ABSTRACT

DESIGN AND OPTIMIZATION OF AN EXPLOSIVE STORAGE POLICY IN INTERNET FULFILLMENT WAREHOUSES

by
Sevilay Onal

This research investigates the warehousing operations of internet retailers. The primary physical process in internet retail is fulfillment, which typically involves a large internet fulfillment warehouse (IFW) that has been built and designed exclusively for online sales and an accompanying parcel delivery network. Based on observational studies of IFW operations at a leading internet retailer, the investigations find that traditional warehousing methods are being replaced by new methods which better leverage information technology and efficiently serve the new internet retail driven supply chain economy. Traditional methods assume a warehouse moves bulk volumes to retail points where the bulks get broken down into individual items and sold. But in internet retail all the middle elements of a supply chain are combined into the IFW. Specifically, six key structural differentiations between traditional and IFW operations are identified: (i) explosive storage policy (ii) very large number of beehive storage locations (iii) bins with commingled SKUs (iv) immediate order fulfillment (v) short picking routes with single unit picks and (vi) high transaction volumes with total digital control. In combination, these have the effect of organizing the entire IFW warehouse like a forward picking area. Several models to describe and control IFW operations are developed and optimized. For IFWs the primary performance metric is order fulfillment time, the interval between order receipt and shipment, with a target of less than four hours to allow for same day shipment. Central to achieving this objective is an explosive storage policy which is defined as: An incoming

bulk SKU is exploded into E storage lots such that no lot contains more than 10% of the received quantity, the lots are then stored in E locations anywhere in the warehouse without preset restrictions. The explosion ratio Ψ_o is introduced that measures the dispersion density, and show that in a randomized storage warehouse $\Psi_o < 0.01$, whereas in an IFW the likely range is $\Psi_o > 0.40$.

Specific research objectives that are accomplished: (i) Develope a descriptive and prescriptive model for the control of IFW product flows identifying control variables and parameters and their relationship to the fulfillment time performance objective, (ii) Use a simulation analysis and baseline or greedy storage and picking algorithms to confirm that fulfillment time is a convex function of E and sensitive to $\check{K}$, the pick list size. For an experimental problem the fulfillment time decrease by 7% and 16% for explosion ratios ranging between $\Psi_o=0.1$ and 0.8, confirming the benefits of an explosive strategy, (iii) Develope the Bin Weighted Order Fillability (BWOF) heuristic, a fast order picking algorithm which estimates the number of pending orders than can be filled from a specific bin location. For small problems (120 orders) the BWOF performes well against an optimal assignment. For 45 test problems the BWOF matches the optimal in 28 cases and within 10% in five cases. For the large simulation experimental problems the BWOF heuristic further reduces fulfillment time by 18% for $\check{K}=13$, 27% for $\check{K}=15$ and 39% for $\check{K}=17$. The best fulfillment times are achieved at $\Psi o=0.5$, allowing for additional benefits from faster storage times and reduced storage costs.

DESIGN AND OPTIMIZATION OF AN EXPLOSIVE STORAGE POLICY IN INTERNET FULFILLMENT WAREHOUSES

by
Sevilay Onal

A Dissertation Proposal
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Industrial Engineering

Department of Mechanical and Industrial Engineering

May 2017

APPROVAL PAGE

DESIGN AND OPTIMIZATION OF AN EXPLOSIVE STORAGE POLICY IN INTERNET FULFILLMENT WAREHOUSES

Sevilay Onal

Dr. Sanchoy K. Das, Dissertation Advisor — Date
Professor of Mechanical and Industrial Engineering, NJIT

Dr. Athanassios Bladikas, Committee Member — Date
Professor of Mechanical and Industrial Engineering, NJIT

Dr. Wenbo Cai, Committee Member — Date
Assistant Professor of Mechanical and Industrial Engineering, NJIT

Dr. Abdel Malek, Committee Member — Date
Professor of Mechanical and Industrial Engineering, NJIT

Dr. Lazar Spasovic, Committee Member — Date
Professor of Civil and Environmental Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Sevilay Onal

Degree: Doctor of Philosophy

Date: May 2017

Undergraduate and Graduate Education:

- Doctor of Philosophy in Industrial Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2017

- Master of Science in Industrial Engineering,
  NYU Polytechnic School of Engineering, Brooklyn, NY 2012

- Bachelor of Science in Industrial Engineering,
  Istanbul University, Istanbul, Turkey, 2008

Major: Industrial Engineering

Presentations and Publications:

Onal, S., et al. (2017). "Modelling and performance evaluation of explosive storage policies in internet fulfilment warehouses." International Journal of Production Research: 1-14. http://www.tandfonline.com/doi/full/10.1080/00207543.2017.1304663

Zhang, Onal and Das S., (2017). "Channel switching in a fixed period fast-fashion supply chain." International Journal of Production Economics (Second review).

Onal, S., et al. (2017). "Product Flows and Decision Models in Internet Fulfillment Warehouses with Explosive Storage." International Journal of Production Process and Control. (In Preparation)

Das R. et al., (2017). "Fulfillment time performance of amazon relative to other online retailers: An empirical analysis" International Journal of Distribution and Logistics. (In Preparation)

Onal and Das S., “Modelling and performance evaluation of explosive storage policies in internet fulfillment warehouses.” Conference Presentation, Institute for Operations Research and the Management Science Annual Conference, November 2016, Nashville, TN, United States.

Onal and Das S., “Evaluating reorder strategies in supply chains with lead time disruption risk.” Conference Presentation, Production and Operations Management Society Annual Conference, May 2015, Washington, D.C., United States.

Zhang, Onal and Das S., “A multi-channel fast fashion supply chain with switch control and replenishment policy.” Conference Presentation, Institute for Operations Research and the Management Science Annual Conference, November 2015, Philadelphia, PA, United States.

Onal and Das S., “Optimization problems in the design and control of internet fulfillment warehouses.” Poster Presentation, Institute for Operations Research and the Management Science Annual Conference, November 2016, Nashville, TN, United States.

Onal and Das, “Simulation analysis of chaotic storage policies in amazon class fulfillment centers.” Poster Presentation, NJIT Dana Knox Student Research Showcase, April 2016, Newark, NJ, United States.

Onal and Das S., “Simulation analysis of chaotic storage policies in amazon class fulfillment centers.” Poster Presentation, Institute for Operations Research and the Management Science Annual Conference, November 2015, Philadelphia, PA, United States.

Onal and Das S., “Evaluating reorder strategies in supply chains with lead time disruption risk.” Poster Presentation, Production and Operations Management Society Annual Conference, May 2015, Washington, D.C., United States.

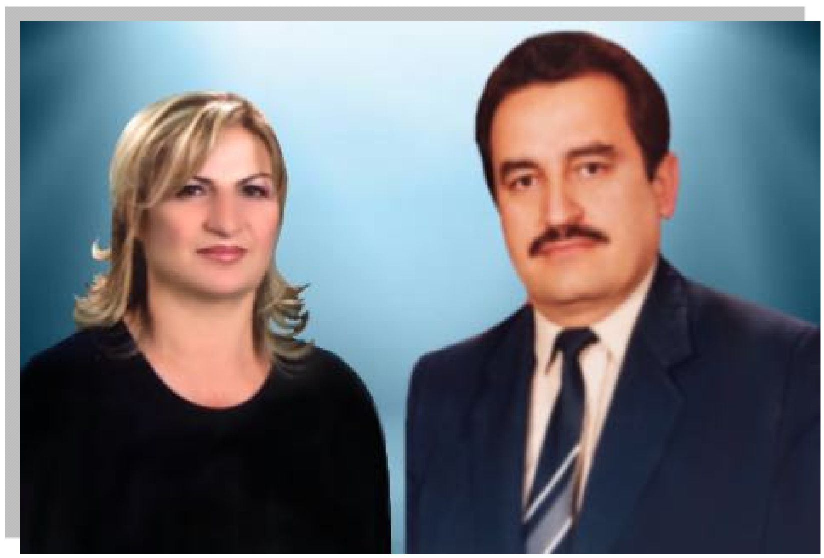
Onal and Das s., “Evaluating reorder strategies in supply chains with lead time disruption risk.” Poster Presentation, NJIT Dana Knox Student Research Showcase, April 2015, Newark, NJ, United States.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my advisor Prof. Sanchoy K. Das for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better advisor and mentor for my Ph.D. study.

Besides my advisor, I would like to thank the rest of my dissertation committee: Prof. Wenbo Cai, Prof. Abdel Malek, Prof. Athanassios Bladikas and Prof. Lazar Spasovic, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

I would like to thank my family: my parents Emine and Naim, my sisters Fusun, Ozlem and Yesim as well as my brother Onur for supporting me spiritually throughout writing this dissertation and with life in general. Also, special thanks to one of my favorite professors Blair Williams who steered me on my journey in the US and my grandparents Sunullah, Refika, Mehmet Arif, Gulsum, and Uncle Tahir Dursun whose love and guidance made me the person I am today.



This thesis is dedicated to my parents

for their love, endless support

and encouragement.

TABLE OF CONTENTS

Chapter **Page**

1 INTRODUCTION………………………………………………………………... 1

1.1 Research Motivation………………………………………………………. 2

1.2 Amazon Class Fulfillment Centers……………………………………….. 3

1.3 Problem Statement…………………….……………………………………. 7

1.4 Research Objectives and Accomplishments………….………….…….. 8

1.4.1 Develop Descriptive And Prescriptive Models For The Control of IFW Operations……..……................................. 8

1.4.2 Simulation confirmation of Performance Advantages of Explosive Storage Policy…………………………………… 9

1.4.3 Develop Advanced Order Picking Algorithms To Minimize Fulfillment Time…………………………………………….. 10

1.5 Research Significance……………………………………………… 11

2 LITERATURE REVIEW………………………………………………… 12

2.1 Warehouse Storage Design………………………………………. 12

2.1.1 Product or Material Flows in the Warehouse Design………. 12

2.1.2 Warehouse Design or the Facility Layout Problem…..…….. 14

2.1.3 Automated Storage/Retrieval Systems…………………….. 16

2.1.4 Recent Work………………………………………………... 18

2.2 Internet Fulfillment Centers………………………………………… 20

2.2.1 Amazon and E-retailing………………………….………….. 20

2.2.2 Related Work…………………...…………………………. 22

Chapter **TABLE OF CONTENTS (Continued)** **Page**

2.3 Random Storage Policy………………………………………………….... 22

2.3.1 The storage Location Assignment Problem………………. 23

2.3.2 Unit-load, Less than Unit-Load Policies and Forward Reserve Problem…………………………………………… 24

2.3.3 Shared Storage Policy and Some of the Other Widely Held Policies……………………………...…………………… 25

2.3.4 Recent Research…………………………………………….. 26

2.4 Order Fulfillment Objective Research Significance……………...…… 26

2.5 Warehouse Order Picking Algorithms………………………………… 28

2.5.1 Routing Policies……………………………….……………. 29

2.5.2 Zoning and Batching……………………….…………….…... 30

2.5.3 Travel Time Estimation and Probabilistic Approximation … 32

2.5.4 Recent Work and Further Reading……….…..………………... 33

3 INTERNET FULFILLMENT WAREHOUSES……………………...…… 35

3.1 Observational Study…………………………………………………... 37

3.2 Key IFW Structural Differentiators…………………………………..… 41

3.2.1 Explosive Storage Policy………………………….……….. 41

3.2.2 Very Large Number of Beehive Storage Locations………… 43

3.2.3 Bins with Commingled SKUs……………………………..…… 44

3.2.4 Immediate Fulfillment Objective…………………………… 46

3.2.5 Short Picking Routes with Single Unit Picks………….……… 46

3.2.6 High Transactions and Total Digital Control……………… 48

TABLE OF CONTENTS
(Continued)

Chapter		Page
	3.3 Modelling Fulfillment Operations……………………...………………	48
	3.3.1 Performance Objective – Minimize Fulfillment Time…………	51
	3.3.2 Uniform Random Stocking List Algorithm……………………	52
	3.3.3 Narrow Band Order Picking Algorithm (NBOP)……….……..	54
4	IFW SIMULATION……………………………………………………….	57
	4.1 Simulation Modelling…………………………………………………..	57
	4.1.1 Categorization of Simulation Models…………………………	57
	4.1.2 Steps of Creating a Simulation……………………………….	58
	4.1.3 Database Needs of a Simulation………………………………	59
	4.2 Simulating IFWs……………………………………………...........	60
	4.2.1 Simulation Objective……………………………………….	60
	4.2.2 Warehouse Operations…………………………………….	60
	4.2.3 Warehouse Design…………………………………………	61
	4.2.4 Input Data Generation……………………………………...	63
	4.2.5 Constructing the Database using MS Access…………………	65
	4.2.6 Simulation Process…………………………………………	67
	4.3 Performance Analysis of Explosive Storage…………………………..	69
	4.1.1 Effect of Increasing Explosion Ratios……………………......	70
	4.1.2 An Optimal Picking Parameter - Ǩ………………..................	72
	4.1.3 Average Number of Stops in a Pick List…………………….	73
	4.4 Results and Conclusions………………………………………………..	75

TABLE OF CONTENTS
(Continued)

Chapter **Page**

5 ADVANCED ORDER PICKING ALGORITHMS FOR IFWs………….……… 76

5.1 MIP Formulation of the Picking Problem…………………………..…. 77

5.2 Decision Variable Space……………………………………………… 79

5.3 Decision Variable Space Reduction…………………………………….. 81

5.3.1 Decision Variable Space Reduction…………………………… 81

5.3.2 Pre-Selection of Bins……………..…………………………… 82

5.3.3 Bin Weighing Method…………………..…………………..….. 83

5.4 Bin Weighted Order Fillability Algorithms (BWOF)………………….. 86

5.4.1 Heuristic 1 – Highest Maximum Fillable Factor……................. 87

5.4.2 Heuristic 2 – Amplified Maximum Fillable Factors………….. 87

5.5 Experimental Design…………………………………………….…..... 88

5.5.1 Factors and Factor Levels…………………………………..…. 89

5.5.2 Factors for Seed Algorithms…………………………..……..... 89

5.6 Solving the MIP………………………………..……………………… 90

5.6.1 OpenSolver…………………………………………………..…. 91

5.7 Experimental Analysis……………………………………………….…. 94

5.7.1 General Performance……………………………………….. 94

5.7.2 Average Pick Time Behavior…………………………………. 98

TABLE OF CONTENTS
(Continued)

Chapter **Page**

5.8 Simulation Analysis of BWOF Algorithms………………….…...… 100

5.8.1 Generating Pick Lists Using BWOF Algorithms…………… 101

5.8.2 General Performance……………………………….....…… 104

5.8.3 Economic Advantages of the BWOF Algorithm…………… 107

6 ACCOMPLISHMENTS AND FUTURE WORK…………………………. 120

6.1 Accomplishments and Significant Findings………………………… 109

6.2 Future Work………………………………………………………… 110

REFERENCES……………………………………………………….. 112

LIST OF FIGURES

Figure		Page
1.1	Amazon fulfillment center………..…………………………...………...	4
1.2	Inbound operations at Amazon…………………………………………	5
1.3	Order picking at Amazon………………………………………………	6
1.4	Automated technology at Amazon……………………………………..	7
2.1	The strategic level: long term decisions (5 years)………………………	13
2.2	The tactical level: medium term decisions (2 years)…………..............	13
2.3	The operational level: short term decisions (1 year)…………………...	14
3.1	Receiving and stocking process…………………………………………	37
3.2	Picking, consolidation, and truck loading assignment………………….	39
3.3	Inventory flow timeline in an IFW……………………………………...	41
3.4	Explosive storage of bulk SKU to multiple bin locations………...…….	43
3.5	Bins with commingled and randomly arranged SKUs………………….	45
3.6	Short and unique picking routes……………………….………………..	47
4.1	Warehouse design………………………………………………...........	62
4.2	Unidirectional zone design……………………………………………...	62
4.3	Input data generation file………………………………………………..	65
4.4	Objects view of tables in MS Access……………………………………	66
4.5	Entity relationship diagram of IFW model……………………………...	67
4.6	Design view of simulation modules…………………………………….	68
4.7	IFW control logic model……………………………………...............	69

LIST OF FIGURES
(Continued)

Figure		Page
4.8	Fulfillment time as function of increasing explosion ratios…………….	71
4.9	Fulfillment time as function of the maximum number of picking stops..	73
4.10	Average number of stops in a pick list………………………………….	74
5.1	Pick band demonstration……………………………………………..	77
5.2	Order-bin location map……………………………………………….	81
5.3	An example to bin weighing………………………………………….	85
5.4	MIP pick list generation procedure……………………………………	88
5.5	OpenSolver problem design…………………………………………..	91
5.6	OpenSolver model construction………………………………………	92
5.7	OpenSolver executing the model……………………………………..	93
5.8	Optimal to seed solution comparison of Type A problems....................	95
5.9	Optimal to seed solution comparison for Type B problems…………….	96
5.10	Optimal to seed solution comparison for Type C problems…………….	97
5.11	Overall performance comparison……………………………………..	97
5.12	Average pick distance for all problem types……………………………	98
5.13	Average pick list completion times…………………………………..	99
5.14	Average pick list completion time behavior for all problem types……..	100
5.15	The comparison of fulfillment time as function of explosion ratio……..	105
5.16	Percentage advantage of the BWOF algorithm…………………………	106
5.17	The comparison of mean stocking time……………............................	107
5.18	Total cost advantage of BWOF…………………………………………	108

LIST OF TABLES

Table		Page
4.1	Key Parameters for the Experimental IFW Problem………………………	69
4.2	Fulfillment Time as Function of Increasing Explosion Ratios…................	72
4.3	Average Fulfillment Time as a Function of the Max Number of Pick Stops	73
4.4	Average Number of Picks………......………………………....................	74
5.1	Experimental Design Problem Types………………………………………	88
5.2	Experimental Design Factors and Factor Levels…………………………..	89
5.3	Factors for Seed Algorithms……………..……………………..................	90
5.4	Problem Parameters……………………………………………………..	104
5.5	The Comparison of Mean Fulfillment Time……………………………….	104
5.6	Percent Improvements in Fulfillment Time Using BWOF Algorithm…….	106
5.7	The Comparison of Mean Stocking Time………...……………..................	107

CHAPTER 1

INTRODUCTION

This research investigates the warehousing operations of internet retailers. Central to these operations are fulfillment centers which receive customer orders and then retrieve, package and ship the items to the customer. While fulfillment centers are not a new or novel concept in themselves, this research will show that these new class of warehouse is structurally and operationally quite different from traditional warehousing configurations and fulfillment methods. This new class is labelled as internet fulfillment warehouses (IFWs), and described as facilities that have been designed and built exclusively for satisfying online retail sales.

U.S online retail sales as a percent of total retail sales have risen from 2.8% in 2006 to 7.2% in 2015 (Dept. of Commerce, 2015), of which less than 20% is currently fulfilled from am IFW. The current research literature on online commerce is primarily focused on the retailing side (Brynjolfsson et al, 2013, Verhoef et al 2015, Chen and Leteney, 2000), and with only limited reported work on the physical fulfillment side. Acimovic and Graves (2015) find that fulfillment warehouses are unique to online retail, and involve picking, packing, and shipping in rapid succession.

Warehousing is the main pillar of retailing with a major role in supply chain. Market competition and fast evolving technologies, require companies to look for new ways to reach customers more efficiently. In the past decade we have witnessed major changes in consumers' shopping habits. As trust was built with online retailers, many brick and mortar retailers assumed early on that simply offering products online, and fulfilling them from their traditional warehouses or store inventory was a sufficient solution.

1.1 Research Motivation

We live in an era where time is limited. Why waste it shopping when we can have our items delivered to our doorstep instead of wasting time and money commuting? This new trend in shopping definitely makes our life easier as consumers. The key factors in making a decision when purchasing a particular item are price, convenience and time. On the opposite side of the market, companies must relentlessly work to attract customers, as well as deal with market pressure to be more competitive. The best option for retailers is to minimize costs and order fulfillment times. But how successful can it be? The traditional warehousing methods are developed assuming a warehouse is in between producer and retailer. It was developed to move the bulk of items to places where it gets broken down into individual items and sold. An online store is where all the middle elements of a supply chain are combined into one. The intent of this research is to answer this question and develop a better responding system for e-retailers' warehouses that are referred to as "Internet Fulfillment Warehouses" (IFWs).

There is a small number of companies that realized this issue early and modified their operations accordingly. However, most retailers fail to see why traditional warehousing practices that are effective in retail stores do not perform so well online.

There have been many empirical studies about e-retailing practices and effects. Price, trust and loyalty comparisons between brick and mortar stores and online stores are the most common topics discussed. However there is little research that quantitatively investigates the issue for improving warehousing operations and shortening order fulfillment time. Existing literature assumes that travel time optimization is the best way to shorten response time to a customer. The goal of this dissertation is to analyze IFWs and optimize actual order fulfillment times which starts with the arrival of a customer order and ends when that specific order is shipped.

Realizing the gap in the existing literature about warehousing methods for the next generation of online retailing, the intent is to investigate the issue further and open doors to a large area for future researchers who could solve many problems which may or may not have been identified by e-retailers yet.

Specific research that is proposed in this study is: (i) *To develop a descriptive and prescriptive model for the control of IFW product and decision flows* by identifying key IFW differentiators, decision variables and parameters, as well as the performance objective (ii) *To obtain a simulation based confirmation of Performance Advantages of Explosive Storage Policy* using baseline (or greedy) storage and picking algorithms where order fulfillment time superiority of explosive storage is investigated to demonstrate that fulfillment time is a convex function of an explosion ratio Ψ which is sensitive to pick list size (iii) *To develop advanced order picking heuristics to minimize fulfillment time* the list of candidate picks is rather large, and the solution space is described by the product of pending orders and active inventory locations. Fast heuristics were developed to determine which orders to serve in the next pick list and which storage locations to fulfill the orders from.

1.2 Amazon Class Fulfillment Centers

Amazon is the leading company known for developing and managing IFWs successfully. In fact, they are the first company to call these class warehouses as “fulfillment centers”. Currently, Amazon operates over 390 active distribution facilities around the world including IFWs, return centers, specialty centers, and redistribution centers. These facilities total over 140 Million square feet of space with over 250,000 employees. Amazon started with two Fulfillment Centers (FCs) in Seattle and Delaware. The original 93,000 square foot Seattle facility that is mainly manually operated currently seems tiny relative to most of the new fulfillment centers being built. Today, an average size Amazon Fulfillment Center is over a million square feet. The figure below shows a size and procedure comparison between classical warehouses and chaotic IFWs.

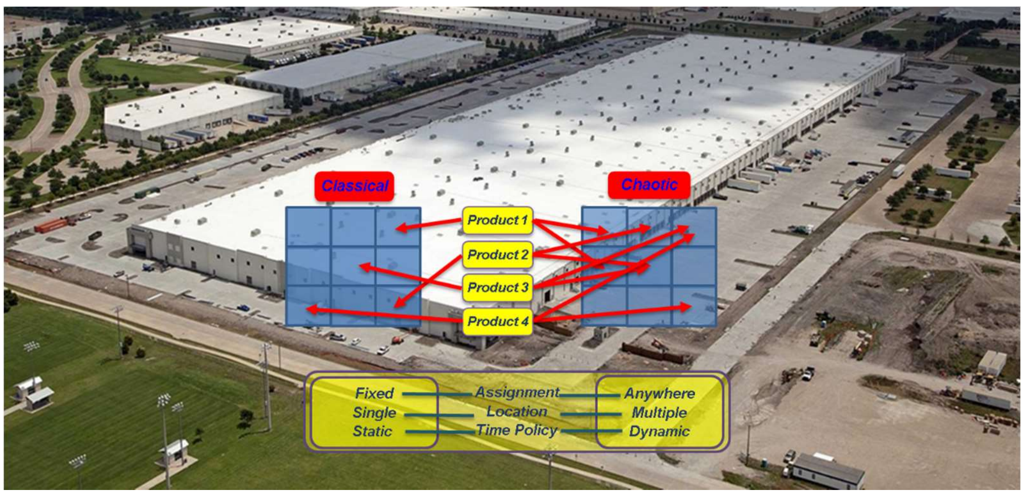


Figure 1.1 Amazon fulfillment center.

Source: Adrian Maties, "Amazon to open fulfillment center in Baltimore next year, create 1,000 full-time jobs." https://www.cpexecutive.com/post/amazon-to-open-fulfillment-center-in-baltimore-next-year-create-1000-full-time-jobs/ accessed April 30, 2017.

One of the fulfillment centers we visited is a 1.2 million sq. ft. Fulfillment Center in Indiana, named SDF8, which of their 30 million SKUs 90% is apparel, 5% is shoes and 5% is accessories. A storage unit in SDF8 consists of a small closet with hanging bars or cardboard drawers. Operations in Amazon can be divided into three sub-operations: (i) Inbound shipment, (ii) Picking and Packing, and (iii) Outbound shipment. This dissertation focuses on the first two operations.

Inbound operations is receipt of goods in the Fulfillment Center from either trusted sellers or others. Trucks arrive with boxes of single or bulk items that workers open, inspect, scan, and put into barcoded totes. Conveyor belts route the totes to different zones of the warehouse, where stockers unload them, and place them randomly into the first available slot in the zone, where they match the barcode with the storage unit. Each shelf is divided into bins. The location of every single item in the Fulfillment Center is known, and can be tracked by computers at any time. Items are shelved randomly where they fit, however, more effective use of storage units is achieved with digital control. SKUs are stocked in many bins throughout the warehouse that make them more accessible to pickers thus reducing worker travel distance.

Figure 1.2 Inbound operations at Amazon.

Source: Marcus Wohlsen "A rare peek inside amazon's massive wish-fulfilling machine." https://www.wired.com/2014/06/inside-amazon-warehouse/#slide-1 accessed April 30, 2017.

All pickers have handheld scanners, which are carried on the "pick mod". The scanners direct the workers to the bin where the ordered items are stored. Meanwhile a red light attached to the storage location blinks to make the bin more visible. The items are picked, scanned, and then placed into a tote, which is also scanned. When a tote is filled, it travels along a conveyor system made up of ramps, long straightaways, and towering corkscrews to be prepped for shipment.

Figure 1.3 Order picking at Amazon.

Source: Marcus Wohlsen “A rare peek inside amazon’s massive wish-fulfilling machine.” https://www.wired.com/2014/06/inside-amazon-warehouse/#slide-7 accessed April 30, 2017.

Starting from 2012, Amazon purchased robots made by Kiva Systems spending $775 million to better handle largescale worldwide orders. With this technology, instead of roaming around and browsing for the items, pickers stand in their stations and collect the items that robots fetch along with the entire pod. Robots are also used for stocking.

Figure 1.4 Automated technology for Amazon.

Source: Emmanuel Amberber, "How amazon's Kiva robots shorten order fulfillment time – 30 minutes instead of hours." https://yourstory.com/2014/12/amazon-kiva-robots/ accessed April 30, 2017.

1.3 Problem Statement

The current IFW knowledge base is limited to a small group of companies, and to one company in particular, Amazon (Lang et al., 2012). Since this knowledge is proprietary the depth and complexity is unknown to the research community. The inability of physical retailers with traditional distribution operations to effectively achieve the speed and efficiency of IFWs is now well recognized. There is an immediate need for detailed analytical models which describe the flow of operations and prescribe key control decisions for IFWs. This broad based open knowledge will allow the wider development of new and powerful models for the design of the internet driven retail economy. A clear identification of IFW modelling differentiators is needed to convert traditional bulk warehousing to unit level fulfillment.

1.4 Research Objectives and Accomplishments

1.4.1 Develop Descriptive and Prescriptive Models for the Control of IFW Operations

Internet retail is generally described as the online marketing and sale of products directly to the consumer. We find that fulfillment operations in an internet retail environment are structured differently. The data for identifying these differences originated from the observational studies which was collected only viewing the operations. No access to operational data or descriptions of control logic were available to the study. The initial reactions from these visits were that the warehouses were operating in a seemingly chaotic mode, and the operations were quite unlike traditional warehousing practices (Bartholdi and Hackman, 2014; Tompkins et al, 2010). Further analysis, though, revealed to us that the warehouses were actually highly efficient and at the frontlines of a new method and operating principle in warehouse design and control. Order fulfillment time is defined as the interval between order receipt and shipment.

Accomplishments:

An analysis of the observational study allowed us to identify and describe key differentiators of IFW operations. The two differentiators found to be the most essential elements are explosive storage policy and immediate fulfillment objective. The differentiators are then used to identify decision variables, parameters and performance indicators that characterize IFWs in order to develop and formulate descriptive and prescriptive analytical models for the control of IFW operations. The models include a stocking algorithm called Uniform Random Stocking List Algorithm that is unique to the explosive storage policy and a basic picking algorithm called Narrow Band Order Picking Algorithm (NBOP) which serves the immediate fulfillment objective.

1.4.2 Simulation Confirmation of Performance Advantages of Explosive Storage Policy

A key analytical question after building a mathematical model is what the likely fulfillment time performance advantages are. Using basic storage and picking algorithms the order fulfillment time superiority of explosive storage is investigated. Model size of the simulation is determined by (i) number of unique SKUs stored (ii) number of storage bin locations and (iii) number of daily customer orders. Explosive storage policy requires considerably more resources from both facility design and information technology context.

Accomplishments:

A simulation model is developed to analyze the performance behavior of linear fulfillment time. A complex system requires a large amount of input data and generates a large amount of output data. For that reason generic simulation packages will not be able to store the output data of these complex models in an organized and structured way. A data driven simulation model is built in MS-Access/VBA platform. The simulation is set up to evaluate the response of two control variables (i) explosion ratio of incoming bulk, (ii) maximum number of stops that a picker should make.

IFW performance behavior is shown to be sensitive to the explosion ratio. Results show that increasing the explosion ratio can reduce linear fulfillment time by as much as 16%, confirming that the IFW storage policy is advantageous. The results also show that fulfillment time behavior is convex as a function of explosion ratio. The NBOP Algorithm prescribes that $\Psi_o = 0.8$ provides the shortest fulfillment times across all pick list sizes, and optimal pick list size in a picklist $\check{K}$=13Both algorithms can be tweaked for further improvements. Based on the simulation results, it can be confidently concluded that explosive storage policy outperforms traditional warehousing practices in terms of fulfillment time, warehouse space and worker utilization.

1.4.3 Develop Advanced Order Picking Algorithms to Minimize Fulfillment Time

Since fulfillment time is linear, minimizing order fulfillment time is equivalent to minimizing the picker's travel distance to complete a pick list similar to the traveling salesman model except priority is given to the earliest received orders. Ideally, one pick list can be fulfilled from one bin. The decision variables are: which orders to serve in the next pick list and which storage locations to fulfill the orders from.

In every period, the IFW receives thousands of orders to be fulfilled from a large number of bins that contain multiple items. The list of candidate picks is very large, and the solution space is described by the product of pending orders and the active inventory locations. Difficulty of a Mixed Integer Programming (MIP) Problem is said to be measured by the number of binary variables. The IFW order picking problem is NP-hard, and the integer decision space is way too large to efficiently find an optimal solution.

Accomplishments:

A MIP problem is created that minimizes fulfillment time for every order by finding the best cluster of items to generate a pick list. Solution time of the MIP is unacceptably long if not impossible. Because of this difficulty, we approach the problem by heuristics to reduce the problem size. First, the problem is narrowed down without any loss of data by generating lists with (i) order elimination, (ii) bin selection. Also, two heuristic algorithms are developed to select a seed to reduce solution time. We call this advanced picking algorithm Bin Weighted Order Fillability (BWOF). The BWOF is first tested using OpenSolver in order to confirm improvement in fulfillment and solution time. Then, the initial picking algorithm NBOP it is replaced with the BWOF in the simulation and the results are compared.

Upon completing experiments using Open Solver, it is found that about 75% of the time the BWOF algorithms found the optimal within 10% in only a couple of seconds. For the cases that the BWOF couldn't find the optimal, the solution is only slightly off from the optimal. In conclusion, the BWOF algorithm has successfully provided better solutions.

Simulation results lead us to an exciting finding about the optimal explosion rate. The BWOF algorithm suggests that optimal explosion ratio Ψ_o drops down to 0.5. The average gain from improving the NBOP to the BWOF is between 12% and 35%. For larger pick list sizes the BWOF algorithm reduces the average fulfillment times much as 50%. Besides fulfillment time advantages, this Ψ_o drop allows us to reduce labor costs in storage operations since higher explosion ratios require more handling in smaller quantities.

1.5 Research Significance

The shift to an internet economy is having significant effects on the design and operation of retail supply chains. Specifically, the disintermediation of a retail points of sale creates new paradigms and opportunities in order fulfillment. Currently, the bulk of the order fulfillment literature is based on bulk movements from distributor to retailer. In the new paradigm we see a complete elimination of the physical retailer, and bulk break-up occurring at the warehouse level. The demand of immediate fulfillment and bulk explosion require new structural models with a new set of operational objectives. This research develops these new models allowing for detailed and continuing research on the operations and decision making structure of IFWs. These advanced models are needed by both traditional omni-channel and purely internet retailers to realign with the new economy. Academic researchers are also able to formulate and optimize specific problems in this context.

CHAPTER 2

LITERATURE REVIEW

2.1 Warehouse Storage Design

A supply chain is the network of all the individuals, organizations, resources, activities, and technology involved in the creation and sale of a product or service, from the delivery of source materials from the supplier to the manufacturer, through the eventual delivery of the product or service to the end user. The three main flows of the supply chain are the product flow, the information flow, and the finances flow (Kahraman, Oztaysi 2014). For the product flow to be flawless, countless factors can be involved to be deliberately considered in every element of a supply chain. Perhaps the most effective factor is the design of a warehouse.

2.1.1 Product or Material Flows in the Warehouse Design

Gu et al. (2007) analyze the material flow problem dividing it into two parts: warehouse design and warehouse operations. The warehouse design problem is classified into five major decisions: overall warehouse structure, sizing the warehouse and its departments, determining the detailed layout within each department, selecting warehouse equipment, and selecting operational strategies. Mohsen and Hassan (2002) lists the detailed decisions as follows: Specifying type and purpose of the warehouse, forecasting and analysis of the expected demand, establishing operating policies, determining inventory levels, class formation (if class based policies are used), departmentalization and general layout, storage partition, design of material handling, storage and sorting systems, design of aisles, determining space requirements, determining the number and location of I/O points, determining the number and location of docks, arrangement of storage, and zone formation. Rouwenhorst et al. (2000) provided a graph representing strategic, tactical and operational level decisions:

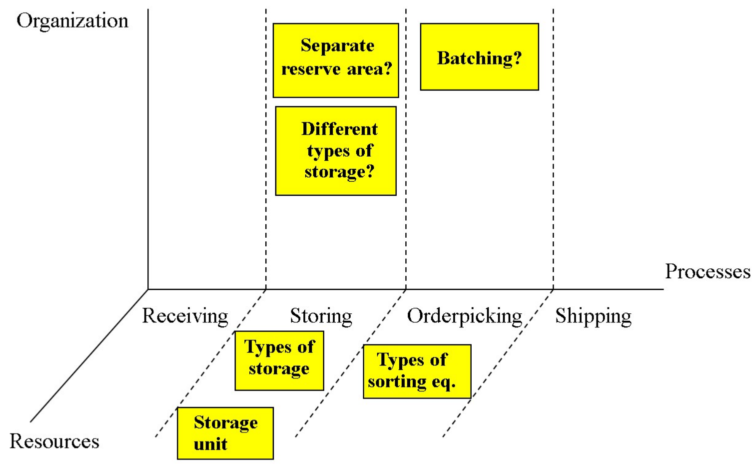


Figure 2.1 The strategic level: Long term decisions (5 years).

Source: Rouwenhorst, B., et al. (2000). "Warehouse design and control: Framework and literature review." European Journal of Operational Research 122(3): 515-533.

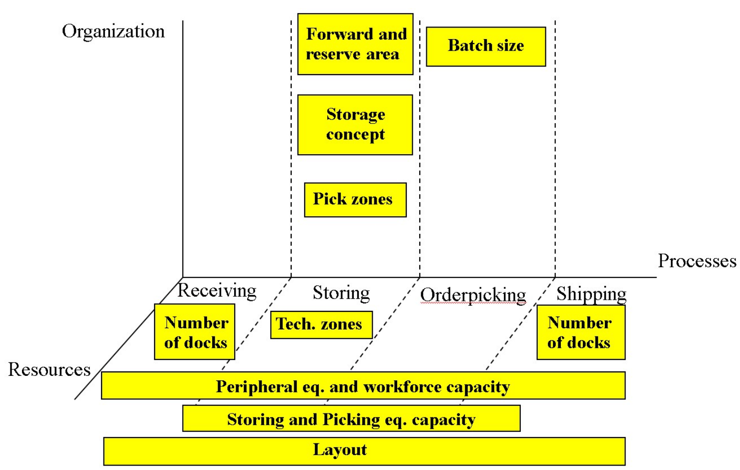


Figure 2.2 The tactical level: Medium term decisions (2 years).

Source: Rouwenhorst, B., et al. (2000). "Warehouse design and control: Framework and literature review." European Journal of Operational Research 122(3): 515-533.

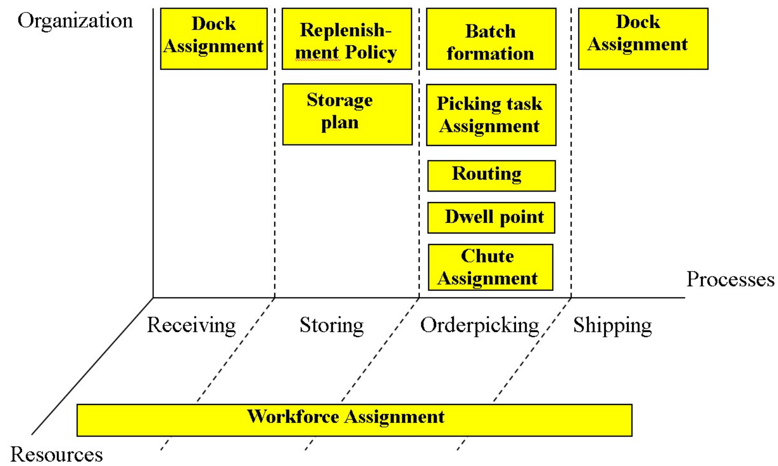


Figure 2.3 The operational level: Short term decisions (1 year).

Source: Rouwenhorst, B., et al. (2000). "Warehouse design and control: Framework and literature review." European Journal of Operational Research 122(3): 515-533.

2.1.2 Warehouse Design or the Facility Layout Problem

Warehouse design is generally called the "facility layout problem" whereby the layout is configured based on interactions among departments such as receiving, picking, storage, sorting, and shipping, among others in the facility (De Koster et al., 2007). The main objective of the facility layout problem is to reduce all non-value adding operations. Layout efficiency in the literature is generally measured in terms of material handling cost (Meller and Gau, 1996), throughput, space utilization and service level (Gu et al., 2007).

Cahn (1948) wrote the first paper in warehouse design, modelling a warehouse with fixed capacity and an initial stock of a certain product, which is subject to known seasonal price and cost variations, to assess the optimal pattern of purchasing (or production), storage and sales. In the following years, "the warehouse problem" has been discussed by Bellman (1956) and Moder and Thornton (1965) to evaluate how floor space utilization is affected by other factors such as slant angle of the pallets. They developed a mathematical model that analyzes floor space efficiency using factors such as the angle of placement of the pallets and width of the aisles, lane depth, and spacing between storage lanes.

Francis (1967) considered a continuous approximation of the storage area without considering aisle structure in order to assess travel time estimates and construction and operating costs of single and dual command cycles in multiple aisle systems. Berry (1968) developed analytic models to evaluate the total space requirement and the average travel distance in order to maximize space utilization for ten parameters including block-stacking patterns with different aisle configurations, lane depths, throughput rates, and number of SKUs. White and Francis (1971) studied the determination of the optimum size for a warehouse used to store products over a finite planning horizon under conditions of deterministic and probabilistic storage demand. The model is formulated as a linear programming problem and transformed via duality theory into an equivalent network flow problem.

Roberts and Reed (1972) compared two alternative configurations to minimize handling and construction costs. Levy (1974) presented analytic models to determine the optimal storage size for a single product with either deterministic or stochastic demand. Bassan et al. (1980) evaluated different layouts using two parameters, the number of cells in a row and the number of rows, and compared them for minimal annual cost. Handling costs and costs associated with the warehouse area and perimeter are taken into consideration to assess expressions for optimal design parameters. It has been found that, for the considered layouts, costs affecting the choice are those associated with the warehouse perimeter and material handling but not the cost of the warehouse area.

Matson and White (1981) studied the design and evaluation of storage system alternatives, including block stacking, single-deep and double-deep pallet racks, and deep lane storage to minimize costs and meet the required service level. Marsh (1978, 1979) used simulation to study space utilization for three different block stacking policies:

straight queuing, upward product set search, and downward product set search. Later in 1983, Marsh compared Marsh (1979) and Berry (1968) models using statistical analysis to determine whether significant differences exist between the two layouts; he found no significant differences between the two layouts except for size.

Roll and Rosenblatt (1983) compared a series of storage policies and their effects on warehouse capacity needs and extended the work of Bassan et al. (1980) to include the additional cost due to the use of grouped storage policies. Roll and Rosenblatt (1984) then searched for a procedure for finding a global optimal solution for a specific formulation of the warehouse design problem. Ashayeri and Gelders (1985) discussed several types of solution procedures in the existing literature and noted that it was rather hard to make a general assumption because each of the models considered a different set of assumptions.

Hung (1984) discussed the economical aspect of warehouse sizing and developed a linear programming model to determine whether such space should be leased or rented from a public warehouse or be privately owned. Park et al. (1989) studied the optimization procedure of three-dimensional, palletized storage systems and compared all alternatives for: control procedures, handling equipment movement in an aisle, storage rules, alternative handling equipment, input and output patterns for product flow, storage rack structure, component costs, and the economics of each storage system.

2.1.3 Automated Storage/Retrieval Systems

In the late 1970's, computerization led to the development of automated storage/retrieval systems (AS/RS). Multiple papers have been published to introduce the benefits of AS/RS including those by Hausman et al. (1976), Graves et al. (1977), and Schwarz et al.(1978). Almost all the papers published are for unit-load storage systems. In the 1980's, the evaluation of the performance of AS/RS was studied several times under many different combinations of assumptions. Hodgson and Lowe (1982) studied a layout problem with the placement of items in a storage rack serviced by a storage/retrieval machine for a class based assignment of storage locations policy. Azadivar (1984) applied a stochastic optimization method to determine the optimum cutoff points for the models developed previously, and compared them using computerized simulation. Later in 1989, Azadivar used a constrained stochastic optimization method with controlled parameters to represent such systems to maximize the throughput. Bozer and White (1984) studied crane travel time models for randomized storage in order to determine expected travel times for both single and dual command cycles.

A warehouse design with storage policy consideration has been discussed in Roll and Rosenblatt (1985) based on a number of studies they published on warehouse sizing and optimizing. Other aspects of automated warehousing systems have been discussed in Waugh and Ankener (1977), Karasawa (1980), Elsayed (1981), Pliskin and Dori (1982), Evans (1984), Linn and Wysk (1984), Perry et al. (1984), Chow (1986), Cox (1986), Hwang (1988), Ashayeri (2002) and several others.

Han and McGinnis (1986, 1987, 1989) analyzed rotary rack operations, carousel applications and sequencing retrievals. Lee and Hwang (1988) studied the design of a unit-load automated carousel system in which each carousel conveyor is served individually by a single storage/retrieval (SIR) device. Later in 1990, they studied continuous analytical models of travel time or both single and dual command cycles under a randomized storage policy. The existing travel-time models of automated storage/retrieval systems (AS/RS) assume average uniform velocity, ignoring the operating characteristics of storage/retrieval (S/R) machines such as the acceleration/deceleration rate and the maximum velocity, and the models are far from being optimal. Hwang and Song (1993) studied sequenced picking operations and travel time models for similar systems, and the impacts of acceleration/deceleration on travel time models for carousel systems. Pandit and Palekar (1993) minimize the expected response time of storage and/or retrieval requests using a queuing model to calculate the total response time including waiting and processing time for different types of layouts.

Malmborg and Al-Tassan (1996) developed analytic models to evaluate the performance of dedicated storage and randomized storage in less than-unit-load warehouses. In 2000, they studied interleaving models for the automated twin shuttle of the AS/RS, and in 2001 improved Zollinger's (1996) rule-of-thumb heuristics for configuring storage racks in the AS/RS design. After the 2000's, warehouse layout problems have been studied less and the number of publications has been reduced accordingly. However, the layout still contains the core of the warehousing problem. Hence, the more recent papers include layout design evaluations in order picking algorithms. Cormier and Gunn (1992), Cormier and Eng (1997), Van den Berg (1999), Van den Berg and Zijm (1999), Rouwenhorst et al. (2000) Hale and Moberg (2003), Gu et al (2007, 2010), and Gagliardi et al. (2012) discuss the literature on warehouse design problems in detail and can provide further information.

2.1.4 Recent Work

Many recent papers address various issues in warehouse optimization and design. Ozturkoglu (2011) analyzed a continuous space model for travel in a unit-load warehouse that allows cross-aisles and picking aisles to take on any angle, and he determined optimal designs for one, two, and three-cross-aisle warehouses, which are called chevron, leaf, and butterfly designs. Ozturkoglu's analysis showed that the chevron design, which is new to theory and to practice, is the best design for many industrial applications. Compagno (2012) studied designing the shape of a warehouse and compared a standard storage-handling system which minimizes the handling planar path to a design that minimizes the overall handling energy consumption. Sooksaksun et al. (2012) developed a one-step warehouse design procedure where an iterative process is run until a design with appropriate performance criteria is found which is different than a traditional two-step design where determination of the aisle layout and dimension is followed by the assignment of items for storage.

Geraldes et al (2012) developed a large mixed-integer nonlinear programming model (MINLP) to capture the trade-offs among the different inventory and warehouse costs in order to achieve a global optimal design satisfying throughput requirements. Cakmak et al (2012), in order to determine the size for a new flow type warehouse and a new u-type warehouse, developed a model that minimizes the travel path taking many parameters into account including the number of doors and their location. Lerher et al (2013) studied multi-objective optimization of AS/RSs. Time-cost- quality relations are discussed and evaluated in order to minimize travel path and cost, and to maximize throughput. Marchet et al (2013) investigated the main design trade-offs for faster deliveries, smaller order sizes, and for material handling solutions using simulation, and they proposed a comprehensive design framework.

Ekren et al. (2015) studied the determination of the best rack design for shuttle-based storage and retrieval systems (SBS/RS) under a class-based storage policy. SBS/RS is a new technology in AS/RS which has been developed for high transaction environments where mini-load cranes may not be able to keep pace with the transaction rate needed over a given number of storage locations.

2.2 Internet Fulfillment Centers

E-commerce technology is different and more influential than other technologies that have been seen in the past the century (Laudon et al., 2007). E-commerce technology progressed quickly with the development of the internet (Li and Yan 2014). However, many of the new e-commerce companies have failed or are struggling for economic survival, and the failure can be related to disregarding the logistics (Bretzke, 2000)

2.2.1 Amazon and E-retailing

Amazon, one of the first e-retailers in the world, started selling books over the internet and rapidly became the first broadly known company for any product category. Amazon was the leader in the online book market until another bookseller, Barnes & Noble, entered into online book retailing in 1997. The competition and online bargains caused book prices to fall by 15% (Bailey, 1998). Since then, many more companies entered online retailing which caused profit margins to decrease significantly. As physical product sales grow, the cost of order fulfillment also increases. As in any other form of retailing, e-retailers need to follow an aggressive policy to survive and be profitable. In an extremely competitive market with very low margins, survival is determined by the volume of sales which induces e-retailers to use two major approaches to market expansion: expanding across product lines and entering in foreign markets. (Chakrabarti _and Scholnick, 2002).

Those two major approaches to market expansion are key choices in the Ansoff matrix (Ansoff, 1957). Another method to expand the market size is providing better services to retain existing customers and attracting new customers. Competition among retailers is forcing organizations to increasingly integrate and automate their business operations such as order processing, procurement, claims processing, administrative procedures, and others (Dayal et. Al, 2001). Amazon is currently using a combination of all of these methodologies to reduce the overall operation cost while fulfilling all of the customer demands in the shortest time possible, providing the best possible customer service, distribution efficiency, and convenience.

Amazon's initial goal in regards to distribution was to eliminate the middleman in the supply chain (Lang et al., 2012). In November 1996, Amazon rented its first large warehouse in South Seattle, 93,000 square feet. Almost one year later, Amazon went cross-country to open its second warehouse in New Castle, Delaware. Online book sales grew from nothing in1995 to more than $2 billion in 2000 (Goolsbee et al., 2002). Considering that average book prices are 6.5% lower on Amazon.com and the rent expense is 7% to 10% lower than in brick and mortar stores, the "virtual" operation is not necessarily more cost-effective (Rosen & Howard, 2000).

In 2001, with growing demand, Amazon increased its distribution space from 300,000 square feet to 2.7 million and hired an additional 600 employees in its Atlanta facility. As of May 2017, Amazon operates 141 active distribution center buildings around the world (including fulfillment centers, sorting centers, returns centers, specialty centers, redistribution centers. These facilities total 91.9 million square feet. Future plans include 16 new distribution center buildings (including new U.S. sorting centers) exceeding 9.2 million square feet to be opened in North America.

Amazon has been constantly looking for improvement to be one step ahead of its competitors. Omni-channel environments where customers shop online and offline at the same retailer are increasingly ubiquitous and have important new implications for demand generation and operational efficiency (Bell et al., 2013). Omni-channel distribution centers serve the online customer through both direct same-day shipping and store-pickup (Rigby

2011, Levy et al 2013). According to Forrester Research the market will grow from $231b in 2013 to $370b in 2017 on CAGR of 10 percent, proving that the internet retail is the fastest growing market. Retailers are attracted to omni-channel strategies because online and offline channels differ in their ability to deliver information and execute product fulfillment (Coughlan et al. 2006). Therefore, retailers of all types and in all locations increasingly interact with consumers through multiple touch points (Brynjolfsson et al. 2013).

2.2.2 Related Work

Linden, Smith and York (2003), published a paper to give recommendation algorithms to e-retailing companies to provide an effective form of targeted marketing by creating a personalized shopping experience for each customer comparing the three main approaches: traditional collaborative filtering, cluster models, and search-based methods with their algorithm which is called item-to-item collaborative filtering.

2.3 Random Storage Policy

Storage is the main function of a warehouse. Once the design of the warehouse is decided to serve the most efficient way to the purpose what the warehouse will be used for, there will be three more fundamental decisions are left: what SKU's will be carried, how frequently for a SKU should be replenished, and what policy should be adopted to store the SKU's in order to optimize the main objective of the warehouse (minimum cost, shortest order picking cycle time, shortest fulfillment time, etc.). The last decision is generally referred as product allocation problem.

De Koster, Le-Duc and Roodbergen (2007) classify storage assignment policies as: random storage, closest open location storage, dedicated storage, full turnover storage and class based storage. Gu et al. (2007) mentioned several examples of how production allocation decisions are made and gave some examples. One of which is allocation of a certain size of space to a certain customer and keeping only the items that will be sent to that specific customer. Another example is dedicated storage where a certain space is reserved for a specific SKU and even if there is no inventory of that item, space cannot be used for storing another SKU. Examples vary with the need of the warehouse.

2.3.1 The Storage Location Assignment Problem

The storage location assignment problem (SLAP) is to assign incoming products to storage locations in storage departments/zones in order to reduce material handling cost and improve space utilization (Gu 2007). Frazelle (1990) lists three main stock location assignment strategies as dedicated storage, randomized storage and class-based storage. Gu 2007, Frazelle, 2002 expands the definition in later studies. They state that the three most frequently used criteria in the case where there is information on SKU's are popularity (Turnover-Based assignment), maximum inventory (Class-Based Turnover assignment) and Cube-Per-Order Index (COI, which is defined as the ratio of the maximum allocated storage space to the number of storage/retrieval operations per unit time). And the most frequently policies used when there is no information on incoming products are closest open location, farthest open location, random storage and longest open location. Petersen (1997) defines random storage as all empty locations have an equal probability of being filled. De Koster et al. (2006), Roodbergen (2001), Le-Duc (2005), Dukic (2004) investigate some certain scenarios in random storage environment.

Literature in the area is very rich and randomized storage policy (RSP) has been used commonly for its ease of use and accuracy on travel time estimation. RSP commonly means allowing multiple SKU's being assigned to the same location over different time periods to increase the warehouse space utilization. The advantage of RSP is cube utilization and warehouse efficiency, the disadvantage is storing quantities of a single item in many different location in the warehouse making inventory control and picking operations a lot more complicated which requires using computerized systems heavily (Ross, 2015). Besides, RSP comes with various assumptions and each research in the literature adopts some of these assumptions based on the case they are working on. The most common assumption is unit load storage where a unit can be thought as a pallet or a storage unit that only one type of on item is stored.

2.3.2 Unit-Load and Less-Than Unit Load Policies and Forward Reserve Problem

Bozer et al (1985) is the first to suggest to split a pallet for more effective picking operations for forward-reserve problem where only some of SKU's are stored in the forward area to reduce the material handling, which later is improved and detailed by Hackman and Rosenblatt (1990) to determine which SKU's should be assigned to forward area. Later Frazelle et al. (1994) extended the problem modelling the size of the forward and reserve areas to minimize the cost of material handling for order picking and replenishment.

Malmborg has made a great contribution to storage, warehousing and inventory systems literature. In one of the papers written in 1998, Malmborg and Al-Tassan are extending the existing literature on the unit load warehousing systems to less than unit load systems and investigating it for dedicated storage, random storage, a combination of closest open location with randomized storage and Cube per Order Index. However, they did not compare their results with unit load approach. Malmborg and Al-Tassan (2000) presents a mathematical model to estimated space requirements and order picking cycle times for less than unit load order picking systems that uses randomized storage.

2.3.3 Shared Storage Policy and Some of the other widely held policies

Shared storage strategies do not reserve slots for specific items, which makes them more convenient when stock levels change over time (Kovacs, 2011). Goetschalckx and Ratliff (1990) the term shared is described as using the same location for sequentially storing different SKU's over a planning horizon, but not always concurrently and show that a duration-of-stay–based policy under shared storage strategy is optimal under an assumption of perfectly balanced inputs and outputs. Cormier and Gunn (1992) states that shared storage policies offer excellent potential for travel time and rack size reductions. The most widely used shared storage strategy is the class-based storage strategies (Hausman et al., 1976; Petersen and Aase, 2004). Kulturel et al. (1999) compared two shared storage assignment policies in an Automated Storage/Retrieval System (AS/RS) by using computer simulation and concluded that the turnover-based policy generally outperforms the duration of stay-based policy. We found little recent research on the topic, but later highlight commingled storage as a key differentiator of IFWs.

Turnover-based storage is studied by many researchers (Jarvis and McDowell, 1991; Caron et al., 2000). Pohl et al. (2011) investigated turnover-based storage policies and warehouse designs with non-traditional aisles. Automated Storage/Retrieval Systems (AS/RSs) is the most common research topic on class-based storage and is usually concerned with determining the number of classes and the boundaries of the warehouse zones. Graves et al. (1977) and Kouvelis and Papanicolaou (1995) derive analytical solutions for class boundaries with two or three classes; and Rosenblatt and Eynan (1989) and Eynan and Rosenblatt (1994) address the n-class case. De Koster et al., 2007; Gu et al., 2007 analyses class-based storage studies in their survey and can be referred for further reading in various storage policies that have not been mentioned in this paper in detail.

2.3.4 Recent Research

Based on a research made by Battista et al. (2013) even though dedicated and random storage policies has been studied for decades, and theoretically dedicated storage policy has found to have an advantage on improving the efficiency (Goetschalckx and Ratcliff, 1990 and Thonemann and Brandeau,1998), there is still not a procedure for systematically analyzing the requirement and designing a warehouse to meet the operational need using the most economic technology (Rowley, 2000; Croom et al., 2000; Pessotto, 2009, Pecchiar, 2012; Azevedo and Carvalho, 2012).

Roy et al. (2012), brought up a new modelling approach for estimating storage/retrieval transaction times in warehouse systems using random storage and closest open location load dispatching that estimates intervals between consolidations of the active storage envelope defined by the most remote occupied storage position in a warehouse.

2.4 Order Fulfillment Objective

In online retailing, the main objective is optimizing the order fulfillment time while minimizing the relative supply chain costs. In comparison to brick and mortar stores, e-retailing has the advantage of being able to accommodate excessive amounts of supply of a large variety of products. The challenge is, the order needs to be delivered to the end customer with in a time frame of selected shipment method, which might be a few hours after purchase. Thus, the “advantage” becomes a big optimization problem in the supply chain. Torabi et al. (2015) list e-retailing decisions as follows: Source of fulfillment having the luxury of aggregating inventory virtually, and being able to fulfill the orders from different locations based on the cost, distance and quantity, Temporary shortage allocation being able to allocating shortages to each customer based on their shipment preferences, Planned substitution being able to sell substitution items even before they are replenished in the warehouse, Order consolidation, order picking, order routing grouping batching or splitting orders to individual items to minimize fulfillment time. Due to the opportunity of converting these decisions into advantages, more traditional retailers are venturing online (Boston Consulting Group, 2000).

Reviewing the existing literature, we realize that it is indisputable that the work of study attains a remarkable growing interest. However, most of the studies focus on descriptive and qualitative models. In this dissertation, we compare a traditional warehouse and a chaotic warehouse based on the respond time of an order, which is the time window between the demand arrival and order fulfillment. Torabi et al. (2015) sketched the window decision opportunity as follows: In the traditional retailing, when a customer arrives, his order needs to be fulfilled right away and there are not too many decisions that the retailer can make. However, in e-retailing, window of opportunity gives us the time to think how to best fulfill the order. Industrial engineering perspective implicates that the shorter the order fulfillment time is, the more requests are satisfied, and assuming there is enough demand in the market. For that reason, time management becomes vital.

2.5 Warehouse Order Picking Algorithms

Order picking means clustering, scheduling, and retrieving items from a warehouse in response to a specific customer request. Order picking is the most costly activity for almost every warehouse and estimated to be as much as 50-75% of the total warehouse operating expense (Coyle et al.,1996 De Koster et al, 2007). In addition, it is typically one of the most time-consuming activities (Tompkins et al. 2003). For these reasons, warehousing professionals consider order picking as the highest-priority area for productivity improvements (De Koster et al., 2007). In a COF warehouse this expense could be significantly higher for these two reasons: in a COF warehouse items are stocked in multiple locations and commingled requiring additional time/effort to complete the pick, in a traditional warehouse storage occurs at the bulk level but as a result of the explosive storage policy both COF storage and picking activities are labor intensive. Recent trends both in manufacturing and distribution in order-picking management is to move to smaller lot-sizes, point-of-use delivery, order and product customization, and cycle time reductions and Many smaller warehouses have been replaced by fewer large warehouses to realize economies of scale.

Order picking policies differ based on the needs of the company, however it is possible to generalize policy decisions that determine the efficiency the order picking operation under three major approaches: storage policies and routing policies SKU pick policies (Petersen et al., 2004) The most commonly used objectives are to maximize the service level subject to resource constraints such as labor, machines, and capital and to minimize the total picking time (Goetschalckx and Ashayeri 1989). In any case, the problem of routing order pickers in a warehouse is actually a special case of the Travelling Salesman Problem (Karp et al., 1985).

2.5.1 Routing Policies

Routing policies determine the rules to create a pick list and the sequence of SKU's to be picked. It has been investigated for decades by many researchers. Since the objective is creating the shortest path, the goal for routing policies is to minimize the total distance travelled. Ratliff and Rosenthal (1983) and Goetschalckx and Ratliff(1988) developed algorithms for routing pickers in a rectangular warehouse. For a restricted set of layouts, researchers have developed efficient routing policies that find a shortest route (De Koster and Van der Poort 1998, Roodbergen and De Koster 2001). Petersen and Schmenner (1999) state that testing of routing and storage policies and their interaction show statically significant differences, therefore in increased efficiency can be gathered selecting a combination of certain routing and storage policies. Therefore, warehouses generally prefer heuristics over optimal routing. Hall (1993), Petersen (1997), Petersen and Schmenner (1999), and Roodbergen and De Koster (2001) discuss several heuristics. Elsayed (1981, 1983) studied the assignment of picks to pickers in an AS/RS environment.

De Koster, Le-Duc and Roodbergen (2007) create a matrix that include five main operating policies: routing, storage, batching, zoning and order release mode based on Goetschalckx and Ashayeri (1989). Matrix shows the level of complexity of order-picking systems, measured by the distance of the representation meaning, the farther a system is located from the origin, the harder the system is to design and control.

De Koster, Le-Duc and Roodbergen (2007) list commonly used heuristics for routing order pickers starting with the S-shape (or traversal) heuristic, meaning that any aisle containing at least one pick is traversed entirely and from the last visited aisle, the order picker returns to the I/O point. Another heuristic is the return method, where an order picker enters and leaves each aisle from the same end visiting only the aisles with picks.

The midpoint method divides the warehouse into two areas. Picks in the font half are accessed from the front cross aisle and picks in the back half are accessed from the back cross aisle. The largest gap strategy is that an order picker enters an aisle as far as the largest gap within an aisle. The gap represents the separation between any two adjacent picks, between the first pick and the front aisle, or between the last pick and the back aisle. Petersen (1997) adds composite and optimal routing methods to the four methods mentioned above and compare them in a random storage warehouse. Commercial warehouse management systems typically prescribe one or more of three well known methods: "S-shape strategy", "Return strategy" and "Zig-zag strategy" (Moeller, 2011). In addition to those methods above, aisle by aisle (Petersen ,1999 and Le-Duc and De Koster, 2004) and composite, a combination of S-shape and return policies is proposed by Petersen (2002) and Le-Duc and De Koster (2004).

2.5.2 Zoning and Batching

Another approach to order sequencing is zoning. In that case, picking area is divided into zones and pickers are allocated within zones. Each picker is assigned a picklist in their assigned zone. Advantages of zoning might be traversing only a small zone, becoming familiar with the stored items within their assigned zone, reduced traffic congestion (De Koster et al. 2007). The topic is discussed in the literature. Speaker (1975) describes the use of zoning in bulk order picking.Sharp et al. (1991) compares the effects in single-order-pick, sort-while-pick, and pick-and-sort systems. Petersen (2002) studies the the zone shape and its effects. Chia Jane (2000) proposes quite a few heuristics to balance picker workload. De Koster (2004) models a zoned pick-and-pass system where each picker retrieves item(s) in a zone and pases the tote to the next zone until the order is completed.

Le-Duc and De Koster (2005) examine the optimal number of zones using mathematical programming and extend the work later on (2012). Dallari et al. (2009) develop a methodology to generate a new classification in order picking systems working on over 68 distribution centers that have been recently built in Italy. The results of the critical analysis allowed developing a design methodology to choose the most suitable OPS Yu and De Koster (2009) proposes an approximation model based on queuing network theory to analyze the impact of order batching and picking area zoning on the mean order throughput time in a pick-and-pass order picking system. Wang et al. (2013) state that synchronized zone order picking system is one of the effective policy to improve the productivity and study the imbalanced workload, built a mathematical model to estimate the throughput time and the workload of each zone. Glock et al. (2015) present an approach to model worker learning in order picking in manual warehouses and resulted that it is beneficial to assign workers with the lowest learning rate in the workforce to the fast moving zone to gain experience.

Order batching is a method of grouping orders to attempt to reduce travel times. Sharp et al. (1991), the two criteria for batching are the proximity of pick locations and time window batching. It also appears in the literature quite a lot of times, however the review in this dissertation will be held briefly since it is not used in our model.

Order batching is generally considered as an NP-hard problem. For that reason operation research techniques and heuristics are mainly used to cluster the orders. De Koster et al. (1999) performed a comparative study for multiple-aisle to picker-to-part systems. Elsayed et al. (1981, 1983, 1989), Hwang et al. (1988) investigated order batching in manual picking systems. Tang and Chew (1997, 1999) Le-Duc and De Koster (2003, 2007) studied the time window order batching in manual warehouses. Elsayed

(1993), Elsayed and Lee (1996) and Won and Olafsson (2005) extended the literature adding a cost analysis by assigning a penalty for late fulfillments.

For further reference, Potts et al (2000), Gu et al. (2007), De Koster et al. (2007) and similar review papers mentioned above can be read.

2.5.3 Travel Time Estimation and Probabilistic Approximation

Travel time estimation has been studied by many researchers. The goal is benefiting mathematical modelling and other operations research practices combined with statistics to calculate the order fulfillment time using the link between the distance travelled and time it takes to collect an order. Two types of travel distance for order picking are used: average picking route distance and total travel distance.

Optimization models in warehouse design, and particularly in order-picking models, typically assume that cycle times are a linear function of travel distance (Petersen and Aase, 2004). There has been a significant volume of research in this area, and a great deal of progress has been made in planning storage policies and order pick lists to minimize the travel length or equivalently minimize the travel time (De Koster et al, 2007; Gu et al 2007). This includes models that factor in turns, aisle crossings, and queuing approximations of congestions.

For further reference, following papers may give a good insight: Bozer and White (1984), Bassan et al. (1980), Francis (1967), Kunder and Gudehus (1975), Larson et al. (1997) and Pandit en Palekar (1993), Hall (1993), Ratliff and Rosenthal (1983), De Koster et al. (1998), Jarvis and McDowell (1991), Le-Duc and De Koster (2004, 2007), Bozer and Cho (2005), Roodbergen and Sharp (2008), Chen et al. (2013, 2014).

A common approach in order picking is to derive a probabilistic approximation of the travel time assuming a fixed travel speed. These include works by Roodbergen et al. (2008) and Le-Duc and De Koster (2005), which are applicable to rectangular warehouses with an S-shaped routing heuristic. Simulation modeling has been used by several researchers to optimize and or evaluate order picking policies, almost all use a linear time model. Peterson (2000) and Petersen and Aase (2004) evaluated multiple picking policies under varying operating conditions.

2.5.4 Recent Work and Further Reading

Gademann and Velde, 2005; and Chen and Wu, 2005 use a location closeness partitioning algorithm to make a pick batch decision, while Pan and Wu (2012) and Chen et al (2013) derive an order picking throughput model with multiple pickers and aisle congestion.

Guo, Yu and De Koster (2016) present a travel distance model considering the required space consumption by storage zoning in comparing the performance of different storage policies for a unit-load warehouse. Rao and Adil (2017) develop analytical travel distance models for class-based and full turnover storage policies under across-aisle, within-aisle and a newly proposed hybrid product placement schemes in unit-load warehouses and present the accuracy compared to simulation results. Li, Huang and Dai (2017) investigate a joint optimization problem involving with order batching based on similarity coefficient between orders and picker routing with local search. Pang and Chan (2016) propose a storage location assignment algorithm based on data mining in a randomized warehouse.

Issues in design and control of order-picking processes in particularly are mentioned in Goetschalckx and Ashayeri (1989), Sharp et al. (1991), Roodbergen (2001) and Wäscher (2004). An extensive bibliography on order-picking systems is gathered in Goetschalckx and Wei (1994) and Roodbergen (2001). Gu et al (2007) provides an extensive review of order picking algorithms and identifies batching and routing as the main areas of research focus. Bartholdi and Hackman (2014) describe order picking as a special case of the travelling salesman problem (TSP). Arc travel times are the key data in TSP and most of the literature assumes a linear activity time. For the methods to be more applicable there is a need for more accurate task time estimates.

CHAPTER 3

INTERNET FULFILLMENT WAREHOUSES

Internet retail is generally described as the online marketing and sale of products directly to the consumer. The complementary physical process is fulfillment, which typically involves a large fulfillment warehouse and a parcel delivery network. Both of these activities are part of the traditional distribution process, but we that in an internet retail environment they are structured differently. This chapter provides detailed analytical insights into internet fulfillment warehouses (IFWs), facilities that have been designed and built exclusively for satisfying online retail sales. The insights are synthesized from observational visits made by the authors to two IFWs operated by a pioneering online retail company. The initial reactions from these visits were that the warehouses were operating in a seemingly chaotic mode, and unlike traditional warehousing practice as documented in classical textbooks (Bartholdi and Hackman, 2014; Tompkins et al, 2010). Further analysis, though, revealed to us that the warehouses were actually highly efficient and at the frontlines of a new method and operating principle in warehouse design and control. We concluded that traditional warehousing methods are evolving into new methods which better leverage information technology to efficiently serve the new online retail driven supply chain economy.

The current IFW warehouse knowledge base is limited to a small group of companies, and to one company in particular (Lang et al., 2012). Since this knowledge is proprietary, the depth and complexity is unknown. Some retailers operate in an Omni channel mode in that they serve both the online customer through both direct shipping and through store-pickup delivery (Rigby 2011, Levy et al 2013). For a purely online retailer, though, there is only one delivery mode direct shipping. Currently, the largest online retailer operates over 360 active distribution facilities around the world including IFWs, return centers, specialty and redistribution centers. These facilities total over 140 million square feet of space with 250,000 employees. Nynke et al (2002) suggests that the two main observable aspects of warehouse complexity are the technologies used and number of SKUs processed. In this context there are two types of IFW warehouses. Man-to-Part: Similar to a classical configuration in that storage racks are stationary and the worker moves to the bin location and Part-to-Man: The storage racks move, usually by a robot swarm, and bring the bin to the stationary worker. In a classical Part-to-Man the pick occurs before the move, whereas in an IFW the entire rack is moved and the pick is done after the move. This research focusses primarily on the Man-to-Part type.

A warehouse process and data/decision flow diagram are created. Although the main operations such as storing and picking in both warehouses were the same, the way they are carried out is found to be quite different. We identified and described key differentiators of IFW operations. It was concluded that the primary differentiator is an explosive storage policy which breaks up and distributes received bulk to multiple storage points in the warehouse. The differentiators are used to develop and formulate an analytical model of IFW product flows. Specifically, algorithms for the explosive stocking process and real time order picking process are presented. The performance objective of these algorithms is to reduce a linear fulfillment time metric. A simulation model is built to examine the performance behavior of the explosive storage policy on an experimental problem. This is used to confirm the advantages of the explosive storage policy in general.

3.1 Observational Study

The observational study of the two IFWs were both Man-to-Part and between 3 and 8 years old. During the visits no access to operational data or descriptions of control logic were available. The observational perimeter was described by the receiving, bulk unpacking, stocking, and order packing/shipping activities. Additionally, several descriptive articles and publicly available videos of IFW operations were reviewed. After each visit a flow model was created, which was then sequentially updated.

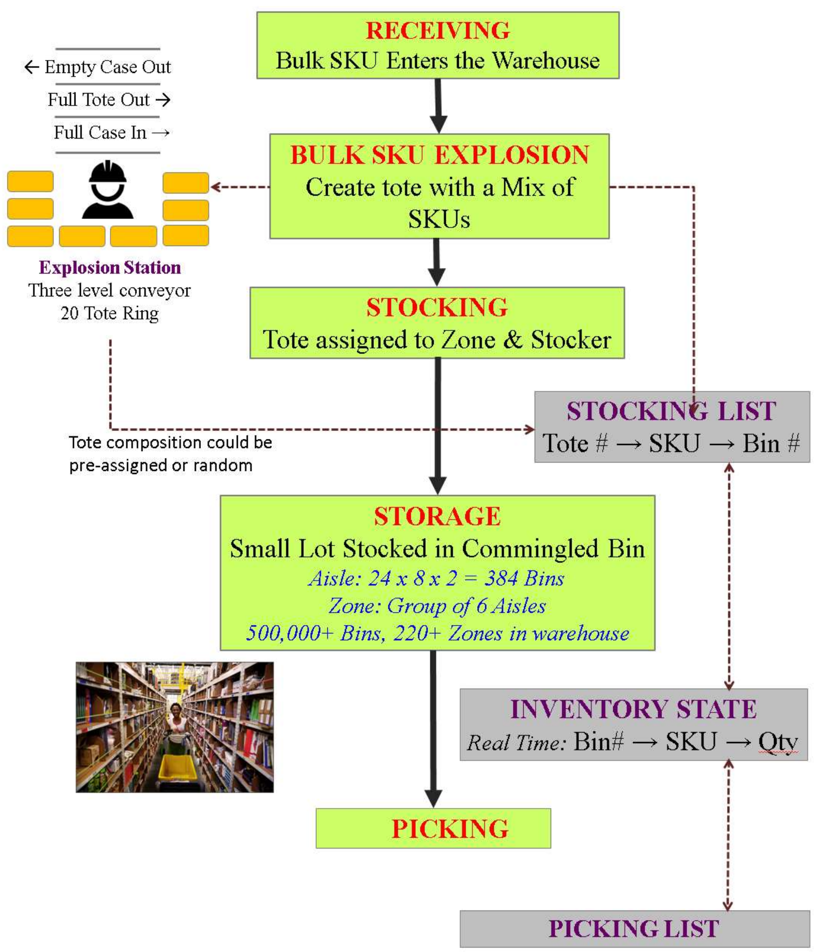


Figure 3.1 Receiving and stocking process.

Receiving: The routine at an IFW begins with receiving of single or bulk items. Traditional warehousing storage policies generally require storing SKUs to locations in bulk except for the forward-reserve configuration where the most commonly retrieved products are stored in the forward area and rarely retrieved products are picked from the reserve area (Manzini, 2012). The forward area of a warehouse functions as a warehouse within a warehouse, so that order picking can be concentrated within a relatively small area where all bulk items are stored in the reserve area (Bartholdi and Hackman, 2008). In this sense, IFWs act like a massive forward area where bulk items are opened, inspected, scanned, and put into barcoded totes for storage.

Bulk SKU Explosion: When splitting bulks into lots, receiving associates benefit from using a three-level conveyor system where full boxes are received at the bottom layer, totes filled with items from the box are placed in the middle layer which is connected to a network of conveyor belts, and emptied boxed are put on the top level to be recycled.

Stocking: Conveyor belts carry totes filled with mixed items to a zone where storage density is lower than others. Zones are assigned based on a series of algorithms which aim to maximize the space utilization and minimize storage and order fulfillment time.

Storage: Stockers pick up full totes, unload them, and place items randomly into the first available slot in the zone. Items are simply shelved randomly where they fit. The most common storage design is split shelves with adjustable dividers. Storage slots are referred as bins. The stocking process is not complete until the stocker match the item barcode with the storage unit. The location of every single item in the Fulfillment Center is known, and can be tracked by computers.

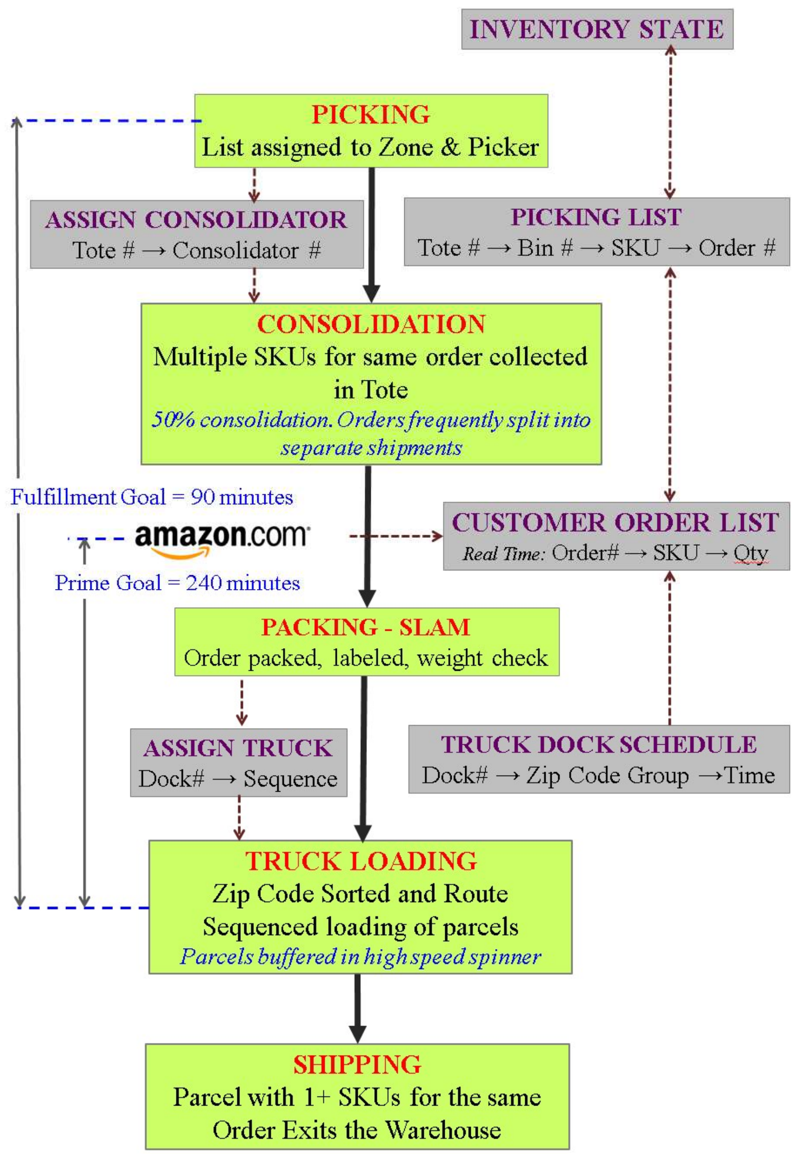


Figure 3.2 Picking, consolidation, and truck loading assignment.

Picking: The picking process is more complex. Unfulfilled customer orders are split based on SKU number. Associated bin locations that meet inventory requirement for each SKU is listed. Pick lists are created with the closest stored items and assigned to pickers. All pickers have handheld scanners which direct the workers to the bin where the ordered items are stored. Meanwhile a red light attached to the storage location blinks to make the bin more visible. Picked items are scanned, and then placed into a tote, which is also scanned.

Consolidation: When a tote is filled, it goes to the consolidation area. If the picked product is the only item in the customer order, it is sent for shipping. Otherwise, there are more decisions to make. If other items in the order are in the same warehouse all items are combined together in a divided portion of the consolidation area to be sent in one box. If not, items are either sent to the customer in multiple shipments or they are sent to another warehouse to be combined with other items.

Packaging & SLAM: Most packaging decisions such as packaging type (box or envelope), size, protection (bubble wrap, air pillow, etc.), and even the tape size are made by computers. However, packing associates are allowed to override the computer. Packed items are then put back on a conveyor where robots print the shipping label and attach it to the package, check if the weight matches the order and control the quality of packaging. If everything is satisfactory the package is carried to the truck loading zone.

Truck Loading & Shipping: Based on the destination, the truck loading zone is divided into the docks where loading/unloading is done. The conveyor belt that leads to the loading dock is made of vertical bars that are controlled digitally. Barcode readers scan the package label, and time the conveyor portion to eject the package when it reaches the truck that should be loaded.

Inventory Flow Timeline: The inventory flow timeline from point of receipt to shipment is shown below. Schematically the flow appears to be identical to a traditional warehouse but in IFWs the actual operations are quite different. The overall timeline is much shorter and both stocking and fulfillment time are measured in hours. Furthermore, due to large number of stocked SKUs and high-volume throughput, inventory time is limited to better manage the warehouse size. We estimate that the inventory turnover of an IFW is much higher than a traditional retail warehouse.

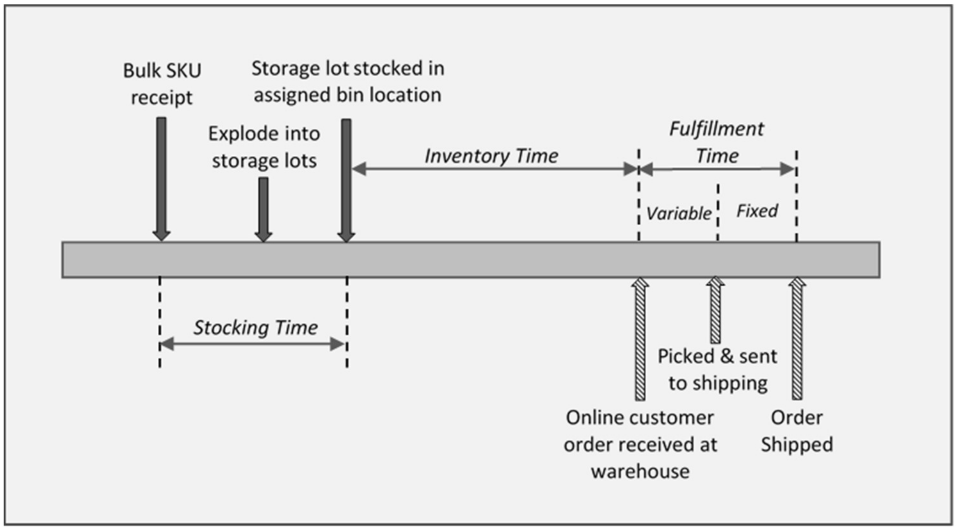


Figure 3.3 Inventory flow timeline in an IFW.

3.2 Key IFW Structural Differentiators

From the observational visits several physical design and operational insights unique to IFWs were identified. We find that an efficient IFW is differentiable from traditional warehouses by the following characteristics:

3.2.1 Explosive Storage Policy

Traditional warehouses store a SKU either in a set of contiguous locations dedicated to the SKU or a random slot for each arriving bulk. Locations are then selected using either a volume based or class based approach (Petersen and Aase, 2004). In either case, at any point of time the actual number of locations where a specific SKU is stored is few (less than 10). This approach preserves the bulk load, and the breakup into units is integrated into order picking activity. In contrast, in an IFW the incoming bulk load is immediately broken into unit SKUs upon arrival at the IFW and there is no bulk storage. The exploded units are then dispersed into a large number of storage bins with each having one or more units of the same SKU (as shown in figure 9). The bins could be random or prescribed by a rule, and each bin could receive several units of the same SKU.

We describe this as an explosive storage policy and define it as: The bulk SKU is exploded into E storage lots of one or more units such that no lot contains more than 10% of the received quantity. The lots are then stored in E non-contiguous bins anywhere in the warehouse. In a traditional policy E=1, while in an explosive policy E>10. Note that in a traditional policy several contiguous locations maybe assigned to a SKU and this should not be considered an explosion.

Let $i \in N$ be the set of unique items or SKUs stored in the warehouse. Let E_i be the explosion factor and V_i the current total warehouse inventory for i, and L_i the number of unique bin locations where it is stocked. Then we introduce the following measures:

$$\textit{Explosion ratio for product } i = \Psi_i = L_i \,/\, V_i \tag{3.1}$$

$$\textit{Warehouse Explosion Ratio} = \Psi_o = \frac{\sum_{i \in N} L_i}{\sum_{i \in N} V_i} \tag{3.2}$$

Note that L_i is not simply equal to E_i. Since batches of the bulk are arriving at some interval, every explosion will send the lots to both existing and new locations. At the same time fulfillment is occurring, as a result L_i is changing constantly and Ψ_i is time variant. The overall warehouse explosion ratio is then derived from inventory weighted function.

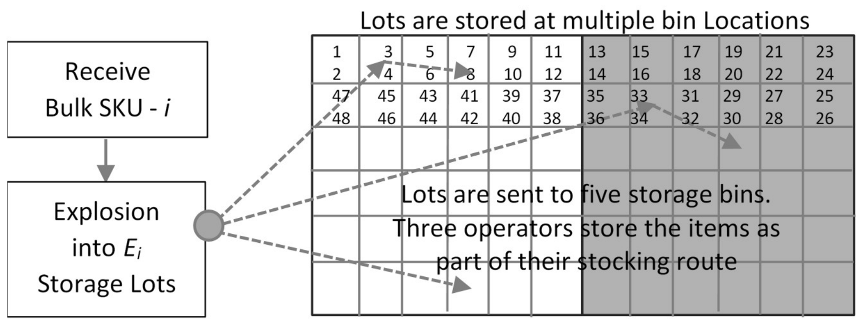


Figure 3.4 Explosive storage of bulk SKU to multiple bin locations.

Since Ψ_i is time variant, target and range values usually refer to the mean. For the case where E_i is the same for all items then the mean Ψ_i is also the same and equal to the overall Ψ_o ratio. In the extreme case where each unit of V_i is stored in a different location then Ψ_i =1. In a traditional warehouse with randomized storage we can expect at most 3 to 4 storage locations with Ψ_i <0.01, whereas in an IFW the likely range is 0.10< Ψ_i <0.50. In the design of the IFW storage policy and the target performance levels, Ψ_o and the associated Ψ_i are critical parameters. These in turn are related to the explosion factors E_i, which are therefore strategic decisions in the IFW design problem.

3.2.2 Very Large Number of Beehive Storage Locations

In traditional warehouses received items are stored in large volume locations with multiple bulk loads of a single SKU. This facilitates the subsequent shipment of the bulk quantities to retail points. In an IFW warehouse, however, the strategy is to store SKUs in unit quantities. , storage locations are therefore configured into small bins. Traditionally, bins have been used in a forward picking area for immediate fulfillment, and the allocated area is relatively small. In contrast the entire IFW warehouse is organized into racks that are divided into many small bins in a sort of beehive pattern. A million square-foot IFW could have a million bin locations. A similar sized traditional warehouse may have only 10,000 locations. This is the most apparent physical difference of an IFW.

Let $b \in M$ be the storage locations in the warehouse, then one motivation for having a large number of locations is seen from the bounding effect of M on Ψ_o. If a location could stock only one SKU at a time, then $\Sigma_i L_i \leq M$. Both Ψ_i and Ψ_o are upper bounded by this constraint. A very large M will allow the IFW to achieve higher explosion ratios.

3.2.3 Bins with Commingled SKUs

Shared storage policies have been used in traditional warehouses and have been studied in the literature. However, the term shared is described as using the same location for sequentially storing different SKU's over a planning horizon, but not always concurrently (Goetschalckx and Ratcliff, 1990). One of the most radical differentiators of an IFW, is that multiple SKUs are simultaneously stored in the same bin. We label this as commingled storage since the commingled SKU's are arranged in an unorganized way within a bin. The picker needs to visually identify the SKU against images and identify codes provided on a hand-held tablet. Intuitively, this appears to be an inefficient arrangement, since classical warehousing practice recommends easy and reliable identification of SKUs for efficient picking. Again, a likely motivation is the effect of commingling on the Ψ_o upper bound introduced in Section 3.2.1 If there is no limit on the number of SKUs commingled in the same bin then $\Sigma_i L_i \leq NM$ or $L_i \leq M$. If we make the realistic assumption that the physical bin size limits the maximum number of commingled SKUs, for example 5, then the bound is $\Sigma_i L_i \leq 5M$. Clearly, commingled storage allows for higher explosion ratios. Figure 3 shows examples of a commingled bin and the arrangement in rows and racks. Clearly, commingled storage allows for higher explosion ratios.

Figure 3.5 Bins with commingled and randomly arranged SKUs.

Source: Retrieved May 1, 2017, from http://www.bluemaize.net/im/appliances/amazon-warehouse-6.jpg.

In combination the first three differentiators have the effect of organizing the entire IFW warehouse like a forward picking area. This gives the observational view that it is seemingly operating in a sort of chaotic mode. The concept of chaos and efficiency occurring concurrently has been mentioned previously in the operations management literature. Bartholdi et al (2009) evaluate a special case of a bucket brigade assembly line where the convergence condition (workers are sequenced from most-slowed to least-slowed) is not met, as result handover points are randomly located and the system is chaotic. They note that a chaotic assembly line will appear to start and to complete products at random-even though the assembly line is completely deterministic. IFWs exhibit SKU dispersion maps that change constantly while SKU storage/pick routes are almost always unique. But the warehouse is under the full deterministic control of the central IFW controller, which is usually very sophisticated and has full transactional knowledge of every SKU movement. This could be characterized as a spontaneous emergence of balance leading to high throughput (Bartholdi et al. 2001).

3.2.4 Immediate Fulfillment Objective

Traditional warehouses operate in a batch mode in that at the start of the day or week several customer orders are pending, the tactical objective then is to fulfill these orders during the day or week. In an IFW orders arrive continuously and are transmitted to picking immediately. For the most aggressive online retailers the fulfillment goal is for same day shipment and possibly next day delivery. The entire IFW, and not just a forward area, is configured as a fast pick zone. This strategy allows them to be highly competitive against a physical retail stores. Often the delivery date has already been promised to the customer when the online order was placed, this in implies little flexibility in fulfillment time delays.

The IFW pick planning window is therefore much shorter, and typically fulfillment times are less than half a day or 4 hours. Outbound trucks leave the warehouse at regular intervals during the day. Let $\check{T}$ be the truck departure interval, then the real time planning window is a fraction of $\check{T}$ since ideally a customer order could ship out on the next truck. Our observations were that this focus on fulfillment time dominated the attitude of all workers at the IFW.

3.2.5 Short Picking Routes with Single Unit Picks

Order picking efficiency has been a key decision process in warehouse operations, and the pick list decision problem is focused primarily on travel time minimization. In an IFW most orders are for only a few units and in most cases for only a single unit. This is explained by the similarity of online customer demand to physical retail demand behavior. The efficiency gains of combining multiple orders for the same SKU are usually not exploitable in an IFW, except when orders arrive within a few hours of each other. The assumption is orders are arriving continuously with a fulfillment objective, as a result waiting to accumulate orders for the same SKU would adversely affect performance.

Typically N is very large and the arrival time between orders for the same SKU is often longer than the order pick planning window. It was observed that when orders include multiple SKUs, an IFW creates a separate order for each SKU. The assumption that a customer order is for a single SKU therefore holds. It was also observed that picked items for the same order are not necessarily aggregated into a common shipment.

The explosive strategy creates an efficient picking solution whereby multiple customer order SKUs are stored in close proximity. As E_i is increased L_i also increases and a customer order can be picked from any of the Li locations. Given that hundreds of orders are active, the probability that a small number of orders, maybe 10 or 12, can be picked from tight picking area, maybe a single row, also increases. A very short pick route can therefore fulfill several orders, and potentially multiple order SKUs could be in the same bin.

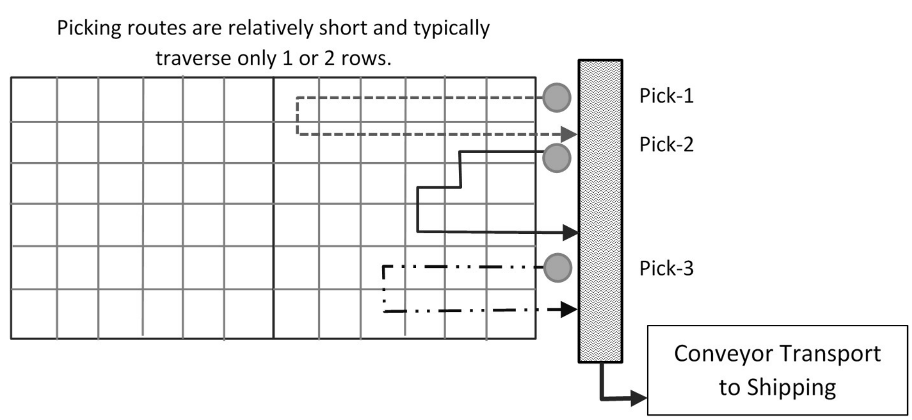


Figure 3.6 Short and unique picking routes.

The explosive strategy creates a stocking dispersion whereby multiple customer order SKUs are stored in close proximity. As E_i is increased L_i also increases and a customer order can be picked from any of the L_i locations. When many orders are active, the probability is high that several can be picked from a single aisle. IFWs are organized in a pick-to-belt configuration, and as shown in figure above, the picker travels along aisles and then enters an aisle. This allows for a very short pick route that traverses just one or two aisles and can fulfill several orders simultaneously, with possibly multiple orders from the same bin. While the pick zone assigned to a picker is large, for each pick instance there are several optimal or very good routes of 10 to 15 picks, and as shown in figure 4 each targets a different part of the assigned zone. This structural change in the picking behavior allows an IFW to achieve its same day shipment objective.

3.2.6 High Transactions and Total Digital Control

We observed a high level of digital activity control in the IFW. The explosive storage and single unit picks result in a higher rate of store/pick movements per unit shipment, and the number of data transactions is also much larger. Little decision making is done at the human level and all movements are modelled and instructed by a central controller. Both stockers and pickers have only short term visibility, possibly less than 15 minutes, of their next actions. As an example, in a picklist of 12 items, the picker only knows the next 4-5 items. Possibly the controller maybe updating the pick list in real time. During the visits little information about the control logic was known to floor level workers. There was also tight control on worker discretions, for example, workers must pick orders in the instructed sequence. IFWs integrate high levels of physical and data automation with high levels of labor.

3.3 Modelling Fulfillment Operations

Based on the observational study we propose a new model to describe the product flows and associated operations in an IFW. Specifically, this model highlights the key differentiators noted in the previous section. Similar to traditional warehousing there are three operational segments (i) receiving – explosion and stocking, (ii) fulfillment – order picking and (iii) shipping – packaging and dispatch. Here we focus only on the first two and consider two classes of workers (i) stockers and (ii) pickers. The model notation is introduced next, followed by a brief description of the operations:

$i \in N$ Items or SKU's stocked in the warehouse

$b \in M$ Sequentially numbered storage locations/bins in the warehouse

z Storage zones defined by a range of contiguous bins $\{b \in z \mid B_{min,z} \leq b \leq B_{max,z}\}$

$s \in S_z$ Stockers assigned to zone z

$p \in P_z$ Pickers assigned to zone z

E_i Explosion factor assigned to item i

$t \in T$ Operational periods (days) in the control model

$r \in R_t$ Incoming bulk received during period t

$j \in J_t$ Customer orders received during period t

$I_{i,b,t}$ Inventory of item i in bin b at the end of period t

Storage Parameters

V_i Volume of item i

G_i Minimum storage lot for item i

β Volume capacity of a storage bin, assumes all bins are identical in size

Receiving: Every period incoming bulk is received for R_t items. Depending on the replenishment policy R_t could be less than 1% or as much as 10% of N. The bulk quantities are unloaded and opened so that pickers can readily pick one or more units for stocking.

The IFW controller explodes each received item into E_i lots such that each is larger than G_i. Each lot is assigned to a specific bin location such that $\Sigma_i I_{i,b,t}V_i \leq \beta$. From the set of pending stock lots, the earliest free stocker is assigned an item stocking list which identifies for each item the SKU, lot quantity and storage bin. The only constraints are that the total list volume is less than χ the stocking cart capacity, and all bins are in the stocker assigned zone. The stocker completes the stocking order and reenters the free stocker queue. This process continues until all received items are stored, or the period is over. Items not stocked during the period will be pending next period. For each incoming bulk r let $U_{r,t}$ be the associated SKU, $A_{r,t}$ the arrival time and $Q_{r,t}$ the quantity received.

Fulfillment: Every period the warehouse receives J_t customer orders many of which could be for the same item. These orders are fulfilled from the stocked inventory at the end of the last period. This not a binding assumption and an advanced model could make the inventory immediately available for fulfillment. When a picker is free, the IFW controller generates an order pick list which identifies the SKU, pick quantity and bin location. The list of candidate picks is very large, and the solution space is described by the product of the number of pending orders and L_i in the picker's zone. The order picking algorithm is driven by the performance objective described below. Note that orders are arriving continuously and are time stamped. The picker completes the picking list and reenters the free picker queue. This process continues until all received items are stored, or the period is over. Items not stocked during the period will be pending next period. For each incoming order j let $W_{j,t}$ be the associated SKU, $C_{j,t}$ the order receipt time and $H_{j,t}$ the order quantity.

3.3.1 Performance Objective – Minimize Fulfillment Time

As noted earlier a key IFW differentiator is immediate fulfillment, with a goal of same day shipment. Figure illustrates the inventory time line for a shipped item. The two performance metrics are order fulfillment time – the interval between order receipt and shipment, and stocking time – the interval between bulk receiving and bin stocking. Clearly long stocking time will effect and limit the number of available picking locations, and in the extreme case could lead to stock outs even when the item is already in the warehouse. The primary market driven objective, though, is simply the fulfillment time measured on either a linear or quadratic scale. Let $\hat{C}_{j,t}$ be the order fulfillment time then:

$$Simple\ Fullfillment\ Time = \sum_t \sum_{j \in J_t} \frac{(\hat{C}_{j,t} - C_{j,t})}{TJ_t} \tag{3.3}$$

$$Quadratic\ Fullfillment\ Time = \sum_t \sum_{j \in J_t} \frac{(\hat{C}_{j,t} - C_{j,t})^2}{TJ_t} \tag{3.4}$$

Note that $\hat{C}_{j,t}$ is on an hourly clock extended across periods. While the observational study provided no further insights on whether a simple, quadratic or some combination metric was used, for the remainder of this dissertation, we will consider only the simple metric. On an operational basis the IFW Controller decision variables are linked to the picking and socking lists that are generated throughout the period. While these decision problems share similarities with the traditional order picking problems, the explosive strategy changes them sufficiently that a new class of problems are created.

3.3.2 Uniform Random Stocking List Algorithm

In random storage policy, the most common stocking objective is location selection based on class of use, space availability and proximity to associated SKUs. In an IFW the focus is on stocking time and location and subsequent influence on fulfillment time. The IFW stocking list problem is therefore different from traditional problems since multiple locations are selected for the E lots from the same bulk. Individual lots are stocked at different times depending on the zones they are sent to. Here we present an initial solution to the problem.

1. Initiate the algorithm and set $t=1$.

2. Explode all arriving bulk into stocking lots. Maximum number of exploded lots is constrained by the storage minimum. This is multiplied by the explosion ratio to derive the target number of lots:

$$For\ r \in R_t \quad set\ \ X_{r,t} = 1 + INT\left[\ o\frac{Q_{r,t}}{G_i}\right]\ where\ i = U_{r,t} \tag{3.5}$$

   $Q_{r,t}$ is then equally distributed across $y = 1$ to $X_{r,t}$ lots with $q_{y,r,t}$ units. To maintain integer lots, odd numbered lots are rounded up, while even lots are rounded down.

$$q_{y,r,t} = RoundUP\left\{\frac{Q_{r,t}}{X_{r,t}}\right\}\Big|\, y\ is\ odd\ \ q_{y,r,t} = RoundDN\left\{\frac{Q_{r,t}}{X_{r,t}}\right\}\Big|\, y\ is\ even\ \forall\ r \text{ and } t \tag{3.6}$$

3. Assign each exploded lot to a storage bin location $\breve{D}_{y,r,t}$ using a uniform random rule. If the bin capacity is exceeded then another bin assignment is generated.

$$\breve{D}_{y,r,t} = RA \quad \{1, M\}\ \ such\ that\ \ V_i\left(q_{y,r,t} + \textstyle\sum_i I_{i,b,t}\,\middle|\ i = U_{r,t}\ \ b = \breve{D}_{y,r,t}\right) \tag{3.7}$$

4. Update the bin inventory $I_{i,b,t} = I_{i,b,t} + q_{y,r,t}$ where $i=U_{r,t}$ and $b=\check{D}_{y,r,t}$. At the end of this step all the exploded lots arriving in t have been assigned to bins and are in the stocking queue described by the vector $\check{R}_{\xi,t} = \{ U_{r,t} , A_{r,t} , q_{y,r,t} , \check{D}_{y,r,t} \}$ where ξ is a counter assigned to all exploded stocking lots in period t, plus those carried over from the previous period.

5. We assume the stocking shift starts at time zero and ends at T_s. Let ϕ_s be the free or available time of a stocker, and at the start of period set $\phi_s = 0$ for all s. The following notation is used to describe a stocking list:

e	Stocking list number, set e=0 at the start of a period
k, K	Sequential stocking stops in a list and the maximum stops
$\check{S}_{e,t}$	Stocker assigned to list e
$F_{k,e,t}$	Stocking lot processed in stop k of list e
$\delta_{1,e,t}$ $\delta_{2,e,t}$	Stocking list start and end times
τ_e τ_w τ_b	Stocking list setup time, stocking time per unit, and travel time per unit bin
λ	Current location of stocker in the list

6. Identify the next available stocker s^*, $\phi_{s^*} = \text{Min}\{ \phi_s \mid \text{all } s\}$. If $\phi_{s^*} > T_s$ then no more lots can be stocked in this period, go to step 10. Let z^* be such that $s^* \in z^*$. Initiate a new stocking list $e=e+1$ and assign $\check{S}_{e,t} = s^*$.

7. Select the first stop in list e to be the earliest pending lot ξ* in the queue such that:

$\{A_{r,t} \in \check{R}_{\xi^*,t}\} = \text{Min}\{ A_{r,t} \mid B_{min,z^*} \leq \check{D}_{y,r,t} \leq B_{max,z^*} \text{ for all pending } \xi\}$
Set $F_{1,e,t}=\xi^*$, $\delta_{1,e,t}=\{A_{r,t} \in \check{R}_{\xi^*,t}\}$, $\delta_{2,e,t} = (\delta_{1,e,t}+\tau_e+\tau_w\{q_{y,r,t} \in \check{R}_{\xi^*,t}\})$, $\hat{U}=\{q_{y,r,t} \in \check{R}_{\xi^*,t}\}(V_i|i=\{U_{r,t} \in \check{R}_{\xi^*,t}\})$ and $\lambda=\{\check{D}_{y,r,t} \in \check{R}_{\xi^*,t}\}$. Remove ξ* from the queue list $\check{R}_{\xi,t}$.

8. Increment the stop to k=k+1. Add lots to list using the closest bin rule, select ξ* from the queue such that:

$\{ |\lambda-\check{D}_{y,r,t}| \in \check{R}_{\xi^*,t}\} = \text{Min}\{ |\lambda-\check{D}_{y,r,t}| \mid B_{min,z^*} \leq \check{D}_{y,r,t} \leq B_{max,z^*} \text{ for all pending } \xi\}$
Set $F_{k,e,t} = \xi^*$, $\delta_{2,e,t} = (\delta_{2,e,t} + \tau_b|\lambda-\check{D}_{y,r,t}| + \tau_w \{q_{y,r,t} \in \check{R}_{\xi^*,t}\})$, $\hat{U} = \hat{U} + \{q_{y,r,t} \in \check{R}_{\xi^*,t}\}(V_i|i=\{U_{r,t} \in \check{R}_{\xi^*,t}\})$ and $\lambda=\{\check{D}_{y,r,t} \in \check{R}_{\xi^*,t}\}$. Remove ξ* from the queue list $\check{R}_{\xi,t}$.

9. If any of the list stopping conditions is met then no more lots can be added to the list: (i) Cart capacity $\hat{U} > \ddot{U}$ (ii) Maximum stops $k=K$ or (iii) Shift has ended $\delta_{2,e,t} > T_s$. Else return to step 7 to add more lots.

10. If there are pending orders in $\check{R}_{\xi,t}$ then go to step 5.

11. If $t<T$ then update $t=t+1$ and go to step 2. Else stop.

3.3.3 Narrow Band Order Picking Algorithm (NBOP)

As noted the primary objective of an IFW is immediate fulfillment, and critical to this is the prescriptive model by which the order picking list is created. We believe significantly different from the traditional problem, and defines a new class of IFW order picking problems. Two decisions are integral to creating the order picking list here, which orders to serve in the next pick list and which storage locations to fulfill the order from. The following notation is used to describe a picking list.

f	Picking list number, set f=0 at the start of a period
k, $\check{K}$	Sequential picking stops in a list and the maximum stops
$\acute{P}_{f,t}$	Picker assigned to list f
$G_{k,f,t}$	Customer order processed in stop k of list f
$\dot{D}_{k,f,t}$	Bin location from which order in stop k of list f is fulfilled
$\delta_{1,f,t}$ $\delta_{2,f,t}$	Picking list start and end times
τ_e τ_w τ_b	Picking list setup time, Picking time per unit, and travel time per unit bin
η	Current location of picker in the list

The order picking queue is described by the vector $\hat{O}_{j,t} = \{W_{j,t}, C_{j,t}, H_{j,t}\}$. When a picker becomes available the algorithm identifies the earliest $C_{j,t}$ in the queue that can be fulfilled from the associated zone. This order is the first pick, and subsequent picks are selected from orders that can be filled within a narrow band of the current location and the earliest $C_{j,t}$. The proposed algorithm is described as:

1. Initiate the algorithm by setting t=1.

2. We assume the picking shift starts at time zero and ends at T_P. Let ϕ_p be the free or available time of a picker, and at the start of period set ϕ_p =0 for all p.

3. Add all orders for t to the order queue $\hat{O}_{j,t}$. If there an unfilled order from the previous period they are also added to $\hat{O}_{j,t}$.

4. Identify the next available picker p^*, ϕ_{p^*} =Min$\{ \phi_p \mid \text{all } p\}$. If $\phi_{p^*} > T_p$ then no more orders can be processed in this period, go to step 9. Let z^* be such that $p^* \in z^*$. Initiate a new stocking list f=f+1 and assign $\acute{P}_{f,t} = p^*$.

5. Select the first stop in list f to be the earliest pending order j^* in the queue such that:

$C_{j^*,t}$ = Min$\{ C_{j,t} \mid I_{i,b^*,t} \geq H_{j,t}$ for some $B_{min,z^*} \leq b^* \leq B_{max,z^*}$ where $i = W_{j,t}$, for $j \in \hat{O}_{j,t}\}$
Set $G_{1,f,t}$=j^* and $\dot{D}_{k,f,t} = b^*$. Since the picking cannot start till the next customer order arrives set $\delta_{1,f,t}$=$Max(\rho+C_{j^*,t}, \phi_{p^*})$, where is a start delay to allow for more incoming orders to be included in the current pick. In a high volume IFW, though, most often we can expect $\delta_{1,f,t}$=ϕ_{p^*}. Set $\delta_{2,f,t} = (\delta_{1,f,t}+\tau_e+\tau_w H_{j^*,t})$, and η=b*. Remove j^* from the queue list $\hat{O}_{j,t}$. and update the inventory $I_{i,b^*,t}$=$I_{i,b^*,t} - H_{j^*,t}$ where $i = W_{j^*,t}$.

6. Increment the stop to f=f+1. Add orders to the pick list by searching for the earliest pending order j* in the queue that can be fulfilled within a band of λ bins from the current location.

$C_{j^*,t}$ = Min$\{ C_{j,t} \mid C_{j,t} \leq \delta_{1,f,t}$, $I_{i,b^*,t} \geq H_{j,t}$ for some $B_{min,z^*} \leq b^* \leq B_{max,z^*}$ and
$\eta-\lambda \leq b^* \leq \eta+\lambda$ where $i = W_{j,t}$, for $j \in \hat{O}_{j,t}\}$

If no j^* meets the condition then the picking list is ended and go to step 7. Else, Set

$\dot{D}_{k,f,t} = b^*$, $\delta_{2,f,t} = (\delta_{2,f,t} + \tau_b|\eta - b^*| + \tau_w H_{j^*,t})$, and η=b*. Remove j^* from the queue list $\hat{O}_{j,t}$. and update the inventory $I_{i,b^*,t} = I_{i,b^*,t} - H_{j^*,t}$ where $i = W_{j^*,t}$.

7. If any of the list stopping conditions is met then no more orders can be added to the list: (i) The queue list $\hat{O}_{j,t}$ is empty (ii) Maximum stops $k=\check{K}$ or (iii) Shift has ended $\delta_{2,f,t} > T_P$. Else return to step 6 to add more orders.

8. If there are pending orders in $\check{R}_{\xi,t}$ then go to step 4.

9. If $t<T$ then update $t=t+1$ and go to step 3. Else stop.

CHAPTER 4

IFW SIMULATION

4.1 Simulation Modelling

A simulation is an artificial reality that is developed to model a real system or process in order to obtain predictive information that is useful in decision making without the risk and expense of building the actual system. Simulations are suitable for problems in which there are no closed-form analytical solutions. By simulating a system for many replications and recording the observations, system statistics can be computed in order to make evaluations and design strategies. Simulation models are classified as: (i) static or dynamic models, (ii) stochastic or deterministic models, and (iii) discrete-event or continuous models.

4.1.1 Categorization of Simulation Models

Harrell and Charles (2004), defines the categories:

> **Static simulation** is not based on time and often involves drawing random samples to generate a statistical outcome, so it is sometimes called Monte Carlo simulation.
>
> **Dynamic simulation** includes the passage of time; a clock mechanism moves forward in time and state variables are updated as time advances.
>
> **Deterministic simulations** have constant inputs and produce constant outputs.
>
> **Stochastic simulations** have random inputs and produce random outputs.
>
> **Discrete-event simulation** is one in which state changes occur at discrete points in time as triggered by events such as the arrival of an entity to a workstation, the failure of a resource, the completion of an activity, and the end of a shift.

Continuous simulation is where state variables change continuously with respect to time and are therefore referred to as continuous-change state variables.

Hybrid simulation is a combination of both discrete-event and continuous simulation capabilities. It is more beneficial since most processing systems that have continuous-change state variables also have discrete-change state variables.

The main difference between continuous and discrete models is that discrete models monitor what happens to an individual item in the system whereas continuous models monitor the entire system. When a warehouse is being simulated, a continuous model is more appropriate since the system is continuously collecting observations of a continuous flow of information and items as time gradually increments by a specified amount of time.

4.1.2 Steps of Creating a Simulation

Steps of creating a simulation are:

1. **Problem Definition:** Identification and definition of the system or procedure that needs to be monitored clearly in order to mirror the system as closely as possible.

2. **Project Planning (Creating the Conceptual Model):** Define objectives, formulate the problem, identify parameters, list performance measures, decide the time frame of the study and break down the task into workable pieces.

3. **Simulation Tool Selection:** Based on the system requirements, determine which simulation tool is more appropriate. Building and running a simulation is a highly time consuming process. Therefore, choosing the right simulation tool is crucial.

4. **Input & Output Data Storage Selection:** Based on the problem size, decide a data storage method.

5. **Input Data Collection & Analysis:** Determine type of input data to collect. If an existing system is going to be modeled, collect existing data. If the system does not yet exist, then create data using appropriate probability distributions.

6. **Output Data Storage:** Create tables, identify database primary and foreign keys, define primary key attributes, create entity relationship diagram by validating keys and relationships.

7. **Formulate and Develop a Model (Creating the Computer Model):** Translate the model into a programming language by drawing schematics and network diagrams of the system. Programming languages like Python, R, SQL, or VBA are more flexible than general purpose simulators like ARENA, Simio, and Simul8. Although, construction of simulation modelling using programming languages may be more challenging.

8. **Verification & Validation of the Model:** Ensure that the model performs as intended, if possible, compare the model's performance under known conditions with the performance of the real system.

9. **Experimental Design:** Decide on problem size, system parameters, and levels of each input variable, determine the desirable number of simulation runs and number of replications needed to gather a sufficient amount of output data to do a statistical analysis.

10. **Perform Simulation Runs:** Plan an organized way of running simulations and documenting simulation results.

11. **Analyze Results:** Interpret results and report an analysis of performance measures. A statistical analysis may be necessary to validate the model and/or test the hypothesis. Once the simulation process is completed, there may be a need to improve and redesign the existing model. In this case, necessary changes must be made and steps must be repeated.

4.1.3 Database Needs of a Simulation

Simulation modelling is used to mimic the behavior of the real system and the main purpose of simulation is to gather and collect observations of the monitored system as a function of time (Taha and Elizandro, 2008). This makes data handling: database selection, storing and accessing the data generated in each run of a simulation experiment a key concept in simulation modelling. There are various simulation software products as well as programming languages that can be linked to databases in order to effectively and efficiently store and retrieve large amounts of simulation data. Particularly continuous simulation modelling of a complex system requires a large amount of input data and generates a large amount of output data. For that reason data logging options of generic simulation packages will not be able to store the output data of these complex models in an organized and structured way. Therefore when collected data is large a separate database should be considered.

IFW simulation is a quite multifaceted and requires a well-rounded database. Input data for the IFW model is generated in MS Excel, then transferred to MS Access in order to generate a collection of database tables. The queries to run the simulation and code to record output data are written in Visual Basic, which is implemented in MS Access.

4.2 Simulating IFWs

4.2.1 Simulation Objective

The first decision in building a simulation is defining the objective. Our objective is building a simulation model that compares an IFW with explosive storage policy to a warehouse that uses known bulk storage policies, and test the hypothesis that is:

> *A fully randomized explosive storage policy warehouse has more efficient picking operations and shorter fulfillment times compared to a traditional fixed location policy warehouse.*

4.2.2 Warehouse Operations

A simulation model should contain all operations in a warehouse that are associated with the main objective in order to accurately mimic the real system. Although, expanding the simulation with non-value adding modules will exhaust the computer memory when running the simulation. Therefore, the decision is what will be included in the simulation that effects the objective directly. The four main warehousing operations and some decisions that are included in the simulation are:

1. **Receiving (Procurement):** What is the initial inventory at the beginning of a simulation experiment? Which SKUs should be replenished based on a realistic demand/sales relationship and in what quantity should SKUs be ordered?

2. **Sales (Order Generating):** What is the demand behavior and what are demand classes? What should be the total number of sales in a specified number of time and what distribution should be used?

3. **Storing:** It is known that IFWs use a randomized storage policy. What level of randomness should be modeled? Is there any kind of computer control in warehouse space utilization? How can zoning be adapted? What is the size and design of the warehouse? How many zones are appropriate? How many storage associates should be assigned?

4. **Picking & Packing & Shipping:** What picking policy should be adapted? How many pickers should be assigned? Should packing and shipping operations be simulated along with picking? What is the size of simulated problem? Since picking time effects fulfillment time of an order primarily, what kind of algorithm should be used?

4.2.3 Warehouse Design

The main interest of this simulation model is finding out fulfillment times of customer orders and optimizing the receiving/sales/picking processes in an IFW. What makes an IFW different than a traditional warehouse is basically the size and storage policies, and therefore, the strongest impact is on storage and picking times. The observed IFWs were 1.2 million square feet, and the entire warehouse was divided into zones. Considering the benchmark, we decided on having 9 zones:

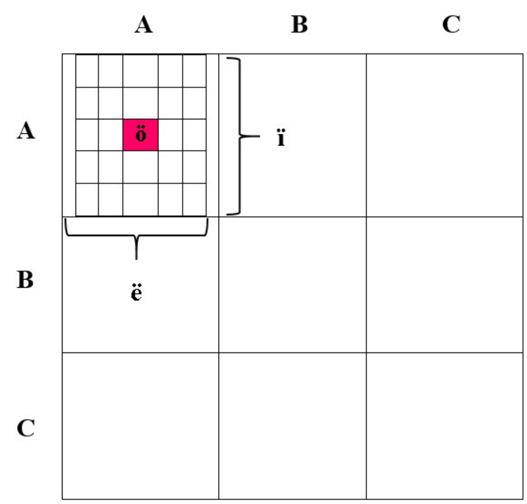


Figure 4.1 Warehouse design.

The base model warehouse has 9 zones identified by AA, AB, CA etc. and 6 aisles divided into 6 bookshelf like pods with 10 shelves each in each pod. Total number of storage locations sum up to:

$Z_{x,y}$= 1,…, z z ∈ Z [1,9}]

S_x = 1,…, ï ï ∈ Ï [1,6] (horizontal identifier of aisles in a zone.)
S_y = 1,…, ë ë ∈ Ë [1,6] (vertical identifier of aisles in a zone.)
S_z = 1,…, ö ö ∈ Ö [1,10] (number of shelves)

M = ï x ë x ö x z = 6 x 6 x 10 x 9 = 3240 storage locations.

The simulation process is consists of two main activities: Receiving and Sales. For ease of calculation, aisles in each zone for both operations are set unidirectionally and we assume storage locations are linearly positioned. A walking path to visit 360 bins in a zone looks like the Figure 4.2:

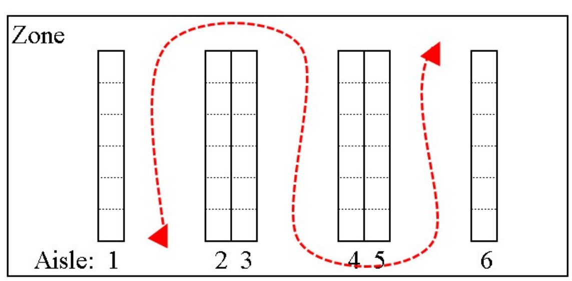


Figure 4.2 Unidirectional zone design.

4.2.4 Input Data Generation

IFWs are busy warehouses with very high transaction rates in all operations mainly because of the size of the warehouse and the explosive storage policy they are using. For this case it was particularly difficult, because there was no initial data from the observed IFWs. Therefore all data used in the model was carefully generated using probability distributions in order to accurately represent the nature of IFWs where customer orders are taken online and minimizing fulfillment time is the main objective. A data file is created in MS Excel in order to create different scenarios with 7 tabs: Interface, Item, Startup inventory, Receiving, Sales, People, and Location.

The interface tab is used to enter parameter values to generate experiments. Parameters include number of SKUs, size of the warehouse, employee/zone and employee/task assignment, number of sales orders received, number of days that simulation runs, search band size pick list size, bin volume and so on.

The item tab acts like an appendix where all essential information for each SKU is stored. The information includes but is not limited to the number of minimum packing requirements (minpack), dimensions (volume) and weight. This uniform random variables are created with uniform distribution using excel functions and they are static for all of the experiments. Item table also has more information in it in order to generate receiving and sales tabs such as monthly sales, number of inbound orders, number of customer orders. These data are gathered exogenously.

Startup inventory is on hand inventory which is about %10 of the monthly sales along with the assigned locations of the items that are being used in the experiment. Location assignment of these items differ for varying explosion ratios. The number of storage locations for each item that initial inventory is stored increases as explosion ratio increases.

Receiving tab has receiving ID, SKU, quantity, and arrival time of inbound items that are being used in an experiment. The quantity is decided to be within 20% of the mean quantity per received order which can be adjusted for different design of experiments.

Sales tab has the sales ID, SKU, quantity and order arrival time. The number of sales orders are calculated based on sales per day and the quantity is considered to be around the mean quantity of sales orders of each item.

People tab has employee ID, shift start-end times for each worker, and labor allocation (role of the worker, picker or storer) in each zone.

Location tab has exact address of a storage location in terms of aisle and zone. It is used to resize the warehouse when creating new experiments. For example, when the total number of storage locations is increased from 900 to 1800, location addresses of the 900th item moves from the 9th zone to 5th zone.

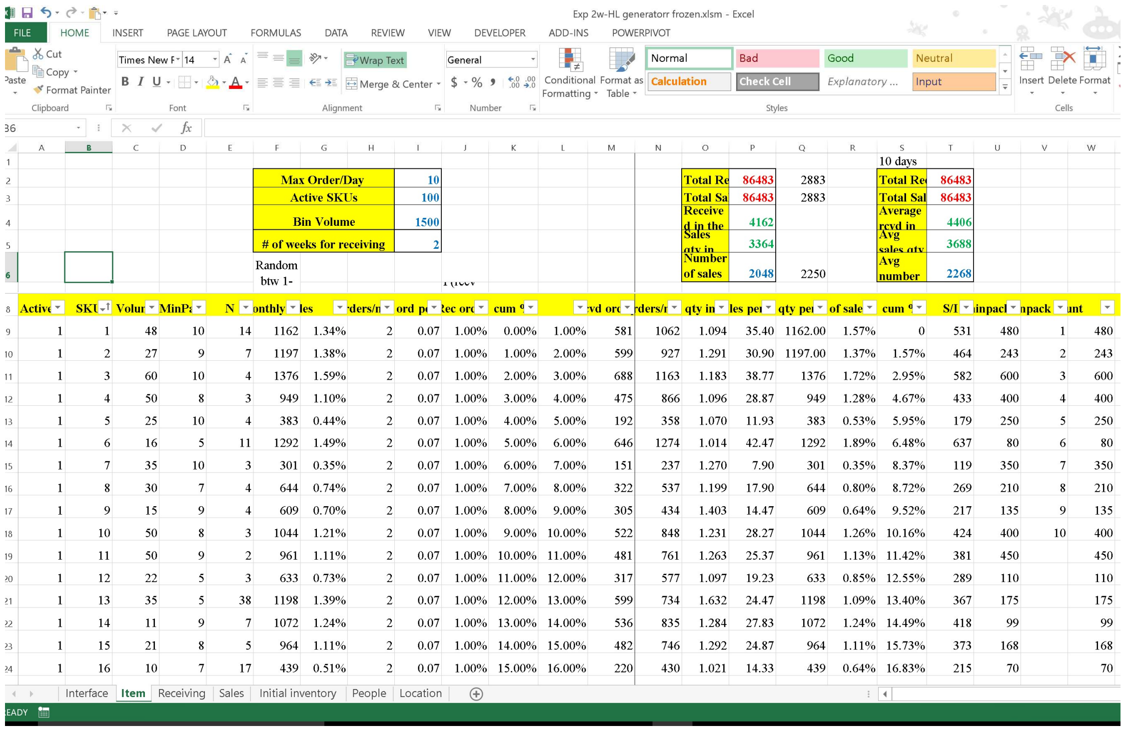


Figure 4.3 Input data generation file.

4.2.5 Constructing the Database using MS Access

The first step creating the simulator is creating the database. A database is basically bunch of tables filled with data that are tied with primary keys. Primary key of a table uniquely identifies each record in the table. Data cannot be stored into a database without first creating a table, and tables do not create a database without establishing relationships using a primary key.

Create Tables: A simulation database requires tables for imported input data, to record output data and to temporarily save data as simulation runs. Each table consists of three columns: field name, data type (number, currency, etc.), and description (optional, to be used to describe the information that is to be entered in this field).

Create Primary Key(s): A primary key is a special relational database table column(s) designated to uniquely identify all table records. A primary key must contain a unique value for each row of data (cannot contain null values).

List of tables can be viewed on the left side using the Navigation Pane:

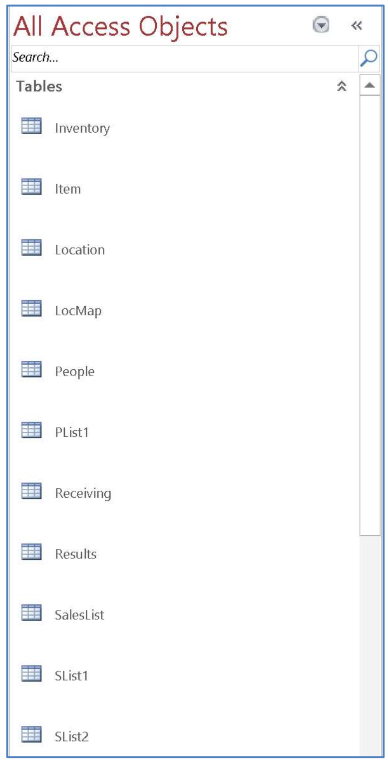


Figure 4.4 Objects view of tables in MS Access.

Visual representation of a relational database structure that illustrates all the entities, attributes and relationships can be viewed in Microsoft Access. It is important to verify that the tables are connected right before entering any information into the database. An entity in a database may represent a person, place, object, event or idea that stores and processes data. An entity in a relational database has its own table. Within the entity's table are the attributes, also referred to as a field or column that characterizes the entity. The Entity Relationship diagram in MS Access represent the entities as rectangles with the list of columns that it contains, starting from the primary key(s).

An example of entity relationship table of IFWs look like:

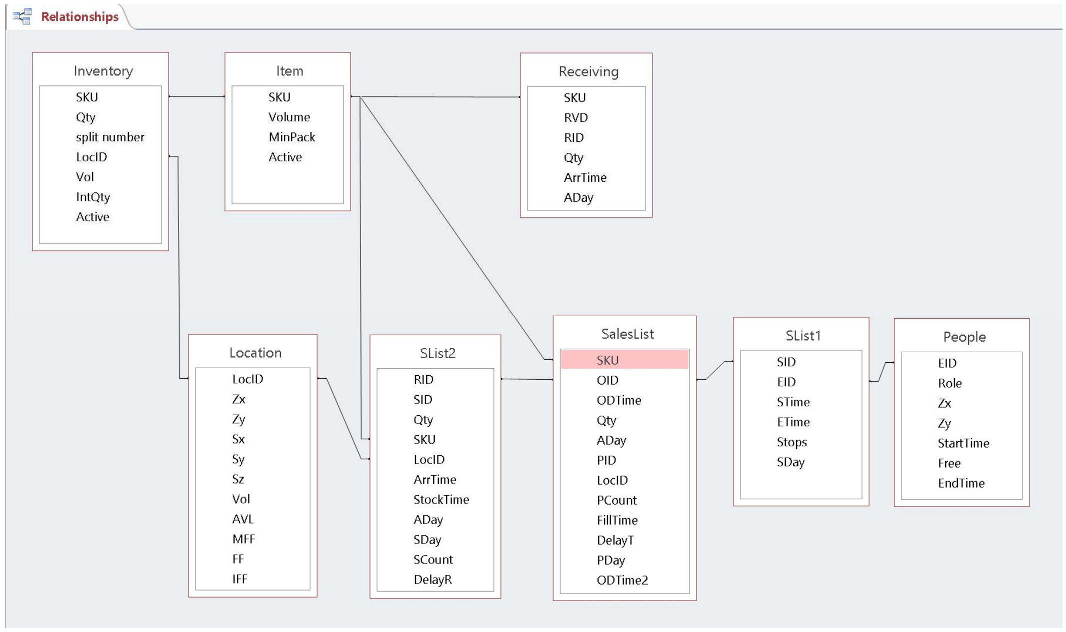


Figure 4.5 Entity relationship diagram of IFW model.

4.2.6 Simulation Process

The simulation process starts with importing the excel data file into access database. Visual Basic is used to code the process which is embedded into MS Access. Startup inventory is imported first, and since the locations in the initial inventory table are already assigned, the task here is to update the "Inventory" table with the given data. Code is written in a module in "Design View" and looks like Figure 4.6:

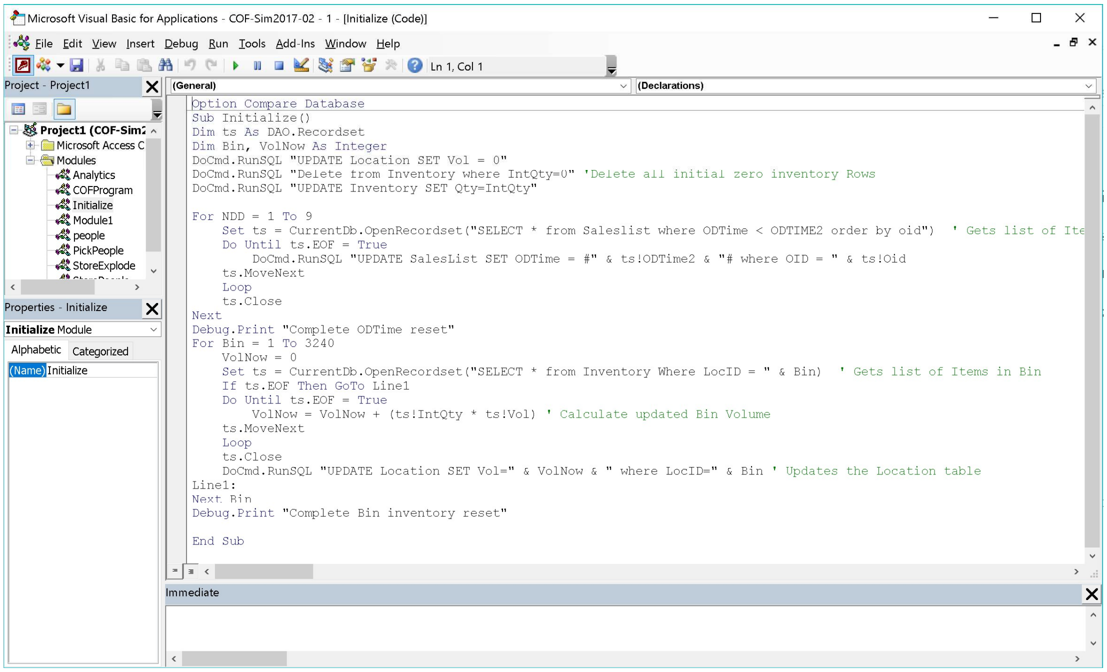


Figure 4.6 Design view of simulation modules.

The simulation model mentioned in Chapter 4 is coded using Visual Basic and then the simulation is run several times for every experiment. The main elements of the control logic are Receiving, Storage and Fulfillment. The two algorithms represented in Chapter 3: the Uniform Random Stocking List Algorithm which helps store received bulk items, and the Narrow Band Picking Algorithm which helps create pick lists are written in Visual Basic. In Analytics module, results are calculated.

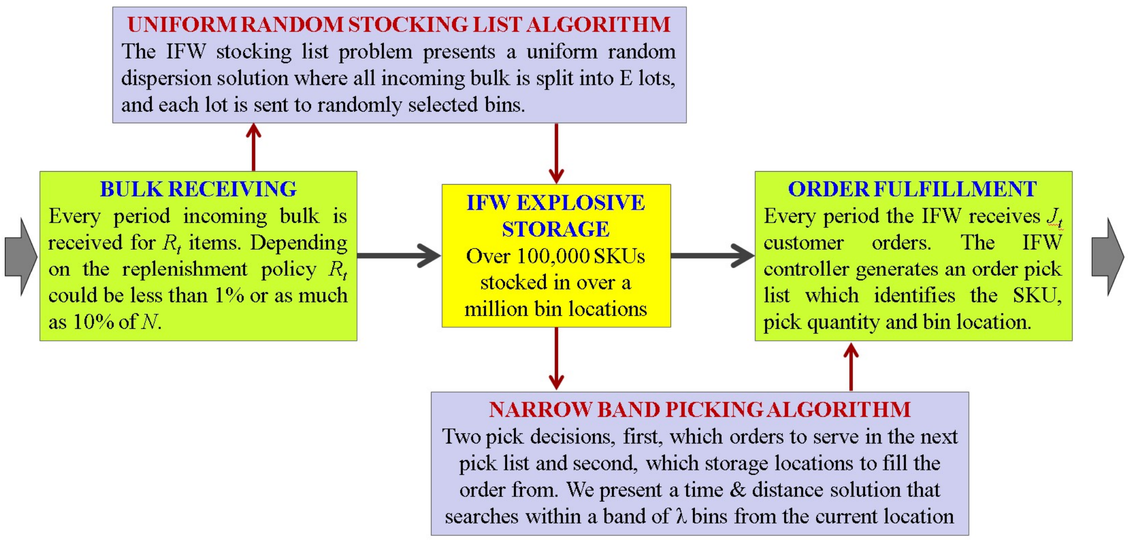


Figure 4.7 IFW control logic model.

4.3 Performance Analysis of Explosive Storage

An explosive storage policy requires considerably more resources both from a facility design and information technology context. A key analytical question then is what the likely fulfillment time performance advantages are. We used a simulation model to analyze the performance behavior of linear fulfillment time. A data driven simulation model was built in MS-Access/VBA platform. Considering the data requirements needed to track the exploded inventory, traditional discrete event simulation models could not be used. Given the processing time limits, the largest feasible model was run, and the parameters are shown in Table 4.1.

Table 4.1 Key Parameters for the Experimental IFW Problem

$N = 400$ SKUs	$M = 3240$ Bins	$\beta = 3000$ in^3
$Z = 9$ (Equal Size)	$S_z = 6$/zone	$P_z = 6$/zone
$T = 9$ days	$T_s = 8$ Hours	$T_p = 8$ Hours
$\sum_t R_t = 220$	$\sum_t J_t = 22000$	$K = 10$

Both incoming bulk and customer orders arrive uniformly during the day. The inventory replenishment strategy was set at 3 weeks, which gives about 220 unique SKU arrival during the 9 day interval. Customer order distribution was 80% for 1 unit, 15% for 2 units and 5% for 3 units and the SKU distribution was also uniform. All items movements in the IFW simulator were prescribed by the stocking and picking algorithms described in the previous section. The two factors in the experimental design were: (i) the explosion ratio Ψ_o which is set at the same Ψ_i value for all SKU – in 0.1 increments from 0.1 to 0.9 and (ii) $\check{K}$ the maximum number of picker stops – set at 10, 13, 15, 17 and 20. For consistency the same R_t and J_t data file was used in all experiments. The only random function in the simulation is Step 3 of the stocking algorithm, which generates different dispersion patterns. The number of repetitions was sets at 25, and the mean results were recorded. Additionally, 10 different R_t and J_t data files were created, but the data condition $\sum_t R_t$ = 220 and $\sum_t J_t$ = 22000 was maintained across all data files. The results in the following section are the mean across all data files. Initial inventory files for all cases were generated to match R_t and J_t such that there no inventory stock outs.

4.3.1 Effect of Increasing Explosion Ratios

Figure 4.8 shows the simulation results for five different settings of $\check{K}$ with increasing explosion ratios. The longest fulfillment time of 76 minutes is used to benchmark the results. For all five $\check{K}$ settings the fulfillment time drops consistently till Ψo =0.8. A Ψ_o =0.1 has very little explosion and is representative of a traditional warehouse with random or dedicated storage. For the experimental problem the fulfillment time decrease between Ψ_o = 0.1 and 0.8 ranged between 7% and 16% from the five solutions. The results confirm that across the range of experimental problems an explosive strategy will significantly improve fulfillment time. We see that largest time reductions occur in the Ψ_o = 0.1 to 0.3 range, and at the minimum IFWs should operate at the upper bounds of this range. The results also show that at Ψo=0.9 the fulfillment starts to trend up. Analysis of the results shows that at 0.8 the picking opportunities are maximized and further explosion does not generates only redundant opportunities. On the down side higher results in smaller qy,r,t which in turn results in many single unit inventory locations. This then limits the picking options for 2 and 3 unit orders, and the fulfillment time delays.

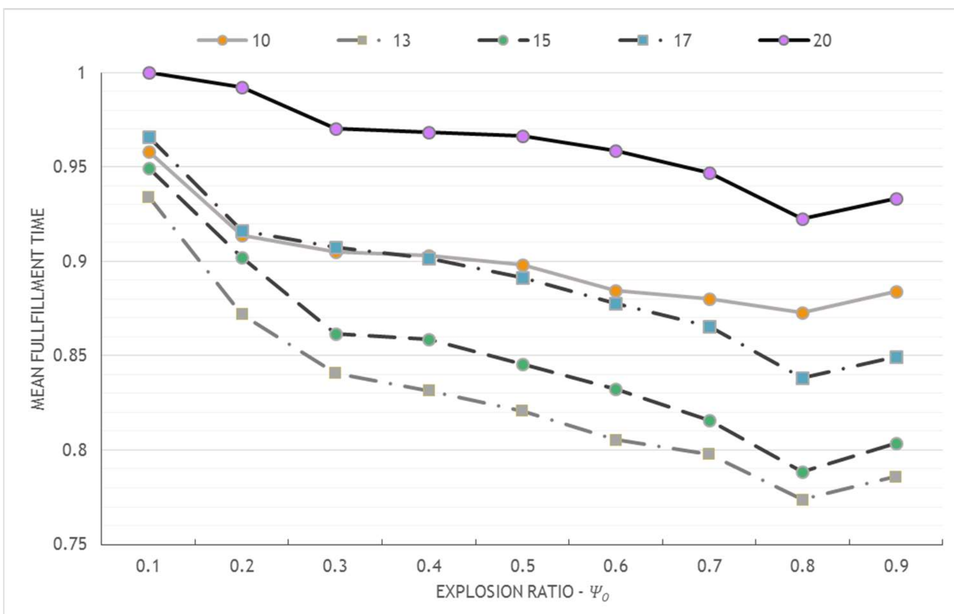


Figure 4.8 Fulfillment time as function of increasing explosion ratios.

Table 4.2 Fulfillment Time as Function of Increasing Explosion Ratios

	Maximum number of stops				
	10	**13**	**15**	**17**	**20**
0.1	0.961	0.937	0.952	0.969	1.000
0.2	0.916	0.874	0.905	0.919	0.987
0.3	0.894	0.843	0.864	0.897	0.973
0.4	0.906	0.834	0.861	0.907	0.973
0.5	0.901	0.823	0.848	0.898	0.969
0.6	0.886	0.808	0.835	0.880	0.961
0.7	0.887	0.800	0.818	0.868	0.949
0.8	0.875	0.776	0.791	0.841	0.925
0.9	0.887	0.788	0.806	0.852	0.936

4.3.2 An Optimal Picking Parameter – Ǩ

In Figure 4.8 we saw that the fulfillment time trend was not monotonic as the maximum number of picking stops $\check{K}$ was increased. Table 4.2 presents the rearranged data, with fulfillment time as a function of $\check{K}$. Clearly, $\check{K}$ does effect the fulfillment time significantly and the behavior is convex, implying that for a given IFW problem there is likely an optimal $\check{K}$ setting. The convexity becomes sharper as we trend towards the best performing Ψ_o setting. For the problem studied here the optimal value occurs at $\check{K}$=13 consistently across all explosion ratios. For the case of Ψ_o =0.8 the fulfillment time rises sharply by 10% when $\check{K}$ is changed from to 13 to 10, and by 15% when changed from 13 to 20. Note that this behavior is specific to the narrow band picking algorithm used here, newer algorithms may shows different sensitivity to $\check{K}$. Analysis of the results shows that when is limited to 10 then many opportunities for a quick pick are neglected. In contrast as is increased to 20 the longer pick cycle delays the fulfillment of the first picks in the list. As an example if the pick cycle time increases by 10 minutes due to the 7 additional picks, then the first 13 picks are all delayed by an additional 10 minutes.

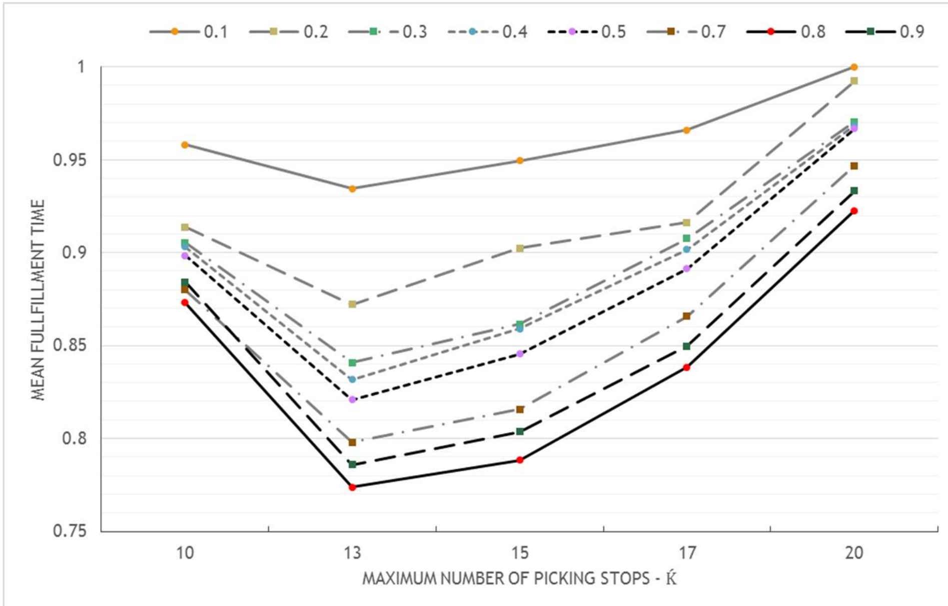


Figure 4.9 Fulfillment time as function of the maximum number of picking stops.

Table 4.3 Average Fulfillment Time as a Function of the Max Number of Pick Stops

	Maximum Number of Stops in a pick list				
	10	**13**	**15**	**17**	**20**
0.1	71.22	69.46	70.58	71.81	74.14
0.2	67.93	64.83	67.08	68.12	74.16
0.3	66.30	62.51	64.04	66.48	72.14
0.4	67.14	61.83	63.86	67.22	72.17
0.5	66.79	61.02	62.85	66.56	71.87
0.6	65.68	59.89	61.88	65.24	71.26
0.7	65.73	59.32	60.64	64.36	70.39
0.8	64.90	57.52	58.61	62.33	68.59
0.9	65.74	58.43	59.75	63.16	69.37

4.3.3 Average Number of Stops in a Pick List

The narrow band algorithm is designed to find picking options within a narrow band from the current picker location. Figure 5 show the results for three $\check{K}$ settings. For all three settings, the average number of picks increases with increasing explosions rations, confirming that that the explosion policy provide many more picking options with a short route. Between Ψ_o =0.1 to 0.4 the number of picks increased by 74%, but between Ψ_o =0.4 to 0.9 the number of picks only increased by 8%. Clearly the explosion to picking advantages are greatest in the early increases. Further analysis of the results, though, reveals the pick quality improves with the larger explosions ratios. In the lower values, many of the picks are delayed orders, and at times for the same SKU. The overall performance impact is faster fulfillment times.

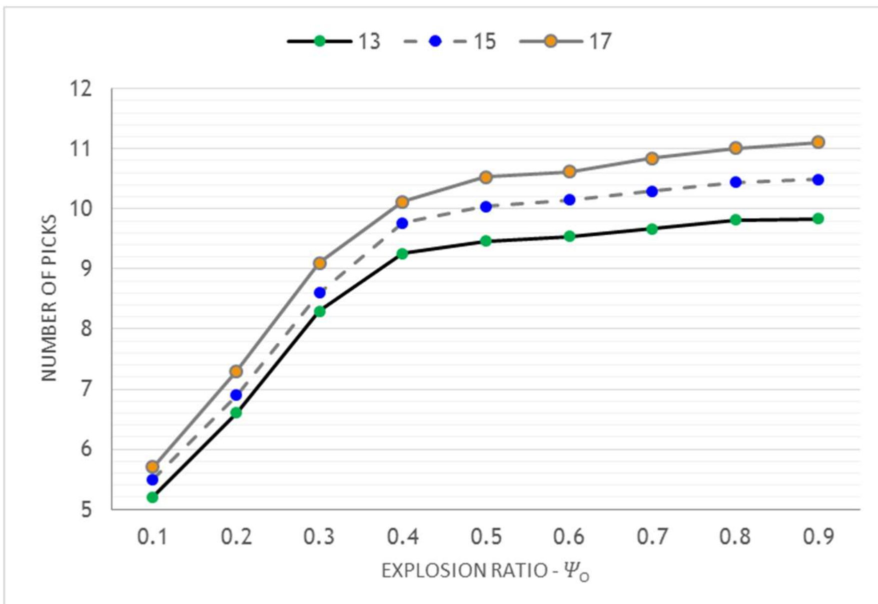


Figure 4.10. Average number of stops in a pick list.

Table 4.4 Average Number of Picks

	Max number of stops		
	13	**15**	**17**
0.1	7.00	7.06	7.13
0.2	8.30	8.66	8.86
0.3	8.89	9.28	9.55
0.4	9.26	9.77	10.12
0.5	9.47	10.04	10.53
0.6	9.54	10.14	10.62
0.7	9.67	10.30	10.84
0.8	9.82	10.44	11.01
0.9	9.84	10.49	11.10

4.4 Results and Conclusions

IFWs are a new and innovative concept in warehousing allowing for efficient and direct fulfillment of online customers orders. During the last decade both pure online and Omni channel retailers have been building IFWs, and we find the IFW operating policies and physical design to be significantly different from traditional warehouses. This model describes the associated receiving and fulfillment product flows. Explosive storage of incoming bulk allows for much quicker fulfillment of incoming customer orders. Two decision algorithms for (i) generating a stocking list and (ii) creating an order picking list are formulated and presented.

A simulation model to evaluate the fulfillment time performance advantages of the explosive policy was built. Experimental runs were conducted on a problem with N=400, M=3240, bulk receipts $\sum_t R_t = 220$ and customer orders $\sum_t J_t = 22000$. The base case of Ψ_o =0.1 was considered equivalent to traditional storage policy. The results show that increasing levels of explosions reduce the linear fulfillment time by as much as 16%, confirming that the IFW storage policy is advantageous. The results also show that fulfillment time behavior is convex as a function of the picking algorithm parameter $\check{K}$. The average number of actual stops is also observed to rise with higher explosion ratios. The performance results are based on the two algorithms presented here. Both algorithms can be further extended to further improve the decision model leading further performance improvements. Additionally, the model can be expended to consider the quadratic fulfillment time.

CHAPTER 5

ADVANCED ORDER PICKING ALGORITHMS FOR IFWs

In Chapter 4, the fulfillment time performance of explosive storage policy was evaluated using a simulation model. In the experimental problems fulfillment time decreased by as much as 16% with higher explosion ratios. However, the NBOP algorithm does not fully exploit all picking opportunities provided by explosive storage policy. The NBOP is a greedy algorithm that finds the most attractive pick and then builds a pick list from it. In this chapter, a more advanced algorithm is developed that evaluates the likelihood of finding multiple picks from an attractive bin which is surrounded by other attractive bins.

To serve the immediate fulfillment objective of IFWs, a hybrid approach of batching, and pick sequencing policy is formed. In the literature the objective is to reduce warehouse workload, therefore the travelling salesman problem is used. In explosive storage identical items are stored in multiple locations throughout the warehouse. The solution strategy must find an inventory cluster from a subset of the earliest ordered items in order to minimize travel and fulfillment time.

Traditional warehouses plan their outbound shipments days or weeks in advance whereas in IFWs the fulfillment process has more fluidity. Ideally each order remains in the system for a few hours and only a small percentage of orders would be delayed to the next day. The factors that may affect average fulfillment time include size and design of the IFW, number of workers, and working hours of the facility etc.

5.1 MIP Formulation of the Picking Problem

In mathematical programming, the objective is always to maximize or minimize a function of some variables that are controlled by the decision maker. They are the unknowns of a mathematical programming model. The variables are often called decision variables because the problem is to decide what value each variable should take. When one or more decision variable is an integer, the problem is called Mixed Integer Programming (MIP) problem.

We consider a case where thousands of customer orders are pending, each of which are associated with a single SKU. The order quantity and the inventory dispersion through the storage zone is known. For a given pick list size Π, the problem is to find a cluster of bins such that there is sufficient inventory within the cluster to fulfill at least Π orders. Let L_{min} and L_{max} be the first and last bins in the cluster, the objective then is to minimize L_{max} - L_{min} which is the pick travel distance. In the ideal case there would be a bin with sufficient inventory to fulfill Π orders, and $L_{max} - L_{min} = 0$.

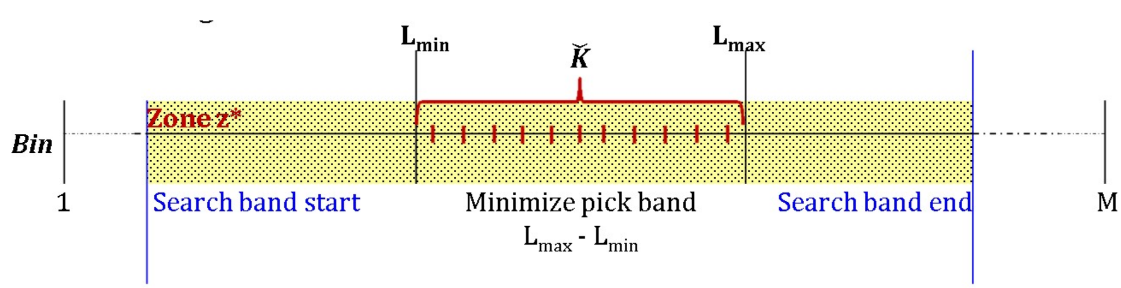


Figure 5.1 Pick band demonstration

The problem is then formulated as follows:

Objective:

Min L_{max} - L_{min}

Subject to:

$\sum_j \sum_b X_{j,b} = \Pi$ $\quad\forall\, j, b$ (5.1)

$\sum_b X_{j,b} \leq 1$ $\quad\forall\, j, b$ (5.2)

$\sum_j H_{i,j}.X_{j,b} \leq I_{i,b}$ $\quad\forall i, b$ (5.3)

$Lmax \ \geq B_{j,b}.X_{j,b}$ $\quad\forall\, j, b$ (5.4)

$Lmin \ \leq B_{j,b}.(M - (M-1)X_{j,b})$ $\quad\forall\, j, b$ (5.5)

Where:

$X_{j,b} \in \{0,1\}$ $\quad\forall\, j, b$ ($X_{j,b} = 1$ if order is assigned to the pick list)

Π Pick list size

$H_{i,j}$ Order quantity for item i in order j. For each j only one i is non-zero

$I_{i,b}$ Inventory of item i in bin b

$B_{j,b}$ Order j is fillable from bin b. $B_{j,b}$ =1 if both $H_{i*,j}$ and $I_{i*,b}$ are non-zero

$b \in M$ The highest bin number in the warehouse

Equation (5.1) ensures that exactly Π orders are selected, while equation (5.2) ensures that an order is filled only once. IFWs are physically very large and therefore divided into many picking zones which have at least one dedicated picker. However, the area assigned to a picker is still very large, so keeping the pick lists limited is assumed to shorten the fulfillment time. The solution of this model results in the assignment of orders to pick lists in order to minimize order fulfillment time.

In Chapter 4, an IFW where more than 22,000 orders are handled a day was investigated, and the role of Π in equation (5.1) is investigated as a parameter and it was found that a pick list with 10 to 15 orders performed best.

Another subject that must be considered is inventory control. Equation (5.3) ensures that ordered quantity of the associated SKU can never exceed the quantity in the bin. As explosion ratio increases, item dispersion increases. Consequently, the probability of finding an item in close proximity is greater. However, greater explosion ratios result in smaller quantities of each item in bins, which may be undesirable if ordered quantity is high.

Equations (5.4) and (5.5) measure the distance travelled to complete a pick list. Locations in a pick zone are assumed to be linear and numbered sequentially. The difference determines the travel distance to complete a pick list. L_{max} is calculated easily because it is the greatest bin number in the generated pick list. Calculating L_{min} may be trivial in cases where $X_{j,b}$ is zero, therefore adjustments are made to prevent it from becoming zero.

5.2 Decision Variable Space

The difficulty of MIP problems is said to be measured by the number of binary variables that is called the decision variable space (DVS). Although traditional storage policies have limited storage locations for SKUs the MIP may still be NP-hard. In the IFW problem, the decision space is much larger with many orders that can be picked from innumerable bins which further increases complexity and difficulty of the MIP problem.

We have identified our decision variables as a pending order j ($1 \leq j \leq J$) to be selected in a pick list to be picked from a bin location ($1 \leq b \leq M$) where it is stored. That is:

$$X_{j,b} \in \{0,1\} \qquad \forall\, j, b \; (X_{j,b} = 1 \text{ if order is in the pick list, 0 otherwise})$$

$$X_{j,b}, \qquad j = 1, 2, \ldots, J; \quad b = 1, 2, \ldots, M.$$

Therefore, the decision variable space is a JxM matrix. Considering the size of IFWs, the beehive-type storage systems, and the number of transactions made, the problem is impossible to solve without modification. The necessity of generating new pick lists as current pick lists are completed and new orders are received requires a reduction in problem size in order to obtain faster solutions.

The original problem in Chapter 4 has J=22,000 customer orders per day and M=3,240 bins to fulfill the orders from. The decision space consists of 22,000 x 3,240 = 71,280,000 decision variables. Since a mathematical programming problem with over a thousand variables is considered NP-hard, a problem with over 70 million variables is impossible to solve with known methods. In the following section, we investigate strategies to reduce the problem size and to shorten the solution time without any loss of data by generating lists with (i) order elimination by selecting orders based on arrival time, (ii) bin pre-selection by picking the orders from only some selected bins where the associated SKUs are stored and (iii) eliminating pre-selected bins that cannot be part of the optimal solution because of either inventory or location requirements. In addition, two heuristic algorithms are developed to select seeds that achieve the optimal solution within seconds.

5.3 Decision Variable Space Reduction

In this section, problem reduction is demonstrated with some examples. The decision variable space is a JxM matrix where J is a number of unfulfilled orders and M is a sequential number that counts storage locations where items are stored. An example of a decision variable space where there are 20 orders and location addresses of the items in each order may be similar to the following chart:

Order #	k (Bin locations)													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	11373	8206	3076	15801	16848	14176	12641	2021	653					
2	4039	3516	5512	14069	9424	8769	13668	16467	17951	767	7978	14291	2996	12040
3	5389	2119	2422	6598	9291	8497	1099	17360	1307	14899	12867			
4	11884	8441	12359	9633	1348	3723	13027	1737	7829	16814	13575	15256	12996	12572
5	9317	4895	9082	6412	337	13727								
6	11073	13492	9800	13062	16436	11643	11254	14148	17777	13804	16014	2738	3372	2072
7	2956	2342	12828	5860	4139	5124	3939	12843	3186	7986	1838	5530	4324	11133
8	17563	7266	16380	16293	8106	2403	17949	4471	14106	16174	1040	2943	16050	11184
9	13922	17990	11725	1836	17238	5630	15539	8053	10800	11037	9344	11307		
10	8589	11171	2468	1258	16601	6617	14496	88	14732	14665	5622	10727	6863	4981
11	2864	1594	12310	8746	7375	9071	8835	17528						
12	4040	10508	6522	926	7308	13331	12813	941	1241	1535	7441	6644	3928	3183
13	4266	444	8902	509	7391	1931	119	16174	596	10487	2518	1053	14908	13874
14	3084	141	13591	3947	2433	6987	13892	638	15842	17444	1799	5529	10982	8034
15	5075	11198	11423											
16	6191	11434	7030	17813	8075	2971	17416	13355	9720	6012	6819	15870	1136	
17	15258	3919	11802	12789	7805	7421	2641							
18	241	12608	9669	9351	8083	17796	3611	7757	11633					
19	1866	6099	11749	17761	2213	8211	17393	5476	4208	13694	17227	848		
20	3288	13814	1839	14005	17519	1434	6122	7435	15836	12887	9045	14508	6490	17877

Figure 5.2 Order-bin location map.

Increasing the size of the problem makes it more suitable to the nature of IFWs. Thus, the model is updated to 6000 commingled bins with 22,000 x 6,000 = 132,000,000 decision variables. The problem is extremely difficult or impossible to solve for an optimal solution as is. The DVS needs to be reduced using order reduction and bin selection methods.

5.3.1 Selection of Customer Orders

The primary goal of IFWs is immediate fulfillment by more efficiently placing orders in the first available truck. Incoming orders are clustered together and assigned to pickers immediately to eliminate any time loss. The visited IFW is one of the largest of its kind with about 400 orders per second on peak days. It is assumed that, on average, 100 orders are processed per second on non-peak days. Therefore, the MIP problem would need to be solved thousands of times a day. In order to minimize solution time of the MIP problem, the size of the decision variable matrix JxM must be reduced by first reducing J, the total number of orders. Let Ĵ be the first orders taken within a specified time interval. The pending order picking queue is described by the vector $\hat{O}_{j,t} = \{W_{j,t}, C_{j,t}, H_{j,t}\}$ where $W_{j,t}$ is the associated SKU, $C_{j,t}$ is order arrival time, and $H_{j,t}$ order quantity. Let $C_{j^*,t}$ be the receiving time of the earliest pending order, and $\hat{t}$ the designated time interval.

$$Z_{j,t} = \begin{cases} 1, & C_{j,t} - C_{j*,t} \leq \hat{t} \\ 0, & otherwise \end{cases} \tag{5.6}$$

$$\hat{J} = \sum Z_{j,t} \qquad \forall\, j \in J \tag{5.7}$$

This selection of orders reduces the decision variable space to ĴxM where $\hat{J} \subset J$. This adjustment will help us reach the immediate fulfillment objective in a more timely fashion. For example, instead of using all 22,000 orders received in the initial case, the earliest 100 orders can be worked with. This reduces the DVS from 132,000,000 to 100 x 6,000 = 600,000 variables.

5.3.2 Pre-Selection of Bins

Explosive storage strategy requires a very large number of storage locations for each SKU and items are stored in small quantities. Relatively, the bin locations that these SKUs are stored in often change over time as orders are fulfilled and new items are stored. Therefore, the pick list generation problem changes dynamically. This process requires complete digital control because of high transaction rates. Even after an elimination of orders, the MIP problem is still too large and needs further reduction. To handle this issue only some candidate pick locations are used for order fulfillment, such that a subset of bins $k \leq M$ that have the associated SKUs with the largest inventory in the warehouse is selected.

Considering our initial example, the pre-selection process reduces the decision space to 100 orders x 10 bins = 1,000 decision variables, which significantly reduces solution time.

5.3.3 Bin Weighing Method

While searching for more reduction methods, it was realized that there were some bins in the decision space that could not be included in the solution space, due to distance and inventory requirements needed to complete the pick list. The idea is very similar to how web search engines work. Web crawlers scan all websites and acquire data, which could be the page title, images, keywords, or any other pages they link to. All data are indexed and ranked based on their appearance on the websites and how often they were searched for. Then, when a query is made, a list of offers is retrieved starting from the highest ranked website to lowest.

Generally, users enter a query of a few keywords (Jansen et al., 2000).The search engine index already has the names of the sites containing the keywords ordered by their ranking. However, a combination of keywords requires more processing to bring out the best search results list. Every web page in the entire list must be weighed according to information in the indexes (Jawadekar, 2011). This is the more complicated part.

To mimic how search engines list the most relevant search results, two measures are created: Maximum Fillable Factor (*mff*) used to rank bins effectively, and Fillability Factor (*ff*) used to weight *mff* to bring out the more loaded bins that are located nearby.

Maximum Fillable Factor counts the number of pending orders that can be fulfilled from each bin in the warehouse. That is:

$$mff_b = \sum_j Y_{j,b} \text{ for } j \in \hat{O}_{j,t} \; \forall \, b \in M \tag{5.8}$$

where $Y_{j,b} \in \{0,1\}$, $Y_{j,b} = 1$ if order j is fillable at bin b

Fillablity Factor weighs *mff* in order to create a scalar of each bin in the warehouse to identify improved search ranges. A search bandwidth $\widehat{\Delta}$ is selected to search for the orders within a [+$\widehat{\Delta}$, - $\widehat{\Delta}$] range. A weight ω is given to all bins within the search bandwidth based on the distance between the central bin and the target bin, which is symbolized with δ. Bins closer to the center bin have a higher weight such that:

$$\omega_\delta = \left(1 - \frac{\delta}{\widehat{\Delta}}\right) \quad \text{for} \quad \delta \in \mp\widehat{\Delta} \tag{5.9}$$

The Fillablity Factor *ff* is calculated for each bin such that:

$$ff_b = \sum_{\widehat{\Delta}} \sum_j \; Y_{j,b-\delta} \; . \; \left(1 - \frac{\delta}{\widehat{\Delta}}\right) \quad \text{for } j \in \hat{O}_{j,t} \; \forall \, b \in M, \text{ and } \delta \in \mp\widehat{\Delta} \tag{5.10}$$

Figure 5.2 shows a simple example of *ff* calculation for bin 14887, assuming all other bins that are not shown are empty:

Bin number	MFF	Distance from central bin	Weight factor	FF
14877	0	10	0	0.2
14878	0	9	0.1	0.7
14879	0	8	0.2	1.2
14880	0	7	0.3	1.7
14881	0	6	0.4	2.4
14882	0	5	0.5	3.1
14883	0	4	0.6	3.8
14884	0	3	0.7	4.5
14885	0	2	0.8	5.2
14886	2	1	0.9	5.9
14887	3	0	1	6.2
14888	0	1	0.9	5.9
14889	0	2	0.8	5.6
14890	2	3	0.7	5.3
14891	0	4	0.6	4.6
14892	0	5	0.5	3.9
14893	0	6	0.4	3.2
14894	0	7	0.3	2.5
14895	0	8	0.2	1.8
14896	0	9	0.1	1.1
14897	0	10	0	0.6

2*(0.9)+3+2*(0.7)=6.2

Figure 5.3 An example to bin weighting.

In order to remove low-ranked bins that will not improve finding the optimal solution, a threshold is defined called Fillability Factor Threshold (*fft*), which is set on the *ff*. Bins that are less than that designated threshold are eliminated as well as the bins in a search bandwidth that cannot complete a pick list. That is:

$$\widehat{Y_{j,b}} = \begin{cases} 0, & \sum_{\hat{\Delta}} \sum_j Y_{j,b-\delta} \cdot \left(1 - \frac{\delta}{\hat{\Delta}}\right) \leq ff \\ & \text{and} \\ & \sum_{\hat{\Delta}} \sum_j Y_{j,b-\delta} \geq \prod \\ 1, & otherwise \end{cases} \tag{5.11}$$

$\widehat{\widehat{K}} = \sum \widehat{Y_{j,b}}$ $\qquad\qquad \forall\, b \in M,\ \delta \in \mp\widehat{\Delta}$

$\widehat{Y_{j,b}} \in \{0,1\}$ $\qquad\qquad \forall\, j, b$ ($\widehat{Y_{j,b}} = 0$ if order j is to be removed from the DVS)

After the reduction method, the DVS is reduced from $\hat{J}$x$\widehat{K}$ to $\hat{J}$x$\widehat{\widehat{K}}$ ($where\ \widehat{\widehat{K}} \leq \widehat{K}$).

5.4 Bin Weighted Order Fillability Algorithms (BWOF)

When solving MIP problems, there are various methods and tricks to shorten solution time. In addition to DVS reduction, another approach is introduced that generates seeds to solve MIP problems instantly. We call this seed generation technique Bin Weighted Order Fillability Algorithm (BWOF). The comparison of the full matrix solution, reduced matrix solution, and instant seed solutions will be presented in the next section.

Seed algorithms construct batches in two phases: seed selection and order congruency (De Koster, Roodbergen, 2007). Seeds act as a starting point in the solution of clustering algorithms. Then order congruency rules determine which unassigned order should be added into the pick list. Seed algorithms are introduced by Elsayed (1981) and Elsayed and Stern (1983) for routing operations in automated warehouses. More algorithms are considered in Hwang et al. (1988), Hwang and Lee (1988), and Pan and Liu (1995), Gibson and Sharp (1992), Rosenwein (1994), Ruben and Jacobs (1999). Existing examples are investigated in traditional warehousing environments where batch picking strategies were adopted. De Koster (1999) investigated and compared some of seed selection rules that are: a random order, an order with large number of positions, an order with longest pick tour, an order which is most distantly-located, and order with the largest difference between the aisle number of the right-most and the left-most aisle to be visited.

When a seed is plugged into MIP Solver, first the seed bin is examined to identify pending orders that can be fulfilled from. Next, the nodes within the search range with the seed are and evaluated in order to find the closest bin that can fulfill an awaiting order. This process continues until the shortest path is found to complete a pick list. We tested two seed algorithms for validation and compared the results with the optimal solution in the following section.

5.4.1 Heuristic 1 – Highest Maximum Fillable Factor

The explosive storage policy and bins with commingled SKUs increase the probability of the number of pending orders that can be picked from a tight picking area as shown in Chapter 3. As our first seed, we use the bin with the highest *ff* since the associated bin is in the center of a range which is surrounded with other bins with high *ff*'s. In an ideal case, one pick list can be completed from one bin, and it would take just a few seconds to generate that particular list. However, due to high labor costs of storage, very high explosion ratios are not recommended. The calculation of seed 1 is as follows:

$$S_1 = Max\,\{ \textstyle\sum_{\hat{\Delta}} \sum_j \; Y_{j,b-\delta} \; . \left(1 - \frac{\delta}{\hat{\Delta}}\right) \quad \text{for } j \in \hat{O}_{j,t} \;\forall\, b \in M, \text{ and } \delta \in \mp\hat{\Delta}\} \quad (5.12)$$

5.4.2 Heuristic 2 – Amplified Maximum Fillable Factor

While testing Seed 1, it was found that the highest *ff* does not always find the best batch in cases where Seed 1 returns same measure for multiple bins. We improved the first algorithm which amplifies the weighing effect, so that the range where more orders can be fulfilled from will return a higher value. This was handled by multiplying each bin *ff* with the total number of orders that are fillable from that bin:

$$S_2 = Max\,\{\textstyle\sum_{\hat{\Delta}} \sum_j \; Y_{j,b-\delta} \, . \left[\sum_{\hat{\Delta}} \sum_j \; Y_{j,b-\delta} \, . \left(1 - \frac{\delta}{\hat{\Delta}}\right) \;\right] \text{for } j \in \hat{O}_{j,t} \;\forall\, b \in M, \text{ and } \delta \in \mp\hat{\Delta}\} \quad (5.13)$$

The procedure to solve the MIP problem is summarized in Figure 5.3:

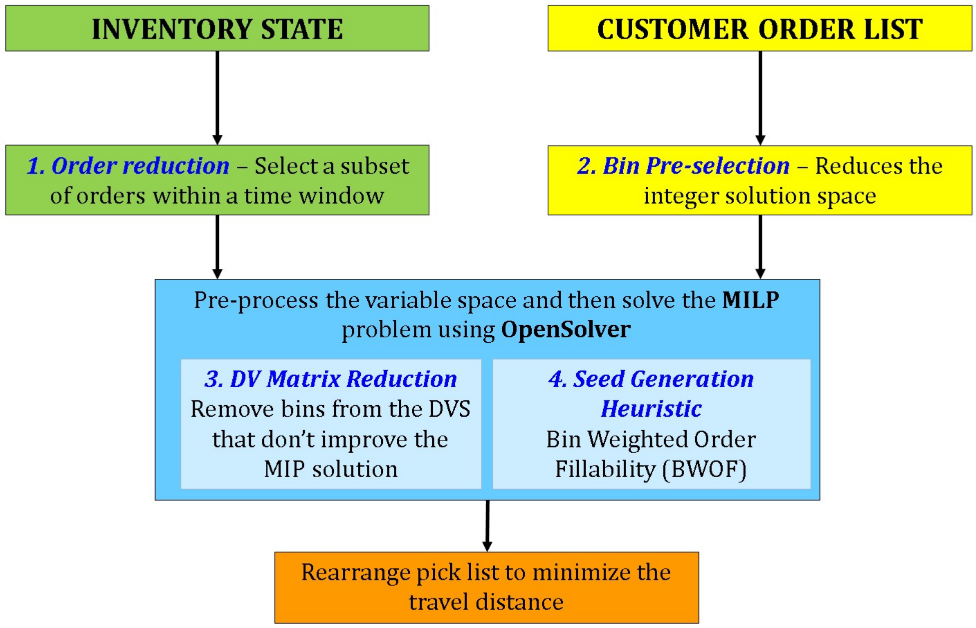


Figure 5.4. MIP pick list generation procedure.

5.5 Experimental Design

Experiments are performed to test MIP solution time advantages of reduction methods, and seed algorithms. Three different warehouse sizes (A, B and C) are considered to analyze the power of our solution methods as problem size increases. Also, we aimed to prescribe a pattern for pick list size and pickers' search bandwidth for varying problems that minimize the travel distance. Design of the cases are given in Table 5.1:

Table 5.1 Experimental Design Problem Types

Case	Number of Pre-selected Orders	Number of Alternative Bin Locations
A	100	10
B	120	12
C	130	14

5.5.1 Factors and Factor Levels

Experiment factors include Pickers Search Bandwidth ($\check{\lambda}$), and Fillability Factor Threshold (*fft*). Experimental factor setting is shown in Table 5.2.

Table 5.2 Experimental Design Factors and Factor Levels

Factors	**Factor Levels**	
Case	A, B, C	(3 Levels)
Problem Type (T)	1, 2, 3, 4, 5	(5 Levels)
Pick List Size ($\prod$)	8, 10, 12	(3 Levels)
Pickers Search Bandwidth ($\check{\lambda}$)	10, 15	(2 Levels)
Fillability Factor Threshold (*fft*)	3.5, 4, 4.5, 5	(4 Levels)

There are 3x5x3x2x4=360 cells are in the results table to test the levels of $\prod$, $\check{\lambda}$ and *fft*.

5.5.2 Factors for Seed Algorithms

For seed algorithms, each case is examined for five different problems and three levels of pick list size: 8, 10, and 12. Hence, there are 3x5x3=45 different problems. All problems are solved for (i) optimal and (ii) the two seed algorithms. Thus, there are 3x5x3x3=135 additional cells in the results table. Experimental factor setting is shown in Table 5.3.

Table 5.3 Factors for Seed Algorithms

Factors	Factor Levels	
Case	A, B, C	(3 Levels)
Problem Type (T)	1, 2, 3, 4, 5	(5 Levels)
Pick List Size (∏)	8, 10, 12	(3 Levels)
Solution for optimal	Full matrix	
Seed Algorithms (S)	1, 2	(2 Levels)

Total number of the results table is 360+135=495 cells.

5.6 Solving the MIP

Upon effectively reducing the decision variable space of the MIP problem, a solution method must be chosen that is proficient and has computational advantages. Even though remarkable progress in solving NP-hard problems has been made in recent years, it is still difficult to handle everyday scenarios that are larger and more complex in practice. Recent features added to MIP solvers at the algorithmic and hardware level have contributed gradually in solving more complex problems. Therefore, it is now possible to solve problems that were considered difficult or impossible to solve in the past (Lima and Grossmann, 2011).

The most widely used solution technique by MIP solvers is the Branch and Bound approach where an upper bound is found by choosing any point in the region, or a point found by a local optimization method and a lower bound is found from convex relaxation, duality, Lipschitz or other bounds. The problem then is partitioned and the nodes that do not improve the existing solution are eliminated until the upper bound defined by integral solutions is equal to the lower bound given by fractional solutions.

5.6.1 OpenSolver

We formulated the MIP using the OpenSolver Advanced version to solve the experimental problems. OpenSolver is a MS Excel Add-in that works just like the built-in solver. It offers a range of optimization engines like COIN-OR, and CBC and can convert problems to AMPL to be solved with NEOS Solver, and Gurobi. It is particularly suitable for solving IFW picking problems, because there is no limit on the number of constraints or problem size. It reads almost all Excel functions and does the calculations on the sheet. It also color codes decision variables, constraints and the objective. A medium size problem built in OpenSolver looks like this:

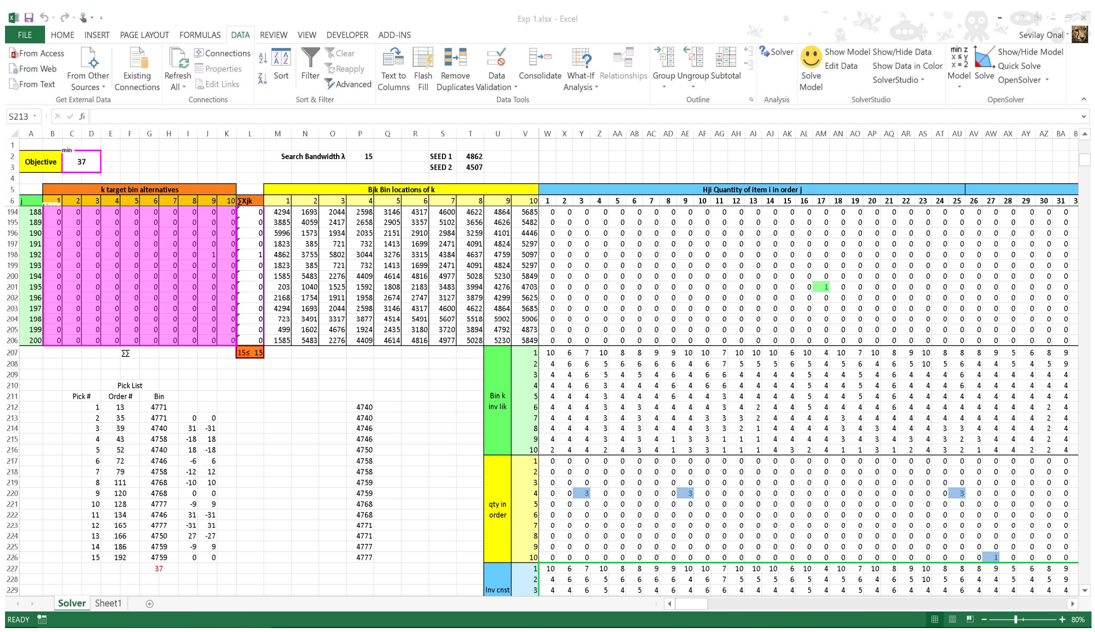


Figure 5.5 OpenSolver problem design.

OpenSolver uses an excel sheet for calculations. The model is built in a different window where the objective, decision variable space, constraints, preferred solver, upper time and iteration limits are defined. This particular problem has 200 unfulfilled customer orders, of which 15 are batched in a pick list to be fulfilled from 10 bin locations

(M7:V206). Binary decision variables define the orders that are in the pick list (B7:K206). Vector (L7:L2016) defines if an order is selected, and cell L207 sets the sum of selected orders to the defined constraint ∏=15. Inventory sufficiency is controlled on the right side which ensures that it never goes below zero. On the bottom left, a pick list that minimizes the shortest distance is generated. The list is ordered by order number by default. Actual route of the pick list is calculated in (P212:P226).

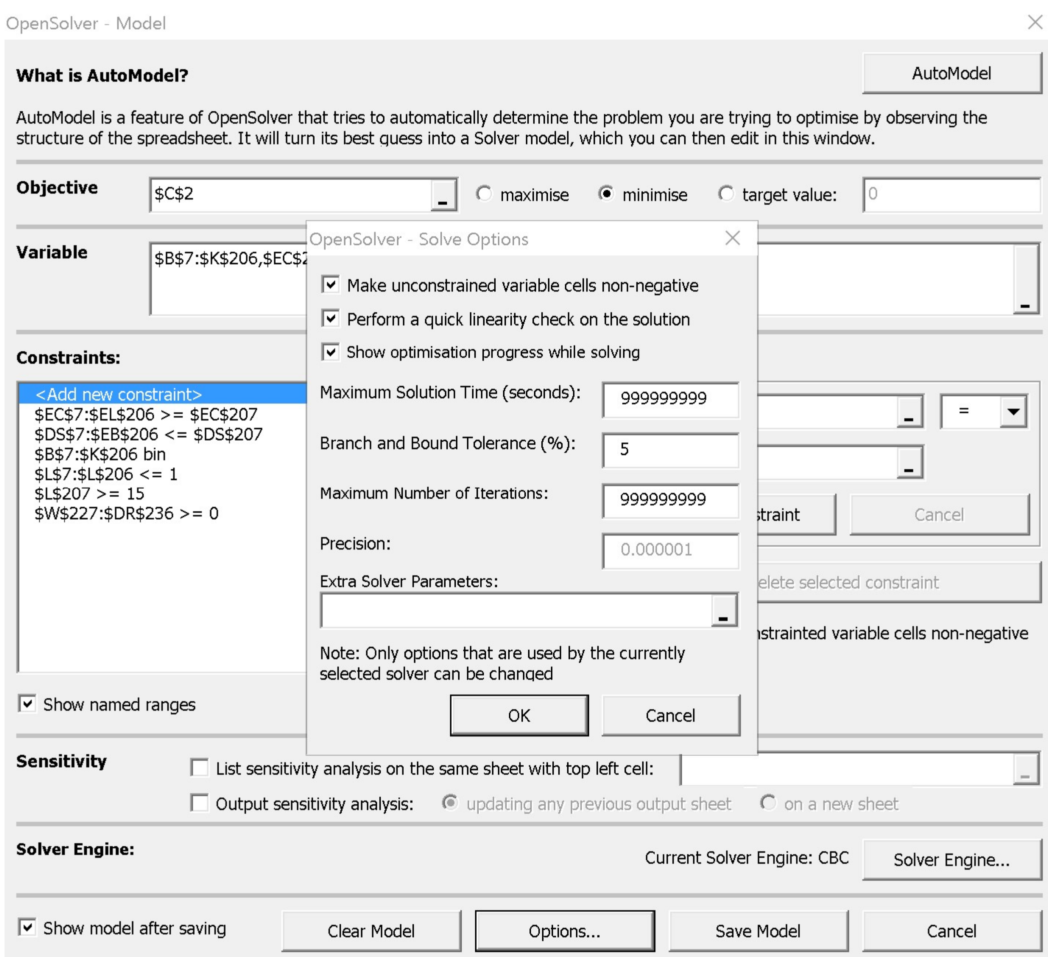


Figure 5.6 OpenSolver model construction.

In the model window, the objective is defined on top; decision variables right underneath the objective; constraints on the left side. Additional options offered are preferred solver, sensitivity analysis, linearity check, maximum solution time, number of iterations, branch and bound tolerance etc. It also offers a progress check window that is updated after evaluating every couple of hundred nodes. When all required information is entered correctly, the solver color codes the matrices to increase visual detectability.

The greatest benefit of using explosive storage and commingled bins is that pickers have the advantage of reusing the same bin to fulfill more than one order. In this particular problem, two orders are fulfilled from bin 4740, two from 4746, one from 4750, two from 4758, two from 4759, two from 4768, two from 4771, and two from 4777 sums up to 15.

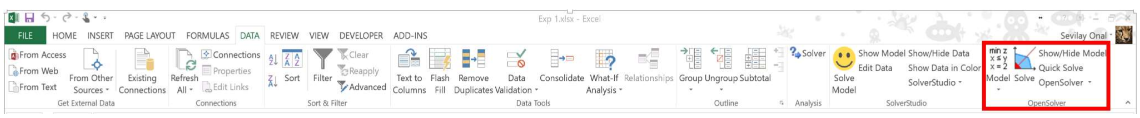


Figure 5.7 OpenSolver executing the model.

The model window is reached by clicking the Model button, and the problem is solved for the optimal by clicking the solve button. When solving a problem using seeds, a constraint is added to set the corresponding decision variable to 1. Consequently, the solver will remove the bins by eliminating the sub-problems using the branch and bound method. When solving reduced matrix problems, the alternative bin location matrix is updated by setting the fillability threshold to the desired level.

5.7 Experimental Analysis

5.7.1 General Performance

Figure 5.7 shows the comparison between the solution methods of our base case problem A, where there are $\hat{J} = 100$ orders to be picked from $\hat{K} = 10$ alternative bins. The DVS matrix has $\hat{J}$x$\hat{K} = 1000$ cells. The reduced matrix is solved by using two levels of search bandwidth and 4 levels of fillability threshold. As the threshold increases, the number of cells to be evaluated is reduced and therefore solution time is shorter. Seed heuristic solutions are found instantaneously.

For problems where the optimal could not be found using OpenSolver we consider the best results as the sub-optimal solutions which allow for comparisons of solution methods. In analyzing the general performance, only the solutions are displayed regardless of execution time and number of decision variables used.

Figure 5.7 shows that out of 15 problems, six problems were not solvable using the full matrix. The reducing method provided the optimal or the suboptimal solution for all problems except for problem A4- 10. The seed solutions were able to find the optimal or the suboptimal in 8 different cases. For problems that seed heuristics couldn't find the exact optimal or suboptimal, the difference is usually not significant. The right two columns show the percent difference between the lowest solution and the seed solution.

Problem	Pick List size	Optimal solution	Reduced Matrix Solution Search Bandwidth		Seed 1 Solution	Seed 2 Solution	Seed 1 %difference	Seed 2 %difference
			10	15				
A1	8	**13**	**13**	**13**	17	14	30.8%	7.7%
	10	**19**	**19**	**19**	35	**19**	84.2%	0.0%
	12	**24**	**24**	**24**	49	**24**	104.2%	0.0%
A2	8	**11**	**11**	**11**	**11**	**11**	0.0%	0.0%
	10	--	**16**	**16**	21	**16**	31.3%	0.0%
	12	--	**27**	**27**	**27**	39	0.0%	44.4%
A3	8	**11**	**11**	**11**	13	**11**	18.2%	0.0%
	10	**18**	**18**	**18**	22	**18**	22.2%	0.0%
	12	**25**	**25**	**25**	54	27	116.0%	8.0%
A4	8	**11**	**11**	**11**	12	13	9.1%	18.2%
	10	--	20	20	**19**	21	0.0%	10.5%
	12	--	**26**	**26**	33	47	26.9%	80.8%
A5	8	**10**	**10**	**10**	15	15	50.0%	50.0%
	10	--	**19**	**19**	27	27	42.1%	42.1%
	12	--	**25**	**25**	36	36	44.0%	44.0%

Figure 5.8 Optimal to seed solution comparison of type A problems.

Problem solution times depend on the computer being used. When using an average computer, a full matrix may take days to weeks and still be impossible to solve. The matrix reduction method generally takes up to a few hours, whereas seed heuristics provide a solution instantly.

As shown in Table 5.7, the matrix reduction method found the minimum solution in 14 out of 15 cases. However, we prefer using seed heuristics since they provide instant solutions. If we consider both seed heuristics, in 8 out of 15 problems, an optimal or suboptimal solution is found within seconds. In only 3 problems the pick route is up to 50% greater than the minimum solution found.

The second case, Problem B, is a larger problem where $\hat{j} = 120$ orders and $\hat{k} = 12$ bins, therefore the full DVS matrix has $\hat{j} \text{ x } \hat{k} = 1440$ cells. The table below displays the results using the provided solution methods:

Problem	Pick List size	Optimal solution	Reduced Matrix Solution Search Bandwidth 10	15	Seed 1 Solution	Seed 2 Solution	Seed 1 %difference	Seed 2 %difference
B1	8	--	19	19	**16**	**16**	0.0%	0.0%
	10	--	31	31	42	**25**	68.0%	0.0%
	12	43	68	68	50	**31**	61.3%	0.0%
B2	8	--	20	20	18	**17**	5.9%	0.0%
	10	--	32	**31**	45	**31**	45.2%	0.0%
	12	**35**	**35**	**35**	66	**35**	88.6%	0.0%
B3	8	--	**13**	**13**	**13**	16	0.0%	23.1%
	10	--	**25**	**25**	43	27	72.0%	8.0%
	12	--	**41**	**41**	77	48	87.8%	17.1%
B4	8	--	19	19	**17**	27	0.0%	58.8%
	10	--	**25**	**25**	52	35	108.0%	40.0%
	12	--	47	47	67	**42**	59.5%	0.0%
B5	8	--	**15**	**15**	17	17	13.3%	13.3%
	10	--	28	27	**26**	29	0.0%	11.5%
	12	--	47	40	**38**	45	0.0%	18.4%

Figure 5.9 Optimal to seed solution comparison for type B problems.

Table 5.8 shows that seed heuristics perform better than the matrix reduction method for type B problems. Seed heuristics found an optimal or a suboptimal solution in 11 out of 15 problems whereas the matrix reduction methods could only find 7. Using the seed algorithm, there is only one problem where the solution is 40% greater than the minimum found. Still, overall performance significantly improved.

The third case, type C is the largest case with $\hat{j} = 130$ and $\hat{k} = 14$ bins. The DVS matrix has $\hat{j}\,\mathrm{x}\,\hat{k} = 1820$ cells. The table below displays the results using the provided solution methods:

Problem	Pick List size	Optimal solution	Reduced Matrix Solution Search Bandwidth 10	Reduced Matrix Solution Search Bandwidth 15	Seed 1 Solution	Seed 2 Solution	Seed 1 %difference	Seed 2 %difference
C1	8	--	26	**25**	40	31	60.0%	24.0%
	10	--	**33**	**33**	65	38	97.0%	15.2%
	12	--	84	84	84	**63**	33.3%	0.0%
C2	8	--	**26**	**26**	37	**26**	42.3%	0.0%
	10	--	**49**	**49**	52	55	6.1%	12.2%
	12	--	79	79	78	**77**	1.3%	0.0%
C3	8	--	**20**	**20**	**20**	**20**	0.0%	0.0%
	10	--	42	42	**35**	**35**	0.0%	0.0%
	12	--	69	69	**66**	**66**	0.0%	0.0%
C4	8	--	**19**	**19**	28	28	47.4%	47.4%
	10	--	**37**	**37**	49	49	32.4%	32.4%
	12	--	73	73	**64**	**64**	0.0%	0.0%
C5	8	--	**21**	**21**	25	25	19.0%	19.0%
	10	--	**31**	**31**	**31**	62	0.0%	100.0%
	12	--	82	82	**51**	88	0.0%	72.5%

Figure 5.10 Optimal to seed solution comparison for type C problems.

In type C problems, seed heuristics perform better than type A but worse than type B. The matrix reducing method and the seed heuristics found the minimum solution in 9 out of 15 cases (not necessarily the same cases). Overall, performance comparison of seed heuristics for all problem types is provided below:

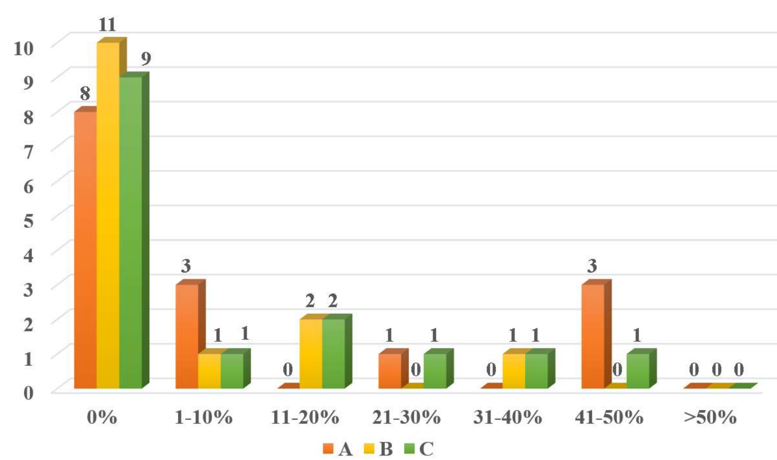


Figure 5.11 Overall performance comparison.

Figure 5.10 shows that out of 45 cases, 28 are optimal or suboptimal were found by using either of the two seed algorithms. In 5 cases the lowest seed solutions are within 10% of the optimal. This sums up to an accuracy of approximately 75% in finding an optimal or a suboptimal solution within 10%. Similarly, 82% accuracy is obtained where the seed solution is within 20% of the optimal. The seed heuristics should show significant improvement over the narrow band pick algorithm and shorten the time to generate and complete a picklist. To measure the efficiency and the power of the seed heuristics, the narrow band picking algorithm in the simulation created in Chapter 4 is updated with the seed heuristics.

5.7.2 Average Pick Time Behavior

In order to make an enhanced comparison, the data obtained from the experiments is standardized using the average distance per pick for each group of problems.

Pick List Size	Minimum Seed Solution	Avgerage Pick Distance	Minimum Seed Solution	Avgerage Pick Distance	Minimum Seed Solution	Avgerage Pick Distance
	A		B		C	
8	14	1.75	16	2.00	31	3.88
	11	1.38	17	2.13	26	3.25
	11	1.38	13	1.63	20	2.50
	12	1.50	17	2.13	28	3.50
	15	1.88	17	2.13	25	3.13
Average		**1.58**		**2.00**		**3.25**
10	19	1.90	25	2.50	38	3.80
	16	1.60	31	3.10	52	5.20
	18	1.80	27	2.70	35	3.50
	19	1.90	35	3.50	49	4.90
	27	2.70	26	2.60	31	3.10
Average		**1.98**		**2.88**		**4.10**
12	24	2.00	31	2.58	63	5.25
	27	2.25	35	2.92	77	6.42
	27	2.25	48	4.00	66	5.50
	33	2.75	42	3.50	64	5.33
	36	3.00	38	3.17	51	4.25
Average		**2.45**		**3.23**		**5.35**

Figure 5.12 Average pick distance for all problem types.

Smaller lists take less time to complete. However, the average pick time is not enough to make a comparison and setup/drop-off times must be taken into consideration. Generating small lists will result in a greater number of total lists to complete all orders received in a day. Consequently, more time will be spent in setup/drop-off. The table below shows a more realistic comparison of the cases where a total number of 15 lists of 8 items, 12 lists of 10 items and 10 lists of 12 items will be generated to fulfill 120 received orders. Assuming setup and drop-off times are measured in terms of distance on a scale from 10 to 18 units, total and average units to 120 orders are calculated as it follows:

Pick List Size	Setup Time	Minimum Seed Solution	Distance to complete 120 orders	Minimum Seed Solution	Distance to complete 120 orders	Minimum Seed Solution	Distance to complete 120 orders
		A		B		C	
8	10	14	164	16	166	31	181
	12	11	191	17	197	26	206
	14	11	221	13	223	20	230
	16	12	252	17	257	28	268
	18	15	285	17	287	25	295
Average			**222.6**		**226**		**236**
10	10	19	139	25	145	38	158
	12	16	160	31	175	52	196
	14	18	186	27	195	35	203
	16	19	211	35	227	49	241
	18	27	243	26	242	31	247
Average			**187.8**		**196.8**		**209**
12	10	24	124	31	131	63	163
	12	27	147	35	155	77	197
	14	27	167	48	188	66	206
	16	33	193	42	202	64	224
	18	36	216	38	218	51	231
Average			**169.4**		**178.8**		**204.2**

Figure 5.13 Average pick list completion times.

For each problem type, as pick list size increases, total time to fulfill the 120 orders decreases. Additionally, as warehouse size increases, time to fulfill the 120 orders also increases, due to the reduced dispersion of the SKUs within the warehouse.

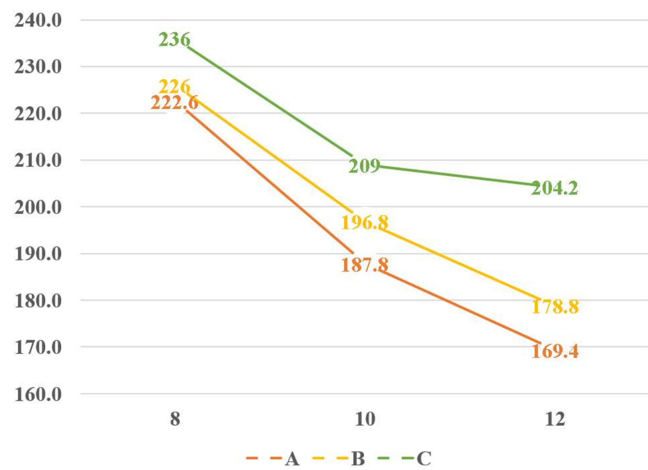


Figure 5.14 Average pick list completion time behavior for all problem types.

Simple results show that seed algorithms reduce order fulfillment times significantly. Larger lists work better to minimize average fulfillment time independent of warehouse size.

5.8 Simulation Analysis of BWOF Algorithms

In consideration of measuring the pick time performance of the BWOF seed algorithms, two developed heuristics: Highest Maximum Fillable Factor and Amplified Maximum Fillable Factor are replaced with the basic Narrow Band Order Picking Algorithm. In Narrow Band Order Picking Algorithm, the best performing pick list sizes were Ǩ =13, 15, 17. Therefore, when running the updated simulation, only Ǩ =13, 15, 17 are tested.

In order to apply the seeds into the simulator, required measures for each bin are added as columns in the Location Table where each bin appears only once. That are:

- *mff*; Maximum fillable factor (number of different SKU's in a bin)
- *ff*; Fillability factor (weighted *mff* within the range)
- *iff*; Improved Fillability Factor (*ff* multiplied with amplification)

5.8.1 Generating Pick Lists Using BWOF Algorithms

The picking algorithm used in the simulation in Chapter 4 is updated with the BWOF seed algorithms. The difference comes from a set of rules to select the first stop in a pick list. The two BWOF heuristics return two seeds. In most cases the two seeds address the same bin. When they are different, both seeds are used to generate two different pick lists, and the list that provides the shortest fulfillment time is selected and assigned to a picker. Therefore, most of the steps in the generation process remained the same such as assigning the first pick to the earliest free picker and the search to increment pick stops in a list's search range. Following notations are added:

ff_{b^*}	Weighted number of orders that are fillable from bin b^*
iff_{b^*}	Improved fillablilty factor of bin b^*
f_1,f_2	Picking list number, set f_1,f_2=0 at the start of a period

Pick list generation process is updated using the BWOF algorithms in order to create the order pick lists from the picking queue vector $\hat{O}_{j,t}$. The proposed algorithm is:

1. Initiate the algorithm by setting t=1.

2. We assume the picking shift starts at time zero and ends at T_P. Let ϕ_p be the free or available time of a picker, and at the start of period set ϕ_p =0 for all p.

3. Add all orders for t to the order queue $\hat{O}_{j,t}$. If there an unfilled order from the previous period they are also added to $\hat{O}_{j,t}$.

4.1 Select the first stop in list f to be the first seed, which is the bin with the highest ff_{b^*} such that:
$ff_{b^*} = Max\,\{ \sum_{\hat{\Delta}} \sum_j \; Y_{j^*,b^*-\delta} \;.\; \left(1 - \frac{\delta}{\hat{\Delta}}\right)$ for $j \in \hat{O}_{j,t}$ and $\delta \in \mp\hat{\Delta}\}$

4.2 Identify the next available picker p^* in the zone where b^* is located.

ϕ_{p^*} =Min{ ϕ_p | all p}. If $\phi_{p^*} > T_p$ then no more orders can be processed in this period, go to step 9. Let z^* be such that $p^* \in z^*$ where $B_{min,z^*} \leq b^* \leq B_{max,z^*}$. Initiate a new stocking list f_1=f+1 and assign $\acute{P}_{f1,t} = p^*$.

4.3 Select the earliest pending order j^* that could be fulfilled from b* in the queue such that:

$C_{j^*,t}$ = Min{ $C_{j,t}$ | $I_{i,b^*,t} \geq H_{j,t}$ for some $B_{min,z^*} \leq b^* \leq B_{max,z^*}$ where $i = W_{j,t}$, for $j \in \hat{O}_{j,t}$}
Set $G_{1,f1,t}$=j^* and $\dot{D}_{k,f1,t} = b^*$.

Since the picking cannot start till the next customer order arrives set $\delta_{1,f1,t}$=$Max(\rho + C_{j^*,t}, \phi_{p^*})$, where is a start delay to allow for more incoming orders to be included in the current pick.

In a high volume IFW, though, most often we can expect $\delta_{1,f1,t} = \phi_{p^*}$. Set $\delta_{2,f1,t}$ = $(\delta_{1,f1,t} + \tau_e + \tau_w H_{j^*,t})$, and η=b*. Remove j^* from the queue list $\hat{O}_{j,t}$. and update the inventory $I_{i,b^*,t} = I_{i,b^*,t} - H_{j^*,t}$ where $i = W_{j^*,t}$.

4.4 Increment the stop to f_1=f+1. Add orders to the pick list by searching b* for earliest pending order j^* in the queue that can be fulfilled within a band of $\pm\widehat{\Delta}$ bins from the current location.

$C_{j^*,t}$ = Min{ $C_{j,t}$ | $C_{j,t} \leq \delta_{1,f1,t}$, $I_{i,b^*,t} \geq H_{j,t}$ for some $B_{min,z^*} \leq b^* \leq B_{max,z^*}$ and
$\eta - \widehat{\Delta} \leq b^* \leq \eta + \widehat{\Delta}$ where $i = W_{j,t}$, for $j \in \hat{O}_{j,t}$}

If no j^* meets the condition then the picking list is ended and go to step 7. Else, Set $\dot{D}_{k,f,t} = b^*$, $\delta_{2,f1,t} = (\delta_{2,f1,t} + \tau_b|\eta - b^*| + \tau_w H_{j^*,t})$, and η=b*. Remove j^* from the queue list $\hat{O}_{j,t}$. and update the inventory $I_{i,b^*,t} = I_{i,b^*,t} - H_{j^*,t}$ where $i = W_{j^*,t}$.

4.5 If any of the list stopping conditions is met then no more orders can be added to the list: (i) The queue list $\hat{O}_{j,t}$ is empty (ii) Maximum stops $k = \check{K}$ or (iii) Shift has ended $\delta_{2,f1,t} > T_P$. Else return to step 6 to add more orders.

5.1 Select the first stop in list f to be the second seed, which is the bin with the highest iff_{b^*} such that:

$iff_{b^*} = Max$ $\{\sum_{\hat{\Delta}} \sum_j Y_{j,b-\delta} \cdot \left[\sum_{\hat{\Delta}} \sum_j Y_{j,b-\delta} \cdot \left(1 - \frac{\delta}{\hat{\Delta}}\right)\right]$ for $j \in \hat{O}_{j,t}$ and $\delta \in \mp\hat{\Delta}$
where $i = W_{j,t}$, for $j \in \hat{O}_{j,t}$}

5.2 Identify the next available picker p^* in the zone where b^* is located.

ϕ_{p^*} =Min{ ϕ_p | all p}. If $\phi_{p^*} > T_p$ then no more orders can be processed in this period, go to step 9. Let z^* be such that $p^* \in z^*$ where $B_{min,z^*} \leq b^* \leq B_{max,z^*}$. Initiate a new stocking list f_2=f+1 and assign $\acute{P}_{f2,t} = p^*$.

5.3 Select the earliest pending order j^* that could be fulfilled from b* in the queue such that:

$C_{j^*,t}$ = Min{ $C_{j,t}$ | $I_{i,b^*,t} \geq H_{j,t}$ for some $B_{min,z^*} \leq b^* \leq B_{max,z^*}$ where $i = W_{j,t}$, for $j \in \hat{O}_{j,t}$}
Set $G_{1,f2,t}=j^*$ and $\dot{D}_{k,f2,t} = b^*$. Since the picking cannot start till the next customer order arrives set $\delta_{1,f2,t} = Max(\rho + C_{j^*,t}\ ,\ \phi_{p^*})$, where is a start delay to allow for more incoming orders to be included in the current pick.

In a high volume IFW, though, most often we can expect $\delta_{1,f2,t} = \phi_{p^*}$. Set $\delta_{2,f2,t}$ = $(\delta_{1,f2,t} + \tau_e + \tau_w H_{j^*,t})$, and η=b*. Remove j^* from the queue list $\hat{O}_{j,t}$. and update the inventory $I_{i,b^*,t} = I_{i,b^*,t} - H_{j^*,t}$ where $i = W_{j^*,t}$.

5.4 Increment the stop to f_2=f+1. Add orders to the pick list by searching b* for earliest pending order j^* in the queue that can be fulfilled within a band of $\pm\widehat{\Delta}$ bins from the current location.

$C_{j^*,t}$ = Min{ $C_{j,t}$ | $C_{j,t} \leq \delta_{1,f2,t}$, $I_{i,b^*,t} \geq H_{j,t}$ for some $B_{min,z^*} \leq b^* \leq B_{max,z^*}$ and
$\eta - \widehat{\Delta} \leq b^* \leq \eta + \widehat{\Delta}$ where $i = W_{j,t}$, for $j \in \hat{O}_{j,t}$}
If no j^* meets the condition then the picking list is ended and go to step 7. Else, Set $\dot{D}_{k,f,t} = b^*$, $\delta_{2,f2,t} = (\delta_{2,f2,t} + \tau_b|\eta - b^*| + \tau_w H_{j^*,t})$, and η=b*. Remove j^* from the queue list $\hat{O}_{j,t}$. and update the inventory $I_{i,b^*,t} = I_{i,b^*,t} - H_{j^*,t}$ where $i = W_{j^*,t}$.

5.5 If any of the list stopping conditions is met then no more orders can be added to the list: (i) The queue list $\hat{O}_{j,t}$ is empty (ii) Maximum stops $k=\check{K}$ or (iii) Shift has ended $\delta_{2,f2,t} > T_P$. Else return to step 6 to add more orders.

6. Compare list completion times and select f^* is completed earlier such that:
f^* = Min {$\delta_{2,f1,t}$, $\delta_{2,f2,t}$}

7. If there are pending orders in $\check{R}_{\xi,t}$ then go to step 4.

8. If $t<T$ then update $t=t+1$ and go to step 3. Else stop.

5.8.2 General Performance

The experimental problem in Chapter 4 is solved using the BWOF algorithm:

Table 5.4 Problem Parameters

N = 400 SKUs	M = 3240 Bins	β = 3000 in^3
Z = 9 (Equal Size)	S_z = 6/zone	P_z = 6/zone
T = 9 days	T_s = 8 Hours	T_p = 8 Hours
$\sum_t R_t = 220$	$\sum_t J_t = 22000$	$K = 10$

Table 5.5 demonstrates a comparison of average fulfillment time (in minutes) of the Narrow Band Order Picking Algorithm (referred as old) and the Bin Weighed Order Fillability Algorithm (referred as new). We compared only the best performing pick list sizes for the NBOP algorithm in the first simulation. That are Ǩ = 13, 15, and 17.

Table 5.5. The Comparison of Mean Fulfillment Time

	13		**15**		**17**	
	old	**seed**	**old**	**seed**	**old**	**seed**
0.1	69.527	71.87837	70.578	74.020	71.692	74.450
0.2	64.469	60.05638	67.079	58.325	69.621	54.925
0.3	62.539	54.13764	64.045	49.629	66.651	48.498
0.4	62.011	50.71415	63.856	47.216	66.816	45.209
0.5	60.805	49.39303	62.853	46.241	66.563	43.008
0.6	59.574	50.837	61882	46.894	64.737	43.098
0.7	59.140	51.65241	60.636	46.420	64.352	42.312
0.8	57.377	53.70685	58.614	46.078	62.426	41.851
0.9	58.265	56.78663	59.749	48.412	63.160	42.560

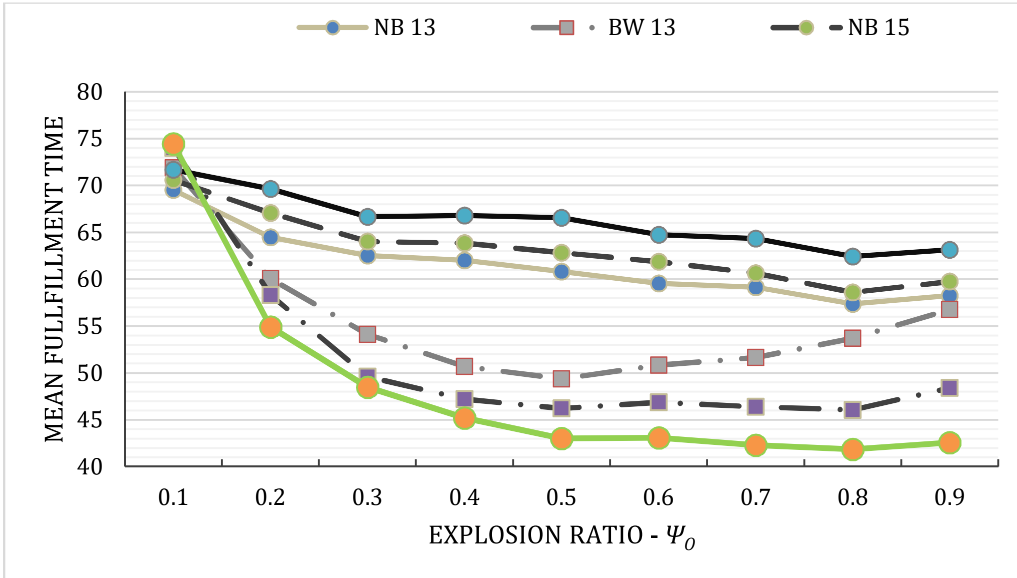


Figure 5.15 The comparison of fulfillment time as function of explosion ratio.

The chart above shows that the BWOF algorithm performs better with explosive storage policy than NBOP algorithm. Chapter 4 results showed that among three pick lists used, Ǩ = 17 has the weakest performance followed by Ǩ = 15 and Ǩ = 13, respectively. An interesting finding is that the new algorithm has the performance order Ǩ flipped. This is because the travel distance of pickers are shortened considerably which reversed the trade-off between setup time to load/unload and travelling time.

Another interesting finding is the optimal explosion rate. Using the old algorithm, it was found that $\Psi_o = 0.8$ provided the shortest fulfillment times across all pick list sizes. Whereas the new algorithm's optimal Ψ_o is 0.5. Besides fulfillment time advantages, this drop allows for a reduction in labor costs of storage operations since higher explosion ratios require more handling in smaller quantities. After $\Psi_o = 0.5$, average fulfillment time raises slightly, but tends to remain flat.

Table 5.6 Percent Improvements in Fulfillment Time Using BWOF Algorithm

	13	**15**	**17**
0.2	6.84%	13.58%	22.79%
0.3	13.43%	23.05%	29.03%
0.4	18.22%	26.83%	34.84%
0.5	18.77%	27.32%	38.74%
0.6	14.67%	25.16%	36.32%
0.7	12.66%	24.04%	37.27%
0.8	6.40%	21.85%	35.86%
0.9	2.54%	19.46%	35.35%
Average	11.69%	22.66%	33.78%

Table 5.6 presents percentage improvements on fulfillment time as a function of $\check{K}$. The average gain of improving the NBOP algorithm to the BWOF algorithm is 11.69%, 22.66%, 33.78% for $\check{K}$ = 13, 15, and 17, respectively. For each list size Ψ_o = 0.5 provides the best results. For $\check{K}$ = 17 and Ψ_o = 0.5, the BWOF algorithm reduces the average fulfillment times much as 39%.

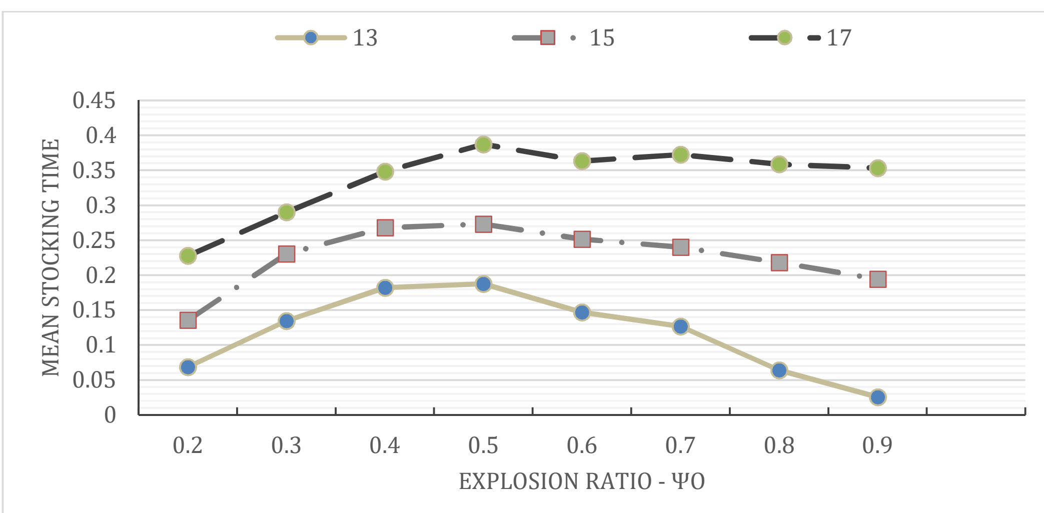


Figure 5.16 Percentage advantage of the BWOF algorithm.

5.8.3 Economic Advantages of the BWOF Algorithm

Simulation results show that the BWOF Algorithm provides significantly shorter fulfillment times with a lower explosion ratio. However, the advantage is not limited with picking operations. The table below shows that the stocking time does not change for different pick list sizes even though it is very sensitive to the changes in explosion ratio.

Table 5.7 The Comparison of Mean Stocking Time

	13		15		17	
	old	**seed**	**old**	**seed**	**old**	**seed**
0.1	72.282	73.11741	71.713	71.841	72.785	71.247
0.2	84.601	83.7918	84.887	85.503	83.933	83.395
0.3	94.766	89.64512	93.591	93.045	94.711	92.840
0.4	98.409	97.68573	98.643	96.658	99.631	98.757
0.5	107.457	104.1496	106.311	104.579	106.245	103.560
0.6	117.361	113.2345	117.367	109.242	118.549	112.710
0.7	150.089	148.3608	147.958	140.809	146.294	143.888
0.8	228.195	220.2046	221.460	209.395	226.065	216.091
0.9	250.532	232.6609	249.137	234.842	249.029	237.555

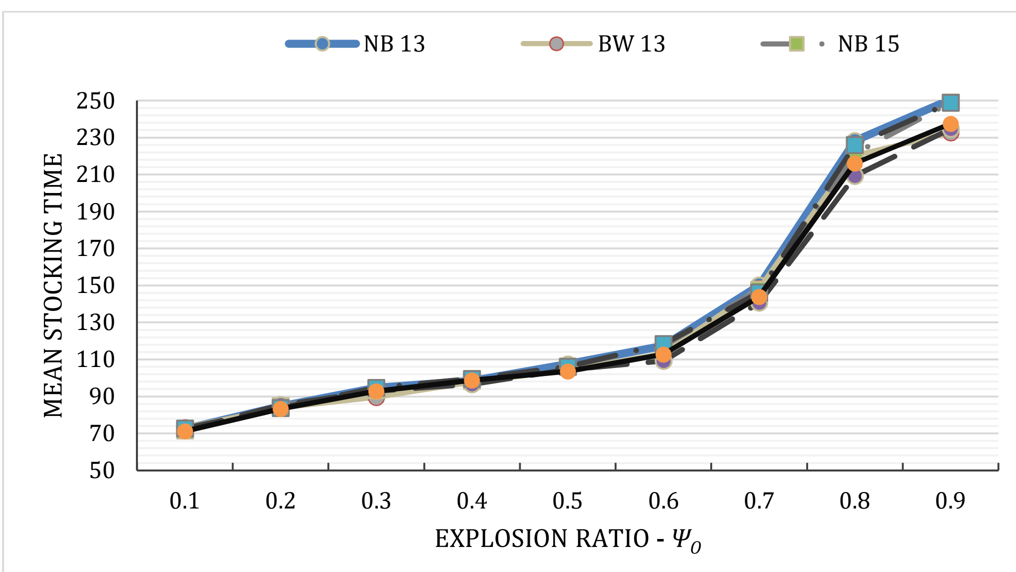


Figure 5.17 The comparison of mean stocking time.

Using the NBOP algorithm where Ǩ = 15, and $\Psi_o = 0.8$, average stocking time of an item is 221.46 minutes. Whereas using the BWOF algorithm the average stocking time drops d to 104.5 minutes. This means the average delay for each received item is about 2 hours.

For each incoming bulk r, let $A_{r,t}$ be the arrival time and $\hat{A}_{r,t}$ be the storage time of the associated SKU, $C_{j,t}$ the order receipt time and $\hat{C}_{j,t}$ the order fulfillment time, $\dot{\alpha}$ be the marginal stocking cost per minute and $\dot{\beta}$ be the marginal picking cost per minute. Total cost of fulfilling an order omitting all other costs is:

$$Total\ Cost\ = \dot{\alpha}_{\Psi_o} \sum_t \sum_{r \in R_t} \frac{(\hat{A}_{r,t} - A_{r,t})}{TR_t} - \dot{\beta}_{\Psi_o} \sum_t \sum_{j \in J_t} \frac{(\hat{C}_{j,t} - C_{j,t})}{TJ_t} \qquad (5.14)$$

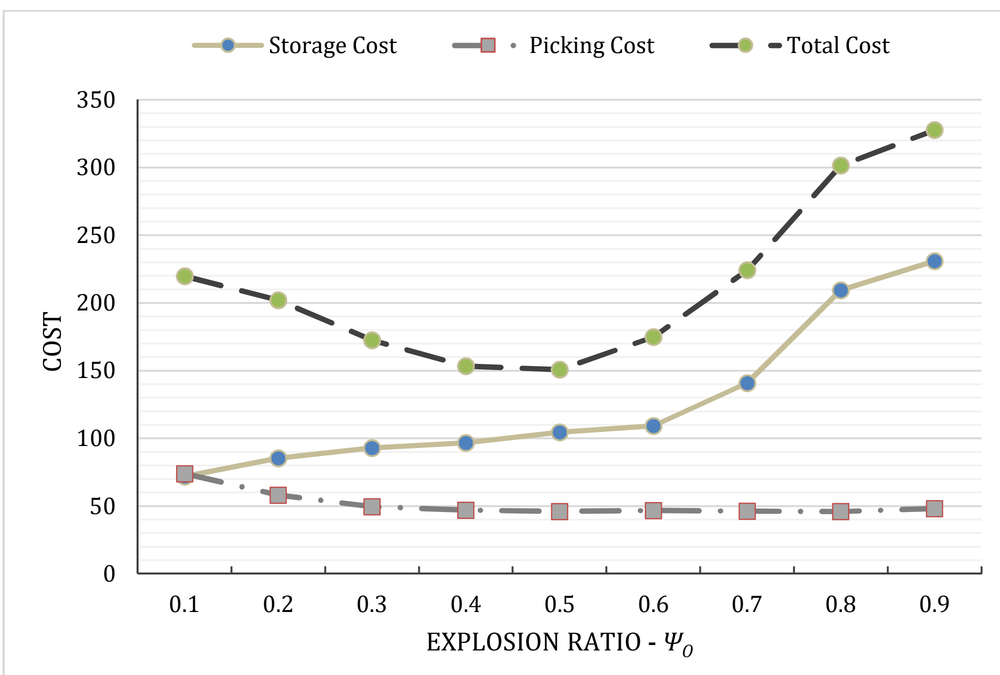


Figure 5.18 Total cost advantage of BWOF.

CHAPTER 6

CURRENT ACCOMPLISHMENTS AND FUTURE WORK

6.1 Current Accomplishments and Significant Findings

The research conducted in the production of this dissertation accomplishes the following significant research objectives:

1. An observational study is carried out at multiple locations of a leading online retail company. IFW operations are identified, and model considerations are formed driven by analytical insights synthesized from the observational study. The specific research objectives are to identify (i) key IFW differentiators and order flows (ii) the decision variables and parameters and (iii) the performance objective.

2. The differentiators are used to identify the decision variables and parameters that characterize IFWs and their performance indicators in order to develop and formulate descriptive and prescriptive analytical models for the control of IFW operations. The model includes a stocking algorithm that is unique to the explosive storage policy and a baseline picking algorithm which serves the immediate fulfillment objective.

3. Using baseline storage and picking algorithms the order fulfillment time superiority of explosive storage is investigated. Model size of the simulation is determined by (i) number of unique SKUs stored (ii) number of storage bin locations and (iii) number of daily customer orders. The performance objective of these algorithms is to reduce a linear fulfillment time objective.

4. A data driven simulation model was built in MS-Access/VBA platform in order to analyze the performance behavior of linear fulfillment time. We have simulated the IFW behavior to evaluate the response to combinations of two controllable variables (i) explosion ratio of incoming bulk, (ii) maximum number of stops that a picker should make.

5. A mixed integer programming problem (MIP) is created that minimizes fulfillment time for every order by finding the best cluster of items and generating a pick list which is used to develop an advanced picking algorithm.

6. The MIP problem is NP-hard, and the integer decision space is way too large to efficiently find an optimal solution. The list of candidate picks is very large, and the solution space is described by the product of the number of pending orders and the active inventory locations. Because of this difficulty, we approach the problem by heuristics to reduce the problem size. Firstly, the problem is narrowed down without any loss of data by generating lists with (i) order elimination meaning only some customer orders, (ii) bin selection which is picking the orders from only some selected bins.

7. A combination of two heuristic algorithms are developed to select a seed to shorten the solution time while minimizing the fulfillment time which we call the Bin Weighed Order Fillability (BWOF)Algorithm.

8. The BWOF is first tested using the OpenSolver in order to confirm improvement in fulfillment and solution time of the problem. It is then, replaced with the baseline picking algorithm in the simulation and the results are compared.

6.2 Future Work

There are many empirical studies about e-retailing practices and effects. However there is little research that quantitatively investigates fulfillment operations in IFWs. Realizing the gap in the existing literature about warehousing methods for the next generation of online retailing, the intent is to investigate the issue further and open doors to a large area for future researchers who could solve many problems which may or may not have been identified by yet. This research conducted in the production of this dissertation has laid the basis for the future research opportunities.

The storage and picking algorithms developed for this research use simple linear fulfillment time for analysis. To further reduce fulfilment times, quadratic metrics may be used. The advantage of using quadratic fulfillment time kicks in when a penalty is set on late fulfillment and the objective function is altered to minimize the penalty. In this case, the developed algorithms will give a higher priority to earliest received orders instead of selecting a list of orders made in a time window.

In Chapter 4, the simulation analysis that uses the Narrow Band Order Picking Algorithm, shows that Ǩ=13, 15, and 17 provide the best results, respectively. Therefore, in Chapter 5 only these three are tested for comparison, and it is concluded that using BWOF Algorithm, larger lists perform better. In order to explore the benefits of BWOF and gather a prescriptive analysis, larger lists than Ǩ=17 should also be tested.

The calculation capacity of the computer used for this research, restricted the simulation to only 22,000 orders of 400 SKU's in a time frame of 9 days. In order to mimic IFWs better, more powerful computers should be used to create larger simulation models.

Improved picking algorithms can be explored. Simple examples may be pre-processing customer orders and the grouping of orders that are stored together before the pick list generation process occurs in order to reduce the calculation load; or, manipulating the search bandwidth, pick list size might be adjusted deciding between the opportunity cost of adding and item to the list or sending picked items to packing.

Baseline stocking algorithm may be replaced with a more advanced model. Instead of a fully randomized model. An algorithm can be developed that controls the storage of items that are sold together or items of same type or family.

An economic analysis may be conducted to explore the benefits or disadvantages of explosive storage and the BWOF policies. These policies require larger warehouse space, more labor and more precision than traditional warehousing policies, however customer satisfaction and convenience of shopping online bring in more business. The relationship may be investigated to gain more insight.

REFERENCES

Acimovic, J. and S. C. Graves (2014). "Making better fulfillment decisions on the fly in an online retail environment." Manufacturing & Service Operations Management 17(1): 34-51.

Ansoff, H. I. (1957). "Strategies for diversification." Harvard Business Review 35(5): 113-124.

Ashayeri, J. and L. Gelders (1985). "Warehouse design optimization." European Journal of Operational Research 21(3): 285-294.

Ashayeri, J., et al. (2002). "A geometrical approach to computing expected cycle times for zonebased storage layouts in AS/RS." International Journal of Production Research 40(17): 4467-4483.

Azadivar, F. (1984). A simulation optimization approach to optimum storage and retrieval policies in an automated warehousing system. Proceedings of the 16th conference on Winter simulation, IEEE Press.

Azadivar, F. (1989). "Optimum allocation of resources between the random access and rack storage spaces in an automated warehousing system." International Journal of Production Research 27(1): 119-131.

Azevedo, S. G., et al. (2012). "Influence of green and lean upstream supply chain management practices on business sustainability." Engineering Management, IEEE Transactions on 59(4): 753-765.

Bailey, J. P. (1998). "Intermediation and electronic markets: Aggregation and pricing in Internet commerce." Doctoral dissertation, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science. Retrieved May 1, 2017, from http://hdl.handle.net/1721.1/9835.

Bartholdi, J. J. and S. T. Hackman (2008). "Allocating space in a forward pick area of a distribution center for small parts" IIE Transactions 40(11): 1046–1053.

Bartholdi, J. J. and S. T. Hackman (2014). "Warehouse & distribution science." Doctoral Dissertation. Supply Chain & Logistics Institute School of Industrial and Systems Engineering, Georgia Institute of Technology.

Bassan, Y., et al. (1980). "Internal layout design of a warehouse." IIE Transactions 12(4): 317-322.

Battista, C., et al. (2014). "Multiproduct slot allocation heuristic to minimize storage space." International Journal of Retail & Distribution Management 42(3): 172-186.

Bell, D. R., et al. (2013). "Inventory showrooms and customer migration in omni-channel retail: The effect of product information." Available at SSRN 2370535.

Bellman, R. (1956). On a routing problem, Defense Technical Information Center Document. Retrieved May 1, 2017, from http://www.dtic.mil/get-tr-doc/pdf?AD=AD0606258.

Berry, J. R. (1968). "Elements of warehouse layout." The International Journal of Production Research 7(2): 105-121.

Bozer, Y. A. (1985). "Optimizing throughput performance in designing order picking systems." Doctoral dissertation, Georgia Institute of Technology. Georgia, USA. Retrieved May 1, 2017, from http://hdl.handle.net/1853/25588.

Bozer, Y. A. and M. Cho (2005). "Throughput performance of automated storage/retrieval systems under stochastic demand." IIE Transactions 37(4): 367-378.

Bozer, Y. A. and J. A. White (1984). "Travel-time models for automated storage/retrieval systems." IIE Transactions 16(4): 329-338.

Bretzke, W.-R. (2000). "Electronic commerce als herausforderung an die logistik." Logistik Management 2(1): 8-15.

Brynjolfsson, E., et al. (2013). "Competing in the age of omnichannel retailing." MIT Sloan Management Review 54(4): 23.

Cahn, A. (1948). "The warehouse problem." Bulletin of the American Mathematical Society, AMER Mathematical Society, Vol. 54, No. 11.

Cakmak, E., et al. (2012). "Determining the size and design of flow type and u-type warehouses." Procedia-Social and Behavioral Sciences 58: 1425-1433.

Caron, F., et al. (2000). "Layout design in manual picking systems: a simulation approach." Integrated Manufacturing Systems 11(2): 94-104.

Casati, F., et al. (2001). E-Business applications for supply chain management: Challenges and solutions. Data Engineering. 17th International Institute of Electrical and Electronics Engineers Conference Proceedings pp. 71-78.

Chakrabarti, R. and B. Scholnick (2002). "Exchange rate expectations and foreign direct investment flows." Weltwirtschaftliches Archiv 138(1): 1-21.

Chen, F., et al. (2013). "An ant colony optimization routing algorithm for two order pickers with congestion consideration." Computers & Industrial Engineering 66(1): 77-85.

Chen, F., et al. (2014). "An ACO-based online routing method for multiple order pickers with congestion consideration in warehouse." Journal of Intelligent Manufacturing: 1-20.

Chen, J. C., et al. (2013). "Warehouse management with lean and RFID application: a case study." The International Journal of Advanced Manufacturing Technology 69(1-4): 531-542.

Chen, M.-C. and H.-P. Wu (2005). "An association-based clustering approach to order batching considering customer demand patterns." Omega 33(4): 333-343.

Chen, S. and F. Leteney (2000). "Get real! Managing the next stage of internet retail." European Management Journal 18(5): 519-528.

Chew, E. P. and L. C. Tang (1999). "Travel time analysis for general item location assignment in a rectangular warehouse." European Journal of Operational Research 112(3): 582-597.

Chia Jane, C. (2000). "Storage location assignment in a distribution center." International Journal of Physical Distribution & Logistics Management 30(1): 55-71.

Chow, W.-M. (1986). "An analysis of automated storage and retrieval systems in manufacturing assembly lines." IIE Transactions 18(2): 204-214.

Compagno, L., et al. (2012). Designing an optimal shape warehouse. Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services, Springer: 248-255.

Cormier, G. and P. Eng (1997). "A brief survey of operations research models for warehouse design and operation." CORS-SCRO Canadian Operational Research Society Bulletin Vol 8.

Cormier, G. and E. A. Gunn (1992). "A review of warehouse models." European Journal of Operational Research 58(1): 3-13.

Coughlan, A., Anderson, E., Stern, L., El-Ansary, A., (2006). Marketing Channels, 7th ed.. Prentice-Hall, New Jersey.

Cox, B. (1986). "Determining economic levels of automation by using a hierarchy of productivity ratios techniques." 7th International Conference on Automation in Warehousing Proceedings pp. 39-49.

Coyle, J. J., et al. (1996). "The management of business logistics." St Paul, MN: West Publishing Company.

Croom, S., et al. (2000). "Supply chain management: an analytical framework for critical literature review." European Journal of Purchasing & Supply Management 6(1): 67-83.

Dallari, F., et al. (2009). "Design of order picking system." The International Journal of Advanced Manufacturing Technology 42(1-2): 1-12.

Dayal, Umeshwar, Meichun Hsu, and Rivka Ladin. (2001). "business process coordination: State of the art, Trends, and Open Issues." Very Large Data Base. Vol. 1.

De Koster, R. and T. Le-Duc (2004). "Single-command travel time estimation and optimal rack design for a 3-dimensional compact AS/RS." R. Meller, MK Ogle, BA Peters, GD taylor, J. Usher (eds.), Progress in Material Handling Research 2005: 49-66.

De Koster, R., et al. (2007). "Design and control of warehouse order picking: A literature review." European Journal of Operational Research 182(2): 481-501.

De Koster, René, Kees Jan Roodbergen, and Ronald Van Voorden. "Reduction of walking time in the distribution center of De Bijenkorf." New Trends in Distribution Logistics. Berlin Heidelberg: Springer 215-234.

De Koster, R. and E. Van der Poort (1998). "Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions." IIE Transactions 30(5): 469-480.

De Koster, R. B., et al. (2012). "Determining the number of zones in a pick-and-sort order picking system." International Journal of Production Research 50(3): 757-771.

Dukic, Doran, and Dadomir Oluio. (2004) "Order-picking routing policies: Simple heuristics, advanced heuristics or optimal algorithm." Strojniski Vestnik 50.11 2.

Dukic, G. and C. Oluic (2007). "Order-picking methods: improving order-picking efficiency." International Journal of Logistics Systems and Management 3(4): 451-460.

Ekren, B., et al. (2015). "Warehouse design under class-based storage policy of shuttle-based storage and retrieval system." International Federation of Accountants, PapersOnLine 48(3): 1152-1154.

Elizandro, D. and H. Taha (2007). "Simulation of industrial systems: Discrete event simulation using Excel/VBA." New York, NY: Auerbach Publications CRC Press.

Elsayed, E. and A. Chen (1993). "Optimal levels of process parameters for products with multiple characteristics." The International Journal of Production Research 31(5): 1117-1132.

Elsayed, E. and M.-K. Lee (1996). "Order processing in automated storage/retrieval systems with due dates." IIE Transactions 28(7): 567-578.

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

Elsayed, E. A. and R. G. Stern (1983). "Computerized algorithms for order processing in automated warehousing systems." The International Journal of Production Research 21(4): 579-586.

Elsayed, E. A. and O. Unal (1989). "Order batching algorithms and travel-time estimation for automated storage/retrieval systems." The International Journal of Production Research 27(7): 1097-1114.

Evans, J. R. (1984). "The factored transportation problem." Management Science 30(8): 1021-1024.

Eynan, A. and M. J. Rosenblatt (1994). "Establishing zones in single-command class-based rectangular AS/RS." IIE Transactions 26(1): 38-46.

Foley, R. and E. Frazelle (1990). "Throughput expressions and bounds for end-of-aisle order picking with activity based storage." HE Transactions.

Francis, R. L. (1967). "On some problems of rectangular warehouse design and layout." Journal of Industrial Engineering 18(10): 595-&.

Frazelle, E. (2002). "Supply chain strategy: the logistics of supply chain management." CA McGraw Hill.

Frazelle, E. H., Hackman, S. T., Passy, U., & Platzman, L. K. (1994). "The forward-reserve problem" pp. 43-61. John Wiley & Sons, Inc.

Gademann, N. and S. Velde (2005). "Order batching to minimize total travel time in a parallel-aisle warehouse." IIE Transactions 37(1): 63-75.

Gagliardi, J.-P., et al. (2012). "On storage assignment policies for unit-load automated storage and retrieval systems." International Journal of Production Research 50(3): 879-892.

Geraldes, C. A., et al. (2012). Warehouse design and planning: a mathematical programming approach. Computational Science and Its Applications–ICCSA 2012, Springer: 187-201.

Glock, C. H., et al. (2014). "The lot sizing problem: A tertiary study." International Journal of Production Economics 155: 39-51.

Goetschalckx, M. and J. Ashayeri (1989). "Classification and design of order picking." Logistics World 2(2): 99-106.

Goetschalckx, M. and J. Ashayeri (1989). "Classification and design of order picking." Logistics World 2(2): 99-106.

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

Goetschalckx, M. and H. D. Ratliff (1990). "Shared storage policies based on the duration stay of unit loads." Management Science 36(9): 1120-1132.

Goetschalckx, M. and R.-P. Wei (1994). "Bibliography on order picking systems." Material Handling Research Center, Georgia Institute of Technology.

Goolsbee, A. and J. Chevalier (2002). "Measuring prices and price competition online: Amazon and Barnes and Noble." Yale ICF Working Paper No. 02-23.

Graves, S. C., et al. (1977). "Storage-retrieval interleaving in automatic warehousing systems." Management Science 23(9): 935-945.

Graves, S. C. and L. B. Schwarz (1977). "Single cycle continuous review policies for arborescent production/inventory systems." Management Science 23(5): 529-540.

Group, B. C. (1999). "The race for online riches–e-retailing in Europe." Silver Spring, Maryland.

Gu, J., et al. (2007). "Research on warehouse operation: A comprehensive review." European Journal of Operational Research 177(1): 1-21.

Gu, J., et al. (2010). "Research on warehouse design and performance evaluation: A comprehensive review." European Journal of Operational Research 203(3): 539-549.

Guo, X., et al. (2016). "Impact of required storage space on storage policy performance in a unit-load warehouse." International Journal of Production Research 54(8): 2405-2418.

Hackman, S. T., et al. (1990). "Allocating items to an automated storage and retrieval system." IIE Transactions 22(1): 7-14.

Hale, T. S. and C. R. Moberg (2003). "Location science research: a review." Annals of Operations Research 123(1-4): 21-35.

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

Han, M.-H. and L. F. McGinnis (1986). Automated Work-in Process Carousels: Modeling and Analysis, Material Handling Research Center, Georgia Institute of Technology.

Han, M.-H. and L. F. McGinnis (1989). "Control of material handling transporter in automated manufacturing." IIE Transactions 21(2): 184-190.

Han, M.-H., et al. (1987). "On sequencing retrievals in an automated storage/retrieval system." IIE Transactions 19(1): 56-66.

Harrell, Charles, 1950-. (2004). "Simulation using ProModel." Fairfield, PA:McGraw-Hill.

Hausman, W. H., et al. (1976). "Optimal storage assignment in automatic warehousing systems." Management Science 22(6): 629-638.

Hodgson, T. J. and T. J. Lowe (1982). "Production lot sizing with material-handling cost considerations." IIE Transactions 14(1): 44-51.

Hoover, S. V. and R. F. Perry (1984). "Validation of simulation models: The weak/missing link". 16th conference on Winter simulation Proceedings pp. 292-295, IEEE Press.

Hung, M. S. and J. C. Fisk (1984). "Economic sizing of warehouses: A linear programming approach." Computers & Operations Research 11(1): 13-18.

Hwang, H., et al. (1988). "Clustering algorithms for order picking in an automated storage and retrieval system." The International Journal of Production Research 26(2): 189-201.

Hwang, H. and S. B. Lee (1990). "Travel-time models considering the operating characteristics of the storage and retrieval machine." The International Journal of Production Research 28(10): 1779-1789.

Hwang, H. and J. Y. Song (1993). "Sequencing picking operations and travel time models for man-on-board storage and retrieval warehousing system." International Journal of Production Economics 29(1): 75-88.

Jansen, B. J., Spink, A., and Saracevic T (2000). “Real life, real users, and real needs: A study and analysis of user queries on the web.” Information Processing & Management 36(2), 207-227.

Jarvis, J. M. and E. D. McDowell (1991). "Optimal product layout in an order picking warehouse." IIE Transactions 23(1): 93-102.

Jawadekar, Waman S. (2011), "8. Knowledge management: Tools and technology", Knowledge Management: Text & Cases, New Delhi: Tata McGraw-Hill Education Private Ltd, p. 278

Öztayşi, Başar, and Özge Sürer. "Supply chain performance measurement using a SCOR based fuzzy VIKOR approach." Supply chain management under fuzziness. Berlin, Heidelberg: Springer, 2014. 199-224.

Karasawa, Y., et al. (1980). "Trade-off analysis for optimal design of automated warehouses." International Journal of Systems Science 11(5): 567-576.

Reinelt, Gerhard, (1994). "The traveling salesman: computational solutions for TSP applications." Berlin, Heidelberg: Springer.

Kouvelis, P. and V. Papanicolaou (1995). "Expected travel time and optimal boundary formulas for a two-class-based automated storage/retrieval system." International Journal of Production Research 33(10): 2889-2905.

Kulturel, S., et al. (1999). "Experimental investigation of shared storage assignment policies in automated storage/retrieval systems." IIE Transactions 31(8): 739-749.

Kunder, R. and T. Gudehus (1975). "Average travel times for one-dimensional order picking." Zeitschrift für Operations Research 19: B53-B72.

Lang, et al. Harrison. (2012). "Amazon.com: Offering Everything from A to Z." Case Study, University of Richmond: Robins School of Business.

Larson, T. N., et al. (1997). "A heuristic approach to warehouse layout with class-based storage." IIE Transactions 29(4): 337-348.

Laudon, Kenneth C., and Carol Guercio Traver (2007). "E-commerce." Pearson/Addison Wesley. Vol. 29.

Le-Anh, T. and M. De Koster (2006). "A review of design and control of automated guided vehicle systems." European Journal of Operational Research 171(1): 1-23.

Le-Duc, T. (2005). Design and control of efficient order picking processes, Erasmus Research Institute of Management (ERIM).

Le-Duc, T. and R. De Koster (2005). "Layout optimization for class-based storage strategy warehouses." Supply Chain Management-European Perspective: 191-214.

Lee, M.-K. and H. Hwang (1988). "An approach in the design of a unit-load automated carousel storage system." Engineering Optimization 13(3): 197-210.

Lerher, T., et al. (2013). "Multi-objective optimization of automated storage and retrieval systems." Annals of the Faculty of Engineering Hunedoara 11(1): 187.

Levy, J. (1974). "The optimal size of a storage facility." Naval Research Logistics Quarterly 21(2): 319-326.

Levy, S. E., et al. (2013). "An analysis of one-star online reviews and responses in the Washington, DC, lodging market." Cornell Hospitality Quarterly 54(1): 49-63.

Li, J., et al. (2017). "Joint optimisation of order batching and picker routing in the online retailer’s warehouse in China." International Journal of Production Research 55(2): 447-461.

Linden, G., et al. (2003). "Amazon. com recommendations: Item-to-item collaborative filtering." Internet Computing, IEEE 7(1): 76-80.

Linn, R. J. and R. Wysk (1984). A simulation model for evaluating control algorithms of an automated storage/retrieval system. Proceedings of the 16th conference on Winter simulation, IEEE Press.

Loomis, J. and L. Caughlan (2006). "The importance of adjusting for trip purpose in regional economic analyses of tourist destinations." Tourism Economics 12(1): 33-43.

Manzini, R. (2012). "Warehousing in the Global Supply Chain" (Chapter 2, page 32), London, England: Springer-Verlag.

Malmborg, C. (1998). "Analysis of storage assignment policies in less than unit load warehousing systems." International Journal of Production Research 36(12): 3459-3475.

Malmborg, C. J. (1996). "An integrated storage system evaluation model." Applied Mathematical Modelling 20(5): 359-370.

Malmborg, C. J. (2001). "Rule of thumb heuristics for configuring storage racks in automated storage and retrieval systems design." International Journal of Production Research 39(3): 511-527.

Malmborg, C. J. and K. Al-Tassan (2000). "An integrated performance model for orderpicking systems with randomized storage." Applied Mathematical Modelling 24(2): 95-111.

Marchet, G., et al. (2013). "Development of a framework for the design of autonomous vehicle storage and retrieval systems." International Journal of Production Research 51(14): 4365-4387.

Marsh, W. (1979). "Elements of block storage design." International Journal of Production Research 17(4): 377-394.

Marsh, W. H. (1978). Space utilization considerations in block storage design.

Marsh, W. H. (1983). "A comparison with Berry." International Journal of Production Research 21(2): 163-172.

Matson, J. O. and J. A. White (1981). Storage system optimization, Defense Technical Information Center Document.

Mellema, P. M. and C. A. Smith (1988). Simulation analysis of narrow-aisle order selection systems. 20th conference of Association for Computing Machinery Proceedings.

Meller, R. D. and K.-Y. Gau (1996). "The facility layout problem: recent and emerging trends and perspectives." Journal of manufacturing systems 15(5): 351-366.

Moder, J. J. and H. Thornton (1965). "Quantitative analysis of factors affecting floorspace utilization of palletized storage." Journal of Industrial Engineering 16(1): 8-&.

Moeller, K. (2011). "Increasing warehouse order picking performance by sequence optimization." Procedia-Social and Behavioral Sciences 20: 177-185.

Mohsen and M. Hassan (2002). "A framework for the design of warehouse layout." Facilities 20(13/14): 432-440.

Nynke, F., et al. (2002). "Linking warehouse complexity to warehouse planning and control structure." International Journal of Physical Distribution and Logistics Management 32(5): 381-395.

Oztürkoglu, O. (2011). "New warehouse designs: angled aisles and their effects on travel distance." Doctoral Dissertation, University of Arkansas.

Pan, J. C.-H., et al. (2012). "Storage assignment problem with travel distance and blocking considerations for a picker-to-part order picking system." Computers & Industrial Engineering 62(2): 527-535.

Pandit, R. and U. Palekar (1993). "Response time considerations for optimal warehouse layout design." Journal of engineering for industry 115(3): 322-328.

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

Pang, King-Wah, and Hau-Ling Chan. (2016) "Data mining-based algorithm for storage location assignment in a randomized warehouse." International Journal of Production Research 1-18.

Pecchiar, G. (2012). "Upgrade to WMS, keep QuickBooks and MYOB." MHD Supply Chain Solutions, 42(1), 24.

Perry, L. Glenn. "The SEC's enforcement activities." The CPA Journal (pre-1986) 54.000004 (1984): 9.

Pessotto, A. (2009). SCM: Supply Chain Management, Università degli studi di Udine, Dipartimento di Ingegneria Elettrica Gestionale e Meccanica.

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

Petersen, C. G. (2000). "An evaluation of order picking policies for mail order companies." Production and Operations Management 9(4): 319-335.

Petersen, C. G. (2002). "Considerations in order picking zone configuration." International Journal of Operations & Production Management 22(7): 793-805.

Petersen, C. G. and G. Aase (2004). "A comparison of picking, storage, and routing policies in manual order picking." International Journal of Production Economics 92(1): 11-19.

Petersen, C. G., et al. (2004). "Improving order-picking performance through the implementation of class-based storage." International Journal of Physical Distribution & Logistics Management 34(7): 534-544.

Petersen II, C. G. and R. W. Schmenner (1999). "An evaluation of routing and volume-based storage policies in an order picking operation." Decision Sciences 30(2): 481.

Pliskin, J. S. and D. Dori (1982). "Ranking alternative warehouse area assignments: A multiattribute approach." IIE Transactions 14(1): 19-26.

Pohl, L. M., et al. (2011). "Turnover-based storage in non-traditional unit-load warehouse designs." IIE Transactions 43(10): 703-720.

Potts, C. N. and M. Y. Kovalyov (2000). "Scheduling with batching: a review." European Journal of Operational Research 120(2): 228-249.

Rao, S. S. and G. K. Adil (2017). "Analytical models for a new turnover-based hybrid storage policy in unit-load warehouses." International Journal of Production Research 55(2): 327-346.

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

Rigby, D. (2011). "The future of shopping." Harvard Business Review 89(12): 65-76.

Roberts, S. D. and R. Reed Jr (1972). "Optimal warehouse bay configurations." AIIE Transactions 4(3): 178-185.

Roll, Y. and M. J. Rosenblatt (1983). "Random versus grouped storage policies and their effect on warehouse capacity." Material Flow 1(3): 199-205.

Roodbergen, K. J. and R. De Koster (2001). "Routing order pickers in a warehouse with a middle aisle." European Journal of Operational Research 133(1): 32-43.

Roodbergen, K. J. and R. Koster (2001). "Routing methods for warehouses with multiple cross aisles." International Journal of Production Research 39(9): 1865-1883.

Roodbergen, K. J., et al. (2008). "Designing the layout structure of manual order picking areas in warehouses." IIE Transactions 40(11): 1032-1045.

Rosen, K. T. and A. L. Howard (2000). "E-retail: Gold rush or fool's gold?" California management review 42(3): 72-100.

Rosenblatt, M. J. and A. Eynan (1989). "Note—Deriving the Optimal Boundaries for Class-Based Automatic Storage/Retrieval Systems." Management Science 35(12): 1519-1524.

Rosenblatt, M. J. and Y. Roll (1984). "Warehouse design with storage policy considerations." International Journal of Production Research 22(5): 809-821.

Rosenblatt, M. J. and Y. Roll (1985). "A Future Value Approach to Determining Project Duration." IIE Transactions 17(2): 164-167.

Ross, D. F. (2015). Statistical Inventory Management. Distribution Planning and Control, Springer: 357-405.

Rouwenhorst, B., et al. (2000). "Warehouse design and control: Framework and literature review." European Journal of Operational Research 122(3): 515-533.

Rowley, J. (2000). "The principles of warehouse design." The Institute of Logistics & Transport, Corby.

Roy, D., et al. (2012). "Performance analysis and design trade-offs in warehouses with autonomous vehicle technology." IIE Transactions 44(12): 1045-1060.

Schwarz, L. B., et al. (1978). "Scheduling policies for automatic warehousing systems: simulation results." AIIE Transactions 10(3): 260-270.

Sharp, Gunter P., Kyung Il-Choe, and Chang S. Yoon. (1991). "Small parts order picking: analysis framework and selected results." Material Handling'90. Springer Berlin Heidelberg. 317-341.

Sooksaksun, N., et al. (2012). "A class-based storage warehouse design using a particle swarm optimisation algorithm." International Journal of Operational Research 13(2): 219-237.

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

Staudt, F. H., et al. (2015). "Warehouse performance measurement: a literature review." International Journal of Production Research 53(18): 5524-5544.

Tang, L. C. and E.-P. Chew (1997). "Order picking systems: batching and storage assignment strategies." Computers & Industrial Engineering 33(3): 817-820.

Thonemann, U. W. and M. L. Brandeau (1998). "Note. Optimal storage assignment policies for automated storage and retrieval systems with stochastic demands." Management Science 44(1): 142-148.

Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2010). "Facilities planning. " Hoboken, NJ: John Wiley & Sons.

Torabi, S., et al. (2015). "Fulfillment source allocation, inventory transshipment, and customer order transfer in e-tailing." Transportation Research Part E: Logistics and Transportation Review 79: 128-144.

Van den Berg, J. P. and W. Zijm (1999). "Models for warehouse management: Classification and examples." International Journal of Production Economics 59(1): 519-528.

Verhoef, et al. (2015). "From multi-channel retailing to omni-channel retailing: introduction to the special issue on multi-channel retailing." Journal of retailing 91.2: 174-181.

Wang, X., et al. (2013). "A Real-Time Synchronized Zone Order Picking System for Balancing Pickers' Workload." ICIC express letters. Part B, Applications: an international journal of research and surveys 4(5): 1495-1502.

Wäscher, Gerhard (2004). "Order picking: a survey of planning problems and methods." Supply chain management and reverse logistics. Springer Berlin Heidelberg, 323-347.

Waugh, R. M. and R. A. Ankener (1977). Simulation of an automated stacker storage system. Proceedings of the 9th conference on Winter simulation-Volume 2, Winter Simulation Conference.

White, J. A. and R. L. Francis (1971). "Normative models for some warehouse sizing problems." AIIE Transactions 3(3): 185-190.

Won, J. and S. Olafsson* (2005). "Joint order batching and order picking in warehouse operations." International Journal of Production Research 43(7): 1427-1442.

Yan, W., et al. (2014). "The relationship between recent stressful life events, personality traits, perceived family functioning and internet addiction among college students." Stress and Health 30(1): 3-11.

Yu, M. and R. B. De Koster (2009). "The impact of order batching and picking area zoning on order picking system performance." European Journal of Operational Research 198(2): 480-490.

Zollinger, H. A., (1996) "Expanded methodology to concept horizontal transportation problem solutions. " In R. J. Graves, L. F. McGinnis, D. J. Medeiros, R. E. Ward and M. R. Wilhelm (eds) Progress in Material Handling Research: 1996 (Charlotte, NC: Material Handling Institute), pp. 651–663.