An algorithm for dynamic order-picking in warehouse operations

Wenrong Lu[a,*], Duncan McFarlane[a], Vaggelis Giannikas[a], Quan Zhang[b,]

[a]*Institute for Manufacturing, University of Cambridge, 17 Charles Babbage Road, Cambridge, CB3 0FS, UK*
[b]*YH Global Logistics Ltd, 1201, Central Commerce Building, 88 Fuhua First Road, Fu Tian Qu, Shenzhen, 518048, P.R. China*

Abstract

Warehousing has been traditionally viewed as a non value-adding activity but in recent years a number of new developments have meant that supply chain logistics have become critical to profitability. This paper focuses specifically on order-picking which is a key factor affecting warehouse performance. Order picking is the operation of retrieving goods from specified storage locations based on customer orders. Today's warehouses face challenges for greater responsiveness to customer orders that require more flexibility than conventional strategies can offer. Hence, dynamic order-picking strategies that allow for changes of pick-lists during a pick cycle have attracted attention recently. In this paper we introduce an interventionist routing algorithm for optimising the dynamic order-picking routes. The algorithm is tested using a set of simulations based on an industrial case example. The results indicate that under a range of conditions, the proposed interventionist routing algorithm can outperform both static and heuristic dynamic order-picking routing algorithms.

Keywords: routing, dynamic picking, warehouse management

1. Introduction

Warehouse management is a long established area — both in terms of industrial practice and academic literature. The attention paid to warehouse performance (and logistics performance generally) has increased in recent years as a result of factors such as a) leaner supply chain separation requiring reduced inventory and faster warehouse response time (Agarwal et al., 2006; Naim and Gosling, 2011); b) greater number of 3PL providers managing larger, more complex warehouses with multiple customers with widely varying requirements (Selviaridis and Spring, 2007; Tian et al., 2010); c) the rapid rise in the number of online retailer transactions, in which orders to the end customer are managed directly by the

---

*Corresponding author. Email: wl296@cam.ac.uk, Telephone: +44 (0)1223 765910, Institute for Manufacturing, University of Cambridge, 17 Charles Babbage Road, Cambridge CB3 0FS, UK

warehouse (Gong and De Koster, 2008; Davarzani and Norrman, 2015).

Across the various operations in a warehouse, order-picking is the most time consuming operation in general (Roodbergen and Koster, 2001) and accounts for around 55-75% of total warehousing costs (Chiang et al., 2011). Therefore, order-picking has the highest priority for productivity improvement (De Koster et al., 2007). The order-picking operation is particularly important in manual picker-to-parts picking systems[1], which are the most common ones (Gong and De Koster, 2008) and account for over 80% of all order-picking systems in Western Europe (De Koster et al., 2007). In a picker-to-parts system, orders are firstly batched to form a pick-list. The list then guides the order-picker to travel along the aisles with a picking device (e.g. picking cart or fork-lifter) and collect requested items from designated storage locations (storage racks or bins) (De Koster et al., 2007).

Although many studies have been conducted on improving the order-picking operation, managing it efficiently remains complex (Gong and De Koster, 2008). On the demand side, the complexity arises from the introduction of new sales channels such as on-line shopping; and on the supply side, it arises from new operating programs, e.g. JIT, cycle-time reduction (Tompkins, 2010; Davarzani and Norrman, 2015). In such novel business models, customers can place an order by a click of the mouse in their computer, expecting inexpensive, rapid and accurate delivery (De Koster, 2003), i.e. they tend to order more frequently but in smaller quantities asking for more customised service. In response, more companies are inclined to accept late orders which leads to tighter windows for timely deliveries (Gong and De Koster, 2008). Moreover, many logistics companies are replacing small warehouses by fewer but larger warehouses to realise the economies of scale (De Koster et al., 2007). Consequently, the time available for order picking is increasingly shorter. Hence, a fast response is critical for warehouses to operate in such a complex environment (Otto and Chung, 2000).

Conventional, Static Order-Picking (SOP) requires batch formation to generate static pick-lists, which is time-consuming and insufficient to cope with the rising number of daily orders and the decreasing lead time (Gong and De Koster, 2008). In order to shorten the response time, Dynamic Order-Picking (DOP) systems that allow for changes of a pick-list during a pick-cycle have been introduced. Such systems can also be useful for managing disruptions occurring in a warehouse. For example, disruptions occurring from the arrival of an urgent order or by noticing inaccuracies between items requested and items actually picked. These systems use constantly and selectively updated pick information of newly arrived orders as well as information regarding the status of the warehouse operations. A

---

[1]The reader is referred to De Koster et al. (2007); Tompkins (2010) for detailed reviews on order-picking systems

DOP system capable of updating pick-lists during execution has been previously proposed by Gong and De Koster (2008) which, however, constrains the picker to travel on a heuristic (non optimal) route and limits the assignment of new orders depending on a picker's current location. As a result, it has been suggested that the performance of such a system can be significantly improved by determining the new picking route in an optimal way (Gong and De Koster, 2008).

Motivated by this, in this paper we present an Interventionist Routing Algorithm (IRA) to address the problem of dynamic picking route optimisation. The algorithm calculates the route of minimised distance an order-picker should follow, upon receiving updated information on the requested items of new orders during the picking operation. Moreover, we investigate potential benefits of using such an algorithm in comparison to a conventional static optimal algorithm and dynamic heuristic algorithm, in terms of the average order completion time and average travel distance per order. The algorithm forms a fundamental part of a DOP, the authors have introduced in previous papers (Lu et al., 2014; Giannikas et al., 2013) and treats new orders as external disruptions and incorporates distributed intelligence to manage them.

The paper is organised as follows: Section 2 provides a the background to order-picking operations. Section 3 introduces the routing problems for both conventional and interventional order-picking, clarifies the terminology, and introduces the notations and important definitions for supporting the dynamic extension of a static optimal algorithm originally proposed by Ratliff and Rosenthal (1983). Section 4 proposes the interventionist routing algorithm and provides an illustrative example. Section 5 presents and discusses the results of applying the algorithm in an industrial scenario. Lastly, the paper concludes with a summary and an outlook on further research topics.

2. Background

In this section, we provide the necessary background on warehouse order picking. We begin by introducing the order-picking problem in warehouse operations and then focus on routing methods as one of the methods that reduce picking travel-time. Finally, we discuss approaches that can be used in order picking in non-static settings.

2.1. Order Picking

An order-picking operation is the operation of retrieving goods from specified storage locations on the basis of customer orders (Roodbergen and Koster, 2001). The most common objective of any order-picking system is to maximise the service level (i.e. factors like the average and variation of order delivery time, order integrity and accuracy) subject to resource

constraints such as labour, machines and capital (Gu et al., 2010). In fact, order-picking is arguably a crucial link to other services in warehousing since it can significantly affect the point of time an order is available to be shipped to the customer (De Koster et al., 2007). If an order is late for its outbound transportation due time, it may have to wait for the next transportation service, thus affecting the whole transportation schedule (Goetschalckx, 1989). Hence, De Koster et al. (2007) emphasise that *"minimising the order retrieval time is a need for any order-picking system"* highlighting the fact that the shorter the *order retrieval time*[2], the higher the flexibility in handling late changes in orders.

Many activities contribute to the order retrieval time, including travelling between storage locations, item searching, item picking, equipment setting-up etc. (Dekker et al., 2004). Among them, *travelling* is the dominant component which accounts for more than 50% of the total retrieval time (Tompkins, 2010). Furthermore, travel time is considered a waste since it costs labour hours and adds no value (Bartholdi and Hackman, 2010). Therefore, in order to improve the performance of order-picking, reducing travel time is critical. Since the travel distance is proportional to travel time for picker-to-parts system (Petersen and Aase, 2004), minimising the travel distance (total or average) of a picking tour is often considered as imperative factor to reduce travel time and consequently improve warehouse operation efficiency (De Koster et al., 2007).

There are four methods to reduce the travel distance of an order-picker (Petersen II, 1999; De Koster et al., 2007): *i*) storage location assignment, *ii*) warehouse zoning, *iii*) order batching and *iv*) pick-routing methods. This paper will focus on the pick-routing methods with an in-depth review in the next sub-section.

2.2. Routing Methods in Order-Picking

This well-researched problem consists of the determination of the optimal sequence and route to pick up a set of requested items, subject to the layout of the warehouse and working methods (Van Nieuwenhuyse and De Koster, 2009). Optimisation is often achieved by minimising the total travel distance or time of the picker(s) or minimising the total material handling costs.

In practice, heuristic procedures are widely applied for the their simplicity and flexibility (De Koster et al., 2007). Several heuristics procedures for single-block warehouses exist (for a detailed review, see Petersen II (1997); De Koster et al. (2007)). Among them, the *largest gap* heuristic procedure outperforms the others when the number of products in a pick-list is small (Petersen II, 1997). However, despite the simplicity and flexibility of the

---

[2]The time needed for the items of an order to be picked from the designated storage locations

heuristics procedures to most order-pickers, optimisation algorithms are still the centre of routing studies since they result in less travel distance (Petersen II, 1999).

To pursue the optimal or near-optimal order-picking route in a typical rectangular, multi-parallel-aisle warehouse, the order-picking routing problem is classified as the Steiner Travelling Salesman Problem (STSP) (De Koster et al., 2007; Theys et al., 2010). There are two general methods to solve the STSP (Theys et al., 2010): the first one is to reformulate a STSP into the classic TSP by computing the shortest paths between every pair of required nodes, by which many well-studied TSP heuristics can be applied to STSP, e.g. Makris and Giakoumakis (2003); Renaud and Ruiz (2007); The second method for the solution of a STSP is by using exact (dedicated) algorithms.

The latter method is preferred to the former although it can only be applied to a rectangular warehouse layout with less than three blocks, for the following reasons:

- The original graph, which represents the warehouse layout and pick locations, may be easier to solve than the TSP on a complete graph (Cornuéjols et al., 1985), especially in warehouse situation where the pickers can only travel along the aisle according to a fixed layout;

- Even if the original graph is sparse, the reformulation to a classic TSP instance increases the number of variables substantially (Letchford et al., 2013);

- An exact algorithm for a series-parallel warehouse does guarantee optimality whereas heuristics for the class TSP do not;

- The output of an exact algorithm is usually a route map. In comparison, the output of algorithms for the classic TSP is usually a sequence of nodes that indicate the order of locations to be visited. However, since there are two possible paths between any pair of nodes in a warehouse, it requires an extra step to convert the sequence into a route map.

Most exact algorithms available currently are based on a graph theoretic algorithm for single-block warehouses that was published in a seminal paper by Ratliff and Rosenthal (1983). Since then, many studies extended the capability of this algorithm in order to solve the STSP in all series-parallel graphs (Cornuéjols et al., 1985), solve the situation where the warehouse cannot be represented by series-parallel graphs (De Koster and Van Der Poort, 1998; Roodbergen and Koster, 2001; Roodbergen and De Koster, 2001), use it in warehouses with two blocks (Roodbergen and Koster, 2001), wide aisles (Goetschalckx and Ratliff, 1988) or warehouses that store similar items at multiple locations (Daniels et al., 1998).

As new information technologies become available to allow information to be conveyed in real time to a picker who is on-site, a different approach to improving order picking

is by dynamically changing the pick-list during the picking operation i.e. during a pick-cycle. In this way, the picker can be informed about new (urgent) orders before returning to the depot for his next tour. Also, up to now, most of the studies on the exact optimal algorithms assume that the picker starts and ends a picking tour in the same depot location and without intervention. It is obvious that these algorithms are not applicable if the pick-list is dynamically changing (rather than remaining static) since they require the picker to pick all items in his current list and return to the depot before new orders can be assigned to him. An algorithm that could be used for allowing pick-list updates while the picker is not at the depot has been presented in De Koster and Van Der Poort (1998). However, it is also not applicable for the dynamic picking problem we are considering for two reasons: a) the algorithm was developed for the purpose of decentralised depositing rather than picking, therefore the picker might finish his pick-cycle on a different location compared to the location which the picker started the pick-cycle from; b) the algorithm facilitates the update of a pick-list only when the picker is located at the head of an aisle, whereas in the problem under consideration, routing methods need to use updated pick-lists regardless of the picker's location.

The issue of dynamically updating a pick-list has been previously examined in (Gong and De Koster, 2008). In their work, the authors use stochastic polling theory to analyse the benefit of a picking system in which the picking information can dynamically change during a pick-cycle In order to calculate the updated route of a picker in this system, a heuristic procedure, that can lead to non-optimised results, is used. The authors notice that the problem of optimal routing in such systems can be of particular interest and we will explore it in the following sections.

3. The Interventionist Order-picking Problem

In the previous section we discussed ways to improve the performance of the order picking operation and we identified a need for an optimal routing algorithm for pickers using dynamic picking lists. In this section, we state the routing problems for both conventional and interventionist order-picking, we clarify the terminology, and we introduce the notations and important definitions for supporting the extension of the algorithm from Ratliff and Rosenthal (1983).

3.1. Problem Statement

The order-picking system considered here is based on a rectangular warehouse that contains a number of parallel aisles with equal length, as shown in Figure 2a. An order-picker uses the aisles to pick up requested items stored in the storage locations which are located

at both sides of the aisles. We only consider "narrow" aisles which allow picking of items from both sides of the aisle without causing extra travelling distance or without changing position (De Koster et al., 2007). Order-pickers will travel with a picking device (e.g. cart, roll pallet, etc.), therefore multiple customer orders can be combined until the capacity of the picking device is exhausted. Aisles can be transversed in both directions by a picker and can allow a change of direction within an aisle (De Koster and Van Der Poort, 1998). Moreover, we consider here a single-picker setting in an one-block warehouse, i.e. there are only two cross-aisles, located at the front and back of the warehouse for the order-picker to travel from one aisle to another. Notice that cross-aisles do not contain any storage locations.

The order-picking system in this paper is a manual picker-to-parts system where order-pickers are guided by pick-lists. The order-picker starts the picking process at the depot, travels through the warehouse and picks items from associated storage locations according to the assigned pick-list. Upon completion of the picking, the order picker returns to the depot to unload the picked items. The pick-list is generated based on the customer orders/requests (with typically one or two order lines), which may arrive anytime during the work-shift. That is, during a work-shift, not all customer orders/requests are known in advance, and may become available over time. Van Nieuwenhuyse and De Koster (2009) define such operating environment as a 'stochastic setting'.

Conventionally, order-picking can only be scheduled based on the known customer orders, so an intervention of the process and a rearrangement of the orders is prohibited if an order picker has started a pick-cycle through the warehouse. Therefore, the routing problem for Static Order-Picking can be described as the following:

Problem 3.1. Routing Problem for Static Order-Picking (SOP)

Given a) a set of storage locations to be visited (listed on the pick-list), and b) a known starting and finishing location (depot), what route should the order-picker follow in order to minimise the distance travelled?

In comparison to conventional order-picking, interventionist order-picking allows the re-arrangement of orders even the order-picker has started a pick-cycle. The process can be described as follows: an order-picker starts the picking operation with an initial pick-list of existing orders. The order-picker does not need to wait for the number of requested products to saturate the capacity of the picking-device in order to start a pick-cycle. This is due to the fact that the pick-list can be updated and rearranged if new orders arrive. Once a new order arrives, the picking process is intervened by adding the new order to the pick-list currently being executed. The location of the order-picker at the time of intervention can be anywhere inside the warehouse. Such pick-lists should typically be displayed on a digital screen (e.g. on an electronic handheld device or a laptop mounted on the picking device), to allow their

dynamic update and the display of the new route.

Therefore, the routing problem for Interventionist Order-Picking can be described as the following:

Problem 3.2. Routing Problem for Interventionist Order-Picking (IOP)

Given a) a revised set of storage locations to be visited (listed on the updated pick-list), and b) an arbitrary current location of the order-picker, what route should the order-picker follow in order to minimise the distance travelled?

Hence, for a single pick cycle, a picker would face Problem 3.1 once —while at the depot— but Problem 3.2 many times potentially —every time a new order is received.

3.2. Notation and Definitions

This section introduces definitions and notation which will be used in the development of an interventionist routing algorithm for order-picking. As this development extends the static algorithm development by Ratliff and Rosenthal (1983), we use similar notation to that work.

Notation

In the developments that follow, we will use the following notation for the model variables:

G: the graph of the warehouse layout and pick locations

m: the number of items to be picked at the moment

n: the number of aisles in the warehouse.

i: index number of each requested item

v: a location in the warehouse, where v_i represents the storage locations for each requested item (with $i = 1, 2, 3, \ldots, m$), v_0 represents the location of depot, and v_p represents the location of the order-picker;

j: index number of each aisle

j_p : index number of the aisle the picker is currently located at

j_d : index number of the aisle of the depot

f: index number of the first pick-aisle[3]

r: index number of the last pick-aisle

a_j, b_j: back (top), front (bottom) endpoints of each aisle

A_j: the subgraph of G which only consists a_j and b_j, and everything within aisle j.

---

[3]By the term "pick-aisle" we refer to an aisle that contains at least one requested item.

L_j: Partial Route Subgraph including aisles $1, \ldots, j$. It consists of two parts, L_j^- and L_j^+. The former includes endpoint vertices a_j and b_j together with everything to the left of a_j and b_j. The latter includes endpoint vertices a_j and b_j and all vertices v_i within aisle j, together with everything to the left of a_j and b_j, i.e. $L_j^+ = L_j^- \bigcup A_j$. This is graphically illustrated in Figure 1.
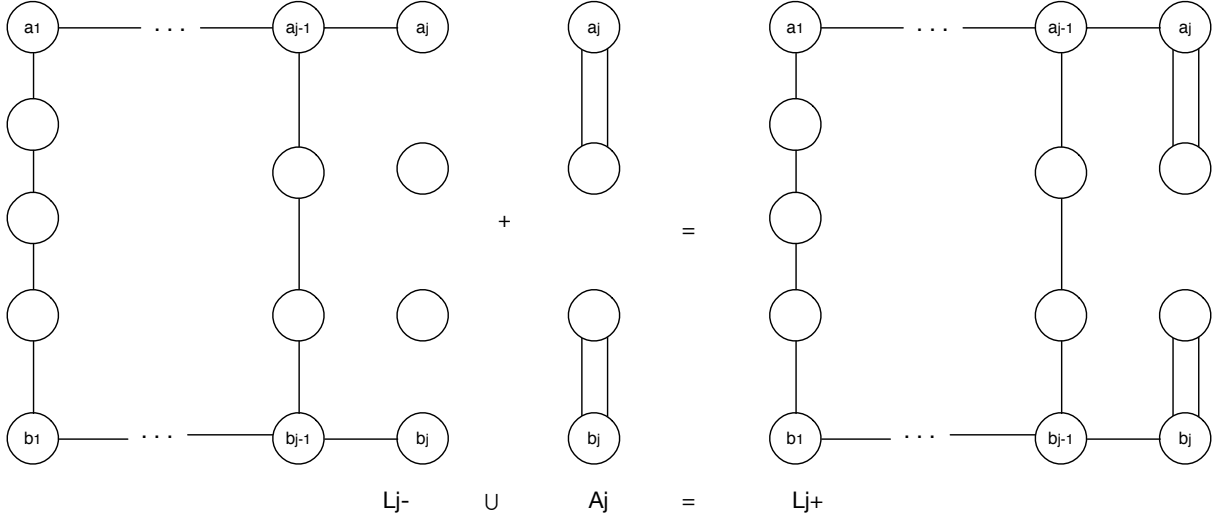

Figure 1: Partial Route Subgraphs

Warehouse Model

The warehouse along with the associated pick locations is modelled here as a graph in (Ratliff and Rosenthal, 1983). Let m denote the number of items to be picked at the moment, and n the number of aisles in the warehouse. We can then define a vertex v_i with $i = 1, 2, 3, \ldots, m$ as each requested item in the associated storage location, and vertices a_j and b_j as the back and front endpoints[4] of each aisle $j = 1, 2, 3, \ldots, n$ [5]. Connecting any two vertices that are associated with adjacent locations in the warehouse, we define graph G to represent the warehouse layout and pick locations in Figure 2a as a graph representation shown in Figure 2b. Each connection represents an undirected and unlimited number of parallel arcs, of which length indicating the direct distance between the two locations. Lastly, the depot is modelled by introducing a depot vertex v_0, with zero length to the endpoints of the corresponding aisle. For instance, since the depot is on aisle 4 in Figure 2a, the depot vertex v_0 is added between a_4 and b_4 with zero length between v_0 and b_4, as shown in Figure 2b.

---

[4]The intersections of an aisle and a cross-aisle

[5]The reader is referred to Appendix A for more details on the numbering system of the aisles

(a) Order-picking layout in a warehouse (b) Graph representation of the order-picking layout
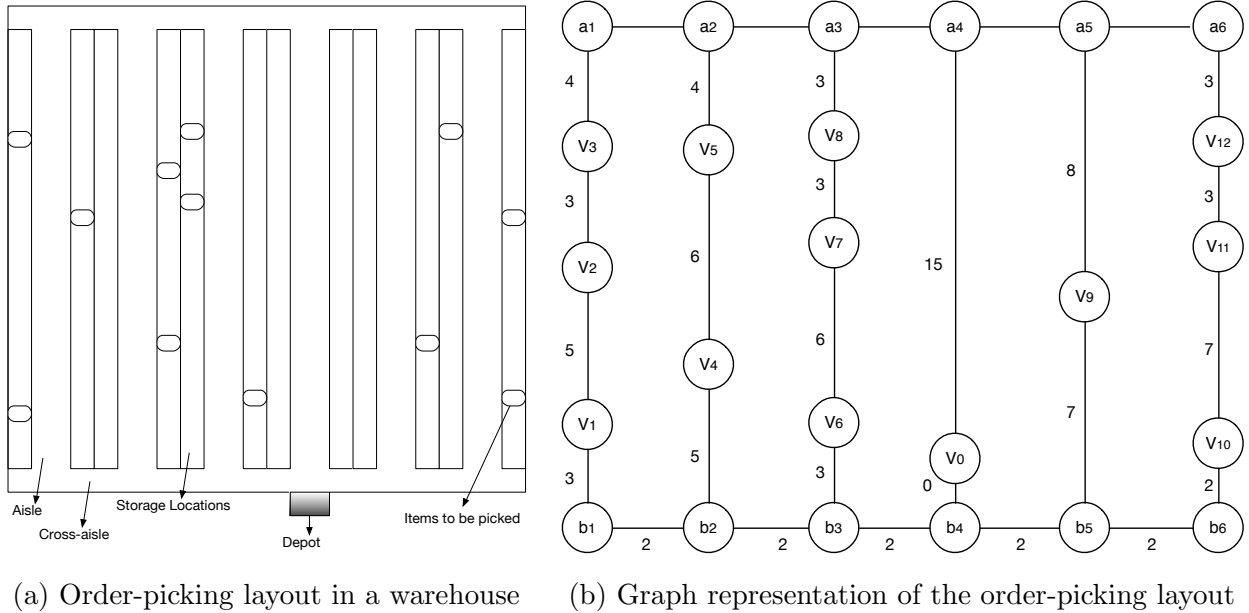
Figure 2: Modelling the order-picking layout

Definitions

Route Subgraph: denoted as L, refers to a possible order-picking route that is a subgraph in the warehouse graph G, which includes each of the vertices v_i for $i = 0, 1, 2, \ldots, m$ at least once. Then the shortest order-picking route is a route subgraph with minimum length.

Partial Route Subgraph (PRS): for any Route Subgraph $L \subset G$, is denoted $L_j \subset L$ for $j = 1, 2, \ldots, n$, that consists of all the vertices (v_i with $i = 1, 2, \ldots, m$) and aisle endpoints (a_j and b_j with $j = 1, 2, \ldots, n$,) to the left of aisle j in G, such that the minimum length route subgraph is obtained by exhaustive searches of all the possible combinations of PRS.

Equivalence class for PRS: Each possible combination of L_j PRS (for $j = 1, 2, \ldots, n$) is classified based on the triple of (degree parity of a_j, degree parity of b_j, and connectivity of the PRS). The possible degree parities can only be zero(0), uneven(U) or even(E), and the connectivity is categorised as 0, 1 or 2 components (denoted as 0C, 1C or 2C respectively).

4. An Interventionist Routing Algorithm

Having established the routing problem for IOP, in this section, we propose an interventionist routing algorithm (IRA) for this problem. We first provide an overview of IRA in

terms of the criteria it needs to address as well as the way IRA addresses them. We then describe the elements of the algorithm and we close this section with an illustrative example that demonstrates the usage of IRA.

4.1. Overview of Algorithm

Criteria to be Addressed by an IOP Algorithm

In order to solve the routing problem for IOP, an algorithm that addresses the following four criteria is needed:

- *Interruptibility*: the ability to re-route the picker when new orders are added to the current pick-list while the picker is on-site. This implies an arbitrary starting point of the route but still requires the order-picker to pick all the requested items in the pick-list and deposit them to the depot.

- *Optimisation*: the resulting travel distance of the order-picking route aims to be minimum;

- *Complexity*: the computation time should be polynomial and also linear to the number of pick-aisles;

- *Centralised depositing*: the order-picker will always start and finish a pick-cycle at the same position which is determined by the pre-defined depot.

Key Features of Proposed Algorithm

In order to address the criteria of optimisation, complexity and centralised depositing, this algorithm was developed based on the optimisation algorithm in Ratliff and Rosenthal (1983), which determines the shortest order-picking route from all possible combinations of partial route subgraph (PRS) of each pick-aisle. However, to address the criteria of interruptibility, the starting location of the picking route is required to be any location in a warehouse. As a result, in order to extend Ratliff and Rosenthal (1983)'s algorithm to accommodate starting from an arbitrary point in the warehouse, the following extensions to that algorithm have been made:

1. **Allow the picker starting location to be inside an aisle**: this has led to new initiation procedures for the algorithm, and new possible arc configurations were identified for a picker to leave an aisle;

2. **Introduce a new travelling area**: the proposed algorithm makes use of two types of travelling areas, One-Way (OW) and Round-Trip (RT) area, were defined based on the picker's location relative to the depot and all other picked items locations. The RT area was studied by Ratliff and Rosenthal (1983);

3. **Add new PRS equivalence classes**: in addition to the six PRS equivalence classes proposed for RT area in (Ratliff and Rosenthal, 1983), we have added 7 new PRS equivalence classes, in which 6 of them are proposed for the newly introduced OW area, and the rest is added to RT area;

4. **Add new route construction tables**: in order to accommodate the new travelling ares and PRS equivalence classes, 5 more tables for the route construction procedures were developed, and the 2 tables from (Ratliff and Rosenthal, 1983) were modified by adding one row each.

In addition to these extensions, in this paper, we also provide detailed guidance for applying the algorithm, including the initiations, calculation of the transitions, selection of tables, and finishing the route construction.

4.2. Algorithm Development

This sub-section will describe IRA in detail. Firstly, we introduce the basic definitions with some of them being modified from Ratliff and Rosenthal (1983). Based on these definitions, we then summarise the steps for constructing the route. Finally, in this sub-section, we discuss the optimality, complexity and running time of the algorithm.

4.2.1. Basic Definitions

Since we intend to enable the intervention by updating newly received orders to the order-picker during the picking operation, the starting position of the route, which is determined by the current location of the order-picker, could be anywhere in the warehouse. Hence the route subgraph L —in comparison to Theorem 1 in Ratliff and Rosenthal (1983)— is no longer a *cycle* in G, and therefore it is re-defined as:

Definition 4.1. Route Subgraph: a subgraph $L \subset G$ is a Route Subgraph for the IOP problem if and only if:

1. vertex v_p and all vertices v_i for $i = 0, 1, 2, \ldots, m$ have positive degree in L;

2. vertices v_p and v_0 both have odd degrees since L is not a cycle;

3. excluding vertices v_p and v_0, every other vertex in L (including a_j and b_j for $j = f, f + 1, \ldots, r - 1, r$) has either an even or zero degree.

Given a route subgraph, the algorithm will calculate the minimum length PRS (partial route subgraph) for every pick-aisle j for $j = f, f + 1, \ldots, r - 1, r$. Such a PRS can be described as follows:

Let L be any subgraph of G, a subgraph $T_j \subset L$ is a L *Partial Route Subgraph (PRS)* if C_j, a *completion* of T_j, exists. C_j is a subgraph of $G - L$ (the graph remains after deleting

12

all edges in L from G) such that $T_j \cup C_j$ is a route subgraph of G. Therefore, the necessary and sufficient conditions for $T_j \subset L_j$ to be an L_j PRS for the IOP problem are the following:

Definition 4.2. Partial Route Subgraph $T_j \subset L_j$ is a L_j Partial Route Subgraph (PRS) if and only if:

1. for all $v_i \in L$, their degrees are all positive in T_j;

2. the start and end vertices, and possibly for a_j and b_j have odd degrees, every other vertex in T_j has even or zero degree;

3. excluding vertices with zero degree, T_j has either no connected component, a single connected component connecting to either or both a_j or/and b_j, or two connected components of which one is connecting to a_j and the other is connecting to b_j.

In comparison to Theorem 2 in (Ratliff and Rosenthal, 1983), the above definition differs in condition (2) which emphasises that the degrees of both the start and the end vertices are odd, such that the sum of the degrees of all vertices in T_j remains even. Therefore, the proof of Theorem 2 can still be applied here.

Suppose there are two PRSs T_j^1 and T_j^2 of L_j, we define them to be equivalent if each completion $C_j \subset G - L_j$ of T_j^1 is also a completion of T_j^2. i.e. $T_j^1 \cup C_j$ is a route subgraph and $T_j^2 \cup C_j$ is also a route subgraph. Therefore, from Theorem 3 of (Ratliff and Rosenthal, 1983):

Definition 4.3. Eqiuvalent Partial Route Subgraph(PRS) For two PRSs T_j^1 and T_j^2 of L_j to be equivalent if and only if:

1. Both a_j and b_j have the same degree parity (i.e. even, odd or zero) in T_j^1 and T_j^2;

2. excluding vertices with zero degree, both T_j^1 and T_j^2 have the same number of connectivity component (i.e. no connected component, a single connected component connecting to either or both a_j or/and b_j, or two connected component of which one is connecting to a_j and the other is connecting to b_j.

We can classified all the L_j PRSs by such equivalency and denote them as *Equivalence classes for PRS* which was defined in section 3.2. However, these equivalence classes for each aisle j are different based on the its relationships with the route starting position and the depot location. i.e. The equivalence classes of aisle j can only be determined by the areas that the associated j aisle is located in as depicted in Figure 3, hence we define the areas as the *travelling areas* as follows:
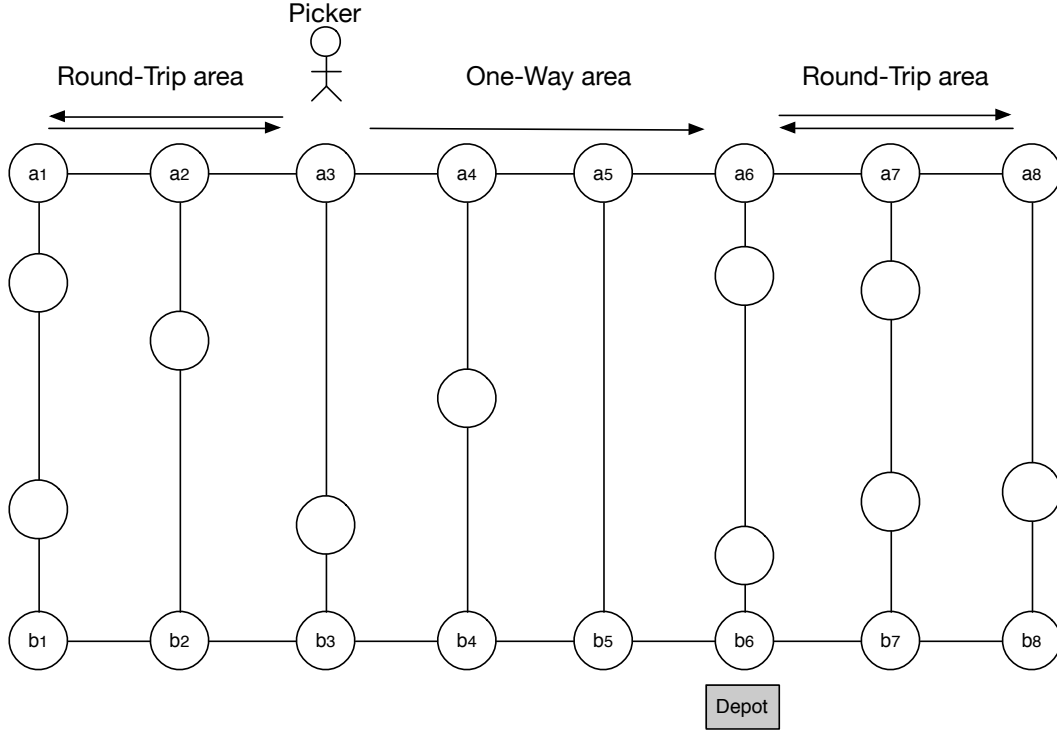
Definition 4.4. Travelling Areas

Figure 3: Categories of aisles

- *One-Way(OW) travelling area*, is an area that the picker can travel through in either a forward or backward manner but not both. If the total number of arcs on cross-aisles in any possible PRS between two adjacent aisles $j-1$ and j is odd, the area between aisle $j-1$ and j is a One-Way travelling area. For example, referring to Figure 3, the area between aisle 3 and 6 is a One-Way travelling area.

- *Round-Trip(RT) travelling area*, is an area that the picker will travel through in both forward and backward manner. If the total number of arcs on cross-aisles in any possible PRS between two adjacent aisles $j-1$ and j is even, the area between aisle $j-1$ and j is a Round-Trip travelling area. In Figure 3, the areas between aisles 1 and 3 and between aisles 6 and 8 are Round-Trip travelling areas.

In a One-Way travelling area, because the picker will only travel through the aisle j once, there is always one and only one odd degree parity in either a_j or b_j of the L_j PRS. Hence, without loss of generality, we can conclude the following:

Observation 1. *The only possible equivalence classes for any possible L_j PRS in a OW area are: (U, 0, 1C), (0, U, 1C), (E, U, 1C), (E, U, 2C), (U, E, 1C), and (U, E, 2C) where, for example, (U, E, 1C) means this PRS has an Uneven degree parity of a_j, Even degree parity of b_j, and the connectivity of 1 Component.*

14

In a Round-Trip area, the sum of the degrees of a_j and b_j must be even, i.e. if there is an odd degree parity of a_j, the corresponding b_j must have an odd degree. If none of the endpoints has odd degree, there is at least one even degree parity in a_j or/and b_j of the L_j PRS. Therefore, without loss of generality, we can conclude the following:

Observation 2. *The only possible equivalence classes for any possible L_j PRS in a RT area are: (U, U, 1C), (U, U, 2C), (0, E, 1C), (E, 0, 1C), (E, E, 1C), and (E, E, 2C).*

In comparison with Corollary 3.1. in (Ratliff and Rosenthal, 1983), we have added the class *(U, U, 2C)* since the Round-trip PRS in interventionist routing may be constructed from One-Way area where the two components of PRS haven't yet been merged into one. Similarly, according to (Ratliff and Rosenthal, 1983), we also omitted the classes:

- *(0, 0, 1C)*, because this class only applies when the length of the aisles and cross-aisles are not the same, which is not the warehouse layout considered in this paper.

- *(0, 0, 0C)*, because this means that none of the aisles in L_j contain an item to be picked. Hence it will always be the equivalence class for L_j^- where j is the first aisle that contains at least one item to be picked. i.e. The PRS construction in the algorithm proposed in this paper will always starts from L_j^+.

4.2.2. Algorithm Description

The three main stages for constructing the optimal picking route using the Interventionist Routing Algorithm (IRA) —initiation, transitions, and route constructions— along with the necessary arc configurations, are explained in Appendix A. The detailed procedure of applying the algorithm is omitted from this paper for length reasons but is documented in its entirely in (Lu et al., 2015). Here, we simplistically summarise IRA in the steps of Algorithm 1 and then graphically in Figure 4.

Algorithm 1: Interventionist routing algorithm (IRA)

Graph model conversion: Convert the depot, current location of the order-picker, and all requested items on the pick-list into the graph model;

Identify different travelling areas: Define the OW and RT areas based on the depot, the storage locations of the requested items and the current location of the order-picker;

if v_p *is within an aisle* **then**

    Assuming j_p is empty **begin**

        TempA1 = Route_Length_Calculation(j_p, f, r, j_d, a_{j_p});

        TempB1 = Route_Length_Calculation(j_p, f, r, j_d, b_{j_p});

    TempA2 = Route length between v_p and a_{j_p} according to required items in j_p;

    TempB2 = Route length between v_p and b_{j_p} according to required items in j_p;

    if *TempA1 + TempA2 < TempB1 + TempB2* **then**

        Choose a_{j_p} as the end-point to exit the aisle j_p;

    else

        Choose b_{j_p} as the end-point to exit the aisle j_p;

else

    Route_Length_Calculation(j_p, f, r, j_d, v_p);

Construct the overall picking-route: Construct the picking route for L_j^+ and L_j^- for all $j = f, f + 1 \ldots, r - 1, r$ based on the arc configurations indicated in the last step of the Procedure;

Procedure: *Route_Length_Calculation(j_p, f, r, j_d, v_p)*

$j = f$;

Determine the PRS equivalence classes for L_j^+: Based on all feasible arc configurations (Figure A.1) and on the vertices within this aisle, calculate the length of PRS for each equivalence class of L_j^+;

while $j < r$ **do**

    $j = j + 1$;

    Calculate the PRS equivalence classes for L_j^-: Based on the travelling area of aisle $j - 1$ and j, choose the appropriate transition table and apply the corresponding arc configurations (from Figure A.2) to the associated PRS equivalence class of L_{j-1}^+. Then among all feasible combinations for each equivalence class of L_j^-, choose the PRS with the minimum length as the representative one and record this combination;

    Calculate the PRS equivalence classes for L_j^+. Based on the travelling area of aisle j, choose the appropriate transition table, apply the corresponding arc configurations (from Figure A.1) to the associated PRS equivalence class of L_j^-. Then among all feasible combinations for each equivalence class of L_j^+, choose the PRS with the minimum length as the representative one record this combination;

Determine the PRS classes for each aisle and cross-aisle: Choose the PRS equivalence class of L_r^+ with the minimum length. Then follow the trace indicated by this equivalence class backward to determine all the classes for each L_j^+ and L_j^- where $j = r, r - 1, \ldots, f + 1, f$;
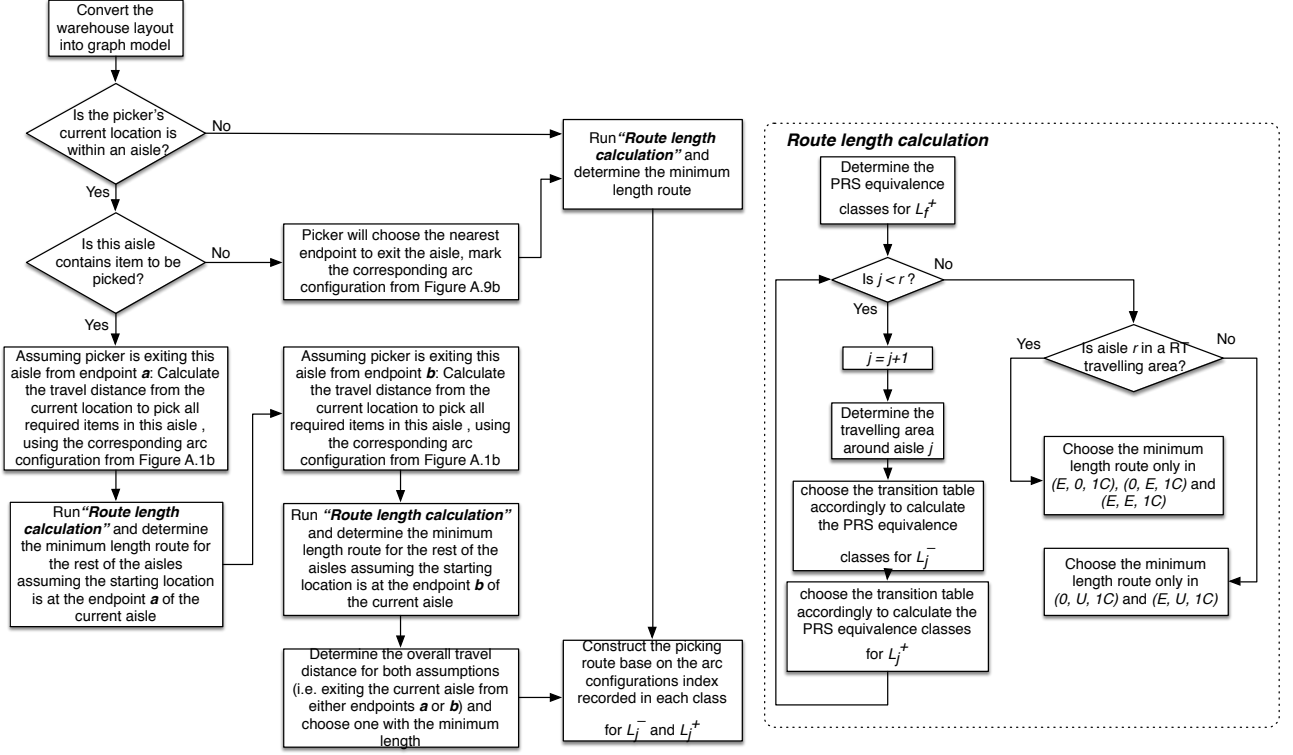

Figure 4: Flow chart for Interventionist Routing Algorithm

The ability of a picker to be re-routed regardless of the picker's current location is enabled by IRA via a) the definition of the OW travelling area and b) the determination of the appropriate endpoint to exit the picker's current aisle based on the optimal route to follow from that endpoint onwards. These two additions form the main difference between this work and the earlier work of Ratliff and Rosenthal (1983).

4.2.3. Analysis of IOP Algorithm Criteria

Since IRA allows the picker to be re-routed regardless of his current location and always complete his pick-cycle at the depot, it is immediately clear that the criteria of *interruptibility* and *centralised depositing* proposed in Section 4.1 have been addressed. Hence, we emphasise the other two IOP algorithm criteria of *Optimality* and *Complexity* here and we also provide information on the computation time for algorithm execution.

Optimality

From the *Route_Length_Calculation* procedure of IRA presented in Section 4.2.2, the algorithm assesses all possible arc configurations for accessing a pick-aisle and for crossing between two adjacent aisles for all $j \in [f, r]$. In other words, the proposed algorithm essentially examines all possible arc configurations of every L_j PRS for constructing a feasible pick-route, and then determines the one with the minimum length as the final pick-route.

17

Therefore, by conducting an exhaustive search for all possible routes, the routing problem for IOP is solved optimally by the proposed algorithm.

Since the algorithm is using an exhaustive search to achieve optimality, the immediate issue is then the complexity of the algorithm, i.e. whether the number of calculations is linear to the number of pick-aisles and whether the computation time is polynomial. We will examine these issues next.

Complexity

In order to achieve the criteria of complexity, the proposed algorithm adapts a similar approach as Ratliff and Rosenthal (1983) to develop an efficient procedure for conducting the exhaustive search. Such efficiency is achieved using the concept of equivalent PRS (as discussed in Section 4.2.1), which allows all possible equivalent PRSs to be classified into different equivalence classes such that the completion graph for each class is the same. Consequently, every time the PRS equivalence classes for L_j^- or L_j^+ are calculated (Section 4.2.2), the required PRS equivalence classes previously constructed can be represented by the PRS arc configuration with the minimum length. In this way, the algorithm does not need to consider other possible arc configurations for the same equivalence class, i.e. IRA is effectively reducing the search space for the exhaustive search. Therefore, the total number of calculations depends on the number of possible combinations for each resulting equivalence class guided by the associated table (i.e. the density of the table).

Although the total number of possible routes increases in a non-linear way as the number of pick-aisles increases, for IRA, the number of calculations increases linearly. This is demonstrated in Figure 5a. The mathematical proof is omitted for length reasons but is documented in (Lu et al., 2015).

Computation time

The computation time was tested via a simulation of about $8,500$ items for warehouses of different sizes in terms of number of aisles. The simulation was run on a computer with a 2GHz Intel Core i7 processor and 8GB of memory. For a regular warehouse of 10 aisles, the mean computation time was less than 7 milliseconds, with a standard deviation of 4 milliseconds and a maximum value of 24 milliseconds. These computation times should not be expected to cause any disruptions in a operational scenario that requires the real-time calculation of a new optimal route. Average computation times for larger warehouses are plotted in Figure 5b which also demonstrates the linearity of the IRA.

4.3. Illustrative Example

To illustrate the mechanics of executing the proposed algorithm, we will apply it to the order-picking layout previously used in Ratliff and Rosenthal (1983). Further, in order to
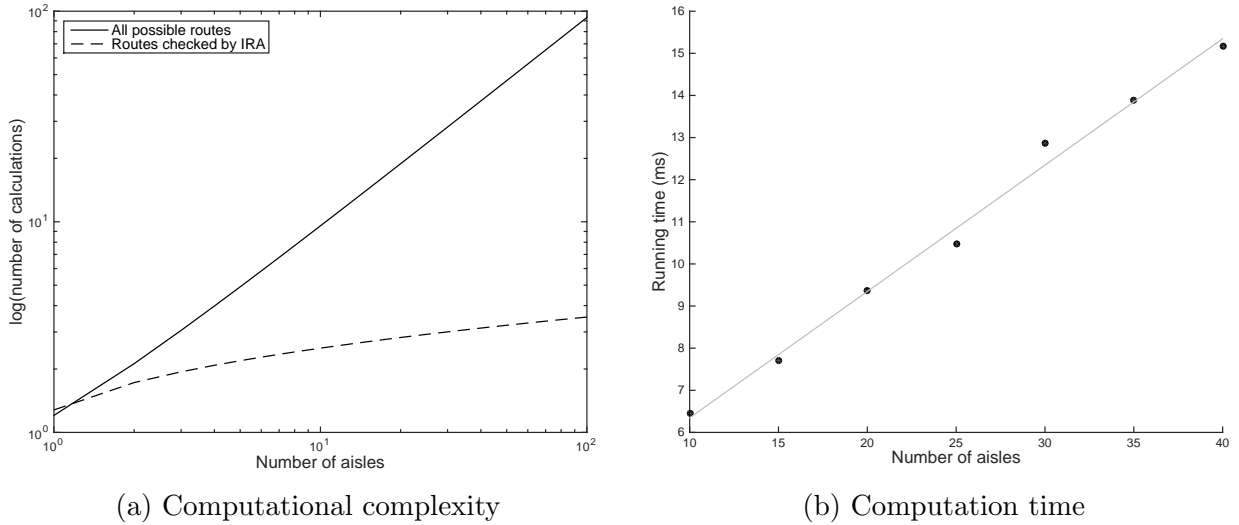
(a) Computational complexity   (b) Computation time

Figure 5: Algorithm complexity

demonstrate the interruptibility of the algorithm, we assume that —at the time his pick-list is updated— the picker's location is as illustrated in Figure 6a.

The associated information regarding the minimum length PRS for each equivalent class is listed in Table 1. For each equivalence class, other than the index column, each cell of the table is formed by three numbers to represent:

- the length of the minimum PRS for the corresponding equivalent class;

- the index of the equivalence class of the predecessor PRS that constructed the current PRS; and

- the corresponding arc configuration for aisles (L_j^+) from Figure A.1, or for cross-overs (L_j^-) from Figure A.2.

Hence, for example, the Aisle 1 entry for Index 7 of "15, - , i" means the minimum PRS of L_j^+ in class $(U, U, 1C)$ has length 15, there was no predecessor PRS , and the corresponding arc configuration is i from Figure A.1.

The algorithm can be applied to this illustrative example by the following steps:

1. The graph model is depicted in Figure 6a.

2. The round-trip and one-way travelling areas are also depicted in Figure 6a.

3. *Setup the algorithm based on the exit point of the current aisle:* we can identify that the order-picker is inside Aisle 2 for this re-routing and since it is assumed that all items within this aisle must be picked before leaving the current aisle, the only possible arc configuration for A_2 is either number (ix) or (x) from Figure A.1 by using the aisle

19

exit of a_2 or b_2 respectively. In order to select the arc configuration for A_2, we firstly determine the travel distance for exiting the current aisle: D_a if picker leave this aisle through a_2 (i.e. apply configuration (ix)), and D_b if picker leave this aisle through b_2 (i.e. apply configuration (x)) which are 12 and 15 respectively. Secondly, we determine the consequent minimum length PRS in both situations. Assuming that there is no item requested in Aisle 2 and the picker is on the node a_2 or b_2 as if the picker has finished the picking in Aisle 2, we calculate the resulting distance for both endpoints. We choose endpoint a_2 to demonstrate the procedure here.

4. *Determine the PRS equivalence classes for $L_f{}^+$:* the first pick-aisle from the left is Aisle 1, i.e. $j = f = 1$. As mentioned above, the equivalent class $(0, 0, 0C)$ can only be applied to $L_1{}^-$ and vice versa, so we will not include such class in the calculation, and will mark all entires of $L_1{}^-$ in $'-'$ as not-applicable in Table 1. Then the minimum length PRS for each equivalent class in $L_1{}^+$ is obtained simply by applying the arc configurations from Figure A.1(a). For example, the equivalence class $(E, 0, 2C)$ will be obtained by applying the arc configuration (ii), resulting a distance of 24. The dash $'-'$ in this column means that there is no predecessor PRS.

5. Increase index number j to 2.

6. *Calculate the PRS equivalence calsses for L_j^-:* since Aisle 2 is within the round-trip area, according to Appendix A, we will use Table A.2a to calculate each equivalent class for $L_2{}^-$. For instance, Table A.2a suggests that there are two ways to obtain an $L_2{}^-$ PRS in equivalent class $(0, E, 1C)$: we could add arc configuration (vii) (length of 4) from Figure A.2 to either the equivalent class of $(0, E, 1C)$ (with length of 22) or $(E, E, 1C)$ (with length of 30) in $L_1{}^+$ PRS. Therefore the two ways constructed the $L_2{}^-$ PRS with length of 26 or 34 respectively for the equivalent class $(0, E, 1C)$. We then choose and record the combination with the minimum length (26), the associated predecessor (9) and the added arc configuration (vii) into Table 1.

7. According to the *steps 4 to 6* of the procedure summarised in Section 4.2.2, we apply the same method to calculate the minimum length PRSs in the L_j^- and L_j^+ equivalence classes for $j = 2, 3, \ldots, 6$ sequentially.

8. *Determine the PRS classes for each aisle and cross-aisle:* since the minimum length route subgraph can only be obtained by the class $(E, 0, 1C)$, $(0, E, 1C)$ or $(E, E, 1C)$ if the right-most picked aisle is a round-trip aisle (Ratliff and Rosenthal, 1983), we can find from Table 1 that the optimised order-picking is the $L_6{}^+$ PRS with length of 75 of class $(E, E, 1C)$. The sequence of the arc configuration can then be tracked back through the table which was indicated by the underlined and bolded entries.

20

9. *Repeat the same method for the case of D_b:* once we finished obtaining the length of the picking-route if the picker leave Aisle 2 at a_2, we then use the same method to calculate the route length of exiting at the node b_2. We will find the resulting total length is 76 and also ends with class $(E, E, 1C)$ in $L_6{}^+$.

10. *Determine the exit point of the current aisle and the resultant route:* Combine with the calculation of $D_a = 12$ and $D_b = 15$, the total distance for leaving Aisle 2 from node a_2 and b_2 will be 87 and 91 respectively. Therefore, the route with leaving Aisle 2 from node a_2 is chosen, and the resultant route can simply be constructed by adding the arc configurations from Figure A.1 and A.2 respectively for each L_j^- and L_j^+ suggested in the calculation table for this node in Table 1, but replace the $(L_2)^+$ as (ix). Figure 6b shows the resulting minimum length route subgraph.
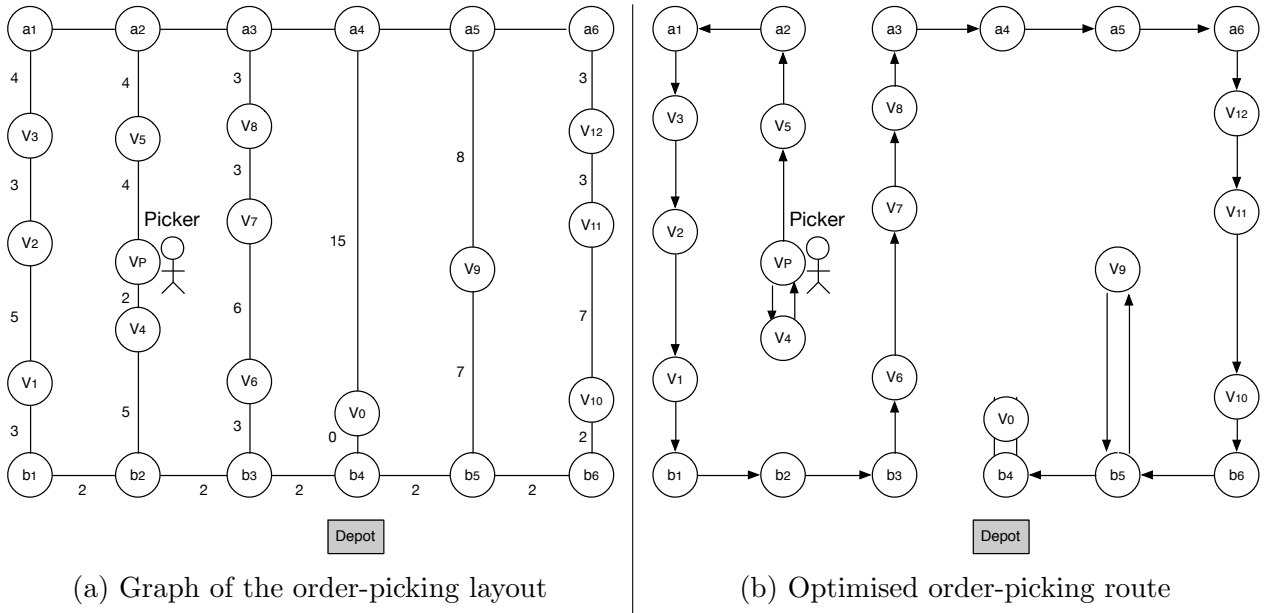

(a) Graph of the order-picking layout   (b) Optimised order-picking route

Figure 6: Example for applying the proposed algorithm

5. Industrial Example

Having illustrated the steps to be taken to re-calculate the optimised route using the proposed algorithm, in this section the algorithm is used to address the changing warehouse needs of a logistics provider. This section specifically aims to demonstrate the performance of the algorithm and to compare the algorithm against other common routing approaches.

5.1. Industrial Scenario

The industrial example studied here refers to the warehouse operations of a collaborative third party logistics provider for various multi-national manufacturers. Among others, the

21

Table 1: Solution to the illustrative problem in Figure 6a

Index	Equivalence classes	Aisle 1		Aisle 2		Aisle 3	
		L_1^-	L_1^+	L_2^-	L_2^+	L_3^-	L_3^+
1	(U, 0, 1C)	—	—	—	—	30, 8, i	54, 1, ii
2	(0, U, 1C)	—	—	—	—	**21, 7, ii**	45, 2, iii
3	(E, U, 1C)	—	—	—	—	25, 7, iii	43, 3, iv
4	(E, U, 2C)	—	—	—	—	—	39, 2, iv
5	(U, E, 1C)	—	—	—	—	44, 10, iv	**36, 2, i**
6	(U, E, 2C)	—	—	—	—	32, 9, iv	48, 1, iv
7	(U, U, 1C)	—	**15, −, i**	**19, 7, v**	**19, 7, vi**	—	—
8	(E, 0 , 1C)	—	24, −, ii	28, 8, vi	28, 8, vi	—	—
9	(0, E, 1C)	—	22, −, iii	26, 9, vii	26, 9, vi	—	—
10	(E, E, 1C)	—	30, −, v	38, 10, viii	38, 10, vi	—	—
11	(E, E, 2C)	—	20, −, iv	28, 11, viii	28, 11, vi	—	—
12	(U, U, 2C)	—	—	—	—	—	—
		Aisle 4		Aisle 5		Aisle 6	
		L_4^-	L_4^+	L_5^-	L_5^+	L_6^-	L_6^+
1	(U, 0, 1C)	**38, 5, i**	68, 1, ii	—	—	—	—
2	(0, U, 1C)	45, 3, ii	45, 2, iii	—	—	—	—
3	(E, U, 1C)	49, 3, iii	49, 3, iii	—	—	—	—
4	(E, U, 2C)	45, 4, iii	45, 4, iii	—	—	—	—
5	(U, E, 1C)	42, 5, iv	42, 5, iii	—	—	—	—
6	(U, E, 2C)	54, 6, iv	**38, 1, iii**	—	—	—	—
7	(U, U, 1C)	—	—	46, 5, v	60, 7, iii	64, 7, v	76, 8, i
8	(E, 0 , 1C)	—	—	53, 3, vi	69, 8, ii	61, 10, vi	87, 8, ii
9	(0, E, 1C)	—	—	49, 2, vii	63, 9, iii	61, 10, vii	85, 9, iii
10	(E, E, 1C)	—	—	57, 3, viii	57, 12, i	65, 10, viii	**75, 12, i**
11	(E, E, 2C)	—	—	53, 4, viii	65, 9, ii	71, 9, viii	77, 8, iv
12	(U, U, 2C)	—	—	**42, 6, v**	**56, 12, iii**	**60, 12, v**	76, 12, iv

logistics provider provides warehouse services for finished products of different retailers and end-customers around China. Over the years, orders have mainly been placed by retailers, and each order typically contains a large number of order lines (product types) and therefore allows a relatively large amount of time for order-picking. Also, since most of the orders are normally placed around 3pm-5pm, the items to be picked in the warehouse are known before the associated order-picking, which normally takes place after 5pm. For this reason, the warehouse currently applies static order-picking strategies[6] when there will be no new orders received during the picking operation.

---

[6]To form order batches and release to order-pickers as pick-lists based on the capacity of the picking-device.

However, due to the rapid development of e-commerce in China, the frequency of orders placed by end-customers has been increased significantly. These orders typically contain only one order line per order with much shorter lead time. Also, since these orders are placed by individual customers online, their arrival time is scattered throughout the day. In such an operating environment, the static order-picking strategy currently in use becomes inefficient due to the high uncertainty of order arrivals and priorities during picking operation. For example, consider an order-picker who needs to wait until a certain amount of orders are received before starting a picking process (in order to avoid numbers of unnecessary pick-cycles (longer travel distance)). This method, although reducing the total travelled distance, it may also result in longer order completion times and even reduce the on-time completion ratio.

The company considers a case in which its pickers will be permitted to start a new pick cycle even if there are not enough orders received to fill their picking device's capacity. Then, during the picking operation, the pick list of a picker will be updated by new incoming orders and the pick route will then be re-calculated accordingly. In the rest of this section, we will compare such a solution against the static algorithm to identify benefits and drawbacks under different operational conditions.

5.2. Experimental Design

In order to compare the performance of the different routing algorithms, a simulation environment was built using NetLogo[7] and MATLAB®[8]. The warehouse under consideration is an one-block warehouse with narrow aisles (i.e. the lateral movement within an aisle is negligible), in which only one picker with finite capacity is conducting the picking process. The new orders received have the same priority, contain one order line (only one item will be requested per order) and their arrival time follows a Poisson Process. Further assumptions for the simulation are:

1. each item/order will have a unique associated storage location;

2. each associated storage location will always have sufficient inventory for the requested item, i.e. different orders may request items from the same storage location;

3. new orders will be added to the picker's picking list regardless of the current location of the picker in the warehouse as long as there is enough capacity in his picking device;

4. all items/orders will occupy the same space on a picking device;

---

[7]https://ccl.northwestern.edu/netlogo/
[8]http://www.mathworks.co.uk/products/matlab/

5. the picker will travel on constant speed around the warehouse.

The simulation was used to compare two sets of algorithms:

SET 1: Comparing the proposed routing algorithm incorporating interventionist pick-list *vs.* Optimal routing algorithm with static pick-list: This set of comparisons emphasises the potential advantages of an interventionist order-picking strategy, as well as determining the appropriate operational condition (in terms of orders arrival rate) for such a strategy.

SET 2: Comparing the proposed routing algorithm with interventionist pick-list *vs.* Largest-gap heuristics with interventionist pick-list: This set of comparison aims to demonstrate the advantage of the proposed routing algorithm by comparing it with a modified Largest-gap routing heuristic method for interventionist order picking.

For the second set of comparisons, the largest-gap heuristic method[9] has been modified in order to enable it to perform the interventionist routing as it has been shown that, for static pick-list, it can provide the best routing performance when the number of picks within each pick-cycle is between 5–25 (Petersen II, 1997). Other heuristic methods could also be considered for enabling interventions, however, they need to be developed/modified accordingly.

The performance of the above algorithms is compared via two metrics: *Average Order Completion Time (AOCT)* and *Average Travel Distance per Order (ATDO).* The order completion time is the time period between receiving the order and depositing the order at the depot by the order-picker. Hence AOCT reflects the average time required for an order before it is ready for the next warehouse operation. In certain settings, AOCT can be used as a metric of the responsiveness of warehouse operations (Van Nieuwenhuyse and De Koster, 2009). Practical examples — for example — include the case when end-customers wait for their orders inside a click-and-collect store or the situation when a truck is waiting for a certain order before it is allowed to depart. One can also consider the case of deciding when a truck will arrive based on estimations of the order completion time, i.e. transportation is scheduled after orders are received. ATDO is the mean value of overall travel distance for a work-shift divided by the total number of orders received in that work-shift. This metric can reflect the cost per order of an order-picking strategy.

---

[9]Routing by Largest Gap heuristics: the order-picker will only transverse the block via the left-most and right-most pick-aisle, and the 'gap' represents the separation between any two adjacent picks, between the first pick and the front cross-aisle, or between the last pick and the back cross-aisle (De Koster et al., 2007)

The results in the next section are all based on 31 work-shifts (operations in a month) for each scenario, with the parameters of the simulation aligning with those recommended by the collaborating company:

- Warehouse layout: 1 block (i.e. 2 cross-aisles), 10 parallel aisles with 15 storage locations on each side. The depot is located at one end of aisle 5. The length of an aisle is 15 meters and the distance between two adjacent aisle is 3 meters.

- Order arrival rate = 0.01–0.09 orders / sec

- Working time per work-shift = 8 hours

- Order-picker travel speed = 1 meter/sec, Picking time per item = 5 sec

- Capacity of the picking device = 20 items/orders

5.3. Results

5.3.1. **SET 1:** Interventionist Order-picking (IOP) VS Static Order-picking (SOP)

In static order-picking, since the pick-list cannot be updated on-route, the order-picker waits until all the capacity of his picking device (e.g. a trolley) has been booked to new orders before starting a pick-cycle. In interventionist order-picking, any newly arrived order will be immediately added to the order-picker's pick-list, regardless of whether he has commenced his pick-cycle or not, as long as the picker has not reached full capacity. In order not to force a picker start a new cycle when only a very small number of orders are waiting to be picked, we assume the order-picker waits until 25% of his capacity has been booked before starting a pick-cycle.

Figures 7 compares the performance on average order completion time and average travelling distance per order respectively for order-arrival rates varying from 0.01 to 0.09 orders per second. By studying Figures 7a and 7b, we can conclude:

- *The reduction in average order completion time achieved by the interventionist strategy, resulted in an increase on travelling distance for the order-picker.* Even though IOP can reduce the average completion time of an order by up to 76% it might also require a picker to travel much more (up to 108%). The reason for this is that in IOP, a newly arrived order does not have to wait until a picker returns to his depot or until he reaches capacity in order to be picked, thus making the picking operation more responsive than SOP. On the other hand, since the picker is required to pick any newly received order during the picking operation, the resulting trip will be longer compared to SOP where all orders are known in advance before the calculation of the picking route. Hence, a selective policy for selecting which orders should intervene picking

might yield a better result (i.e. our performance results can be seen to be a lower bound on the best performance achievable with developing various policies for IOP).

- *The differences between interventionist and static order-picking systems became smaller when order arrival rates increase.* As the order arrival rate increases, there is less waiting time for the order-picker until he starts a pick-cycle in static strategy, hence the difference in order completion time compared to the interventionist strategy becomes smaller. At the same time, by the time a picker returns to his depot in IOP, more orders are already waiting to be picked thus being able to calculate a shorter picking route. In fact, when the order arrival rate is greater than 0.07 orders per second, in which the order-picker can not be intervened (most of the times) as his capacity has already been booked by the time he starts a new pick-cycle, the performances for both strategies are almost identical. As a result, the interventionist strategy can only offer significant benefits when the order arrival rate is between 0.01 and 0.07 orders per second in warehouses similar to ones studied here.

(a) Average order completion time   (b) Average travel distance per order
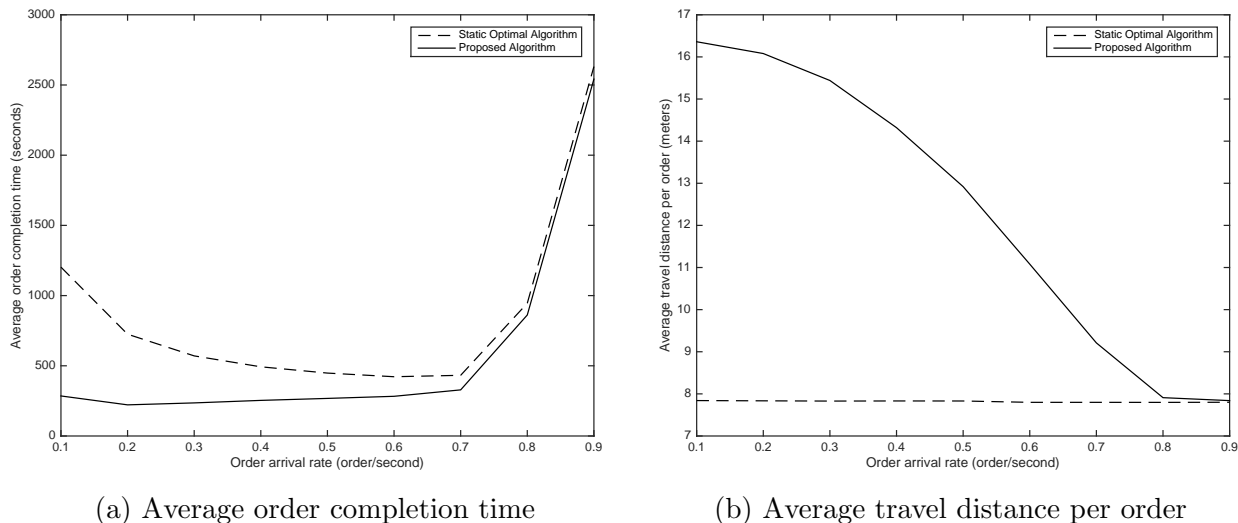
Figure 7: SET 1: proposed algorithm against static optimal algorithm for different order arrival rates

To further investigate the relationship between the two performance measures under other operational control parameters, we will now study the impact the order-picker's capacity has on them. Apart from affecting the number of orders a picker can pick in a pick-cycle, a picker's capacity also modifies the trigger for starting new pick-cycles in a both strategies. The order arrival rates used for this part of the study are 0.03 and 0.06 based on the average values calculated by the case company during their off-peak and peak seasons respectively.

In Figure 8, we depict the difference between IOP compared to SOP for order completion time and travel distance per order for different capacity levels of the picking device

$(5, 10, 15, 20)$. The two graphs indicate that even though reducing a picker's capacity also reduces the benefit IOP can earn in terms of AOCT, it also reduces the loss on ATDO significantly. For example, reducing the capacity from 20 to 5 reduces the benefit on order completion time around 14%, however, the order-picker has to travel 69% less. Similar observations can be made for the case with more frequent order arrivals ($\lambda = 0.06$), where it could be worth reducing the picker capacity from 20 to 10. However, any further reduction on capacity will cause the loss in order completion time outweigh the benefit in reducing travel distance.
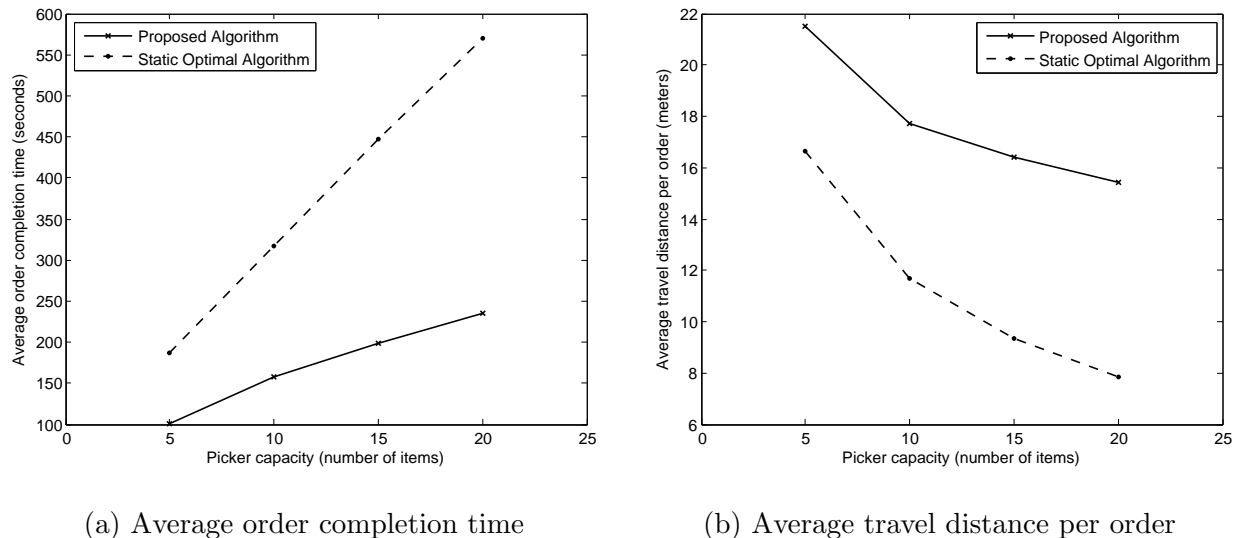

(a) Average order completion time   (b) Average travel distance per order

Figure 8: Performance comparison between static optimal and proposed algorithms for different capacity levels ($\lambda = 0.03$)

5.3.2. **SET 2:** *Comparing Interventionist Order-picking(IOP) strategies: Proposed Interventionist Routing Algorithm VS Modified Largest Gap Heuristics*

This section will demonstrate the advantage of the proposed interventionist routing algorithm (IRA) under the category of interventionist strategy. IRA will be compared against a Modified Largest Gap Heuristic (MLGH) routing method which was modified to enable re-routing after an intervention in the picking process, i.e. the order-picking process can be intervened at any point if an order arrives, and the the associated re-routing will still follow the principles of conventional Largest Gap heuristics.As discussed in the previous section, the orders arrival rate studied in this section will range between 0.01–0.07 orders per second in order to create an environment that can facilitate interventions.

Figure 9 compares the performance of the proposed algorithm against the MLGH, in terms of average order completion time and average travelling distance per order respectively. By studying Figure 9a and 9b, we can conclude:

27

- The proposed IRA outperforms MLGH for all order-arrival rates both in terms of AOCT and of ATDO. Their difference gets its maximum value when $\lambda \simeq 0.03$ with 15% benefit in AOCT and 12% in ATDO.

- Figure 9b shows a slight decrease on the benefit of ATDO when the there are more than 0.04 order arrivals per second. This is because when the ratio between the number of items needed to be picked in one pick-cycle and the total number of storage locations increases, the difference in the total travel distance between the two routing methods decreases.

- Figure 9a shows the AOCT benefit of applying IRA rises when the order-arrival rate increases to 0.07. This is because after a certain point during a work shift, the awaiting orders at the depot have accumulated in a way that each new pick-cycle begins with full capacity, therefore making the picking process static rather than interventional. In this case, what is actually compared is an optimal routing algorithm against largest gap heuristics in a static setting.

(a) Average order completion time
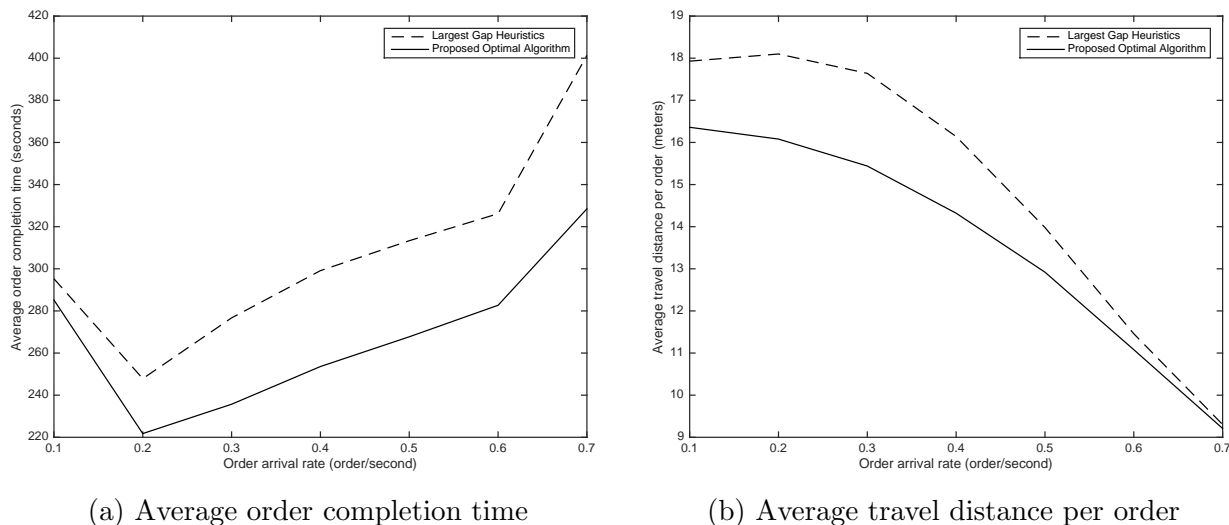

(b) Average travel distance per order

Figure 9: SET 2: proposed algorithm against modified largest gap heuristics for different order arrival rates

Apart from examining the benefit of IRA on average terms, we also looked more detail into the performance of both algorithms in each work-shift[10]. For this examination the less advantageous order arrival rate among the ones that are more frequent in the case company ($\lambda = 0.06$) was chosen. The results showed that in every working shift, the proposed IRA outperformed the MLGH in terms of both AOCT and ATDO.

---

[10]In the simulation the work-shifts used to compare the algorithms are identical

6. Conclusions and Discussion

In this paper, we presented an interventionist routing algorithm for the dynamic order-picking problem involving new orders arriving randomly during the picking process —thus extending the seminal routing algorithm proposed by Ratliff and Rosenthal (1983). The proposed interventionist picking differs from the dynamic picking studied by Gong and De Koster (2008) in that we allow the picking operation to be intervened by newly arrived orders even though the required items are not further downstream in a picker's current route. In this way the picking operation can be supplemented by any order received.

The performance of the proposed algorithm was tested using a set of simulations base on an industrial case example, that resulted in the following practical conclusions:

- when the number of orders that trigger a new pick-cycle is fixed, an interventionist approach can result in shorter average order completion times than a static one due to its ability to add new orders to a pick-list during the picking process. However, such reduction results in additional travel distance for the order-picker;

- the trade-off between the above performance measures can be modified by deciding the maximum capacity of the picking device that will be used in a warehouse;

- within an interventionist approach, the proposed optimised algorithm outperforms the largest gap heuristics in both travel distance and order completion time, thus emphasising the importance of optimisation in an interventionist approach.

The aim of this paper is to introduce and explain the algorithm and present a practical example of how it could potentially be applied. An important area requiring investigation is the identification of ways to regulate the algorithm's usage. This is important because the mechanism that regulates the interventions can have an effect both on order completion time and travel distance. For example, only the most appropriate orders among the newly-arriving ones, such as the ones on the current path of the picker, could be considered for intervention. This latter issue of how to determine a beneficial picking strategy using the proposed algorithm will be examined in a different paper (Giannikas et al., 2015).

Future extensions of the algorithm will examine its application to different warehouse layouts, e.g. multiple blocks. However, we note that the most important area of future research is on ways in which the algorithm can be used for deploying interventionist strategies in fully operational warehouses with multiple pickers and multiple storage locations for each product type. This could add extra levels of complexity since the most appropriate picker and storage location has to be determined for each new order before any intervention takes place. The aim is to achieve more resilient, flexible and customer-oriented warehouse operations.

Acknowledgements

References

Agarwal, A., Shankar, R., Tiwari, M., 2006. Modeling the metrics of lean, agile and leagile supply chain: An ANP-based approach. European Journal of Operational Research 173, 211–225.

Bartholdi, J.J., Hackman, S.T., 2010. Warehouse & distribution science: release 0.92. volume 30332. The Supply Chain and Logistics Institute, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Chiang, D.M.H., Lin, C.P., Chen, M.C., 2011. The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres. Enterprise Information Systems 5, 219–234.

Cornuéjols, G., Fonlupt, J., Naddef, D., 1985. The traveling salesman problem on a graph and some related integer polyhedra. Mathematical programming 33, 1–27.

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

Davarzani, H., Norrman, A., 2015. Toward a relevant agenda for warehousing research: literature review and practitioners' input. Logistics Research 8.

De Koster, R., 2003. Distribution strategies for online retailers. Engineering Management, IEEE Transactions on 50, 448–457.

De Koster, R., Le-Duc, T., Roodbergen, K.J., 2007. Design and control of warehouse order picking: A literature review. European Journal of Operational Research 182, 481–501.

De Koster, R., Van Der Poort, E., 1998. Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions. IIE transactions 30, 469–480.

Dekker, R., De Koster, M., Roodbergen, K.J., Van Kalleveen, H., 2004. Improving order-picking response time at ankor's warehouse. Interfaces 34, 303–313.

Giannikas, V., Lu, W., McFarlane, D., 2015. Interventionist order-picking strategies for single-picker warehouse operations In preparation.

Giannikas, V., Lu, W., McFarlane, D., Hyde, J., 2013. Product intelligence in warehouse management: A case study, in: Industrial Applications of Holonic and Multi-Agent Systems, Springer. pp. 224–235.

Goetschalckx, Marc; Ashayeri, J., 1989. Classification and design of order picking. Logistics Information Management 2, 99–106.

Goetschalckx, M., Ratliff, H.D., 1988. Order picking in an aisle. IIE transactions 20, 53–62.

Gong, Y., De Koster, R., 2008. A polling-based dynamic order picking system for online retailers. IIE Transactions 40, 1070–1082.

Gu, J., Goetschalckx, M., McGinnis, L.F., 2010. Research on warehouse design and performance evaluation: A comprehensive review. European Journal of Operational Research 203, 539–549.

Letchford, A.N., Nasiri, S.D., Theis, D.O., 2013. Compact formulations of the steiner traveling salesman problem and related problems. European Journal of Operational Research 228, 83–92.

Lu, W., Giannikas, V., McFarlane, D., Hyde, J., 2014. The role of distributed intelligence in warehouse management systems, in: Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics, Springer. pp. 63–77.

Lu, W., McFarlane, D., Giannikas, V., 2015. Interventionist Routing Algorithm for single-block warehouse: Application and Complexity. Technical Report. University of Cambridge. Report No. CUED/E-MANUF/TR.35, ISSN: 0951-9338. Available online at `http://www.ifm.eng.cam.ac.uk/uploads/Research/DIAL/Resources/Reports/LMG15.pdf`.

Makris, P., Giakoumakis, I., 2003. k-interchange heuristic as an optimization procedure for material handling applications. Applied Mathematical Modelling 27, 345–358.

Naim, M.M., Gosling, J., 2011. On leanness, agility and leagile supply chains. International Journal of Production Economics 131, 342–354. Innsbruck 2008.

Otto, J.R., Chung, Q., 2000. A framework for cyber-enhanced retailing: Integrating e-commerce retailing with brick-and-mortar retailing. Electronic Markets 10, 185–191.

Petersen, C.G., Aase, G., 2004. A comparison of picking, storage, and routing policies in manual order picking. International Journal of Production Economics 92, 11–19.

Petersen II, C.G., 1997. An evaluation of order picking routeing policies. International Journal of Operations & Production Management 17, 1098–1111.

Petersen II, C.G., 1999. The impact of routing and storage policies on warehouse efficiency. International Journal of Operations & Production Management 19, 1053–1064.

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

Renaud, J., Ruiz, A., 2007. Improving product location and order picking activities in a distribution centre. Journal of the Operational Research Society 59, 1603–1613.

Roodbergen, K.J., De Koster, Rene, a., 2001. Routing order pickers in a warehouse with a middle aisle. European Journal of Operational Research 133, 32–43.

Roodbergen, K.J., Koster, R.É., 2001. Routing methods for warehouses with multiple cross aisles. International Journal of Production Research 39, 1865–1883.

Selviaridis, K., Spring, M., 2007. Third party logistics: a literature review and research agenda. The International Journal of Logistics Management 18, 125–150.

Theys, C., Bräysy, O., Dullaert, W., Raa, B., 2010. Using a TSP heuristic for routing order pickers in warehouses. European Journal of Operational Research 200, 755–763.

Tian, Y., Ellinger, A.E., Chen, H., 2010. Third-party logistics provider customer orientation and customer firm logistics improvement in china. International Journal of Physical Distribution & Logistics Management 40, 356–376.

Tompkins, J.A., 2010. Facilities planning. Wiley.

Van Nieuwenhuyse, I., De Koster, R., 2009. Evaluating order throughput time in 2-block warehouses with time window batching. International Journal of Production Economics 121, 654–664.

Appendix A. Route Constructions

This section explains the procedures of applying the Interventionist Routing Algorithm (IRA).

Before constructing the order-picking route, we firstly need to examine the possible route (arc) configurations for a picker to access or leave a particular pick-aisle, and to cross between two adjacent pick-aisles.

In the configuration for pick-aisle, a minimum length route cannot have more than two arcs between any pair of vertices (Ratliff and Rosenthal, 1983). Therefore, as shown in Figure A.1 beside the 6 possible arcs configurations (defined in (Ratliff and Rosenthal, 1983)) to access an aisle if the picker is at either endpoint of the aisle, we introduced 4 possible configurations to leave an aisle if the picker is in the aisle. It is noted that case (iv) is configured by determining the largest gap between the two item vertices in that aisle because it can only be applied if there are more than two storage locations to be visited in the aisle; and it is the only configuration that may not be unique.
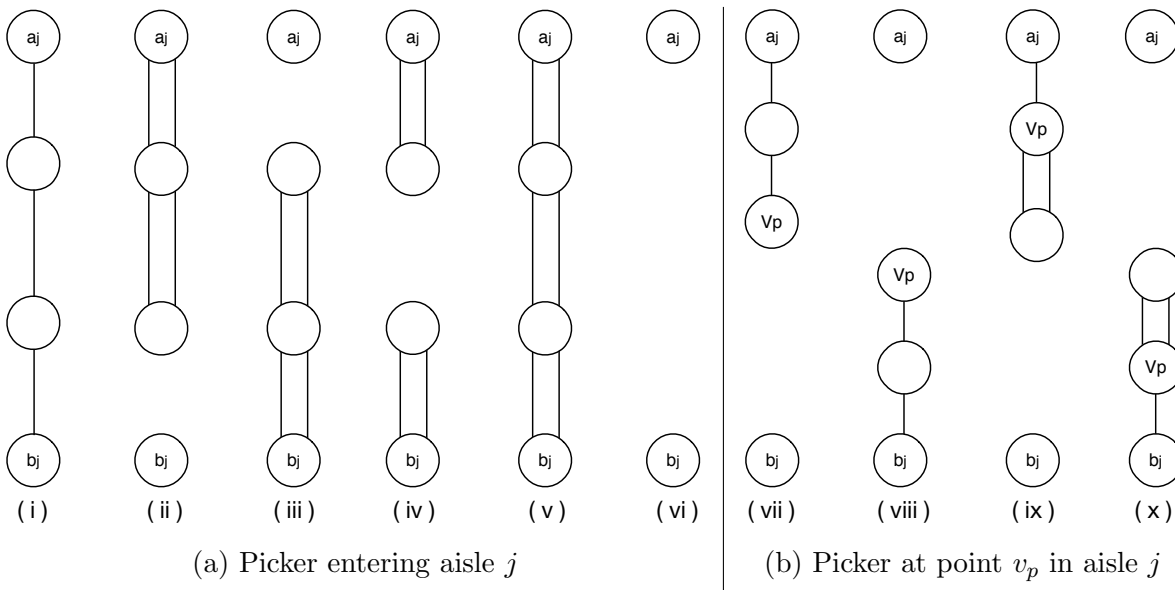

Figure A.1: Possible arc configurations for a picker to access or leave any aisle j in a minimum route subgraph

Similarly, in the configuration for cross-aisles, a minimum length route cannot have more than two arcs between any pair of aisle endpoints vertices. Since the total number of arc between two adjacent aisles should always be odd for one-way and even for round-trip, there are only 4 possible arc configurations for cross-overs in one-way and another 4 possible configurations in Round-trip areas, as shown in Figure A.2.

Having identified all possible arc configurations, we can now describe the three main stages of IRA.

Stage 1: Initiation

In order to construct the order-picking route, we first need to convert the depot, current location of order-pickers and all requested items on the pick-list into the graph model as described in Section 3.2. We number the aisles in an increasing matter from left to right unless a) the picker's current location is on the right-hand side of the depot aisle or b) the picker's current location is on the depot aisle and there are no items to be picked on the
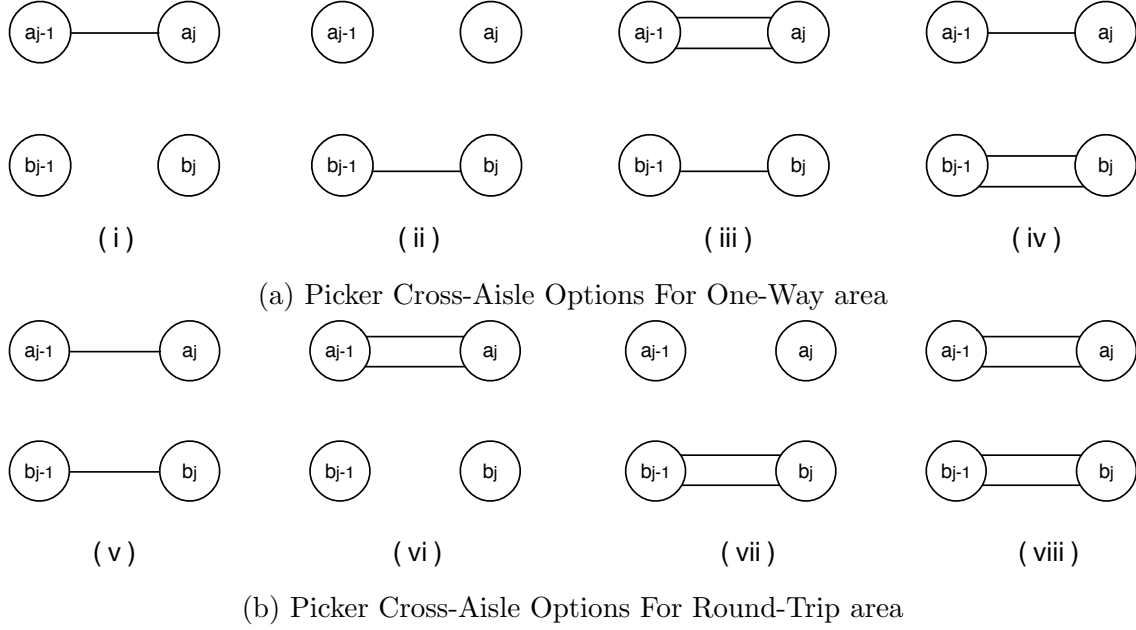
(a) Picker Cross-Aisle Options For One-Way area

(b) Picker Cross-Aisle Options For Round-Trip area

Figure A.2: Possible arc configurations for cross-aisle between aisle $j-1$ and aisle j in minimum route subgraph

right-hand side of the depot. In these special cases, we number the aisles in an increasing matter from right to left.

Secondly, we determine the situations by the current location of the picker when a customer order/request is added and a new order-picking route is required. This algorithm assumes that a new route will not be regenerated when the picker is traveling on the cross-aisle in between any two adjacent aisles. This is because a) the distance between two adjacent aisles is relative small compared to the length of an aisle, and b) the cross-aisles in practice are generally too narrow for an order-picker with a travelling device to change the direction. Therefore, a new order-picking route is constructed in two situations:

Situation 1: the order-picker is inside a pick-aisle;

Situation 2: the order-picker is at one of the endpoints (a_j or b_j) of an aisle.

In Situation 1, in order to decide which endpoint the order-picker should exist the pick-aisle, the algorithm will:

1. assume no item is requested from the current pick-aisle in question;

2. calculate the potential travel distances of leaving from either exit;

3. determine the overall travel distance in addition to the distance of travelling to the associated exit;

34

4. choose the exit (endpoint) with the shortest overall travel distance and generate the associated routing.

In order to generate the route for Situation 1, the algorithm will calculate that for Situation 2 twice by assuming the order-picker is at the endpoint a and then b respectively. (Therefore, for Situation 2, we can go to the next step directly.)

We denote aisles from left to right, the first pick-aisle as aisle f and the last one is aisle r, the algorithm will then construct all equivalence classes of the L_j PRS for each aisle $j = f, f + 1, \ldots, r - 1, r$ in sequence. Each equivalence classes of $L_f{}^+$ PRSs (i.e. $(U, U, 1C), (E, 0, 1C), (0, E, 1C), (E, E, 2C), (E, E, 1C)$) can be determined by one of the 5 arcs configurations in Figure A.1a correspondingly.

If the order-picker is to the right of the depot or the in the depot aisle, and requested items are all located his left, the algorithm will 'flip' the warehouse by renumbering the aisles starting from the right-most pick aisle to the left-most one.

Stage 2: Transitions

The transition between aisle $j - 1$ to aisle j are considered in two steps: from L_{j-1}^+ to L_j^- and then from L_j^- to L_j^+.

The transition from L_{j-1}^+ to L_j^- determines which of the 8 possible configurations in Figure A.2 can be added to the equivalence classes of the L_{j-1}^+ PRS in order to obtain the L_j^- PRS in each of its equivalence classes. For instance, usingTable A.2a below, the L_j^- PRS in *(E,E,2C)* class can be obtained by applying the arc configuration (viii) in Figure A.2 to three of the equivalence classes of L_{j-1}^+ PRS, *(E,0,1C)*, *(0,E,1C)* and *(E,E,2C)* class respectively. The minimum length PRS of L_j^- in *(E,E,2C)* class can then be obtained by taking the combination that derives the shortest travel distance for this class.

Such transition is enabled by applying the appropriate transition tables based on the travelling area that aisle $j - 1$ and j are located in. For example, if aisle $j - 1$ and j are both in Round-Trip(RT) area, and the One-Way(OW) area does not exist before aisle $j - 1$, apply Table A.2a. Whereas If aisle $j - 1$ and j are both in RT area, and the OW area exists before aisle $j - 1$, apply Table A.2b.

Similarly, this transition from L_j^- to L_j^+ determines which of the 6 possible ways in Figure A.1a can be added to each equivalence classes of the L_j^- PRS in order to obtain the L_j^+ PRS in each of its equivalence classes. Such transition is also enabled by applying the appropriate transition tables based on the travelling area that aisle $j - 1$ and j are located in.

Note that all the transition tables together with the detailed description and procedure of how to choose the appropriate tables is omitted from this paper for length reasons but is documented in its entirely in (Lu et al., 2015).

Stage 3: Route construction

Table A.2: Transition from L_{j-1}^+ to L_j^- in Round-Trip equivalent classes from adding each of the arc configuration in Figure A.2b

	L_{j-1}^+ Equivalence	Arc Configuration from Figure A.2			
#	Classes	(v)	(vi)	(vii)	(viii)
1	(U, U, 1C)	(U, U, 1C)	—[a]	—[a]	—[a]
2	(E, 0, 1C)	—[a]	(E, 0, 1C)	—[a]	(E, E, 2C)
3	(0, E, 1C)	—[a]	—[a]	(0, E, 1C)	(E, E, 2C)
4	(E, E, 1C)	—[a]	(E, 0, 1C)	(0, E, 1C)	(E, E, 1C)
5	(E, E, 2C)	—[a]	—[a]	—[a]	(E, E, 2C)
6	(U, U, 2C)	(U, U, 2C)	—[a]	—[a]	—[a]

Table A.2a contains rows 1–5

Table A.2b contains rows 1–6

L_{j-1}^+ Equivalence	Arc configuration from Figure A.2			
Classes	(v)	(vi)	(vii)	(viii)
(U, U, 1C)	—[a]	(E, 0, 1C)	(0, E, 1C)	(E, E, 1C)
(E, 0, 1C)	(U, U, 2C)	—[a]	—[a]	—[a]
(0, E, 1C)	(U, U, 2C)	—[a]	—[a]	—[a]
(E, E, 1C)	—[a]	—[a]	—[a]	—[a]
(E, E, 2C)	(U, U, 2C)	—[a]	—[a]	—[a]

Table A.2c: if the picker starts the tour on the head of the depot aisle.

---

[a]No completion can connect the graph

After determining the L_r^+ PRS for each equivalence class, the minimum length route subgraph (i.e. the minimum order-picking route) is the minimum length L_r^+ PRS in (E, 0, 1C), (0, E, 1C) and (E, E, 1C) if aisle r is in a RT area (Ratliff and Rosenthal, 1983), or in (0, U, 1C) and (E, U, 1C) if aisle r is in OW area, given that $f \neq d$ in both cases.

Eventually, each equivalence class of L_j^- and L_j^+ (for $j = f, f+1, \ldots, r-1, r$) PRS can be determined by following the trace of obtaining the chosen equivalence class of L_r^+ PRS, and therefore the associated order-picking route can be constructed from aisle f to aisle r accordingly.