2004

Order batching and picking optimization in terms of supply chain management (SCM)

Jaeyeon Won
Iowa State University

Order batching and picking optimization in terms of supply chain management (SCM)

by

Jaeyeon Won

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Program of Study Committee:
Sigurdur Olafsson, Major Professor
Doug Gemmill
Jo Min
Nilakanta Sree
Suzuki Yoshinori

Iowa State University

Ames, Iowa

2004

Graduate College
Iowa State University

This is to certify that the doctoral dissertation of

Jaeyeon Won

Has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

DEDICATION

I would like to dedicate this thesis to my wife and my parents, who encouraged me in this study and waited over a number of years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Warehousing and distribution are at the center of supply chain management (SCM), which is conceived with the management of logistics, manufacturing, production, inventory, distribution, and retailing. The emerging focus is on improving product movement, reducing inventory, and maximizing consumer satisfaction to achieve marketplace advantage. In this thesis, two classical control problems of bin packing problem (BPP) and traveling salesman problem (TSP) are investigated with respect to considerations motivated by supply chain management. For example, the supply-related activities such as the time of product movement, predictive order delivery time, and control of product movement across the supply chain can be operated more efficiently. In distribution center (DC), at the time of forming a batch, by considering the order holding time and picking time in the system, an algorithm based on the first-come, first served based (FCFS) rule is presented. In the proposed algorithm, both order picking factor and order holding factor are used in order to reflect the throughput of the system and response of orders.

Furthermore, some simple batching policies for considering the order holding time in the system are presented. They may result in meeting its customers' requirements for larger numbers of smaller orders and rapid turnaround. They reflect the well-known supply chain management (SCM) concept by taking the customer's need into consideration. Their performance measure of order response time that consists of order picking time and order holding time is evaluated and compared.

Finally, this thesis also develops a new optimization-based joint order batching and picking framework for warehousing and distribution systems. The nested partitions (NP) method that integrates global sampling of the feasible region and local search heuristic is applied to this problem. To speed up the computation, the improved NP-algorithm is developed. Also, by taking the special structure of the problem in account, an improved NP method in terms of small computational time is developed.

Within the supply, in general, the responsiveness versus efficiency tradeoff that companies make is one of deciding factors how much information to share with the other companies and how much information to keep private. The more information about products supply, customer demand, market forecast, and production schedules that companies share with each other, the more responsive to the market. Balancing this phenomenon is the concern that each company has about revealing information that could be used against it by a competitor. The potential costs associated with increased competition can harm the profitability of a company.

CHAPTER 1. INTRODUCTION

Warehousing and distribution are at the center of a new paradigm based on concepts of logistics, manufacturing, production, inventory, distribution, and retailing, which are often referred to jointing as supply chain management (SCM). The emerging focus is on improving product movement and on maximizing consumer satisfaction to achieve marketplace advantage. Today, manufacturers and distributors are judged not only by the quality of their products, but also by how quickly and efficiently they deliver goods to their customers or end users. While corporate executives are focusing more attention and resources on optimizing supply chain performance, the traditional warehouse and distribution center (DC) are now acknowledged as an area where improved management can be achieved.

The supply chain, also known as the logistics network, is a channel of suppliers, manufacturers, distribution centers or warehouses and customers, as well as raw materials and finished products that flow between the facilities. It encompasses all activities associated with the flow and transformation of goods from the raw materials to the end users, as well as the associated information flow. Information and material flows both up and down the supply chain. Supply chain management (SCM) can be defines as "the set of approaches utilized to efficiently integrate suppliers, manufacturers, warehouses, and stores, so that merchandise is produced and distributed at the right quantities, to the right locations, and at the right time, in order to minimize system-wide costs while satisfying service level requirements" (David, 2000). It is a process-oriented, integrated approach to procuring, producing, and delivering products and services to customers. SCM covers the management of material, information, and funds flows. In essence, supply chain management is all about delivering the right product to the right place, at the right time and at the right price - which is one of the most powerful engines of business transformation.

Increasing frequencies and volumes of orders has complicated management of efficient, effective product movement throughout the supply chain. At the same time, increasing customer demands are driving manufacturers and distributors to manage communication more effectively between channels of the supply chain. These trends have led to significant changes in the traditional role of the warehouse. As companies optimize their supply chains, integrators or third party logistic providers are offering solutions to manage the entire supply chain for their customers. Warehouses are becoming more specialized to accommodate value-added services like mass customization, customer service and return goods processing. The days of mass production to reduce production costs while inventory costs have gone high in favor of mass customization with efficient chains of value-added processes is fading away.

Mass customization can be defined as the flexibility to meet the demands of a customer base whose needs are diverse and/or changing. (Raines, 1995). The goal of mass customization is to combine aspects of the craft and mass production systems to create a high volume of varied products, with high quality at low cost. This system has been well documented in several sources (Monden, 1983; Shingo, 1981; Womack et al, 1990). Its inventory levels are very low compared to those in mass production facilities, allowing for rapid reaction to customers and changes in the marketplace and more customized products.

A growing number of organizations react to the demand for custom-made product, with a short product life cycle, quickly delivered and of excellent quality, by the strategic choice of mass customization. Also, faster customer service, greater product diversity, shorter product life cycles and globalization have all dramatically increased the complexity of running a business.

One clear trend in many markets, especially the computer industry, is that product life cycle becomes shorter. The company has to adjust new product development speed to the trend. Some problems may emerge in association with the adjustment. In order to be competitive in market, the company has to avoid cost increases with short product life cycle since prices of new products are held as they were or sometimes lowered. New facility investments should be restrained and inventory also should be lowered, in addition that product attractiveness should be competitive with rivals' products. It means the company has to catch the customer's needs quickly and adjust production capability per time unit given investment level to the shortened product life cycle.

One of the basic missions of warehouses or distribution centers is to meet the customer orders. The diversity and the volume of the stock-keeping unit (SKU) in the warehouse in which the items are requested are factors affecting the warehousing system. There are many factors to be considered for the warehousing system to be operated effectively and efficiently. Although it may seem that the only function of a warehouse is warehousing – that is, the temporary storage of goods – warehouses also perform many other functions, such as, receiving, storage, holding inventory, packing, and shipping.

In the past, the role of warehouses or distribution centers was very simple and static. They receive, store and retrieve products, keep inventory for safety, perform order picking and ship the customer's requirements. The stock keeping units (SKU) within the warehouse or distribution center (DC) have waited for a long time until their withdrawals are made. Today, market forces are changing the traditional concept of "warehousing". As a result of changing order profiles, ever more demanding customer requirements and a much greater recognition of the role logistics plays in reducing costs across the entire supply chain, high-speed/high-throughput distribution centers (DCs) have become the norm in most industries.

Several researchers introduced the design, planning and control problem of warehousing systems as new research topics (Graves et al., 1977, Hausman, et al., 1976, Schwarz et al, 1978). In particular, the operation problem of warehousing systems has received considerable interests. With simulation and analytical approaches, they have made an effort to explain and solve the warehousing problems. Efficient operation decisions in warehouse lead to improving the efficiency of the warehousing system in that the operational problems are narrow and short term by comparison.

Two control problems of warehousing operations are batching of orders and order picking or sequencing. Batching is a basic method for reducing the mean travel time per order and increasing the throughput. By forming a batch efficiently, the resource of people, machine and time can be utilized fully. Many batching algorithms or heuristics have been presented in the literature for minimizing travel time. Recently, Rosenwein (1996) presented and compared various heuristics for order batching with multiple aisles warehousing system in order to measure the proximity of orders. In most parts, the focus was on the warehouse efficiency, in other words, a batch that gives the minimal travel time or produces the maximum throughput. Also, picking items of orders within a batch is a classical problem. This problem can be represented by the traveling salesman problem (TSP) in the warehouse. Sequencing the visit of storage locations to be visited efficiently can get rid of unnecessary travel time, in turn, improve the system throughput.

The problem of finding the optimal solution among a finite number of solutions often arises in real life. Unfortunately, this kind of problem is extremely difficult to solve. In most applications the number of alternatives is huge, and only its small portion is considered. Furthermore, these problems often have uncertain structure that can be utilized in identifying the optimal solution. As a result, heuristic algorithms are used. Also, there is a drawback of local optimal which gives worse performance than the global optimum in these algorithms. In this research, the choice of the estimation of the optimal solution when random search methods are applied to solve discrete stochastic optimization problems is discussed. At the present time, such optimization methods usually estimate the optimal solution using either the feasible solution the method is currently exploring or the feasible solution visited most often so far by the method.

In Chapter 3, these two classical control problems are investigated with respect to considerations motivated by supply chain management. At the time of forming a batch, by considering the order holding time in the system, a joint order batching and picking algorithm based on the first-come, first served based rule is presented. Its results are compared with the optimal solution that is derived from solving two mathematical problems of bin packing problem (BPP) and traveling salesman problem (TSP). In Section 3.4, both order picking factor and order holding factor are used in order to reflect the throughput of the system and response of orders in the proposed algorithm.

Furthermore, some simple batching policies for considering the order holding time in the system are presented in Chapter 4. They may result in meeting its customers' requirements for larger numbers of smaller orders and rapid turnaround. Its performance measure of order response time that consists of order picking time and order holding time is evaluated and compared.

Finally, in Chapter 5, a recently proposed random search method called the nested partition (NP) method is described and applied to the supply chain problem. This method tries to seek an optimal solution to obtain increasingly more precise estimates of the objective function values at the different points in the feasible region. At any given time, the feasible solution that has the best estimated objective function is used as the estimate of the optimal solution. The advantages of using this approach for estimating the optimal solution is discussed and numerical results showing that modifying an existing random search method to use this approach for estimating the optimal solution is shown to yield improved performance.

In a rapidly changing business environment, far greater demands are on order to provide products and service quicker. Within the warehouse, order batching and order picking function is commonly considered as one of the most critical functions in a supply chain (Tompkins and Smith, 1998). Through the supply chain, there are some absolutely critical and predictive questions the warehouse system should accurately and quickly answer; when will specific orders really completed and what is the best schedule that can be executed now? Based upon theses questions, current research can provide the starting point of research that study on the efficiency of the warehousing system and response of customer service in supply chain, and the scope of interest can be extended throughout the supply chain.

The objective of this research effort is to investigate and seek solutions for the performance improvement associated with the warehouse efficiency and order response. With today's emphasis on meeting customer requirements and improving service, getting the right product to the right customer at the right time has become a standard. Time is of the utmost importance for high volume, delivery-sensitive environments. Speed and accuracy of transactions, information flow, and distribution processes are essential for industries that deal with short shelf -life and product life cycles (Dilger 1999).

CHAPTER 2. LITERATURE REVIEW

2.1 Introduction

The items stored and retrieved to serve the customer orders distinguish warehouses or distribution centers. The diversity of the stock-keeping unit (SKU) in the warehouse and the volume in which the items are requested can be factors affecting the warehousing system. There are many things to be taken into account for the warehousing systems to be operated internally and externally, and there are several factors to affect on the warehousing systems. Although it may seem that the only function of a warehouse is warehousing – that is, the temporary storage of goods – warehouses also perform many other functions, such as, receiving, storage, packing, and shipping.

Several researchers introduced the design, planning and control of warehousing systems as new research topics (Graves *et al.* 1977, Hausman, et al., 1976, Schwarz et al, 1978). Since then, the operation of warehousing systems has received considerable interest in the literature. This is because the 1970's was the era that interest of warehousing management shifted from productivity improvement of mass production to inventory reduction.

Several authors presented a literature review and made a survey on the warehouses or distribution centers. Matson *et al.* (1982) performed a general survey of material handling research with a range of topics covering robotics, conveyor theory, transfer lines, flexible manufacturing systems (FMS) and equipment selection, in addition to models applicable to warehousing. Ashayeri *et al.* (1985) addressed warehousing specially, and they gathered information about analytical methods and simulation approaches for the problems of warehouse design optimization. Cormier *et al.* (1992) reviewed the recent literature concerned with the optimization of the warehouse design and operations. They mentioned three warehousing models: throughput capacity models, storage capacity models, and warehouse design models. This research is closely related to the throughput capacity model. Throughput capacity models focus on the picking policies, batching policies, storage, and assignment policies. Jeroen (1999) presented a literature survey on methods and techniques for the planning and control of warehousing systems even if they are not easy to solve. The author focused on the storage location assignment problem at the tactical level, and as control issues the author mentioned three problems of batching of orders, routing and dwell point positioning. He pointed out many batching heuristics have been presented in the literature for minimizing travel time, and most heuristics used seed order and proximity batching method.

2.2 Order Batching (OB)

Batching is a popular and efficient strategy for reducing the mean travel time per order and thus improving the system performance. A batch is a set of orders that is picked in a single tour. The total items of the orders in the batch may not exceed the storage capacity of the order-picking vehicle. Furthermore, we may maximize the system throughput by establishing large batches with orders at nearby pick locations. However, large batches will give rise to large response times. Moreover, only selecting orders at nearby pick locations might excessively delay orders at the far end of the aisle in the warehouse.

Vinod (1969) presented two integer programming (IP) formulations for the batching problem; one formation has a linear objective function, and the other a quadratic objective function. The author used the string property for proving to be necessary for the minimum of within group's sum of squares, and generalized the property to higher dimensional case. Armstrong *et al.* (1979) considered proximity batching with fixed batch sizes and presented an integer programming (IP) model. They used dummy orders for a full batching and took into account a zone picking operation utilized by a liquor distribution center. Kusiak (1986) presented integer and quadratic programming formulations of the batching problem. An eigenvector-based approach is described for finding an approximate solution.

Several authors presented heuristic methods for batching orders. These methods basically follow three steps: 1) a method of initiating batches, 2) a method of allocating orders to batches, 3) a stopping rule to determine when a batch has been completed.

Elsayed (1981) measured proximity of the orders by considering the number of common locations of the items. Elsayed (1983) presents four heuristic algorithms for handling orders in automatic warehousing systems. The algorithms select the orders that will be handled in one tour in order to minimize the total distance traveled by the S/R machine or picking vehicle. The performance of any of four algorithms described depends on the structure of the orders and the capacity of the S/R machine. Gibson and Sharp (1992) presented an order batching procedures through computer simulation to compare two new procedures for batching orders in order retrieval system against a baseline procedure. Bartholdi and Platzman (1988) presented a method for batching orders based on spacefilling curves (SFC), which is a continuous mapping from a point on the unit circle onto the unit square. A rectangular range is identified for each order and is summarized as scalar by a 4-dimensional SFC transformation.

Elsayed and Lee (1996) described batching procedures in a single aisle facility with the objective of minimizing tardiness. They investigated batching effect on the automated and storage retrieval system (AS/RS) where due date is specified for each retrieval order. Several rules are developed for sequencing and batching orders to tours such that the total tardiness of retrievals per group of orders is minimized.

Rosenwein (1996) presented and compared various heuristics for order batching in multiple aisles warehousing system. The author used a couple of metrics (the minimum additional aisle (MAA) and center of gravity (COG) distance metric) that approximates the relative closeness of a pair of orders. The former measures the number of extra aisles that must be visited when an order is added to a batch. The latter evaluates the average of the aisle numbers and batches the orders for which this average is nearest. Elsayed *et al.* (1989) compared four heuristics, under the assumptions that the total number of orders is normally distributed, while the total number of items in an order and quantity of each item are uniformly distributed random variables. They presented that the best algorithm involves first sorting each order as large or small with respect to a predetermined fraction of vehicle capacity.

2.3 Order picking (OP)

Jarvis and McDowell (1991) concentrate on the product layout design problem for a single depot configuration for a discrete order picking system in which a single picker moves parallel aisles (a rectangular warehouse). Since they assume that once a picker enters a vertical aisle he must traverse the entire aisle. They determine the best product layout for two special cases of a single depot configuration – a depot at the beginning of the aisle or one in the center.

Chew and Tang (1999) also examine a discrete order picking system in a rectangular warehouse. Their focus is on examining the effects of a batch sizes in a real time system. Their probability model fixes the number of items per order (the batch size), and then develops a queuing model where incoming orders queue and batched before picking.

Two most popular methods of gathering orders in a warehouse are order picking (OP) and zone picking (ZP) for an order picking procedure (Heragu, 1997). In order picking an operator or order-picking vehicle is responsible for picking all the items in a batch or tour, whereas in zone picking an operator assigned to a zone (more than an aisle) is responsible for picking all the items of the orders in a batch. In zone picking the order picker has to travel within a zone. The objective of order picking is to sequence the locations of the items to be visited in order to minimize the travel time of an order-picking vehicle. It can get rid of the unnecessary time in the order picking process. In this thesis, only the order picking problem is considered.

Gudehus (1973) described the widely used band heuristic. This heuristic divides the rack of the warehouse into two horizontal bands or area. First, the locations on the lower band are visited on increasing x-coordinate; subsequently the locations on the upper band are visited on decreasing x-coordinate. The authors claim any even number of bands can be used. Bozer (1985) derived an analytical expression for the expression for the expected tour length of the band heuristic, as well as the optimal number of bands as functions of the number of picks.

Ratliff and Rosenthal (1983) developed a procedure for finding the picking tour that minimizes travel distance from order picking graph. In order to form a tour in a warehouse with the 50 aisles the authors utilize a well-known traveling salesman problem (TSP). Randolph (1993) evaluates strategies for routing a manual picker through a simple warehouse and derives equations that relate route length to warehouse attributes. The four routing strategies showed similar simulation results. Peterson (1997) evaluated the performance of five routing heuristics in comparison with the algorithm of Ratliff and Rosenthal (1983). They gave at best average 10 % over optimal for various warehouse factors, e.g., shapes, locations of the I/O station and picking size.

Goetschaleks *et al.* (1988) presented an efficient algorithm for order picking in a warehouse with non-negligible aisle width. In wide aisles two way travel is possible, traffic can turn and pass and it is possible to use forklift truck for lifting. They suggested more than 30 % of savings are possible by picking both sides of the aisle rather than picking one side first and returning to pick the other side. After that, Goetschaleks *et al.* (1989) considered the problem of determining the optimal stop positions of an order picking vehicle in an aisle when the order picker is allowed to perform multiple picks per stop. They suggested an efficient dynamic programming algorithm for the instance that the travel time of the order picker is measured with the rectilinear metric.

Comier (1987) described an order picking problem in which the objective is to minimize the total weighted tardiness incurred when items are not delivered to the pickup/dropoff (P/D) point before their respective due dates. A dynamic programming based heuristic is presented where each job is a tour of the order picking vehicle, and each tour consists of several items each having different due dates but a common completion time.

Akl *et al.* (1978) presented a fast algorithm for finding the convex hull. Boer *et al.* (1990) presented a heuristic that uses the convex hull of the rack locations in the interior of the convex hull. Also they proposed an improved version of the band heuristic that blocks out a central portion of the rack. Golden *et al.* (1985) discussed the property that every TSP whose travel times are measured by the Euclidean metric has an optimal solution in which the nodes on the boundary of the convex hull are visited in the same sequence as if the boundary of the convex hull itself is traced

Kanet *et al.* (1986) proposed a mixed zero-one non-linear programming formulation for the problem of selecting from alternate picking locations so a to minimize some combination of breakdown cost as well as fixed and variable picking costs. The authors assumed the variable costs is a function of travel time while the fixed picking cost depends on such things as pallet loading and unloading times.

2.4 Nested partitioning method

Shi and Olafsson (2000) proposed a new randomized optimization framework called the Nested Partitions (NP) method for solving global optimization problems. The NP method systematically partitions the feasible regions and concentrates the search in regions that are the most promising. The most promising region in each iteration is selected based on information obtained from random sampling of the entire feasible region and local search. They developed the method for discrete problems and then extended to continuous global optimization.

Shi and Olafsson (1999) provided several methods of both partitioning and sampling for the traveling salesman problem. The efficiency of the NP method depends on good solutions tending to be clustered together, in turn, depends on how to partition and randomly sample an arbitrary feasible region. They incorporated the 2-opt exchange local search heuristic into the promising index with good results and showed that the NP method was a powerful alternative in solving combinatorial problems.

Olafsson and Shi (2000) addressed a scheduling problem of simultaneously allocating flexible resources, and sequencing jobs in cellular manufacturing systems where the cells are configured in parallel. They applied the nested partitions (NP) method, reformulated the problem so that the NP method may take advantage of the special structure of the problem in both the global and local search for efficiency and developed a new sampling algorithm that can be used obtain good feasible schedules.

Shi *et al* (2001) applied the nested partitions (NP) method to the product design problem and presented a new optimization framework for this problem. They incorporated several known heuristics into this framework to speed its convergence and showed several examples using the NP method.

2.5 Supply chain management (SCM) and warehousing

Current trends in warehousing and distribution logistics are motivated by recent development in supply chain management (SCM). The supply chain management (SCM) pursues a demand-driven organization of the supply chain with small inventories and reliable short response times throughout the supply chain. All deliveries are driven by the sales downward in the supply chain. Such an organization requires a close cooperation among the companies in the supply chain and the immediate feedback of sales data. Nowadays, information technology (IT) enables these developments through electronic data interchange

(EDI) and software systems such as the MRP-based enterprise resource planning (ERP) systems and warehousing management systems (WMS).

The new market needs the operation of warehouses to adapt rapidly. The warehousing systems demand an increased productivity. On the other hand, the rapidly changing market requires a rapid customer response. Hence, the tradeoff between efficiency of the warehousing system and response of the customer orders must be considered. However, few papers have been published that deal with this tradeoff.

Lee *et al.* (1995) considered the problem of minimizing earliness and tardiness penalties when all storage and retrieval requests have a common due date. Jaikumar *et al.* (1990) addressed the problem of the relocating pallets with a high expectancy of item retrieval in an AS/RS to locations closer to the I/O station during off-peak times. They presented an efficient algorithm that minimizes the number of relocations in order to meet the expected throughput. Egbelu *et al.* (1999) presented an algorithm to relocate the unit loads in order to process the retrieval orders during the idle time. They gave a reduced order response time when the proposed procedure is applied to the sample problem in the simulation.

In this research, when the orders from suppliers and customers are processed a problem that takes into account the picking time and holding time of the orders simultaneously in order to improve the order response time is considered. Past literatures did not consider the response time of the orders. They typically focused on only the efficiency or throughput of the warehouse. Depending upon the warehouse manager's decision, the different weights on the picking time and holding time are given in order to reflect the order holding time. This approach can give an approximate solution that attacks a tradeoff between the efficiency of the warehouse and response of the customer orders.

CHAPTER 3. JOINT ORDER BATCHING AND PICKING

In a typical warehouse or a distribution center, products or stock-keeping units (SKU) need be drawn from specified storage locations of a rack. This order picking (OP) process, which is the process of retrieving products or items that have been requested in an order, is driven by customer orders. Each order consists of a number of line items and each line item represents one stock-keeping unit that has to be drawn and shipped to the customer. The order picking tries to determine the sequence in which product locations are visited so that total travel distance is minimized (Cormier et al., 1992). Because the order picking operation consumes as much as 50% of all labor activities in a warehouse, it is not surprising that order picking has received much interest in the literature (Heragu, 1997).

Also, in order to improve the warehouse efficiency, an additional order batching (OB) method by which the orders are assigned to order picker should be considered (Ruben, 1999). Batching is an efficient planning strategy for reducing the mean travel time per order. A batch consists of a set of orders that are picked in a single tour. The orders in a batch cannot exceed the capacity of order picking vehicle. Furthermore, it may maximize the system throughput by establishing large batches with orders at nearby picking locations. However, large batches will give rise to large response time, in turn, customers may incur a long waiting time. Moreover, only selecting orders at nearby pick locations might excessively delay orders at the far end of the warehouse.

Nowadays, one of the current trends in warehousing and distribution logistics is supply chain management (SCM). Supply chain is the channel among suppliers, manufacturers, distributors, and customers. Companies, both manufacturing and service, are creators of value, not simply makers of products. Supply chain management focuses on globalization and information management tools that integrate procurement, manufacturing, operations, warehousing and logistics from raw materials to customer satisfaction. The SCM pursues a demand-driven organization of the supply chain with small inventories and reliable short response time throughout the supply chain. All deliveries are driven by the sales downward in the supply chain. Such an organization requires a close cooperation among the supply chain.

Today, warehouse managers demand an increased productivity in the system. In the past, the problems about the design, planning and control of warehousing systems as new research field have been introduced and received considerable interest since 1970s (Gudehus, 1973; Graves et al., 1977; Hausman et al., 1976; Schwarz et al., 1978). These problems have been modeled as mathematical optimization problems or simulation models based on specific performance metrics, i.e., cost, time or throughput. In order to increase an efficiency of warehouse systems, they do not explicitly take into account customer's point of view.

On the other hand, the customers or retailers demand an improved order response time. They might want the prompt order processing or quick response for their orders. Furthermore, due to the development of information technology (IT) such as an electronic data interchange (EDI), lead times can be decreased by reducing some portion of the response time linked to order processing, i.e., picking time and holding time in the queue. Furthermore, many companies are trying to find suppliers with shorter order processing time.

Accordingly, the trade-off between warehouse efficiency and order urgency must be observed. This trade-off may be achieved by forming appropriate batches taking into account an order picking time in terms of warehouse efficiency and order holding time in terms of order promptness at the same time in supply chain management (SCM). Forming large batches will cause a large response time, and few batches will give rise to inefficient warehouse management. So the more weight is given on the order picking time as a production time, and the relatively less weight is given on the order picking time as a non-production time. This concurrent approach considers an order picking (OP) problem and an order batching (OB) problem jointly.

3. 1 The Layout of a rack and computation of a bin location

An automated or semi-automated warehouse consists of a storage rack on each side of an aisle. Each rack has a number of storage bins, and these storage bins are used for storing various items or products in the warehouse. A picker or vehicle can serve both sides of an aisle for retrieval or storage at the same time. An aisle enables a picker or vehicle to retrieve a list of items from their respective locations of a rack.

Figure 3.1 shows the typical layout of an aisle and a rack in a warehouse. The rack contains a large number of storage bins and also shows that a picker or retrieval vehicle can travel along the aisle vertically and horizontally at the same time. Figure 3.2 shows the numbering scheme of bins in a rack. In general, the rack has a number of bins described by the number of bin columns and the number of bin rows. Thus a rack of N bin column and M bin rows has the number of MN bins on each rack. If two racks face each other on an aisle, bin location on one rack can have exactly the same bin location on the other rack.

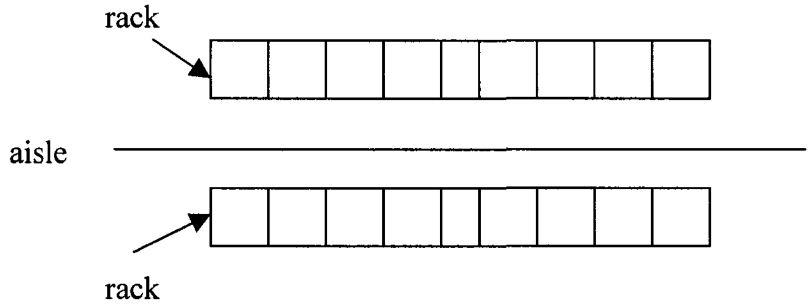

Figure 3.1 Top view of an AS/RS aisle

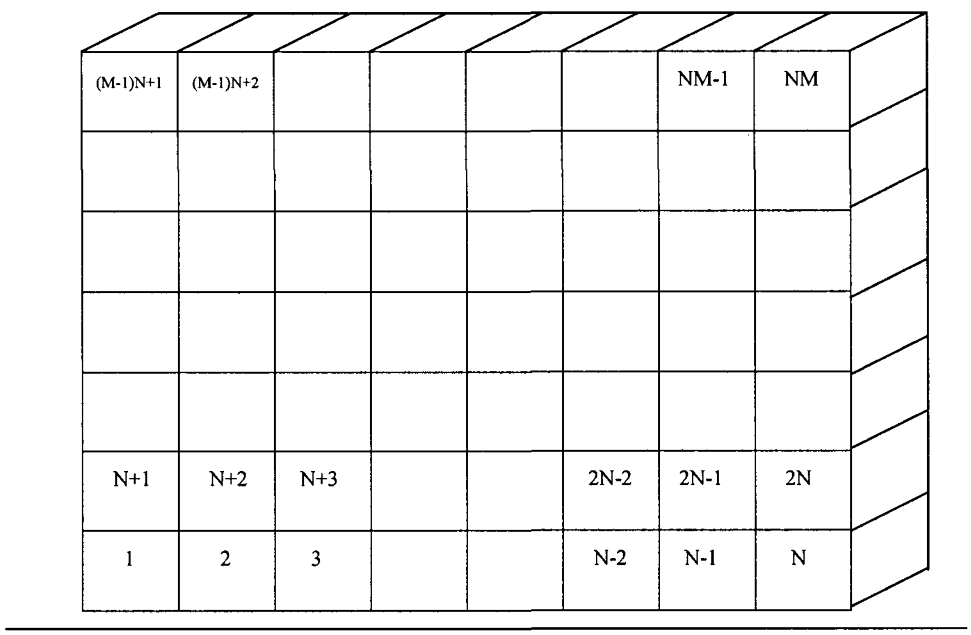

Figure 3.2 Side view of a rack showing bins and a bin numbering scheme

(Courtesy of Egbelu, 1991)

If the rack begins L feet away from the reference point along the horizontal axis and the bottom of the first row of bins is H feet above the reference point, the location of the center of a bin with respect to the reference point can be calculated as follows. The center of a bin is assumed to be located at the bottom of a bin (Egbelu, 1991).

$$(X_j, Z_j) = \left\{ \left(j - N\left[\frac{j}{N}\right]^- - \frac{1}{2} \right)\frac{w}{12} + L + X_r \, , \left(\left[\frac{j}{N}\right]^+ - 1 \right)\frac{h}{12} + H + Z_r \right\}$$

where $\left[\frac{j}{N}\right]^-$ = greatest integer less than $\frac{j}{N}$, and

$\left[\frac{j}{N}\right]^+$ = smallest integer greater than or equal to $\frac{j}{N}$,

j = index for the bin whose center coordinates are to be calculated with respect to the reference point

N = length of a rack in bins

M = height of a rack in bins

w = width of a bin in inches

h = height of a bin in inches

(X_r, Z_r) = coordinates of the reference point in an aisle

(X_j, Z_j) = coordinates of the bin j in an aisle

Based upon this information, the unique location of each bin can be calculated and utilized in calculation of the picking time in traveling salesman problem (TSP). As mentioned before, a tour or batch consists of many orders to be processed, and each order can have multiple items.

Order picking in conventional warehouse environments involves determining a sequence in which to visit the unique locations where each item of an order in a tour or batch is stored, and thus is often modeled as a traveling salesman problem (TSP). This TSP can be stated as follow. A picker or retrieval vehicle retrieves products or lineitems listed in the orders of a tour or a batch from their respective locations. The items in an order are called lineitems or picks. Beginning from a pickup/drop-off (P/D) point or input/output (I/O) point location, the picker retrieves the lineitems listed in the order in a particular sequence and then places the retrieved orders in a batch back at the I/O point. To increase the throughput rate of the warehouse system, managers try to find a sequence that minimizes the time required to pick the items of the orders in a tour or batch (Heragu, 1997; Goetschalckx *et al.*, 1988; Bozer *et al.*, 1990; Heragu *et al.*, 1994).

3. 2 BPP and TSP

Most complicated warehousing and logistics problems, for example, the bin- packing problem (BPP) and traveling salesman problem (TSP) can be formulated as classical combinatorial optimization problem. Both BPP and TSP are rather easy to state and formulate, but hard to solve. They can be applied to various real world problems. Because they are NP-hard it is unlikely that polynomial time algorithms will be developed for their optimal solution.

3.2.1 Bin packing problem (BPP)

The purpose of the bin packing problem (BPP) is to find the minimum number of batches or bins. The problem is to assign each order to a bin such that the sum of the item of the order in a bin does not exceed the capacity of the vehicle, while minimizing the number of bins used. The bin packing problem (BPP) can be stated as follow: given n items and n batches or vehicles, with

w_j = weight or volume of a item j

(assume that the weight or volume of all the items is same)

C = vehicle capacity in each batch

assign each item to each batch so that the total weight or volume of the items in each batch does not exceed C and the number of batches used is a minimum. A mathematical model can be described as follows:

Minimize $Z = \sum_{i=1}^{n} y_i$

Subject to $\sum_{j=1}^{n} w_j X_{ij} \leq Cy_i \qquad i \in N = \{1,\ldots,n\}$

$\sum_{i=1}^{n} X_{ij} \qquad j \in N = \{1,\ldots,n\}$

$y_i = 0 \text{ or } 1 \qquad i \in N = \{1,\ldots,n\}$

$X_{ij} = 0 \text{ or } 1 \qquad i, j \in N = \{1,\ldots,n\}$

where

$$y_i = \begin{cases} 1, & \text{if vehicle of batch } i \text{ is used} \\ 0, & \text{otherwise} \end{cases}$$

$$X_{ij} = \begin{cases} 1, & \text{if lineitem } j \text{ is assigned to batch } i \\ 0, & \text{otherwise} \end{cases}$$

In this research, the role of bin packing problem is to determine the assignment of orders to batches satisfying the capacity constraint. It gives the minimum number of batches or tours, but it allows a good number of optimal solutions if the number of orders is huge and the size of the orders is small. So the heuristic methods such as the first-fit (FF) or first-fit decreasing (FFD) in order to solve the bin packing problem (BPP) should be used due to the time or computational restriction.

3.2.2 Traveling salesman problem (TSP)

The traveling salesman problem (TSP) can be stated as follows. Given a set of the n storage locations that corresponds to the items to be retrieved and the distance between each pair of these locations, find the shortest path such that the picking operation begins and ends at the same location. Also the order picking needs visit every storage locations once and only once. Its mathematical model with some notations can be described as follows.

k, l : index for locations to be visited in a tour

h : vehicle's horizontal speed

v : vehicle's vertical speed

(x_k, y_k) : coordinates of location k

u_k, u_l : arbitrary real numbers for location k and l

The role of the TSP is to find the sequence of the storage locations to be visited in a tour once the batches are formed. It is known that the TSP is NP-complete (Lawler, 1985). Although optimal algorithms for the TSP exist, they are suited for solving small-size problems only. Even medium-sized problems give computational burden to the computer system. Moreover, since the order picking operation is a planning problem that is to be addressed on a frequent time window basis (e.g., hourly) a quick solution is needed. Hence, the heuristic methods are to be used to solve the traveling salesman problem (TSP) or order picking problem.

Minimize $$Z = \sum_{k=1}^{n} \sum_{\substack{l=1 \\ k \neq l}}^{n} d_{kl} w_{kl}$$

Subject to
$$\sum_{\substack{k=1 \\ k \neq l}}^{n} w_{kl} = 1 \qquad \text{for each } l$$

$$\sum_{\substack{l=1 \\ l \neq k}}^{n} w_{kl} = 1 \qquad \text{for each } k$$

$$u_k - u_l + n w_{kl} \leq n - 1 \qquad \text{for } 2 \leq k \neq l \leq n$$

$$d_{kl} = \max\left[\frac{|x_k - x_l|}{h}, \frac{|y_k - y_l|}{v}\right] \qquad \text{for } 1 \leq k \neq l \leq n$$

$$w_{ij} = 0 \text{ or } 1 \qquad \text{for each } k,\ l,\ k \neq l$$

u_k is arbitrary real numbers $\qquad$ for each k

3.3 Problem formulation

Consider the following warehouse environment. Orders from customers or from a factory floor arrive and specify a set of products and quantities for each order. The problem to be formulated is to determine how to batch or group the orders listed with the order requirements and to minimize the order picking time and order holding time within a batch. This batch should take into account the warehouse efficiency and order response or promptness. Although there are other considerations in choosing which orders to batch or group to maximize the system utilization and to minimize the order response time, this research considers the problem of minimizing the time of picking the orders and of holding the orders within a batch.

Some notations and assumptions for the problem should be defined.

Notations

k, l : index for bin locations to be visited in a tour or batch

h, v : horizontal and vertical speed of a vehicle

(x_k, y_k) : coordinates of bin location k

u_k, u_l : arbitrary real numbers for location k and l

B : index for batch

$$y_{kl}^{B} = \begin{cases} 1, & \text{if location } k \text{ is visited immediately after } l \text{ in a batch } B \\ 0, & \text{otherwise} \end{cases}$$

a_k : arrival time of order k

h_j^B : order holding time of job j in a batch B

$$X_k^B = \begin{cases} 1, \text{ if order } k \text{ is assigned to a batch } B \\ 0, \text{ otherwise} \end{cases}$$

C : vehicle capacity

w_k : number of items in order k

ϖ_1 : picking factor for order picking, $0 \leq \varpi_1 \leq 1$

ϖ_2 : holding factor for order holding, $0 \leq \varpi_2 \leq 1$

<u>Assumptions:</u>

1. A picker or retrieval vehicle can travel along the aisle vertically and horizontally at the same time, i.e., it has a Tchebyshev movement
2. Items stored in a rack are known with certainty
3. The bin number occupied by the product is known
4. picking time for the picker to retrieve the item from a storage location is assumed to be constant and known
5. A picker or retrieval vehicle has a dedicated aisle
6. There is one input/output (I/O) point before each aisle
7. An order picking tour starts and ends at the I/O point in an aisle
8. The number of items in each order is assumed to be less than the capacity of the vehicle
9. It is assumed that all orders in a tour or batch are completed, in other words, an order cannot be split

The formulation of this problem can then be expressed as follow. The below model is based upon the formulations of the traveling salesman problem (TSP) and the bin packing problem (BPP). As mentioned above, in order to minimize mean travel time per order an efficient picking method for orders in a tour should be considered. The TSP can figure out the problem. Also, in order to increase efficiency of the warehouse system, efficient batching methods are needed. However, because the bin packing problem tries to only assign the orders to a specific batch, it does not consider the sequence of orders in queue. For just minimal number of bins or batches the BPP does not take into account the waiting time of orders. In other words, some orders may be grouped into the last batch even if they arrived early in the system because they contribute to minimize the number of batches or tours which is objective function of the bin packing problem (BPP). Hence, a variation should be considered.

The distance used in this research is assumed to be the Tchebychev metric instead of the usual Euclidean or rectilinear metric. The Tchebychev metric allows movement in both directions simultaneously. Therefore, the time it takes to travel from a point with one specific coordinates to another coordinates depends on the horizontal and vertical distances between the two points as well as the horizontal and vertical speeds of the two independent motors.

The decision variable y_{kl}^{B} takes on a binary value of 0 or 1 depending on whether or not the location k is visited immediately after the location l in an order picking sequence of a batch B.

Minimize $Z = \varpi_1 \sum_{k=1}^{n}\sum_{l=1}^{n} d_{kl}^B y_{kl}^B + \varpi_2 \sum_{k=1}^{n} h_k^B$

Subject to

$$\sum_{\substack{k=1 \\ k\neq l}}^{n} y_{kl}^B = 1 \qquad \forall l, \forall B \qquad (1)$$

$$\sum_{\substack{l=1 \\ l\neq k}}^{n} y_{kl}^B = 1 \qquad \forall k, \forall B \qquad (2)$$

$$u_k - u_l + n \cdot y_{kl}^B \leq n-1 \qquad \text{for } 2 \leq k \neq l \leq n,\ \forall B \qquad (3)$$

$$d_{kl}^B = \max\left[\frac{|x_k - x_l|}{h}, \frac{|y_k - y_l|}{v}\right] \qquad \text{for } 1 \leq k \neq l \leq n,\ \forall B \qquad (4)$$

$$h_j^B = \max_{\forall k}\left\{X_k^B \cdot a_k\right\} - a_j \qquad \forall B \qquad (5)$$

$$\sum_{k=1}^{n} w_k X_k^B \leq C \qquad \forall B \qquad (6)$$

$$\sum_{\substack{k=1 \\ k\neq l}}^{n}\sum_{\substack{l=1 \\ l\neq k}}^{n} y_{kl}^B = \sum_{k=1}^{n} w_k X_k^B \qquad \forall B \qquad (7)$$

$$\sum_{k=1}^{n} X_k^B = 1 \qquad \forall B \qquad (8)$$

If it is assumed that *n* orders are arrived, the number of batches is *n* at the worst case, and the minimum number of batches is n/C. The objective function minimizes the total picking time of tours or batches and holding time of orders of the tour. ϖ_1 and ϖ_2 represent picking factor and holding factor, respectively. They give weights on each component. If $\varpi_1 = 1$ and $\varpi_2 = 0$ then the above formulation minimizes only the picking time of a batch. On the contrary, if $\varpi_1 = 0$ and $\varpi_2 = 1$ then it tries to minimize the order holding time of a batch. By giving proper values to each factor depending upon the warehouse system, e.g., $\varpi_1 = 0.7$ and $\varpi_2 = 0.3$, the model can reflect the efficiency of the warehousing system and the response of customer orders.

The first two constraints (1) and (2) represent that each arc in a tour or batch has exactly two endpoints, i.e., one on either side of it. The constraint (3) is a sub-tour elimination constraint or no sub-tour allowed in each batch. In other words, the optimal solution must only have one tour connecting all the points in one batch. The fourth constraint represents that the distance metric is the Tchebyshev metric. It has two independent motors for horizontal and vertical direction. The constraint (5) computes order holding time within a batch. The constraint (6) specifies that because the picker or retrieval vehicle has its capacity it cannot exceed its maximum items to be retrieved. The constraint (7) represents the relationship of the number of locations visited in a tour or batch and the sum of items in the orders of a batch. Finally, The constraint (8) ensures one order should be included in a batch *B*.

It is known that both the traveling salesman problem (TSP) and bin-packing problem (BPP) are NP-complete which is the complexity class of decision problems for which answers can be checked for correctness by an algorithm whose run time is polynomial in the size of the input (Lawler, 1985). Although there are optimal algorithms for the TSP and BPP, they are capable of solving only small size problems. Moreover, because both order picking problem and order batching problem are operational problems that are to be addressed on a frequent basis and quick solutions are required, it is clear that heuristics instead of an optimal algorithm are to be used to solve the joint problem. (Gibson *et al.* 1992; Hwang *et al.* 1988; Pan *et al.* 1995)

3.4 The proposed JBP algorithm

The objective of this chapter is to efficiently batch the orders and pick the items within the orders from the warehouse system such that the sum of the order response time is minimized. This problem can be considered to find the optimal between-batch time. It takes into account the order picking time for warehouse efficiency and order holding time for quick response. By selecting the efficient between-batch time the warehouse efficiency of deducing the picking time and customer satisfaction of reducing the order waiting time from the supply chain perspective can be improved.

The purpose of the proposed Joint Batching Problem(JBP) algorithm below is to find the between-batch time that meets the above objective. Given many orders to be processed, a value of an optimal between-batch time is selected. It reflects the warehouse efficiency and customer urgency. The following additional assumptions are added for this algorithm.

1) Orders are served as first come first served (FCFS) basis
2) a heuristic method for solving the traveling salesman problem (TSP) is used

The reason for using the FCFS rule is that it is easy to implement and reflects the order response well. Below are notations for the proposed algorithm.

t : time for between-batch

B : index for a batch

k : index for an order

∇t : the increment of between-batch time

a_k : the arrival time of an order k

a_L^B : the last order arrival time in a batch B

P_t^B : order picking time for a batch B with between-batch time t

H_t^B : order holding time for a batch B with between-batch time t

h_k^B : order holding time of job k in a batch B

R_t^B : order response time for a batch B with between-batch time t

With these notations at hand, we can now describe the generic version of the JBP algorithm.

Step 0. Initialize the parameters.

Set ∇t to be an appropriate value, i.e., a minute or 5 minutes. Give ϖ_1 and ϖ_2 a specific value, respectively. Put the minimum and the maximum batch time for t. Too small value for between-batch time will deteriorate the efficiency of a warehouse system and too large between-batch time will not give any benefit to order batching due to the vehicle capacity.

Step 1. Find a batch based on between-batch time t and the vehicle capacity C.

Set $P_t^B = H_t^B = R_t^B = 0$. Group the orders within between-batch into one batch B. If the total number of items of all orders in the batch B exceeds the vehicle's capacity, discard the most recent orders which violate the condition of the capacity of a vehicle from the batch B backward. Let this specific batch be B'.

Step 2. Calculate the order picking time, $P_t^{B'}$, of the batch B'.

The picking time, $P_t^{B'}$, is calculated by solving the traveling salesman problem (TSP). The calculation of total travel time or picking time is called whenever the new batch or one tour in traveling salesman problem is formed.

Step 3. Calculate the batch holding time, $H_t^{B'}$, of the batch B'.

It is the sum of an individual order holding time, $h_k^{B'}$. Because the arrival time of an order is known and the batch B' is already formed in Step 2 the individual order holding time can be calculated as follows.

$$h_k^{B'} = a_L^{B'} - a_k$$

Step 4. Calculate the order response time, $R_t^{B'}$, of the batch B'.

The order response time is the sum of order picking time, $P_t^{B'}$, from Step 2 and order holding time, $H_t^{B'}$, from Step 3. This value is kept to get the total order response time within a between-batch time t. A total order response time within a between-batch t is a performance measure in this research. Set R_t^B = $R_t^B + R_t^{B'}$.

Step 5. If there are still orders to be processed in an order list, then delete the orders in the current batch B' and go to Step 1. Otherwise, Record the total order response time of R_t^B obtained from Step 4 and go to Step 6.

Step 6. Change the between-batch time $t = t + \nabla t$. Also, the initialize the related values and go to Step 1. If the t is reached to the maximum between-batch time, stop. Find the between-batch time with the minimum order response time, R_t^B, obtained from Step 5.

At Step 2, the nearest neighbor algorithm was used for solving the TSP problem. This is a simple heuristic method for constructing a tour:

Step 1. Start with the I/O point of the rack

Step 2. If the current tour doesn't include the all the pick locations of a batch, pick the next location to visit from among the ones NOT visited yet, and of all such locations pick the one that is closest

Step 3. Stop when the tour contains all the locations to be visited

This algorithm allows us to find answers fairly quickly.

3.5 Numerical example

An example problem is provided to illustrate the joint order batching and picking problem. The storage structure under consideration has two storage racks with 1000 bins (50 columns and 20 rows) in each rack. The dimension of a bin is 4 foot length and 4 foot height.

The units have equal sizes and weights while the picker or retrieval machine has the same horizontal and vertical speed of 50 ft/min. In this problem, the picking factor ϖ_1 and holding factor ϖ_2 are assumed to be 1. In other words, the same weight is given to those parameters. Also, the input or start point is located 10 ft away from the beginning of an aisle. It is also a drop-off point where the picker discharges the items from the tour or batch. There is no acceleration for the retrieval machine. For an item pickup or retrieval from the item location the constant pickup time of 0.25 min is assumed. It is assumed that order arrival rate is 30 per hour, and the number of items in each order has an uniform distribution of U(1,5). The vehicle capacity of 30 is assumed.

For a comparative purpose, the optimal method using the bin packing problem (BPP) and traveling salesman problem (TSP) is considered. For order batching, the BPP is considered. It tries to pack as many orders as possible within the range of the vehicle capacity. However, the disadvantage of choosing the newest-arrival order for the first batch can exist and cause the long order holding time, in turn, the long order response time. Another disadvantage is that BPP can result in multiple solutions. Because the BPP gives the minimum number of packs or batches for the problem, if there are many orders with 1 or 2 items, then it produce a lot of optimal solutions. However, it can be one method for measuring the warehouse efficiency.

Table 3.1 shows order data. Here, the order alias means another name of an order number for convenience.

Table 3.1 Raw order data

order number	order alias	order arrival time	number of itmes	order number	order alias	order arrival time	number of itmes
1	1	9:01	4	16	G	9:32	4
2	2	9:02	4	17	H	9:32	5
3	3	9:04	2	18	I	9:36	4
4	4	9:06	3	19	J	9:38	2
5	5	9:09	4	20	K	9:39	2
6	6	9:13	2	21	L	9:41	1
7	7	9:18	5	22	M	9:41	5
8	8	9:18	4	23	N	9:45	2
9	9	9:19	1	24	O	9:49	3
10	A	9:21	5	25	P	9:52	2
11	B	9:21	5	26	Q	9:53	1
12	C	9:23	4	27	R	9:54	2
13	D	9:25	3	28	S	9:55	3
14	E	9:25	5	29	T	9:56	5
15	F	9:26	3	30	U	9:59	3

Table 3.2 One of the optimal solutions from solving the BPP

Batch	Orders	Total itmes
1	1, 5, 6, 8, D, G, I, R, S	30
2	3, B, H, K, L, M, N, O, T	30
3	2, 4, 9, A, C, E, F, J, U	30
4	7, P, Q	8

Table 3.2 shows one of the optimal solutions from solving the bin packing problem (BPP) and traveling salesman problem (TSP) sequentially. The optimal solution gives 4 batches for the above problem. From the Table 3.1, the order picking time and order holding time can be obtained by solving their respective problems. This solution below is one of the multiple optimal solutions.

Table 3.3 shows the results of the aforementioned joint order picking and order holding problem for an optimal method. Here, PT, HT and RT represent the order picking time, the order holding time, and the order response time, respectively. As can be noticed, the value of order holding time (HT) under the optimal solution in Table 3.3 is larger than that of Table 3.4

Table 3.3 Result of an optimal method using BPP and TSP model

Batch	PT	HT	RT
1	59.3	234.0	293.3
2	42.7	298.0	340.7
3	48.3	231.0	279.3
4	14.1	69.0	83.1
Total	164.4	832.0	996.4

This is because the optimal solution does not consider the order arrival time of the customers and focus on the vehicle utilization of a warehouse. To use this information and reduce this holding time is one of our concerns. The reason for the large picking time (PT) in Table 3.3 is from the just one simulation output compared to the relatively small picking time (PT) in Table 3.4.

Table 3.4 Result of using the JBP algorithm

BT	NB	PT	HT	RT
5	9	114.0	51.0	165.0
6	8	113.2	67.0	180.2
7	7	109.8	85.0	194.8
8	7	97.3	102.0	199.3
9	6	93.1	123.0	216.1
10	5	88.7	133.0	221.7
11	5	89.5	127.0	216.5
12	5	91.2	116.0	207.2
13	5	91.4	132.0	223.4
14	5	101.5	166.0	267.5
15	5	97.3	178.0	275.3

Table 3.4 shows the result of using the JBP heuristic algorithm described above. Here, BT means the time for between-batch and NB means the number of batches calculated when the JBP algorithm was used. This is the result from one replication. Figure 3.3 shows the pictorial view of Table 3.6.

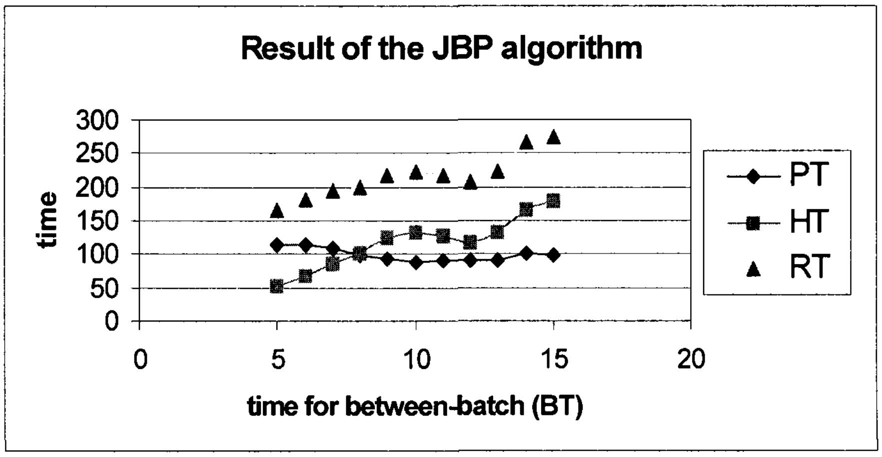

Figure 3.3 Graphical representation of Table 3.4

3.6 Effect of weights on order response time

As can be seen in Figure 3.3, the order response time (RT) gives the minimum value when the time for between-batch is 10 min. The order picking time (PT) shows a small variation compared to the variation of the order holding time. Also, as expected, the order holding time increases as the time for between-batch increases. This comes from the fact that the longer the between-batch time is, the larger the waiting time in queue is. To paraphrase, a large batch size gives the short order picking time to some extent, but does the long order holding time.

In the previous example, order picking factor of $\varpi_1 = 1$ and order holding factor of $\varpi_2 = 1$ were used. In other words, the same weights are given to the picking and holding time. From the above example, the order response time that is the sum of picking time and holding time seems to be dependent on an order holding time.

For $\varpi_1 = 1$ and $\varpi_2 = 0$, the mathematical model minimizes only the picking time of a batch. Also $\varpi_1 = 0$ and $\varpi_2 = 1$ minimizes only the order holding time of a batch. By giving proper values to each factor depending upon manager's experience, e.g., $\varpi_1 = 0.7$ and $\varpi_2 = 0.3$, the proposed JBP model can reflect truly the efficiency of the warehouse system by stressing on the picking time and the response or promptness of customer orders by reducing the waiting time in the system.

Table 3.5 Simulation result for $\varpi_1 = 1.0$ and $\varpi_2 = 1.0$

BT	PT	HT	RT
3	119.28	32.48	151.76
4	108.71	45.24	153.95
5	102.42	57.20	159.62
6	97.35	70.04	167.39
7	94.03	83.56	177.59
8	90.77	95.44	186.21
9	88.61	108.40	197.01
10	86.10	112.42	198.53
11	84.98	124.00	208.98
12	84.26	132.52	216.78
13	81.39	143.10	224.49
14	80.55	150.32	230.87
15	80.21	155.80	236.01
16	79.67	158.48	238.15
17	78.94	151.66	230.60

Another sample problem with 50 simulation replications is considered. In this problem the different weights of ϖ_1 and ϖ_2 are given compared to the previous problem with $\varpi_1 = 1$ and $\varpi_2 = 1$. For a demonstrative purpose, (1.0, 1.0) and (0.8, 0.2) cases of the (ϖ_1, ϖ_2) are shown. Table 3.5 and Table 3.6 show the simulation output with $\varpi_1 = 1.0$, $\varpi_2 = 1.0$ and $\varpi_1 = 0.8$, $\varpi_2 = 0.2$, respectively. They show the order response time of as a function of changes in the between-batch time. The between-batch time ranges from 3 minutes to 17 minutes. Figure 3.4 and Figure 3.5 show the pictorial representation of Table 3.5 and Table 3.6, respectively. When ϖ_1 and ϖ_2 have the same weight factor, the order response time (RT) is dominated by the order holding time (HT) rather than the order picking time (PT). If an order holding time increases, the order picking time also increases. However, in case of $\varpi_1 = 0.8$ and $\varpi_2 = 0.2$, the result is a little bit different, i.e., the order holding time does not dominate the order response time and give small variation. Approximately, at 10 min of the between-batch time, the simulation output gives the best result of 91.37 min. In other words, in order to complete the customer orders, the order response time of 91.37 min is needed. As a production time in terms of the warehouse efficiency, the order picking time contributes 80 % of order response time, and the order holding time in terms of supply chain management does 20 %.

Table 3.6 Simulation output for $\varpi_1 = 0.8$ and $\varpi_2 = 0.2$

BT	PT	HT	RT
3	95.42	6.50	101.92
4	86.97	9.05	96.01
5	81.93	11.44	93.37
6	77.88	14.01	91.89
7	75.22	16.71	91.94
8	72.61	19.09	91.70
9	70.89	21.68	92.57
10	68.88	22.48	91.37
11	67.98	24.80	92.78
12	67.41	26.50	93.91
13	65.11	28.62	93.73
14	64.44	30.06	94.50
15	64.17	31.16	95.33
16	63.73	31.70	95.43
17	63.15	30.33	93.48

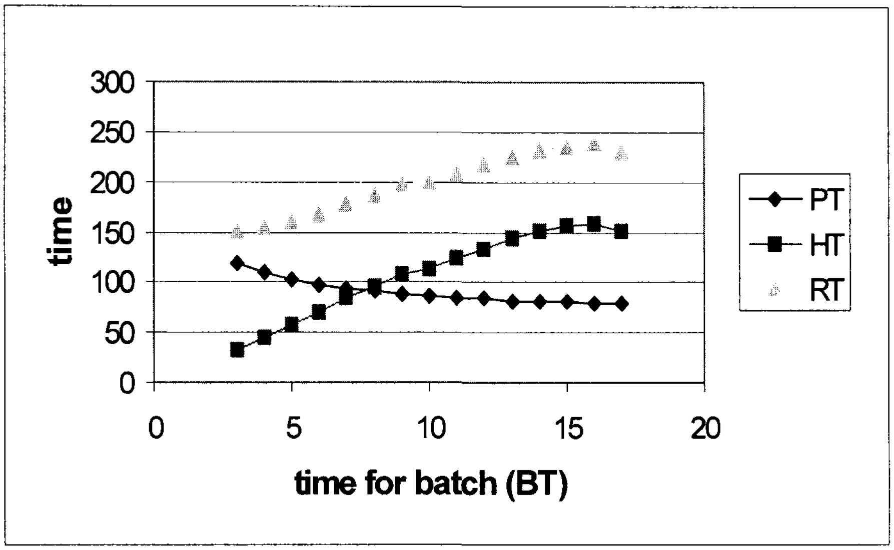

Figure 3.4 Simulation output for $\varpi_1 = 1.0$ and $\varpi_2 = 1.0$

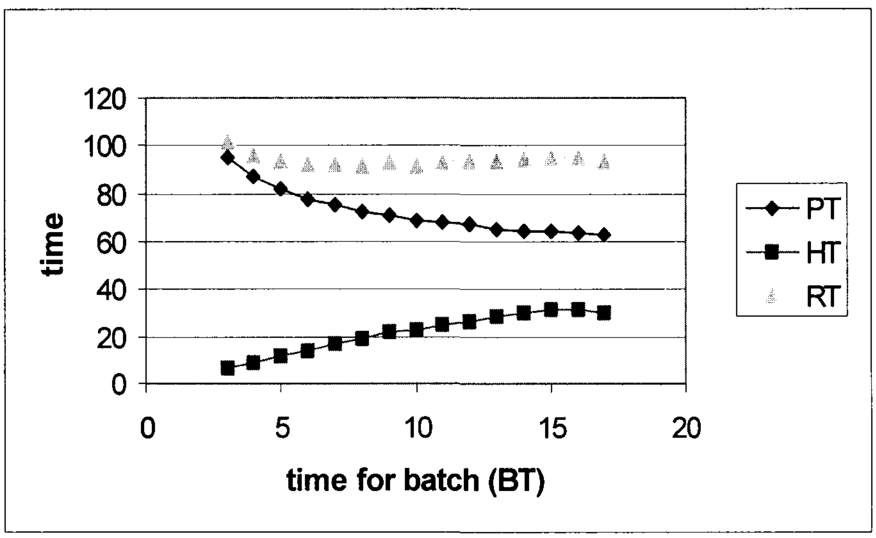

Figure 3.5 Simulation output for $\varpi_1 = 0.8$ and $\varpi_2 = 0.2$

3.7 Effect of high volume orders on order response time

In a typical warehouse or distribution center with high-volume orders, several thousand orders are picked per day. Sometimes the pickers or picking devices operate almost continuously and the transaction (order picking and item storage) rates are high. Because the operation of such systems is very expensive, managers are interested in maximizing the throughput by adjusting the design and operation of the warehouse.

In this section, the effect of high-volume of orders on order response time of the proposed model is investigated. Table 3.6 shows the simulation output of the system with the order arrival rate of 100 per hour. This experiment was replicated 50 times.

Table 3.7 Simulation output with high-volume order system of 100 per hour for $\varpi_1 = 0.8$ and $\varpi_2 = 0.2$

BT	PT	HT	RT
5	255.0	86.1	341.1
6	251.4	96.0	347.4
7	249.2	93.2	342.4
8	248.7	92.4	341.1
9	247.6	86.8	334.4
10	247.8	85.9	333.7
11	247.7	83.8	331.6
12	247.8	83.8	331.6
13	247.8	83.8	331.6
14	247.8	83.8	331.6
15	247.8	83.8	331.6

The order holding time and order picking time in a high-volume order system show little variations. Because the proposed JBP algorithm basically uses the FCFS rule and the arrival rate is high, the batching policy of based upon an arrival time does not give much benefit. This means the proposed algorithm is applied to a medium-sized order arrival system effectively. Figure 3.6 is a pictorial representation of Table 3.7. Unlike the previous results, because there are many orders to be picked from a storage rack in a small amount of time unit, the batching policy-based on the FCFS rule does not do good, and picking time and holding time are almost flat over the range of batch time.

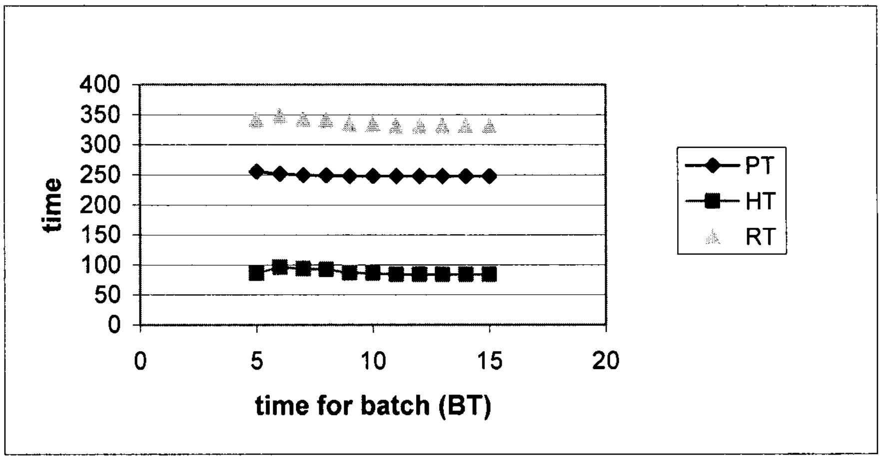

Figure 3.6 Simulation output with high-volume order arrivals of 100 per hour

3.8 Conclusion

In this chapter, the joint order batching and order picking problem was studied. Its mathematical formulation is based upon the bin packing problem (BPP) and traveling salesman problem (TSP). Although the exact algorithms for the BPP and the TSP exist, because the order batching and order picking problems need quick solutions because of operational policies, the proposed JBP heuristic algorithm for an alternative was developed. The heuristic method tries to find the between-batch time to minimize the order response time. The order response time is consisted of the order picking in terms of the warehouse efficiency and the order holding in terms of supply chain management. The former represents the time to retrieve the items of orders in a tour. The latter corresponds to the time for orders to wait in queue in processing them in a tour.

Two correcting factors of order picking factor and order holding factor are considered and applied to the model. Depending on the warehouse manager and supply chain status, the specific order picking value as a production time and holding value as a non-production time can be utilized. By giving the appropriate values to the picking factor and the holding factor, the warehouse system could be described in reality. The best between-batch time could be obtained by using the JBP heuristic method.

Also, the effect of high-volume of order arrivals on order response time was investigated. It does not give little benefit. Because of the high order arrival in a small amount of time unit, the batching policy-based on the FCFS rule does not give much benefit. The order picking time and holding time for high order arrival rate is almost flat over the range of batch time.

Finally, the current trends in respect to the supply chain management (SCM) is reacting to the global market requirements of the Cross-Docking customer service for a warehouse or distribution center, while maximizing the value added from machines and material, and minimizing the total cost occurred from the related activities. Customers want quick service or response. As mentioned above, the optimal batching method did not give very good response time, i.e., long waiting or holding time within the warehousing system.

The challenge is to process orders in a way that maximizes utilization of capacity, yet minimizes order response time to increase the level of customer service and satisfaction. Excess capacity and inventory is no longer an acceptable strategy for coping with variability in order content, arrival and processing times.

In the next chapter, simple yet effective order batching policies are introduced. They are easy to implement and practical for batching. If the number of orders to be completed is focused the number of order-based batching policies can be a candidate. Also, with the tolerable limit-based order batching policies, even if the vehicle capacity is not full, the order batching can be started according to the tolerable limit.

CHAPTER 4 SIMPLE ORDER BATCHING POLICIES

4. 1 Introduction

As previously discussed, the retrieval of items is an important function in warehouse systems. For example, we might store pallets (unit loads) in a warehouse and retrieve individual cartons (items) in response to a specific request or order. It is often better to group several small orders into a large amount of work, called a batch, to realize an increase in labor efficiency and to increase the response time for satisfying customer orders.

There are a number of different ways to process orders. The simplest way is to process one order at a time. Using this strategy, it takes a long time to complete all the orders. Thus, if there are many small orders, it can be more efficient to pick by group (pick-and-sort). There is also an intermediate form, in which several orders are picked together in one tour and sorted by the picker (sort-while-pick). Batching is the process of combining one or more customer orders into one batch. All orders that are to be processed are joined together, after which they are sorted into separate pick orders. This means that in one route through the warehouse system several orders are picked. During the picking process, the items may be sorted into the original customer orders.

In Chapter 3, a heuristic algorithm that can be used to heuristically solve the batching and picking problems at the same time in warehousing system is presented. For an order batching problem, the fixed time interval approach was used. For an order picking problem, a heuristic algorithm for the well-known traveling salesman problem (TSP) in order to minimize the travel time of a vehicle was applied. Also, in order to reflect the warehouse efficiency and promptness of the order response, both order picking factor and order holding factor are introduced.

Also, the order holding time is mentioned to be important with respect to the supply chain management (SCM). Increasingly, companies are realizing that they are competing on the basis on time. Most research has focused on the time related to warehousing activities, rather than non-activity related time, for example, order holding time in the warehousing system. Related to the order batching and order picking problems, it is worthwhile to consider batching policies that consider orders based on first-come first-serve in order to reduce the order holding time and improve the customer's satisfaction. To paraphrase, reducing the time required to provide the customer with products is on of the major forces that is leading organizations to participate in supply chain management initiatives.

In this chapter, simple, yet efficient batching policies to trigger batching are presented. Three naive heuristic methods for batching are presented. The first heuristic method is about the number of order-based batching method that is based on the first-come first-served (FCFS) rule, and the second one is the tolerbale-limit based method. The last one uses well-known first-fit (FF) and first-fit decreasing (FFD) heuristic method for the bin packing problem (BPP).

4. 2 Number of order-based batching

The number of order-based batching strategy tries to group the orders into batches based on how many orders there are in a batch. Because the parameter of the number of orders to be processed is given, the vehicle can process at most the maximum number of orders at a time. Also the sum of items in the batch cannot exceed the capacity of the picking vehicle or storage/retrieval (S/R) machine. This strategy can be applied if the picking system can wait for orders to be processed until the storage bins for each order on the vehicle are full. Because this feature utilizes the full capacity of the vehicle in a sense, it can reduce the number of batches. Its downside is a long waiting time in the system. As in Chapter 3, the naïve FCFS rule is used. Figure 4.1 shows the pseudocode of the order-based batching and italicized part corresponds to the condition for the orders to be formed as a batch.

```
While orders are unassigned Do
Get a new batch
        While the batch is less than full AND the orders are less
          than the maximum number of orders
                Select an unassigned order based on FCFS
                      and assign it the batch
        Reduce the capacity
```

Figure 4.1 Order-based heuristic pseudocode

Table 4.1 shows the results of the suggested number of order-based heuristic method. The right three columns correspond to the adjusted values by the picking and holding factors. Figure 4.2 corresponds to the pictorial representation of Table 4.1 with order picking factor of 0.9 and order holding factor of 0.1 as in Chapter 3. Where PT, HT, and RT mean the picking time (PT), holding time (HT) and response time (RT) of the batch, respectively. As the result indicates the small number of orders for a batch or tour give rise to long picking time, but short holding time. According to the simulation, 8 orders per batch gave the smallest response time of 90.29 min in order to process all orders in the specific time window.

Table 4.1 Results of number of order-based batching
with $\varpi_1 = 0.9$ and $\varpi_2 = 0.1$

	original			adjusted		
	PT	HT	RT	PT	HT	RT
5	89.94	112.44	202.38	80.95	11.24	92.19
6	84.70	143.23	227.93	76.23	14.32	90.55
7	84.46	163.94	248.40	76.01	16.39	92.41
8	79.03	191.62	270.65	71.13	19.16	90.29
9	79.14	208.64	287.78	71.23	20.86	92.09
10	78.39	229.53	307.92	70.55	22.95	93.50
11	77.16	235.24	312.40	69.44	23.52	92.97
12	77.10	238.20	315.30	69.39	23.82	93.21
13	77.10	239.42	316.52	69.39	23.94	93.33

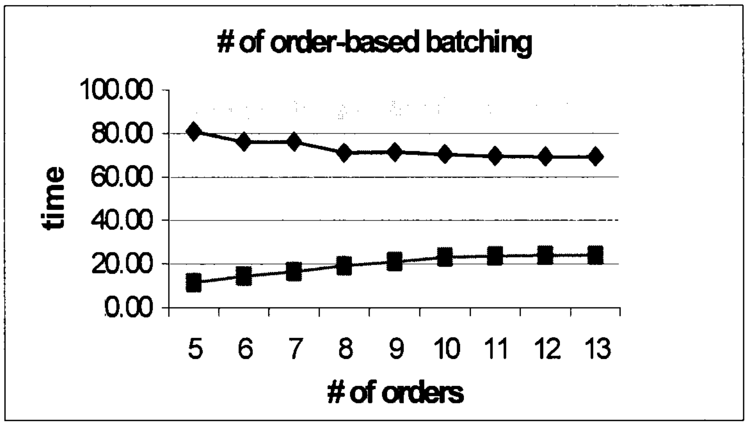

Figure 4.2 Graph of the last three columns of Table 4.1
with $\varpi_1 = 0.9$ and $\varpi_2 = 0.1$

4. 3 Tolerable limit-based batching

The objective of this batching method is to group the orders into batches based on the tolerable limit in a batch. Like before, the sum of items in every batch cannot exceed the capacity of the picking vehicle. A tolerable limit of a batch can be stated as follow: it is a maximum allowable waiting time of the first order in a batch for the vehicle to complete a picking process of the batch. In other words, it is the difference between the arrival time of the first order and that of the last order that can be formed and picked in a batch. For example, if the tolerable limit is set to 10 min, then the maximum time between the first and last order arrival in the batch should be less than or equal the 10 min. If there are a large order arrival rate in the system, its effect does not have on the order batching. This approach can be used in order to reduce the waiting time of the orders in the batch. So even if there are not enough orders to be processed in a batch the order picking process can begin in terms of the order response.

```
While orders are unassigned Do
Get a new batch
          While the batch is less than full AND the tolerable limit is not
             violated
                    Select an unassigned order based On FCFS
                              and assign it the batch
          Reduce the capacity
```

Figure 4.3 Tolerable limit-based heuristic pseudocode

Table 4.2 Result of tolerable limit-based batching

	original			adjusted		
	PT	HT	RT	PT	HT	RT
7	94.03	84.76	178.79	84.63	8.48	93.10
8	90.77	98.08	188.85	81.69	9.81	91.50
9	88.73	112.04	200.77	79.86	11.20	91.06
10	86.26	123.48	209.74	77.63	12.35	89.98
11	84.96	168.00	252.96	76.46	16.80	93.26
12	84.39	177.30	261.69	75.95	17.73	93.68
13	81.55	194.22	275.77	73.40	19.42	92.82
14	80.65	204.60	285.25	72.59	20.46	93.05
15	80.16	211.00	291.16	72.14	21.10	93.24

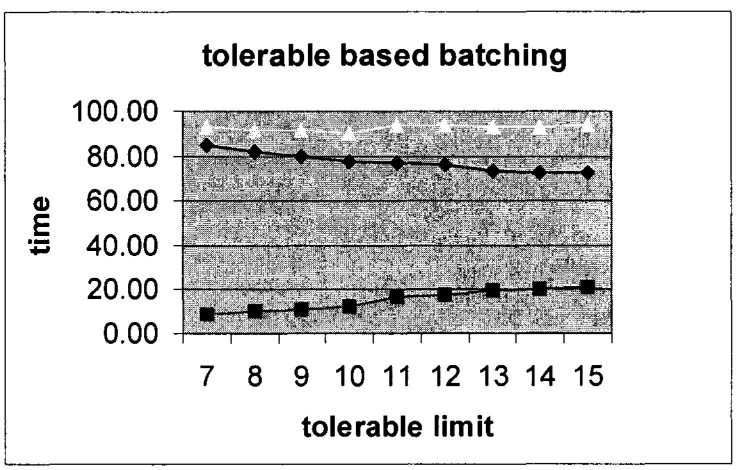

Figure 4.4 Graph of the last three columns of Table 4.2
with $\varpi_1 = 0.9$ and $\varpi_2 = 0.1$

According to the simulation results, 10 minutes of tolerable limit gives the smallest order response time of 89.98 per batch in the specific time window. After that the objective value increases to some extent, and then goes to steady value because of the vehicle's capacity. Even if the tolerable limit is set to be long, the capacity of the vehicle prevents the order response time from increasing continuously.

4. 4 First-fit and first-fit decreasing heuristic for batching

Many heuristics have been developed for this bin packing problem (BPP). One of the simple and popular approximate approaches to the bin packing problem is the first-fit (FF) algorithm. It assigns items to bins according to the order they appear in the list without using any knowledge of subsequent items in the list. The items in this heuristic are assigned to the lowest indexed bin into which it fits. When the current item cannot fit into any initialized bin, A new bin is introduced. Figure 4.5 shows the pseudocode for first-fit (FF) heuristic method for BPP.

```
While orders are unassigned Do
    Get a new bin
    While the batch is less than full
        Select an unassigned order based On FCFS
        Find out the bin with the lowest bin index that it fits
        If success, assign it to the bin and reduce the capacity
        Else get a new bin
```

Figure 4.5 FF-based heuristic pseudocode

In contrast to the first-fit (FF) heuristic, first-fit decreasing (FFD) first sorts the items in non-increasing order of their size (the number of items in the orders in this research) and conforms to the first-fit (FF) procedures in the remainder. The FF heuristic is called *on-line* algorithm, and the FFD heuristic is called *off-line* algorithm. Because the FFD approach does not take into account the arrival time of each order it gives a good batching or packing efficiency, but a long order response time in terms of supply chain management (SCM). To paraphrase, the latter gives better results in terms of compactness of a bin because it tries to use the unused space of a bin, which in turn can produce a few number of batches compared to the former FF heuristic method.

```
Sorts the orders in non-increasing order
While orders are unassigned Do
        Get a new bin
        While the batch is less than full
Select an unassigned order based On FCFS
Find out the bin with the lowest bin index that it fits
If success, assign it to the bin and reduce the capacity
        Else get a new bin
```

Figure 4.6 FFD-based heuristic pseudocode

As the result indicates the picking time (PT) for the FF and FFD heuristics are almost same, but not the holding time (HT). Because the FFD heuristic tries to fill the bin as much as possible it ignores the order arrival time. It cause the order picking process to be delayed. So it gives rise to long order holding time (HT), in turn order response time (RT).

Table 4.3 Result of FF and FFD heuristic batching

	Origianl			Adjusted		
	PT	HT	RT	PT	HT	RT
FF	77.26	288.38	365.64	69.53	28.84	98.37
FFD	76.91	711.70	788.61	69.22	71.17	140.39

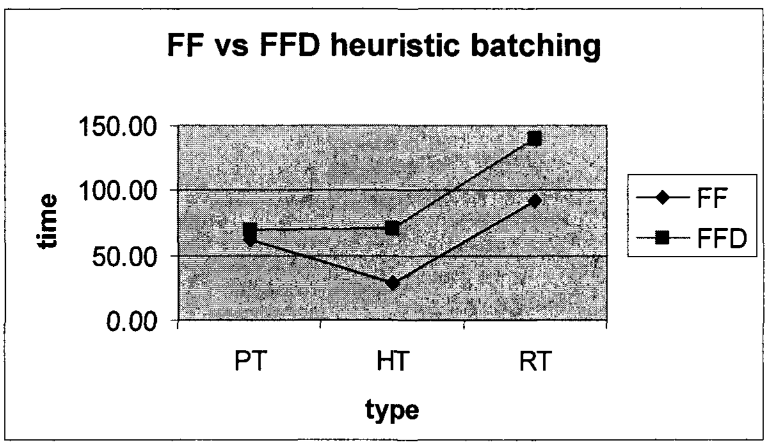

Figure 4.7 Graphical representation of Table 4.3 (last three columns)

4.5 Conclusion

Batch picking results in savings over single order picking whenever items on different orders can be picked simultaneously, especially, when those items are located in close proximity to one another in the warehouse. However, in a certain warehouse system, the time factor to affect on the next stage can be more essential than the distance factor. The customers would want their orders to be processed quickly, whereas the warehouse managers would want their equipment and overall systems to be operated efficiently and economically. This trade-off should be taken into account at the same time.

In this chapter, several simple order batching methods to trigger the batching and picking to be used at the next process are considered. All these batching policies can be applied to the warehouses or distribution centers according to the system environment. Like the results from the Chapter 3, according to the warehouse manager's decision, a different picking and holding factors are assigned.

For order picking representing the warehouse efficiency, the first fit (FF) and first-fit decreasing (FFD) heuristic methods or the order-number of batching methods can be utilized in the warehouse systems. Whereas for order holding representing the order promptness, the tolerable limit-based batching method gave the smallest holding time. As mentioned earlier, the FF and FFD heuristics for the bin packing problem (BPP) gave the smallest order picking time, but largest order holding time.

Chapter 5. The NP METHOD FOR THE JHPP

5.1 Introduction

In Chapter 3, the joint holding and picking problem (JHPP) was developed and its solution methodology was discussed in terms of a supply chain management (SCM) viewpoint. Also, the effects of order picking and holding factors on the response time of the problem are considered, and the effect of high volume orders on the problem was discussed. In Chapter 4, several simple order-batching policies were considered. They were the number of order-based batching, tolerable limit-based batching, and well-known first-fit heuristic batching. The starting point was the well-known first-come first served (FCFS) rule. The primary limitation of the FCFS rule is that an order with long waiting time will tend to delay or increase overall waiting time of other orders.

As discussed in the chapter 3 and chapter 4, the joint holding and picking problem (JHPP) corresponds to a combinatorial optimization, that is, obtaining the optimal solution among a finite set of solutions. Sometimes some of them are hard to get the solution, and some, even if a solution is found, may show much worse performance than other solutions of the problem. This fact makes it inappropriate to apply local optimization technique. A popular and effective way for escaping such local solution is to use random algorithm (Shi and Olafsson, 2000). This has given rise to many randomized optimization methods for optimization problems. In this chapter, the metaheuristic called nested partitions (NP) method for global optimization is presented. This method is primarily focused on solving problems that have a finite feasible region. The NP method is divided into four main steps:

partitioning, random sampling, selecting a promising region, and backtracking. Each of these steps can be implemented in a generic fashion, but can also be combined with heuristic methods and adapted to take a advantage of the special structure of the joint holding and picking problem (JHPP). In general, an important advantage of the NP method is that it can combine almost existing heuristic methods to speed its convergence (Shi et al, 1999, Olafsson and Shi, 2000).

In this chapter, the nested partitioning (NP) method for the joint order batching and picking problem (JBPP) is introduced and implemented. As an alternative of the previous heuristic methods, the NP method gives better performance compared to the heuristic results to the joint batching and picking problem. Empirical results indicate that the new methodology performs well relative to simple heuristics with respect to solution quality in some problems. Then the improved NP method that integrates the existing heuristic method of nearest neighborhood search to increase its efficiency is presented. Its computational time to find the solution can be greatly decreased for large problems.

5. 2 The NP algorithm for the JHPP

The basic idea behind the nested partitions (NP) method is to systematically partition the feasible region into subsets and focus the computational effort to those subsets that are considered promising. The motivation for this approach is that some of the feasible region may have big chance to contain global optima. This method can be understood as an optimization framework that uses partitioning to divide the feasible space into regions that can be analyzed individually and aggregates the results from each region to determine how to continue the search, that is, how to concentrate the computational effort. In other words, the

NP method adaptively samples from the entire space of possible feature subsets and concentrates sampling efforts by systematic partitioning of this space.

To implement the partitioning, the NP method maintains in the k-th iteration what is called the most promising region, that is, a sub-region that is considered the most likely to contain the best solution. This most promising region is partitioned into a given number of subregions and what remains is aggregated into one region called the surrounding region. Thus a disjoint collection of sets covering the entire feasible region is considered. The subregions and the surrounding region are sampled using typical random sampling method, and the sampling information used to determine which region should be the most promising region in the next iteration. If one of the subregions contains the best solution, this region is now selected as the new most promising region and is partitioned into the smaller subregions for the next iteration. If the surrounding region contains the best solution this is selected as an indication that the last move might not have been the best move. So the algorithm backtracks to what was the most promising region in the previous iteration. This partitioning creates a tree of subsets that we refer to as the partitioning tree. The distance of the current promising region from the top of the tree, which corresponds to the minimum number of iterations it takes to get to this region is called the depth of the region. Once the maximum depth is reached, the algorithm terminates. The notation used by NP method is below:

Θ = feasible region

$\Sigma = \{ \sigma \subseteq \Theta | \sigma$ is a valid region given a fixed partitioning $\}$

$\Sigma_0 \subset \Sigma = \{ \sigma \subseteq \Theta | \sigma$ is at maximum depth $\}$

$\sigma(k)$ = the most promising region in the k-th iteration

$d(\sigma)$ = the depth of region $\sigma \in \Sigma$

$s(\sigma)$ = the superregion of $\sigma \in \Sigma$

Partitioning

As the first step of each iteration, the current most promising region, $\sigma(k) \in \Sigma$, is partitioned into $M_{\sigma(k)}$, and aggregate the surrounding region into one region $\Theta \setminus \sigma(k)$. The generic way to partition the solution space is considered. Given *n* orders, the whole solution space becomes all permutations of $\{1, 2, \ldots, n\}$. First, this solution pace is divided into *n* equal parts by fixing the first order in the batch to be on of 1, 2, …, or *n*. This scheme can further partition each such subregion into *n* -1 parts by fixing the second order as any of the remaining *n* - 1 orders. This procedure can be repeated until the maximum depth is reached, when all the orders in the system are fixed. In this way, the subregions at maximum depth contain only single solutions.

Random sampling

The next step of the NP method is to randomly sample from each of the subregions and the aggregated surrounding region. The method used to get random samples from each region in each iterations is not fixed by the NP algorithm. However, sampling methods will greatly affect the efficiency of the NP algorithm. The generic random sampling is simply to pick the order that each orders have equal chance to be chosen (uniform sampling).

Calculating the promising index

Once each region has been sampled the next step is to utilize the sample points to estimate the promising index of the each region. Assume that a promising index function, $I(\sigma_j)$, has been defined and after selecting k orders that satisfies the condition of the vehicle capacity, the promising index, $I(\sigma_j)$, that will be used to determine the most promising region. For each region σ_j, j = 1, 2, ..., $M_{\sigma(k)}+1$, its performance is evaluated. Here, the promising index function, $I(\sigma_j)$, can be defined as to be the best performance value in the region

$$I(\sigma_j) = \min_{\theta \in \sigma_j} f(\theta), j = 1, 2, \ldots, M_{\sigma(k)}+1,$$

where $f(\theta)$ = sum of holding time and picking time.

Backtracking

The simplest method of backtracking rule would be to always to go to the immediate superregion of the current most promising region. If the depth of the current region is less than the maximum depth, then the algorithm backtracks to a superregion of the best solution found during the current iteration. This superregion is determined such that it ahs less depth than the current region. If the current region is at maximum depth, then the algorithm backtracks to a superregion of the best solution found in the current iteration.

5.3 Numerical results

Here we apply the generic NP algorithm to a joint holding and picking problem (JHPP) where the objective is to schedule the orders that will minimize the response time of customer orders in the warehouse or production facilities.

Since the JHPP problem can be formulated as a 0 – 1 IP problem, and it is well known that a branch and bound method can be used to solve such problems. A natural way of partitioning is to follow the branching procedure typically used for such problems. If this approach is used, a region $\sigma \in \Sigma$ of depth $d(\sigma)$ corresponds to a branch, whose root node is of depth, $d(\sigma)$. The order schedules in σ correspond to the leaf nodes of the branch. The promising index $I : \Sigma \to R$ can be a summary statistic, such as the best or smallest response time,

$$I(\sigma) = \min_{X \in \sigma} Z(X), \qquad \sigma \in \Sigma$$

This promising index function is naturally estimated by the best response time among all the order schedules sampled from the region. Backtracking is also simple, and backtracking to the next larger region is used. This corresponds to moving to a branch whose root node is of one less depth than the root node of the current most promising region (branch).

Consider a feasible region that consists of 8 orders $\eta_0 = \Sigma = \{1, 2, 3, 4, 5, 6, 7, 8\}$ per hour in the joint holding and picking problem. Each order has its arrival time for calculating a waiting time of the orders in a batch and has its corresponding number of items to be retrieved and shipped together. Also a picking time can be obtained if a batch is formed. The whole solution space becomes all permutations of {1, 2, … , 8}. First, this solution space is divided into 8 equal node or regions by fixing the first orders on the tour to be one of 1, 2, … , 8. We can further partition each such subregions into 8 – 1 = 7 orders by fixing the second orders as any of the remaining 7 orders on the tour. This procedure can be repeated until the maximum depth is reached, when all the locations on the tour are fixed. In this way, the subregions at the maximum depth contain only single solution. Fig 1. shows the example of a partition of a generic NP method. Here {2,3} mean first two orders are fixed, but remaining 6 orders are not assigned in the tour.

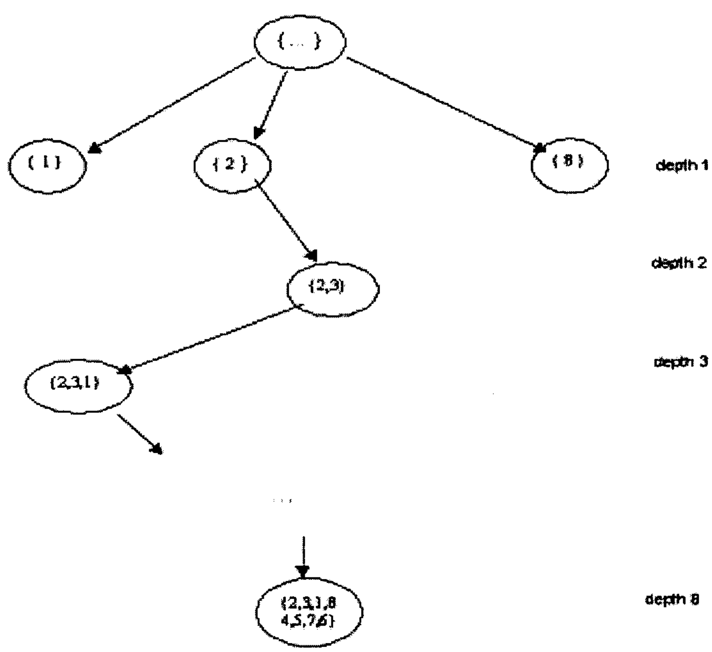

Figure 5.1. Partition tree of the JBPP generated by the NP method

A random sampling procedure is used to select 5 random samples from each of the regions, and calculate the corresponding performance values of minimizing the response time. After selecting 5 random points from each subregion, determination of the most promising region is made. For backtracking, the NP algorithm backtracks to a superregion of the best solution found during the iteration.

Table 5.1 Comparison of several performance results

# of Order	# of items	Optimal	CPU1	FCFS	NP	S.E	CPU1
6	1	160.62	2.38E+02	164.38	160.62	0	2.94E+02
	2	78.72	2.70E+02	82.56	78.72	0	2.94E+02
	3	68.35	2.67E+02	72.91	68.38	0.02	3.35E+02
7	1	122.11	3.21E+02	181.51	122.11*	0	3.49E+02
	2	92.64	3.65E+02	131.12	92.64*	0	2.21E+02
	3	77.84	3.88E+02	84.02	77.94	0.06	2.19E+02
8	1	112.24	6.13E+03	146.64	112.20*	0	3.27E+02
	2	101.54	5.23E+03	146.66	101.60*	0.58	3.54E+02
	3	84.69	9.92E+02	126.78	87.61*	3.97	2.11E+02
9	1	115.21	3.54E+03	127.29	116.00	0	3.32E+02
	2	101.3	4.51E+03	142.2	109.01*	4.73	< 100
	3	72.51	7.56E+03	101.62	72.55*	4.07	5.78E+02
10	1	126.3	1.29E+04	146.5	143.42	1.60	2.49E+02
	2	79.04	4.85E+04	127.18	85.79*	2.76	2.79E+02
	3	65.27	6.52E+04	114.99	89.48*	5.35	4.77E+02
11	1	119.45	3.54E+04	125.35	123.85	0.09	4.76E+02
	2	81.34	2.08E+04	109.92	105.20	1.65	4.38E+02
	3	78.56	3.81E+04	102.09	92.53*	3.42	6.89E+02
12	1	123.56	2.31E+04	161.84	159.00	1.18	7.24E+02
	2	101.56	3.31E+05	115.82	107.27	3.59	6.07E+02
	3	67.87	4.28E+05	113.43	102.61*	6.36	1.13E+03

CPU1 : CPU time for optimal solution
CPU2 : CPU time for NP method
S.E : Standard Error of NP results
* : Significance between FCFS and NP at 95 %

Here, just two parameters are considered for convenience in this example; number of orders per hour and number of items per order (Table 5.1). The minimum number of orders is 6 and the maximum number of orders is 12 orders per hour. The three items per order are used. As a termination condition, 100 program iterations were used.

According to the results from Table 5.1, FCFS result doesn't show any specific patterns. For small-size problem some results are good, but some results are so-so (for example, number of order is 7 and number of item is 1 or 2). But as the problem size goes large, the result is poor. Parameter values or pattern that FCFS works well cannot be found. This means that FCFS is just a rule of thumb, not a good measure of performance.

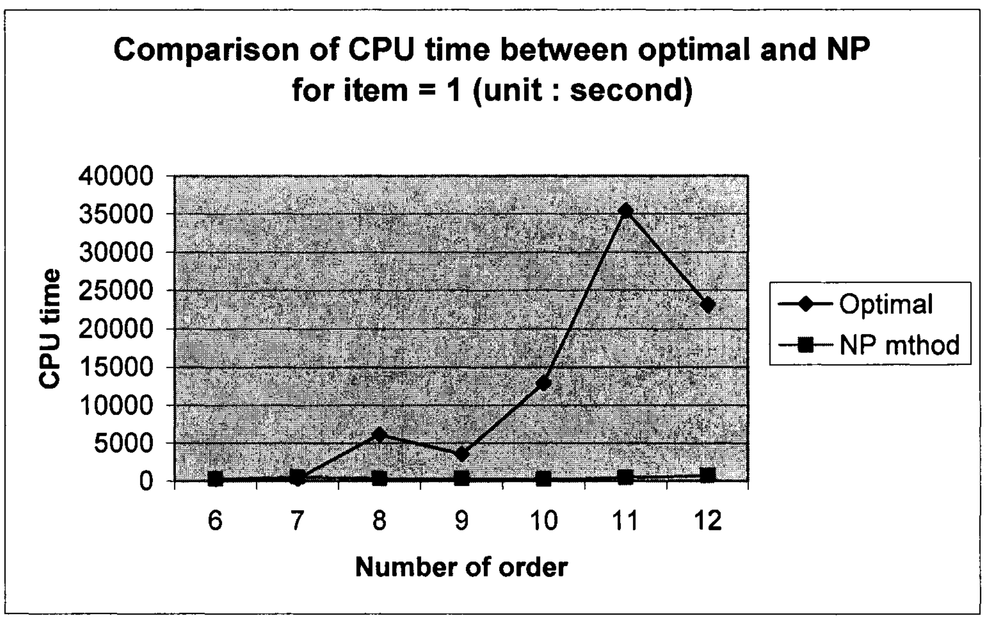

Figure 5.2 Comparison of CPU time between optimal and NP method for item = 1

However, the NP method shows very good result for small problems ($n < 10$). In such problem settings, the solution is same as the optimal solution. Even if the NP result is dependent upon several factors such as the program iterations, as the problem size gets larger, the difference of the performance measure between optimal solution and NP result is smaller than the big difference of the performance measure between optimal solution and FCFS result, respectively. So, in case of not getting an optimal solution and anticipating that the FCFS rule gives the poor results, the NP method can be a good alternative.

Table 5.2 shows the performance difference between the optimal solution and FCFS values (Dif_1), and that between the optimal solution and NP result. Diff1 shows the poor results of FCFS. Regardless of the problem size, the result is not reliable. However, for small problem, the difference between optimal solution and NP method is small (in some cases, NP method gives the optimal solution). But as the problem size increases its performance is deteriorated. The percentages indicate how much the solution is above the known optimal solution.

In this section, the generic NP optimization framework was introduced and implemented to the JHPP. Table 5.1 and 5.2, and Figure 5.1 show that the NP method gives near optimal solutions for the problems where the number of order is small. The performance of the FCFS rule shows the poor result as the problem size goes larger. But for the problem size, i.e., $n > 11$, the experiment doesn't show big difference between FCFS rule and NP method. This means the FCFS policy can be used for large size problems for a quick solution. However, this weakness can be overcome by changing the termination condition of the program. For it is not possible to solve the big size problems exactly within the reasonable amount of time.

Table 5.2 Performance difference

# of Order	# of Item per order	Dif_1	Dif_2
6	1	2.34%	0.00%
	2	4.88%	0.00%
	3	6.67%	0.04%
7	1	48.64%	0.00%
	2	41.54%	0.00%
	3	7.94%	0.13%
8	1	30.65%	0.00%
	2	44.44%	0.06%
	3	49.70%	3.45%
9	1	10.49%	0.69%
	2	40.38%	7.61%
	3	40.15%	0.06%
10	1	15.99%	13.56%
	2	60.91%	8.54%
	3	76.18%	37.09%*
11	1	4.94%	3.68%
	2	35.14%	29.33%*
	3	29.95%	17.78%*
12	1	30.98%	28.68%*
	2	14.04%	5.62%
	3	67.13%	51.19%*

Dif_1 : Percentage difference between optimal solution and FCFS

Dif_2 : Percentage difference between optimal solution and NP result

* : Significance between Optimal and NP result at 95 %

5.4 Improved NP algorithm

The NP method applied so far can be considered a generic optimization algorithm. The "generic" partitioning of the solution space means that it does not consider the objective function when the partitioning step of the NP method is in progress. This may result in difficulties in distinguishing between regions and consequently the algorithm may be hard to find any particular region to focus on the computational efforts. However, the advantage of the "generic" partitioning is that the search tree is in general predictable and is highly regular in terms of branching degrees and searching depths (Shi and Olafsson, 1997). When a generic partitioning is used, there is a chance to retreat frequently and not settle down in a particular region.

On the other hand, the NP method is capable of supporting different partitioning scheme, sampling methods, promising indices, and backtracking rules. An important advantage of the NP method is that it can incorporate almost any known efficient local heuristic to speed its convergence (Shi et al., 1999, Olafsson and Shi, 2000). The NP method is likely to perform the optimization tasks much better if good solutions tend to cluster together for a given region. In this section, the improved NP method for reducing the converging time is introduced to combine two optimization algorithms: the nested partitions (NP) method and nearest neighborhood search. In each iteration, the NP method has a subset of feasible region that is considered the most promising region. Such a region is defined by a partial tour or orders. Each of the regions and surrounding region is sampled using some random sampling scheme that assigns random levels to the remaining orders and estimated performance function values of the randomly selected orders are used to estimate the promising index for each region.

The improved NP algorithm using the nearest neighborhood search is described for the joint holding and picking problem (JHPP). The role of nearest neighborhood search is help to identify the good solutions. Because the good solution is closely related to the small waiting or holding time of orders, the nearest neighborhood search is applied to find next order in a tour. This results in finding efficient partitioning of the algorithm. This implementation takes advantage of the special structure of the JHPP both in the global sampling and local search.

First the partitioning scheme is explained. The basic idea behind partitioning is to completely schedule one job at each depth so that a region of depth *d* is defined by *d* orders being scheduled, or fixed, and the remaining n - *d* orders being free. Here the rule of scheduling selecting an order giving the least holding time given all the orders that have already been fixed is used. In other words, orders that results in a small holding time tend to be scheduled earlier than the orders that are fixed at greater depth. This implies a structure on the feasible region and allows for special structure to be incorporated into global and local search phases. For a finite feasible region, a good partitioning strategy always exists and can be obtained by enumeration of the feasible region, but it may not be known a priori.

Given a region $\sigma \in \Sigma$ and an order $j \in \Theta$, let $N_j^u(\sigma)$ denote the set of unassigned orders, N_j^a denote the set of the assigned orders in the system and $H_j(\sigma)$ be the sum of holding times of assigned orders in $\sigma \in \Sigma$. And let a_j be the arrival time of unassigned order *j*. Then find the argument of an order that gives the smallest holding time.

Partitioning Algorithm

Step 1. Orders that are fixed in σ are fixed to the same orders for each of the subregions

$$N_j^a = N_j^a, \quad d(\sigma)$$

Step 2. Find the argument that gives the smallest holding time in $N_j^u(\sigma)$

$$\hat{j} = \arg\min_{N_j^u}\{a_j - H_j(\sigma)\}$$

Step 3. Add the unassigned order obtained from Step 2 to N_j^a, and increment a depth count 1,

$$N_j^a = N_j^a + \hat{j}, \qquad d(\sigma) + 1,$$

Based on the improved NP algorithm described above, the same problem is revisited. The results are shown in Table 5.3, Table 5.4 and Figure 5.3. They show the optimal value, the first come first served (FCFS) solution, the generic NP solution, and improved NP solution of a response time, respectively. Also, their average CPU time in seconds is also reported. Table 5.3 shows performance of the NP algorithm and improved NP algorithm,

Table 5.3 Performance of the NP algorithm and improved-NP algorithm

# of Order	# of Item	Optimal	NP	S.E	NP-cpu	Imp-NP	S.E	Imp-NP cpu
	1	160.62	160.62	0	2.94E+02	160.62	0	2.63E+02
6	2	78.72	78.72	0	2.94E+02	78.72	0	2.41E+02
	3	68.35	68.38	0.02	3.35E+02	68.43	0.12	3.32E+02
	1	122.11	122.11	0	5.49E+02	122.11	0	3.31E+02
7	2	92.64	92.64	0	2.21E+02	92.64	0	2.08E+02
	3	77.84	77.94	0.06	2.19E+02	77.86	0.16	2.08E+02
	1	112.24	112.20	0	3.27E+02	112.24	0	1.37E+02
8	2	101.54	101.60	0.58	3.54E+02	101.54	0.58	3.07E+02
	3	84.69	87.61	3.97	2.11E+02	85.47	3.87	2.21E+02
	1	115.21	116.00	0	3.32E+02	116.39	0	1.84E+02
9	2	101.3	109.01	4.73	< 100	105.80	4.75	< 100
	3	72.51	72.55	4.07	5.78E+02	72.83	4.07	2.34E+02
	1	126.3	143.42*	1.60	2.49E+02	141.40*	1.61	2.81E+02
10	2	79.04	85.79	2.76	2.79E+02	99.75*	2.76	2.22E+02
	3	65.27	89.48*	5.35	4.77E+02	74.28	5.05	3.33E+02
	1	119.45	123.85	0.09	4.76E+02	123.86	0.09	4.47E+02
11	2	81.34	105.20*	1.65	4.38E+02	101.00*	1.55	4.28E+02
	3	78.56	92.53*	3.42	6.89E+02	92.30*	3.42	5.91E+02
	1	123.56	159.00*	1.18	7.24E+02	158.85*	1.18	4.81E+02
12	2	101.56	107.27	3.59	6.07E+02	105.82	4.59	5.05E+02
	3	67.87	102.61*	6.36	1.13E+03	84.52*	5.36	3.72E+02

* : Significance between optimal and NP methods at 95 %

including the optimal solution of the several problems with different parameters. The CPU time found by the NP algorithm always shows a much better than that of the optimal solution, and the results from improved NP algorithm which integrated the local search into the NP framework tend to have better performance (smaller CPU time) than the generic NP method.

Table 5.4 shows the average CPU time in seconds. The CPU time required to obtain the optimal solution increases as the problem size increases. But both NP methods are static. This can be explained by the program termination. In this program, the termination condition is set to the specific iteration. So if the iteration count of the program reaches its maximum the program terminates. However, if the simulation program increases the size of the iteration the CPU time will also be increased. The response time through the improved NP method shows almost same results as the results by the generic NP algorithm for most problems. This is because within the specified termination counter, both generic NP and improve NP method give same result.

Table 5.4 CPU time for optimal value, NP method and improved NP method

(number of items per order = 1)

(unit : second)

Order	Optimal	NP	Imp-NP
6	2.38E+02	2.94E+02	2.63E+02
7	3.21E+02	5.49E+02	3.31E+02
8	6.13E+03	3.27E+02	1.37E+02
9	3.54E+03	3.32E+02	1.84E+02
10	1.29E+04	2.49E+02	2.81E+02
11	3.54E+04	4.76E+02	4.47E+02
12	2.31E+04	7.24E+02	4.81E+02

In Table 5.4, the CPU time of the improved NP algorithm shows statistical significance ($p = 0.027$) than the generic NP method at the 95 % confidence interval. As mentioned earlier, because the improved NP method incorporates the special structure of the nearest neighborhood method when partitioning it can decrease the computational time of finding its solution by partitioning the feasible region efficiently. Figure 5.3 shows the percentage above optimal solution to FCFS rule, generic NP method, and Improved-NP method for item = 1. As can be seen in Figure 5.3, the value of Improved NP gives the best results. This means the result from the improved-NP method tries to imitate the optimal solution. Similar results for item = 2 and 3 can be found in Figure 5.4. and Figure 5.5.

From Figure 5.3, Figure 5.4, and Figure 5.5, two conclusions can be drawn. First, for small size problems (less than 10 of orders), the generic and Improved-NP methods give the almost optimal solution. Secondly, as the problem size increases both NP methods do worse than when there are few orders. This can be overcome by increasing the program iterations. As Shi and Olafsson (1999) indicate the NP algorithm converges to optimal solution with probability one. Another reason for poor results is that as the size of orders goes large the holding time is increased, and in turn, increased order response time.

The benefit of the improved NP method is that the desirable solution can be found quickly, and its computational time outperforms the one obtained through the generic NP method as the problem size grows large. This fact gives a strong indication that the improved NP algorithm may be very useful in application needed a quick solution.

Lastly, the effect of the program iterations on the performance value from the generic and Improved NP method is presented. As the number of program iteration increases, the result from both generic and Improved-NP method approaches to the optimal solution as in Figure 5.6, Figure 5.7 and Figure 5.8.

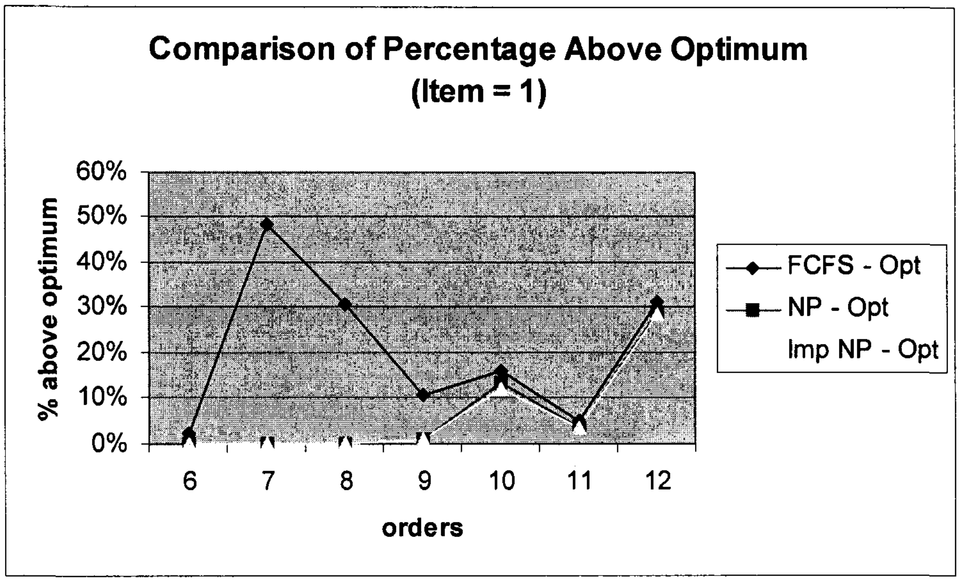

Figure 5.3 Graphical representation of percentage above optimum for item = 1

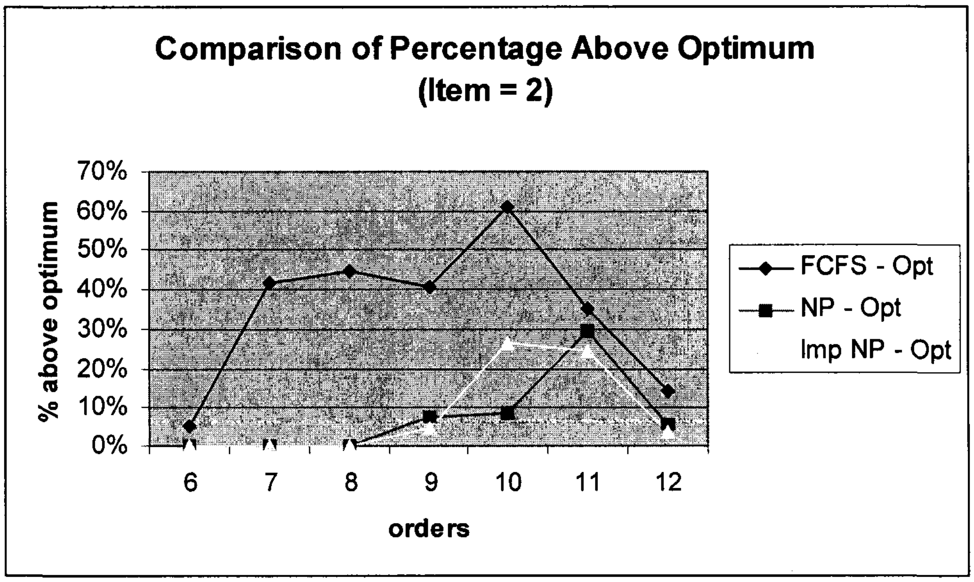

Figure 5.4. Graphical representation of percentage above optimum for item = 2

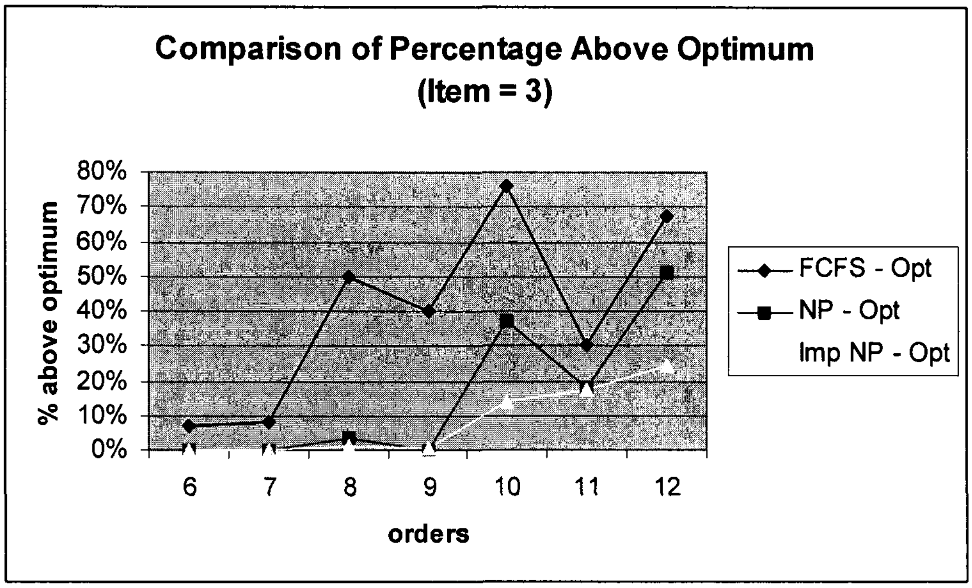

Figure 5.5. Graphical representation of percentage above optimum for item = 3

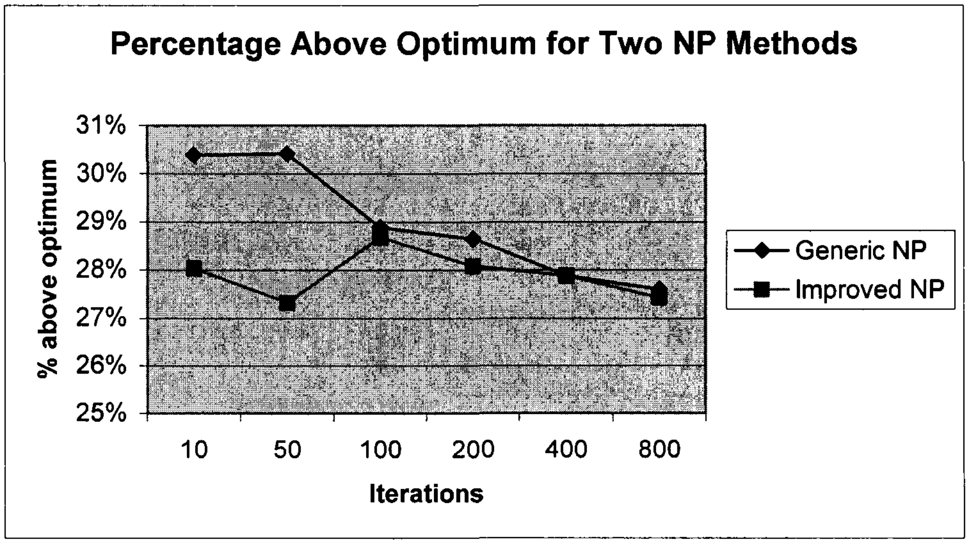

Figure 5.6 Percentage above optimum for the number of iterations (Item = 1)

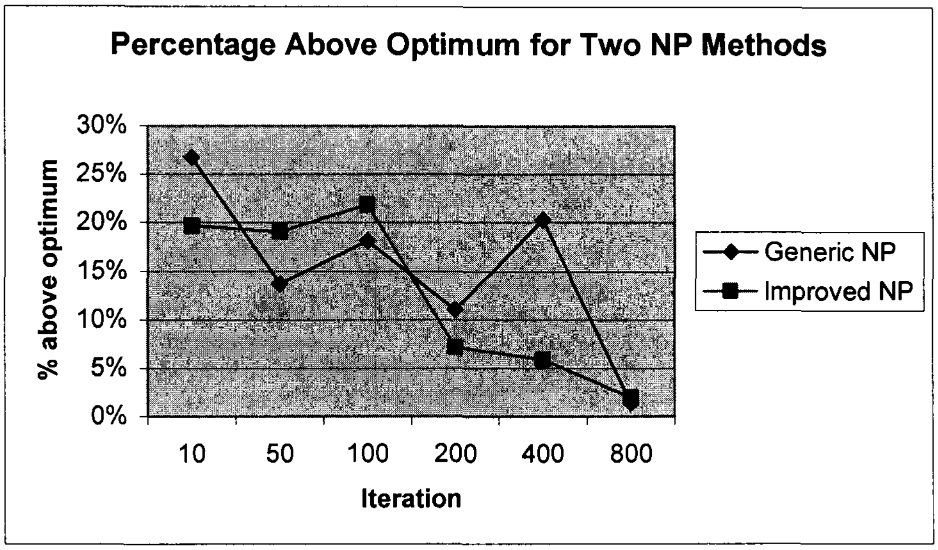

Figure 5.7 Percentage above optimum for the number of iterations (Item = 2)

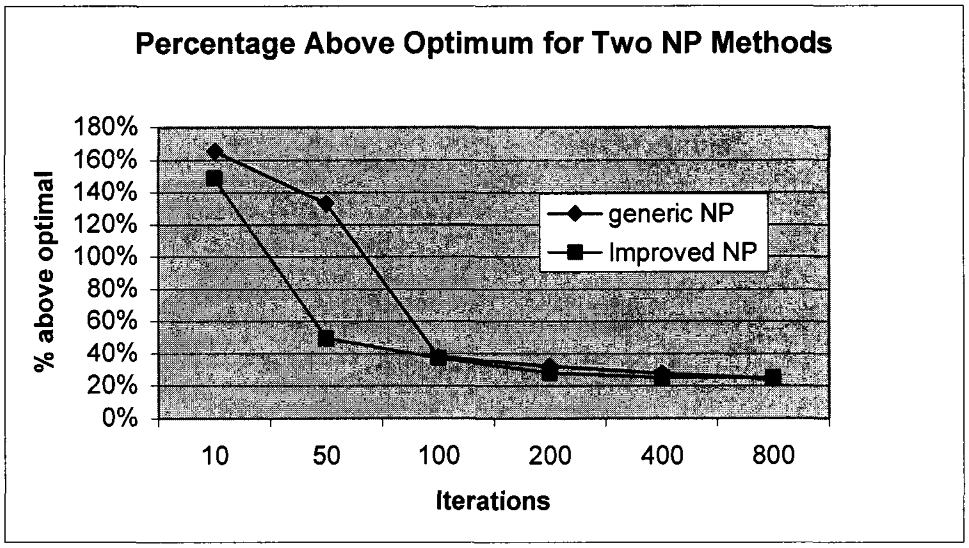

Figure 5.8 Percentage above optimum for the number of iterations (Item = 3)

5.6 Conclusions and extensions

With tighter supply chain integration it is possible to make high quality prediction for orders. By solving the batching/picking problem arrived at the warehouse or system, an approximate delivery time once an order is placed can be announced. For example, manufacturer can make use of its supplier's production and delivery schedule to improve its own production schedule. Also a warehouse manager can improve its inventory turnover by the information sharing of customers.

In this chapter, an optimization method called the nested partitioning (NP) is described and implemented for the joint order batching holding problem. The NP method gave much better performance compared to the simple heuristic results. It can be applied effectively to predict the delivery time of orders to customers. Also the improved NP method that integrates the existing heuristic method into the NP framework in order to improve its efficiency is presented. With this method applied to order batching and picking problem its computational time to find the solution can be dramatically decreased.

In particular, for big-sized problems the improved NP method can be used to obtain a realistic solution rather than simple heuristics or generic NP method. The reason that the improved NP method gives similar result of the generic NP method is the program count is fixed. The NP method can be applied to general or different objective functions, such as, total profit, utilization of the system for future research. Also, when the pick density, or the items per order, is large, our research can be extended further.

CHAPTER 6. CONCLUSION AND FUTURE RESEARCH

In this thesis an order batching and picking problem was considered in terms of supply chain management (SCM). In a rapidly changing business environment, order batching and order picking is commonly considered the most critical activity in supply cain. The importance of order picking is becoming more apparent as new e-commerce operations prosper to compete with traditional business. Consider for example e-commerce bookstore service. They distinguish themselves from traditional bookstores in that they must absorb the cost to pick customer orders, whereas for a traditional grocery store the customer performs this function for free.

The objective of this research is to investigate and find solutions of the joint order batching and picking problem for the performance improvement associated with the warehouse efficiency and customer service of order response. At the time of forming a batch by considering the holding time and picking time of orders at the same time, the joint order batching and picking problem is developed. The proposed joint batching picking problem is to find the between-batch time that considers the warehouse system and customer response. Based the order information that transferred from the warehouse management system (WMS), the optimal order batch time can be identified. Also, the effects of weights on order response time are investigated. By assigning an appropriate weights based on the supervisor's decision to picking operation and batching operation, more efficient management and processing of orders are possible.

Also, three simple order batching methods for the joint batching and picking problem are suggested. They are the number of order based batching, the tolerable limit-based order batching and first-fit (FF) heuristic batching. Batch picking results in savings over single order picking. Each method can be utilized efficiently in order to control the warehouse systems dependent upon the warehouse environment.

Finally, a new optimization method called the nested partitioning (NP) is described and implemented for the joint order batching holding problem. The NP method produced better performance compared to the FCFS results. Hence, rather than using the simple FCFS rule, a smart way of NP can be suggested. It can also be applied effectively to predict the delivery time of orders to customers. Furthermore, the improved NP method that integrates the existing heuristic method of the nearest neighborhood search into the NP framework in order to improve its efficiency is presented. The advantage of this method to generic NP method is that the improved NP method can reduce the time to find a reasonable solution. In particular, for big sized problem, this improved NP method can be applied.

Our research can be extended to several ways. First, when the pick density is large, or the order arrivals vary hour by hour, the effect of batching and picking can be investigated. Our research has restricted the number of items to less than four items, and the number of containers for order and assumed a constant order arrival. But this is not true for real life.

Second, when the production shop is located beside the warehouse, the shop can request the order directly. In this environment, production schedule and distribution schedule can be combined in our problem setting. For his combined problem, a more realistic and efficient solution can be obtained by the improved NP method.

By the tight supply chain and collaborative information, good planning, and realistic replenishment make the effective supply chain management firm. Further improvements come from supplier links to make then firm, fast and flexible for the benefit of the entire supply chain.

REFERENCES

Armstrong, R.D., Cook, W.D., and Saipe, A.L. , Optimal batching in a semi-automated order picking system, *Journal of Operational Research Society,* 30(8), 711-720, 1979

Ashayeri, J., and Goetschalckx, M., Analysis and design of order pick systems, *9th International Conference on Automation in Warehousing*, 125-135, 1988

Ashayeri, J. and Gelders, L.F., Warehouse Design Optimization, *European Journal of Operational Research*, 21, 285-294, 1985

Bachers, R., Dangelmeier, W., and Warnecke, H. J., Selection and use of orderpicking strategies in a high-bay warehouse, *Material Flow,* 4, 233-245, 1988

Barrett, B.G., A further digression on the over-automated warehouse: some evidence, *Interfaces,* 8(1), 46-49, 1977

Bozer, Y. A., Schorn, E. C., and Sharp, G. P., Analyzing picker-to-part order picking problems, *Proceedings of the 7th International Conference on Automation in Warehousing*, IFS Publications, 155-164, 1986

Bozer, Y. A., Schorn, E. C., and Sharp, G. P., Geometric approaches to solve the chebyshev travelling salesman problem, *IIE Transactions,* 22(3), 238-252, 1990

Bozer, Y. A., and Sharp, G. P., An empirical evaluation of general purpose automated order accumulation and sortation system used in batch picking, *Material Flow,* 2(2), 111-113, 1985

BrynzJr, H., and Johansson, M. I., Design and performance of kitting and order picking systems, *International Journal of Production Economics,* 41, 115-125, 1995

Aaron, F., Marchet, G. and Perego, A., Routing policies and COI-based storage policies in picker-to-part systems, *International Journal of Production Research,* 36(3), 713-732, 1998

Choe, K. I., Aisle-based order pick systems with batching, zoning and sorting, PhD Thesis, Georgia Institute of Technology, Atlanta, GA., 1991

Choe, K. I., Sharp, G. P., and Serfozo, R. S., Aisle-based order pick systems with batching, zoning and sorting, *Progress in Material Handling Research,* 1992, R. J. Graves et al. (eds.), Material Handling Institute, Charlotte, 245-276, 1993

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

Cormier, G. and Gunn, E. A., A review of warehouse models, *European Journal of Operational Research*, 58, 3-13, 1992

Daniels, R. L., Rummel, J. L. and Schantz, R., A model for warehouse order picking, *European Journal of Operational Research,* 105, 1-17, 1998

De Koster, R. and Van der Meer, J. R., Centralized versus decentralized control of internal transport, *Advances in distribution logistics*, B. Fleischmann et al. (eds.), Springer Verlag Berlin, 375-401, 1998

De Koster, R., Van der Poort, E.S., and Wolters, M., Efficient orderbatching methods in warehouses, *International Journal of Production Research*, 37(5), 515-531, 1998

De Koster, R., and Van der Poort, E.S., Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions, *IIE Transactions*, 30, 567-577, 1998

Elsayed, E. A., Algorithms for optimal material handling in automatic warehousing systems, *International Journal of Production Research,* 19(5), 525-535, 1981

Elsayed, E. A., Dube, R., and Proctor, C.L., Heuristic algorithms for handling orders within a warehouse, *Units and Bulk Materials Handling,* ASME Publication, 225-233, 1980

Elsayed, E. A., and Lee, M. K., Order processing in automated storage/retrieval systems with due dates, *IIE Transactions,* 28, 567-577, 1996

Elsayed, E. A., Lee, M. K., Kim, S., and Scherer, E., Sequencing and batching procedures for minimizing earliness and tardiness penalty of order retrievals, *International Journal of Production Research,* 31(3), 727-738, 1993

Elsayed, E. A., and Stern, R. G., Computerized algorithms for order processing in automated warehousing systems, *International Journal of Production Research,* 21(4), 579-585, 1983

Elsayed, E. A., and Unal, O. I., Order batching algorithms and travel-time estimation for automated storage/retrieval systems, *International Journal of Production Research,* 27(7), 1097-1114, 1983

Frazelle, E. H., Suggested techniques enable multi-criteria evaluation of material handling alternatives, *Industrial Engineering,* 17, February, 42-48, 1985

Frazelle, E.A., and Sharp, G.P., Correlated assignment strategy can improve order-picking operation, *Industrial Engineering,* 4, 33-37, 1989

Gademann, A.J.R.M., Van den Berg, J. P., and Van der Hoff, H. H., An order batching algorithm for wave picking in a parallel-aisle warehouse, *IIE Transactions,* 28, 567-577, 1997

Goetschalckx, M., and Ratliff, H. D., Order picking in an aisle, *IIE Transactions,* 20(1), 53-62, 1988a

Goetschalckx, M., and Ratliff, H. D., Sequencing picking operations in a man-aboard order picking system, *Material Flow,* 4, 255-263, 1988b

Gray, A. E., Karmakar, U. S., and Seidmann, A., Design and operation of an order-consolidation warehouse: models and application, *European Journal of Operations Research,* 58(1), 14-36, 1992

Graves, S. C., Hausmans, W. H., and Schwarz, L.B., Storage retrieval interleaving in automatic warehousing systems, *Management Science*, 23(9), 935-945, 1977

Hall, R. W., Distance approximations for routing manual pickers in a warehouse, *IIE Transactions,* 25(4), 76-87, 1993

Hausman, W. H., Schwarz, L. B., and Graves, S. C., Optimal storage assignment in automatic warehousing systems, *Management Science*, 22(6), 629-638, 1976

Hwang, H. B., and Lee, M. K., Clustering algorithms for order picking in an automated storage and retrieval system, *International Journal of Production Research,* 26(2), 189-201, 1988

Hwang, H. B., and Lee, M.K., Order batching algorithms for a man-on-board automated storage and retrieval system, *Engineering Costs and Production Economics,* 13, 285-294, 1988

Jeroen, P, B., A literature survey on planning and control of warehousing systems*, IIE Transactions*, 31, 751-762, 1999

Kanet, J. J., and Ramirez, R. G., Optimal stock picking decisions in automatic storage and retrieval system, *Omega International Journal of Management Science,* 14(3), 239-244, 1986

Ogburn, J. E., Order picking economy: batch vs. individual, *Proceedings of the International Material Handling Forum,* Houston, TX, 1984

Pan, C., H., and Liu, S., Y., A comparative study of order batching algorithms, *Omega International Journal of Management Science,* 23(6), 691-700, 1995

Park, Y. H., and Webster, D. B., Modelling of three-dimensional warehouse systems, *International Journal of Production Research,* 27(6), 985-1003, 1989

Petersen, C. G., An evaluation of order picking routing policies, *International Journal of Operations & Production Management,* 17(11), 1098-1111, 1997

Petersen, C.G., and Vaughan, T.S., The effect of warehouse cross aisles on order picking efficiency, *International Journal of Production Research,* September, 121-129, 1997

Randolph W., Hall distance approximation for routing manual pickers in a warehouse, *IIE Transactions*, 25(4), 1993

Ratliff, H. D., and Rosenthal, A. S., Orderpicking in a rectangular warehouse: a solvable case of the traveling salesman problem, *Operations Research,* 31(3), 507-521, 1983

Rosenwein, M. B., A comparison of heuristics for the problem of batching orders for warehouse selection, *International Journal of Production Research,* 34(3), 657-664, 1996

Sadiq M. T., Lander G. D., and Taylor G. D., An assignment algorithm for dynamic picking systems, *IIE Transaction*, 28, 607-616, 1996

Speaker, R.L., Bulk order picking, *Industrial Engineering,* 7(12), 14-18, 1975

Schwarz, L.B., Graves, S.C., and Hausman, W. H., Scheduling policies for automatic warehousing systems: simulation results, *AIIE Transactions*, 10(3), 260-270, 1978

Tompkins, J. A. and Smith J. D. The warehouse management handbook, Tompkins Press 1998

Vinod H. D. Integer programming and the theory of grouping, *Mathematica,* 506-519, 1979

APPENDIX A Algorithm of nested partitioning (NP)

Step 1. Partitioning.

Let $M_{\sigma(k)}$ denote the number of subregions of $\sigma(k)$, the current most promising region. The number $M_{\sigma(k)}$ may depend on $\sigma(k)$ but not on the iteration *k*, that is, if $\sigma(k_1)=\sigma(k_2)$ then $M_{\sigma(k_1)}=M_{\sigma(k_2)}$. Partition $\sigma(k)$ into $M_{\sigma(k)}$ subregions $\sigma_1(k),\ldots,\sigma_{M_{\sigma(k)}}(k)$, and aggregate the surrounding region $\Theta \setminus \sigma(k)$ into one region $\sigma_{M_{\sigma(k)}+1}(k)$.

Step 2. Random Sampling

Let N_j denote the number of sample points from region $\sigma_j(k)$. The number of sample points may depend on both the current most promising region and the iteration. Use a random sampling procedure to select N_j points from each of the regions $\sigma_j(k), j = 1,2, \ldots, M_{\sigma(k)}+1$,

$$\theta^{j1},\theta^{j2},\ldots,\theta^{jN_j}, \quad j = 1,2, \ldots, M_{\sigma(k)}+1,$$

and calculate the corresponding performance values,

$$f(\theta^{j1}), f(\theta^{j2}),\ldots, f(\theta^{jN_j}), \quad j = 1,2, \ldots, M_{\sigma(k)}+1,$$

The only requirement on the random sampling procedure is that each point in the region has a positive probability of being selected. Hence there is much flexibility in selecting a sampling procedure.

Step 3. Estimating the promising index

Given a promising index function, $I : \Sigma \to R$, estimate the promising index of each region. For example, assume that the promising index function is definedas the best performance value in the region,

$$I(\sigma) = \min_{\theta\in\sigma} f(\theta), \quad \sigma \in \Sigma$$

For each region σ_j, $j = 1,2, \dots, M_{\sigma(k)} + 1$, estimate the promising index $I(\sigma_j)$ by using sample points obtained in the previous step,

$$\hat{I}(\sigma_j) = \min_{i \in \{1,2,\dots,N_j\}} f(\theta^{ji}), \quad j = 1,2, \dots, M_{\sigma(k)} + 1,$$

Notice that $\hat{I}(\sigma_j)$ is a random variable.

Step 4. Backtracking

Determine the most promising region $\sigma_{\hat{j}}$, where

$$\hat{j} \in \arg \min_{j \in \{1,2,\dots,M_{\sigma(k)}+1\}} \hat{I}(\sigma_j),$$

If two or more regions are equally promising, the tie can be broken arbitrarily. If this index corresponds to a region that is a subregion of $\sigma(k)$, then let this be the most promising region in the next iteration. Otherwise, if the index corresponds to the surrounding region, backtrack to the superregion of the current most promising region. Notice that this region has one less depth than the current most promising region.

If $\hat{j} < M_{\sigma(k)} + 1$ then

let $\sigma(k+1) = \sigma_{\hat{j}}(k)$, let $d(k+1) = d(k) + 1$;

else,

let $\sigma(k+1) = s(\sigma(k))$, let $d(k+1) = d(k) - 1$;

APPENDIX B Comparison of CPU time between Optimal and NP Method

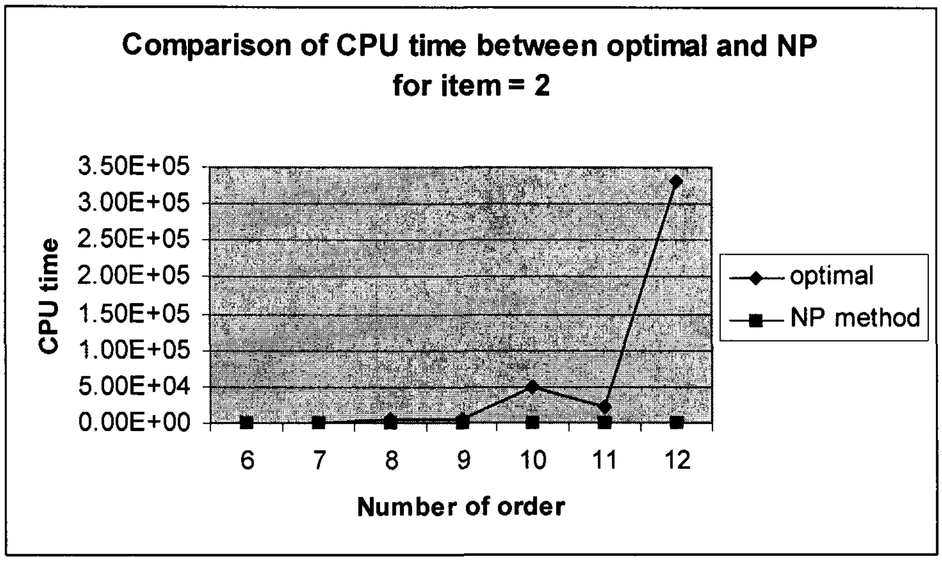

Figure A. Comparison of CPU time between optimal and NP method for item = 2

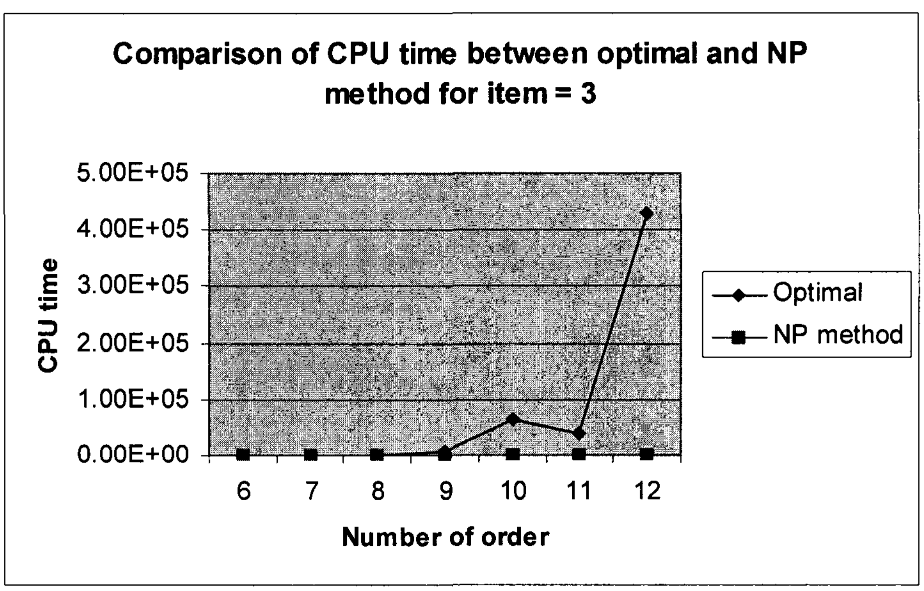

Figure B. Comparison of CPU time between optimal and NP method for item = 3

ACKNOWLEDGEMENTS

My thanks and appreciation to Dr. Sigurdur Olafsson for persevering with me as my advisor and mentor throughout the time it took me to complete this research and write the dissertation.

The members of my dissertation committee, Dr. Doug Gemmill, Dr. Suzuki Yoshinori, Dr. Jo Min, and Dr. Nilakanta Sree have generously given their time and expertise to better my work. I thank them for their contribution and their good-natured support.

I am grateful to many persons who supported and waited and prayed for me.