Optimisation of replenishment processes

The case of Udifar II

Duarte Meira Trindade Barros

Departamento de Engenharia e Gestão, Técnico Lisboa, Universidade de Lisboa, Portugal.

Abstract: Over the last 40 years, there has been an increased access of the Portuguese population to health care services, with the consequence that health costs have increased significantly. Governments have been implementing several measures aimed essentially at controlling health costs such as: introducing the use of generics; reducing the margins of sales of medicines; fixing public procurement prices; etc. The players of drug supply chain do not have control over drug prices, so it is vital that they become increasingly more efficient. In this new context Distribution Centres have assumed a vital role in the drug chain, since they guarantee that the medicines are delivered at the right time and place. Udifar II's warehouse is analysed with the goal of evaluating and improving its internal logistics by up-grading the automatic picking replenishment system, to increase productivity and reduce operating costs. For this, the Interventionist Routing Algorithm is implemented and tested in 30 real picking routes of Udifar II, and it was verified that the algorithm is useful and it is possible to reduce the distances travelled and consequently reduce costs.

Keywords: Interventionist Routing Algorithm, Order Picking, Distribution Centre, Logistics, Pharmaceutical Supply Chain.

1. Introduction

Living conditions in Portugal have improved significantly in recent years (after the 25th of April 1974), with reflections at various levels, including the democratization of access to health care and medicines. This allowed a significant growth in the pharmaceutical market. There has also been different legislative changes with direct implications for all parties in this chain. The distribution of pharmaceutical products is very important for the current design of the supply chain, enabling connections between the pharmaceutical companies and the points of sale of the medicines, which ensures the availability of drugs, and contributes to the sustainability of this business. Competition in this market started thus to be more aggressive at all levels of activity (Romão, 2012). As a result, the drug distribution sector faces a major challenge: although it maintains the level of sales (sales value), the marginal gains are much lower than in the past, so distributors need to reduce their fixed and operating costs, in order to increase or even maintain profit. Chiang et al (2011) highlight that the costs of distribution centres (DC) are strongly allocated to order picking - between 55% and 75%. Thus, if optimisation is possible, even if having only a small impact on margins costs, it might result in savings. These unitary savings, multiplied by the numerous daily orders to be met, might translate into significant gains. This study focuses on drug distributors, with the particular case on Udifar II, seeking to find solutions to improve the efficiency of the automatic picking replenishment process, given its current situation.

2. Context and Background

Wholesale Distribution of Medicines, commonly known as Warehousing, consists on the supply, storage or supply of medicines for processing, resale or use in medical services, healthcare facilities and pharmacies. This business also integrates other activities like billing, collection and management of returns (damaged or expired products - reverse logistics). In recent years, the pharmaceutical sector has undergone numerous changes - due to the political environment, economics, financial and professional. These changes not only affected pharmaceutical laboratories, but also pharmacies and the associated drug supply chain (storekeepers and distributors). Due to the changes and restrictions imposed in this sector, there has been a tendency of market concentration in the stockholding sector, with several mergers and acquisitions of distribution companies. To ensure their survival, storekeepers and distributors have also been forced to change their operational behaviour by making large investments in their information systems. Every process associated with responding to requests from pharmacies became automatic as possible, by means of sophisticated and complex logistical structures.

Udifar II - Distribuição Farmacêutica, SA is a distribution and logistics service company for pharmaceutical products, being a full line wholesaler, with almost all the medicines available in the market. The warehouse of Udifar II located at Agualva-Cacém has an area of 6.000 m2. There are five major operations taking place in this warehouse. Sequentially: reception of merchandise, storage, order picking, control and shipment. All these operations have to be executed efficiently, driven by the need to achieve an appropriate quality of service. To achieve this, Udifar II receives requests of products, guarantees that the products are available in the warehouse and prepares the products for shipment in time for delivery. If they are not capable of executing the loading in time or do not have the required products, the company may be jeopardising a client relationship. Regarding the order picking operation Udifar II uses four types of picking: automatic, semi-automatic, manual and inverse picking. The automatic picking is fully automatic and the products placed in this picking system are the most requested by clients. Udifar II has three treadmills, with channels on both sides. The products are stored vertically on each channel and then are ejected to the treadmill, which drives the products to the tub, waiting in the conveyor. Parallel to the channels, we can find stands which contain the same products in the channels. The purpose is to execute fast replenishment in case of low stock in the channels. The operators in this area have the mission of re-stocking the channels when they reach 20% of the original capacity, by First-Come-First-Served (FCFS) method. Even though this is a very expensive system it is highly efficient and allows a high turnover of Udifar II's products. From the analysis of the internal logistics of the Udifar II's warehouse it was clear that the priority area of improvement is the replenishment of automatic picking. A possibility is to apply an algorithm that processes the known information about the automatic feeders to be replenished and creates an ordered list of them, defining a path as small as possible, so that operators can perform all replenishments rationally. By doing so, the company believes that the outcomes may be very rewarding in terms of service level and flexibility within the picking process.

3. State of the art

A DC is a warehouse that is specifically designed for this purpose, since tactical changes may be very expensive and problematic after construction, with the objective of accumulation and distribution of final products and guarantees the delivery of the exact amount needed by the costumers (Hadjinsky, 2013). According to Baker (2008), a DC holds the key role of gathering materials from different suppliers and executing value added activities to fulfil the customer needs and high service levels. To obtain a good balance and equilibrium between supply and demand, quick response via automation (Henn et al, 2013) and reliable transportation is required (Baker et al, 2007). On these facilities, there are four main processes in action (Gu et al, 2007): inbound transportation, warehousing, order picking and shipment.

Order picking is the most time-consuming operation of all the DC's operation (Roodbergen et al, 2001). Order picking is the collection of products requested by customers, and can be performed automatically or manually (Marchet et al, 2015). It represents between 55-75% of all DC costs (Chiang et al, 2011). Thus, order picking improvement is highly important, given that any improvement represents significant cost savings and efficiency increase (De Koster et al, 2007).

Currently, with the emergence of the e-commerce, customer behaviour has changed remarkably (McFarlane et al, 2016), since customers started to place orders at any hour and wish to receive their orders at a specific time (Liam et al, 2015). There is also the possibility of customers cancelling their orders, which causes the supplier to deal with unexpected occurrences (Gong et al, 2008). Therefore, through the improvement of order picking, the DC can reduce lead time, improve responsiveness and, if the route upgraded is efficient, it can reduce order picking travel time up to 35% according to van Voorden et al (1999).

There are several methods regarding the improvement of order picking, for instance, heuristics and algorithms. Gong et al (2008) propose a Static Order Picking (SOP) that requires batch formation (grouping) of client requests to develop pick-lists. Nevertheless, the authors argue that it is not the best procedure given the rising number of order. Instead, they recommend Dynamic Order Picking (DOP) systems that can change throughout the picking process. DOP is always updating picking information which enables the warehouse to be easily managed given any unexpected circumstance. Although the proposed procedure is expected to improve the pickers travel efficiency, it is based on a heuristic (eventually non-optimal) solution that limits the picking-list depending of the operator position (Davarzani et al, 2015). The optimisation of the order picking process is based on two approaches (Gu et al, 2010): storage optimisation and picking optimisation.

Starting by the SOP, the Traveling Salesman Problem (TSP) is a possible method, for the DC's improvement. In the TSP, given a starting point and a set of cities, the travel salesman must find a minimum cost route that visits all the cities (Bernardino et al, 2017). The solution of a TSP represents a Hamiltonian cycle, that is a route through all edges-weighted in a graph, in which the travel salesman visits each edge just once (Tong et al, 2014). This algorithm can be adapted to rectangular warehouses, named Steiner TSP (De Koster et al, 2007). Another method could be applying the Order Batching Problem (OBP). According to Wäscher (2004), OBP aims to minimise the total length of picking tours by grouping the orders. The orders must then be assigned to batches while determining the route for every batch that minimizes length and time. The Sequential Zone Picking is an order picking system that was developed to improve DCs performance (Frazelle et al, 1994). This system has been used in various DCs and has the advantage of being easy to implement. Despite handling only one order at a time and having the disadvantage of reducing the pick-rate of pickers, this procedure has the advantage of maintaining order integrity throughout all the picking process (Petersen, 2000). Inverse picking is a pick-to-part system that inverts the systems of order picking, i.e., the Stock Keeping Units (SKU) are delivered successively by a conveyer system that is connected to an automated storage and retrieval system. Finally, different methods to improve the DC's efficiency can be united (Scholz et al, 2017) like the Batch Assignment and Sequencing Problem (Pinedo 2016), Joint Order Batching, Assignment and Sequencing Problem (Hen 2015) and Joint Order Batching, Assignment and Sequencing and Routing Problem.

On the other hand, given the introduction of new sales channels, such as e-commerce, customers are trending to make late orders instantly from a computer or an application (Tompkins, 2010), and demanding tighter delivery time windows. To deal with fast response times during picking, dynamic order picking systems are a good solution since they allow pick-list changes during picking execution (Lu et al, 2016). These systems are also good for solving problems like the arrival of urgent orders and to detect inconsistencies between requested orders and the items picked. Gong et al (2008) proposes a dynamic system to address this problem, but only manages to solve it through a heuristic that limits the assignment of orders depending on the picker's location. To address this problem, Lu et al (2016) develop an Interventionist Routing Algorithm (IRA) that calculates the minimal distance route that the picker must travel, even after receiving new information, regarding new requests during the picking process. Zhang et al (2017) mention that on-line batching solutions are required to improve efficiency and customer service level and suggested a time window batching divided into fixed time window batching and variable time window batching.

4. Methodology - IRA

As highlighted in section 2 and 3, the problem addressed in this study is on the picker's route regarding the replenishment of the automatic picking. To solve this problem, an algorithm is used for the routing optimisation in order picking, IRA (Lu et al, 2016). It considers that the warehouses, to which it applies, are rectangular and have parallel aisles, all of equal length. Aisles can be travelled in both forward and backword directions and there can be changes of direction within an aisle (i.e., the picker does not have to fully traverse the aisle). This algorithm was initially proposed by Lu et al (2016) for a single picker working in a one block warehouse, meaning that there are 2 cross-aisles that allow the passage from one aisle to the other. IRA is based on a manual order picking system, in which the picker is guided by a list of orders. The operator starts the operation at any point of the warehouse, and then starts the process of fulfilling the orders that are in the list, and after all orders have been completed, the picker returns to a depot point. The list is built according to the orders, which can be received dynamically before or after the picker begins its tour, since not all orders are known at the beginning of the tour. In practice, the list of orders can be updated during the picker tour to meet new order more efficiently.

Notations

The following notation is used in the algorithm:

- G: matrix with binary values, that registers all the warehouse points and the SKU that needs to be attended.
- m: number of the items to be picked.
- n: number of aisles in the warehouse.
- i: index for the requested item.
- v: a location in the warehouse, where v_i represents the storage locations for each requested item (with I = 1, 2, …, m), v_0 represents the depot, and v_p represents the location of the picker.
- j: index for an aisle.
- p: index for the aisle the picker is currently located at.
- d: index for the aisle of the depot.
- f: index for the first pick-aisle.
- r: index for the last pick aisle
- a_j - back (top) b_j - front (bottom), endpoints of each aisle.
- A_j: subpart of G which only consists of a_j and b_j, and all SKU's within the aisle j.
- L_j: Partial Route Subpart including aisles 1, …, 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 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, $L_j+ = L_j- \cup A_j$.

In this algorithm, the warehouse and pick locations are modulated in a binary matrix. The letter m indicates the number of items to be picked and n the number or aisles. Then, it is possible to associate vertices vi, i = 1, ..., m, to the storage item locations and the points aj and bj, j = 1, ..., n, representing respectively the top and bottom of each aisle. G is then the matrix that represents the layout of the warehouse, including the SKU and pick locations.

Definitions

- Route Subpart: denoted as L, refers to a possible order picking route that is a subpart in the warehouse matrix G, which includes at least once the points v_i for i = 0, 1, 2, ..., m. The shortest order picking route is a route subpart with minimum length.
- Partial Route Subpart (PRS): for any Route Subpart $L \subset G$, it is denoted $L_j \subset L$ for j = 1, 2,..., n, that consists of all the points (v_i with i = 1, 2,..., m) and aisle endpoints (a_j and b_j with j = 1, 2,..., n,) to the left of aisle j in G, such that the minimum length route subpart is obtained by exhaustive searches of all the possible combinations of PRS.
- Equivalence class for PRS: Each possible combination of PRS L_j (for j = 1, 2, ..., 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 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).

To solve this routing problem, the IRA considers the following criteria:

- Interruptibility: the possibility of being able to readjust the picker route, when a new order is added to the existing pick list and when the picker has already started the route. This criterion implies that the picker can start at an arbitrary position of the graph, but continues go through all points, ending in the depot;
- Optimisation: the result obtained should be as little as possible;
- Complexity: computer processing time is polynomial and linear to the number of pickers;
- Centralised depositing: the order picker starts and ends the pick cycle in the depot point, which is determined and predefined at the beginning of the algorithm.

Considering these four criteria, and to determine the shortest path in the picker route within all PRS combinations of each aisle this algorithm is based on the optimisation algorithm proposed by Ratliff et al. (1983). However, Ratliff et al's algorithm does not consider the interruptibility criterion, so Lu et al (2016) modified the algorithm so that the picker can start at any point in the warehouse. To do this, the following extensions have been added to the initial algorithm:

- Allow starting location of the picker to be inside the aisle: which consequently led to new arc configurations between the vertices, for a picker to be able to exit an aisle and led to new algorithm initiation procedures;
- Enter the new traveling area, One Way (OW): there are 2 types of traveling areas, OW and Round-Trip (RT), which are defined according to the position of the picker relative to the location of the depot and other items;
- Add new PRS equivalence classes: 7 new PRS equivalence classes have been added, in which 6 of them are proposed for the new traveling area, OW, and the rest for RT;
- Add new route construction tables: 5 new route construction tables were developed to deal with new travelling areas and PRS equivalences, and 2 new tables by Ratliff et al (1983) were modified.

The bullet points above mention the travelling areas, which are related to starting point of the route and the location of the depot. The travelling areas are defined as:

- OW- is an area where the picker travels only one way between corridors. If the total number of arcs in adjacent cross-aisles in any PRS is odd, then the area between these 2 runners is OW;
- RT- is an area in which the picker moves in cross-aisles in both directions. If the total number of arcs, in the cross-aisles, in any PRS between two adjacent runners is even, then the area between these runners is an RT.

These travelling areas are assigned to aisles according to their relationship to the vp and depot locations. It also implies that the equivalence classes, defined in section 4.1.2, are related to the travelling areas and some equivalences can only be used in one travelling area. In the OW, the picker only passes through aisle j once, and its possible equivalences are: (U, 0.1C,), (0, U, 1C), (E, U, 2C), (U, E, 1C) and (U, E, 2C). For example (U, E, 1C) means that PRS has an uneven degree parity in aj, even parity in bj and connectivity of 1 component. In RT, the picker pass through the corridor twice and its possible equivalences are: (U, U, 1C), (U, U, 2C), (0, E, 1C), (E, 0,1C), (E, E, 1C) and (E, E, 2C).

The configuration of the arcs for the entry and exit of the picker in a certain aisle, or between 2 adjacent corridors is another subject considered in IRA. In route configuration, a route with the minimum path cannot have more than two arcs between 2 vertices (Ratliff et al, 1983). Therefore, there are 6 possible arc configurations to enter an aisle and 4 to leave it (Figure 1).

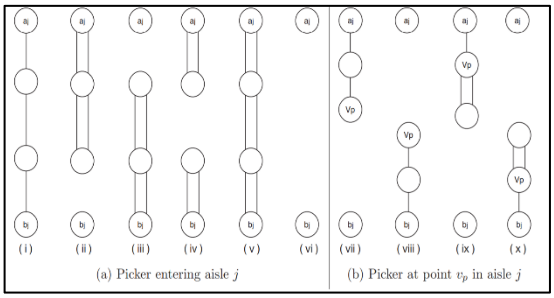

Figure 1 – Picker's route possibilities for travelling inside an aisle- Lu et al, 2016

Following the same reasoning for cross-aisles connections, there can also be no more than 2 arcs between 2 vertices. Since the connections between corridors may have to be even (RT) or odd (OW), there are a total of 8 possible configurations (Figure 2).

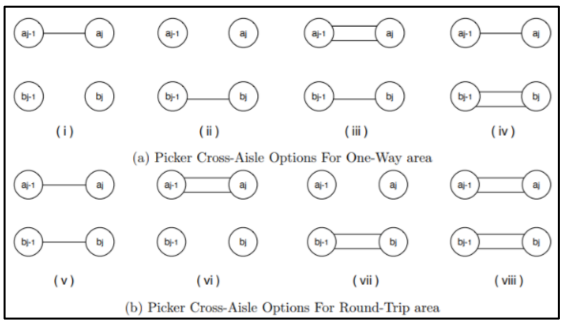

Figure 2 – Picker's route possibilities for crossing inside an aisle- Lu et al, 2016

After having all the arcs' configurations identified, the 3 phases of the IRA are described in the following section.

Algorithm Description

Given the definitions and notations presented above, it is now possible to detail the algorithm. IRA consists of 3 steps: 1) initiation; 2) transitions, and 3) route construction. After executing these steps, it is possible to establish the route with the minimal distance travelled.

To begin with, it is necessary to build a binary matrix in which the depot, the location of the picker and the required items are represented by "1"-initiation step the aisles are numbered from left to right incrementally, with two exceptions: i) the picker is on the right side of the depot or, ii) the picker is in the depot and there are no items on the left side. In both cases, the aisle numbers are increasingly numbered from right to left.

When a new order is added to the pick list, the binary matrix is updated and a new route is obtained. Routes are not optimized when a picker is crossing cross-aisles since these distances are insignificant compared to that of a normal runner. Two situations may occur for the beginning of a route construction regarding the location of the picker:

1. the picker is inside a pick-aisle;
2. the 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 exit the pick-aisle, the algorithm: a) assumes that no item is requested from the current pick-aisle in question; b) calculates the potential travel distances of leaving through either exit; c) determines the overall travel distance in addition to the distance of travelling to the associated exit; d) chooses the exit (endpoint) with the shortest overall travel distance and generate the associated routing. In situation 1, the algorithm calculates the route as in situation 2, for the point a_j and b_j, then adds the corridor exit distances. At the end, the route with the least sum of distances is chosen.

The second step is transitions, in which IRA calculates how the picker must travel through the aisles and how to leave them. There are two types of transition: L_{j-1}+ for L_j- and L_j- for L_j +. In the first transition (L_{j-1}), it is determined which of the 8 possible ways of the figure 2 can be added to the equivalence class of L_{j-1} + PRS. The value of the minimum distance PRS is obtained by choosing the lowest value of the PRS of these classes, which is the combination with the shortest distance.

- If aisle j − 1 and j are both in RT area, and the OW area does not exist before aisle j − 1, Table 2a applies.
- If aisle j − 1 and j are both in RT area, and the OW area exists before aisle j − 1, Table 2b applies.
- If aisle j − 1 is in RT area or j − 1 = f, and aisle j is in OW area, Table 3a or 3b applies if the order-picker starts from endpoint a or b respectively.
- If aisle j − 1 and j are both in OW area, Table 5 applies.
- if aisle j − 1 is in OW area, and aisle j is in RT area, Table 6 applies.
- if aisle j − 1 is the depot aisle, and the picking route will start from the endpoint a (i.e. the order-picker will start from the head of the depot aisle a), Table 2c applies.

In the second transition, it is determined which of the 6 possible ways of the figure 1 can be added to the equivalence class of L_j- PRS in each of their equivalence classes.

- If aisle j − 1 and j are both in RT area, and the OW area does not exist before aisle j − 1, Table 1a applies.
- If aisle j − 1 and j are both in RT area, and the OW area exists before aisle j − 1, Table 1b applies.
- If aisle j − 1 is in RT area or j − 1 = f, and aisle j is in OW area, Table 4a or 4b applies if the order-picker starts from endpoint a or b respectively.
- If aisle j − 1 and j are both in OW area, Table 4c applies.
- if aisle j − 1 is in OW area, and aisle j is in RT area, Table 1b applies.
- if aisle j − 1 is the depot aisle, and the picking route will start from the endpoint a (i.e. the order-picker will start from the head of the depot aisle a), Table 1c applies.

#	L_j^- Equivalence Classes	Arc Configuration from Figure 1a (i)	(ii)	(iii)	(iv)	(v)	(vi)[a]
1	(U, U, 1C)	(E, E, 1C)	(U,U,1C)	(U, U, 1C)	(U, U, 1C)	(U, U, 1C)	(U, U, 1C)
2	(E, 0, 1C)	(U, U, 1C)	(E, 0, 1C)	(E, E, 2C)	(E, E, 2C)	(E, E, 1C)	(E, 0, 1C)
3	(0, E, 1C)	(U, U, 1C)	(E, E, 2C)	(0, E, 1C)	(E, E, 2C)	(E, E, 1C)	(0, E, 1C)
4	(E, E, 1C)	(U, U, 1C)	(E, E, 1C)	(E, E, 1C)	(E, E, 1C)	(E, E, 1C)	(E, E, 1C)
5	(E, E, 2C)	(U, U, 1C)	(E, E, 2C)	(E, E, 2C)	(E, E, 2C)	(E, E, 1C)	(E, E, 2C)
6	(0, 0, 0C)[b]	(U, U, 1C)	(E, 0, 1C)	(0, E, 1C)	(E, E, 2C)	(E, E, 1C)	(0, 0, 0C)
7	(U, U, 2C)	(E, E, 1C)	(U, U, 2C)	(U, U, 2C)	(U, U, 2C)	(U, U, 1C)	(U, U, 2C)

Table 1a contains rows 1–6
Table 1b contains rows 1–7
Table 1c contains rows 1,2,3,4,6,7

[a] This is not a feasible configuration if there is any item to be picked in aisle j.
[b] This class can occur only if there are no items to be picked to the left of aisle j.

Figure 3 – IRA: Arc configurations for applying Table 1- Lu et al, 2015

#	L_{j-1}^+ Equivalence Classes	Arc Configuration from Figure 2 (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 2a contains rows 1–5
Table 2b contains rows 1–6

L_{j-1}^+ Equivalence Classes	Arc configuration from Figure 2 (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 2c: if the picker starts the tour on the head of the depot aisle.

[a]No completion can connect the graph

Figure 4 – IRA: Arc configurations for applying Table 2 in IRA- Lu et al, 2015

L_{j-1}^{+} Equivalence Classes	Arc configuration from Figure 2 (i)	(ii)	(iii)	(iv)
(U, U, 1C)	—[a]	(0, U, 1C)	(E, U, 1C)	—[a]
(E, 0, 1C)	(U, 0, 1C)	—[a]	—[a]	—[a]
(0, E, 1C)	—[a]	—[a]	—[a]	(U, E, 2C)
(E, E, 1C)	(U, 0, 1C)	—[a]	—[a]	(U, E, 1C)
(E, E, 2C)	—[a]	—[a]	—[a]	(U, E, 2C)

Table 3a: for picker start at the head (a-point) of the aisle

L_{j-1}^{+} Equivalence Classes	Arc configuration from Figure 2 (i)	(ii)	(iii)	(iv)
(U, U, 1C)	(U, 0, 1C)	—[a]	—[a]	(U, E, 1C)
(E, 0, 1C)	—[a]	—[a]	(E, U, 2C)	—[a]
(0, E, 1C)	—[a]	(0, U, 1C)	—[a]	—[a]
(E, E, 1C)	—[a]	(0, U, 1C)	(E, U, 1C)	—[a]
(E, E, 2C)	—[a]	—[a]	(E, U, 2C)	—[a]

Table 3b: for picker start at the head (a-point) of the aisle

[a]No completion can connect the graph

Figure 5 – IRA: Arc configurations for applying Table 3 in IRA- Lu et al, 2015

#	L_j^{-} Equivalence Classes	Arc configuration of Figure 1 (i)	(ii)	(iii)	(iv)	(v)	(vi)[a]
1	(U, 0, 1C)	(E, U, 1C)	(U, 0, 1C)	(U, E, 2C)	(U, E, 2C)	(U, E, 1C)	(U, 0, 1C)
2	(0, U, 1C)	(U, E, 1C)	(E, U, 2C)	(0, U, 1C)	(E, U, 2C)	(E, U, 1C)	(0, U, 1C)
3	(E, U, 1C)	(U, E, 1C)	(E, U, 1C)	(E, U, 1C)	(E, U, 1C)	(E, U, 1C)	(E, U, 1C)
4	(E, U, 2C)	(U, E, 1C)	(E, U, 2C)	(E, U, 2C)	(E, U, 2C)	(E, U, 1C)	(E, U, 2C)
5	(U, E, 1C)	(E, U, 1C)	(U, E, 1C)	(U, E, 1C)	(U, E, 1C)	(U, E, 1C)	(U, E, 1C)
6	(U, E, 2C)	(E, U, 1C)	(U, E, 2C)	(U, E, 2C)	(U, E, 2C)	(U, E, 1C)	(U, E, 2C)

Table 4a contains rows 1,2,3,5,6
Table 4b contains rows 1,2,3,4,5
Table 4c contains rows 1–6

[a]This is not a feasible configuration if there is any item to be picked in aisle j.

Figure 6 – IRA: Arc configurations for applying Table 4 in IRA- Lu et al, 2015

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

Table 5a contains rows 1,2,3,5,6
Table 5b contains rows 1–5
Table 5c contains rows 1–6

[a]No completion can connect the graph

Figure 7 – IRA: Arc configurations for applying Table 5 in IRA- Lu et al, 2015

L_{j-1}^{+} Equivalence Classes	Arc configuration from Figure 2 (i)	(ii)	(iii)	(iv)
(U, 0, 1C)	(U, U, 2C)	—[a]	—[a]	—[a]
(0, U, 1C)	—[a]	—[a]	(0, E, 1C)	—[a]
(E, U, 1C)	—[a]	(E, 0, 1C)	(0, E, 1C)	(E, E, 1C)
(E, U, 2C)	—[a]	—[a]	—[a]	(E, E, 2C)
(U, E, 1C)	(U, U, 1C)	—[a]	—[a]	—[a]
(U, E, 2C)	(U, U, 2C)	—[a]	—[a]	—[a]

[a]No completion can connect the graph

Figure 8 – IRA: Arc configurations for applying Table 6 in IRA- Lu et al, 2015

To understand which travelling areas associated to the aisles and which tables to apply, Figure 3 represents a flow chart for the processes of the algorithm. The figure represents the details procedure of IRA after the construction of the binary matrix. Figures 10 and 11 represent how the route constructions operate and how to use them. It is possible to see that there is one procedure for the route construction in OW area and 3 route construction procedures for RT areas, for different cases.

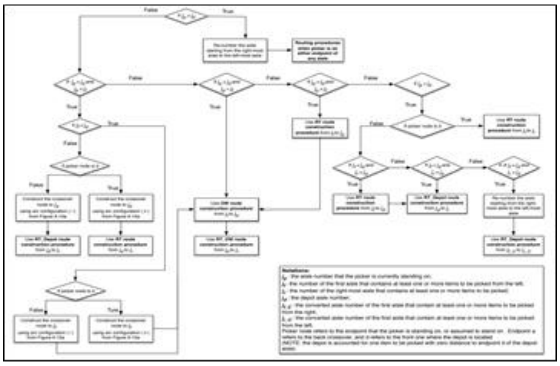

Figure 9 – IRA's flowchart- Lu et al, 2015

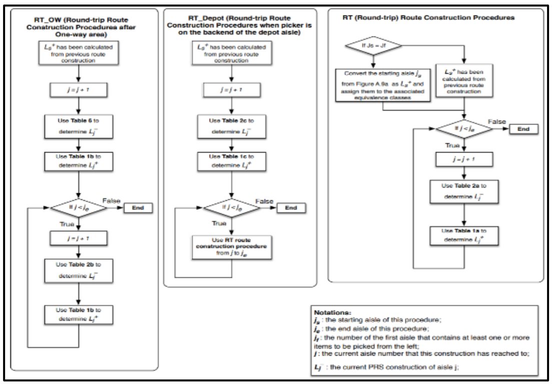

Figure 10 – Round Trip Route Constructions- Lu et al, 2015

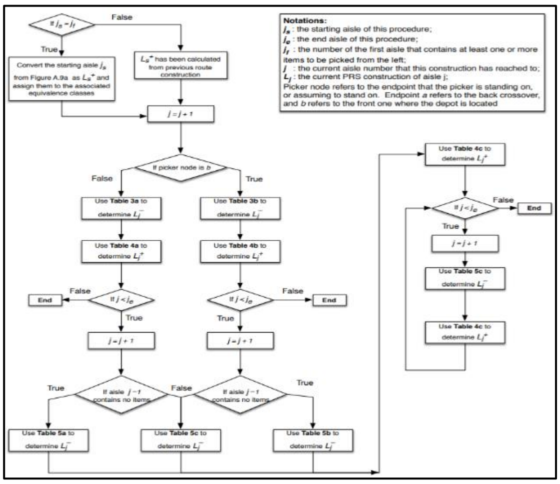

Figure 11 – One Way route construction- Lu et al, 2015

The final step of the algorithm is the route construction. After the transitions step is completed, a final table is achieved with different solutions that contain the final equivalence class of the aisle, the total distance travelled, the route applied on that aisle and the predecessor equivalence class. Then the route construction step is developed in two parts: the first one is to sum the distance value travel by the picker to leave the first aisle, and the second is to choose the minimum distance travelled of the table, that is a feasible solution. If the last aisle is in a RT area, the feasible solutions are the ones that have the equivalence class: (E,0,1c), (0,E,1c) and

(E,E,1c); and if it is in a OW area: (0,U,1c) and (E,U,1c).

To obtain the final route travelled by the picker, it is chosen the minimum feasible cumulative distance, and the previous routes and equivalences are traced back through the predecessor equivalence. At the end of this process, it is possible to see the minimal route the picker has to travel, from the aisle f to r.

Multi-picker expansion

```
Binary Matrix= ([a1,a2,a3,a4,a5,a6],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [0,0,0,0,0,0],
                [b1,b2,b3,b4,b5,b6]))
```

Figure 12 – Example of Binary Matrix

IRA is an algorithm designed for single picker systems. However, most warehouses operate with multiple pickers, so this method must be expanded to handle these situations. The best way to achieve this end is, upon the arrival of a new request, execute the algorithm on the routes of all pickers. After the execution, if the final route is equal to the initial one, it means that the picker would already go through this point and nothing changed either on the route or on the total distance travelled. If distances are different, the decision-maker needs to decide to which picker he wants to add the item. The most efficient solutions in this case are: 1) add the order to the picker that has the smallest route, or 2) add the order to the picker that changes less its initial route. This is a feasible and efficient procedure due to the small execution time of IRA.

5. Computational Experiments

Is important to remember how automatic refilling of Udifar II works. The current system generates routes through the FCFS system, which can imply large unnecessary travel. As discussed in chapter 2 when there is a lack of product in automatic picking, the system detects this need and signals it, in a refill list. The list is constructed considering only the order by which each need is verified (FCFS). It was based on these lists that the required data was collected to determine the actual routes.

There are 30 routes used by the pickers in the automatic picking refuelling task which were randomly collected in the week of July 23rd to 27th, 2018, in a real operating context of Udifar II.

Orders with 8 requests (average of products per request) are used.

The algorithm is coded in Python using Jupyter Notebook. The arrangement of the Udifar II automatic picking warehouse leads to the matrix (Figure 12), where "0" represents the possible SKU's and "1" will represent the dispensers that need replenishment. In addition, it should be noted that the distance between aisles is 4 meters. It was defined that the end-of-route point of all routes is point b_4 since it is an intermediate point (there are a total of 6 aisles) and therefore, with good flexibility to start new routes, in case of new requests.

The 30 orders, mentioned in the last section, are executed by IRA's algorithm.

Table 1 shows the results obtained, comparing the current solutions with the ones obtained by the algorithm. This table includes 6 columns:

1. Route index number of the sample considered (from 1 to 30);
2. The actual distance travelled, in meters by the picker;
3. The results of IRA algorithm, in meters;
4. IRA's running time, in seconds;
5. The optimisation obtained using the route suggested by IRA, in meters versus the actual distance travelled;
6. The optimisation obtained using the route suggested by IRA, in percentage.

ROUTE	Udifar II PICKERS TOUR (m)	IRA		SAVINGS	
		TOUR (m)	running TIME (seconds)	(m)	(%)
1	130	74	1,6244	56	43,1
2	116	68	2,2309	48	41,4
3	147	69	2,4685	78	53,1
4	149	95	2,5306	54	36,2
5	153	113	2,1762	40	26,1
6	182	92	3,0672	90	49,5
7	155	115	3,0048	40	25,8
8	137	94	2,3813	43	31,4
9	137	99	2,8292	38	27,7
10	160	81	2,7816	79	49,4
11	128	55	1,7266	73	57,0
12	189	92	3,0400	97	51,3
13	164	116	2,8689	48	29,3
14	161	115	2,9908	46	28,6
15	142	96	3,4655	46	32,4
16	154	112	2,9100	42	27,3
17	152	101	2,0703	51	33,6
18	169	97	2,8079	72	42,6
19	173	85	3,1635	88	50,9
20	156	90	3,0206	66	42,3
21	142	94	2,9973	48	33,8
22	142	110	2,7914	32	22,5
23	185	104	1,5032	81	43,8
24	156	112	2,8192	44	28,2
25	157	109	2,7559	48	30,6
26	103	93	2,3596	10	9,7
27	181	114	3,0493	67	37,0
28	188	106	2,7544	82	43,6
29	157	85	3,1105	72	45,9
30	182	116	3,0180	66	36,3
Average	154,9	96,7	2,6773	58,2	37,0

Table 1 – Results of the computational experiments

Discussions of Results

For the 30 instances, the average distance travelled by pickers is 154.9 meters (the minimum and maximum distances are 103 and 189 meters, respectively). Using the IRA algorithm to determine the routes, it is possible to obtain a very significant difference in distance associated with each route, and the average distance travelled was 96.7 meters, with a minimum and maximum of 55 and 116 meters, respectively. These results, for the samples considered, lead to an average saving of 58.2 meters, which corresponds to 37%, relative to the distances travelled by the pickers, which allows to conclude that the IRA algorithm is quite effective in optimising routes. The large variations of the real sample distances versus the calculated distances can be explained by the fact that Udifar II uses an FCFS system, which means that the picker may have to cross the same aisles several times, leading to inefficiencies in the task of replenishing the automatic picking system.

Regarding processing time, IRA took an average of 2.6773 seconds, on an i7 processor with 12 gigabytes of memory, proving to be a fast, flexible and efficient algorithm. Although this value is quite different from the execution time obtained by Lu et al. (2016), 7 milliseconds, it remains quite reliable. This difference can be explained by a more efficient programming of the algorithm. The efficiency of the implementation is not a concern in this work. And thus one can conclude that IRA is a powerful algorithm to be implemented in this environments.

To sum up, with the implementation of IRA, Udifar II automatic picking replenishment system would become much more efficient and effective, because with the current FCFS system, the picker passes through each point at most twice and the routes are optimised based on the totality of the requests for each list. IRA is a better algorithm for Udifar II, since it determines very quickly each route and allows a better use of resources (the picker travels less distance, consuming less time in the replenishment of each list). Implementing IRA translates in significant gains, due to the high number of routes performed in this type of task, as well as increasing the picker’s happiness and consequently improving the flexibility of this procedure. On the other hand, the speed of calculation of the IRA makes it possible to make the replenishment system even more dynamic and effective, since IRA makes it possible to recalculate routes. As an example, a picker is walking the route assigned, but the system has detected a new material failure in a certain location. Almost instantaneously, the picker's route can be recalculated - considering this new need - and allowing the failure to be solved, together with the others, through the smallest possible space, as soon as possible. This possibility is crucial because in situations where there is no product available, manual intervention is necessary to complete the clients order. This procedure results in loss of time, inefficient resource consumption and entropy throughout the process chain.

Although the IRA algorithm has been applied in the specific case of Udifar II, it can also be used in other similar environments, and it is expected that the results will be similar to those obtained in the case of Udifar II.

6. Conclusions

The health sector in Portugal is highly competitive and regulated, with tight sales margins defined by the state, through specific legislation, leading in recent years to reformulation of the sector to reduce costs associated with all the operations and processes involved throughout the whole supply chain. The drug storekeepers are no exception to this reality and considering the very high number of daily orders they must to respond, any optimisation process results in significant gains due to the scale effect.

It is analysed the case of the drug distribution companies specifically Udifar II DC, regarding its operation. As a result of this analysis some processes are identified that present problems or that show a margin of improvement. The replenishment of automatic picking is selected, defining as the main focus of efficiency improvement in Udifar II warehouse.

To better understand the importance of the subject addressed, the pharmaceutical market in Portugal is analysed. In relation to Udifar II, its internal operations are described: reception of merchandise, storage, order picking and shipment; being also analysed the possible areas of improvement of the company. The present study focuses on the optimisation of picker routes in automatic picking replenishment.

A literature review related to this issue is carried out. It covers the themes of DC, order picking and all the tools that can contribute to improve its performance. The IRA and VNA algorithms stand out from the others analysed not only by the speed of execution but also by their effectiveness.

Because it is considered that the IRA algorithm is a powerful algorithm to be applied in the case study, it is proposed for the automatic picking replenishment system of Agualva - Cacém DC of Udifar II. IRA is implemented in Python and tested using real instances from Udifar II. For each instance, a new route is defined to accommodate new orders received on the system and the associated distances are determined. When comparing to the results using the current policy of Udifar II, the results show an average reduction of the distances travelled of about 37%, corresponding, on average, to a reduction of 58.2 meters in each route, a result that is considered very significant. Moreover, the running time of the algorithm is between 2-3 seconds which shows that is a good algorithm to be implemented in the company.

In view of the results, it can be stated that the current FCFS strategy in Udifar II is not the most appropriate not only because it leads to longer travel distances in each route, but also because it is static, not allowing, for example, the insertion of the most frequently requested products.

The implementation of the algorithm in the Udifar II allows to increase the efficiency in the automatic picking replenishment, reducing the distance travelled by the pickers and consequently the time consumed by them in the execution of these routes, releasing resources that can be more effectively used, increasing the productivity at the level of replenishment. On the other hand allowing also to incorporate a high degree of flexibility in this activity since new orders that arise during the realization of a certain route can be considered immediately.

It should also be noted that the optimisation of the entire automatic picking replenishment process also reduces the time in which the stock out of the automatic picking is active. It should be recalled that the operators who, at the end of the automatic picking line, check if the clients requests are complete, i.e., if they contain all the products in the order list, when they realize the lack of some product they have to collect the missing product, which translates into process entropy, inefficiency and loss of time. Thus, improving the efficiency of the automatic picking replenishment process also reduces the number of incomplete client requests at the end of the tour, allowing the process to take place more fluidly and with fewer stops and delays.

It can be concluded that, despite the competitive environment in which the Udifar II DC works, it is possible to achieve a significant optimisation in the automatic picking replenishment process. Changing the current philosophy of FCFS for IRA‘s algorithm increase the efficiency of the warehouse, enabling a better use of resources and cost reduction.

Since the situation analysed refers to a one-block store, it is recommended to future studies to investigate the possibility of expanding IRA’s algorithm to warehouses that have more than one block dimension. If the IRA's applicability to this hypothesis is confirmed, the IRA algorithm can be used in most of the warehouses in the definition of routes.

7. References

Baker, P., Halim, Z., 2007. *An exploration of warehouse automation implementations: cost, service and flexibility issues*. Supply Chain Management: International Journal 12 (2), 129-138.

Baker, P., 2008. *The design and operation of distribution centres within agile supply chains*. International Journal of Production Economics 111(1), 27-41.

Bernardino, R., Paias, A., 2017. *Solving the family traveling salesman problem*. European Journal of Operational Research 267 453-466.

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(2), 219–234.

Davarzani, H., Norman, A., 2015. *Toward a relevant agenda for warehousing research: literature review and practitioners’ input*. Logistics Research, 8(1), 1-18.

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.

Frazelle, E. H., Apple, J. M., 1994. *Warehouse operations* in J. A. Tompkins & D. A. Harmelink (Eds.), The distribution management handbook, 22.1–22.36.

Gong, Y., De Koster, R., 2008. *A polling-based dynamic order picking system for online retailers*. IIE Transactions, 40(11), 1070-1082.

Gu, J., Goetschalckx, M., McGinnis; L.F., 2007. *Research on warehouse operation: a comprehensive review*. European Journal Operational Research, 177 (1), 1-21.

Gu, J., Goetschalckx, M., McGinnis; L.F., 2010. *Research on warehouse design and performance evaluation: A comprehensive review*. European Journal of Operational Research, 203(3), 539-549.

Hadjinsky A., 2013. *Logistics: Textbook for High society*. - Moscow: Marketing.

Henn, S., Schmid, V., 2013. *Metaheuristics for order batching and sequencing in manual order picking systems*. Computers & Industrial Engineering, 66, 338-351.

Henn, S., 2015. *Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses*. Flexible Services and Manufacturing Journal, 27(1), 86–114.

Lam, H., Choy, K., Ho, G., Cheng, S. W., Lee, C., 2015. *A knowledge-based logistics operations planning system for mitigating risk in warehouse order fulfillment*. International Journal of Production Economics, 763-779.

Lu, W, McFarlane, D., & Giannikas, V., 2015. Technical Report. University of Cambridge. Report No. CUEDE/E-MANUF/TR.35, ISSN: 0951-9338.

Lu, W., McFarlane, D., Giannikas, V., Zhang, Q., 2016. *An algorithm for dynamic order-picking in warehouse operations*. European Journal of Operational Research, 248(1), 107–122.

Marchet, G., Melacini, M., Perotti, S., Jan. 2015. *Investigating order picking system adoption: a case-study-based approach*. International Journal Logistics Research Applications 18 (1), 82-98.

McFarlane, D., Giannikas, V., Lu, W., 2016. *Intelligent logistics: involving the customer*. Computers in Industry, 81, 105-115.

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

Pinedo, M. L. 2016. *Scheduling: Theory, algorithms, and systems* (5th). Springer. Cham et al.

Ratllif, H.D., & Rosenthal, A. S., 1983. *Order-picking in a retangular house: solvable case of the traveling salesman problem*. Operation Research 31, 507-521.

Romão, A.L.S., 2012. *O Sector de Distribuição Farmacêutica em Portugal Estrutura e Estratégias*. Phd in Economics, Universidade do Minho.

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

Scholz, A., Schubert, D., Wäscher, G., 2017. *Order picking with multiple pickers and due dates simultaneous solution of order batching, batch assignment and sequencing, and picker routing problems*. European Journal of Operational Research, 263(2), 461–478.

Tompkins, J. A., White, J. A, Bozer, Y. A., Tanchoco, J. M. A., 2010. *Facilities planning* (4th ed.). Hoboken, NJ: John Wiley & Sons.

Tsai, C. Y., Liou, J. J., Huang, T. M., 2008. *Using a multiple-ga method to solve the batch picking problem: considering travel distance and order due time*. International Journal of Production Research, 46, 6533-6555.

Wascher G., 2004. *Order picking: a survey of planning problems and methods. Supply chain management and reverse logistic*s. Berlin: Springer; 2004. p. 323–47.

Zhang, H., Tong, W., Xu, Y., Lin, G., 2014. *The Steiner Traveling Salesman Problem with online edges blockages*. European Journal of Operational Research, 243, 30-40.

Zhang, J., Wang, X., Chan, F.T.S., Ruan, J., 2017. *On-line order batching and sequencing problem with multiple pickers: A hybrid rule Based algorithm.* Applied Mathematical Modelling, 45, 271-284.