

# COMP4106-Project: Warehouse navigation bot

---

Carleton University  
Winter 2019

Artificial Intelligence  
John Oommen  
Assignment 2

By: David N. Zilio (Skote2)

Proposal: Wednesday 7<sup>th</sup> March 2019 Demo: TBD

TA: Karim

The course project is an arbitrary AI project selected by me, the student. The Project is worth 30% of the course grade and expected to take 2 times the amount of time of a regular assignment.

## Proposal

---

### Defining the problem domain

Amazon is known for their unique and advanced solutions to the problem of inventory management. Amazon manages their warehouses with bots. The bots wander the warehouse to stock it, and collect inventory for orders. Given the nature of how Amazon receives orders it often needs to retrieve inventory immediately after an order is made so that the product can make shipment times from two days to as low as an hour, near their more advanced warehouses.

### Motivation for the problem

The efficient prioritization, management and timely achievement of the tasks at hand ensure a warehousing system with no bottleneck. By getting items to customers faster, Amazon has managed to create a world with near instant access to magnitudes more items than retail could provide. A better marketplace which is more convenient and saves people time while giving them access to lower prices is only made possible with the strong backbone of a responsive warehouse. The warehouse cannot bottleneck and while it provides a reliably faster but also cheaper alternative to retail.

Ultimately this is a difficult challenge. The semantics of managing logistics has crippled many a company before and will cripple many more.

### The bots' AI

The solution used must be efficient which means metrics must be used against it

### Means of gauging performance

One of the obvious metrics is whether the bot has taken the optimal path through its shopping list. To determine the optimal path the factory can be turned into a finite state space, therefore being observed as a graph which would have the optimal path found by Dijkstra's algorithm.

Determination of whether a bot is making valuable decisions would be based on times for when an item was ordered, the priority it was ordered with and the cost of the item.

Therefore performance metrics can be listed as:

1. Distance traveled to deliver shopping list over Dijkstra's calculated distance
2. bonus for clustering of multiple items collected
3. Value of delivered items as the: sum of all items' (dollar value times priority multiplier times (total time minus time left))

### **Potential means of the bots' implementation**

- Hive mind
- neural nets
- genetic algorithms
  - competitive bots breeding
- state space solution
- collecting multiple items at once
- picking up new items for its "shopping list" while actively searching so that higher priority items get out faster, if necessary.
  - item time restrictions
  - item values

A number of different designs could be implemented first thing is to determine the base of whether to implement a hivemind infrastructure or to implement a client server based approach. For the hivemind one algorithm would control all of the bots in the factory given them directions step by step or a list of instructions. The alternative approach is to have bots act as agents who can either be given shopping lists by a overarching algorithm or select the items they want to collect themselves.

The hivemind implementation leaves less interesting implementation options, it restricts the algorithm implementations to a single overarching algorithm in charge of the bots' movement. The same applies for the option of having the server determine the agent's next shopping list item. If the bot can choose its own path and choose the next shopping items it can make for a very interesting system where the bots could all be running different algorithms. Although potentially less functional this design would provide far more variations of the data and allow for further investigation into what makes a good algorithm.

I believe I'll implement client server and if I have the time the bots will pick the shopping list items they want. The bots will do their pathing and prioritization by a genetic algorithm which will produce different algorithms running across many bots. This will create a space where the bots can compete for breeding potential based on their scores. The underlying algorithm which is being bred will likely be a simple feed forward neural network design. The neural net would be instantiated to random values over some standard variation and have a randomized design. The more successful neural nets will be allowed to breed and reimplemented into the least successful bots. The process will be allowed to go on constantly running the warehouse until a desirable performance metric can be met by the general population of the bots.