Optimization of the Picking Process in Inventories

Le Phuong Quynh
Supervisor: Dr. Jüttner Alpár
Péter Madarasi

Eotvos Lorand University

December 2020

Overview

1. Introduction
2. Project Overview and Objectives
3. Set up
4. Evaluations and Analysis
5. Conclusions

Introduction

- Order picking, the process of retrieving items from their storage locations to fill customer orders, is known as the most time consuming and laborious component of the warehouse activities. The project aim to simulate this process in the warehouse. More specifically, picker-to-parts systems (i.e, the order picker travels along the aisles to retrieve products) are considered, as these systems account for the large majority of all order picking systems in Western Europe.

Previous work

- An optimizing problem: to reduce working time inside warehouse by avoiding collisions in picking products process
- A collision may arise when at a moment there are two or more workers wanting to get product at the same place. "First come first serve" rule states that a worker who comes first will have priority and all others need to wait until this man leaves

Scope of work

- First, as mentioned above, fixing bugs is the most important task before having to any further improvement.
- Second, cleaning code and implementing object oriented program (OOP).
- Third, evaluating the results.

Basic Conditions

1. There are variety of goods in the system. Each location in the warehouse can have only one type of goods. The demand for a variety of goods is fixed and known, at this time we also assume that we need not to care about the amount of goods in stock, i.e, the remaining goods always exceeds demand.
2. To simplify the processing, here have only one speed for all workers, said a maximum speed. Moreover, except the collision mentioned at the beginning, we consider all other traffic jams during working time are neglectable. It helps us to have a fixed information of traveling time between any two locations.

Basic Conditions

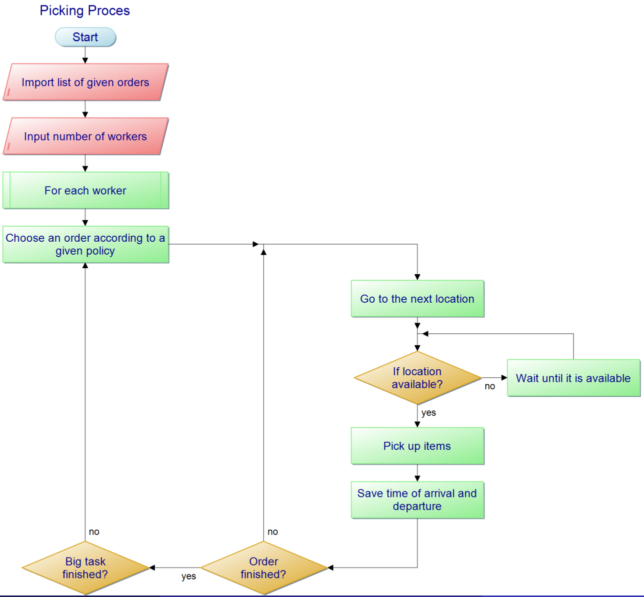
3. **Picker to Order**: the picker is responsible for collecting items related to an order. With a detailed list of the items to be collected, the picker travels to the locations where the goods are stored, picks the needed amount and move the the next location. After finishing an order, the picker comes back to the starting point to leave items for packaging and delivery process then get a new order here.

Programming Set Up

Develop a Controller class to hold a dictionary of pickers and their picking processes in the warehouse

- A dictionary is initially empty.
- Picker takes an order from big task according to a policy, new order is taken only if the previous order is done.
- Picker can be looked up by key in the dictionary, which is numbered from 1.
- A value in dictionary is the process of that picker from the start to the end including locations, arriving and leaving time that locations and waiting time if collision happens.
- User can look up a location at a time to check if there is any collision.

Programming Set Up



Programming Set Up

In this report, I implemented 3 new policies for choosing the next order for a picker. All of three policies will consider the time point where a picker has finished the previous task and already comeback to the starting location.

At that specific time point, we can check for every location that if it is occupying by other pickers or not. If yes, we can calculate the time needed till that location would be free, which is call estimated queuing time

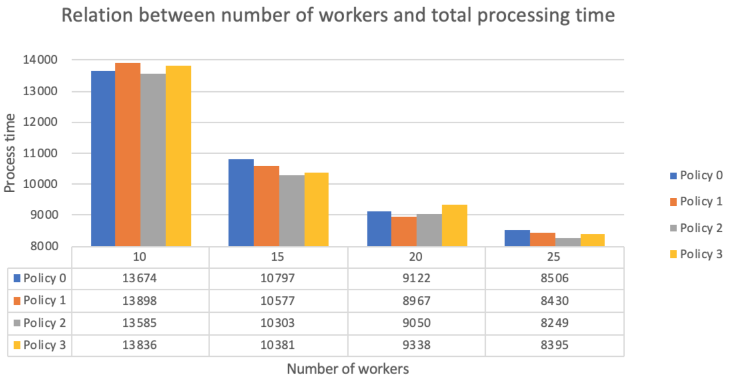
Programming Set Up

- Policy 1: minimum sum of queuing time from locations
- Policy 2: minimum average queuing time, i.e, we take sum of queuing time divide by number of locations contained in that order and choose the one has min result
- Policy 3: minimum ratio between queuing time and the estimated processing time of an order (total time a worker will need if we get rid of all collisions)
- Policy 0: simply the case of choosing a random order from list of given orders.

Sample

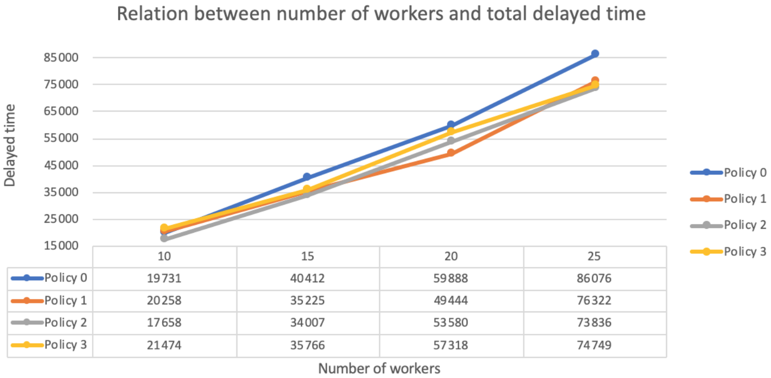
We use a sample including 185 orders for an inventory containing 20 product locations applying to policies 1, 2 and 3. We also change pickers in range from 10 to 25 people to see the trend of picking process if we increase or decrease the number of workers

Result



	10	15	20	25
🟦 Policy 0	13674	10797	9122	8506
🟧 Policy 1	13898	10577	8967	8430
⬜ Policy 2	13585	10303	9050	8249
🟨 Policy 3	13836	10381	9338	8395

Result



	10	15	20	25
Policy 0	19731	40412	59888	86076
Policy 1	20258	35225	49444	76322
Policy 2	17658	34007	53580	73836
Policy 3	21474	35766	57318	74749

Conclusions

Order picking is often considered as the most crucial warehouse activity. Any inefficiency in order picking can lead to unsatisfactory service and high operational cost for its warehouse.
The simulation system is considered in this project is a simple one. We can extend it in many directions. Planning for the next semester, I will keep continuing improve the algorithms for more realistic calculating collisions.

THANK YOU FOR LISTENING