**	**%LO**	**%UO**	**%UL**	**CPU**	**%LO**	**%UO**	**%UL**	**CPU**
pre1	46.57	1263	25.31	0.95	35.15	550	27.71	0.95	39.65	65
pre2	47.36	78	30.83	0.00	44.57	737	30.83	0.00	44.57	64
pre19	45.33	973	28.96	0.00	40.77	1158	29.66	0.00	42.17	67
pre28	60.91	1838	50.26	0.03	101.12	666	50.34	0.03	101.41	65
pre31	57.64	1084	46.48	0.79	88.31	668	47.34	0.79	91.40	66
pre32	63.16	63	54.55	0.00	120.03	1694	59.23	0.00	145.31	66
Average	**53.49**	**883**	**39.40**	**0.29**	**71.66**	**912**	**40.85**	**0.29**	**77.42**	**66**
pre3	40.86	23	15.47	0.00	18.31	624	21.53	0.00	27.44	45
pre8	48.71	468	31.94	0.00	46.94	689	33.94	0.00	51.37	63
pre15	46.41	1421	23.04	0.32	30.36	899	23.04	0.32	30.36	55
pre24	54.80	159	55.82	0.00	126.34	827	55.82	0.00	126.34	68
pre36	60.39	74550	42.82	0.47	75.69	604	42.82	0.47	75.69	65
pre37	53.51	357	33.51	4.35	56.94	480	33.51	4.81	57.64	54
Average	**50.78**	**12830**	**33.77**	**0.86**	**59.10**	**687**	**35.11**	**0.93**	**61.47**	**58**
npre3	38.40	13	16.77	0.00	20.15	655	23.34	0.00	30.44	47
npre8	47.54	680	34.09	0.00	51.73	708	36.22	0.00	56.80	64
npre15	43.84	22889	24.69	0.72	33.73	924	24.69	0.72	33.73	55
npre24	55.00	625	58.35	0.00	140.08	836	58.35	0.00	140.08	67
npre36	59.70	14753	43.22	1.24	78.29	611	43.22	1.24	78.29	63
npre37	51.97	342	35.14	4.56	61.22	482	35.14	4.57	61.24	53
Average	**49.41**	**6550**	**35.38**	**1.09**	**64.20**	**703**	**36.83**	**1.09**	**66.76**	**58**
Overall	**51.23**	**6754**	**36.18**	**0.75**	**64.99**	**767**	**37.60**	**0.77**	**68.55**	**61**

Lastly, in this section, we present results regarding ***Every*** and ***Latest*** heuristics with respect to ***LR1***, ***LR2***, ***ELR1*** and ***ELR2*** under their best settings in Table 5.4. We do not need these heuristics' results in that stage since we know the optimal solutions of the instances for benchmarking of our LR based approaches. However, we will need these heuristics when experimenting with large instances and it will be more clear why we implemented these heuristics to the preliminary instances when we come to experiments with large instances. Table 5.4 shows that *Every* heuristic performs very poorly compared to all LR based approaches, being over %100 distant from the best upper bound obtained by LR based approaches, on average. *Latest* heuristic, on the other hand, generates results far better than *Every*. However, again our LR based procedures perform significantly better than *Latest*, on average. We did not report CPU times for *Every* and *Latest* since CPU times for both heuristics are well under one second for all the instances considered.

Table 5.4 Results for *Every* and *Latest* heuristics with respect to our LR algorithms

	%Every				%Latest			
	***LR1* (*a* /2, 30)**	***LR2* (*a* /2, 10)**	***ELR1***	***ELR2***	***LR1* (*a* /2, 30)**	***LR2* (*a* /2, 10)**	***ELR1***	***ELR2***
pre1	59.75	59.75	61.60	61.60	9.59	9.59	10.85	10.85
pre2	60.42	53.02	64.95	64.95	5.24	0.39	8.22	8.22
pre19	61.95	61.95	61.95	61.95	11.90	11.90	11.90	11.90
pre28	189.46	189.46	189.46	189.46	12.83	12.83	12.83	12.83
pre31	129.36	123.04	129.36	129.36	20.94	17.61	20.94	20.94
pre32	138.94	113.33	138.94	138.94	19.14	6.37	19.14	19.14
Average	**106.65**	**100.09**	**107.71**	**107.71**	**13.27**	**9.78**	**13.98**	**13.98**
pre3	102.50	102.50	102.50	102.50	7.36	7.36	7.36	7.36
pre8	85.50	85.50	85.50	85.50	7.30	7.30	7.30	7.30
pre15	69.49	69.49	70.09	69.49	12.69	12.69	13.09	13.09
pre24	147.49	111.02	147.49	147.49	30.99	11.69	30.99	30.99
pre36	171.88	168.68	170.61	170.61	22.14	20.70	21.57	21.57
pre37	172.29	169.89	171.09	169.89	15.31	14.29	14.80	14.29
Average	**124.86**	**117.85**	**124.55**	**124.25**	**15.96**	**12.34**	**15.85**	**15.77**
npre3	119.50	119.50	119.50	119.50	5.18	5.18	5.18	5.18
npre8	97.99	95.00	97.99	97.99	5.54	3.95	5.54	5.54
npre15	81.49	81.49	81.49	81.49	11.23	11.23	11.23	11.23
npre24	153.40	117.52	153.40	153.40	29.34	11.02	29.34	29.34
npre36	183.34	179.87	179.87	179.87	21.67	20.18	20.18	20.18
npre37	185.04	183.68	183.72	183.68	14.47	13.92	13.94	13.92
Average	**136.79**	**129.51**	**135.99**	**135.99**	**14.57**	**10.91**	**14.23**	**14.23**
Overall	**122.77**	**115.82**	**122.75**	**122.65**	**14.60**	**11.01**	**14.69**	**14.66**

5.3 Further experiments

This section involves the last phase of our experimentation described at the beginning of the chapter. In this section, we present the results obtained by solving large test instances with *N*=50 and *T*=30. We first give the experimental settings used, and then the computational results.

5.3.1 Experimental settings

In this part, we use 80 test instances as described in section 5.1. However, for the instances at hand, the following data are added.

- Fixed order cost (f_{0t}): $\sum_{i \in N} h_0 d_i$
- Fixed vehicle dispatching cost (f_t)
    - $0.8 * \sum_{i \in N} h_i d_i$
    - $0.2 * \sum_{i \in N} h_i d_i$

The instances with high fixed vehicle dispatching cost (f_t) are named with a prefix "H" (e.g. H1, H2, etc.) while those with low fixed vehicle dispatching cost (f_t) are named with a prefix "L" (e.g. L1, L2, etc.).

Since we know that ***LR1*** algorithm is not a viable option in terms of computation time for such large instances (50 retailers and 30 periods), we only use ***LR2*** algorithm besides ***ELR2*** algorithm. Remember that, in the preliminary experiments section, we have fixed total number of iterations to 100 and halve subgradient optimization constant after 10 consecutive non-improving iterations for ***LR2*** algorithm. Now, we select 50 iterations as a termination criterion for ***LR2*** because we have seen that the convergence of the algorithm to a local solution does not take much iterations in our preliminary experiments. Furthermore, since the enumeration tree gets intractably larger as the number of periods increases, we enumerate only the first six periods of the planning horizon for the enhanced version of ***LR2*** (***ELR2***). We let the ***ELR2*** to perform five iterations on each feasible node of the last level of the tree. We also tried another version of ***ELR2*** algorithm, in which all possible combinations of five periods selected with almost equally distant from each other (i.e., periods 1, 8, 15, 22 and 30) are enumerated. However, we did not report them since the results (CPU times) are considerably worse than (longer than) those of ***ELR2*** algorithm with enumerating only the first six periods.

5.3.1 Experimental results

In this section, we present the run results as to the above described algorithms and settings. In the following, we give the set of extra measures that we use for determining the performance of algorithms.

The measures defined in section 5.2 are used in this part, as well. However, we consider the following measure in addition to those defined in section 5.2.

§ **%LLPR:** The percentage gap between the best lower bound (LB) found by the LR based algorithm and the LP relaxation value (LPR) of IRO, i.e. %(LB-LPR)/LB.

Computational results for instances with high f_t and low f_t are averaged over 10 instances with respect to each combination of parameters (c_{ij}, h_i, h_0) and presented in Tables 5.5 and 5.6, respectively. We did not report any result regarding the feasible solution of IRO with CPLEX in the tables because CPLEX could not achieve to find even a feasible solution for any of the instances at hand within the two hour time limit. This also shows the difficulty of the problem we are dealing with.

Table 5.5 Results for *LR2* and *ELR2* on large instances with high f_t

						LR2					*ELR2*				
c_{ij}	h_i	h_0	CPU LPR	CPU *Latest*	CPU *Every*	%LLPR	%UL	%Latest	%Every	CPU *LR2*	%LLPR	%UL	%Latest	%Every	CPU *ELR2*
[0, 500]	[0.6, 1]	0.3	633.57	9.18	0.70	28.75	19.65	0.22	107.27	2791.84	29.17	18.46	0.64	108.14	6130.95
		0.8	641.40	9.09	0.72	27.87	24.67	0.42	106.09	3072.08	28.34	23.27	0.90	107.07	6823.08
	[0.1, 0.5]	0.3	658.02	9.10	0.76	27.92	40.20	-0.39	333.10	2777.85	28.32	38.38	0.38	336.30	5949.78
		0.8	638.02	9.08	0.72	27.23	52.75	0.63	311.29	3071.11	27.70	51.12	1.07	313.08	6790.81
[0, 1000]	[0.6, 1]	0.3	625.48	9.15	0.72	28.55	28.11	0.04	109.72	2642.84	28.94	26.79	0.52	110.71	5655.50
		0.8	642.29	9.14	0.71	27.72	32.90	0.31	108.68	2721.39	28.12	31.26	1.01	110.11	6027.21
	[0.1, 0.5]	0.3	649.78	9.14	0.73	27.38	62.79	-0.48	306.96	2630.11	27.71	60.94	0.23	309.85	5508.48
		0.8	640.45	9.01	0.69	26.80	75.44	0.06	288.68	2814.76	27.09	73.78	0.65	291.00	5964.81
Average			641.12	9.11	0.72	27.78	42.06	0.10	208.97	2815.25	28.17	40.50	0.68	210.78	6106.33

Table 5.6 Results for *LR2* and ELR2 on large instances with low f_t

						LR2					*ELR2*				
c_{ij}	h_i	h_0	CPU LPR	CPU *Latest*	CPU *Every*	%LLPR	%UL	%Latest	%Every	CPU *LR2*	%LLPR	%UL	%Latest	%Every	CPU *ELR2*
[0, 500]	[0.6, 1]	0.3	634.39	8.77	1.00	29.04	13.38	0.19	111.52	2640.86	29.16	12.86	0.48	112.12	5736.12
		0.8	655.40	9.12	0.72	28.19	18.03	0.69	110.67	2950.36	28.33	17.63	0.85	111.01	6590.28
	[0.1, 0.5]	0.3	646.42	9.15	0.69	28.22	33.47	-0.07	349.94	2688.62	28.30	32.85	0.29	351.60	5755.11
		0.8	651.66	9.11	0.73	27.53	46.59	0.64	324.08	3057.58	27.68	45.69	1.07	325.96	6686.59
[0, 1000]	[0.6, 1]	0.3	626.64	9.05	0.69	28.85	21.86	0.00	113.77	2528.78	28.93	21.19	0.44	114.71	5390.07
		0.8	649.64	9.11	0.70	28.02	26.74	0.25	112.44	2711.41	28.10	26.06	0.68	113.35	5850.49
	[0.1, 0.5]	0.3	657.45	9.04	0.71	27.69	56.56	-0.48	318.98	2573.24	27.69	56.04	-0.12	320.58	5436.45
		0.8	644.79	9.05	0.73	26.97	69.33	0.28	299.78	2757.47	27.07	68.45	0.70	301.43	5937.28
Average			645.80	9.05	0.75	28.06	35.75	0.19	217.65	2738.54	28.16	35.09	0.55	218.85	5922.80

When we look at the gap between the best upper and lower bounds in Table 5.5, we see that on the average, gap is %42.06 for the *LR2* algorithm, while it is slightly improved to %40.5 by the *ELR2* algorithm. These are not tight bounds and does not give much idea on the quality of the best feasible solution (upper bound) generated by the algorithms. Nevertheless, lower bounds generated by the *LR2* and *ELR2* are still far better than the LP relaxation of the IRO (approximately %28), as stated by %LLPR columns. Almost all the results with low f_t (Table 5.6) are slightly better than those with high f_t (Table 5.5) but similar comments are still valid.

Our main inference from the preliminary experiments is that our LR based algorithms are successful in finding good upper bounds (feasible solutions), whereas lower bounds generated are not tight enough. To justify this inference on large instances, we will use the heuristics *Latest* and *Every*.

Table 5.5 and 5.6 show that *LR2* algorithm is far superior to the *Every* heuristic, ranging from %106 to %333 for instances with high f_t and from %111 to %350 for instances with low f_t. *ELR2* algorithm achieves better results than *LR2* as expected. On the other hand, the results generated by the *Latest* heuristic are very close to those obtained by using *LR2* and *ELR2* algorithms. *ELR2* algorithm produces better results than *Latest* in almost all the data combinations except one in instances with low f_t (%-0.12) yet the results are again very close to those obtained by the *Latest* (On average, %0.68 for instances with high f_t and %0.55 for instances with low f_t). These results contradict with the results of small instances, where *LR2* and *ELR2* significantly surpass *Latest*. To shed light on this contradiction, we should investigate the features of the *Latest* heuristic and instances at hand.

Latest tends to minimize inventory holding cost at retailers by replenishing them at the last possible moment. As stated by Bertazzi et al. (2002), an optimal policy in *Latest* tends to replenish each retailer just before the inventory level becomes negative (stockout occurs). The only reason to replenish a retailer earlier than last possible moment may be the vehicle capacity or the limited amount of product available at the supplier. In our problem settings, we treat vehicle capacity as uncapacitated and have the opportunity to determine order amounts as well as their timings at the supplier. Furthermore, we solve routing of retailers to be replenished in each period (T many TSPs) as well as order decisions at supplier (an uncapacitated lot-sizing problem) to optimality. Hence, we expect *Latest* to perform well in our problem. However, we know that our algorithms are far superior to the *Latest* on smaller test instances in preliminary experiments while performances are almost equivalent in the instances we considered in this section. We explain this situation by investigating the weights of the cost components on the total cost of the problem instances.

To find percentages of the cost components on total cost, we consider the results of the 12 small instances used in preliminary experiments and their corresponding larger instances, from which they are derived. The results are shown in Table 5.7. In this table, %Supp, %Trans and %Ret denote average percentages of fixed order and inventory holding costs at the supplier, transportation cost, and inventory holding cost at the retailers on total cost of the best (upper bound) feasible solution found, respectively.

Table 5.7 shows that inventory holding cost at retailers predominates other cost terms in large instances, while transportation cost has the largest weight and is slightly higher than the inventory holding cost at retailers in small instances. In both small and large instances, supplier's cost is the lowest cost term, thus insignificant with respect to other costs. For the large instances, percentage of retailers' cost on total cost is about %75 on average, which means that minimizing that cost approximately minimizes the total cost. Thus, considering that *Latest* tends to minimize inventory holding cost at retailers, we can claim that *Latest* finds good quality solutions in large instances. In this respect, figures of the Table 5.7 account for the high %Late values for small test instances and low %Late values for large test instances. Our algorithms surpass *Latest* on small instances where %Trans is comparable to %Ret, whereas they and *Latest* perform similarly on large instances in which %Ret far outweighs other cost terms. For instances with high %Trans (at least greater than %Ret), our algorithm finds more opportunities to consolidate replenishments to retailers through cost efficient routes that compensate the higher holding cost at retailers compared to latest replenishment policy. This leads to far better performance for our algorithm with respect to *Latest*. Note that almost whenever %Ret increases, %Late decreases in a given instance.

Considering that *Latest* might find optimal solution under special conditions (uncapacitated vehicle and unlimited inventory at the supplier as well as high percentage of retailers' cost on total cost), we can deduct that *Latest* performs very good in the selected large instances. Consequently, our algorithms *LR2* and *ELR2* have also good performance since they find slightly better results than Latest does, on average.

Table 5.7 Percentages of different cost components on total cost

	Small Instances				Large Instances			
	%Supp	%Trans	%Ret	%Latest	%Supp	%Trans	%Ret	%Latest
pre3-H3	4.58	35.83	59.59	7.36	2.69	15.50	81.81	0.24
pre8-H8	3.36	45.52	51.12	7.30	2.78	15.12	82.10	0.21
pre15-H15	6.45	41.02	52.53	12.69	7.01	17.35	75.64	0.32
pre24-H24	4.40	67.81	27.80	11.69	7.02	24.12	68.86	-0.17
pre36-H36	11.41	56.61	31.97	20.70	15.40	21.24	63.36	0.68
pre37-H37	10.26	49.89	39.85	14.29	12.29	21.65	66.06	-0.40
Average	**6.74**	**49.45**	**43.81**	**12.34**	**7.86**	**19.16**	**72.97**	**0.15**
npre3-L3	4.96	30.45	64.59	5.18	2.86	9.93	87.20	0.13
npre8-L8	5.04	41.22	53.74	3.95	2.95	10.09	86.96	0.05
npre15-L15	6.91	36.85	56.25	11.23	7.56	11.26	81.18	0.00
npre24-L24	4.53	66.82	28.65	11.02	7.15	20.13	72.71	0.32
npre36-L36	11.89	54.81	33.30	20.18	16.00	17.73	66.28	0.82
npre37-L37	11.15	47.09	41.77	13.92	13.11	17.22	69.67	-0.91
Average	**7.41**	**46.20**	**46.38**	**10.91**	**8.27**	**14.40**	**77.33**	**0.07**
Overall	**7.08**	**47.83**	**45.10**	**11.63**	**8.07**	**16.78**	**75.15**	**0.11**

Tables 5.5 and 5.6 also show the sensitivity of solutions generated by *LR2* and *ELR2* with respect to different parameters (unit inventory holding costs at supplier

(h_0) and retailers (h_i), distance based transportation cost (c_{ij})). For instance, we can easily state that both *LR2* and *ELR2* are more successful than *Latest* in instances, where unit holding cost at supplier is greater than unit holding cost at retailers. This conclusion is valid for instances with both high and low dispatching cost.

To examine the impact of different cost parameters on LR algorithms, we compare the percent gap between best UB and UL (%UL columns) in Tables 5.5-5.6. For the *LR2* algorithm on instances with high dispatching cost, increasing one of the cost parameters (except h_i) and keeping the rest constant lead to increase in the %UL values. The best %UL has been achieved when h_0 and c_{ij} are at their lowest value and h_i at its highest value. In contrast, the worst %UL has been obtained when h_0 and c_{ij} are at their highest value and h_i at its lowest value. The same interpretations are valid for *ELR2*. Furthermore, when dispatching cost decreases (low f_t value), %UL values improve and all the comments regarding sensitivity of solutions to unit holding cost at supplier and retailers as well as distance based transportation cost are the same.

As far as the CPU times are concerned, one can observe from Tables 5.5-5.6 that CPU time needed to solve LP relaxation of the IRO is over ten minutes, which shows the difficulty of solving the IRO to optimality. As expected, CPU times needed by *Latest* and *Every* are very low. Since *Latest* finds good quality solutions in short time on those instances, it can be a good alternative to solve IRO whenever holding cost at retailers is a predominating component of the total cost. On the other hand, our LR algorithms finds good quality solutions in reasonable times considering the length of horizon. In addition, LR algorithms also produce lower bounds in that time. *ELR2* runs about two times *LR2* does. Furthermore, in *ELR2*, one can obtain better results by enumerating more periods or increasing the iteration number on each node, which require compromises from CPU time.

CHAPTER 6

CONCLUSION AND FURTHER STUDIES

In this study, we addressed the integrated inventory management and vehicle routing problem with deterministic order-up-to level policy (IRO). We have proposed a flexible mathematical formulation for the IRO, which can handle different kinds of cost structures.

For the efficient solution of the mathematical formulation proposed, we have developed a Lagrangean relaxation (LR) based solution approach, which disaggregates the problem into N single retailer replenishment problems, VT distribution planning problems and simple supplier's order and inventory problems. At the core of the solution approach is the single retailer replenishment problem. A polynomial time algorithm with worst case complexity $O(T^2)$ has been developed for the single retailer replenishment problem. Another challenging problem that arises after relaxation is the distribution problem (*DIST*). Since *DIST* is a complex problem, its solution to optimality is not possible in a reasonable time for large instances by using general purpose solvers. Therefore, we have developed valid inequalities to that problem to make its LP relaxation solution tighter and obtain a good approximation.

Furthermore, we have investigated a special case of the general problem consisting of multiple retailers. As a special case, single supplier-single retailer problem has been analyzed and a polynomial time algorithm with worst case complexity $O(T^3)$ has been developed for the special case problem.

We have implemented several variants of the LR based solution approach on small and large test instances. On the small instances, we have determined the proper settings for LR variants to achieve the best implementation. In addition, we have proposed enhanced versions of the LR based approach that improves solutions generated.

Results on relatively small instances revealed that our LR based solution procedures produce high quality feasible solutions (upper bounds) whereas lower bounds generated are not tight enough.

As for large instances, we could not even find a feasible solution to the instances using CPLEX within hours. From the literature, we have adapted two heuristics, *Every* and *Latest*, to our context and compared our approach with them. According to our computation results, our LR based solution procedures are significantly superior to *Every*, whereas *Latest* generates solutions very close to ours in some instances but not in all instances. We have shown that solution quality of *Latest* depends on the relative weight of inventory holding cost at retailers on total cost. That is, whenever the weight is high, it is expected from *Latest* to produce high quality solutions. In a set of instances, we have shown that if the weight of holding cost at retailers on total cost is not high, our algorithms generate considerably better solutions than *Latest*.

Our computational study was limited to instances with a single vehicle although our problem as well as model formulation are general and involve multiple vehicles. Although testing our LR based procedures on cases with a fleet of vehicles could be nice, multiple vehicles might create difficulties for enhanced LR procedure as the enumeration tree will get enormously large.

Mathematical formulation proposed is flexible in handling several cost structures. It may be interesting to generate test instances for different cost structures such as transportation cost proportional to amount shipped rather than distance based transportation cost and test our solution procedures on these instances in a future study.

In this study, we have considered that the supplier can decide on its order policy. It is more realistic to have production planning issues with capacity limitations on production at the lower echelon. Another research issue may be to study our problem with plant(s) instead of a supplier. In other words, production planning decisions at the plant(s) should be incorporated into the model.

Another possible research issue is to extend our work to involve multiple items. Indeed, our model can easily be extended to account for multiple items if replenishment of each item can be made independent of other items.

With regard to mathematical formulation, one can find many different ways to formulate the problem leading to different solution approaches. For instance, there are alternative ways for ensuring route integrity (eliminating subtours), which have been widely studied in vehicle routing problem (VRP) and traveling salesman problem (TSP) literature. Adaptation of such formulations appeared in VRP and TSP literature may bring about interesting formulations and necessitate development of specific solution approaches.

In the past few years, polyhedral approaches have been successfully implemented for several hard optimization problems. Thus, a possible research direction may be to study the polyhedra of the problem.

REFERENCES

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

Anily, S., and A. Federgruen. 1993. Two-echelon distribution systems with vehicle routing costs and central inventories. *Operations Research* **41** 37-47.

Anily, S. 1994. The general multi-retailer EOQ problem with vehicle routing costs. *European Journal of Operational Research* **79** 451-473.

Baita, F., W. Ukovich, R. Pesenti, and D. Favaretto. 1998. Dynamic routing-and-inventory problems: A review. *Transportation Research A* **32** (8) 585-598.

Bell, W.J., L. Dalberto, M. Fisher, A. Greeenfield, R. Jaikumar, P. Kedia, R. Mack, and P. Prutzman. 1983. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces* **13** 4-23.

Bertazzi, L., G. Paletta, and M.G. Speranza. 2002. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science* **36** (1) 119-132.

Blumenfeld, D.E., L.D. Burns, J.D. Diltz, and C.F. Daganzo. 1985. Analyzing trade-offs between transportation, inventory and production costs on freight networks. *Transportation Research* **19B** (5) 361-380.

Blumenfeld, D.E., L.D. Burns, C.F. Daganzo, M.C. Frick, and R.W. Hall. 1987. Reducing logistics costs at General Motors. *Interfaces* **17** (1) 26-47.

Burns, L.D., R.W. Hall, D.E. Blumenfeld, and C.F. Daganzo. 1985. Distribution strategies that minimize transportation and inventory costs. *Operations Research* **33** (3) 469-490.

Campbell, A., L. Clarke, A. Kleywegt, and M.W.P. Savelsbergh. 1998. The inventory routing problem, T.G.Crainic, G.Laporte, eds. *Fleet Management and Logistics*. Kluwer Academic Publishers, London, UK. 95-113.

Chan, L.M.A., A. Federgruen, and D. Simchi-Levi. 1998. Probabilistic analysis and practical algorithms for inventory-routing models. *Operations Research* **46** (1) 96-106.

Chandra, P. 1993. A dynamic distribution model with warehouse and customer replenishment requirements. *Journal of Operations Research Society* **44** (7) 681-692.

Chandra, P., and M.L. Fisher. 1994. Coordination of production and distribution planning. *European Journal of Operational Research* **72** (3) 503-517.

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

Dror, M., and M. Ball. 1987. Inventory / Routing: Reduction from an annual to a short-period problem, *Naval Research Logistics* **34** 891-905.

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

Federgruen, A., and D. Simchi-Levi. 1995. Analysis of vehicle routing and inventory-routing problems. M.O. Ball et al., eds., *Handbooks in OR &MS*, vol. 8, Chapter 4.

Fisher, M.L. 1981. The Lagrangian relaxation method for solving integer programming problems. *Management Science* **27** (1) 1-18.

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

Fisher, M.L. 1985. An applications oriented guide to Lagrangian relaxation. *Interfaces* **15** (2) 10-21.

Fumero, F., and C. Vercellis. 1999. Synchronized development of production, inventory, and distribution schedules. *Transportation Science* **33** (3) 330-340.

Gallego, G., and D. Simchi-Levi. 1990. On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer R-systems. *Management Science* **36** (2) 240-243.

Georgia Institute of Technology. March 2005. Concorde TSP Solver. Available from website *http://www.tsp.gatech.edu/concorde*.

King, R.H., and R.R. Love. 1980. Coordinating decisions for increased profits. *Interfaces* **10** (6) 4-19.

Kleywegt, A.J., V.S. Nori, and M.W.P. Savelsbergh. 2002. The stochastic inventory routing problem with direct deliveries. *Transportation Science* **36** (1) 94-118.

Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. 1985. *The traveling salesman problem: A guided tour of combinatorial optimization*. John Wiley & Sons.

Martin, C.H., D.C. Dent, and J.C. Eckhart. 1993. Integrated production, distribution and inventory planning at Libbey-Owens-Ford. *Interfaces* **23** (3) 68-78.

Pınar, Ö., and H. Süral. 2004. An integrated approach to inventory routing problem (Bütünleşik envanter yönetimi ve güzergah belirleme problemi). *Proceedings of the 24th National Conference on Operational Research and Industrial Engineering*. Gaziantep University Press. 91-93.

Sarmiento, A.M., and R. Nagi. 1999. A review of integrated analysis of production-distribution systems. *IIE Transactions* **31** 1061-1074.

Thomas, D.J., and P.M. Griffin. 1996. Coordinated supply chain management. *European Journal of Operational Research* **94** 1-15.

Viswanathan, S., and K. Mathur. 1997. Integrating routing and inventory decisions in one-warehouse multiretailer multiproduct distribution systems. *Management Science* **43** (3) 294-312.

Wagner, H.M., and T.M. Whitin. 1958. Dynamic version of the economic lot size model, *Management Science* **5** (1) 89-96.

APPENDIX A

PRELIMINARY EXPERIMENTS FOR *LR* BASED SOLUTION PROCEDURES

In this appendix, we present numerical values obtained on small instances using LR based solution procedures, *Every* and *Latest* heuristics, besides optimal solution values and LP relaxation values computed by CPLEX.

Table A.1 Detailed results on preliminary experiments

Instance	Opt	LPR	*Every*	*Latest*
pre1	9038.51	4828.98	14744.15	10114.21
pre2	9935.69	5230.47	16388.80	10751.98
pre19	9872.02	5397.11	15987.90	11047.22
pre28	4796.73	1875.06	13888.60	5413.72
pre31	5922.54	2509.08	13691.85	7219.71
pre32	5939.10	2188.22	14191.00	7075.78
pre3	7515.75	4445.13	15219.50	8069.00
pre8	8419.13	4318.10	15617.50	9033.32
pre15	11635.83	6236.21	19854.35	13200.47
pre24	4714.14	2130.77	11666.95	6174.95
pre36	5341.38	2115.95	14522.15	6524.07
pre37	5490.51	2552.33	15531.25	6577.19
npre3	6933.75	4271.42	15219.50	7293.00
npre8	7888.13	4137.93	15617.50	8325.32
npre15	10861.83	6100.13	19854.35	12168.47
npre24	4604.14	2072.07	11666.95	5954.95
npre36	5125.38	2065.53	14522.15	6236.07
npre37	5235.51	2514.82	15531.25	6237.19

Table A.2 Detailed results on preliminary experiments for *LR1* algorithms

	***LR1* (*a* /1.005)**		***LR1* (*a* /2, 30)**		***LR1* (*a*/ 2, 10)**	
Instance	**UB**	**LB**	**UB**	**LB**	**UB**	**LB**
pre1	9229.29	6964.12	9229.29	7329.43	9229.29	7062.87
pre2	10539.40	7772.97	10216.21	8168.73	10539.40	7773.60
pre19	9872.02	7644.23	9872.02	7918.69	9872.02	7768.29
pre28	4798.13	3393.67	4798.13	3495.15	4798.13	3204.19
pre31	6103.49	3793.18	5969.54	4116.34	6103.49	4026.75
pre32	5939.10	4155.36	5939.10	4442.84	6021.62	4003.42
pre3	7515.75	6436.78	7515.75	6457.89	7515.75	6289.06
pre8	8419.13	6614.09	8419.13	6839.13	8419.13	6620.11
pre15	11713.87	8469.77	11713.87	8649.24	11713.87	8562.14
pre24	4714.14	3035.23	4714.14	3377.29	4714.14	3322.09
pre36	5404.98	3356.22	5341.38	3463.52	5341.38	3509.05
pre37	5718.26	3672.49	5703.86	3710.31	5703.86	3725.95
npre3	6933.75	6179.72	6933.75	6178.60	6933.75	6067.27
npre8	7888.13	6390.41	7888.13	6529.66	7888.13	6356.79
npre15	10939.87	7872.57	10939.87	8335.11	10939.87	8160.17
npre24	4604.14	3085.53	4604.14	3315.24	4604.14	3261.58
npre36	5125.38	3170.46	5125.38	3223.82	5125.38	3396.73
npre37	5463.26	3647.44	5448.86	3657.01	5463.26	3604.61

Table A.3 Detailed results on preliminary experiments for *LR2* algorithms

	***LR2* (*a* /1.005)**		***LR2* (*a* /2, 30)**		***LR2* (*a*/ 2, 10)**	
Instance	**UB**	**LB**	**UB**	**LB**	**UB**	**LB**
pre1	9229.29	5209.04	9229.29	5543.79	9229.29	5684.72
pre2	10710.28	5557.47	10710.28	5899.83	10710.28	6111.72
pre19	9872.02	5669.77	9872.02	5864.87	9872.02	6186.59
pre28	4865.33	1686.52	4798.13	1835.96	4798.13	2063.29
pre31	6372.39	2436.79	6138.79	2686.40	6138.79	2816.41
pre32	6654.18	1899.18	6652.22	2277.38	6652.22	2436.76
pre3	7515.75	4885.38	7515.75	5032.19	7515.75	5248.10
pre8	8419.13	4507.53	8419.13	4875.52	8419.13	5006.42
pre15	11713.87	6516.87	11713.87	6897.99	11713.87	7085.22
pre24	5646.47	1625.35	5612.61	2027.66	5528.73	2339.36
pre36	5404.98	2245.54	5404.98	2328.96	5404.98	2449.45
pre37	5754.74	2700.39	5754.74	2842.35	5754.74	2912.45
npre3	6933.75	4866.54	6933.75	5041.10	6933.75	5074.38
npre8	8008.83	4582.98	7888.13	4719.60	8008.83	4826.27
npre15	10939.87	6516.87	10939.87	6739.58	10939.87	6949.12
npre24	5459.13	1625.35	5363.73	2280.30	5363.73	2280.30
npre36	5289.27	2048.07	5188.98	2282.50	5188.98	2398.78
npre37	5499.74	2629.79	5474.99	2813.80	5474.99	2874.61

Table A.4 Detailed results on preliminary experiments for *ELR* algorithms

	ELR1		***ELR2***	
Instance	**UB**	**LB**	**UB**	**LB**
pre1	9124.04	6751.22	9124.04	6533.49
pre2	9935.69	6872.35	9935.69	6872.35
pre19	9872.02	7013.06	9872.02	6943.62
pre28	4798.13	2385.73	4798.13	2382.26
pre31	5969.54	3170.00	5969.54	3118.94
pre32	5939.10	2699.18	5939.10	2421.10
pre3	7515.75	6352.79	7515.75	5897.65
pre8	8419.13	5729.80	8419.13	5561.83
pre15	11673.05	8954.47	11673.05	8954.47
pre24	4714.14	2082.73	4714.14	2082.73
pre36	5366.42	3054.42	5366.42	3054.42
pre37	5729.24	3650.54	5754.74	3650.54
npre3	6933.75	5770.79	6933.75	5315.65
npre8	7888.13	5198.80	7888.13	5030.83
npre15	10939.87	8180.47	10939.87	8180.47
npre24	4604.14	1917.73	4604.14	1917.73
npre36	5188.98	2910.42	5188.98	2910.42
npre37	5474.24	3395.54	5474.99	3395.54

APPENDIX B

CONVERGENCE GRAPHS FOR *LR1*

Here, figures regarding convergence of *LR1* algorithm on small instances are presented.

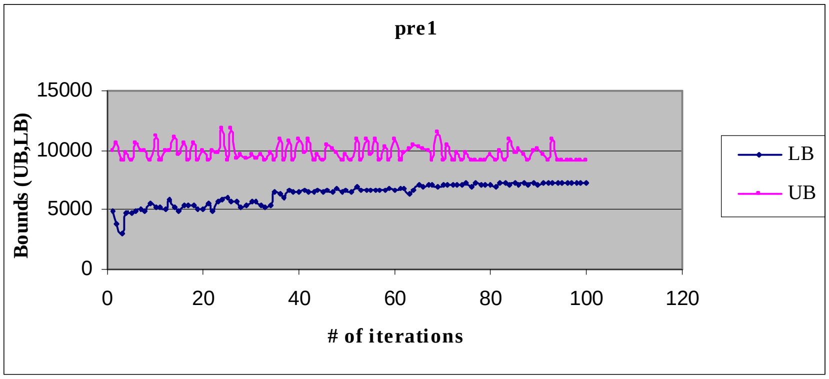

Figure B.1 Bounds versus no. of iterations graph for pre1

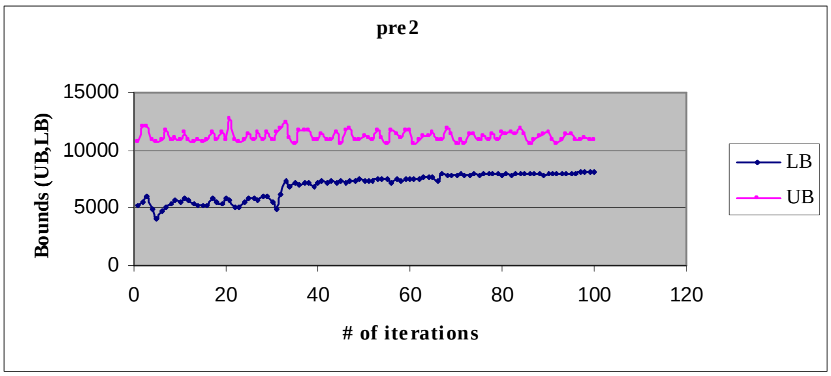

Figure B.2 Bounds versus no. of iterations graph for pre2

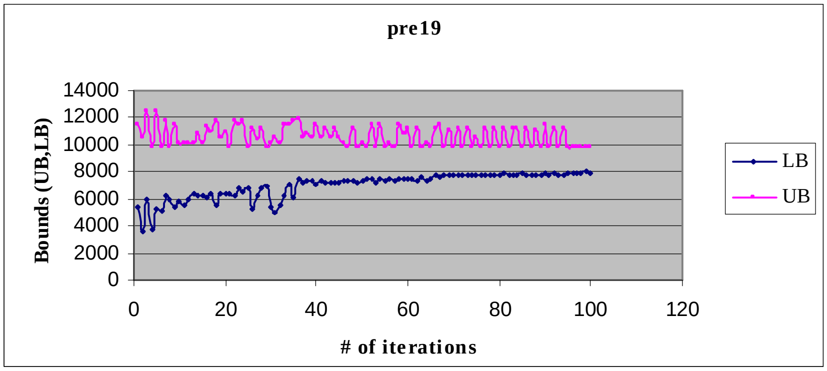

Figure B.3 Bounds versus no. of iterations graph for pre19

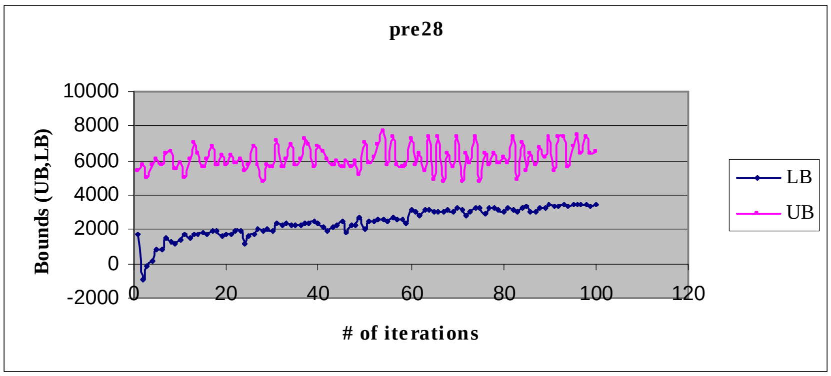

Figure B.4 Bounds versus no. of iterations graph for pre28

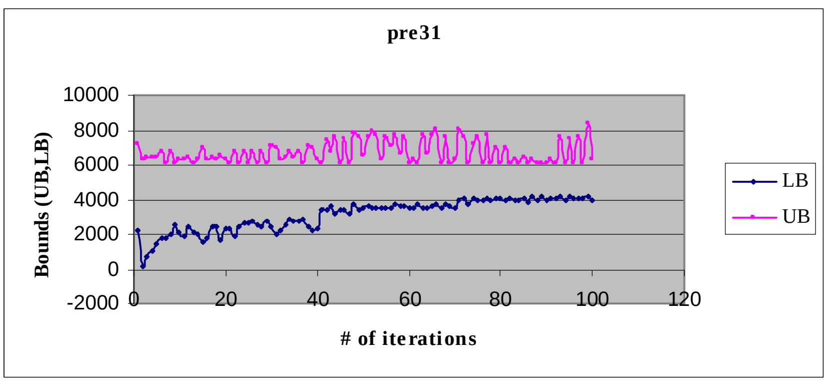

Figure B.5 Bounds versus no. of iterations graph for pre31

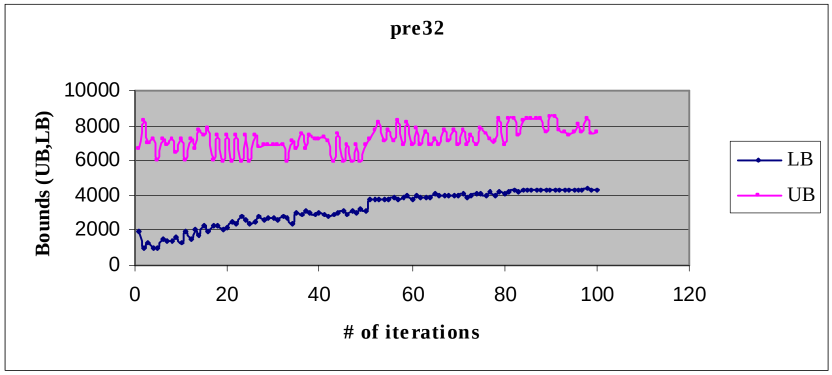

Figure B.6 Bounds versus no. of iterations graph for pre32

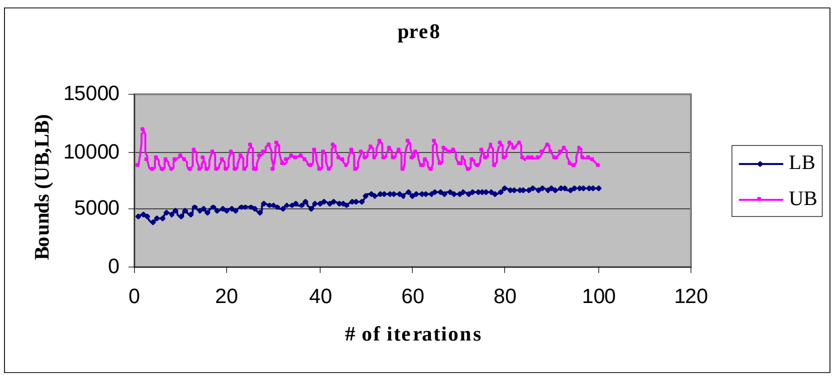

Figure B.7 Bounds versus no. of iterations graph for pre8

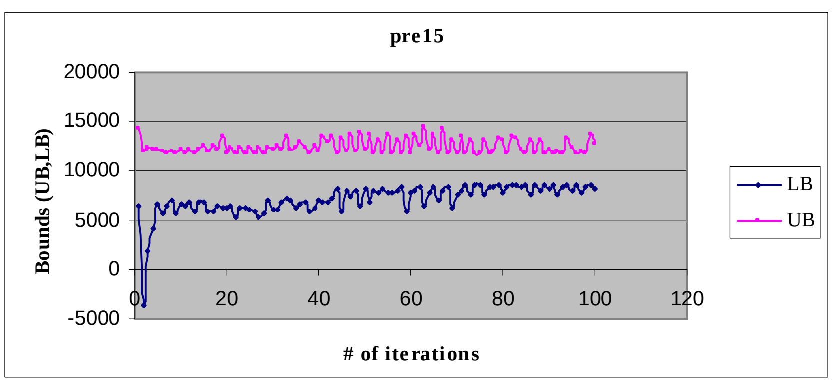

Figure B.8 Bounds versus no. of iterations graph for pre15

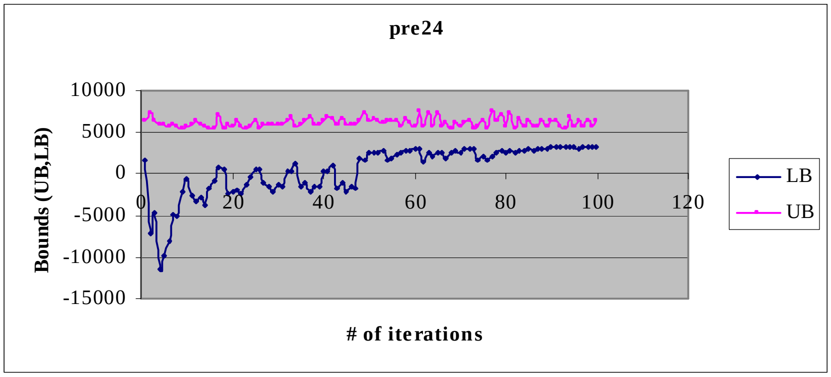

Figure B.9 Bounds versus no. of iterations graph for pre24

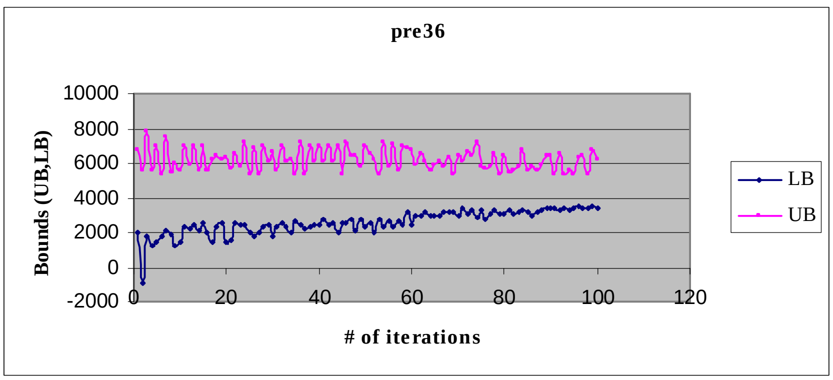

Figure B.10 Bounds versus no. of iterations graph for pre36

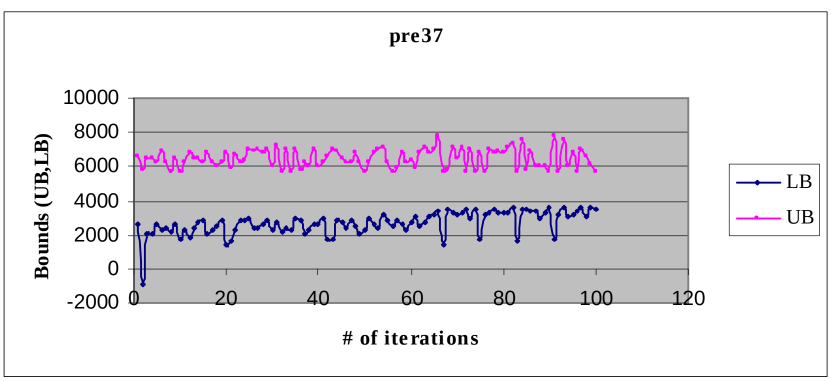

Figure B.11 Bounds versus no. of iterations graph for pre37

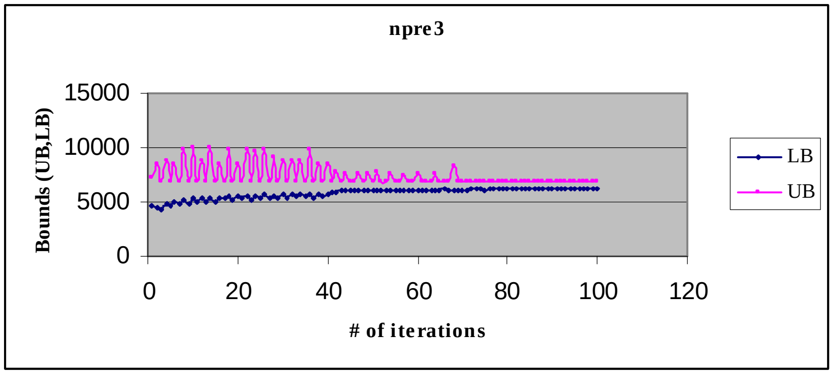

Figure B.12 Bounds versus no. of iterations graph for npre3

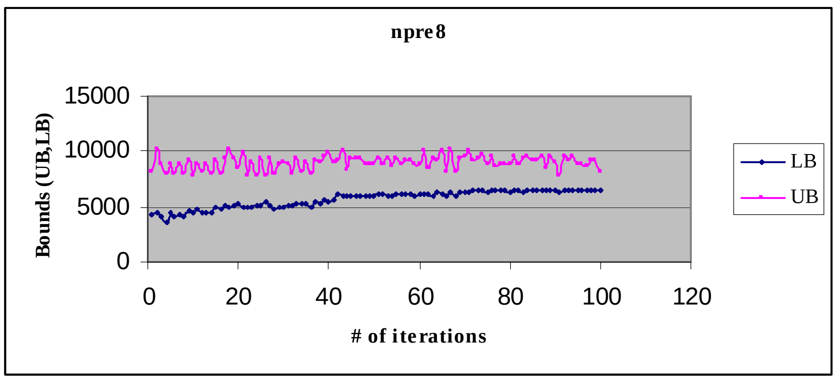

Figure B.13 Bounds versus no. of iterations graph for npre8

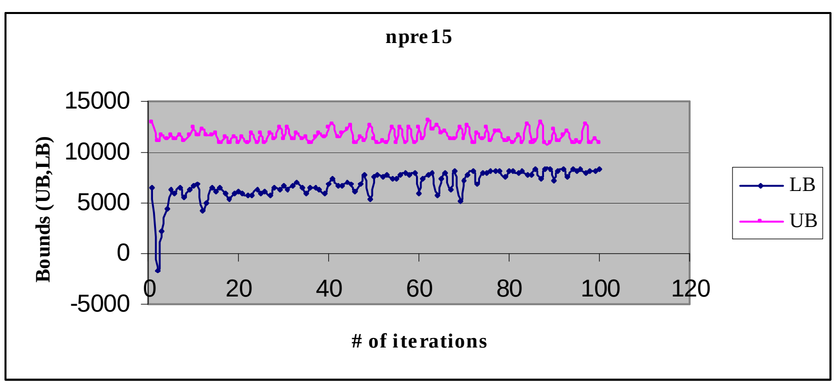

Figure B.14 Bounds versus no. of iterations graph for npre15

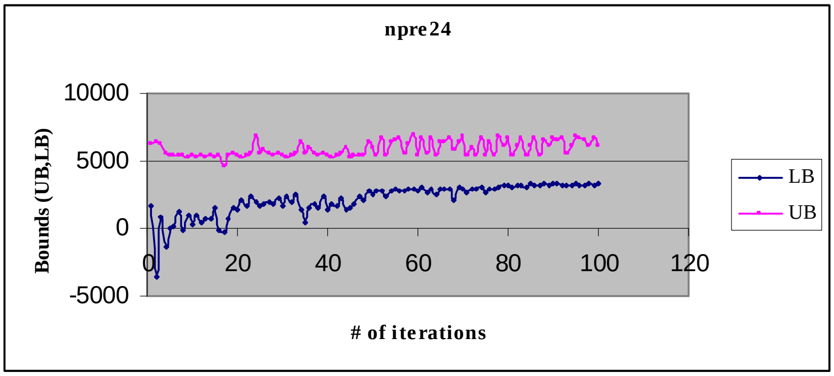

Figure B.15 Bounds versus no. of iterations graph for npre24

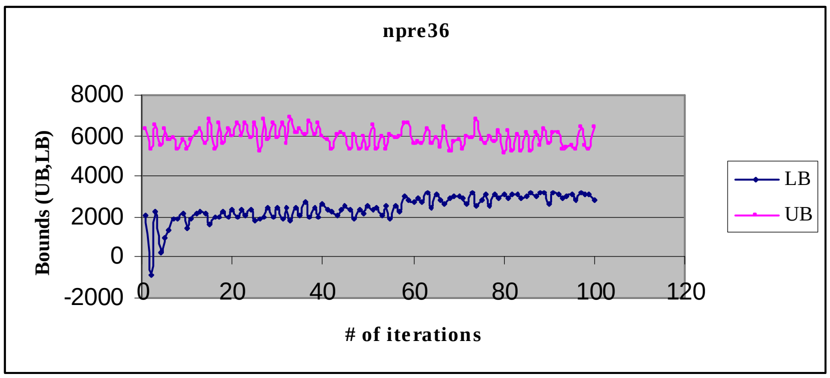

Figure B.16 Bounds versus no. of iterations graph for npre36

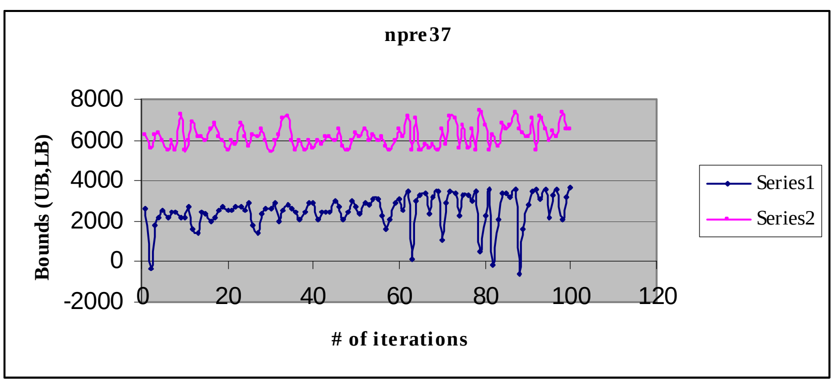

Figure B.17 Bounds versus no. of iterations graph for npre37

APPENDIX C

CONVERGENCE GRAPHS FOR *LR2*

In this appendix, figures regarding convergence of *LR2* algorithm on small instances are presented.

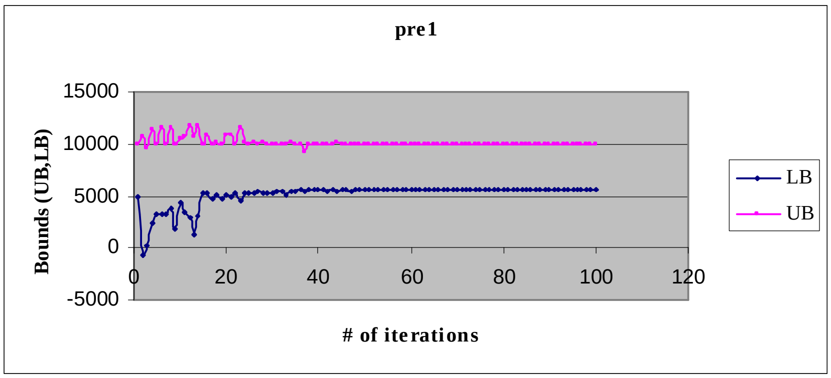

Figure C.1 Bounds versus no. of iterations graph for pre1

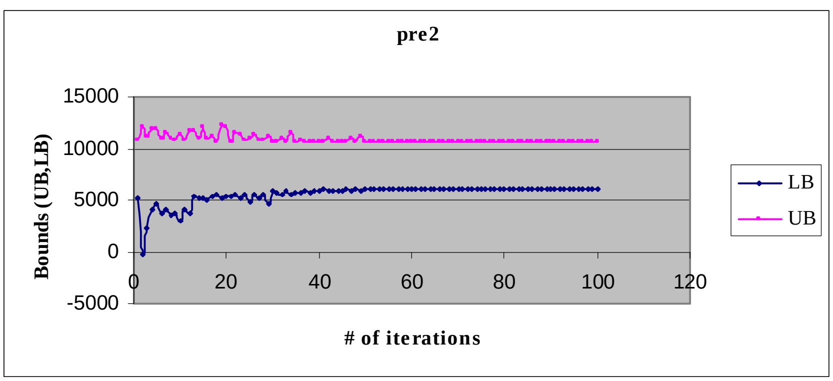

Figure C.2 Bounds versus no. of iterations graph for pre2

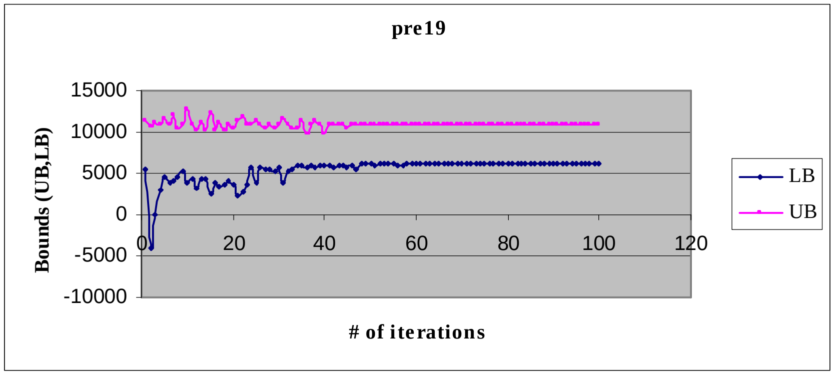

Figure C.3 Bounds versus no. of iterations graph for pre19

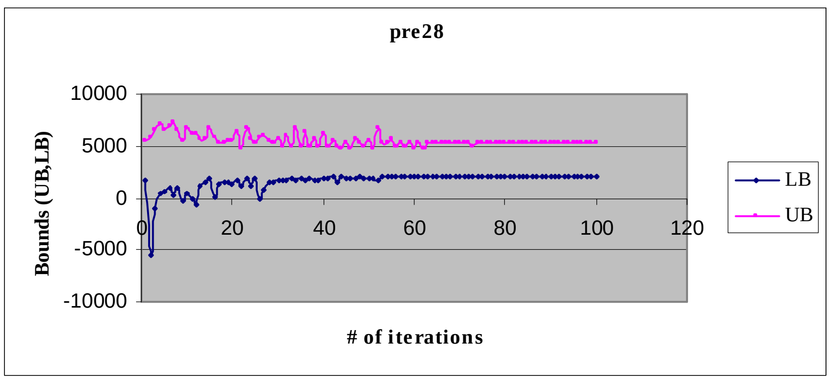

Figure C.4 Bounds versus no. of iterations graph for pre28

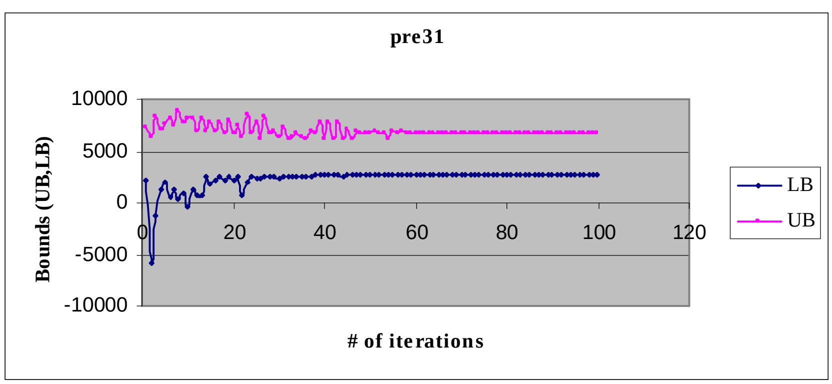

Figure C.5 Bounds versus no. of iterations graph for pre31

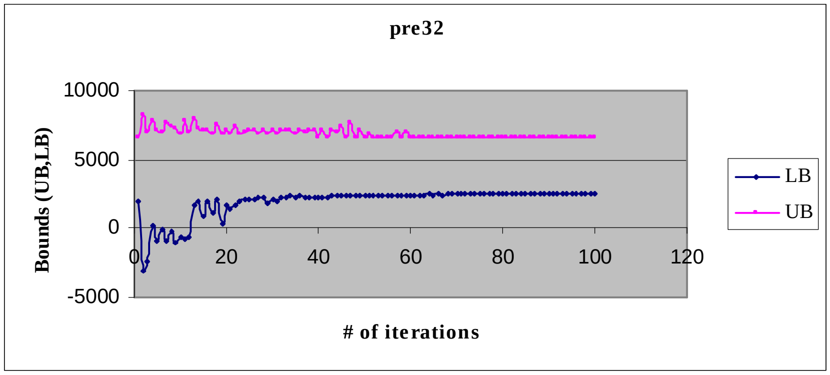

Figure C.6 Bounds versus no. of iterations graph for pre32

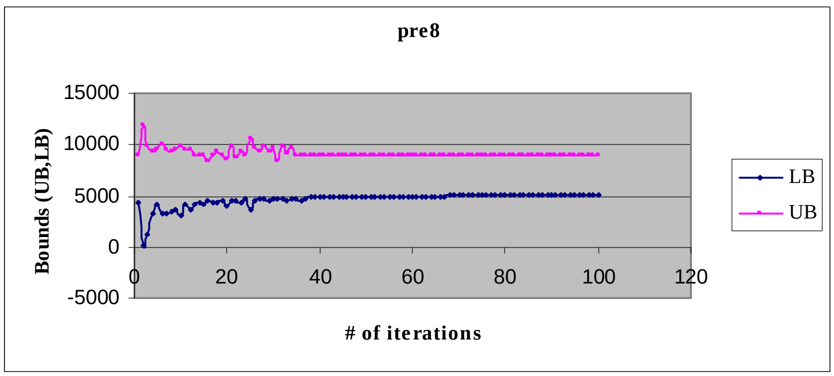

Figure C.7 Bounds versus no. of iterations graph for pre8

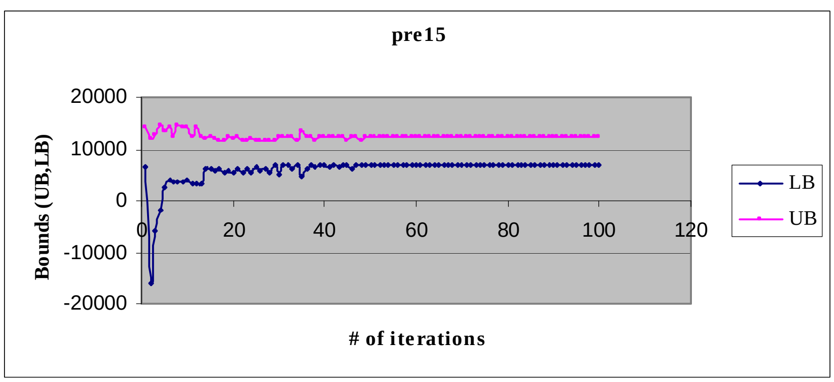

Figure C.8 Bounds versus no. of iterations graph for pre15

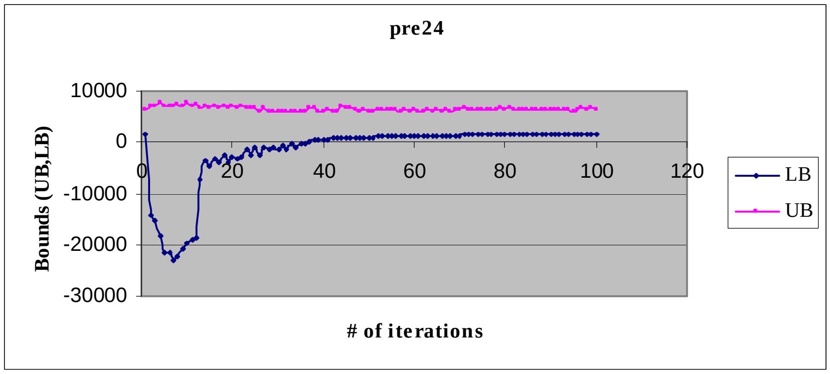

Figure C.9 Bounds versus no. of iterations graph for pre24

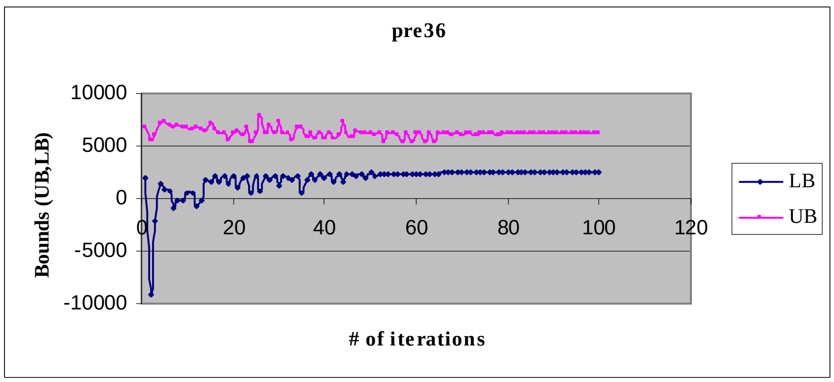

Figure C.10 Bounds versus no. of iterations graph for pre36

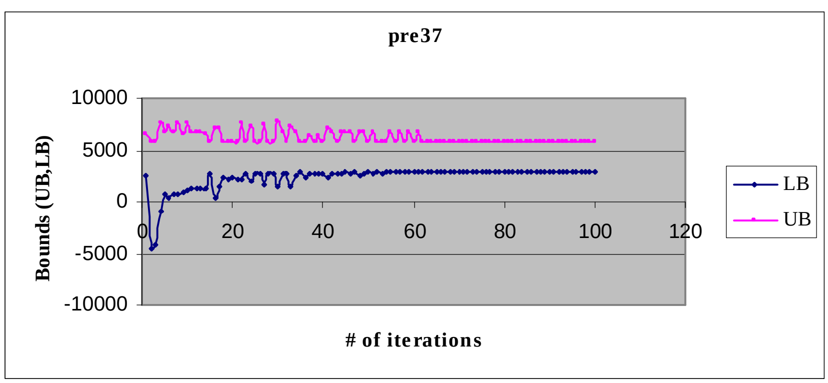

Figure C.11 Bounds versus no. of iterations graph for pre37

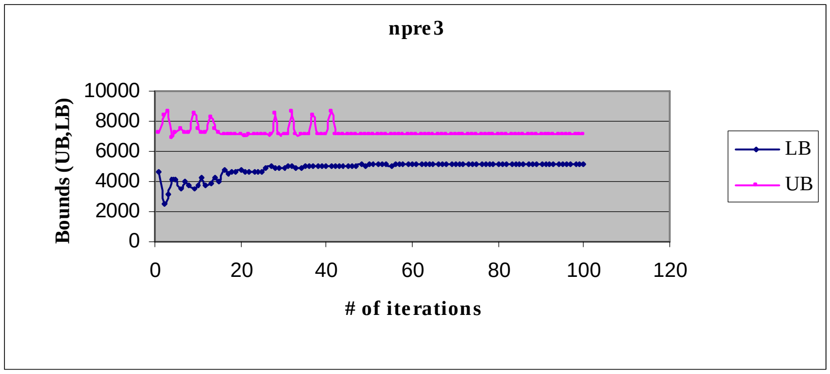

Figure C.12 Bounds versus no. of iterations graph for npre3

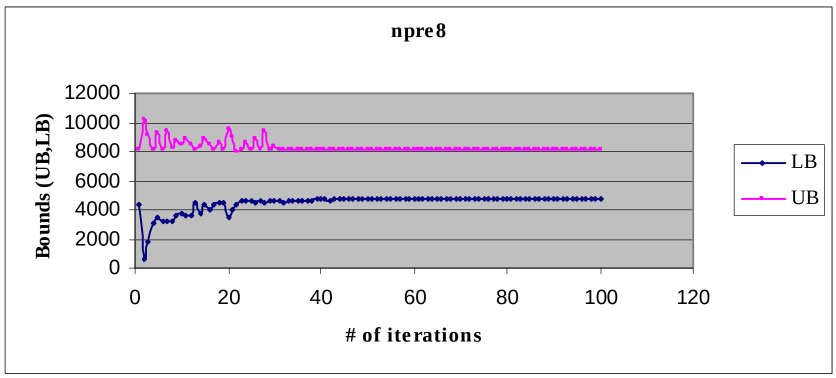

Figure C.13 Bounds versus no. of iterations graph for npre8

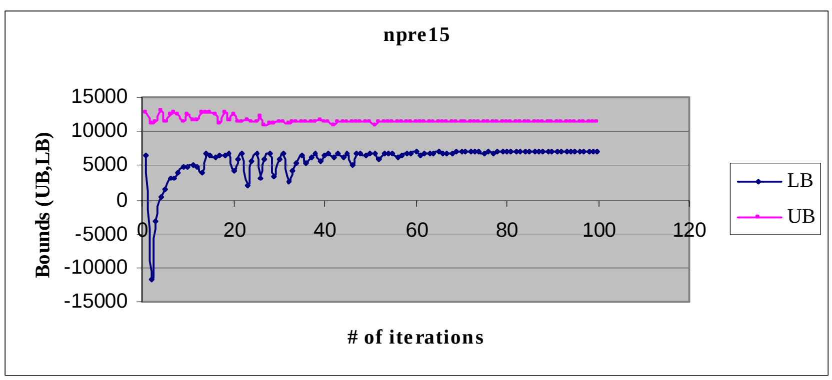

Figure C.14 Bounds versus no. of iterations graph for npre15

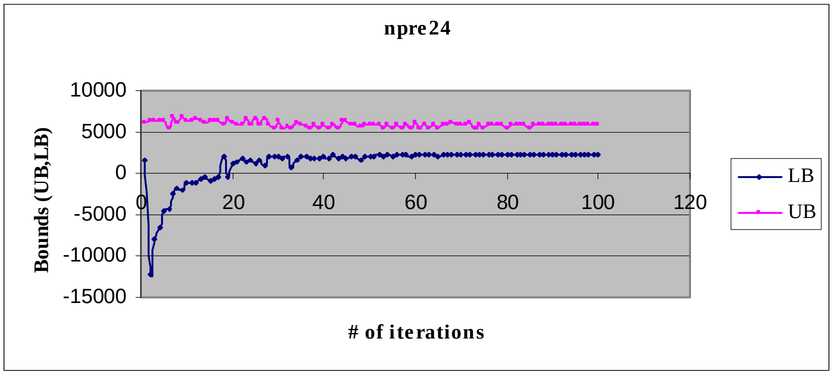

Figure C.15 Bounds versus no. of iterations graph for npre24

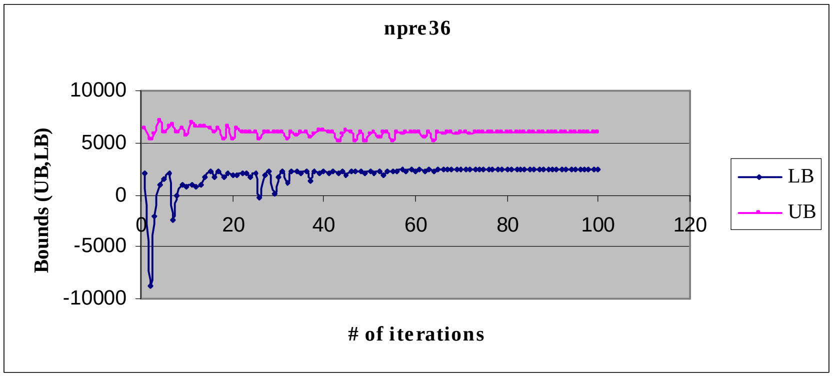

Figure C.16 Bounds versus no. of iterations graph for npre36

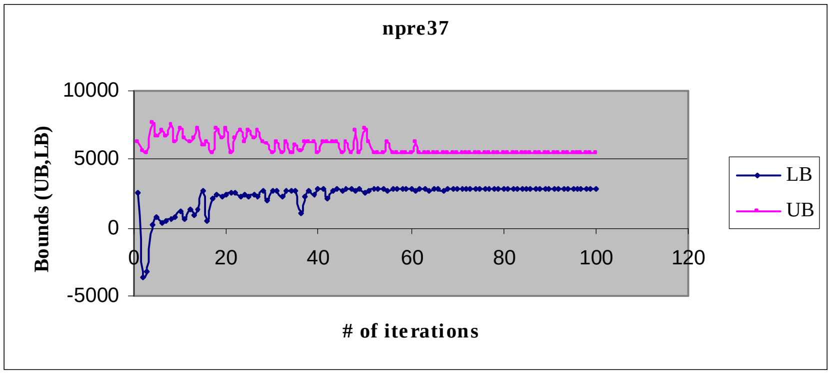

Figure C.17 Bounds versus no. of iterations graph for npre37

APPENDIX D

EXPERIMENTS ON LARGE INSTANCES WITH *LR2* AND *ELR2*

In this appendix, we present numerical values obtained on large instances using *LR2*, *ELR2*, *Every* and *Latest* heuristics, besides LP relaxation values computed by CPLEX.

Table D.1 Results on instances with high fixed vehicle dispatching cost

				LR2		*ELR2*	
Instances	**LPR**	***Every***	***Latest***	**UB**	**LB**	**UB**	**LB**
H1	351891.74	1135236.30	546802.95	545820.35	465131.04	542917.55	468231.28
H2	355804.75	1263766.00	580413.35	578967.35	494383.73	577023.35	497941.20
H3	254625.51	978588.10	461774.75	460673.05	381094.44	458501.15	383660.18
H4	326336.38	1095045.60	523338.15	526625.19	433969.73	519776.05	436680.61
H5	206435.68	791109.00	402211.25	401092.15	324564.69	400308.19	325624.89
H6	321747.35	1138607.70	534250.45	532975.15	446669.57	530246.65	448795.76
H7	288223.13	1009728.30	491652.35	488483.05	413196.92	487579.05	415940.56
H8	329863.56	1052818.10	511730.90	510675.50	422992.25	510239.40	425621.82
H9	282722.11	999965.10	506973.85	506125.35	423941.99	503805.25	426187.41
H10	314283.33	1048189.50	515315.90	511978.70	431636.28	511522.90	434072.13
H11	356368.69	1173691.70	568743.95	567231.35	466191.64	564457.31	469557.28
H12	360542.21	1307666.40	605514.35	603766.35	495480.22	600755.95	499455.20
H13	259089.34	1018863.50	484533.25	481401.45	382381.28	479065.51	385049.18
H14	330346.01	1136780.80	549338.65	551079.81	434731.16	544957.41	438119.61
H15	211213.15	827914.20	423166.25	421801.25	325581.84	419460.05	326893.89
H16	326663.68	1182138.10	559783.45	556116.65	447713.36	553381.97	450296.76
H17	293117.72	1046968.50	512408.85	511086.85	414241.15	507600.45	417224.56
H18	333403.51	1092896.10	537368.90	534359.50	424151.85	533487.30	427003.82
H19	287531.25	1041555.30	530277.35	527586.35	425082.22	525252.17	427621.41
H20	318263.44	1087484.50	539128.40	533562.40	432798.27	534700.72	435427.13
H21	134658.40	1026246.30	235528.45	236107.03	175913.22	233618.63	176934.78
H22	122787.58	1152181.00	233611.35	234697.55	172697.51	233243.39	174004.20
H23	88136.72	873498.10	189839.75	190149.42	131767.16	189520.58	132526.18
H24	116367.00	983160.60	219698.65	220067.15	153532.16	219386.71	154747.11
H25	79242.24	689724.00	177576.75	178859.25	120122.84	175456.79	120172.89
H26	115521.19	1030592.70	221215.95	220186.05	157532.39	220637.19	158459.76
H27	112099.27	911793.30	217321.35	219818.02	159959.31	217511.40	160955.06
H28	120537.21	945538.10	219258.40	218790.03	153188.38	218799.91	153972.32
H29	114382.99	887360.10	228375.35	228965.75	168571.87	226468.55	169152.41
H30	122875.71	945424.50	226770.40	229978.62	164811.32	226560.29	165905.13

Table D.1 (Continued)

				LR2		*ELR2*	
Instances	**LPR**	***Every***	***Latest***	**UB**	**LB**	**UB**	**LB**
H31	136454.47	1064701.70	257469.45	256026.65	176952.13	252828.98	178260.78
H32	124979.69	1196081.40	258712.35	255915.55	174314.39	255696.59	175518.20
H33	90085.52	913773.50	212598.25	211951.18	133129.88	209398.24	133915.18
H34	118546.07	1024895.80	245699.15	243395.46	154948.63	243395.46	156186.11
H35	80971.40	726529.20	198531.75	196957.35	121354.40	195170.74	121441.89
H36	117602.85	1074123.10	246748.95	245064.95	157668.35	243664.84	159960.76
H37	113959.99	949033.50	238077.85	239022.45	161487.97	238577.45	162239.06
H38	122266.72	985616.10	244896.40	243855.11	154399.58	244025.96	155354.32
H39	116642.58	928950.30	251678.85	245624.23	169995.49	247235.43	170586.41
H40	124776.89	984719.50	250582.90	251909.61	166256.78	249931.19	167260.13
H41	352775.85	1229916.30	581568.95	578678.95	465179.75	577988.31	468531.28
H42	356502.56	1346956.00	613666.35	614370.75	494526.75	611442.25	498247.20
H43	255754.95	1068438.10	495200.75	494392.45	381428.45	491410.15	383966.18
H44	327500.44	1184775.60	561281.15	560661.21	434833.36	559339.00	436896.61
H45	208427.67	874629.00	437366.25	441711.95	325523.42	435379.19	326374.89
H46	322733.02	1224227.70	570187.45	568563.15	446984.86	565704.15	448981.76
H47	289335.97	1099158.30	524363.35	524484.86	413486.64	523622.73	416138.56
H48	330687.74	1140208.10	550896.90	549966.50	423472.25	549271.40	425711.82
H49	283836.98	1089215.10	539319.85	541424.75	424564.23	535876.05	426307.41
H50	315593.31	1130599.50	551272.90	547329.70	431767.23	546865.32	434396.13
H51	357252.80	1268371.70	603509.95	599165.98	467013.93	598397.21	469857.28
H52	361240.01	1390856.40	638767.35	633348.15	496375.01	631058.20	499761.20
H53	260218.78	1108713.50	517959.25	516991.05	382885.39	512441.65	385355.18
H54	331510.07	1226510.80	587281.65	585007.21	436047.57	584905.50	438335.61
H55	213205.14	911434.20	458321.25	462733.69	327160.45	451987.64	327643.89
H56	327649.35	1267758.10	595720.45	594736.25	448203.85	588674.97	450482.76
H57	294230.55	1136398.50	545119.85	543570.86	414635.35	538638.26	417422.56
H58	334227.69	1180286.10	576534.90	573650.50	424761.46	573650.50	427093.82
H59	288646.12	1130805.30	562623.35	562453.75	425852.95	557343.17	427741.41
H60	319573.42	1169894.50	575085.40	569435.20	432965.56	567616.32	435751.13
H61	135542.50	1120926.30	270294.45	267434.48	176277.55	265427.94	177234.78
H62	123485.38	1235371.00	266864.35	269441.51	173218.99	266446.87	174310.20
H63	89266.15	963348.10	223265.75	225875.78	132041.06	223505.54	132832.18
H64	117531.05	1072890.60	257641.65	257771.25	154210.21	257077.71	154963.11
H65	81234.22	773244.00	212731.75	214045.86	121131.42	212374.35	120922.89
H66	116506.85	1116212.70	257152.95	258384.35	158152.34	256622.29	158645.76
H67	113212.11	1001223.30	250032.35	253623.49	160605.54	251984.19	161153.06
H68	121361.39	1032928.10	258424.40	261689.72	152977.16	258365.72	154062.32
H69	115497.86	976610.10	260721.35	261354.55	168744.59	259618.83	169272.41
H70	124185.69	1027834.50	262727.40	261653.00	165065.53	262295.63	166229.13
H71	137338.57	1159381.70	292235.45	293460.48	177668.51	287218.51	178560.78
H72	125677.50	1279271.40	291965.35	294094.21	174653.08	289087.98	175824.20
H73	91214.95	1003623.50	246024.25	243732.08	133561.13	242699.12	134221.18
H74	119710.13	1114625.80	283642.15	284099.06	155776.09	282789.66	156402.11
H75	82963.39	810049.20	233686.75	231520.44	122268.05	230800.74	122191.89
H76	118588.51	1159743.10	282685.95	281650.55	159266.74	282685.53	160146.76
H77	115072.83	1038463.50	270788.85	271900.05	161974.33	272086.54	162437.06
H78	123090.90	1073006.10	284062.40	284847.95	154688.33	282283.20	155444.32
H79	117757.45	1018200.30	284024.85	282633.63	170417.64	280949.83	170706.41
H80	126086.87	1067129.50	286539.90	286811.70	166804.99	287868.13	167584.13

Table D.2 Results on instances with low fixed vehicle dispatching cost

				LR2		*ELR2*	
Instances	**LPR**	***Every***	***Latest***	**UB**	**LB**	**UB**	**LB**
L1	349762.36	1097076.30	518818.95	517427.35	463557.82	516481.75	464415.28
L2	353504.98	1221436.00	549371.35	547925.35	492718.36	547392.35	493708.20
L3	251668.87	939768.10	433306.75	432205.05	378871.78	430938.45	379778.18
L4	323587.91	1053975.60	493220.15	491900.45	431881.38	490756.45	432573.61
L5	202897.08	754809.00	375591.25	374472.15	321912.84	373374.25	321994.89
L6	319010.04	1096367.70	503274.45	501999.15	443719.17	500678.65	444571.76
L7	285788.08	971868.30	463888.35	464413.75	411370.59	461077.05	412154.56
L8	327279.57	1013908.10	483196.90	482141.50	421136.51	481665.00	421730.82
L9	279326.53	957365.10	475733.85	474697.35	421029.54	473985.25	421927.41
L10	311792.83	1009549.50	486979.90	487002.00	429246.40	484439.00	430208.13
L11	354239.30	1135531.70	540759.95	536716.35	464831.35	536716.35	465741.28
L12	358242.44	1265336.40	574472.35	570626.35	493523.54	570093.35	495222.20
L13	256132.70	980043.50	456065.25	452829.05	380334.27	452358.65	381167.18
L14	327597.54	1095710.80	519220.65	515556.45	433297.20	515473.65	434012.61
L15	207674.56	791614.20	396546.25	393028.65	323184.23	392193.25	323263.89
L16	323926.36	1139898.10	528807.45	525341.65	445315.61	523958.65	446072.76
L17	290682.66	1009108.50	484644.85	480845.05	412616.17	478859.05	413438.56
L18	330819.52	1053986.10	508834.90	505825.50	422249.78	505825.50	423112.82
L19	284135.67	998955.30	499037.35	496615.75	422799.23	495507.75	423361.41
L20	315772.94	1048844.50	510792.40	507610.20	430748.80	506119.72	431563.13
L21	133861.55	1011966.30	225056.45	225504.25	175261.19	223545.50	175506.78
L22	121969.38	1137121.00	222567.35	224675.51	172205.43	221994.59	172498.20
L23	87081.10	859638.10	179675.75	179323.85	130918.05	179692.30	131140.18
L24	115351.12	967980.60	208566.65	208395.55	153107.45	208760.71	153229.11
L25	77932.08	676284.00	167720.75	167342.15	118707.97	166343.55	118828.89
L26	114535.52	1015382.70	210061.95	209965.25	156808.49	209483.19	156938.76
L27	111151.87	897063.30	206519.35	206352.25	159298.74	205682.11	159482.06
L28	119604.82	931498.10	208962.40	208460.43	152352.62	208460.43	152568.32
L29	113046.28	870590.10	216077.35	215557.55	167473.02	215131.69	167475.41
L30	121957.24	931174.50	216320.40	217811.10	164331.09	216500.83	164480.13
L31	135657.62	1050421.70	246997.45	244710.98	176293.64	244284.98	176832.78
L32	124161.49	1181021.40	247668.35	248189.03	173110.15	244652.59	174012.20
L33	89029.90	899913.50	202434.25	202021.18	132362.28	200502.38	132529.18
L34	117530.20	1009715.80	234567.15	232826.56	154604.70	232826.56	154668.11
L35	79661.24	713089.20	188675.75	185705.75	120159.93	184983.28	120097.89
L36	116617.18	1058913.10	235594.95	234296.85	158353.76	232844.30	158439.76
L37	113012.60	934303.50	227275.85	226624.25	159950.25	227284.45	160766.06
L38	121334.33	971576.10	234600.40	232241.27	153311.40	232241.27	153950.32
L39	115305.87	912180.30	239380.85	235688.05	168882.15	236566.83	168909.41
L40	123858.42	970469.50	240132.90	240812.37	165769.49	237201.96	165835.13

Table D.2 (Continued)

				LR2		ELR2	
Instances	**LPR**	**Every**	**Latest**	**UB**	**LB**	**UB**	**LB**
L41	350646.46	1191756.30	553584.95	554270.00	463895.06	551229.35	464715.28
L42	354202.79	1304626.00	582624.35	581177.35	493129.59	580582.35	494014.20
L43	252798.31	1029618.10	466732.75	465811.05	379358.13	464553.05	380084.18
L44	324751.97	1143705.60	531163.15	532982.85	432395.34	530165.70	432789.61
L45	204889.07	838329.00	410746.25	410146.35	322976.87	408011.25	322744.89
L46	319995.70	1181987.70	539211.45	537982.15	444421.21	535977.15	444757.76
L47	286900.92	1061298.30	496599.35	498128.55	411967.42	493671.45	412352.56
L48	328103.75	1101298.10	522362.90	521432.50	421138.16	520978.80	421820.82
L49	280441.40	1046615.10	508079.85	507826.25	421746.71	507471.35	422047.41
L50	313102.81	1091959.50	522936.90	524618.50	429760.31	519348.90	430532.13
L51	355123.41	1230211.70	575525.95	573778.68	465271.90	571734.31	466041.28
L52	358940.24	1348526.40	607725.35	605919.14	494648.12	603509.95	495528.20
L53	257262.13	1069893.50	489491.25	488719.65	380858.17	486031.65	381473.18
L54	328761.59	1185440.80	557163.65	555061.45	433978.08	553601.65	434228.61
L55	209666.54	875134.20	431701.25	431813.65	324229.86	428652.25	324013.89
L56	324912.03	1225518.10	564744.45	561324.65	445764.36	559371.45	446258.76
L57	291795.50	1098538.50	517355.85	516937.65	413111.98	514179.03	413636.56
L58	331643.70	1141376.10	548000.90	545116.50	422768.73	545116.50	423202.82
L59	285250.54	1088205.30	531383.35	529368.75	423359.51	528660.69	423481.41
L60	317082.92	1131254.50	546749.40	547852.20	430957.61	542451.60	431887.13
L61	134745.66	1106646.30	259822.45	259096.38	175799.37	256767.98	175806.78
L62	122667.18	1220311.00	255820.35	258564.07	172257.62	257950.32	172804.20
L63	88210.54	949488.10	213101.75	212911.25	131420.38	214131.98	131446.18
L64	116515.18	1057710.60	246509.65	246775.96	153579.61	246451.71	153445.11
L65	79924.07	759804.00	202875.75	201887.85	120148.18	203481.20	119578.89
L66	115521.18	1101002.70	245998.95	248422.05	157366.67	244998.49	157124.76
L67	112264.71	986493.30	239230.35	243092.71	159724.46	241398.92	159680.06
L68	120429.00	1018888.10	248128.40	250442.47	152538.87	247988.02	152658.32
L69	114161.15	959840.10	248423.35	248635.45	167656.15	249229.25	167595.41
L70	123267.22	1013584.50	252277.40	254723.80	164465.86	252320.63	164804.13
L71	136541.73	1145101.70	281763.45	280117.25	175957.45	276825.71	177132.78
L72	124859.30	1264211.40	280921.35	282919.05	173483.48	280945.56	174318.20
L73	90159.34	989763.50	235860.25	234051.45	132793.48	231687.60	132835.18
L74	118694.26	1099445.80	272510.15	271864.06	155130.60	271657.66	154884.11
L75	81653.23	796609.20	223830.75	220808.34	121478.06	219280.38	120847.89
L76	117602.84	1144533.10	271531.95	268942.95	158585.17	271904.43	158625.76
L77	114125.43	1023733.50	259986.85	265133.85	161362.24	261399.45	160964.06
L78	122158.51	1058966.10	273766.40	274094.95	153486.29	272728.64	154040.32
L79	116420.74	1001430.30	271726.85	268332.23	169270.35	266597.03	169029.41
L80	125168.40	1052879.50	276089.90	275011.90	164927.90	277812.80	166159.13

APPENDIX E

EXPERIMENTS ON DETERMINING WEIGHTS OF COST COMPONENTS

This appendix involves cost distribution of inventory holding cost at supplier, transportation cost and inventory holding cost at retailers on total cost for a set of small and corresponding large instances.

Table E.1 Values of cost components on total cost (UB)

	Small Instances				Large Instances			
Instances	Supp	Trans	Ret	UB	Supp	Trans	Ret	UB
pre3-H3	344.10	2693.00	4478.65	7515.75	12370.00	71406.00	376897.05	460673.05
pre8-H8	283.30	3832.00	4303.83	8419.13	14207.60	77205.00	419262.90	510675.50
pre15-H15	755.40	4805.00	6153.47	11713.87	29550.00	73181.00	319070.25	421801.25
pre24-H24	243.00	3749.00	1536.73	5528.73	15446.40	53090.00	151530.75	220067.15
pre36-H36	616.80	3060.00	1728.18	5404.98	37743.60	52055.00	155266.35	245064.95
pre37-H37	590.20	2871.00	2293.54	5754.74	29387.20	51740.00	157895.25	239022.45
npre3-L3	344.10	2111.00	4478.65	6933.75	12370.00	42938.00	376897.05	432205.05
npre8-L8	404.00	3301.00	4303.83	8008.83	14207.60	48671.00	419262.90	482141.5
npre15-L15	755.40	4031.00	6153.47	10939.87	29696.40	44262.00	319070.25	393028.65
npre24-L24	243.00	3584.00	1536.73	5363.73	14908.80	41956.00	151530.75	208395.55
npre36-L36	616.80	2844.00	1728.18	5188.98	37480.80	41535.00	155281.05	234296.85
npre37-L37	610.20	2578.00	2286.79	5474.99	29700.00	39029.00	157895.25	226624.25