UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Sequential zone picking systems and the integration of frequent item pairs in storage assignment

Trang H. Trinh

M.Sc. Thesis
August 2021

UT Supervisor:
Prof. Dr. Johannes Schmidt-Hieber

Kramp Supervisor:
Oliver Meisch
Nikki Nijenhuis

Graduation Committee:
Prof. Dr. Johannes Schmidt-Hieber (UT)
Dr. Jan-Kees van Ommeren (UT)
Nikki Nijenhuis (Kramp)

Statistics (STAT) Research Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

PREFACE

With this thesis, my time as a Master's student at the University of Twente has come to an end. The pages that lie in front of you are the result of a research project at Kramp Group BV that lasted for almost eight months. This project reminded me of the beauty of Data Science, in which I was able to dive deep into a new domain of knowledge about order picking systems, despite a rough start. My primary company supervisor went on his parental leave earlier than expected, and without him, the initial discussion with the stakeholders posed different expectations from the logistics team than what we had in mind. I have struggled to bridge the gap between our ultimate goal to deliver something that has sufficient academic content and the business demands. I was lost in the world of data in which the data entries are only kept for a week from its logging date, or the same attribute can have different names across different tables. But all the moments of hardship were made worthwhile when the simulation started running properly, the drawn results were meaningful to the business, or simply when I was able to fully explain a theorem in literature. Eight months were not enough to implement on all of the ideas I had in mind for the topic.

In truth, I could not have made it without a strong support group. First of all, I would like to thank Oliver Meisch, Nikki Nijenhuis, Daniel Schmelzer, and Tra Nguyen from Kramp for their direct support and guidance. They also helped me prioritize tasks within the project scope and proofread the report. Secondly, my gratitude goes to my academic supervisor, Johannes Schmidt-Hieber, for various valuable discussions although he is not familiar with the knowledge domain of order picking systems. Additionally, I am deeply grateful to my fiance, Jackson, whom has been there for me during the ups and downs of the project and offered to read the paper as many times as needed. Last but not least, I owe a big thanks to all my friends in Enschede, for the delicious dinners and fun conversations to keep me sane during the pandemic. Thank you all for your unwavering support and love.

ABSTRACT

Order picking, defined as the retrieval of products from their storage location in response to a specific customer order, is the most labor-intensive and time-critical operation in manual warehouse systems. The project aims to analyze theoretically and empirically sequential zone picking systems in which the picking area is divided into several picking zones or stations and connected by conveyor. The items of an order are distributed to one or more boxes, and the items in a box are retrieved progressively by visiting stations in a sequence. To this end, two queuing models with finite capacity are designed respectively for theoretical and empirical settings. In the empirical study, frequent item pairs (FIP) are derived from a year of historical transaction data at Kramp and integrated into the existing storage assignment of the company's warehouse in Strullendorf, Germany. A simulation is then built to study the impact of the FIP integration on the warehouse performance. Multiple key performance indicators are specified to quantify the impact. The simulation results show that FIP helps improve the warehouse performance; however, the effect may become negative when employing a greedy approach.

Contents

List of Figures

List of Tables

1 INTRODUCTION

Since the 1980s with the arrival of new management philosophies, such as Just-In-Time, partnership, and lean production, the warehouse has taken on a strategic role in attaining the logistics goals of tighter inventory control, shorter response time, and greater product variety (Coyle, Langley, and Bardi, 1996; Gu, Goetschalckx, and McGinnis, 2007). The warehouse operations have been constantly improved, especially in terms of order throughput time, to satisfy the customer's demand for faster (same-day or overnight) and better delivery services at cheaper prices. Among the operations, order picking has been regarded as the highest-priority area for responsiveness and productivity improvements in warehouse management (Petersen, 2000; de Koster, Le-Duc, and Roodbergen, 2007). Order picking, defined as the retrieval of products from their storage location in response to a specific customer order, is the most labour-intensive and time-critical operation in manual warehouse systems (Tompkins et al., 2010; Manzini, Bozer, and Heragu, 2015). The widespread adoption of the information and communication technology, such as barcoding, radio frequency communications, and warehouse management systems (WMS), has provided new opportunities to support the picking operations (Gu, Goetschalckx, and McGinnis, 2007).

In 2012, manual picker-to-parts order picking systems (OPS) form over 80% of all order picking systems in Western European warehouses in 2012 due to their easy installation and high reconfigurability at low initial and maintenance cost (de Koster, Le-Duc, and Roodbergen, 2007; de Koster, 2012). Despite the technology advancement and the increasing popularity of parts-to-picker OPS, a large number of recent research papers still focuses on the manual picker-to-parts. This is due to the fact that the manual picker-to-parts involves humans in almost all operations, there are many rooms for improvement. These systems employ low-level storage systems, i.e., products are stored in bins on shelves, storage drawers in cabinets, or cartons on flow racks (Coyle, Langley, and Bardi, 1996). The shelf height is limited by the reaching height of an average human being. Order pickers retrieve the requested items from storage racks or bins by traveling along the aisles. As previous studies claim that 50% of an order picker's time is spent on traveling, a myriad of research has concentrated on reducing travel time at tactical or operational levels (de Koster, Le-Duc, and Roodbergen, 2007; Tompkins et al., 2010).

Two of the main research threads that combine tactical and operational level are *zoning* and *storage assignment*.

Zoning is developed to group aisles into work zones or stations. It aims to reduce the traffic congestion and familiarize pickers with the item locations in the station which consequently minimizes the intra-station travel time. In order to eliminate the inter-station travel time of pickers, a conveyor system is installed to transfer the boxes between stations. One variant of zoning is sequential approach of progressive order assembly in which the order is assigned to a box, and the box visits all relevant stations sequentially. A picker at the first station retrieves items in their station then puts the box back to the conveyor to pass it to the next destination.

Storage assignment represents a strategy to distribute and locate items in the warehouse. This strategy is closely related to the layout of the warehouse, i.e., the design of its floors, aisles, modules, and shelf spaces, as well as the depot location. Intuitively, if the frequently-purchased-together products are located relatively close to each other, significant reductions in travel time and operational cost may be attained. As a result, cluster-based storage assignment (CBSA) with frequent item pairs has been studied extensively thanks to the unprecedented development of the computing power to mine enormous historical transaction data. Nevertheless, the literature on CBSA is very limited, especially when considering both zoning and warehouse layout.

The objective of this report is to analyze the *sequential zone picking systems* (SZPS) in both theoretical and empirical settings. To this end, two queuing models with finite capacity are designed to represent each setting. The theoretical setting is based on the closed queuing model with single segment by van der Gaast et al. (2020) while the empirical resembles Kramp's warehouse in Strullendorf, Germany. In the latter, frequent item pairs (FIP) are derived from a year of transaction data and integrated into the existing storage assignment. This report is the first to attempt to solve the item clustering problem by using the connected components in graph theory. Then a heuristic approach is employed to tackle both assign clusters to stations (zoning) and items to shelf spaces (warehouse layout). A simulation is built to study the impact of the FIP integration on the system performance.

The structure of the report is as follows. First, a general introduction to modeling order picking system as a network of queues and cluster-based storage assignment with frequent item pairs is given in Chapter 2. Chapter 3 elaborates on the product-form stationary distribution of the jump-over blocking approximation of the single-segment order picking systems with capacity constraint and block-and-recirculate protocol using the theoretical setting from van der Gaast et al. (2020). Chapter 4 describes the storage layout and dimensioning of the three-floor warehouse of Kramp Group, as well as storage assignment and picker information. The chapter also builds a simulation model with two scenarios and five main key performance indicators (KPIs) for the warehouse. Chapter 5 explains the data mining and clustering process for the frequent item pairs and develops an algorithm to incorporate the FIP cluster into the current storage assignment. Three scenarios are then simulated to reveal how FIP influences the KPIs via storage rearrangement. Chapter 6 concludes the results and suggests future work on the topic.

2 LITERATURE SURVEY

This chapter reviews the literature on order picking systems with sequential zoning and a conveyor to connect stations as a queuing network in Section 2.1. Section 2.2 discusses the previous cluster-based storage assignment using frequent item pairs. Although, the literature on both topics was limited despite their impact on the warehouse performance, it has received more attention from researchers, especially on cluster-based storage assignment (de Koster, Le-Duc, and Roodbergen, 2007). The limitations are addressed and the contribution of this report is explained.

2.1 Sequential Zone Picking Systems (SZPS) as a Network of Queues

SZPS with a conveyor poses difficulties for analysis (de Koster, 1994). In an exact analysis, all possible positions of transport boxes on the conveyor must be included in the state space. Since the systems can contain a considerable number of boxes, the state space becomes enormous and hence unsuitable for exact analysis. This justifies the need for a fast framework to evaluate various layout alternatives in designing sequential systems. Many researchers have attempted to approximate picking systems with an infinite buffer, i.e., without blocking and recirculation, based on a network of queues to analyze the system performance. Two common performance statistics in the literature are minimizing the average travel distance and minimizing the throughput time of an order (de Koster, Le-Duc, and Roodbergen, 2007).

de Koster (1994) pioneers this research thread. The author models the SZPS as a queuing network consisting of nodes that correspond to conveyor pieces and picking stations, all are preceded by uncapacitated queues. The network employs a Poisson arrival process, exponentially distributed service rate at all stations, and Markovian routing probability without looping. In most scenarios, the number of conveyor pieces equals the number of picking stations plus one as demonstrated in Figure 2.1. Each conveyor piece is assumed to have constant speed, hence approximated by a number of parallel exponential servers which is equal to its capacity in boxes. The performance of each node is computed by solving the traffic equations using the Gauss-Seidel iterative method.

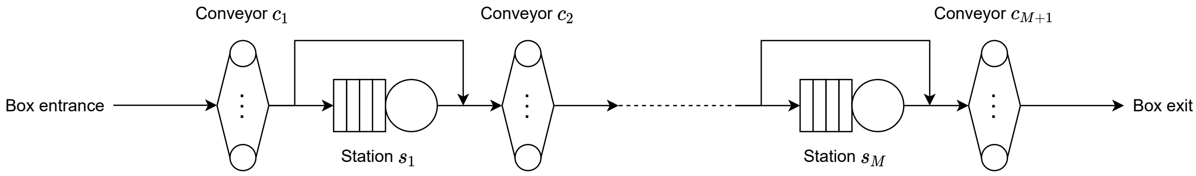


Figure 2.1: SZPS viewed as a sequence of picking stations connected by conveyor pieces (de Koster, 1994; Melacini, Perotti, and Tumino, 2011).

An additional difficulty besides the enormous state space when performing exact analysis of the SZPS lies in obtaining the exact distribution of service times at stations. This difficulty gives rise to the most important work in this research thread. Yu and de Koster (2008) extend the Jackson queuing network modeling of de Koster (1994) by allowing a general order arrival process and general service time distributions. The distribution of both the service time at a pick station and the inter-arrival time between two boxes is assumed to be characterized by its mean and Squared Coefficient of Variance (SCV). SCV is equal to the square of the division of the standard deviation by the mean. This extension helps represent real-life warehouses more accurately and provides a deeper understanding of the impact of different order picking policies on SZPS. The network is approximated by Whitt's analyzer for $G/G/c$ queuing network (Whitt, 1982). This approximation method yields acceptable results when analyzing the real-life SZPS at a part distribution center of an international motor production company.

Based on de Koster (1994), Melacini, Perotti, and Tumino (2011) show one of the first attempts to incorporate the system cost structure, while respecting the service level in terms of average order throughput time. The costs include annualised conveyor cost, annualised equipment cost of each picking station and the conveyor, annualised cost for a square metre of picking area, and annual salary of a picker. The model also adopts the single-block picking aisles for each station and decomposes the service time of a box at a station into three components: (i) the constant setup time per box, (ii) the travel time with return routing policy and constant walking speed, and (iii) the constant picking time per item regardless of the quantity. The return routing directs pickers to enter and leave the main aisles with items to be picked at the front aisle. This framework allows a quick estimate of the cost effectiveness of a given solution, which helps warehouse managers make better decisions on the layout design.

Because the pickers can temporarily place the boxes on the floor if the station buffer becomes saturated, it is reasonable for all three of the above-mentioned open queuing networks for SZPS to assume infinite station capacity. However, this is not usually the case as the boxes may arrive while all pickers are on a picking tour, resulting in an overloaded station buffer and a congested conveyor piece preceding the station. van der Gaast et al. (2020) models the SZPS as a closed queuing network with finite buffer and block-and-recirculate policy to more closely resemble real-world systems. The queuing network is visualized in Figure 2.2. A concern arises as the network is highly intractable due to the capacity constraints and blocking mechanism. The authors resolve this concern by constructing an approximation model with a jump-over blocking protocol developed by van Dijk (1988). Chapter 3 will further discuss the model setup and elaborate the product-form stationary distribution of the jump-over networks.

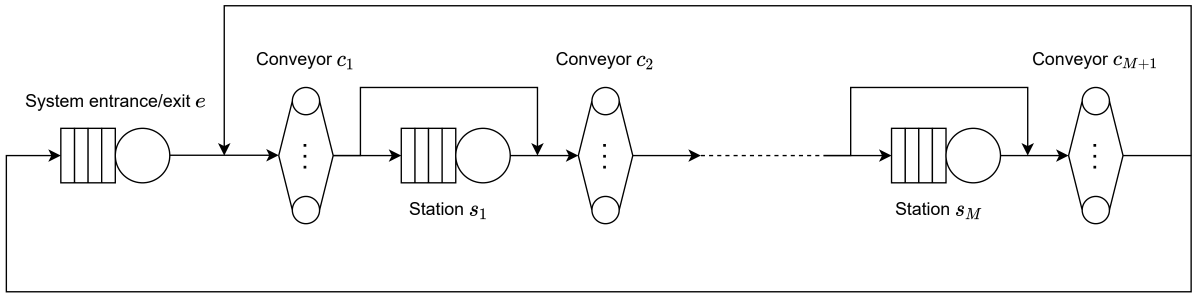


Figure 2.2: Closed queuing networks with block-and-recirculate protocol and single segment.

The closed block-and-recirculate networks and its jump-over approximation have several limitations. Firstly, due to the assumed exponential service time at a station, the picking time may be close to zero, which is unrealistic because a picker requires a minimum amount of time for initial tour setup and box post-processing. Moreover, the picking time takes into account only the amount of items to be picked, while neglecting the station layout and the storage assignment. Secondly, the generalizability of the model to open networks with multiple box classes and multiple node types may not be feasible. van Dijk (1988) claims that the product-form stationary distribution with jump-over blocking can be extended to the classic open networks with Poisson arrival streams at any node and no capacity constraints, for which the partial balance per node holds. However, his research is limited to single box class and single node type. van der Gaast et al. (2020) proposed to use a semi-open queuing network to facilitate the external arrivals while accounting for the population constraints. When the system reaches its capacity, the incoming box is routed to an external queue. Unfortunately, the stationary distribution of this semi-open network does not admit product form due to the external queue, even with Poisson arrivals and exponential services (Jia and Heragu, 2009). Finally, in the case of multiple segments, if the destination segment is fully occupied but the destination station within that segment is not, the box must still recirculate on the main conveyor at least once before attempting to reenter the congested segment. Therefore, instead of performing an exact analysis on the open block-and-recirculate networks with multiple segments, Chapter 4 intends to conduct a simulation study inspired by a Kramp Group's warehouse in Strullendorf, Germany.

2.2 Cluster-based Storage Assignment (CBSA) with Frequent Item Pairs

A storage assignment policy is a set of rules to determine the allocation of products to storage locations (de Koster, Le-Duc, and Roodbergen, 2007). The policy has a significant impact on the order retrieval time in the warehouse; therefore, this research thread has been receiving increasing attentions from researchers worldwide. A particular storage assignment policy can be classified into one of six commonly used types, namely random storage, closest open location storage, dedicated storage, full-turnover storage, class-based storage, and cluster-based storage. The characteristics of each type are summarized in Table 2.1. Compared to the random storage policy, the class-based and full-turnover policies can reduce picking travel time substantially by putting additional focus on individual items' attributes, such as turnover or ordered quantity. However, a concern about increasing order picking time arises when two frequently-requested-together items are potentially assigned to storage locations far from each other. Therefore, cluster-based storage assignment (CBSA) has been designed to integrate the potential pairwise item affinity derived from historical orders or demand forecasts.

Since the storage assignment problem is classified as non-deterministic polynomial-time hard (NP-hard), the most popular solution method is heuristic approach (Mirzaei, Zaerpour, and de Koster, 2021). The cluster-based storage assignment is often decomposed into two sub-problems: first clustering products to maximize a measure of joint demand affinity between products, followed by allocating the clusters to storage locations. All defined affinity measures involve pairwise item relations. The maximum cluster size is the predefined station capacity and the number of clusters is set to equal the number of picking stations. When a cluster achieves its maximum capacity, it is no longer considered for further clustering. In the second step, the most commonly used objective function is to minimize the travel distance/time in the case of assigning items in the clusters to shelf spaces or to minimize the number of station visits in the case of zoning.

Jane and Laih (2005) proposes a natural cluster model which is a relaxation of the NP-hard p-median homogenous cluster problem. A pairwise similarity measurement between items is defined as the times that both items are requested by the same order. The model allocates item pairs to stations in the descending order of similarity to increase the zone picking utilization. Regarding the problem sizes, Jane and Laih (2005) conducts a case study at a distribution center with 680 items, 22,538 orders, and 8 stations.

Table 2.1: Storage assignment, summarized from de Koster, Le-Duc, and Roodbergen (2007) and Gu, Goetschalckx, and McGinnis (2007).

Policy	Definition	Advantage	Disadvantage	Remark
Random storage	Incoming product is assigned a uniformly distributed random empty location in the warehouse.	High space utilization.	Increased travel distance.	Will only work in a computer-controlled environment.
Closest open location storage	The first empty location that is encountered by the employee will be used to store the products.	Easy to implement.	Racks are full around the depot and gradually more empty towards the back	Might perform equivalently to the randomized storage under certain conditions (Hausman, Schwarz, and Graves, 1976).
Dedicated storage	Each product is stored at a fixed location.	Pickers become familiar with product locations.	A location is reserved even for products that are out of stock. Lowest space utilization	More beneficial when products have different properties.
Full-turnover storage	Products are distributed over the storage area according to their turnover: the higher the sales rates, the closer to the depot.	Easiest implementation when combined with dedicated storage	Regular reshuffling of stock due to a constantly changing demand rates and the product assortment	Other variants cube-per-order index (COI) rule or frequency-based storage.

Table 2.1 (cont.)

Policy	Definition	Advantage	Disadvantage	Remark
Class-based storage	Products are grouped into classes based on popularity (COI or pick frequency), and each class is then assigned to a dedicated area of the warehouse.	Combination of dedicated storage and full-turnover storage	Requires more rack space than randomised storage.	Class-based storage is outperformed by full-turnover storage with regards to the travel distance in a manual order-picking system due to experimental simulation results by Petersen, Aase, and Heiser (2004).
Cluster-based storage	Certain products can be grouped and allocated to a subsection of the warehouse according to some shared properties.	Take advantage of the correlation between products. Easily combine with other strategies.	Computationally more expensive.	A metric to measure the strength of joint demand needs to define. An optimal assignment with all combinations pairs is only possible for warehouses with a very limited number of products (Mirzaei, Zaerpour, and de Koster, 2021).

Garfinkel (2005) develops a heuristic approach in which the clustering is performed by the previously proposed techniques, such as sequential clustering, simultaneous clustering, or particle movement. From this initial construction, the author then performs one of the improvement procedures using two or cyclic item exchange moves in which an item in one station is swapped with an item from another station. The result is presented from a real data set with 10,644 products, 74,202 multi-product orders for a total of 288,870 order-product pairs. This problem size is remarkably larger than the sizes reported in the prior literature which are no more than 1,000 items.

Data mining, or particularly association rule mining (ARM), is a relatively new technique used to discover the interesting patterns or correlations between items contained in large volumes of data. ARM has been integrated into storage assignment in OPS and shown promising results, particularly due to the unprecedented computing power capabilities (Li et al., 2021). Two commonly used indexes of ARM in CBSA literature are support and lift. Support and its variant called support count evaluate the popularity of a rule, in which support count has been employed as the similarity measure in early 2000s CBSA literature. Lift illustrates the type of relationship between products. Li, Moghaddam, and Nof (2016) claims that the lift values greater than, less than, and equal to 1.0 respectively represent complementary, substitute, and independent relationships.

$$\text{Association rule: item } i \rightarrow \text{ item } j$$

$$\text{Support count} = \text{Number of orders containing both } i \text{ and } j$$

$$\text{Support} = \frac{\text{Support count}}{\text{Total number of orders}}$$

$$\text{Lift} = \frac{\text{Support count} \times \text{Total number of orders}}{\text{Number of orders containing } i \times \text{Number of orders containing } j}$$

$$\{i, j\} \text{ are frequent item pairs}$$

$$\Leftrightarrow \text{Support count} \geq \text{Minimum support count threshold (or support threshold for short)}$$

In general, when the number of orders is higher than the number of items and the average order size is small, the support of an association rule can be significantly small. Hence, support count is more suitable for analysis. Note that all three measures are symmetric for i and j. If a minimum support count threshold, hereafter referred to as *support threshold*, is set to define the frequency level. Items i and j are called a frequent item pair if and only if the support count of the pair is at least the support threshold.

Ming-Huang Chiang, Lin, and Chen (2014) combines the concepts of support and lift into an association measure called weighted support count (WSC). Based on the measure, the authors develop two methods, namely modified class-based heuristic (MCBH) and association seed based heuristic (ASBH), to maximize the item association allocated in the same aisle and to reduce the travel distance. The data set includes 338,113 daily orders from a distribution center in which 787 distinct products are stocked in the picking area. It is collected over the period of one year instead of one quarter as in Garfinkel (2005) and Jane and Laih (2005). Li, Moghaddam, and Nof (2016) also employs WSC to represent the intensity and nature of pairwise item relations. Four warehouse configurations are taken into account in which only the first configuration for small-sized warehouses is the real data collected from a family care retailer with 200 orders and 20 items. Li et al. (2021) uses classic ARM with minimum support threshold in addition to assigning items to weight classes. The items are distributed into shelf spaces in the descending order of weight and individual support coefficients. The data set contains 7,881 unique items with 624 transactions. The minimum support threshold is set to attain the best performance for the transaction data in which sufficient item sets with highly correlated items within each item set are detected.

Early 2000s research has been mainly devoted to allocating items from the beginning with an empty warehouse while recent research shows interest in improving the existing storage assignment. Zhang (2016) proposes an insertion algorithm to iteratively improve the result of other storage assignment strategies step by step. However, the experiment result indicates that the CBSA from scratch performs better than the improvement-based CBSA. Wang, Zhang, and Fan (2020) develops data-based approach (DBA) to improve the initial storage assignment by reassigning items across the aisles and subsequently to appropriate modules within each aisle. The proposed DBA is able to quickly find a candidate pair for item swapping to reduce the travel distance at each iteration by taking advantage of the characteristics of historical transactions. Table 2.2 summarizes several studies on CBSA in picker-to-parts OPS and highlights the research gap. Stock splitting implies that an item can be stored in multiple shelf spaces in the warehouse. S-shape routing indicates that a picker traverses every aisle containing pick item(s) entirely in the shape of an S.

Two papers, Garfinkel (2005) and Jane and Laih (2005), are the studies which take into account zoning, i.e., assigning items to picking stations, yet neglecting the exact location of the items in the stations. Out of six papers using the real-life data from Table 2.2, only Jane and Laih (2005) was able to access the current key performance indicator status where they obtained the data. Only one out of seven papers, Li et al. (2021), employs ARM with minimum support threshold yet the authors do not study the impact of decreasing the threshold on the warehouse performance. Moreover, all proposed mathematical models consider either a single objective or a combination of an item affinity and a performance measure, which may lead to a sub-optimal decision.

Table 2.2: Overview of papers using cluster-based storage assignment.

Paper	Comparison	Objective/Key performance indicator	Routing	Zoning	Stock splitting	Improvement-based CBSA	Solution	Real-life data
Jane and Laih (2005)	Existing policy, random storage	Zone picking utilization	-	Yes	No	No	Heuristic	Yes
Garfinkel (2005)	Random and frequency-based storage	Number of station visits	-	Yes	Yes	Yes	Heuristic	Yes
Ming-Huang Chiang, Lin, and Chen (2014)	Class-based storage	Weighted support count/ travel distance	S-shape	No	No	No	Heuristic, MCBH and ASBH	Yes
Li, Moghaddam, and Nof (2016)	Class-based storage	Sum of affinity and turnover/ travel distance	S-shape	No	No	No	Greedy genetic algorithm	Yes
Zhang (2016)	Full-turnover storage	Travel distance	Return	No	No	Yes	Heuristic, sum and static seed	No
Wang, Zhang, and Fan (2020)	Frequency-based storage	Travel distance	S-shape	No	No	Yes	Heuristic, DBA	Yes
Li et al. (2021)	Class-based storage	Travel distance	S-shape	No	No	No	Heuristic, ARM with minimum support threshold	Yes

This thesis contributes by simulating a complex SZPS with zoning and multiple segments as a network of queues. Multiple performance objectives, including the number of station visits, the travel distance of pickers, and the order throughput time, are considered. The real-life transaction data is obtained to construct the frequent item pairs (FIP) graph and generate clusters according to the connected components algorithm in graph theory. The FIP graph also helps reveal and visualize the natural relationship between items. A heuristic algorithm is then built to assign clusters to stations and reassign items to shelf spaces according to an existing storage assignment. The performances of the cluster-based storage assignment with two minimum support count thresholds are then compared with the current policy and with each other in Chapter 4.

3 CLOSED QUEUING NETWORKS WITH BLOCK-AND-RECIRCULATE AND SINGLE SEGMENT

This chapter summarizes the research of van der Gaast et al. (2020) on the single-segment SZPS with capacity constraints, block-and-recirculate protocol, and multiple box classes. Stations with best-selling items may become congested during peak periods due to their limited capacity. Conveyor blockages caused by boxes waiting to enter a full buffer can propagate throughout the system, which leads to unbalanced workload between stations and increased throughput times. The block-and-recirculate protocol has been widely adopted in warehouses to prevent blockages. Instead of forcing the box to enter a full buffer, the protocol allows boxes to recirculate on the conveyor loop and, if necessary, receive picks from other stations before attempting to reenter the previously congested station. However, because of the finite capacity, the queuing models with this blocking mechanism are highly intractable with no exact formulas for the stationary distribution. As a result, a jump-over blocking protocol with equivalent Markovian routing and matching network flow is developed to establish an approximation network. The chapter starts with general notation and assumptions for the queuing network in Section 3.1. Section 3.2 introduces the block-and-recirculate protocol and derives the routing probability of the network employing the protocol. Section 3.3 sets up the equivalent jump-over blocking protocol to the block-and-recirculate and elaborates on the proof that he jump-over network admits a product-form stationary distribution. The proof makes use of Kelly's Theorem for time-reversed processes and is an extension to van Dijk (1988)'s original proof with multiple box classes and three different queue types.

3.1 General Notation and Assumptions

van der Gaast et al. (2020) model the SZPS as a closed cyclic queuing network with one entrance/exit e, M stations with capacity constraints, and $M + 1$ conveyor nodes connecting either two adjacent stations or the entrance/exit and the first/last station, respectively. Each box is assumed to have infinite capacity; therefore, a box can represent an order regardless of the order size. A saturated system with N boxes presenting at any time is assumed to have an infinite supply of orders at the entrance. This assumption guarantees that a box entering the entrance/exit e is instantaneously assigned to a new order and getting ready to leave node e. At all nodes, boxes are served according to First-In-First-Out (FIFO) discipline. The closed queuing network with block-and-recirculate protocol and single segment is visualized in Figure 2.2. The general notation with additional constraints and assumptions is summarized as follows.

- N: the box population in the closed queuing network.
- M: the number of stations in the network.
- e: the entrance/exit.
- $S = \{s_1, \dots, s_i, \dots, s_M\}$: the set of stations, indexed by s_i where $i \in [1, M]$.
- $C = \{c_1, \dots, c_i, \dots, c_{M+1}\}$: the set of conveyors, indexed by c_i where $i \in [1, M+1]$.
- $Q = \{e\} \cup S \cup C$: the set of all nodes in the network, indexed by j.
- $\mathbf{r} \subseteq S$: the class of a box, i.e., the set of stations it has to visit. After visiting station $s_i \in \mathbf{r}$, its class changes to $\mathbf{r} \setminus \{s_i\}$. When $\mathbf{r}$ reduces to $\emptyset$, the box has finished all picks and is ready to leave the system.
- μ_e: the rate of exponential distribution at which new boxes are released from the entrance/exit e, as long as the total number of boxes in other nodes of the system is less than N. The rate reflects the preparation rate of a box before entering the system. Moreover, there is an infinite supply of orders at the entrance.
- $\mu_{c_i},\ c_i \in C$: the rate of the exponentially distributed service time at conveyor node c_i, i.e., a box spends on average $1/\mu_{c_i}$ seconds on c_i.
- $\mu_{s_i},\ s_i \in S$: the rate of the exponentially distributed order picking time at station s_i to capture the variations of both the pick time and the number of assigned items per box.
- $d_{s_i},\ s_i \in S$: the finite number of servers, or order pickers, at a station s_i. A picker picks one box at a time, i.e., no batch servicing. One station must have at least one picker, i.e., $1 \leq d_{s_i} < \infty$.
- $q_{s_i},\ s_i \in S$: the buffer size at station s_i, i.e., the maximum number of boxes in the queue of station s_i. Incoming boxes are blocked when the total number of boxes in the buffer equals q_{s_i}.
- $n_j,\ j \in Q$: the number of boxes in the queue of node j. Since the box population is N, $\sum_{j \in Q} n_j = N$. Due to the capacity constraint at the station $s_i \in S$, $n_{s_i} \leq d_{s_i} + q_{s_i}$.

- $x_j = (x_{j1}, \ldots, x_{jl}, \ldots, x_{jn_j}),\ j \in Q$: the detailed state of node j where x_{jl} is the class of the box in position l of node j. The state includes both boxes in service and boxes in the buffer.

- $x = (x_j : j \in Q)$: the state of the queuing network.

- $\mathbb{S}(N)$: the state space of the closed network with N recirculated boxes.

- $\psi_{\mathbf{r}},\ \mathbf{r} \subseteq S$: the probability that a newly released box belongs to class $\mathbf{r}$. These release probabilities can be obtained from historical order data or forecasts. As it is trivial to release a box with no picks into the system, $\psi_{\emptyset} = 0$ can be assumed.

- $\lambda_{j\mathbf{r}},\ j \in Q,\ \mathbf{r} \subseteq S$: the visit ratio of a class $\mathbf{r}$ box to node j; in other words, the mean number of times a class $\mathbf{r}$ box joins the queue in node j.

- $p_{j\mathbf{r},k\mathbf{v}}(x),\ j,k \in Q,\ \mathbf{r},\mathbf{v} \subseteq S,\ x \in \mathbb{S}(N)$: the state-dependent routing probability that a class $\mathbf{r}$ box travels from node j to node k and changes its class to $\mathbf{v}$, given the state x of the system.

- $q(x, y),\ x, y \in \mathbb{S}(N)$: the transition rate from state x to state y.

When the network is first initiated, N boxes of class $\emptyset$ waiting at the entrance are ready to be fed. The queue characteristics of different node types in the system is summarized in Table 3.1. The infinite number of servers at conveyor node c_i implies that there is no queue in front of the node and a box starts travelling on the conveyor as soon as it arrives. The queue discipline and service time at all nodes are box-class independent. Due to the box-class dependent routing (which is demonstrated in the next section), the exact order of boxes at each node needs to be included in the current state x in order to predict the system's future state.

Table 3.1: Comparison of the queue characteristics of different node types in the network.

Node	Number of servers	Buffer size	Queue discipline
Entrance/exit e	1	∞	FIFO
Conveyor c_i	∞	∞	FIFO
Station s_i	$1 \leq d_{s_i} < \infty$	$0 \leq q_{s_i} < \infty$	FIFO

3.2 Block-and-Recirculate Protocol

The block-and-recirculate protocol is illustrated in Figure 3.1 in which class $\mathbf{r}$ boxes with $s_i \in \mathbf{r}$ either enter station s_i and leave with class $\mathbf{r} \setminus \{s_i\}$ when the queue is not full or skip station s_i and maintain class $\mathbf{r}$ when the queue is full.

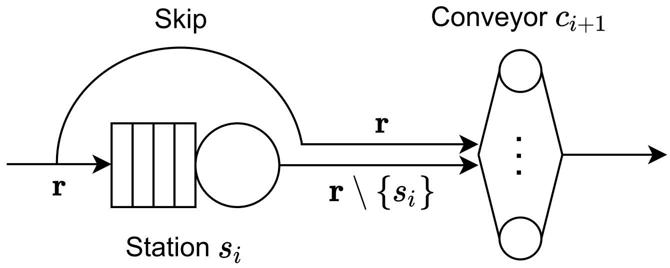


Figure 3.1: Illustration of station blocking in a block-and-recirculate network.

3.2.1 Routing Probability

The routing probability of the block-and-recirculate network is state-dependent due to the finite station capacity. Additionally, the box routing through the system is Markovian since the future state of the box is independent of the past provided its present state. Its non-zero entries are described as follows.

- When there is an empty box, i.e., a box of class $\emptyset$, at the entrance e, the network assigns to the box a new order which has not been fulfilled yet. The box then receives its new class $\mathbf{r}$ with probability $\psi_{\mathbf{r}}$ and moves from the entrance e to the first conveyor node c_1,

$$p_{e\emptyset,c_1\mathbf{r}}(x) = \psi_{\mathbf{r}}. \tag{3.1}$$

- From conveyor node c_i, the class $\mathbf{r}$ box enters the buffer of station s_i if $s_i \in \mathbf{r}$ and the buffer is not full,

$$p_{c_i\mathbf{r},s_i\mathbf{r}}(x) = 1, \quad i = 1, \ldots, M,\ s_i \in \mathbf{r},\ \text{and } n_{s_i} < d_{s_i} + q_{s_i}. \tag{3.2}$$

  It is assumed that the first d_{s_i} boxes in the state x_{s_i} are always in service, i.e., an idle picker starts retrieving items for a box as soon as it arrives. The boxes remain in the state x_{s_i} until their pick service from station s_i are completed.

- From conveyor node c_i, the class $\mathbf{r}$ box moves to the next conveyor piece c_{i+1} if $s_i \notin \mathbf{r}$ or $s_i \in \mathbf{r}$ but the buffer is full. The class of the box remains unchanged,

$$p_{c_i\mathbf{r},c_{i+1}\mathbf{r}}(x) = 1, \quad i = 1, \ldots, M,\ s_i \notin \mathbf{r},\ \text{or } n_{s_i} = d_{s_i} + q_{s_i}. \tag{3.3}$$

- From station s_i, after all picks are performed, the box enters the subsequent conveyor node c_{i+1}, and its class changes to $\mathbf{v} = \mathbf{r} \setminus \{s_i\}$,

$$p_{s_i\mathbf{r},c_{i+1}\mathbf{v}}(x) = 1, \quad i = 1, \ldots, M, s_i \in \mathbf{r},\ \mathbf{v} = \mathbf{r} \setminus \{s_i\}. \tag{3.4}$$

- After visiting the last conveyor node c_{M+1}, all boxes with incomplete picks $\mathbf{r} \neq \emptyset$ are recirculated to the first conveyor node c_1,

$$p_{c_{M+1}\mathbf{r},c_1\mathbf{r}}(x) = 1, \quad \mathbf{r} \neq \emptyset. \tag{3.5}$$

- Boxes with complete picks move to the exit e from conveyor c_{M+1},

$$p_{c_{M+1}\emptyset,e\emptyset}(x) = 1. \tag{3.6}$$

The routing probabilities (3.2), (3.3), and (3.4) implies that class $\mathbf{r}$ boxes skipping station s_i or leaving a conveyor node maintain their class while boxes entering and leaving station s_i always change to class $\mathbf{r} \setminus \{s_i\}$ before continuing its circulation on conveyor c_{i+1}. Since class $\mathbf{r}$ boxes can reach from each node to any other nodes in one or more transitions, the routing matrix is irreducible.

Because of the finite buffers and the state-dependent routing probability, there is no exact formula for the stationary distribution of the block-and-recirculate network. The next section describes how to construct the jump-over network to approximate the performance of the block-and-recirculate network, starting from the routing probability to traffic equations and transition rates.

3.3 Jump-over Blocking Protocol

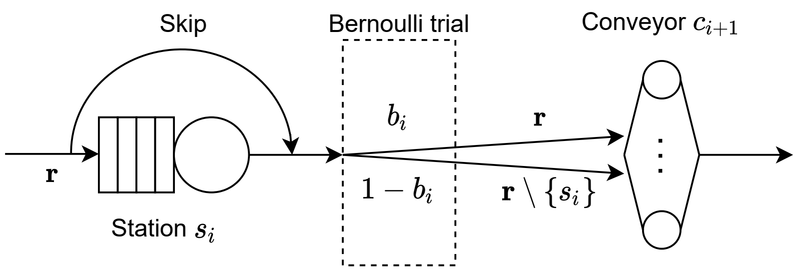


Figure 3.2: Illustration of station blocking in a jump-over network.

In order to approximate the block-and-recirculate protocol, jump-over blocking is chosen due to the similar characteristics of "overtaking full stations, skipping, and blocking and rerouting" (van der Gaast et al., 2020). Moreover, jump-over protocol admits a product-form stationary distribution for single-class networks of queues (van Dijk, 1988). The product-form solution allows the performance metric for the collection of components to be written as a product of the metric across the components. This makes the computation inexpensive to evaluate the system for large numbers of components. As demonstrated in Figure 3.2, the jump-over network allows class $\mathbf{r} \subseteq S$ boxes with $s_i \in \mathbf{r}$ to reduce their class to $\mathbf{r} \setminus \{s_i\}$ based on a Bernoulli trial regardless of whether they visit or skip station s_i. The detailed process is described as follows. A class $\mathbf{r}$ box requesting service at station s_i while station s_i is saturated, i.e., $n_{s_i} = d_{s_i} + q_{s_i}$, will instantly skip s_i and move to the subsequent conveyor node c_{i+1} as if it was served at station s_i at an infinite speed. Independent of whether the box visits or skips station s_i, a Bernoulli trial is defined to determine whether its class changes before entering conveyor node c_{i+1}: the class is maintained with probability b_{s_i} and changed to $\mathbf{r} \setminus \{s_i\}$ otherwise. By setting b_{s_i} equal the blocking probability of station s_i under the block-and-recirculate policy, i.e., the fraction of boxes encountering a full queue at station s_i, the flows of class $\mathbf{r}$ and class $\mathbf{r} \setminus \{s_i\}$ boxes visiting conveyor piece c_{i+1} in both networks become equivalent. Although the blocking probability is unknown in advance, it can be estimated iteratively by applying the Mean Value Analysis algorithm.

3.3.1 Routing Probability

Based on the description of the jump-over network, the nonzero routing probability under jump-over protocol is constructed as follows (cf. (3.2)–(3.4)), together with (3.1), (3.5), and (3.6) from the block-and-recirculate network. These probabilities are state-independent, i.e., they do not depend on whether the station buffer is full or not.

$$p_{c_i\mathbf{r},s_i\mathbf{r}} = 1, \qquad i = 1,\ldots,M, s_i \in \mathbf{r}, \tag{3.7}$$

$$p_{c_i\mathbf{r},c_{i+1}\mathbf{r}} = 1, \qquad i = 1,\ldots,M, s_i \notin \mathbf{r}, \tag{3.8}$$

$$p_{s_i\mathbf{r},c_{i+1}\mathbf{r}} = b_{s_i}, \qquad i = 1,\ldots,M, s_i \in \mathbf{r}, \tag{3.9}$$

$$p_{s_i\mathbf{r},c_{i+1}\mathbf{v}} = 1 - b_{s_i}, \qquad i = 1,\ldots,M, s_i \in \mathbf{r}, \mathbf{v} = \mathbf{r} \setminus \{s_i\} \tag{3.10}$$

3.3.2 Traffic Equations

The traffic equations for multiple box classes under a closed jump-over network are computed as follows,

$$\lambda_{j\mathbf{r}} = \sum_{k\in Q}\sum_{\mathbf{v}\subseteq S} \lambda_{k\mathbf{v}} p_{k\mathbf{v},j\mathbf{r}}. \tag{3.11}$$

$\lambda_{j\mathbf{r}}$ is given by the sum of arrivals of any arbitrary class $\mathbf{v} \subseteq S$ boxes from every other node in the network than j, denoted by k, with its class changing to $\mathbf{r}$ after its service at node k. By substituting the routing probabilities from Eq. (3.1), (3.5)–(3.10), the traffic equations are simplified to,

$$
\begin{aligned}
\lambda_{e\emptyset} &= \lambda_{c_{M+1}\emptyset}, && && (3.12)\\
\lambda_{c_1\mathbf{r}} &= \lambda_{e\emptyset}\psi_{\mathbf{r}} + \lambda_{c_{M+1}\mathbf{r}}, && \mathbf{r} \subseteq S && (3.13)\\
\lambda_{c_{i+1}\mathbf{r}} &= \lambda_{s_i\mathbf{r}} b_{s_i}, && i = 1, \dots, M, s_i \in \mathbf{r} \subseteq S && (3.14)\\
\lambda_{c_{i+1}\mathbf{r}} &= \lambda_{c_i\mathbf{r}} + \lambda_{s_i\mathbf{r}\cup\{s_i\}}(1 - b_{s_i}), && i = 1, \dots, M, s_i \notin \mathbf{r} \subseteq S && (3.15)\\
\lambda_{s_i\mathbf{r}} &= \lambda_{c_i\mathbf{r}}, && i = 1, \dots, M, s_i \in \mathbf{r} \subseteq S && (3.16)
\end{aligned}
$$

whereas all other visit ratios equal zero. Eq. (3.12) is derived from the routing probability (3.6), i.e., a box visits the exit e only if all picks are completed and its class has been reduced to $\emptyset$ before visiting the last conveyor node c_{M+1}. Eq. (3.13) results from the fact that a box of class $\mathbf{r} \subseteq S$ is routed to the first conveyor node c_1 if and only if it is either newly released from the entrance e or recirculated from the last conveyor c_{M+1}. Regarding the visit ratio $\lambda_{c_{i+1}\mathbf{r}}$, a class $\mathbf{r}$ box visits conveyor node c_{i+1} for $1 \leq i \leq M$ only after it passes station s_i. There are two possible scenarios happening at station s_i. If $s_i \in \mathbf{r}$, the box must have been prevented from entering station s_i with the blocking probability b_{s_i}, which results in Eq. (3.14). Otherwise, when $s_i \notin \mathbf{r}$, the box is routed to c_{i+1} either from c_i when it does not need picks from station s_i, or from station s_i after all picks at the station are completed. In the latter case, the box has the previous state of $\mathbf{r} \cup \{s_i\}$ and manages to visit the station s_i with probability $1 - b_{s_i}$. These two scenarios lead to Eq. (3.15). For the last traffic equation (3.16), all box visits to station s_i must be routed from the preceding conveyor piece c_i providing that $s_i \in \mathbf{r}$. A box maintains its class when it leaves a conveyor node, i.e., the class $\mathbf{r}$ of a box remains unchanged.

The routing matrix is irreducible due to the fact that a box, regardless of its class, can reach from an arbitrary node to any other nodes in one or more transitions. As a result, when the throughput of a randomly chosen queue is normalised to a constant, other throughputs can be expressed relative to the reference queue; therefore, the traffic equations have a unique state-independent solution.

3.3.3 Transition Rates

In order to derive the transition rate $q(x, y)$, the notation $x - \mathbf{r}_{jl} + \mathbf{v}_{km}$ is used to specify the state obtained from state x when the class $\mathbf{r}$ box in position $l \in [1, n_j]$ of node j, i.e., $\mathbf{r} = x_{jl}$, leaves node j, changes its class to $\mathbf{v} \subseteq S$ and enters position m of node $k \in Q$. Since there is no box class priority, whenever a box joins a queue k its position is at the end of the queue, i.e., $m = n_k + 1$.

In state x, a newly released class $\mathbf{r}$ box from entrance e certainly moves to conveyor piece c_1. Since there is one server at the entrance/exit e, only the transition rate involving the box in the first position of e move to the last position of c_1 is nonzero. The resulted state due to this event is $x - \emptyset_{e1} + \mathbf{r}_{c_1 n_{c_1}+1}$, and the transition rate equals the rate at which a new box is released from the entrance e multiplying with the probability of that box belonging to class $\mathbf{r}$. This event occurs provided that the total number of boxes in all nodes except node e is smaller than N; in other words, $n_e > 0$. When the system is stable, n_e increments every time a box of class $\emptyset$ arrives at the entrance/exit e from conveyor c_{M+1}. By the routing probability (3.1),

$$q(x, x - \emptyset_{e1} + \mathbf{r}_{c_1 n_{c_1}+1}) = \mu_e p_{e\emptyset, c_1\mathbf{r}} = \mu_e \psi_{\mathbf{r}}, \quad n_e > 0. \tag{3.17}$$

If the class $\mathbf{r}$ box in position l of conveyor node c_i, $i \leq M$ completes its transportation on the conveyor and $s_i \notin \mathbf{r}$, the box continues to the next conveyor c_{i+1} without visiting station s_i, hence from the routing probability (3.8),

$$q(x, x - \mathbf{r}_{c_i l} + \mathbf{r}_{c_{i+1} n_{c_{i+1}}+1}) = \mu_{c_i} p_{c_i\mathbf{r}, c_{i+1}\mathbf{r}} = \mu_{c_i}, \quad s_i \notin \mathbf{r}. \tag{3.18}$$

In case $s_i \in \mathbf{r}$ and there is room in the buffer, the box enters the position $n_{s_i}+1$ of station s_i. The transition rate for this event is derived from the routing probability (3.7),

$$q(x, x - \mathbf{r}_{c_i l} + \mathbf{r}_{s_i n_{s_i}+1}) = \mu_{c_i} p_{c_i\mathbf{r}, s_i\mathbf{r}} = \mu_{c_i}, \quad s_i \in \mathbf{r}, \text{ and } n_{s_i} < d_{s_i} + q_{s_i}. \tag{3.19}$$

Otherwise when the buffer is saturated, the box is immediately routed to the conveyor piece c_{i+1} with its class either changing to $\mathbf{r} \setminus \{s_i\}$ with probability $1 - b_{s_i}$ or unchanged with probability b_{s_i} based on (3.9) and (3.10),

$$\begin{aligned} q(x, x - \mathbf{r}_{c_i l} + \mathbf{v}_{c_{i+1} n_{c_{i+1}}+1}) &= \mu_{c_i} p_{c_i\mathbf{r}, s_i\mathbf{r}} p_{s_i\mathbf{r}, c_{i+1}\mathbf{v}} \\ &= \begin{cases} \mu_{c_i}(1 - b_{s_i}), & s_i \in \mathbf{r},\ \mathbf{v} = \mathbf{r} \setminus \{s_i\}, \\ \mu_{c_i} b_{s_i}, & s_i \in \mathbf{r},\ \mathbf{v} = \mathbf{r}, \end{cases} \quad \text{and } n_{s_i} = d_{s_i} + q_{s_i}. \end{aligned} \tag{3.20}$$

If a picker at station s_i finishes picking items for the class $\mathbf{r}$ box in position l, the box traverses to the next conveyor node c_{i+1}. Since the number of pickers (or servers) at station s_i is finite, only the transition rates involving boxes at location $1 \leq l \leq \min(d_{s_i}, n_{s_i})$ are nonzero. Hence from routing probabilities (3.9) and (3.10),

$$\begin{aligned} q(x, x - \mathbf{r}_{s_i l} + \mathbf{v}_{c_{i+1} n_{c_{i+1}}+1}) &= \mu_{s_i} p_{s_i\mathbf{r}, c_{i+1}\mathbf{v}} \\ &= \begin{cases} \mu_{s_i}(1 - b_{s_i}), & \mathbf{v} = \mathbf{r} \setminus \{s_i\}, \\ \mu_{s_i} b_{s_i}, & \mathbf{v} = \mathbf{r}, \end{cases} \end{aligned} \tag{3.21}$$

Finally, the class $\mathbf{r}$ box in position l of the last conveyor c_{M+1} finishes transportation. If its class $\mathbf{r}$ reduces to an empty set, the box enters the exit e; otherwise, it continues its circulation to the first conveyor c_1. Therefore, according to the routing probabilities (3.5) and (3.6),

$$q(x, x - \mathbf{r}_{c_{M+1}l} + \mathbf{r}_{jn_j+1}) = \begin{cases} \mu_{c_{M+1}}, & \mathbf{r} = \emptyset,\ j = e, \\ \mu_{c_{M+1}}, & \mathbf{r} \neq \emptyset,\ j = c_1. \end{cases} \tag{3.22}$$

Transition rates (3.18)–(3.20), and (3.22) involve the class $\mathbf{r}$ box leaving position l at conveyor node c_i, $i = 1, \ldots, M+1$. Since all conveyor nodes have infinite servers, all boxes in the state are served simultaneously, i.e., position l can be any values within the range $[1, n_{c_i}]$.

All other transition rates of the jump-over network for a particular state x and a particular class $\mathbf{r}$ box equals zero. The total transition rate from state x is then computed as follows, where $\mathbb{1}_{(.)}$ is the indicator function.

$$\begin{aligned}
q(x) &= \sum_{y \in \mathbb{S}(N)} q(x, y) \\
&= \sum_{\mathbf{r} \subseteq S} q(x, x - \emptyset_{e1} + \mathbf{r}_{c_1 n_{c_1}+1}) \mathbb{1}_{(n_e > 0)} \\
&\quad + \sum_{i=1}^{M} \sum_{l=1}^{n_{c_i}} \Big[q(x, x - \mathbf{r}_{c_i l} + \mathbf{r}_{c_{i+1} n_{c_{i+1}}+1}) \mathbb{1}_{(s_i \notin \mathbf{r})} \\
&\qquad\qquad + q(x, x - \mathbf{r}_{c_i l} + \mathbf{r}_{s_i n_{s_i}+1}) \mathbb{1}_{(s_i \in \mathbf{r})} \mathbb{1}_{(n_{s_i} < d_{s_i} + q_{s_i})} \\
&\qquad\qquad + \sum_{\mathbf{v} \subseteq S} q(x, x - \mathbf{r}_{c_i l} + \mathbf{v}_{c_{i+1} n_{c_{i+1}}+1}) \mathbb{1}_{(s_i \in \mathbf{r})} \mathbb{1}_{(n_{s_i} = d_{s_i} + q_{s_i})} \Big] \\
&\quad + \sum_{l=1}^{n_{c_{M+1}}} \Big[q(x, x - \mathbf{r}_{c_{M+1} l} + \mathbf{r}_{e n_e+1}) \mathbb{1}_{(\mathbf{r} = \emptyset)} + q(x, x - \mathbf{r}_{c_{M+1} l} + \mathbf{r}_{c_1 n_{c_1}+1}) \mathbb{1}_{(\mathbf{r} \neq \emptyset)} \Big] \\
&\quad + \sum_{i=1}^{M} \sum_{l=1}^{\min(d_{s_i}, n_{s_i})} \sum_{\mathbf{v} \subseteq S} q(x, x - \mathbf{r}_{s_i l} + \mathbf{v}_{c_{i+1} n_{c_{i+1}}+1}) \qquad (3.23) \\
&= \sum_{\mathbf{r} \subseteq S} \mu_e \psi_{\mathbf{r}} \mathbb{1}_{(n_e > 0)} \\
&\quad + \sum_{i=1}^{M} \sum_{l=1}^{n_{c_i}} \Big[\mu_{c_i} \mathbb{1}_{(s_i \notin \mathbf{r})} + \mu_{c_i} \mathbb{1}_{(s_i \in \mathbf{r})} \mathbb{1}_{(n_{s_i} < d_{s_i} + q_{s_i})} + \big[\mu_{c_i}(1 - b_{s_i}) + \mu_{c_i} b_{s_i} \big] \mathbb{1}_{(s_i \in \mathbf{r})} \mathbb{1}_{(n_{s_i} = d_{s_i} + q_{s_i})} \Big] \\
&\quad + \sum_{l=1}^{n_{c_{M+1}}} \Big[\mu_{c_{M+1}} \mathbb{1}_{(\mathbf{r} = \emptyset)} + \mu_{c_{M+1}} \mathbb{1}_{(\mathbf{r} \neq \emptyset)} \Big] + \sum_{i=1}^{M} \sum_{l=1}^{\min(d_{s_i}, n_{s_i})} \Big[\mu_{s_i}(1 - b_{s_i}) + \mu_{s_i} b_{s_i} \Big] \\
&= \mu_e \mathbb{1}_{(n_e > 0)} \sum_{\mathbf{r} \subseteq S} \psi_{\mathbf{r}} + \sum_{i=1}^{M} \sum_{l=1}^{n_{c_i}} \mu_{c_i} + \sum_{l=1}^{n_{c_{M+1}}} \mu_{c_{M+1}} + \sum_{i=1}^{M} \sum_{l=1}^{\min(d_{s_i}, n_{s_i})} \mu_{s_i} \\
&= \mu_e \mathbb{1}_{(n_e > 0)} + \sum_{i=1}^{M+1} n_{c_i} \mu_{c_i} + \sum_{i=1}^{M} \min(d_{s_i}, n_{s_i}) \mu_{s_i}. \qquad (3.24)
\end{aligned}$$

The total transition rate from state x can be broken down into four summations, corresponding to the entrance/exit e, conveyor nodes c_1 to c_M, the last conveyor node c_{M+1}, and stations s_1 to s_M, as specified in (3.23). The first part describes the sum of the transition rate of a box being released from the entrance/exit e over all its possible class $\mathbf{r}$ under a condition that $n_e > 0$. By substituting (3.17) and applying the initial assumption that $\sum_{\mathbf{r}\subseteq S} \psi_{\mathbf{r}} = 1$, this part is simplified to $\mu_e \mathbb{1}_{(n_e>0)}$. The second part of (3.23) takes into account all the transition rates of a box at position l of conveyor c_i where $i = 1, \dots M$, n_{c_i} possible values for position l in each node, and three possible scenarios (3.18)–(3.20). The third part demonstrates two cases of a box at position $l \in [1, n_{c_{M+1}}]$ leaving the last conveyor c_{M+1}, i.e., whether its class has reduced to $\emptyset$ yet (3.22). The last part is related to the transition rates of a box after its picks at station s_i are complete (3.21). These four summations are simplified to (3.24).

3.3.4 Product-form Stationary Distribution for Each Node as Stand-alone Queue

If the equilibrium condition exists, each node in a queuing network acts as if it were a stand-alone queue. It implies that the stationary distribution of a node in the network are independent of other nodes' states. This section then aims to prove the product-form stationary distributions for three distinctive cases, namely the entrance/exit e, the station s_i, and the conveyor piece c_i. The state transition diagram for each type of node is illustrated and the global balance equations are employed to verify the proposed stationary distributions. The common notation for the proofs is defined as follows.

- $\pi_j(x_j),\ j \in Q$: the probability that the stationary process of node j is in state x_j.
- $\pi(x),\ x \in \mathbb{S}(N)$: the probability that the stationary process of the jump-over network is in state x.
- $x_j - x_{jk} = (x_{j1}, \dots, x_{j,k-1}, x_{j,k+1}, \dots, x_{jn_j}),\ k \in [1, n_j]$: the state obtained from x_j by removing a box in position k of node j. The class of the removed box is x_{jk}.
- $x_j + \mathbf{r}_{jk} = (x_{j1}, \dots, x_{j,k-1}, \mathbf{r}_{jk}, x_{jk}, \dots, x_{jn_j}),\ k \in [1, n_j + 1],\ \mathbf{r} \subseteq S$: the state obtained from x_j by inserting an arbitrary class $\mathbf{r}$ box in position k of node j.

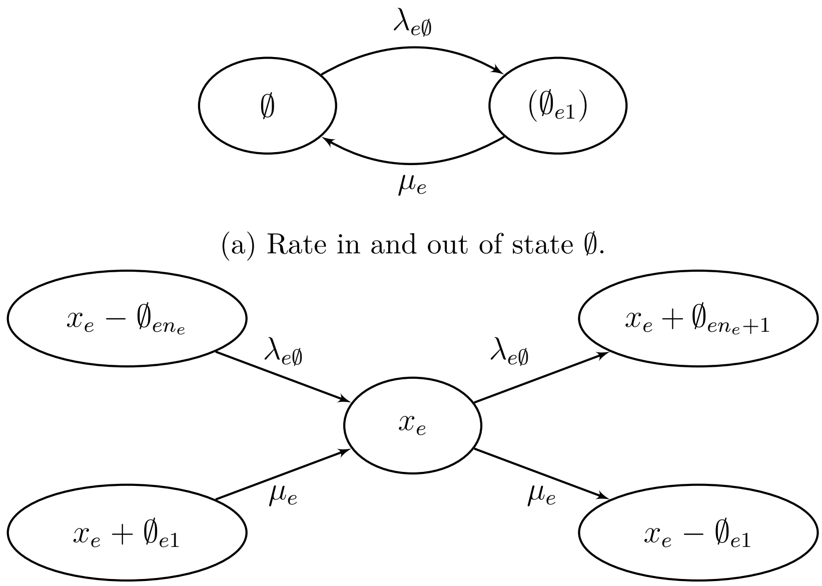
Three theorems, 3.1, 3.2, and 3.3, derive the stationary distribution for the entrance/exit e, the conveyor nodes c_i, and the stations s_i, respectively. The characteristics of each node is given in the bracket. Each theorem is followed by the state transition diagram and the corresponding proof. In all cases, the probability that the stationary process of node j in state $x_j = \emptyset$ is $\pi_j(\emptyset) = 1/G_j$ in which G_j is the normalization constant of the stationary distribution at node j.

Theorem 3.1 (Single-server infinite-capacity queue with multiple box classes)**.** *The stationary distribution of the stand-alone entrance/exit e is of the following form,*

$$\pi_e(x_e) = \frac{1}{G_e} \prod_{l=1}^{n_e} \frac{\lambda_{ex_{el}}}{\mu_e}, \tag{3.25}$$

where $G_e = \sum_{x_e} \pi_e(x_e)$ is the normalization constant, provided that the queue is stable,

$$\sum_{\mathbf{r} \subseteq S} \frac{\lambda_{e\mathbf{r}}}{\mu_e} < 1.$$



(b) Rate in and out of state $x_e \neq \emptyset$.

Figure 3.3: State transition diagram of the entrance/exit e with one server and infinite capacity involving a particular state x_e.

Proof. For a single-server infinite-capacity node, e.g., the entrance/exit e, the state transition diagram is drawn in Figure 3.3. Since only boxes with state $\emptyset$ can enter the entrance/exit e, the class of a removed or inserted box is always $\emptyset$. Due to the single server, only the first box in state x_e is in service. In Figure 3.3a, state $\emptyset$ can reach and be reached from state $(\emptyset_{e1})$ only by the arrival of a class $\emptyset$ box and the release of an arbitrary class $\mathbf{r}$ box onto conveyor c_1, respectively. For a state $x_e \neq \emptyset$, the transitions into and out of the state involve both box arrival to enter the last position and box release from the first position as shown in Figure 3.3b.

From (3.25) of Theorem 3.1, the relation between the stationary distribution of node e at state x_e and states $x_e - x_{ek}$ for $k = 1, \ldots, n_e$ and $x_e + \mathbf{r}_{ek}$ for $k = 1, \ldots, n_e + 1$ is derived as follows,

$$\begin{aligned}\pi_e(x_e - x_{ek}) &= \pi_e(x_{e1}, \ldots, x_{e,k-1}, x_{e,k+1}, \ldots, x_{en_e})\\ &= \frac{1}{G_e}\prod_{l=1}^{k-1}\frac{\lambda_{ex_{el}}}{\mu_e}\cdot\prod_{l=k+1}^{n_e}\frac{\lambda_{jx_{el}}}{\mu_e}\\ &= \pi_e(x_e)\cdot\frac{\mu_e}{\lambda_{ex_{ek}}},\end{aligned} \tag{3.26}$$

$$\begin{aligned}\pi_e(x_e + \mathbf{r}_{ek}) &= \pi_e(x_{e1}, \ldots, x_{e,k-1}, \mathbf{r}_{ek}, x_{ek}, \ldots, x_{en_e})\\ &= \frac{1}{G_e}\prod_{l=1}^{k}\frac{\lambda_{ex_{el}}}{\mu_e}\cdot\frac{\lambda_{e\mathbf{r}}}{\mu_e}\cdot\prod_{l=k+1}^{n_e}\frac{\lambda_{ex_{el}}}{\mu_e}\\ &= \pi_e(x_e)\cdot\frac{\lambda_{e\mathbf{r}}}{\mu_e}.\end{aligned} \tag{3.27}$$

The global balance equations at node e at given state $\emptyset$ and state $x_e \neq \emptyset$ are,

$$\text{Figure 3.3a} \Leftrightarrow \qquad \lambda_{e\emptyset}\pi_e(\emptyset) = \mu_e\pi_e(\emptyset_{e1}) \tag{3.28}$$

$$\text{Figure 3.3b} \Leftrightarrow (\lambda_{e\emptyset} + \mu_e)\,\pi_e(x_e) = \lambda_{e\emptyset}\pi_e(x_e - \emptyset_{en_e}) + \mu_e\pi_e(x_e + \emptyset_{e1}), \qquad \text{for } x_e \neq \emptyset. \tag{3.29}$$

By substituting (3.25), (3.26), and (3.27) into (3.28) and (3.29),

$$(3.28) \Leftrightarrow \qquad \lambda_{e\emptyset}\frac{1}{G_e} = \mu_e\frac{1}{G_e}\cdot\frac{\lambda_{e\emptyset}}{\mu_e} \tag{3.30}$$

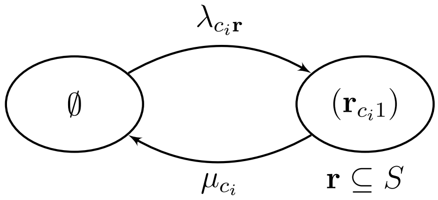
$$\begin{aligned}(3.29) \Leftrightarrow (\lambda_{e\emptyset} + \mu_e)\,\pi_e(x_e) &= \lambda_{e\emptyset}\pi_e(x_e)\frac{\mu_e}{\lambda_{e\emptyset}} + \mu_e\pi_e(x_e)\frac{\lambda_{e\emptyset}}{\mu_e}, \qquad \text{for } x_e \neq \emptyset,\\ &= \mu_e\pi_e(x_e) + \lambda_{e\emptyset}\pi_e(x_e)\end{aligned} \tag{3.31}$$

Therefore, the equalities (3.30) and (3.31) hold, implying that (3.25) satisfies the global balance equations. It can be concluded that $\pi_e(x_e)$ is the stationary distribution for the stand-alone entrance/exit e. □

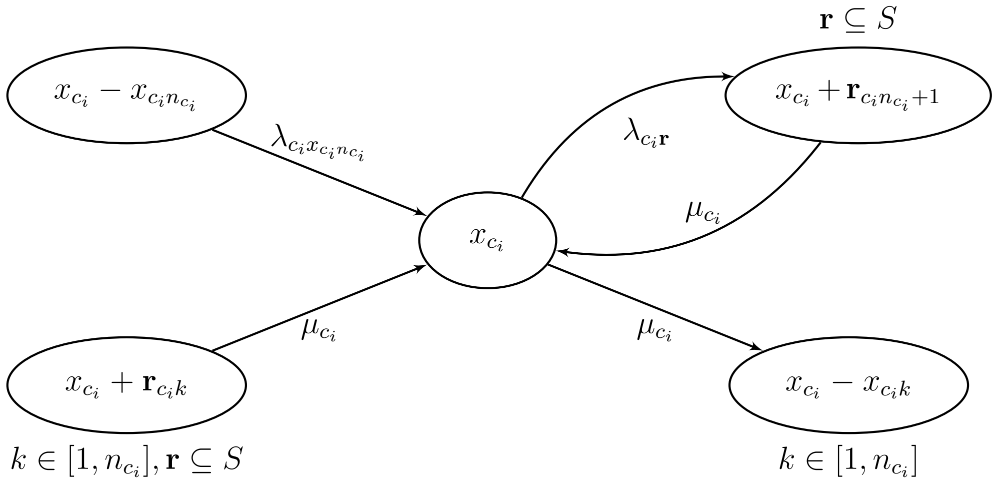
Theorem 3.2 (Infinite-server infinite-capacity queue with multiple box classes)**.** *The stationary distribution of the stand-alone conveyor piece $c_i \in C$ is of the following form,*

$$\pi_{c_i}(x_{c_i}) = \frac{1}{G_{c_i}}\prod_{l=1}^{n_{c_i}}\frac{\lambda_{c_i x_{c_i l}}}{\mu_{c_i}}\cdot\frac{1}{n_{c_i}!}, \tag{3.32}$$

where $G_{c_i} = \sum_{x_{c_i}}\pi_{c_i}(x_{c_i})$ is the normalization constant.



(a) Rate in and out of state $\emptyset$.



(b) Rate in and out of state $x_{c_i} \neq \emptyset$.

Figure 3.4: State transition diagram of conveyor c_i with infinite servers and infinite capacity involving a particular state x_{c_i}.

Proof. For infinite-server infinite-capacity conveyor nodes, the state transition diagram is illustrated in Figure 3.4. Thanks to the infinite number of servers, once an arbitrary box enters a conveyor node $c_i, i = 1, \ldots, M+1$, it is immediately served. As a result, the transitions into and out of state x_{c_i} involves the service completion of all positions rather than just the first box as in the case of a single-server infinite-capacity queue.

From (3.32) in Theorem 3.2, the relation between the stationary distribution of node c_i at state x_{c_i} and states $x_{c_i} - x_{c_i k}$ for $k = 1, \ldots, n_{c_i}$ and $x_{c_i} + \mathbf{r}_{c_i k}$ for $k = 1, \ldots, n_{c_i}+1$ is derived as follows,

$$
\begin{aligned}
\pi_{c_i}(x_{c_i} - x_{c_i k}) &= \pi_{c_i}(x_{c_i 1}, \ldots, x_{c_i, k-1}, x_{c_i, k+1}, \ldots, x_{c_i n_{c_i}}) \\
&= \frac{1}{G_{c_i}} \prod_{l=1}^{k-1} \frac{\lambda_{c_i x_{c_i l}}}{\mu_{c_i}} \cdot \prod_{l=k+1}^{n_{c_i}} \frac{\lambda_{c_i x_{c_i l}}}{\mu_{c_i}} \cdot \frac{1}{(n_{c_i}-1)!} \\
&= \pi_{c_i}(x_{c_i}) \cdot \frac{n_{c_i}\mu_{c_i}}{\lambda_{c_i x_{c_i k}}}, \qquad (3.33)
\end{aligned}
$$

$$
\begin{aligned}
\pi_{c_i}(x_{c_i} + \mathbf{r}_{c_i k}) &= \pi_{c_i}(x_{c_i 1}, \ldots, x_{c_i, k-1}, \mathbf{r}_{c_i k}, x_{c_i k}, \ldots, x_{c_i n_{c_i}}) \\
&= \frac{1}{G_{c_i}} \prod_{l=1}^{k} \frac{\lambda_{c_i x_{c_i l}}}{\mu_{c_i}} \cdot \frac{\lambda_{c_i \mathbf{r}}}{\mu_{c_i}} \cdot \prod_{l=k+1}^{n_{c_i}} \frac{\lambda_{c_i x_{c_i l}}}{\mu_{c_i}} \cdot \frac{1}{(n_{c_i}+1)!} \\
&= \pi_{c_i}(x_{c_i}) \cdot \frac{\lambda_{c_i \mathbf{r}}}{(n_{c_i}+1)\mu_{c_i}}.
\end{aligned} \tag{3.34}
$$

Together with (3.33), and (3.34), the proposed product form (3.50) can solve the global balance equations regarding state $\emptyset$ and state $x_{c_i} \neq \emptyset$ of conveyor node c_i corresponding to Figure 3.4a and 3.4b, respectively, as follows.

$$
\begin{aligned}
\text{Figure 3.4a} \Leftrightarrow \quad & \sum_{\mathbf{r} \subseteq S} \lambda_{c_i \mathbf{r}} \pi_{c_i}(\emptyset) = \mu_{c_i} \sum_{\mathbf{r} \subseteq S} \pi_{c_i}(\mathbf{r}_{c_i 1}) \\
\Leftrightarrow \quad & \frac{1}{G_{c_i}} \sum_{\mathbf{r} \subseteq S} \lambda_{c_i \mathbf{r}} = \mu_{c_i} \sum_{\mathbf{r} \subseteq S} \frac{1}{G_{c_i}} \cdot \frac{\lambda_{c_i \mathbf{r}}}{\mu_{c_i}} \cdot \frac{1}{1!} \\
\Leftrightarrow \quad & \sum_{\mathbf{r} \subseteq S} \lambda_{c_i \mathbf{r}} = \sum_{\mathbf{r} \subseteq S} \lambda_{c_i \mathbf{r}},
\end{aligned} \tag{3.35}
$$

$$
\begin{aligned}
\text{Figure 3.4b} \Leftrightarrow \left(\sum_{\mathbf{r} \subseteq S} \lambda_{c_i \mathbf{r}} + \sum_{k=1}^{n_{c_i}} \mu_{c_i} \right) \pi_{c_i}(x_{c_i}) &= \lambda_{c_i x_{c_i n_{c_i}}} \pi_{c_i}(x_{c_i} - x_{c_i n_{c_i}}) \\
&\quad + \mu_{c_i} \sum_{k=1}^{n_{c_i}+1} \sum_{\mathbf{r} \subseteq S} \pi_{c_i}(x_{c_i} + \mathbf{r}_{c_i k}) \\
\Leftrightarrow \left(\sum_{\mathbf{r} \subseteq S} \lambda_{c_i \mathbf{r}} + n_{c_i} \mu_{c_i} \right) \pi_{c_i}(x_{c_i}) &= \lambda_{c_i x_{c_i n_{c_i}}} \pi_{c_i}(x_{c_i}) \frac{n_{c_i} \mu_{c_i}}{\lambda_{c_i x_{c_i n_{c_i}}}} \\
&\quad + \mu_{c_i} \sum_{k=1}^{n_{c_i}+1} \sum_{\mathbf{r} \subseteq S} \pi_{c_i}(x_{c_i}) \frac{\lambda_{c_i \mathbf{r}}}{(n_{c_i}+1)\mu_{c_i}} \\
&= n_{c_i} \mu_{c_i} \pi_{c_i}(x_{c_i}) + (n_{c_i}+1) \pi_{c_i}(x_{c_i}) \sum_{\mathbf{r} \subseteq S} \frac{\lambda_{c_i \mathbf{r}}}{n_{c_i}+1} \\
&= n_{c_i} \mu_{c_i} \pi_{c_i}(x_{c_i}) + \pi_{c_i}(x_{c_i}) \sum_{\mathbf{r} \subseteq S} \lambda_{c_i \mathbf{r}}.
\end{aligned} \tag{3.36}
$$

Since the two equalities, (3.35) and (3.36), hold, the stand-alone conveyor node c_i is proved to admit the product-form stationary distribution $\pi_{c_i}(x_{c_i})$. □

Theorem 3.3 (Finite-server finite-capacity queue with multiple box classes)**.** *The stationary distribution of the stand-alone station $s_i \in S$ is of the following form,*

$$
\pi_{s_i}(x_{s_i}) = \frac{1}{G_{s_i}} \prod_{l=1}^{n_{s_i}} \frac{\lambda_{s_i x_{s_i l}}}{\mu_{s_i}} \cdot \frac{1}{\gamma(n_{s_i})}, \tag{3.37}
$$

with

$$
\gamma(n_{s_i}) = \begin{cases} n_{s_i}!, & \textit{if } 0 \leq n_{s_i} < d_{s_i}, \\ d_{s_i}!(d_{s_i})^{n_{s_i} - d_{s_i}}, & \textit{if } n_{s_i} \geq d_{s_i}. \end{cases} \tag{3.38}
$$

where $G_{s_i} = \sum_{x_{s_i}} \pi_{s_i}(x_{s_i})$ *is the normalization constant, provided that the queue is stable,*

$$\sum_{\mathbf{r} \subseteq S} \frac{\lambda_{s_i \mathbf{r}}}{\mu_{s_i}} < d_{s_i}, \quad \forall s_i \in S.$$

Proof. For finite-server finite-capacity station node s_i, $i = 1, \ldots, M$, the state transition diagram is demonstrated in Figure 3.5 in which the first two cases in Figures 3.5a and 3.5b are equivalent to the two cases of a infinite-server, infinite-capacity queue. In the third case shown in Figure 3.5c, when $d_{s_i} \leq n_{s_i} < d_{s_i} + q_{s_i}$, the transitions into and out of state x_{s_i} only involve boxes at position $k \in [1, d_{s_i}]$, due to the finite number of pickers. Regarding Figure 3.5d where $n_{s_i} = d_{s_i} + q_{s_i}$, the finite buffer of station s_i becomes saturated. Therefore, it is impossible to generate a state by inserting a class $\mathbf{r}$ box into any position k of state x_{s_i}, i.e., state $x_{s_i} + \mathbf{r}_{s_i k}$ does not exist.

From (3.37) and (3.38) in Theorem 3.3, the relation between the stationary distribution of node s_i at state x_{s_i} and states $x_{s_i} - x_{s_i k}$ and $x_{s_i} + \mathbf{r}_{s_i k}$ is derived as follows. The condition for k for each cases is specified in Figure 3.5.

$$\begin{aligned} \pi_{s_i}(x_{s_i} - x_{s_i k}) &= \pi_{s_i}(x_{s_i 1}, \ldots, x_{s_i, k-1}, x_{s_i, k+1}, \ldots, x_{s_i n_{s_i}}) \\ &= \frac{1}{G_{s_i}} \prod_{l=1}^{k-1} \frac{\lambda_{s_i x_{s_i l}}}{\mu_{s_i}} \cdot \prod_{l=k+1}^{n_{s_i}} \frac{\lambda_{s_i x_{s_i l}}}{\mu_{s_i}} \cdot \frac{1}{\gamma(n_{s_i} - 1)} \\ &= \pi_{s_i}(x_{s_i}) \cdot \frac{\min(d_{s_i}, n_{s_i}) \mu_{s_i}}{\lambda_{s_i x_{s_i k}}}, \end{aligned} \tag{3.39}$$

$$\begin{aligned} \pi_{s_i}(x_{s_i} + \mathbf{r}_{s_i k}) &= \pi_{s_i}(x_{s_i 1}, \ldots, x_{s_i, k-1}, \mathbf{r}_{s_i k}, x_{s_i k}, \ldots, x_{s_i n_{s_i}}) \\ &= \frac{1}{G_{s_i}} \prod_{l=1}^{k} \frac{\lambda_{s_i x_{s_i l}}}{\mu_{s_i}} \cdot \frac{\lambda_{s_i \mathbf{r}}}{\mu_{s_i}} \cdot \prod_{l=k+1}^{n_{s_i}} \frac{\lambda_{s_i x_{s_i l}}}{\mu_{s_i}} \cdot \frac{1}{\gamma(n_{s_i} + 1)} \\ &= \pi_{s_i}(x_{s_i}) \cdot \frac{\lambda_{s_i \mathbf{r}}}{\min(d_{s_i}, n_{s_i} + 1) \mu_{s_i}}. \end{aligned} \tag{3.40}$$

The global balance equations corresponding for four cases of Figure 3.5 are of the following forms.

$$(a) \qquad \sum_{\substack{\mathbf{r} \subseteq S \\ s_i \in \mathbf{r}}} \lambda_{s_i \mathbf{r}} \pi_{s_i}(\emptyset) = \mu_{s_i} \sum_{\substack{\mathbf{r} \subseteq S \\ s_i \in \mathbf{r}}} \pi_{s_i}(\mathbf{r}_{s_i 1}) \tag{3.41}$$

$$(b) \left(\sum_{\substack{\mathbf{r} \subseteq S \\ s_i \in \mathbf{r}}} \lambda_{s_i \mathbf{r}} + \sum_{k=1}^{n_{s_i}} \mu_{s_i} \right) \pi_{s_i}(x_{s_i}) = \lambda_{s_i x_{s_i n_{s_i}}} \pi_{s_i}(x_{s_i} - x_{s_i n_{s_i}}) + \mu_{s_i} \sum_{k=1}^{n_{s_i}+1} \sum_{\substack{\mathbf{r} \subseteq S \\ s_i \in \mathbf{r}}} \pi_{s_i}(x_{s_i} + \mathbf{r}_{s_i k}) \tag{3.42}$$

$$(c) \left(\sum_{\substack{\mathbf{r} \subseteq S \\ s_i \in \mathbf{r}}} \lambda_{s_i \mathbf{r}} + \sum_{k=1}^{d_{s_i}} \mu_{s_i} \right) \pi_{s_i}(x_{s_i}) = \lambda_{s_i x_{s_i n_{s_i}}} \pi_{s_i}(x_{s_i} - x_{s_i n_{s_i}}) + \mu_{s_i} \sum_{k=1}^{d_{s_i}} \sum_{\substack{\mathbf{r} \subseteq S \\ s_i \in \mathbf{r}}} \pi_{s_i}(x_{s_i} + \mathbf{r}_{s_i k}) \tag{3.43}$$

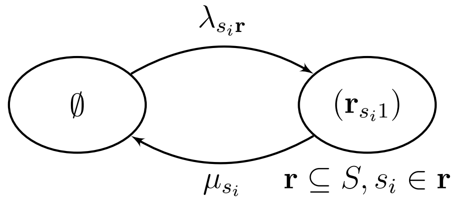
$$(d) \qquad \sum_{k=1}^{d_{s_i}} \mu_{s_i} \pi_{s_i}(x_{s_i}) = \lambda_{s_i x_{s_i n_{s_i}}} \pi_{s_i}(x_{s_i} - x_{s_i n_{s_i}}) \tag{3.44}$$

When $n_{s_i} < d_{s_i}$, $\gamma(n_{s_i}) = n_{s_i}!$, the proposed product-form stationary distribution of station s_i in the cases of (a) and (b) is similar to that of conveyor nodes. Hence, with a similar method as presented in infinite-server infinite-capacity queue, it can be proved that the proposed product-form (3.50) satisfies the global balance equations (3.41), and (3.42). By substituting (3.39) and (3.40), the equations (3.43) and (3.44) are simplified to,

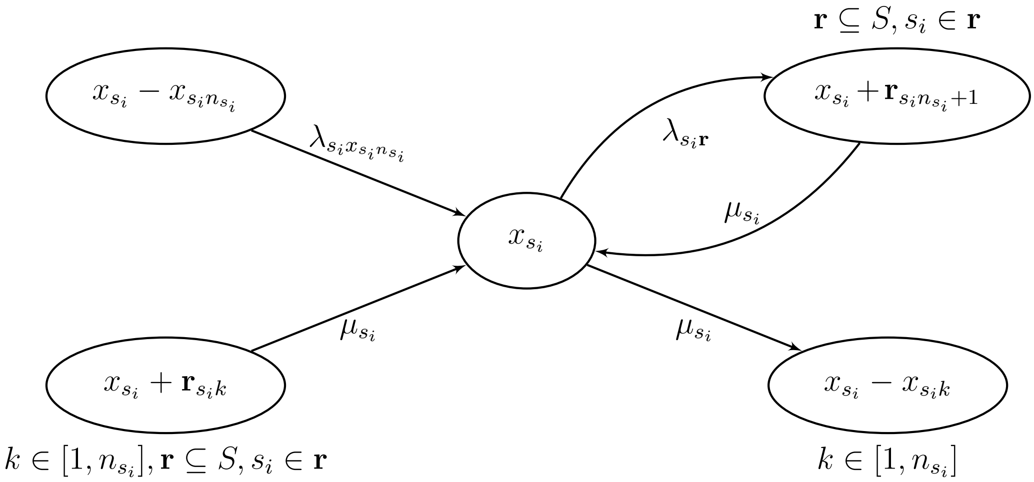
$$\begin{aligned}
(3.43) \Leftrightarrow \left(\sum_{\substack{\mathbf{r} \subseteq S \\ s_i \in \mathbf{r}}} \lambda_{s_i \mathbf{r}} + d_{s_i} \mu_{s_i} \right) \pi_{s_i}(x_{s_i}) &= \lambda_{s_i x_{s_i n_{s_i}}} \pi_{s_i}(x_{s_i}) \cdot \frac{\mu_{s_i} d_{s_i}}{\lambda_{s_i x_{s_i n_{s_i}}}} \\
&\quad + \mu_{s_i} \sum_{k=1}^{d_{s_i}} \sum_{\substack{\mathbf{r} \subseteq S \\ s_i \in \mathbf{r}}} \pi_{s_i}(x_{s_i}) \cdot \frac{\lambda_{s_i \mathbf{r}}}{\mu_{s_i} d_{s_i}} \\
&= d_{s_i} \mu_{s_i} \pi_{s_i}(x_{s_i}) + \mu_{s_i} d_{s_i} \sum_{\substack{\mathbf{r} \subseteq S \\ s_i \in \mathbf{r}}} \pi_{s_i}(x_{s_i}) \cdot \frac{\lambda_{s_i \mathbf{r}}}{\mu_{s_i} d_{s_i}} \\
&= d_{s_i} \mu_{s_i} \pi_{s_i}(x_{s_i}) + \pi_{s_i}(x_{s_i}) \sum_{\substack{\mathbf{r} \subseteq S \\ s_i \in \mathbf{r}}} \lambda_{s_i \mathbf{r}}, \qquad (3.45) \\
(3.44) \Leftrightarrow \qquad d_{s_i} \mu_{s_i} \pi_{s_i}(x_{s_i}) &= \lambda_{s_i x_{s_i n_{s_i}}} \pi_{s_i}(x_{s_i}) \frac{\mu_{s_i} d_{s_i}}{\lambda_{s_i x_{s_i n_{s_i}}}}, \qquad (3.46)
\end{aligned}$$

as in both cases $\min(d_{s_i}, n_{s_i}) = \min(d_{s_i}, n_{s_i} + 1) = d_{s_i}$.

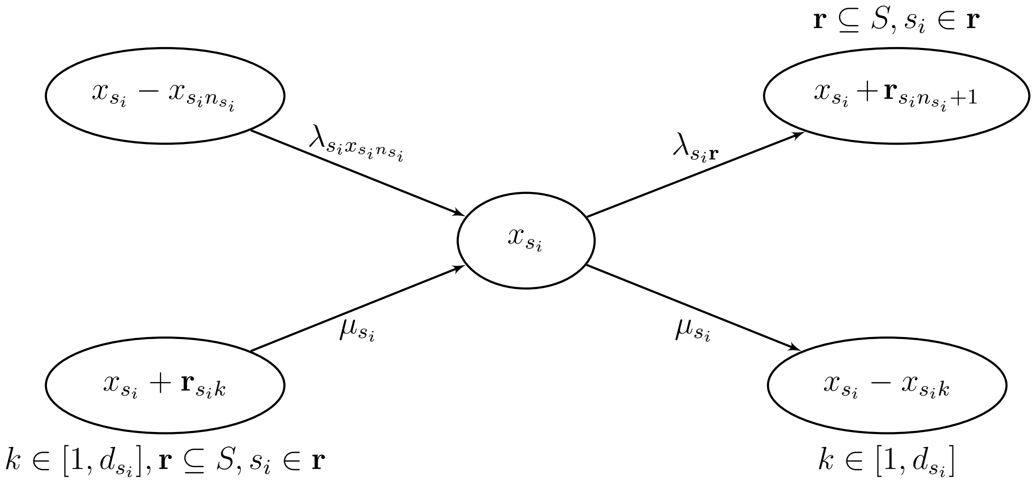
Since (3.45) and (3.46) hold, the proposed product-form equilibrium (3.50) also holds for station s_i. □



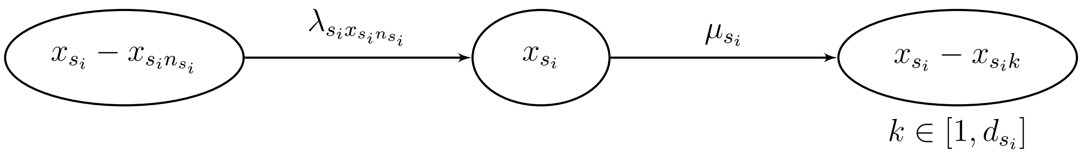
(a) Rate in and out of state $\emptyset$.



(b) Rate in and out of state $x_{s_i} \neq \emptyset$ when $n_{s_i} < d_{s_i}$.



(c) Rate in and out of state $x_{s_i} \neq \emptyset$ when $d_{s_i} \leq n_{s_i} < d_{s_i} + q_{s_i}$.



(d) Rate in and out of state $x_{s_i} \neq \emptyset$ when $n_{s_i} = d_{s_i} + q_{s_i}$.

Figure 3.5: State transition diagram of station s_i with finite servers d_{s_i} and finite buffer q_{s_i} involving a particular state x_{s_i}.

3.3.5 Stationary Distribution for Jump-over Networks

van der Gaast et al. (2020) uses Kelly's Theorem regarding the time-reversed process from Kelly (1979), stated in Lemma 3.4, to prove that the stationary distribution of the jump-over network admits the product form as proposed in Theorem 3.5. The proposed product-form distribution is combined from three theorems, 3.1, 3.2, and 3.3, developed in the previous subsection. The proof is elaborated on the computation of the time-reversed transition rates.

Lemma 3.4 (Kelly's Theorem)**.** *If there are a collection of numbers $\bar{q}(x, y)$ for $x, y \in \mathbb{S}(N)$ such that*

$$\bar{q}(x) = q(x), \quad x \in \mathbb{S}(N), \tag{3.47}$$

and a collection of positive numbers $\pi(x)$, $x \in \mathbb{S}(N)$, summing to unity, such that

$$\pi(x)q(x, y) = \pi(y)\bar{q}(y, x), \quad x, y \in \mathbb{S}(N), \tag{3.48}$$

then $\bar{q}(x, y)$ are the transition rates of the time-reversed process and $\pi(x)$ is the stationary distribution of both processes.

Theorem 3.5 (Product-form stationary distribution of jump-over networks)**.** *The stationary distribution of the single-segment jump-over network with multiple box classes possesses the following product form, under the condition that $\sum_{j \in Q} n_j = N$,*

$$\pi(x) = \frac{1}{G} \prod_{j \in Q} \pi_j(x_j) \tag{3.49}$$

where G is the normalization constant, and $\pi(x)$ and $\pi_j(x_j)$ is the stationary distribution of the system and its particular node j, or in other words, fraction of times at equilibrium the system and node j spends in state x and state x_j, respectively. $\pi_j(x_j)$ is defined as,

$$\pi_j(x_j) = \begin{cases} \dfrac{1}{G_j} \displaystyle\prod_{l=1}^{n_j} \frac{\lambda_{jx_{jl}}}{\mu_j}, & j = e, \\ \dfrac{1}{G_j} \displaystyle\prod_{l=1}^{n_j} \frac{\lambda_{jx_{jl}}}{\mu_j} \cdot \frac{1}{n_j!}, & j \in C, \\ \dfrac{1}{G_j} \displaystyle\prod_{l=1}^{n_j} \frac{\lambda_{jx_{jl}}}{\mu_j} \cdot \frac{1}{\gamma(n_j)}, & j \in S, \end{cases} \tag{3.50}$$

with

$$\gamma(n_j) = \begin{cases} n_j!, & \text{if } 0 \leq n_j < d_j, \\ d_j!(d_j)^{n_j - d_j}, & \text{if } n_j \geq d_j. \end{cases} \tag{3.51}$$

Proof. In the case of jump-over networks, the proposed collection $\pi(x)$ is stated in (3.50). van der Gaast et al. (2020) uses (3.48) to determine the rate $\bar{q}(x, y)$ and (3.47) to verify the derived time-reversed transition rates.

At the entrance/exit e, the only inflow is class $\emptyset$ boxes arriving from conveyor node c_{M+1}. From (3.48), (3.50), and (3.22), the transition rate $\bar{q}(x, x - \emptyset_{en_e} + \emptyset_{c_{M+1}l})$ for $l = 1, \ldots, n_{c_{M+1}} + 1$, provided that $n_e > 0$, can be derived as follows.

$$\begin{aligned}
\pi(x)\bar{q}(x, x - \emptyset_{en_e} + \emptyset_{c_{M+1}l}) &= \pi(x - \emptyset_{en_e} + \emptyset_{c_{M+1}l})q(x - \emptyset_{en_e} + \emptyset_{c_{M+1}l}, x) \\
&= \pi(x - \emptyset_{en_e} + \emptyset_{c_{M+1}l})q(y, y - \emptyset_{c_{M+1}l} + \emptyset_{en_e}) \\
&= \pi(x - \emptyset_{en_e} + \emptyset_{c_{M+1}l})\mu_{c_{M+1}}
\end{aligned} \tag{3.52}$$

Note that state x has n_e boxes at entrance/exit e and $n_{c_{M+1}}$ boxes at conveyor c_{M+1}; hence, state $y = x - \emptyset_{en_e} + \emptyset_{c_{M+1}l}$ has $n_e - 1$ boxes at node e and $n_{c_{M+1}} + 1$ at node c_{M+1}. The transition rate $q(y, y - \emptyset_{c_{M+1}l} + \emptyset_{en_e})$ satisfies the condition for (3.22). Based on (3.26), (3.27), (3.33), (3.34), and the traffic equation (3.12), (3.52) becomes

$$\begin{aligned}
\bar{q}(x, x - \emptyset_{en_e} + \emptyset_{c_{M+1}l}) &= \mu_{c_{M+1}} \frac{\pi(x - \emptyset_{en_e} + \emptyset_{c_{M+1}l})}{\pi(x)} \\
&= \mu_{c_{M+1}} \frac{\pi_e(x_e - \emptyset_{en_e})\pi_{c_{M+1}}(x_{c_{M+1}} + \emptyset_{c_{M+1}l})}{\pi_e(x_e)\pi_{c_{M+1}}(x_{c_{M+1}})} \\
&= \mu_{c_{M+1}} \frac{\mu_e}{\lambda_{e\emptyset}} \frac{\lambda_{c_{M+1}\emptyset}}{\mu_{c_{M+1}}(n_{c_{M+1}} + 1)} \\
&= \frac{\mu_e}{n_{c_{M+1}} + 1}.
\end{aligned} \tag{3.53}$$

In the first conveyor node c_1, class $\mathbf{r} \neq \emptyset$ boxes arrive from either the entrance/exit e or the last conveyor node c_{M+1}. With a similar argument to (3.52) and (3.53), the time-reversed transition rates involving c_1 can be obtained from the transition rates (3.17) and (3.22),

$$\begin{aligned}
\bar{q}(x, x - \mathbf{r}_{c_1 n_{c_1}} + \emptyset_{e1}) &= \mu_e \psi_{\mathbf{r}} \frac{\pi(x - \mathbf{r}_{c_1 n_{c_1}} + \emptyset_{e1})}{\pi(x)} \\
&= \mu_e \psi_{\mathbf{r}} \frac{\pi_e(x_e + \emptyset_{e1})\pi_{c_1}(x_{c_1} - \mathbf{r}_{c_1 n_{c_1}})}{\pi_e(x_e)\pi_{c_1}(x_{c_1})} \\
&= \mu_e \psi_{\mathbf{r}} \frac{\lambda_{e\emptyset}}{\mu_e} \frac{\mu_{c_1} n_{c_1}}{\lambda_{c_1\mathbf{r}}} \\
&= \frac{n_{c_1}\mu_{c_1}}{\lambda_{c_1\mathbf{r}}} \lambda_{e\emptyset}\psi_{\mathbf{r}},
\end{aligned} \tag{3.54}$$

and for $l = 1, \ldots, n_{c_{M+1}} + 1$,

$$\begin{aligned}
\bar{q}(x, x - \mathbf{r}_{c_1 n_{c_1}} + \mathbf{r}_{c_{M+1}l}) &= \mu_{c_{M+1}} \frac{\pi(x - \mathbf{r}_{c_1 n_{c_1}} + \mathbf{r}_{c_{M+1}l})}{\pi(x)} \\
&= \mu_{c_{M+1}} \frac{\pi_{c_{M+1}}(x_{c_{M+1}} + \mathbf{r}_{c_{M+1}l})\pi_{c_1}(x_{c_1} - \mathbf{r}_{c_1 n_{c_1}})}{\pi_e(x_e)\pi_{c_1}(x_{c_1})} \\
&= \mu_{c_{M+1}} \frac{\lambda_{c_{M+1}\mathbf{r}}}{\mu_{c_{M+1}}(n_{c_{M+1}} + 1)} \frac{\mu_{c_1} n_{c_1}}{\lambda_{c_1\mathbf{r}}} \\
&= \frac{n_{c_1}\mu_{c_1}}{\lambda_{c_1\mathbf{r}}} \frac{\lambda_{c_{M+1}\mathbf{r}}}{n_{c_{M+1}} + 1}.
\end{aligned} \tag{3.55}$$

Conveyor node $c_{i+1},\ i=1,\ldots,M$, has the inflow of class $\mathbf{r}$ boxes from both c_i and s_i. There are five possible scenarios as follows.

(i) Class $\mathbf{r}$, $s_i \notin \mathbf{r}$, boxes arrive from c_i and do not need to visit station s_i, then by the transition rate (3.18), for $l=1,\ldots,n_{c_i}+1$,

$$\begin{aligned}\bar{q}(x, x-\mathbf{r}_{c_{i+1}n_{c_{i+1}}}+\mathbf{r}_{c_i l}) &= \mu_{c_i}\frac{\pi(x-\mathbf{r}_{c_{i+1}n_{c_{i+1}}}+\mathbf{r}_{c_i l})}{\pi(x)}\\ &= \mu_{c_i}\frac{\pi_{c_i}(x_{c_i}+\mathbf{r}_{c_i l})\pi_{c_{i+1}}(x_{c_{i+1}}-\mathbf{r}_{c_{i+1}n_{c_{i+1}}})}{\pi_{c_i}(x_{c_i})\pi_{c_{i+1}}(x_{c_{i+1}})}\\ &= \mu_{c_i}\frac{\lambda_{c_i\mathbf{r}}}{\mu_{c_i}(n_{c_i}+1)}\frac{\mu_{c_{i+1}}n_{c_{i+1}}}{\lambda_{c_{i+1}\mathbf{r}}}\\ &= \frac{n_{c_{i+1}}\mu_{c_{i+1}}}{\lambda_{c_{i+1}\mathbf{r}}}\frac{\lambda_{c_i\mathbf{r}}}{n_{c_i}+1}. \qquad (3.56)\end{aligned}$$

(ii) $s_i \notin \mathbf{r}$ and class $\mathbf{r}\cup\{s_i\}$ boxes attempt to visit station s_i when the queue is full $n_{s_i}=d_{s_i}+q_{s_i}$. With probability $1-b_{s_i}$, the boxes arrive at c_{i+1} from c_i with its class changing to $\mathbf{r}$. By the transition rate (3.20), for $l=1,\ldots,n_{c_i}+1$,

$$\begin{aligned}\bar{q}\Big(x, x-\mathbf{r}_{c_{i+1}n_{c_{i+1}}}+\mathbf{r}\cup\{s_i\}_{c_i l}\Big) &= \mu_{c_i}(1-b_{s_i})\frac{\pi\Big(x-\mathbf{r}_{c_{i+1}n_{c_{i+1}}}+\mathbf{r}\cup\{s_i\}_{c_i l}\Big)}{\pi(x)}\\ &= \mu_{c_i}(1-b_{s_i})\frac{\pi_{c_i}\Big(x_{c_i}+\mathbf{r}\cup\{s_i\}_{c_i l}\Big)\pi_{c_{i+1}}(x_{c_{i+1}}-\mathbf{r}_{c_{i+1}n_{c_{i+1}}})}{\pi_{c_i}(x_{c_i})\pi_{c_{i+1}}(x_{c_{i+1}})}\\ &= \mu_{c_i}(1-b_{s_i})\frac{\lambda_{c_i\mathbf{r}\cup\{s_i\}}}{\mu_{c_i}(n_{c_i}+1)}\frac{\mu_{c_{i+1}}n_{c_{i+1}}}{\lambda_{c_{i+1}\mathbf{r}}}\\ &= \frac{n_{c_{i+1}}\mu_{c_{i+1}}}{\lambda_{c_{i+1}\mathbf{r}}}\frac{\lambda_{c_i\mathbf{r}\cup\{s_i\}}}{n_{c_i}+1}(1-b_{s_i}). \qquad (3.57)\end{aligned}$$

(iii) With $s_i \notin \mathbf{r}$, class $\mathbf{r}\cup\{s_i\}$ boxes enter the buffer of station s_i if $n_{s_i} < d_{s_i}+q_{s_i}$. With probability $1-b_{s_i}$, the boxes arrive at c_{i+1} from s_i with its class changing to $\mathbf{r}$. By the transition rate (3.21), for $l=1,\ldots,\min(d_i, n_{s_i}+1)$,

$$\begin{aligned}\bar{q}\Big(x, x-\mathbf{r}_{c_{i+1}n_{c_{i+1}}}+\mathbf{r}\cup\{s_i\}_{s_i l}\Big) &= \mu_{s_i}(1-b_{s_i})\frac{\pi\Big(x-\mathbf{r}_{c_{i+1}n_{c_{i+1}}}+\mathbf{r}\cup\{s_i\}_{s_i l}\Big)}{\pi(x)}\\ &= \mu_{s_i}(1-b_{s_i})\frac{\pi_{s_i}\Big(x_{s_i}+\mathbf{r}\cup\{s_i\}_{s_i l}\Big)\pi_{c_{i+1}}(x_{c_{i+1}}-\mathbf{r}_{c_{i+1}n_{c_{i+1}}})}{\pi_{s_i}(x_{s_i})\pi_{c_{i+1}}(x_{c_{i+1}})}\\ &= \mu_{s_i}(1-b_{s_i})\frac{\lambda_{s_i\mathbf{r}\cup\{s_i\}}}{\mu_{s_i}\min(d_{s_i}, n_{s_i}+1)}\frac{\mu_{c_{i+1}}n_{c_{i+1}}}{\lambda_{c_{i+1}\mathbf{r}}}\\ &= \frac{n_{c_{i+1}}\mu_{c_{i+1}}}{\lambda_{c_{i+1}\mathbf{r}}}\frac{\lambda_{s_i\mathbf{r}\cup\{s_i\}}}{\min(d_{s_i}, n_{s_i}+1)}(1-b_{s_i}). \qquad (3.58)\end{aligned}$$

(iv) Boxes with class $\mathbf{r}, s_i \in \mathbf{r}$ can arrive at c_{i+1} from c_i when $n_{s_i}=d_{s_i}+q_{s_i}$ with blocking probability b_{s_i}. With a similar argument to *(ii)*, for $l=1,\ldots,n_{c_i}+1$,

$$\bar{q}\Big(x, x-\mathbf{r}_{c_{i+1}n_{c_{i+1}}}+\mathbf{r}_{c_i l}\Big) = \frac{n_{c_{i+1}}\mu_{c_{i+1}}}{\lambda_{c_{i+1}\mathbf{r}}}\frac{\lambda_{c_i\mathbf{r}}}{n_{c_i}+1}b_{s_i}. \qquad (3.59)$$

(v) In case of $n_{s_i} < d_{s_i} + q_{s_i}$ and $s_i \in \mathbf{r}$, class $\mathbf{r}$ boxes can arrive at c_{i+1} from s_i with blocking probability b_{s_i}. Similarly to *(iii)*, for $l = 1, \dots, \min(d_{s_i}, n_{s_i} + 1)$,

$$\bar{q}\Big(x, x - \mathbf{r}_{c_{i+1}n_{c_{i+1}}} + \mathbf{r}_{s_i l}\Big) = \frac{n_{c_{i+1}}\mu_{c_{i+1}}}{\lambda_{c_{i+1}\mathbf{r}}} \frac{\lambda_{s_i\mathbf{r}}}{\min(d_{s_i}, n_{s_i} + 1)} b_{s_i}. \tag{3.60}$$

Finally, in station s_i, class $\mathbf{r}$ boxes arrive from conveyor c_i if $s_i \in \mathbf{r}$ and $n_{s_i} < d_{s_i} + q_{s_i}$. By the transition rate (3.19), for $l = 1, \dots, n_{c_i} + 1$,

$$\begin{aligned}
\bar{q}\Big(x, x - \mathbf{r}_{s_i n_{s_i}} + \mathbf{r}_{c_i l}\Big) &= \mu_{c_i} \frac{\pi(x - \mathbf{r}_{s_i n_{s_i}} + \mathbf{r}_{c_i l})}{\pi(x)} \\
&= \mu_{c_i} \frac{\pi_{c_i}(x_{c_i} + \mathbf{r}_{c_i l})\pi_{s_i}(x_{s_i} - \mathbf{r}_{s_i n_{s_i}})}{\pi_{c_i}(x_{c_i})\pi_{s_i}(x_{s_i})} \\
&= \mu_{c_i} \frac{\lambda_{c_i\mathbf{r}}}{\mu_{c_i}(n_{c_i} + 1)} \frac{\mu_{s_i}\min(d_{s_i}, n_{s_i})}{\lambda_{s_i\mathbf{r}}} \\
&= \frac{\lambda_{c_i\mathbf{r}}}{n_{c_i} + 1} \frac{\min(d_{s_i}, n_{s_i})\mu_{s_i}}{\lambda_{s_i\mathbf{r}}} \\
&= \frac{\min(d_{s_i}, n_{s_i})\mu_{s_i}}{n_{c_i} + 1}
\end{aligned} \tag{3.61}$$

in which the last equality is derived from the traffic equation (3.16).

This completes the description of the nonzero time-reversed transition rates $\bar{q}(x, y)$. The last step of the proof is to prove that

$$\bar{q}(x) = q(x) = \mu_e \mathbb{1}_{(n_e > 0)} + \sum_{i=1}^{M+1} n_{c_i}\mu_{c_i} + \sum_{i=1}^{M} \min(d_{s_i}, n_{s_i})\mu_{s_i}. \tag{3.62}$$

The first term of the sum corresponds to the total time-reversed transition rate of events in x occurring in the entrance/exit e provided that $n_e > 0$,

$$\sum_{l=1}^{n_{c_{M+1}}+1} \bar{q}(x, x - \emptyset_{e n_e} + \emptyset_{c_{M+1} l}) \mathbb{1}_{(n_e > 0)} = \sum_{l=1}^{n_{c_{M+1}}+1} \frac{\mu_e}{n_{c_{M+1}} + 1} \mathbb{1}_{(n_e > 0)} = \mu_e \mathbb{1}_{(n_e > 0)}.$$

Regarding the second term of the sum in (3.62), for $i = 1$, the total rate of events in x coming from c_1 is equal to

$$\begin{aligned}
&\bar{q}(x, x - \mathbf{r}_{c_1 n_{c_1}} + \emptyset_{e1}) + \sum_{l=1}^{n_{c_{M+1}}+1} \bar{q}(x, x - \mathbf{r}_{c_1 n_{c_1}} + \mathbf{r}_{c_{M+1} l}) \\
&= \frac{n_{c_1}\mu_{c_1}}{\lambda_{c_1\mathbf{r}}} \lambda_{e\emptyset}\psi_{\mathbf{r}} + \sum_{l=1}^{n_{c_{M+1}}+1} \frac{n_{c_1}\mu_{c_1}}{\lambda_{c_1\mathbf{r}}} \frac{\lambda_{c_{M+1}\mathbf{r}}}{n_{c_{M+1}} + 1} \\
&= \frac{n_{c_1}\mu_{c_1}}{\lambda_{c_1\mathbf{r}}} \Big(\lambda_{e\emptyset}\psi_{\mathbf{r}} + \lambda_{c_{M+1}\mathbf{r}}\Big) \\
&= n_{c_1}\mu_{c_1},
\end{aligned}$$

in which the last equality follows from the traffic equation (3.13). For $i = 1, \dots, M$, if $s_i \notin \mathbf{r}$, it follows from (3.56)–(3.58) and traffic equations (3.15) and (3.16) that the accumulated time-reversed rate of events resulting from c_{i+1} is

$$
\begin{aligned}
&\sum_{l=1}^{n_{c_i}+1} \bar{q}(x, x - \mathbf{r}_{c_{i+1}n_{c_{i+1}}} + \mathbf{r}_{c_i l})\mathbb{1}_{(s_i \notin \mathbf{r})} \\
&\quad + \sum_{l=1}^{n_{c_i}+1} \bar{q}\Big(x, x - \mathbf{r}_{c_{i+1}n_{c_{i+1}}} + \mathbf{r} \cup \{s_i\}_{c_i l}\Big)\mathbb{1}_{(s_i \notin \mathbf{r})}\mathbb{1}_{(n_{s_i} = d_{s_i} + q_{s_i})} \\
&\quad + \sum_{l=1}^{\min(d_{s_i}, n_{s_i}+1)} \bar{q}\Big(x, x - \mathbf{r}_{c_{i+1}n_{c_{i+1}}} + \mathbf{r} \cup \{s_i\}_{s_i l}\Big)\mathbb{1}_{(s_i \notin \mathbf{r})}\mathbb{1}_{(n_{s_i} < d_{s_i} + q_{s_i})} \\
&= \mathbb{1}_{(s_i \notin \mathbf{r})}\mathbb{1}_{(n_{s_i} = d_{s_i} + q_{s_i})}\frac{n_{c_{i+1}}\mu_{c_{i+1}}}{\lambda_{c_{i+1}\mathbf{r}}}\Big(\lambda_{c_i \mathbf{r}} + \lambda_{c_i \mathbf{r} \cup \{s_i\}}(1 - b_{s_i})\Big) \\
&\quad + \mathbb{1}_{(s_i \notin \mathbf{r})}\mathbb{1}_{(n_{s_i} < d_{s_i} + q_{s_i})}\frac{n_{c_{i+1}}\mu_{c_{i+1}}}{\lambda_{c_{i+1}\mathbf{r}}}\Big(\lambda_{c_i \mathbf{r}} + \lambda_{s_i \mathbf{r} \cup \{s_i\}}(1 - b_{s_i})\Big) \\
&= \mathbb{1}_{(s_i \notin \mathbf{r})}\mathbb{1}_{(n_{s_i} = d_{s_i} + q_{s_i})}n_{c_{i+1}}\mu_{c_{i+1}} + \mathbb{1}_{(s_i \notin \mathbf{r})}\mathbb{1}_{(n_{s_i} < d_{s_i} + q_{s_i})}n_{c_{i+1}}\mu_{c_{i+1}} \\
&= \mathbb{1}_{(s_i \notin \mathbf{r})}n_{c_{i+1}}\mu_{c_{i+1}}.
\end{aligned}
$$

Similarly, if $s_i \in \mathbf{r}$, the rate of events coming from c_{i+1} is equal to $\mathbb{1}_{(s_i \in \mathbf{r})}n_{c_{i+1}}\mu_{c_{i+1}}$, which implies that the total time-reversed transition rate of these particular events can be simplified to $n_{c_{i+1}}\mu_{c_{i+1}}$. Therefore, the second term of the sum is the combined rate of events in state x coming from any conveyor node c_i for $i = 1, \ldots, M+1$.

Lastly, the third term of the sum in (3.62) involves the time-reversed transition rate of events in state x resulting from station s_i, $i = 1, \ldots, M$. This can be proved by using (3.61) and the traffic equation (3.16),

$$
\sum_{l=1}^{n_{c_i}+1} \bar{q}(x, x - \mathbf{r}_{s_i n_{s_i}} + \mathbf{r}_{c_i l}) = \sum_{l=1}^{n_{c_i}+1} \frac{\min(d_{s_i}, n_{s_i})\mu_{s_i}}{n_{c_i} + 1} = \min(d_{s_i}, n_{s_i})\mu_{s_i}.
$$

All in all, the total time-reversed transition rate $\bar{q}(x)$ is proved to be equal to $q(x)$; hence, the jump-over network admits the product-form stationary distribution (3.50) according to Lemma 3.4 or Kelly's Theorem. □

The key performance statistics of the network, such as the mean throughput time and chain visit ratios, can be obtained analytically or iteratively using Mean Value Analysis algorithm as presented in van der Gaast et al. (2020). The authors demonstrated that jump-over networks can more accurately simulate the performance of block-and-recirculate networks than previous infinite-capacity approximation models introduced by de Koster (1994), Yu and de Koster (2008), and Melacini, Perotti, and Tumino (2011). To accommodate the fact that one warehouse can have numerous floors, jump-over networks with multiple segments are set up using similar methodology as the single-segment networks with each segment acting as a station on the main conveyor.

4 OPEN QUEUING NETWORKS WITH BLOCK-AND-RECIRCULATE AND MULTIPLE SEGMENTS

This chapter studies the order picking process in the shelving area of Kramp's warehouse in Strullendorf, Germany. Due to the external box arrival to the shelving area, the open queuing networks resembling this OPS is intractable. Therefore, a simulation is built to replicate the network, as well as to integrate the warehouse layout into the service time at a station. Section 4.1 describes the network with three segments corresponding to three floors. The shelving area and the block-and-recirculate policy are first explained, followed by the description of how the conveyor connects each segment in 4.1.1. The next subsection examines the station layout and storage assignment, which are not taken into account in the previous chapter. Afterwards, the picker characteristics, especially the batch servicing and routing strategy, are specified in 4.1.3. The last subsection lists five key performance indicators (KPIs), or performance statistics, for the queuing network. Based on the information of the queuing network, Section 4.2 develops an event-based simulation in which the data model and event relationship diagram are explained in 4.2.1 and 4.2.2, respectively. Moreover, 4.2.3 carries an exploratory data analysis on relations among the KPIs and describes how the service time on the conveyor, i.e., conveyor time, is modelled based on the historical data. Finally, two simulation scenarios, with and without stock splitting, are executed in 4.2.4 and their results are compared to the current warehouse performance. Scenario 2 without stock splitting achieves better results and its storage assignment is chosen as the base to integrate the frequent item pairs in the next chapter.

4.1 Queuing Network Description

In order to accommodate the external arrival process and the warehouse layout into the queuing networks from Chapter 3, the order picking system in the shelving area of Kramp's warehouse in Strullendorf is chosen as a base for the simulation.

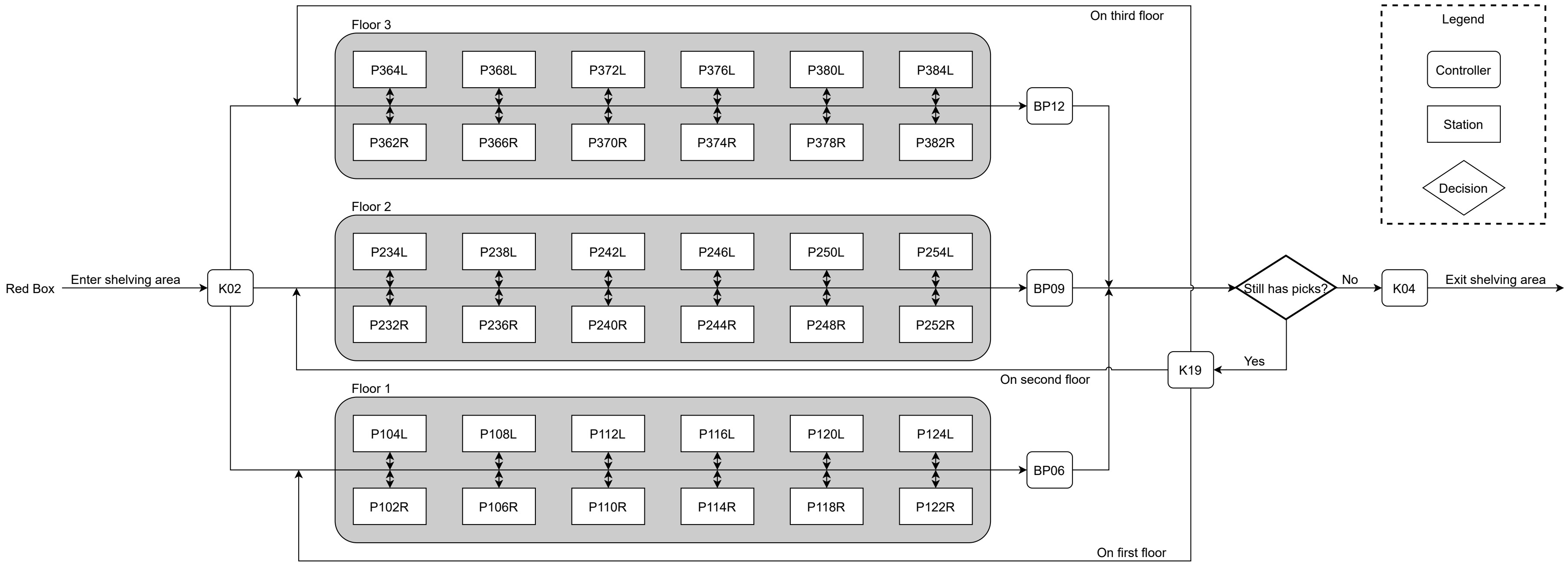


Figure 4.1: Conveyor system in the shelving area with the main controllers to determine box route in the system. When a box passes a main controller, its information and passing timestamp are logged in the database. Three end floor controllers, BP06, BP09, and BP12, have a scale to check whether a box has reached its target weight with all picks yet.

4.1.1 Multiple Segments and Block-and-Recirculate Policy

In the Strullendorf warehouse, the shelving area spans three floors, each floor is considered as a segment and consists of twelve stations. The stations on a floor are divided into two groups: six to the right of the conveyor and six to the left. Station identification starts with the letter P, followed by the floor number and a unique station identifier on the floor, and ends with letter R or L to indicate whether it is located on the right or left of the conveyor.

The conveyor system in the shelving area is demonstrated in Figure 4.1, in which the arrows represent the conveyor direction. The warehouse management system, hereby referred to as WMS, assigns the orders to boxes. Each box contains one or more items of a particular order. In the case of a large order, the items can be distributed into several boxes. A box can contain items from multiple orders if and only if these orders are placed by the same customer. An order is classified as express if it requires same-day or overnight delivery; otherwise, it is considered as a normal order. If the order is express/normal, its assigning boxes are classified as express/normal correspondingly.

A box enters the shelving area by passing the start controller K02. Here WMS determines which floor the box should be directed towards according to its assigned picks. If the box needs picks from multiple floors, there is an externally provided algorithm in place to set the floor sequence. Since the algorithm is a part of the purchased WMS service, it is a black box to Kramp. The box is then transported to the first floor in the sequence. At the first corresponding left and right stations, or twin stations, there is a controller to determine whether the box needs picks from one station, both or neither. Regarding the first case, the box is directed to a buffer of the necessary station. If the buffer is full, the box skips the station and continues on the conveyor. In the second case, the box enters a buffer of a randomly chosen station. If the buffer of said station is full, the controller checks if the buffer of the other station is also full. If yes, the box continues on the conveyor; otherwise, it enters a buffer of the other station. After the picks are completed, the box is deposited back to the conveyor belt before the buffer entrances so it can be assigned to the other station in the pair if necessary. In the case that neither are needed, the box moves without entering either stations. In all cases, the box then travels to the next set of stations and the same procedure applies for all twin stations on the floor.

Once the box reaches the end of the floor, an end floor controller has a scale to check whether the box's current weight matches its target weight with all picks yet. These controllers are called BP06, BP09, and BP12 respectively for the first, second, and third floor. If yes, the box is directed to the end controller K04 and exits the system. Otherwise, it must recirculate to the immediate controller K19, which uses the same algorithm as controller K02 to redirect the box to the required floor. From there, the procedure repeats until all the picks are performed and the box exits the shelving area. During peak periods, a box can recirculate multiple times. The control manager can manually force the box to its blocked station and ask the picker to immediately retrieve items for the box. The start controller K02, immediate controller K19, end controller K04, and three end floor controllers BP06, BP09, and BP12 form the main controllers in the system. Whenever a box passes one of these controllers, its information with the passing timestamp is logged in the database.

The comparison between the OPS at Strullendorf warehouse and the closed network in Chapter 3 is drawn as follows.

- Both order picking processes are manual picker-to-parts with a conveyor system.
- The block-and-recirculate policy of the open queuing network is similar to that of the closed network as in Figure 3.1.
- Instead of using a joint system entrance/exit e, the open networks adopts controller K02 as the system entrance and K04 as the system exit to accommodate the box arrival and departure processes.
- Instead of a cyclic structure, the open network employs a star structure where K02/K19 acts as a central hub. As demonstrated in Figure 4.2, the star structure, together with the weight check at the end of each floor, helps reduce a box's travel time when: (i) the box is blocked from one station, it can visit not only other required stations on the same floor but also stations on the other floors before attempting to reenter the congested station; (ii) the box can exit the shelving area from the end floor controller without reverting back to the entrance/exit.

4.1.2 Station Layout and Storage Assignment

Each station has two buffers, namely normal and express with the capacities of 10 and 6 boxes respectively, to contain the corresponding box type. The station layout admits the single-block form with eight main aisles perpendicular to the conveyor, and two cross aisles, front and back, parallel to the conveyor. The station layout for a pair of twin stations is illustrated in Figure 4.3. Each main aisle has two shelves, one on the left and one on the right. The right shelf on one aisle stands against the left shelf on the next aisle. In the right stations, each shelf consists of 15 modules of size 40cm x 120cm, numbered AA and from 0 to 13. At the conveyor end of the back-to-back shelves

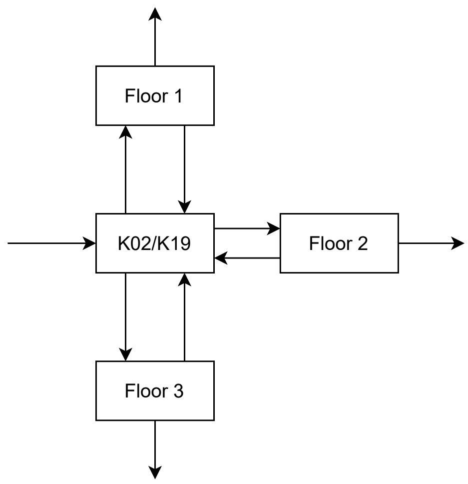


Figure 4.2: The current warehouse layout as a star network.

stands a smaller module numbered 15 of size 40cm x 80cm parallel to the conveyor. In the left stations, each shelf consists of 14 modules, numbered from 17 to 30. There are an additional 10 modules numbered 31 standing against the back wall. All the modules in the left stations are of size 40cm x 120cm. Each module has several shelf spaces in various LHM types starting with the character M, e.g., M01 or M03. LHM types are defined based on the dimension of the shelf spaces. One or more LHM types are also assigned to items to indicate which shelf spaces are compatible to store the items. Depending on the floor structure, several modules may be missing due to the pillars, stairs, or the conveyor setup. The full representation of a floor is plotted in the Appendix A.

The current storage assignment follows the closest open location storage, i.e., a new item is assigned to the open shelf space which is closest to the conveyor and matches its LHM type(s). For an item already present in the warehouse, its current shelf space is filled to its maximum capacity first. If there is still some quantity left, closest open location storage is then applied. Therefore, an item can be stored in multiple shelf spaces in one or more stations. When determining which station(s) a box needs picks from, WMS attempts to minimize the number of spaces taken up by one item. This implies that the box is assigned to have its items retrieved from the shelf space with the least stock in the case of multiple shelf spaces. Hence, the box may need to visit several stations to retrieve the sufficient quantity for an item. Moreover, a box with two items may stop at two stations although there is one common station in the items' storage locations.

The station layout and storage assignment are not taken into account in the closed queuing networks in Chapter 3. With these integrated into the open networks, the box service time at one station can be estimated more accurately based on the true location of the pick items instead of being drawn from an exponential distribution. The service time calculation will be elaborated in the next subsection.

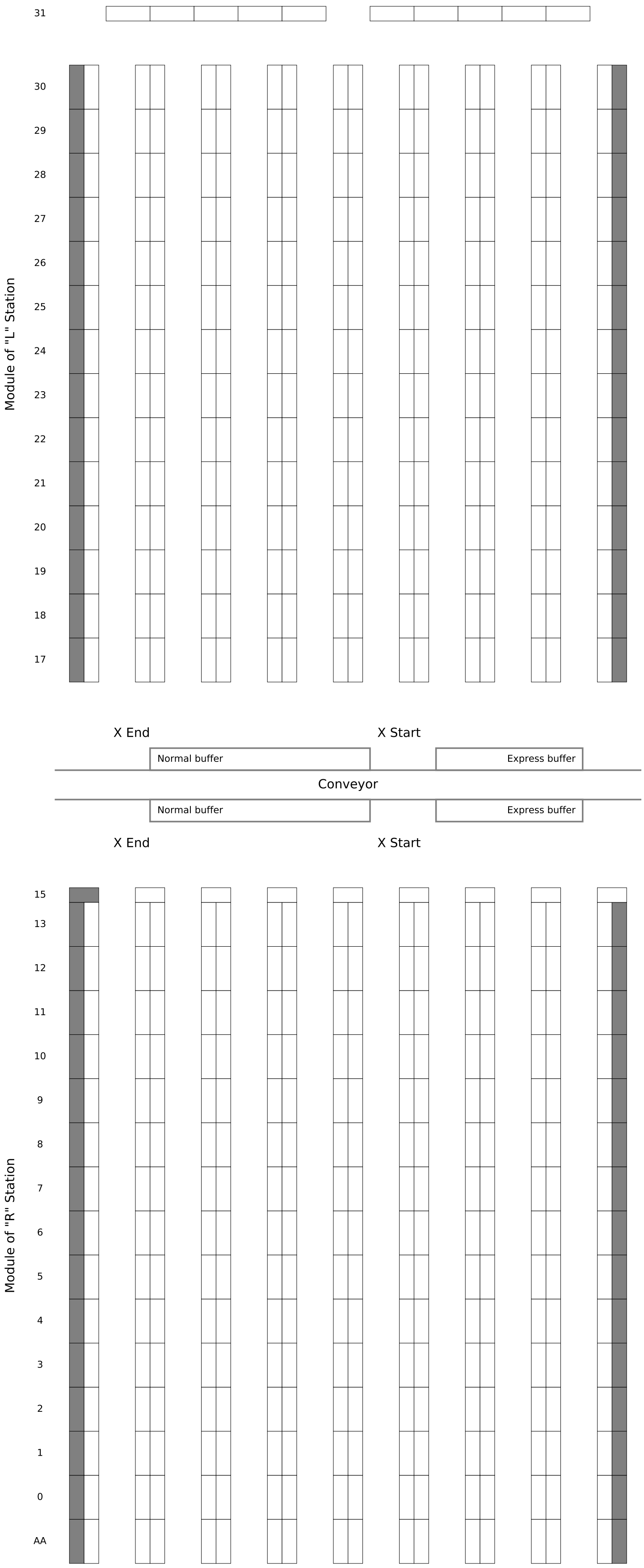


Figure 4.3: Station representation with “X Start” and “X End” the start and end points of a pick tour, respectively. The grey modules belong to the adjacent stations.

4.1.3 Picker and Routing Strategy

Since order picking is an outbound process, only outbound pickers who are responsible for item retrieval participate in the simulation. The number of pickers varies from 30 to 39 per day, and the first outbound pickers start their shift at around 10:45. Order picking in the morning from 08:00 to 10:45 is handled by the inbound pickers, i.e., pickers whose main responsibility is to distribute the items to the corresponding shelf spaces. A picker does not have a designated station but continuously moves to other stations if their current station has no boxes in the buffers. However, one station can have maximum one present picker to prevent aisle blocking. Figure 4.4 visualizes the working environment of a picker, as well as the station layout under a picker's perspective. Each picker is accompanied by a commission cart which can accommodate maximum five boxes and a scanner to access the information about the box's items with the corresponding pick quantity and location. The commission cart allows pickers to perform batch servicing, which potentially reduces their travel time especially in the case of common items or items with close proximity among boxes of the same pick tour. In addition to the FIFO queue discipline, the open queuing network employs a queue priority due to two types of boxes. Pickers choose express boxes over the normal ones for a pick tour and two types of box cannot be combined in one tour. The batch servicing and the box priority are two additional features for the box service in the open network as compared to the closed one.

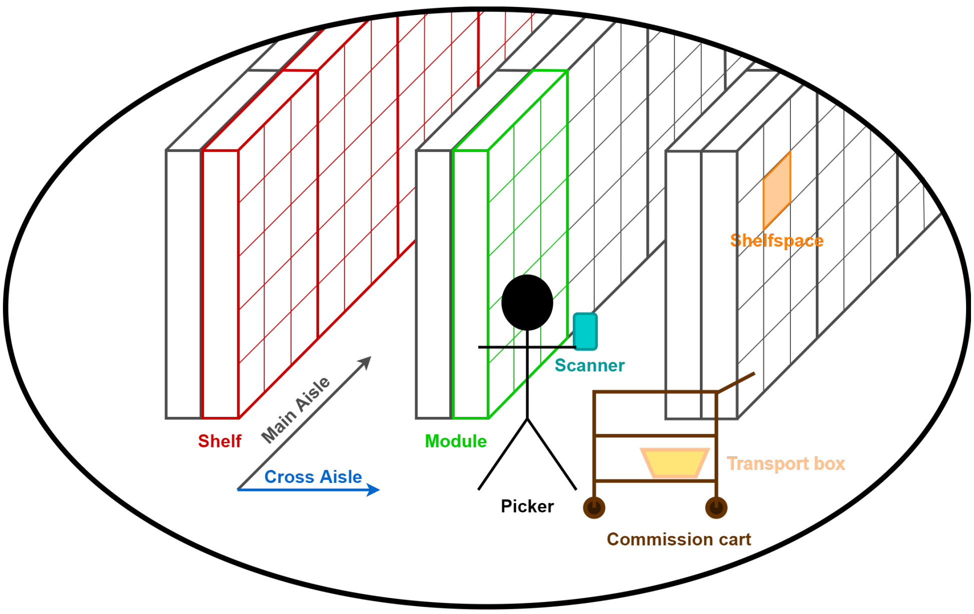


Figure 4.4: Working environment of a picker in the shelving area.

Regarding the routing policy, from the start point of a station, the picker travels down the aisle to the first pick location, retrieves the items, then determines whether they should return to the front cross aisle or take an U-turn at the back cross aisle to reach the next pick location. The procedure continues until all the picks are performed.

From the last visited aisle, the picker walks to the end point to drop the boxes onto the conveyor. The literature survey in Figure 2.2 suggests that instead of implementing the optimal travel time calculation, return and S-shape heuristics are suitable substitutions for analysis due to less computational cost. In the simulation, return routing is chosen due to the closest open location storage strategy. Since shelf spaces near the conveyor are likely to be occupied while those towards the back wall are more empty, it is not necessary to travel all the way to the end of the aisle. However, there is a chance that a frequently purchased item is located near the back cross aisle, which implies that the return routing is only a sub-optimal strategy.

To determine the service time for a particular box in one station, the approach from Melacini, Perotti, and Tumino (2011) is adopted. The box service time is decomposed into four components. The parameters for the components are estimated through the order picking observation in the warehouse,

- the constant setup time per box of 11 seconds,
- the travel time with station layout and storage assignment from the previous subsection, return routing policy, and constant walking speed of 200 centimeters per second,
- the constant picking time per item of 21 seconds, regardless of the quantity,
- the constant post processing time per box of 8 seconds, as a picker needs to put the items into the corresponding box on the commission cart.

4.1.4 Key Performance Indicators (KPIs)

Instead of studying the intractable stationary distribution for the open network, the simulation focuses on the warehouse performance. The following notation is employed.

- N: the number of simulated boxes.
- $B = \{1, \ldots, b, \ldots, N\}$: the set of boxes, indexed by b.
- I: the set of items stored in the shelving area, indexed by i as the item number.
- S: the set of 36 stations specified in Figure 4.1, indexed by s as the station identification.
- $x_{ib},\ i \in I,\ b \in B$: 1 if item i is required for box b; 0 otherwise.

- $y_{is},\ i \in I,\ s \in S$: 1 if item i is stored in station s; 0 otherwise. If the item has not been stored in the warehouse yet, $\sum_{s\in S} y_{is} = 0$. Otherwise, items with stock splitting implies $1 \leq \sum_{s\in S} y_{is} \leq 36$ while in the case of no stock splitting, $\sum_{s\in S} y_{is} = 1$.

- $z_{ibs},\ i \in I,\ b \in B,\ s \in S$: 1 if box b visits station s to pick item i; 0 otherwise. With stock splitting, $z_{ibs} \leq x_{ib}y_{is}$; otherwise, $z_{ibs} = x_{ib}y_{is}$.

- $PpB_b,\ b \in B$: the number of picks for box b. Assume that there are no pick errors, i.e., a picker is able to pick the correct quantity for an item on the first attempt,

$$PpB_b = \sum_{i\in I}\sum_{s\in S} z_{ibs}.$$

- $SpB_b,\ b \in B$: the number of stops for box b, or the number of stations box b needs to visit,

$$SpB_b = \sum_{s\in S} \mathbb{1}_{\left(\sum_{i\in I} z_{ibs} \geq 1\right)},$$

where $\mathbb{1}_{(.)}$ is the indicator function.

- $PpS_b,\ b \in B$: the average number of picks per stop/station for box b,

$$PpS_b = \frac{PpB_b}{SpB_b}.$$

- $TpB_b,\ b \in B$: the throughput time for box b, i.e., the number of seconds box b spends in the system, starting from entering controller K02 to leaving controller K04.

- $BQpB_b,\ b \in B$: the number of blocked queues box b encounters in the system. A box can be blocked from one queue multiple times.

- $RpB_b,\ b \in B$: the number of floor recirculations box b makes in the system. One recirculation is accummulated every time a box is redirected to a floor which has been visited before. For example, if a box's floor route is $\{1, 3, 2, 2, 3, 2\}$, its RpB equals three, two for the second floor and one for the third floor.

Since a box must return to a floor if it encounters one or more blocked queues on that floor, this implies that RpB_b and $BQpB_b$ increase simultaneously with the amount of increase in $BQpB_b$ always equal to or higher than that in RpB_b. Therefore, $RpB_b \leq BQpB_b$, $\forall b = 1, \ldots, N$.

Five key performance indicators (KPIs) are of interest in the simulation. Their definition and formula are listed as follows.

- $aSpB$: average stops per box over N boxes,

$$aSpB = \frac{1}{N}\sum_{b=1}^{N} SpB_b.$$

- $aPpS$: average picks per stop over N boxes,

$$aPpS = \frac{\sum_{b=1}^{N} PpB_b}{\sum_{b=1}^{N} SpB_b}.$$

  It can be derived that $aSpB \cdot aPpS = \frac{1}{N}\sum_{b=1}^{N} PpB_b$. Therefore, if PpB_b remains unchanged, increasing $aSpB$ implies a reduction in $aPpS$.

- $aTpB$: average throughput time per box over N boxes,

$$aTpB = \frac{1}{N}\sum_{b=1}^{N} TpB_b. \tag{4.1}$$

- $aBQpB$: average blocked queues per box over N boxes,

$$aBQpB = \frac{1}{N}\sum_{b=1}^{N} BQpB_b. \tag{4.2}$$

- $aRpB$: average recirculation per box over N boxes,

$$aRpB = \frac{1}{N}\sum_{b=1}^{N} RpB_b. \tag{4.3}$$

Since $RpB_b \leq BQpB_b,\ \forall b = 1, \ldots, N$, it implies that $aRpB$ is also smaller than or equal to $aBQpB$.

4.2 Simulation Model

In order to develop a simulation model, the relevant data is gathered, cleaned, and transformed into a data model. From there, the event relationship diagram demonstrates how the data is fed into the simulation and how the boxes and the pickers interact in the queuing network. An exploratory data analysis is conducted to understand the relationship between KPIs before the simulation results are described.

4.2.1 Data Model

Based on the queuing network description, eight main entities, namely Warehouse, WarehouseRepresentation, Conveyor, Station, Box, BoxManager, Picker, and PickerManager, are defined for the simulation data model. The entity relations are visualized in the schematic class diagram in Figure 4.5. The one-to-one and many-to-many relationships are represented by a simple straight line without and with numbers above the line, respectively. The straight line with a thin diamond at one end specifies the one-to-many relationship where an element of the entity at the diamond end may be linked to many elements of the other entity.

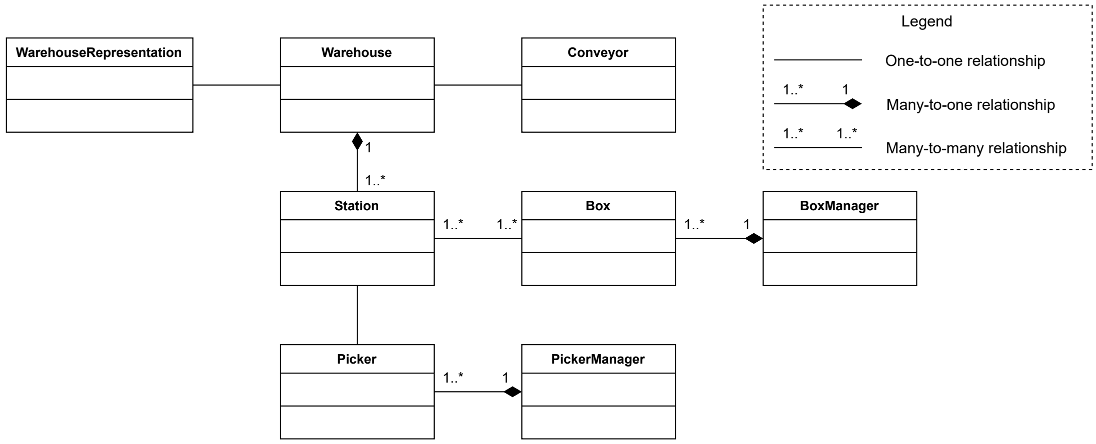


Figure 4.5: Schematic class diagram of the data model.

Warehouse represents the shelving area; hence, it manages the whole order picking system and loads the storage assignment into stations. One warehouse possesses multiple stations, i.e., one-to-many relationship for Warehouse–Station, and one conveyor system connecting the stations, i.e., one-to-one relationship for Warehouse–Conveyor. A station can be visited by many boxes and a box may need picks from several stations; therefore, the relationship between Station and Box entities is many-to-many. BoxManager and PickerManager admit the one-to-many relationship with Box and Picker respectively. Since the maximum number of assigned pickers to a station is one, the one-to-one relationship is ideal for Station and Picker. WarehouseRepresentation entity is associated with Warehouse in one-to-one relationship in which it visualizes the floor layout and the pick spread on the floor or station level according to the information given from the warehouse.

The relevant data for the simulation is currently stored in Oracle and MSSQL database. However, information concerning transport scans, i.e., when a box passes which controller, is scheduled to be erased 6 days after the logged date. Other information regarding boxes and pickers are gathered from several tables. SimulationDataExtract entity is built to extract and transform the necessary data from Oracle and MSSQL database and load them incrementally to Google BigQuery. It helps unify the data source and reduce the computational cost. The data preparation, transformation, and loading of the simulation is presented in Figure 4.6. The data transformation includes various data preprocessing steps exemplified by unifying the key's name in different tables, filtering out boxes with incomplete transport scans and time entering the shelving area before the start working time of the first outbound picker, and deriving the order on the conveyor for each twin station. After the transformed data is loaded to Google BigQuery, a SimulationDataLoader is responsible to read and transform those data into the correct form for the main entities, namely Warehouse, Conveyor, Station, BoxManager with Boxes, and PickerManager with Pickers, to consume. These entities are fed into the simulation to produce the performance statistics, or KPIs, which are then visualized by Warehouse-Representation entity.

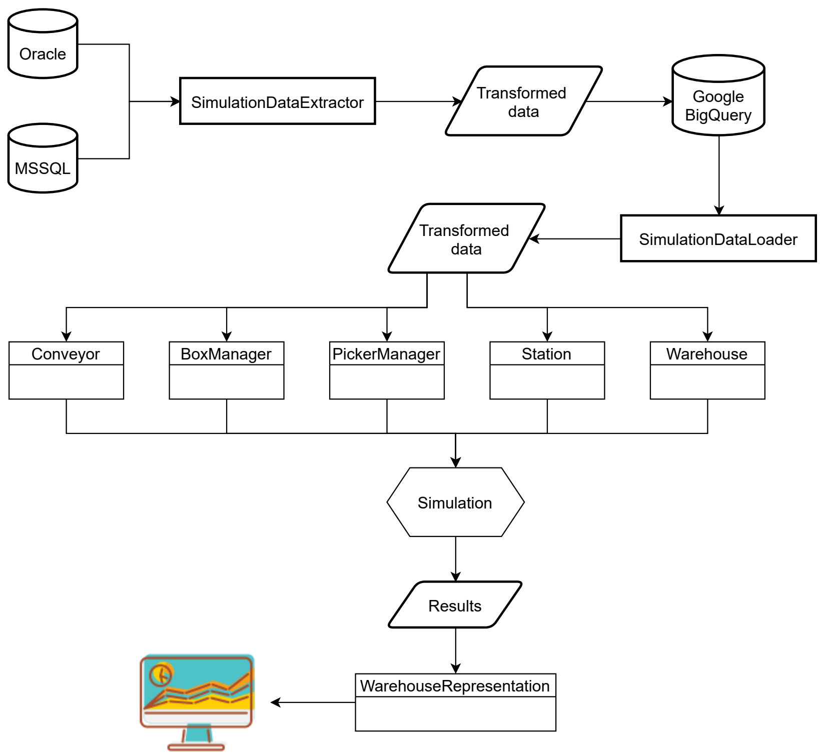


Figure 4.6: The process of extracting, transforming, and loading data into the simulation.

4.2.2 Event Relationship Diagram

The event relationship diagram is plotted in Figure 4.7 in which a white rectangular box represents an entity while a grey one corresponds to an event. The diamond shape indicates a warehouse state check and the arrows imply the event flows. Two entities, namely BoxManager and PickerManager, from the previous subsection are responsible for feeding boxes and scheduling pickers, respectively, into the shelving area.

When a box enters the warehouse, event `BoxEnteringController` is triggered with additional parameter `controller=K02`. The value of this parameter can be set as the start controller K02 or the end floor controllers BP06, BP09, and BP12 in Figure 4.1. When the box needs to recirculate, the intermediate controller K19 always follows the end floor controllers, hence it is unnecessary to trigger another `BoxEnteringController`.

Regardless of the parameter, the event checks whether all the picks for the box were performed. If yes, the box exits the system by passing through the end controller K04 and activating the `BoxExitingSystem` event. Otherwise, `BoxEnteringController` redirects the box to a floor required for the picks. Since WMS is bought as a service, its floor assignment is a black box for Kramp; therefore, the simulation employs a simple random floor assignment. The box then visits the first twin stations on the assigned floor with `BoxArrivingStation` event. Here the system checks whether the box needs picks from either station and whether the station has available spaces in the buffer corresponding to the box's type, i.e., normal or express. If no, the box continues its path on the conveyor by visiting the next twin stations with another `BoxArrivingStation` event or the end floor controller with a `BoxEnteringController` event, depending on whether the current twin stations are the last stations on the floor. Otherwise, the box enters the buffer of one of the twin stations. If there is no assigned picker at the station, the warehouse locates idle pickers and directs the closest to the station in question by activating `PickerArrivingStation`. When a picker is present at the station and they are idle, they are assumed to wait 10 seconds for more boxes to arrive then start choosing boxes and arrange the tour with `PickerChoosingBoxesToPick`. These 10 seconds are different from the average 11 seconds for setting up the tour. Afterwards, the items are retrieved and post-processed with `PickerRetrievingItemsForBoxes` and the box is released back to the conveyor by the event `PickerReleasingBoxes`. The warehouse again checks whether the current station is one of the last twin stations on the floor and determines where to redirect the box next.

On the other hand, a picker is scheduled to work by the PickerManager. When their shift starts, a `PickerStartWorking` is triggered, they are immediately looking for work in stations with no pickers (due to the picker constraint of maximum one picker per station) by `PickerFindingWork`. It is followed by `PickerArrivingStation` if a station is found. If there are boxes in the buffer, a picker starts choosing which boxes to pick with `PickerChoosingBoxesToPick`, followed by `PickerRetrievingItemsForBoxes` and `PickerReleasingBoxes`. The latter event then checks two consecutive conditions: (i) whether the picker's working shift has ended, if yes, trigger the `PickerFinishWorking` event; (ii) whether it is time for them to take a break, if yes, activate the `PickerTakingBreak` event. The break time conditions are that (i) the picker must work a shift of at least 6 hours, (ii) they have worked for at least half of their shift duration, and (iii) there are some pickers still working to prevent all pickers from having break simultaneously. If at least one of these conditions is negative, the picker either continues picking if new boxes have arrived at the station, or looks around for more work. After break time, the picker goes back to work by `PickerFindingWork`. When the last box leaves the warehouse, it notifies all the working pickers to get off work. This concludes a working day for the outbound pickers.

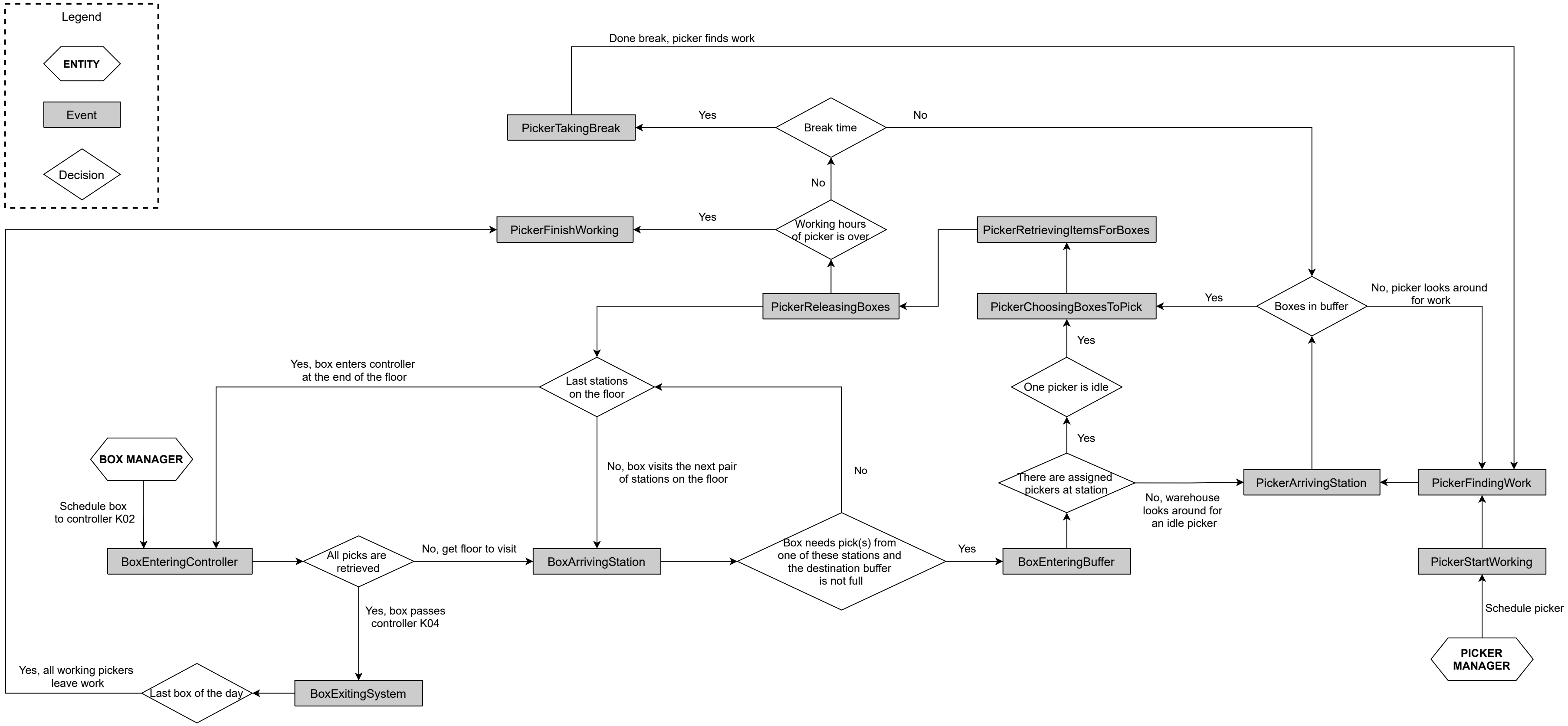


Figure 4.7: Event relationship diagram for queuing network simulation.

4.2.3 Exploratory Data Analysis

Before simulating the queuing network, an exploratory data analysis is conducted to examine the relations between stops per box SpB and other figures, namely picks per stop PpS, box throughput time TpB, the number of blocked queues per box $BQpB$, and the number of recirculations per box RpB, from the historical data on `31-05-2021`. The date was picked since there was a reasonably large number of boxes but a relatively small number of pickers working in comparison to other dates in late May and early June. This implies that boxes on `31-05-2021` tend to experience more blockages and recirculations than other days. Therefore, if the warehouse performance can be improved on the simulated day, the performance on other days is highly likely to be improved as well.

The number of items and stops per boxes on `31-05-2021` are plotted in Figure 4.8. As observed from the figure, when the number of items per box is less than 9, there are boxes with the number of stops per box SpB greater than the number of items per box. This is due to the fact that a box must visit multiple shelf spaces of an item when the ordered quantity of the item is greater than the stored quantity in its shelf space of the least quantity. Moreover, from the right half of Figure 4.8, the higher number of items per box, the more likely that these items are stored in the same station, the smaller increment in the number of stops the box requires.

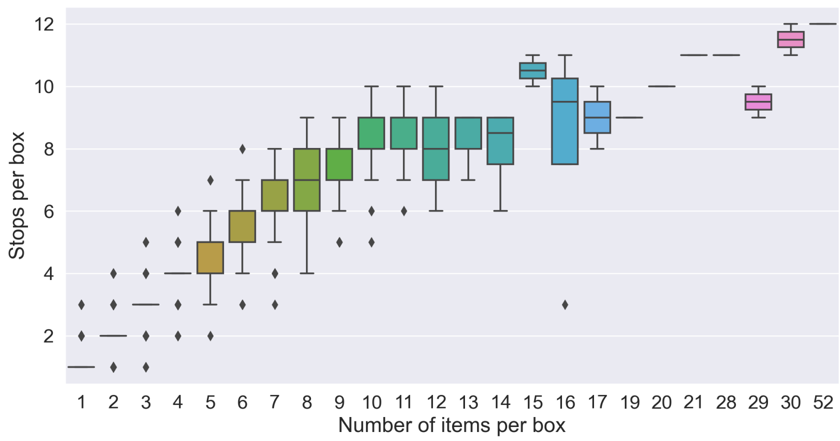
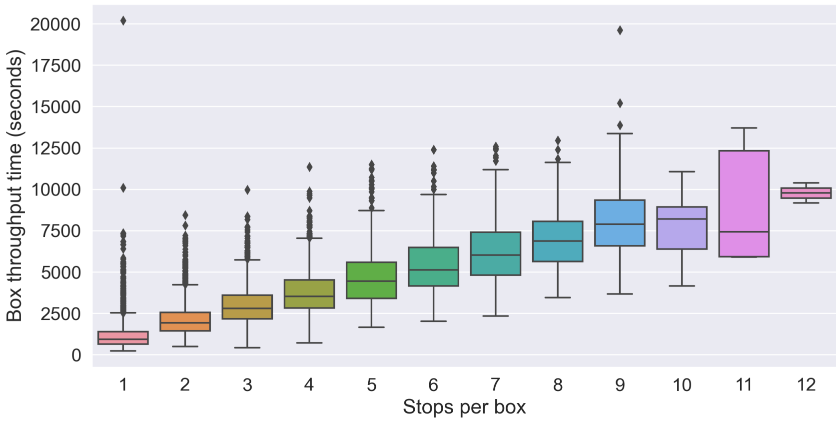
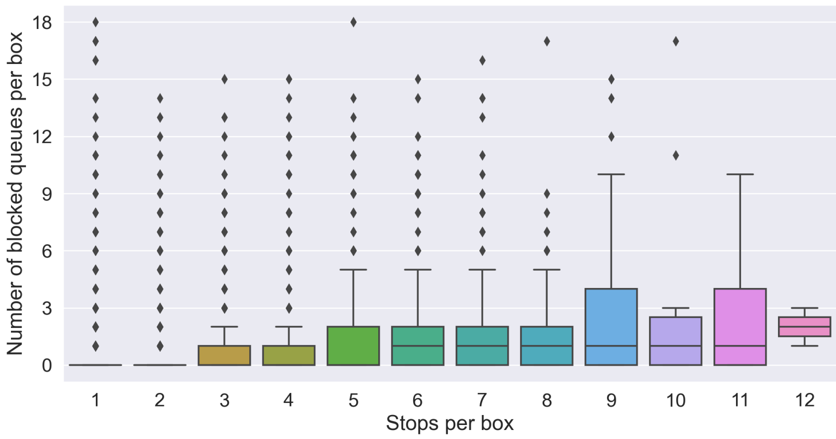


Figure 4.8: The number of items per box and its corresponding number of stops, on `31-05-2021`.

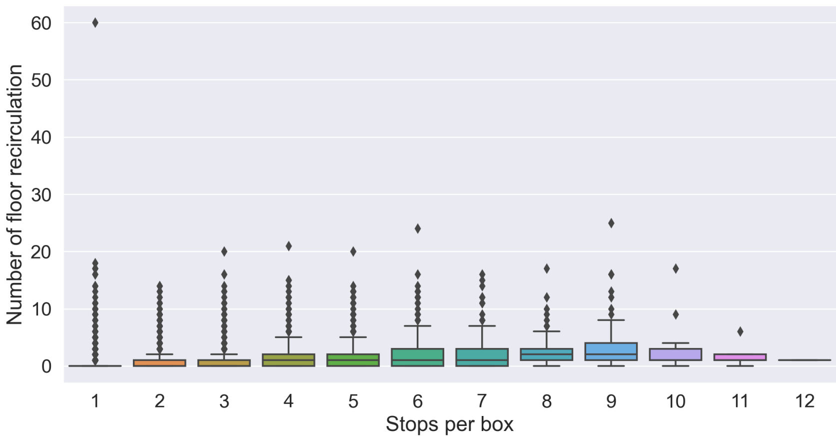
Figure 4.9 shows that SpB exhibits a positive correlation with the box throughput time TpB, the number of blocked queues per box $BQpB$, and the number of floor recirculation RpB. It means that if a box stops at fewer stations, it takes less time to travel through the shelving area and encounters less full buffer and recirculation. However, with the same SpB, RpB obtains several extreme outliers with values much larger than $BQpB$. This results from the error of the controller scan which keeps a box recirculating on a floor with no information about the queues logged. The error can be observed in Figure 4.9(a) and (c) when a box needed only one stop, yet it recirculated 60 times and spent approximately 20,000 seconds in the system.



(a) With box throughput time.



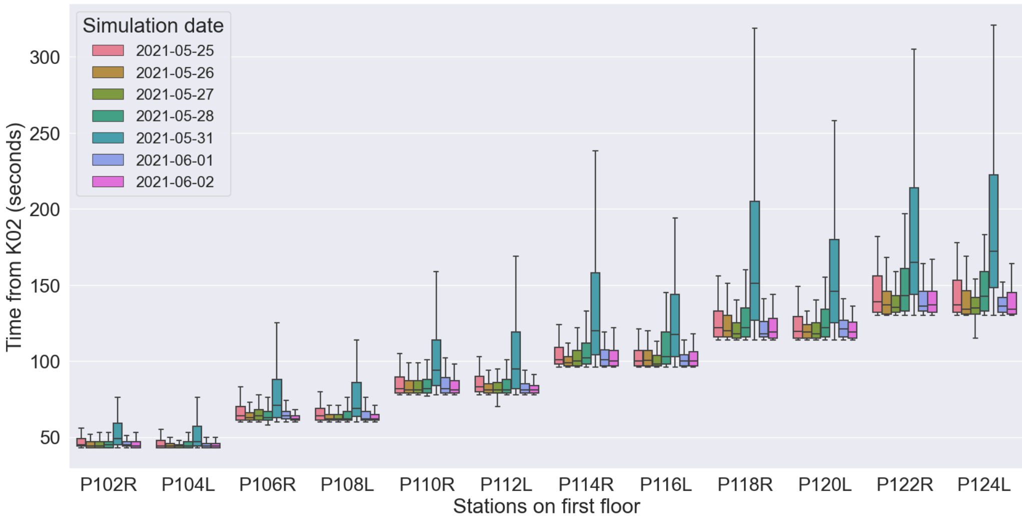
(b) With number of blocked queues.



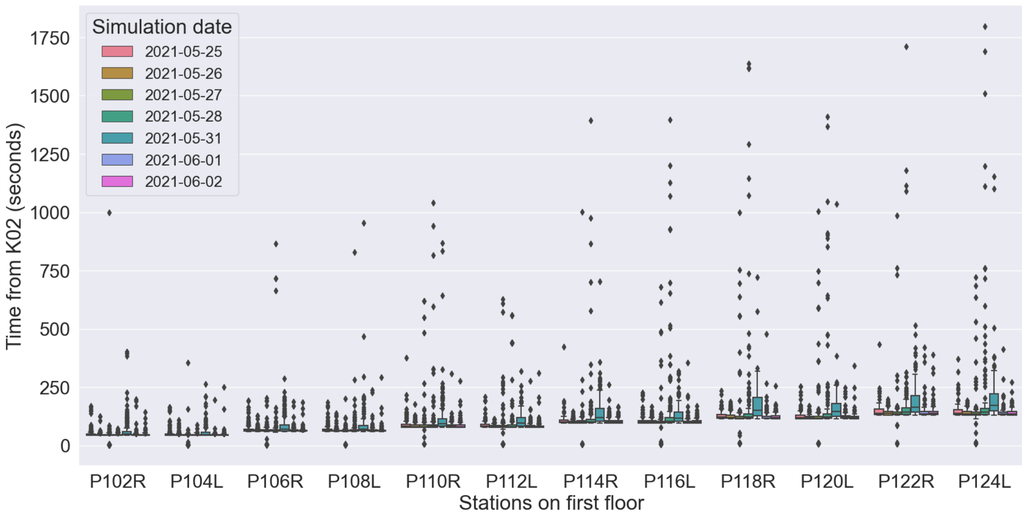
(c) With number of floor recirculation.

Figure 4.9: Relationship between stops per box and other statistics, on `31-05-2021`.

The conveyor time between two arbitrary points on Figure 4.1 varies throughout the day and day to day, which implies that the conveyor time needs to be tailored based on the chosen simulation day. Despite the absence of picking-related actions, the conveyor time from the start controller K02 and one station on the first floor still fluctuates significantly as observed in Figure 4.10. In order to estimate the travel time between stations, the start controller K02 is used as the base starting point to eliminate the picking time in the previous station. Figure 4.10 shows that the travel times between K02 and one station are approximately equivalent to that between K02 and its twin station. Additionally, the conveyor time from K02 to twin stations increases proportionally with their position on the floor. As a result, the conveyor time is modelled as the sum of a minimal travel time and an random exponentially distributed component to account for the congestion, malfunction, or delay. However, the simulation is still not able to capture all of the variations of the travel time on the conveyor due to a large amount of extreme outliers.



(a) Without outliers



(b) With outliers

Figure 4.10: Conveyor time between the start controller K02 and stations on the first floor.

4.2.4 Simulation Results

To simulate the current order picking system, two scenarios have been built. Scenario 1 describes the simulation with multiple locations per item or stock splitting, while Scenario 2 employs single location per item or without stock splitting. Both scenarios assume that there is no pick error, i.e., pickers retrieve the right amount for the items on the first attempt. Moreover, if a normal box has been recirculated more than three times, its type changes to express. An additional assumption applied on Scenario 2 to accommodate the single location is that the item stock is unlimited, i.e., the shelf space is filled immediately when it becomes empty. On `31-05-2021`, the number of simulated boxes and outbound pickers are 6,847 and 36, respectively. The acceptable error range is chosen to be 10% due to the complexity of the process.

Due to the randomness when assigning boxes to floors, each scenario is simulated for 100 runs. Five KPIs are calculated for each run, and the results are summarized in Table 4.1 in which the half width of the 95% confidence interval of the KPIs in 100 runs is given in brackets. The result comparison between the current status of the warehouse performance and the two scenarios are summarized as follows.

- The number of picks and stops in Scenarios 1 and 2 is smaller than those of the current status due to the outbound restriction and the assumption that there is no pick error. The number of picks and stops in Scenario 2 is the smallest due to additional assumption of a single location per item.
- The decreases in both $aSpB$ and $aPpS$ lead to the reduction in $aTpB$ in Scenario 1 and 2. However, the errors for $aTpB$ in both scenarios are relatively larger than those for $aSpB$ and $aPpS$, yet within the acceptable error range.
- The average number of blocked queues $aBQpB$ is supposed to be always larger than the average number of recirculation $aRpB$, which is not the case in the current status. This is due to the controller error that sometimes the box just runs through a floor without any actions logged in the database. Taking the ratio of $aBQpB$ and $aRpB$ induced from Scenario 1 and 2, $aBQpB$ of the current status may range from 1.24 to 1.29, i.e., the error in $aBQpB$ of these two scenarios reduces to 20.2%-25% and -2.3%-1.6%, respectively.
- The $aRpB$ of Scenario 1 is almost 20% higher than that of the current status due to the fact that several extreme outliers of the conveyor time between two particular points on Figure 4.1 cannot be captured in the simulation, as well as the current status receiving a smarter floor assignment of a box from WMS. By employing a simpler storage assignment, Scenario 2 offsets the difficulties with a smaller number

of picks and stops. Therefore, the $aRpB$ in Scenario 2 is closer to that of the current status that in Scenario 1.

With the adjusted $aBQpB$ taken into account, Scenario 2 achieves a closer resemblance to the current status than Scenario 1, although the set up of Scenario 1 is closer to the current status than of Scenario 2. Since single location of the items is also a common assumption in cluster-based storage assignment literature from Chapter 2, Scenario 2 is used as the base to study the impact of frequent item pairs in storage assignment on the warehouse performance.

Table 4.1: Simulation result for Scenario 1 and 2 on `31-05-2021` and the errors compared to the current status, after 100 runs. The half width of the 95% confidence interval of the KPIs in 100 runs is given in bracket.

	Current status	Scenario 1	Scenario 2	Error 1 (%)	Error 2 (%)
Number of boxes	6,847	6,847	6,847	-	-
Number of picks	21,196	21,135	20,805	-0.288	-1.845
Number of stops	19,356	19,195	19,131	-0.832	-1.162
$aSpB$	2.827	2.803	2.794	-0.849	-1.167
$aPpS$	1.095	1.101	1.088	0.548	-0.639
$aTpB$ (in seconds)	2,793.89	2,769.05 ($\pm$9.517)	2,633.70 ($\pm$6.680)	-0.889	-5.734
$aBQpB$	0.99	1.55 ($\pm$0.024)	1.26 ($\pm$0.016)	56.566	27.273
$aRpB$	1.06	1.27 ($\pm$0.019)	1.08 ($\pm$0.014)	19.811	1.887

5 STORAGE ASSIGNMENT WITH FREQUENT ITEM PAIRS (FIP)

This chapter outlines the impact of frequent item pairs (FIP) in storage assignment on the performance of Strullendorf warehouse. The first section describes the data mining process for FIP, the FIP graph construction, and the algorithm to assign clusters in FIP graph into an existing storage assignment, respectively in Subsections 5.1.1, 5.1.2, and 5.1.3. The second section develops two statements and three scenarios to validate the impact in the first two subsections. Subsection 5.2.3 summarizes the simulation results of three proposed scenarios and reveals a positive improvement on all KPIs when compared to the base simulation result of Scenario 2 from the previous chapter. However, decreasing the support threshold to involve more items in FIP clusters may have a detrimental impact on some KPIs, therefore it must be handled with caution.

5.1 FIP and its Integration in Storage Assignment

Aligned with the literature, the report uses historical order data to extract the information about FIP. The date range of 365 days is chosen to obtain the complete overview of item affinity. The report then takes advantage of the well-developed literature on undirected graph and its connected component detection algorithm to solve the item clustering problem. The FIP graph is constructed and the connected components, or clusters, are visualized and assigned to stations according to a heuristic algorithm.

5.1.1 Historical Order Data

In order to determine the frequent item pairs, the historical order data is retrieved for all the customers associated with Strullendorf as their default warehouse and all the items present in the current storage assignment. The frequent item pairs should be determined one week in advance so that there is a sufficient amount of time to relocate the items in the warehouse. Therefore, the date range for the order data is chosen to be 365 days from

24-05-2020 to 24-05-2021 since the simulated date is 31-05-2021. There are 599,903 multi-item transactions involving 141,362 items stored in the shelving area. 45% of the number of transactions consists of either one or two items while the largest order has 313 items, as shown in Figure 5.1. Based on the order size distribution and the fact that the number of transactions is approximately five times higher than the number of items, support count is more suitable than support as the measure of item affinity. A minimum support count threshold, or support threshold for short, is chosen to filter pairs with significant co-appearance in the transaction history. The data retrieval procedure is summarized in Figure 5.2.

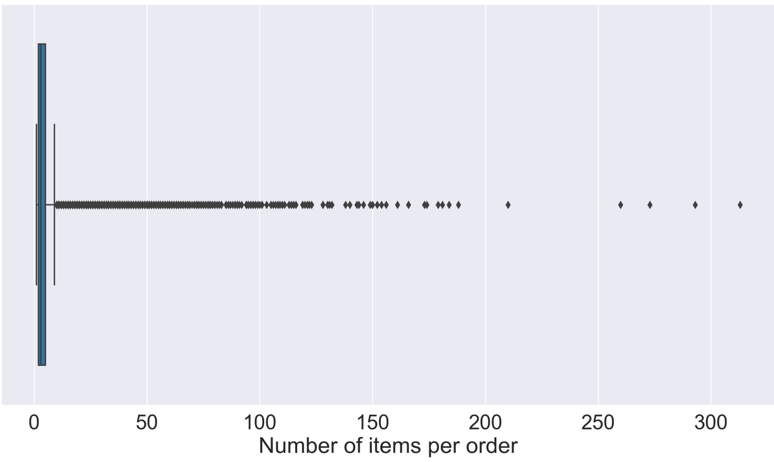


Figure 5.1: Order size distribution within the date range from 24-05-2020 to 24-05-2021.

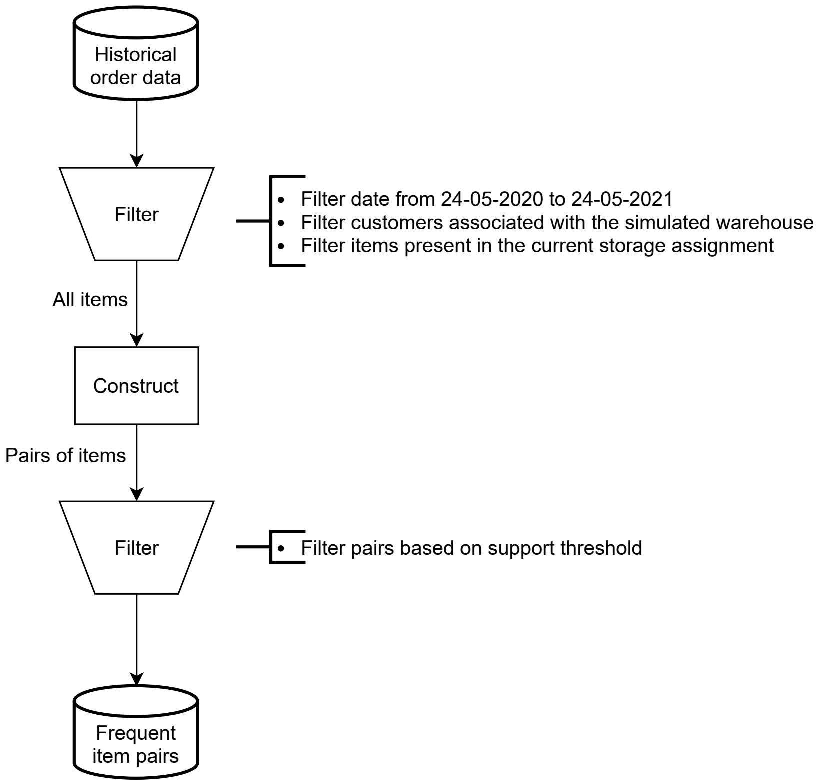


Figure 5.2: Data retrieval for frequent item pairs.

5.1.2 FIP Graph Construction

Consider each item as a node with its item number as the node's label. If two items are ordered together in the last 365 days for at least a certain number of transactions, an edge with weight equal to the support count of the two items is drawn between the corresponding nodes. Based on the node and edge definition, an undirected graph is constructed. The frequent item pairs graph is visualized in Figure 5.3 with the support threshold of 50, i.e., all edges with weight smaller than 50 are excluded.

The graph consists of many connected components, hereby referred to as clusters, with different structures such as pairs, lines, complete or tree-like graphs. The small order size and the huge item portfolio lead to the dominance of pairs in the graph. Within a cluster, the higher weighted degree of a node, i.e., the higher sum of the weight of all edges connected to a node, the more important it is for the cluster. Additionally, items in one cluster are likely to belong to the same brand and relate to each other as either alternative or complementary items. In the alternative case, they may share a common name pattern.

For pair clusters, the items demonstrate the complementary relationship. It can be (i) horizontal in the case of BA85 and BA86 which are respectively positive and negative battery terminals, or 55903210 and 55903310 which are a left and right mower blade, or (ii) vertical SV2S06LS and MV6L are the cutting ring and the swivel nut as the cutting ring connection, while 794421 and 697292 are the air filter and its compatible pre-filter. Of four mentioned pairs, only 697292 is listed as an accessory item of 794421 on the web shop. Unfortunately the web shop relationship is only one way, 794421 is not associated with 697292. This finding suggests that the FIP graph might be beneficial for not only the storage assignment in the warehouse but also other functionalities in the company.

For clusters consisting of three items, the graph structure can be either triangle or line. The cluster of 490190027, 490650721, and 110137046 exemplifies the first case in which they are fuel filter, oil filter, and pre-filter of the same brand, Kawasaki. The triangle structure signals that the items are complementary and may have been ordered together for at least 50 times in the last 365 days. The prediction is quite close as the data shows that the number of orders including all three items is 48. In contrast, the line structure implies that it is highly unlikely for all three items to appear in the same order, and the items at the two ends are likely to demonstrate an alternative relationship. For example, 1093109 (serrated bolt M6x16 with 100 units), 13961 (flange nut M6), and 1093108 (serrated bolt M6x16 with 50 units) are clustered in a line of the corresponding order. Since items 1093109 and 1093108 are the same product with different units sold, they are indeed alternative and it is uncommon for them to appear in the same transaction.

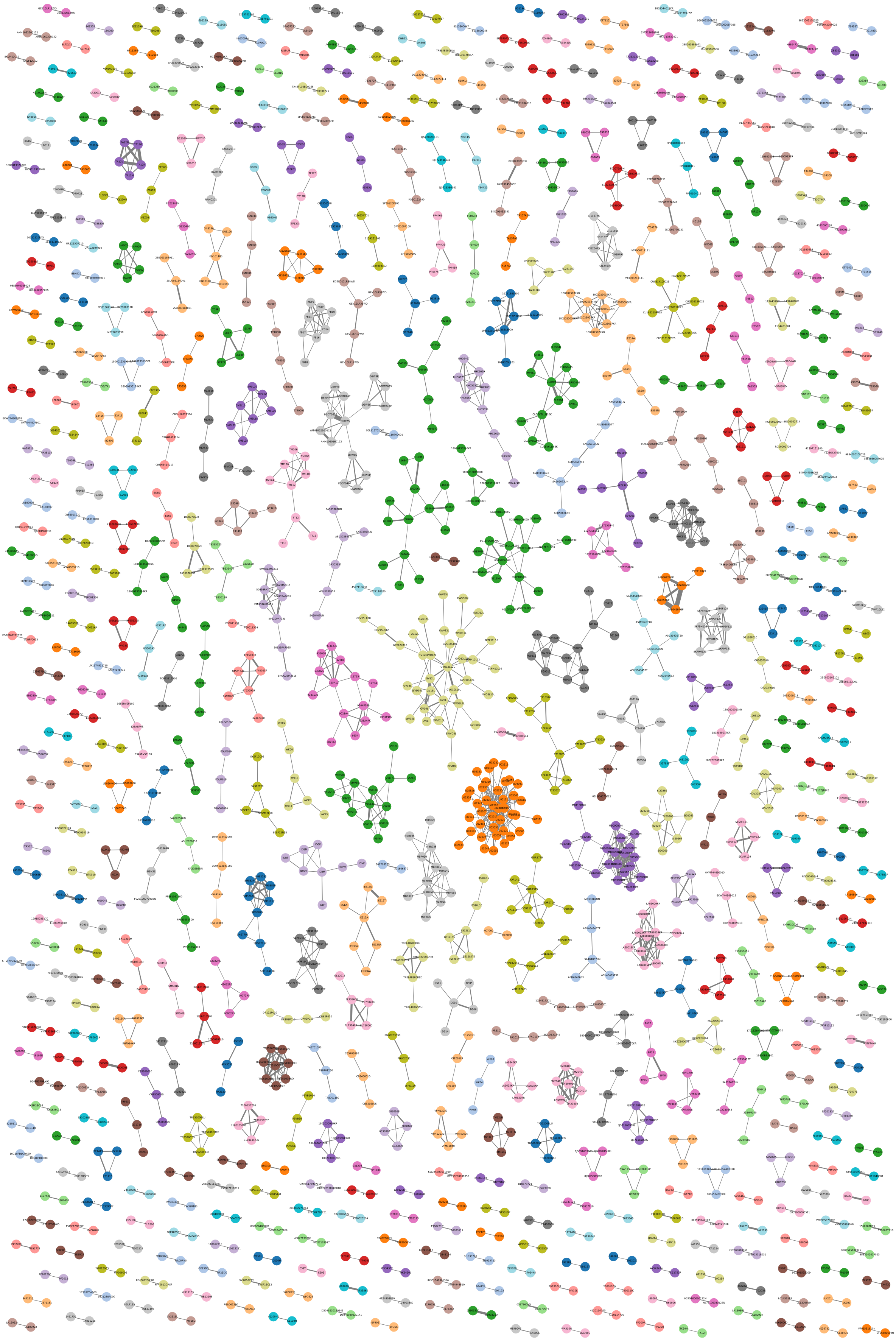


Figure 5.3: Graph visualization of frequent item pairs with support threshold of 50. Each node represents an item and two nodes are connected by an edge if over the considered period, the two corresponding items have been ordered together at least 50 times.

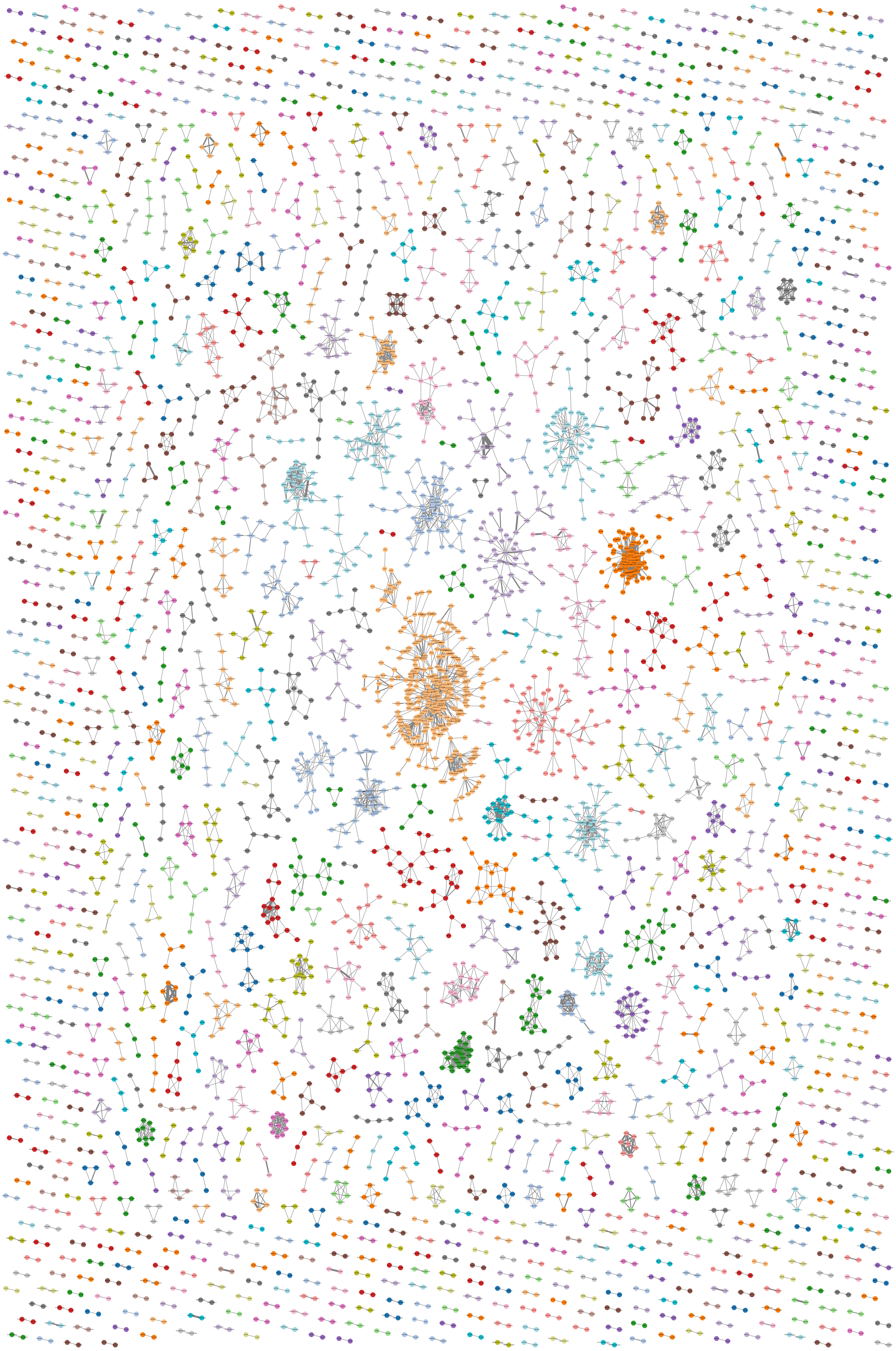


Figure 5.4: Graph visualization of frequent item pairs with support threshold of 25. Each node represents an item and two nodes are connected by an edge if over the considered period, the two corresponding items have been ordered together at least 25 times.

There are also many complete clusters in Figure 5.3, i.e., each node connects with every other node in the cluster. Interestingly, in most cases, the items in those clusters are the alternative products with (i) same size yet different colors, illustrated by the cable tie cluster consisting of TR25100GREY (grey), TR25100BLU (blue), TR25100BRO (brown), TR25100ORA (orange), TR25100GREE (green), TR25100RED (red), TR25100VIO (violet), and TR25100YEL (yellow); or (ii) same length yet different widths, such as these 50m long masking tapes, namely 7100135761 (36mm), 7100135737 (24mm), 7100135720 (48mm), and 7100135739 (18mm).

Various toy versions of agricultural equipment, for example tractors, mowers, and slurry tanks, form the largest cluster in terms of weight. There are 39 items with a total edge weight of 10,668, which is around 1.34 times the weight of the next largest and 2.22 times the weight of the third largest. However, the larger the cluster, the more likely that a node has only one connected edge, and the less close-knit the cluster becomes.

Figure 5.4 visualizes the FIP clusters when the support threshold halves in value from 50 to 25. The number of items involved in the FIP clusters almost triples from 1,693 to 4,835. The dominant types of clusters remain cluster as a pair or line. The previous third largest cluster has merged with its vertically complementary clusters thanks to newly added items, all of which are related to couplings and manual lubrication parts, such as adaptors, cutting rings, reusable inserts for push-in fittings, etc., to form the new largest cluster. Its cluster weight soars to around 32,000. The previous largest cluster can only grow horizontally by appending additional toys. This behavior is common in clusters with alternative products.

5.1.3 Station Assignment of Clusters

To rearrange items involved in FIP clusters in an existing storage assignment, those items are assumed to be removed from their current shelf spaces. Moreover, no stock splitting is allowed, i.e., one item is assigned to only one shelf space. New variables are introduced in addition to variables in Subsection 4.1.4 to formulate a mathematical model for the storage assignment problem.

- F: the set of FIP clusters, indexed by u and v.
- M_s: the set of free shelf spaces in station s with items involved in FIP clusters removed, indexed by ms.
- L: the set of possible LHM types, indexed by l.
- w_u, $u \in F$: the weight of cluster u.

- $p_{iu},\ i \in I,\ u \in F$: 1 if item i belongs to cluster u; 0 otherwise. Since one item is associated with maximum one cluster, $\sum_{u \in F} p_{iu} \leq 1$. If the figure equals $\sum_{u \in F} p_{iu} = 0$, item i does not exhibit a sufficiently strong affinity with any other items based on a particular support threshold. In this case, item i keeps its current storage assignment.

- $d_{bs},\ b \in B,\ s \in S$: the distance a picker needs to travel to retrieve picks for box b in station s, in centimeters. $d_{bs} = 0$ when box b does not require any picks from station s, i.e., $\sum_{i \in I} z_{ibs} = 0$. When $\sum_{i \in I} z_{ibs} > 0$, d_{bs} is computed according to the return routing policy. Here no batch servicing is assumed, i.e., there is maximum one box per pick tour.

- $T_i \subseteq L,\ i \in I$: a list of compatible LHM type to item i. Since one item can have at least one compatible LHM type, T_i has at least one element.

- $t_{ms} \in L,\ ms \in M_s$: the LHM type of shelf space ms in station s.

- $a_{us},\ u \in F,\ s \in S$ (decision variable): 1 if cluster u is assigned to station s; 0 otherwise. Assume that a cluster is assigned to one station exclusively.

- $ss_{ims},\ i \in I,\ ms \in M_s$ (decision variable): 1 if item i is stored in shelf space ms of station s; 0 otherwise. The necessary condition for $ss_{ims} = 1$ is that LHM type of shelf space m in station s is one of the suitable types for item i.

To balance the cluster weight across stations, clusters with frequent item pairs must be distributed evenly to stations while maintaining their structure, i.e., no cluster division is allowed. That is, the higher the cluster weights w_u and w_v, the less likely clusters u and v are present in the same station. To achieve that goal, a mathematical model is constructed. The first term of the objective function (5.1) $a_{us}a_{vs}(w_u + w_v)$ equals $w_u + w_v$ if and only if clusters u and v are both assigned to station s, i.e., $a_{us} = a_{vs} = 1$. The second term computes the total travel distance needed to pick all items for boxes in all stations. The distance unit is chosen to be in centimeters so that the magnitude of the second component has the similar range to that of the first component. This helps balance the importance between the two terms. Minimizing the objective function (5.1) then ensures that big clusters are not stored in the same station and the travel distance for box b in each station is minimal. As a result, the cluster-based storage assignment with zoning is properly handled as compared to the literature in Table 2.2. Constraints (5.2) and (5.3) restricts each cluster and item such that it must and can only be located into one station and one shelf space, respectively. Constraints (5.4) and (5.5) guarantee that items are stored in the same station as its cluster and in a compatible shelf space to its corresponding LHM types. Constraint (5.6) prevents the number of items with LHM type l moving to station s from exceeding the number of available shelf spaces of type l in the station.

$$\min \quad \sum_{s\in S}\sum_{u\in F}\sum_{\substack{v\in F\\ v\neq u}} a_{us}a_{vs}(w_u + w_v) + \sum_{s\in S}\sum_{b\in B} d_{bs} \tag{5.1}$$

$$s.t. \quad \sum_{s\in S} a_{us} = 1, \qquad u \in F, \tag{5.2}$$

$$\sum_{s\in S}\sum_{ms\in M_s} ss_{ims} = 1, \qquad i \in I, \tag{5.3}$$

$$\sum_{ms\in M_s} ss_{ims} = a_{us}, \qquad i \in I,\ s \in S,\ u \in F,\ p_{iu} = 1, \tag{5.4}$$

$$ss_{ims} \leq \mathbb{1}_{(t_{ms}\in T_i)}, \qquad i \in I,\ ms \in M_s,\ s \in S, \tag{5.5}$$

$$\sum_{u\in F}\sum_{i\in I}\sum_{ms\in M_s} a_{us}p_{iu}ss_{ims}\mathbb{1}_{(t_{ms}=l)} \leq \sum_{ms\in M_s} \mathbb{1}_{(t_{ms}=l)}, \qquad s \in S,\ l \in L, \tag{5.6}$$

$$a_{us} \in \{0,1\}, \qquad u \in F,\ s \in S, \tag{5.7}$$

$$ss_{ims} \in \{0,1\}, \qquad i \in I,\ ms \in M_s,\ s \in S. \tag{5.8}$$

The model has two limitations. Firstly, due to the strict constraints, there may be no solution to the problem in some variable combinations. Secondly, the lower the support threshold, the higher number of items involved in FIP clusters needs to be removed from the initial storage assignment so that all possible shelf spaces are taken into account when reassigning. This removal is not realistic in the warehouse as it requires a significant amount of manpower. To efficiently solve the model, a heuristic algorithm is developed. The pseudo code is demonstrated in Algorithm 1. Since each box contains some or all items of one order, the algorithm produces an improved storage assignment which helps reduce the average number of stops per box *aSpB*. The key point to the algorithm is to find the suitable destination station for a cluster that maximizes the chance of the whole cluster is stored together, while balancing the workload and total cluster weight among stations. Since the algorithm has chosen the station with the most available shelf spaces compatible to items in the cluster, in the case when the chosen station still does not have sufficient spaces, the items with the smallest weighted degree have to remain in their current location. This resolves the first limitation. Regarding the second limitation, the algorithm move items to an empty shelf spaces in the destination station, then update M_s before dealing with the next cluster.

Algorithm 1: Assign clusters to stations and items to shelf spaces.

```
Input  : Current storage assignment, clusters of frequent item pairs, free shelf
         spaces at station (fsss), LHM types per item.
Output: Clusters by station and item movement.
Initiate cluster weight per station starting with 0;
Initiate item movement;
for cluster in descending order of cluster weight do
    Get station with enough free shelf spaces of corresponding items' LHM type,
      the least cluster weight, and the smallest number of items currently present
      in the station;
    Update cluster weight per station;
    if there are items of the cluster which are not in the station yet, i.e., outside
      items then
        if there are items of the cluster that belong to the station, i.e., inside
          items then
            Get aisles of those inside items and available shelf spaces in those
              aisles;
            Assign outside items to the closest open shelf spaces of the
              corresponding LHM type in the descending order of item degree in
              the cluster;
            Update item movement and fsss;
        end if
        if there are no inside items or the aisles of inside items do not have
          enough free shelf spaces for outside items then
            Get all free shelf spaces in the station;
            Assign (remaining) outside items to the closest open shelf spaces of
              the corresponding LHM type in the descending order of item degree
              in the cluster;
            Update item movement and fsss;
        else
            Keep the current assignment of the outside items;
        end if
    end if
end for
```

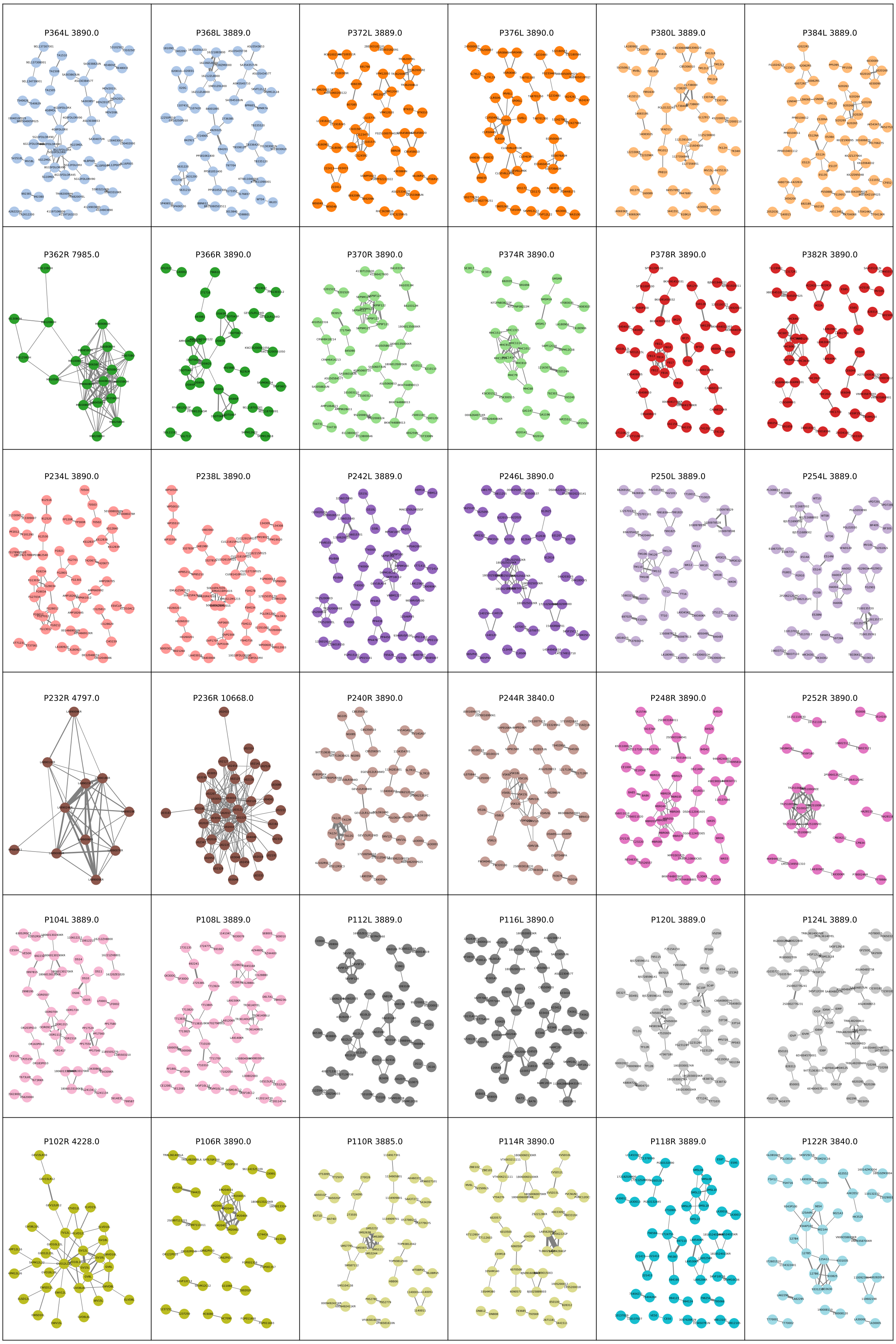


Figure 5.5: Assign clusters from Figure 5.3 to stations based on Algorithm 1. Each rectangle represents a station and is labelled with the station identification and the station weight.

One of the algorithm inputs is the current storage assignment in which no cluster-based analysis has been applied. Suppose that the FIP cluster with support threshold of 50 is already applied to the storage assignment. If the warehouse performance shows improvement after the integration, it is desired to decrease the support threshold to involve more items in the FIP clusters. In this case, the objective function of the algorithm when choosing the suitable station should be modified to minimizing the number item moves to accommodate the newly involved items. This function will help minimize the chance of rearranging a whole cluster to a new station. If a new item arrives at the warehouse, it should be stored in a station with the highest total support count. The total support count is defined as the aggregated sum of all possible pairs' support count between the new item and each item in a particular station if it is greater than the support threshold.

The outcome of the algorithm is visualized in Figure 5.5 when the support threshold is set at 50. Station weight is defined as the weight sum of all FIP clusters assigned to a station. Since cluster splitting is not allowed, there are four stations, each containing only a single cluster, namely P236R (10,668), P362R (7,985), P232R (4,797), and P102R (4,228) with their station weight specified in the bracket. The cluster weight in other stations remains at around 3,900. This imbalanced station weight distribution may lead to the unequal pick workload among stations. The stations with a single distributed cluster may become overloaded while others suffer from box starvation. This potential impact has not been studied in previous literature on cluster-based storage assignment as shown in Table 2.2. Hence, it will be validated in the next section through a simulation study.

5.2 Simulation with FIP Integration in Storage Assignment

With the output generated by Algorithm 1, the storage assignment with FIP integration is incorporated into the simulation with the additional KPIs and scenarios to fully understand its effect on the warehouse performance.

5.2.1 Additional KPIs

To quantify the impact of frequent item pairs (FIP) in storage assignment on the warehouse performance, three additional KPIs are introduced. These KPIs are to involve other warehouse entities, namely pickers and stations.

- *Total travel time of pickers*, which accounts for the picking tour only and excludes the time pickers move from one station to another for more work.

- *Pick spread across stations*, defined as the standard deviation of the number of picks per station. The larger the pick spread, the more imbalanced the station workload becomes. The number of picks in station $s \in S$ is computed as $\sum_{j \in B} \sum_{i \in I} z_{ijs}$. Therefore, pick spread across stations can be formulated as follows.

$$\sqrt{\frac{1}{|S|} \sum_{s \in S} \left(\sum_{i \in I} \sum_{b \in B} z_{ibs} - \bar{z} \right)^2}$$

  where $|S|$ is the number of stations in the system, i.e., 36, and $\bar{z} = \frac{1}{|S|} \sum_{s \in S} \sum_{i \in I} \sum_{b \in B} z_{ibs}$ is the mean number of picks per station.

- *Correlation between stations' number of items and number of picks*, computed as,

$$\frac{\sum_{s \in S} \left(\sum_{i \in I} y_{is} - \bar{y} \right) \left(\sum_{i \in I} \sum_{b \in B} z_{ibs} - \bar{z} \right)}{\sqrt{\sum_{s \in S} \left(\sum_{i \in I} y_{is} - \bar{y} \right)^2 \sum_{s \in S} \left(\sum_{i \in I} \sum_{b \in B} z_{ibs} - \bar{z} \right)^2}}$$

  where $\bar{y} = \frac{1}{|S|} \sum_{s \in S} \sum_{i \in I} y_{is}$ is the mean number of items per station. In the case that the most frequently purchased items form a popular cluster, the correlation between station's number of items and number of picks may decrease. The station with the cluster assigned to may have less items yet the number of picks increases due to item popularity. Meanwhile, other stations receive more items from Algorithm 1 due to the cluster weight balance, yet the number of picks may decrease as the items are not as popular as items of the popular cluster.

According to the literature survey in Chapter 2 and the KPI description in Chapter 4, the expected impact of FIP in storage assignment on warehouse performance is specified in two statements, 5.1 and 5.2. Since the correlation between stations' number of items and number of picks has not been used in literature before, the expected effect on this KPI is unknown and excluded from the statements.

Statement 5.1. *Frequent item pairs in storage assignment helps decrease aSpB and increase aPpS, which consequently leads to a reduction in aTpB, aBQpB, and aRpB.*

Statement 5.2. *Frequent item pairs in storage assignment helps decrease the total travel time of pickers, yet increase the pick spread across stations.*

5.2.2 Scenario Description

From the previous chapter, Scenario 2 without stock splitting resembles the current status better than Scenario 1 with stock splitting in terms of five KPIs from Subsection 4.1.4. Therefore, based on Scenario 2, this section introduces an additional three scenarios, namely Scenario 2.1, 2.2, and 2.3, to validate the statements. The characteristics of all five scenarios from Chapter 4 and 5 are summarized in Table 5.1.

Table 5.1: Summary of scenarios from Chapter 4 and 5.

Scenario	Stock splitting	FIP in storage assignment	Support threshold	Number of clusters integrated
1	Yes	No	-	-
2	No	No	-	-
2.1	No	Yes	50	1
2.2	No	Yes	50	All
2.3	No	Yes	25	All

Scenario 2.1 integrates the biggest FIP cluster with the support threshold of 50 into the existing storage assignment in Scenario 2. Specifically, items belonging to the biggest FIP cluster are reassigned to station P236R according to Algorithm 1.

Scenario 2.2 employs the station assignment of clusters as specified in Figure 5.5 while keeping the location of other items unchanged. The support threshold in this scenario is the same as in Scenario 2.1.

Scenario 2.3 is developed to accommodate the desire of lowering the support threshold to maximize the KPI improvement by involving more items in the FIP clusters. In this scenario, Algorithm 1 is employed with a support threshold of 25 and the starting storage assignment of Scenario 2. The two largest clusters obtain similar weights of roughly 32,000, which is three times greater that the largest cluster in Scenario 2.2. These clusters are distributed separately to station P362R and P246L. The third largest with a weight of 11,478 is moved to P232R. The 33 remaining stations contain multiple clusters and share a common total cluster weights of approximately 7,350.

5.2.3 Simulation Results

The simulation results from three scenarios described in Subsection 5.2.2 are summarized in Table 5.2 and compared with those figures of Scenario 2 in Table 5.3. All four scenarios share the same number of boxes (6,847) and number of picks (20,805). Since Scenario 2 and 2.1 also have a common number of stops (19,131), the five main KPIs, namely $aSpB$, $aPpS$, $aTpB$ $aBQpB$, and $aRpB$, are almost equivalent in those two scenarios.

Table 5.2: Simulation result for Scenario 2, 2.1, 2.2, and 2.3 on `31-05-2021`, after 100 runs. The half width of the 95% confidence interval of the KPIs in 100 runs is given in bracket.

	Scenario 2	Scenario 2.1	Scenario 2.2	Scenario 2.3
Items followed current SA	163,540	163,501	161,847	158,705
Items followed SA with FIP	-	39	1,693	4,835
Support threshold for FIP	-	50	50	25
Number of clusters assigned	-	1	556	1,292
Number of moved items	-	38	1,648	4,680
Number of boxes	6,847	6,847	6,847	6,847
Number of picks	20,805	20,805	20,805	20,805
Number of stops	19,131	19,131	18,888	18,597
$aSpB$	2.794	2.794	2.759	2.716
$aPpS$	1.088	1.088	1.101	1.119
$aTpB$ (in seconds)	2,633.70 ($\pm$6.680)	2,635.01 ($\pm$6.929)	2,560.48 ($\pm$7.677)	2,568.99 ($\pm$7.718)
$aBQpB$	1.26 ($\pm$0.016)	1.26 ($\pm$0.017)	1.17 ($\pm$0.019)	1.22 ($\pm$0.018)
$aRpB$	1.08 ($\pm$0.014)	1.08 ($\pm$0.014)	1.01 ($\pm$0.015)	1.08 ($\pm$0.016)
Total travel time of pickers (in seconds)	189,246.84 ($\pm$104.353)	189,159.65 ($\pm$109.137)	185,175.47 ($\pm$106.189)	186,777.05 ($\pm$114.833)
Pick spread across stations	47.54	48.82	55.63	71.04
Correlation between stations' number of items and number of picks	0.55	0.51	0.67	0.38

Table 5.3: Comparison of Scenario 2.1, 2.2, and 2.3 on `31-05-2021` with Scenario 2, after 100 runs, in %.

	Scenario 2.1	Scenario 2.2	Scenario 2.3
Proportion of items followed current SA	99.976	98.965	97.044
Proportion of items followed SA with FIP	0.024	1.035	2.956
$aSpB$	-	-1.270	-2.791
$aPpS$	-	$+1.195$	$+2.849$
$aTpB$	$+0.050$	-2.780	-2.457
$aBQpB$	-	-7.143	-3.175
$aRpB$	-	-6.481	-
Total travel time of pickers	-0.046	-2.151	-1.305
Pick spread across stations	$+2.692$	$+17.017$	$+49.432$

Table 5.4: Comparison of SpB in Scenarios 2 and 2.1.

Item number	Associated with the biggest cluster	Scenario 2	Scenario 2.1
U02324	Yes	P232R	P236R
U03050	Yes	P236R	P236R
PTO940CJGP	No	P250L	P250L
LP7KR	No	P120L	P120L

(a) Decreasing SpB

Item number	Associated with the biggest cluster	Scenario 2	Scenario 2.1
U03015	Yes	P110R	P236R
PQB86	No	P110R	P110R
7037751BMYP	No	P102R	P102R
XPZ772XEP	No	P104L	P104L

(b) Increasing SpB

With the largest FIP cluster in Figure 5.3 assigned to station P236R, the number of stops in Scenario 2.1 is still equal to that in Scenario 2. This is due to the following reasons. In Scenario 2.1, since item U03050 of the largest cluster is already present in P236R, the number of item moves is only 38 out of 39 items in the cluster. There are 74 boxes containing items associated with the biggest cluster, only one box of which involving more than one item. Its SpB reduces from 4 to 3. Meanwhile, another box has its SpB increase from 3 to 4. All SpBs of the remaining 72 boxes are similar in both scenarios, which implies that $aSpB$ remains unchanged between Scenario 2 and 2.1. The storage assignment of items in two mentioned boxes are listed in Table 5.4.

In Scenario 2.2, 556 FIP clusters are distributed to 36 stations according to Figure 5.5. The number of stops is reduced from 19,131 to 18,888, which leads to a decrease of 1.27%

in $aSpB$ and an increase of 1.20% in $aPpS$. In Scenario 2.3, 4,835 items, roughly three times the number of items in Scenario 2.2, are involved in FIP clusters. The number of stops and $aSpB$ are further reduced by 2.79% while $aPpS$ increases by 2.85%. Therefore, the more FIP clusters incorporated into the storage assignment and the lower the support threshold, the larger number of items needs to be rearranged yet the better improvement in $aSpB$ and $aPpS$ can be achieved. Unfortunately, in comparison to Scenario 2, $aTpB$, $aBQpB$, and $aRpB$ exhibit a less promising improvement in Scenario 2.3 than in Scenario 2.2 due to the following reasons.

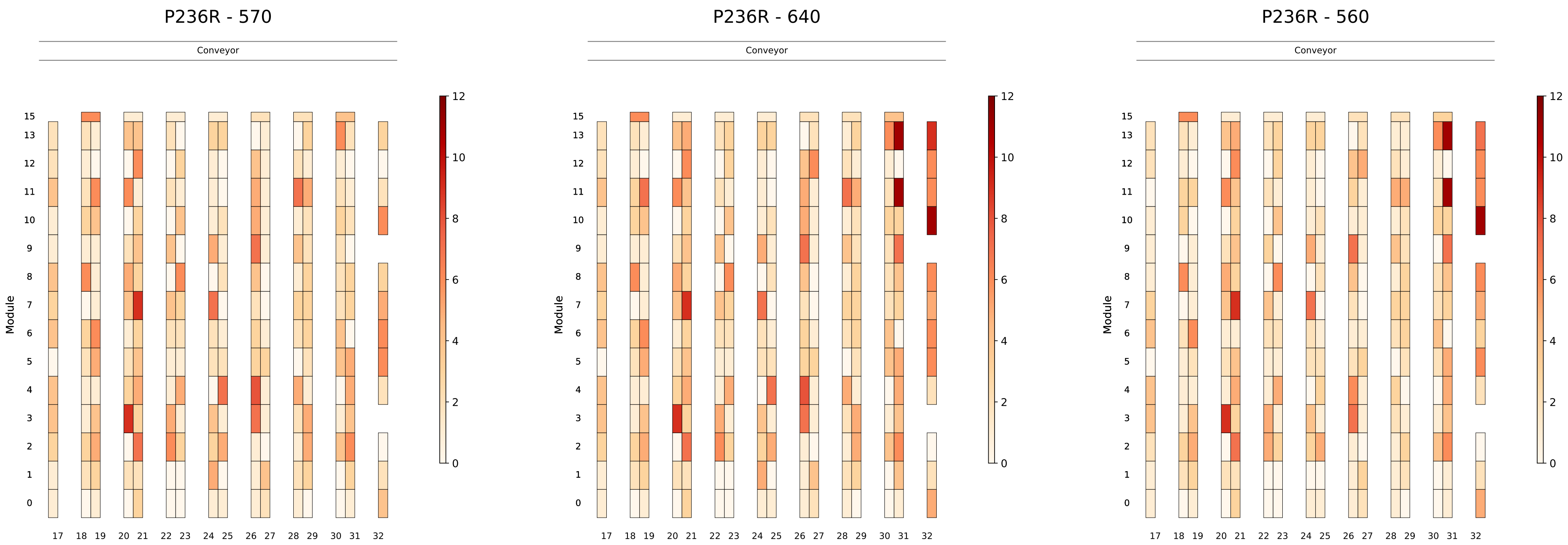
- The quotient between the highest station weight and the common station weight is much larger in Scenario 2.3 (4.35 times - 32,000 versus 7,350) than in Scenario 2.2 (2.74 times - 10,670 versus 3,890), which results in a more imbalanced workload and a notably higher pick spread across stations.
- The cluster sizes grow significantly when decreasing the support threshold from 50 to 25, with the size of the biggest cluster expanding almost 7.5 times. This leads to more items being assigned to the modules further down the aisles, since there may not be enough available shelf spaces closer to the conveyor.
- The bigger the cluster, the stronger affinity items in the station that the cluster is assigned to showcase. This explains the decrease in the correlation between stations' number of items and the number of picks, from 0.67 to 0.38. A stronger item correlation in Scenario 2.3 therefore imposes a negative impact on three main KPIs, $aTpB$, $aBQpB$, and $aRpB$, as well as the total travel time of pickers and the pick spread across stations, when comparing to Scenario 2.2.

The pick distribution in P236R in Scenario 2, 2.1 and 2.2 are plotted in Figure 5.6. Since P236R is the right station, module 15 is the closest to the conveyor. It is observed that in Scenario 2, there are unfortunately more picks at the modules located near the end of the aisles. By applying Algorithm 1, the modules with the highest number of picks are now located at the end of the rightmost aisle near the conveyor. This explains a slight reduction of 0.05% in the pickers' total travel time in Scenario 2.1 compared to Scenario 2. Due to 38 item moves to P236R, the number of picks in the station increases from 570 to 640, leading to an increase in the pick spread across stations of 2.7%. In Scenario 2.2 when all clusters are incorporated into the storage assignment, the biggest cluster is still located in P236R. This implies that no other items will be assigned to P236R due to the cluster weight balance objective of Algorithm 1. Therefore, some items in P236R are moved to other stations, i.e., the number of picks in station P236R decreases to 560.

Although the number of picks in P236R in Scenario 2.2 is even smaller than that in Scenario 2, the pick spread across stations demonstrates a growth of 17%. This results from the fact that FIP clusters are determined based on a year of data, while the simulation is conducted for only one day. Only 74 out of 6,847 boxes on `31-05-2021` are related to the biggest cluster, which implies that more boxes are involved with the items from smaller clusters on the simulated day. Despite the 17% increase in the pick spread across stations in Scenario 2.2, the improvements in other KPIs are good enough to offset it. This is not the case for Scenario 2.3 in which only two KPIs, $aSpB$ and $aPpS$, are better than in Scenario 2.2 while the number of items to be reassigned and the pick spread across stations triple. This implies that decreasing support threshold may negatively influence the warehouse performance.

The simulation outcomes may be improved if Scenario 2.3's storage assignment is based on Scenario 2.2 with a different objective function, for example, minimizing the item moves instead of using Scenario 2's storage assignment with Algorithm 1. Therefore, it raises a need to develop an improved version of the algorithm to incorporate different objective functions for various starting storage assignments. Besides moving items to available shelf spaces, the improved algorithm can consider item swapping when the free shelf spaces are too far from the conveyor. By doing so, FIP clusters can be located in the most convenient modules. This might contribute to even more significant reduction in pickers' travel time. Additionally, the simulation can be executed for different values of support threshold to identify the optimal for the warehouse performance. However, since the simulation focuses on one particular day, the optimal threshold may not be generalized, or in other words, the overfitting problem may arise.

In conclusion, Statements 5.1 and 5.2 hold for Scenario 2.2 but not Scenario 2.3. As a result, it is recommended to start integrating FIP into the existing storage assignment with a relatively high support threshold first to validate the influence of FIP on the warehouse performance in a wider date range before attempting to reduce the threshold. The threshold should not be too high such that only few frequent item pairs are detected, i.e., the FIP integration in storage assignment cannot make a difference in the warehouse performance.



(a) Scenario 2. (b) Scenario 2.1. (c) Scenario 2.2.

Figure 5.6: Pick distribution in station P236R with its number of picks in three different scenarios, namely 2, 2.1, and 2.2. This is to compare how the pick distribution in one station differs when integrating FIP into the storage assignment with the same support threshold of 50, but different number of clusters involved.

6 CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

Although order picking is the most critical operation in manual picker-to-parts warehouses, very limited literature has integrated blocking into modelling the order picking systems and studied the combined impact of zoning, storage assignment, and warehouse layout. This report focuses on two types of order picking system with block-and-recirculate protocol: *(i)* a closed cyclic queuing network with single segment from van der Gaast et al. (2020) and *(ii)* an open queuing network with star structure and multiple segments inspired by Kramp's Strullendorf warehouse. Regarding the closed network, under several assumptions, a jump-over blocking mechanism can resemble the block-and-recirculate protocol, which allows the product-form stationary distribution to be derived. As a result, the system performance metrics can be computed analytically or iteratively. Since the close network only takes into account zoning, the report extends the network to accommodate the real-life storage assignment and warehouse layout from Kramp, as well as the external arrival and departure of boxes, into an open queuing network. To that end, a simulation is developed since the stationary distribution of the network has become intractable with no exact formulas.

Two scenarios, with and without stock splitting, are built to simulate the current situation at Kramp based on five key performance indicators (KPIs), namely average stops per box $aSpB$, average picks per stops $aPpS$, average throughput time per box $aTpB$, average blocked queues per box $aBQpB$, and average recirculation per box $aRpB$. Scenario 2 without stock splitting or with single location is chosen as a base to evaluate the system performance with frequent item pairs (FIP) integration in storage assignment. This is due to the fact that Scenario 2 can resemble the current warehouse status based on the five KPIs better than Scenario 1.

For the cluster-based storage assignment, the FIP graph is constructed and the items are grouped into FIP clusters according to the connected components in graph theory. In the FIP graph, nodes represent items, and an edge between two nodes is formed when the support count of two items based on the 365-day historical transaction data reaches at least a certain support threshold. A mathematical model is formulated and a heuristic algorithm is developed to assign clusters to stations and items to shelf spaces. Three additional scenarios, three additional KPIs (total travel time of pickers, pick spread across stations, and correlation between stations' number of items and number of picks) are introduced to quantify the impact of integrating FIP clusters into the storage assignment of Scenario 2. Scenario 2.1 and 2.2 use the same support threshold of 50, yet the former involves only the largest cluster while the latter integrates all clusters in the storage assignment of Scenario 2 using Algorithm 1. Scenario 2.3 applies the storage assignment of Scenario 2 with all FIP clusters when the support threshold is 25. The expected impact is specified in two statements according to the literature survey. Statement 5.1 implies that FIP integration helps decrease $aSpB$ and increase $aPpS$, which consequently leads to a reduction in $aTpB$, $aBQpB$, and $aRpB$. Statement 5.2 proposes that FIP integration helps decrease the total travel time of pickers while increasing the pick spread across stations.

The simulation results show that given a certain support threshold, FIP integration may improve the warehouse performance. FIP integration always decreases $aSpB$ and $aPpS$ and increases pick spread across stations. An remarkable increase in pick spread across stations might lead to several negative effects including starvation in some stations while others might experience a surge in the number of picks. This results in a higher number of blocked queues and recirculation a box encounters on average, and consequently in longer box throughput time. Therefore, a careful approach must be employed and the simulation should be executed for a longer period to validate the significance of the FIP influence and to prevent overfitting when finding the optimal support threshold.

6.2 Recommendations

Based on the limitations of the analysis, the impact of the integration of FIP in storage assignment on the performance of the sequential zone picking systems would benefit from the following further research topics.

- The seasonal pattern of frequent item pairs might be examined and extracted from the transaction history for the next three-month period in the last two years instead of the last 365 days. Hence, the FIP algorithm can be reran every three months to adjust the storage assignment.
- The simulation could be extended to include the shuttle area as one segment with a single station. The shuttle area employs a parts-to-picker order picking system in

which items are moved on the conveyor from the storage area to the picking bays where the pickers are waiting to retrieve. This area has much larger storage capacity than the stations in the shelving area, i.e., the big FIP clusters could be moved to the shuttle area to better balance the workload among stations in the shelving area where the manpower is more expensive.

- Algorithm 1 might be improved to accommodate the storage assignment with FIP as the starting storage assignment, or extended to incorporate item swapping in one station so that the more frequently purchased items can be located closer to the conveyor even when all the shelf spaces there are occupied. Another potential idea is to combine the cluster-based storage assignment with a policy with better performance than the closest open location storage from Table 2.1.
- If the externally provided floor assignment algorithm by the warehouse management system is revealed and integrated into the simulation, a closer simulation result to reality might be achieved. Together with an improved box assignment of orders with frequent item pairs in the same box, the warehouse performance might be improved even further.
- The advantages of sequential and synchronized zone picking systems might be incorporated so that one order can link to multiple boxes, each of which would visit only one floor. This idea may decrease the order throughput time in exchange for lower box utilization, i.e., each box has fewer items, and more time spent on packing.

REFERENCES

Coyle, John J, C John Langley, and Edward J Bardi (1996). *The Management of Business Logistics*. West Publishing.

de Koster, René B. M. (1994). "Performance approximation of pick-to-belt orderpicking systems". In: *European Journal of Operational Research* 72.3, pp. 558–573. ISSN: 0377-2217. DOI: https://doi.org/10.1016/0377-2217(94)90423-5.

de Koster, René B. M. (2012). "Warehouse Assessment in a Single Tour". In: *Warehousing in the Global Supply Chain: Advanced Models, Tools and Applications for Storage Systems*. Ed. by Riccardo Manzini. London: Springer London, pp. 457–473. ISBN: 978-1-4471-2274-6. DOI: https://doi.org/10.1007/978-1-4471-2274-6_17.

de Koster, René B. M., Tho Le-Duc, and Kees Jan Roodbergen (2007). "Design and control of warehouse order picking: A literature review". In: *European Journal of Operational Research* 182.2, pp. 481–501. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2006.07.009.

Garfinkel, Maurice (2005). *Minimizing multi-zone orders in the correlated storage assignment problem*. Georgia Institute of Technology.

Gu, Jinxiang, Marc Goetschalckx, and Leon F. McGinnis (2007). "Research on warehouse operation: A comprehensive review". In: *European Journal of Operational Research* 177.1, pp. 1–21. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2006.02.025.

Hausman, Warren H., Leroy B. Schwarz, and Stephen C. Graves (1976). "Optimal Storage Assignment in Automatic Warehousing Systems". In: *Management Science* 22.6, pp. 629–638. ISSN: 00251909, 15265501. URL: http://www.jstor.org/stable/2629933.

Jane, Chin-Chia and Yih-Wenn Laih (2005). "A clustering algorithm for item assignment in a synchronized zone order picking system". In: *European J. Oper. Res.* 166.2, pp. 489–496. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2004.01.042.

Jia, Jing and Sunderesh S. Heragu (2009). "Solving semi-open queuing networks". In: *Oper. Res.* 57.2, pp. 391–401. ISSN: 0030-364X. DOI: https://doi.org/10.1287/opre.1080.0627.

Kelly, Frank P. (1979). *Reversibility and stochastic networks*. USA: Cambridge University Press.

Li, Jiaxi, Mohsen Moghaddam, and Shimon Y Nof (2016). "Dynamic storage assignment with product affinity and ABC classification—a case study". In: *The International Journal of Advanced Manufacturing Technology* 84.9, pp. 2179–2194. DOI: https://doi.org/10.1007/s00170-015-7806-7.

Li, Yue et al. (2021). "A heuristic storage location assignment based on frequent itemset classes to improve order picking operations". In: *Applied Sciences* 11.4, p. 1839. DOI: https://doi.org/10.3390/app11041839.

Manzini, Riccardo, Yavuz Bozer, and Sunderesh Heragu (2015). "Decision models for the design, optimization and management of warehousing and material handling systems". In: *International Journal of Production Economics* 170. Decision models for the design, optimization and management of warehousing and material handling systems, pp. 711–716. ISSN: 0925-5273. DOI: https://doi.org/10.1016/j.ijpe.2015.08.007.

Melacini, Marco, Sara Perotti, and Angela Tumino (2011). "Development of a framework for pick-and-pass order picking system design". In: *The International Journal of Advanced Manufacturing Technology* 53.9-12, pp. 841–854. DOI: https://doi.org/10.1007/s00170-010-2881-2.

Ming-Huang Chiang, David, Chia-Ping Lin, and Mu-Chen Chen (2014). "Data mining based storage assignment heuristics for travel distance reduction". In: *Expert Systems* 31.1, pp. 81–90. DOI: https://doi.org/10.1111/exsy.12006.

Mirzaei, Masoud, Nima Zaerpour, and René B. M. de Koster (2021). "The impact of integrated cluster-based storage allocation on parts-to-picker warehouse performance". In: *Transportation Research Part E: Logistics and Transportation Review* 146, p. 102207. ISSN: 1366-5545. DOI: https://doi.org/10.1016/j.tre.2020.102207.

Petersen, Charles G (2000). "An evaluation of order picking policies for mail order companies". In: *Production and Operations Management* 9.4, pp. 319–335. DOI: https://doi.org/10.1111/j.1937-5956.2000.tb00461.x.

Petersen, Charles G., Gerald R. Aase, and Daniel R. Heiser (2004). "Improving order-picking performance through the implementation of class-based storage". In: *International Journal of Physical Distribution & Logistics Management*.

Tompkins, James A et al. (2010). *Facilities planning*. John Wiley & Sons.

van der Gaast, Jelmer P. et al. (2020). "Capacity analysis of sequential zone picking systems". In: *Oper. Res.* 68.1, pp. 161–179. ISSN: 0030-364X. DOI: https://doi.org/10.1287/opre.2019.1885.

van Dijk, Nico M. (1988). "On Jackson's product form with "jump-over" blocking". In: *Oper. Res. Lett.* 7.5, pp. 233–235. ISSN: 0167-6377. DOI: https://doi.org/10.1016/0167-6377(88)90037-5.

Wang, Meng, Ren-Qian Zhang, and Kun Fan (2020). “Improving order-picking operation through efficient storage location assignment: A new approach”. In: *Computers & Industrial Engineering* 139, p. 106186. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2019.106186. URL: https://www.sciencedirect.com/science/article/pii/S0360835219306552.

Whitt, Ward (1982). “Approximating a point process by a renewal process. I. Two basic methods”. In: *Oper. Res.* 30.1, pp. 125–147. ISSN: 0030-364X. DOI: https://doi.org/10.1287/opre.30.1.125.

Yu, Mengfei and René B. M. de Koster (2008). “Performance approximation and design of pick-and-pass order picking systems”. In: *IIE Transactions* 40.11, pp. 1054–1069. DOI: https://doi.org/10.1080/07408170802167613.

Zhang, Yipeng (2016). “Correlated storage assignment strategy to reduce travel distance in order picking”. In: *IFAC-PapersOnLine* 49.2. 7th IFAC Conference on Management and Control of Production and Logistics MCPL 2016, pp. 30–35. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2016.03.006.

A FLOOR REPRESENTATION

The representation of Strullendorf warehouse is visualized in Figure A.1, A.2, and A.3 for floor 1, 2 and 3, respectively. The modules and shelves are labelled vertically and horizontally, respectively. The horizontal dashed line is used to separate the twin stations. The vertical dashed lines represent the station boundary. Each station is assigned a unique color. On each floor, the conveyor starts from the bottom left corner; therefore, stations P102R, P232R, and P362R have less shelves than other stations. The conveyor then turns right and runs along the horizontal dashed line, then turns right again at the end of stations P122R, P252R, and P382R. This explains the modules with irregular positioning in those stations. The rectangular white space equivalent to 10 modules within a station represents the stairs in order for pickers to save time while moving across floors. The two missing adjacent modules which belong to two adjacent stations represent the pillar on the floor.

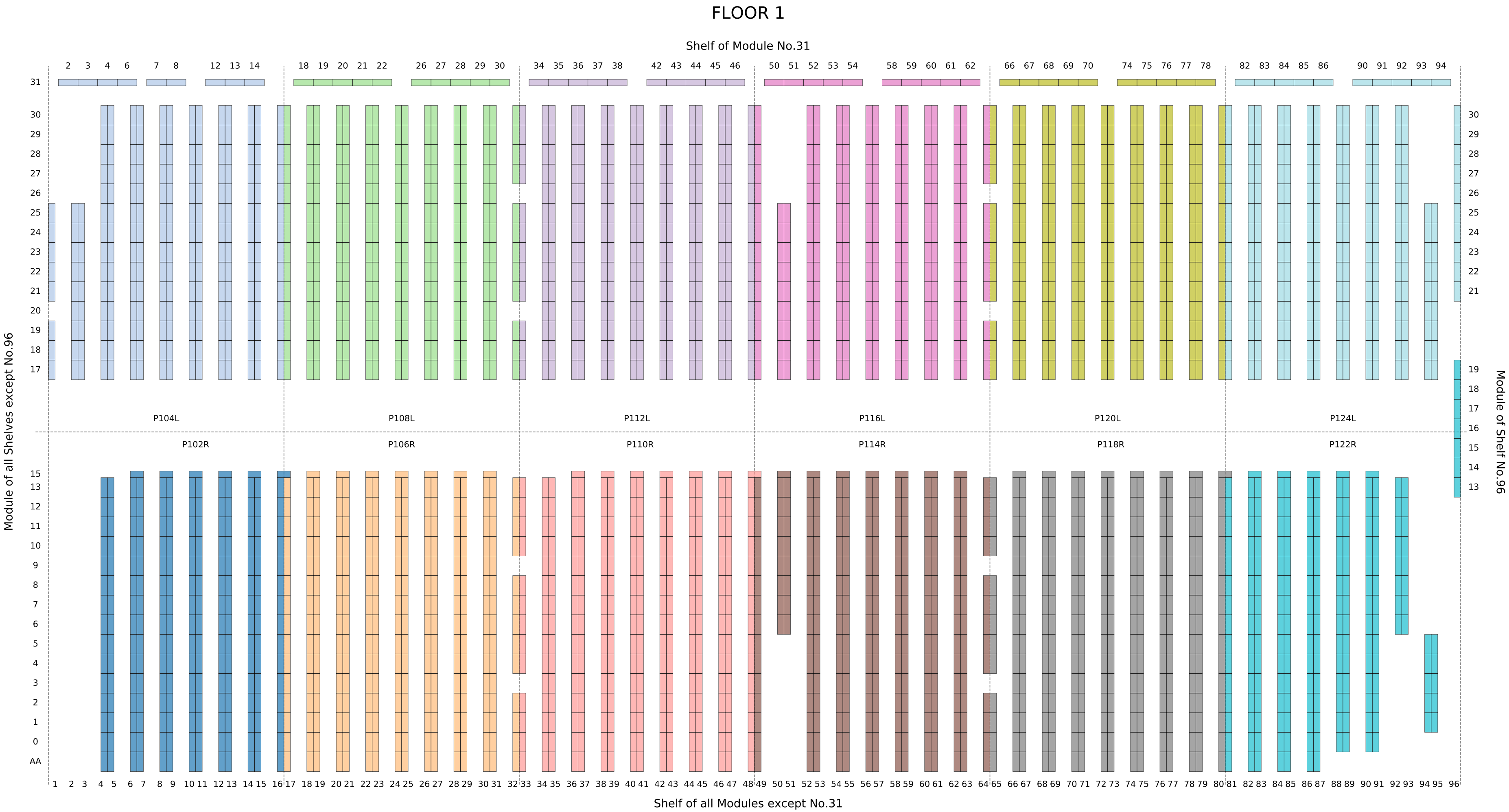


Figure A.1: The representation of the first floor in the Strullendorf warehouse.

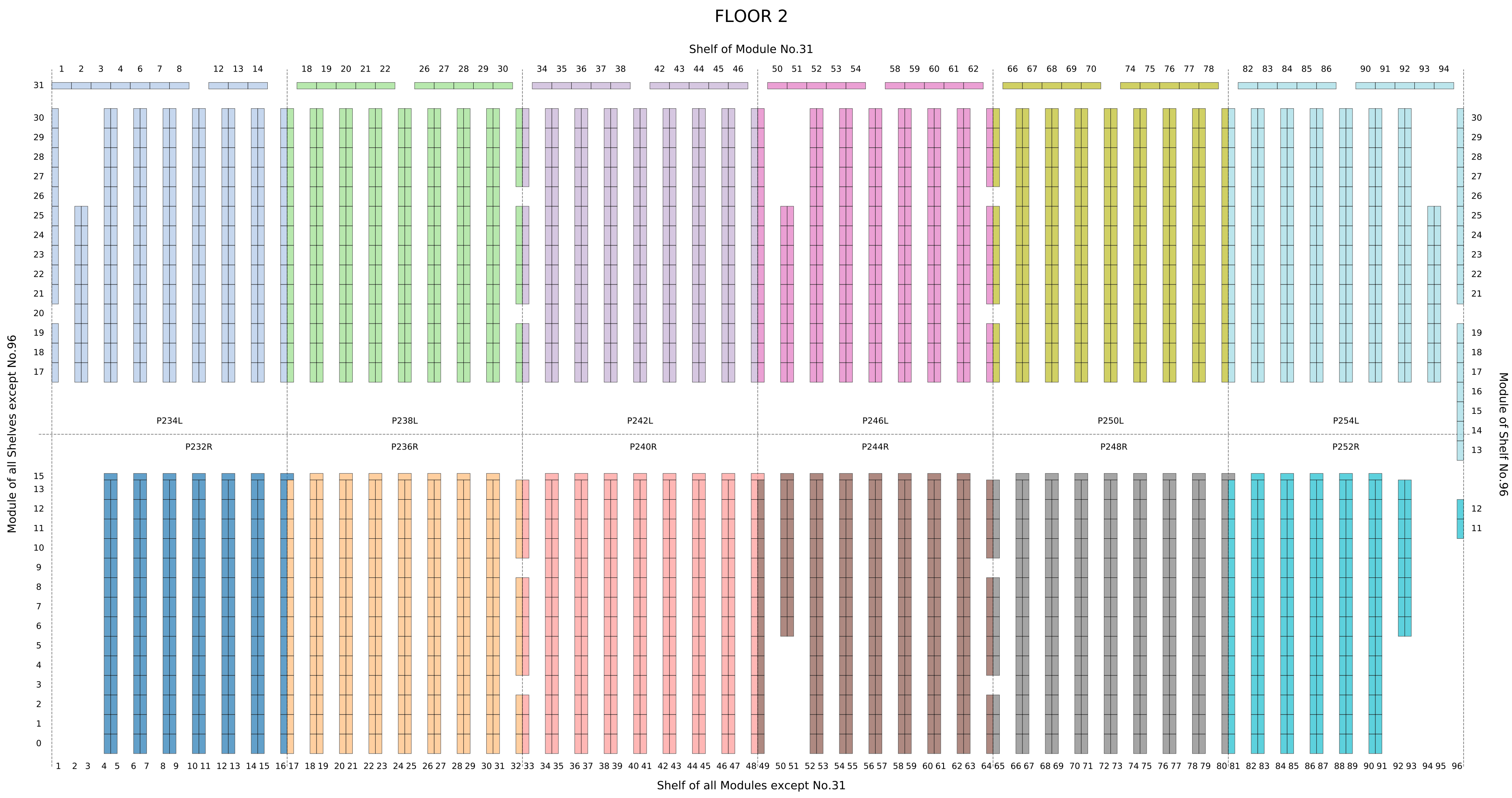


Figure A.2: The representation of the second floor in the Strullendorf warehouse.

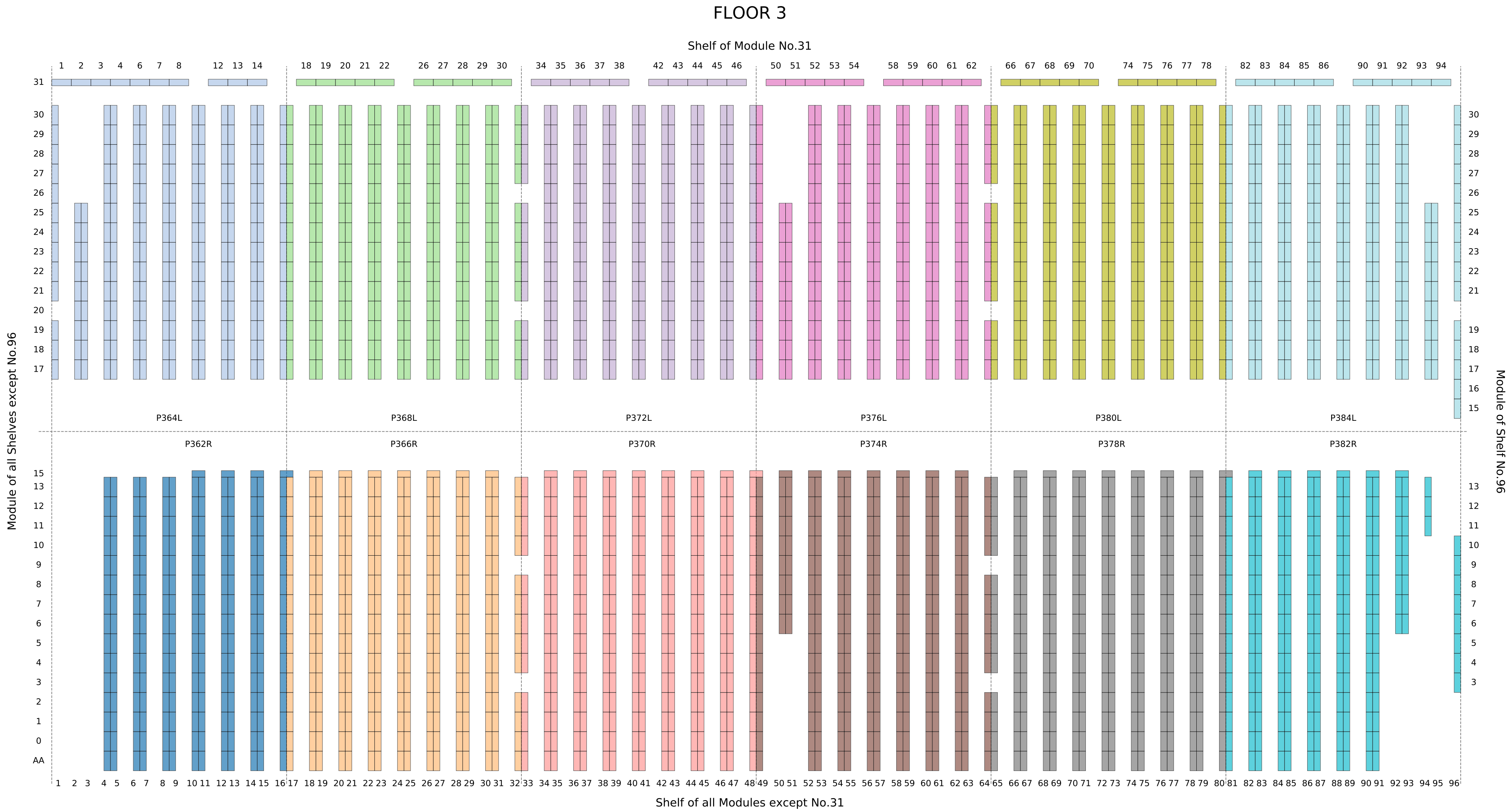


Figure A.3: The representation of the third floor in the Strullendorf warehouse.