WAREHOUSE STRATEGY

Richard Lawton

17/04/2020

Updated on 2020-04-30

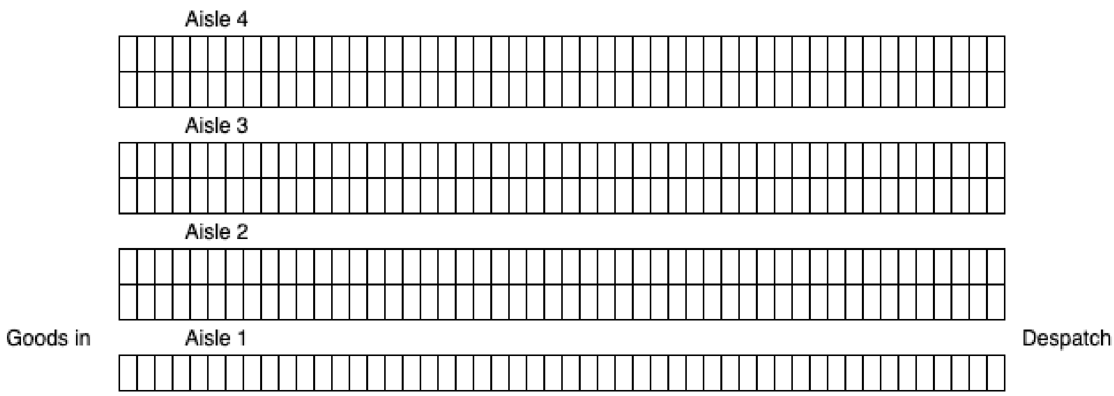
INTRODUCTION

The efficiency (and hence cost) of a warehousing operation is directly related to the ease of placement and retrieval of goods. This investigation sets up a notional warehouse and investigates the effect of a range of storage location strategies through Monte Carlo simulation. Is the optimum strategy to place product randomly, or should certain locations be reserved for certain products? The model is designed to simulate a multi-product, bulk warehouse that handles only single pallet loads: no breaking of pallets occurs.

METHODOLOGY

Warehouse geometry

The warehouse configuration was set for this project as an array of 10 aisles, 5 tiers high and holding 100 storage locations (bins) along each aisle. (These dimensions are parameterised in the model and could be altered.)



1. Simple warehouse

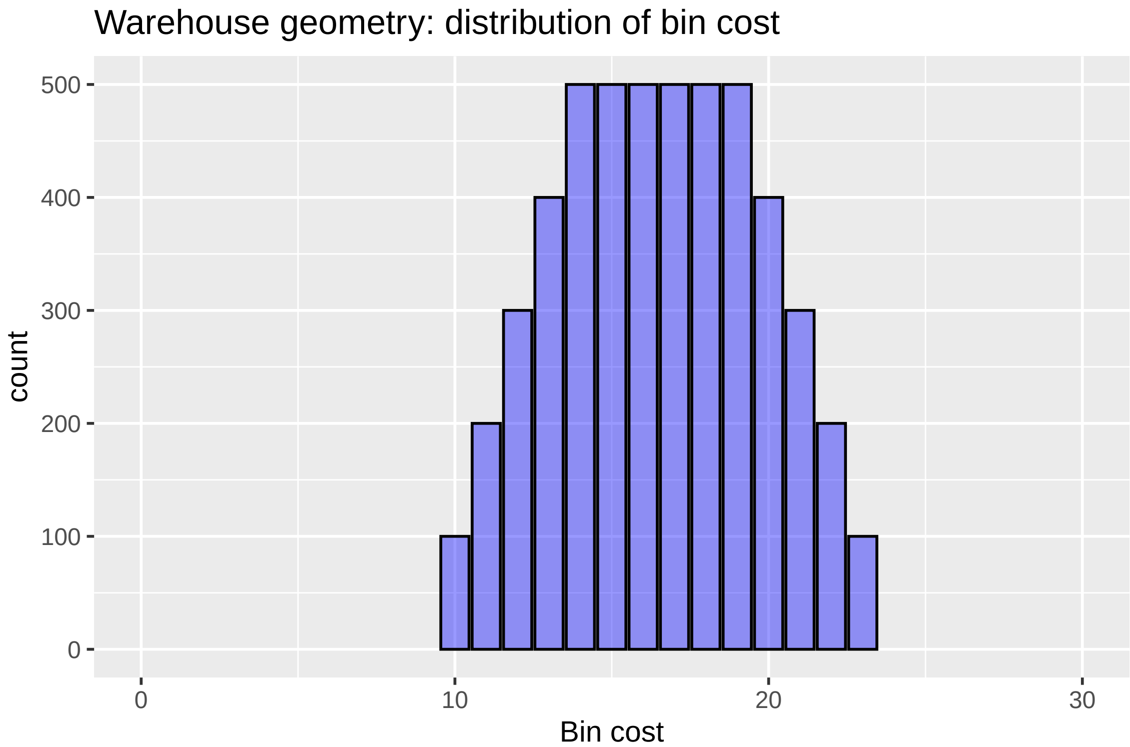
Within this geometry, goods receiving is conceived to be located at one end of the first aisle and despatch at the far end of the same aisle. The baseline (easiest) journey for placing a pallet then consists of travel from goods receiving down aisle 1 to the bin location. For despatch the pallet has to travel down the remainder of the aisle to the far end. Whichever bin it is stored in, the pallet will ultimately travel the full length of the aisle. All bins at a given level on a given aisle will thus have equal ease of access.

Bins in other aisles beyond aisle 1 will necessitate a greater distance of travel, because lateral movements will also be involved, increasing with number of the aisle. Likewise upper tiers will incur additional vertical travel. A marginal cost is therefore associated with every bin location, calculated as:

$$cost(n_{aisle}, n_{tier}) = n_{aisle} - 1 + h(n_{tier} - 1) + 100/a$$

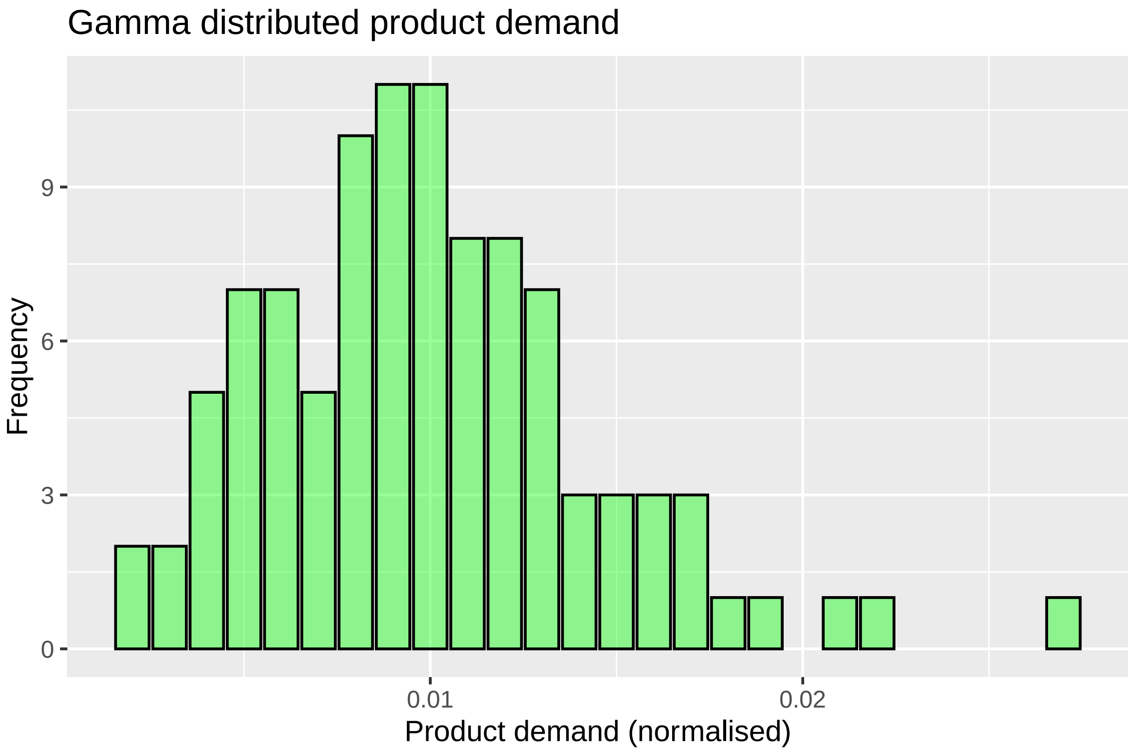
where h is a parameter weighting the cost of vertical movement relative to horizontal, and a is cost of moving to the next aisle relative to moving to the next bin. For this simulation h was set at 1 and a at 10.

This geometry gives rise to the following frequency of bin costs:



Product demand

The focus of the investigation lies in strategies for handling products with different demand levels. Should more popular products be placed in cheaper locations? It seems intuitively correct, since more movements will then take place in lower cost bins. In order to model this, we created a range of products (in this case 100) with product demand distributed along a gamma curve. This skewed distribution was selected to give a lower limit cut-off at 0 (a product cannot have negative demand!), a peak of products with moderate demand and a tail towards higher demand products. The demands were randomised, and regenerated for each run of the model. A sample distribution is shown below:



Product stock was initially placed in the warehouse in porportion to the product demand, at a level of 80% loading to capacity (again parameterised for future investigation).

Monte Carlo simulation

Once the warehouse was constructed, 10000 stock movements of a single pallet were simulated on a randomised basis. The movements were placement or despatch (on a 50/50 probability), with product chosen randomly in proportion to the product demand. The possibilities of stockouts or stock overflows were catered for, but never encountered. With each movement the bin cost was accumulated towards an overall total. Multiple runs of the simulation for a given set of parameters produced very little variation in results: it is therefore unnecessary to perform more than single runs.

Stock placement strategies

1.) Baseline

The baseline for this study is the assumption that all bins are equal. Stock is initially placed in random order in the warehouse. Subsequent incoming stock is placed randomly in vacant bins, and despatch is chosen randomly from bins containing the required product.

2.) Easy access

Stock is initially placed in random order in the warehouse, filling the bin locations from the lowest cost up. Subsequent incoming stock is placed in the lowest cost empty bin. Despatch is drawn from the lowest cost bin containing the required product.

3.) Prioritised

This scheme envisages prioritising higher demand products: the highest demand products are given exclusive access to the lowest cost bins, followed by the second highest demand product etc. Incoming product may not be placed in a bin reserved for a product with higher demand, but is placed in the lowest cost bin to which it is allowed access.

4.) Fast access

This scheme envisages a fast access zone consisting of fixed number of low cost bins reserved for each product, regardless of demand. For these simulations the number of fast-access bins was set at 5. Stock arriving is placed in the fast-access bins reserved for the product in question, or, if they are full already, in the lowest cost bin available beyond the fast access zone. Stock is drawn from the lowest cost bin.

Stock rotation strategies

1.) None

No attempt is made to rotate stock.

2.) FIFO

In order to maintain stock rotation, the pallet of product that was placed first is always selected for despatch. If two or more pallets have the same placement cycle number (which can only occur with initial loading at cycle=0) then the lowest cost bin is selected.

3.) Modified FIFO

FIFO turns out to be an expensive strategy, as we shall see. Modified FIFO is designed to moderate the cost, while maintaining a level of stock rotation. A notional age limit is set beyond which stock must be rotated. For this simulation the limit was set at 2000 cycles. Stock that is old, but still under this limit may be left sitting in a high cost location while fresher stock drawn from low cost bins, but stock over this limit must be drawn for despatch.

The logic of this can be understood if we consider a very simple warehouse of one product with n_0 bins at cost $= 0$, and n_1 bins at cost $= c$. All the bins are initially stocked. We then envisage repeated cycles of drawing one unit and then restocking with one unit.

With the baseline strategy the product will always be drawn and replaced in the zero cost bins. The net average cost is therefore zero, but no stock rotation occurs and the product in the higher cost bins will never move.

With the FIFO strategy, the first n_0 cycles will draw from the zero cost bins, but then the oldest stock will be located in the higher cost bins, and the next n_1 cycles will draw from the high cost bins. The oldest product is now located in the low cost bins, and the whole sequence will be repeated. The net average cost will therefore be total cost incurred/number of cycles $= c \times n_1/(n_0 + n_1)$.

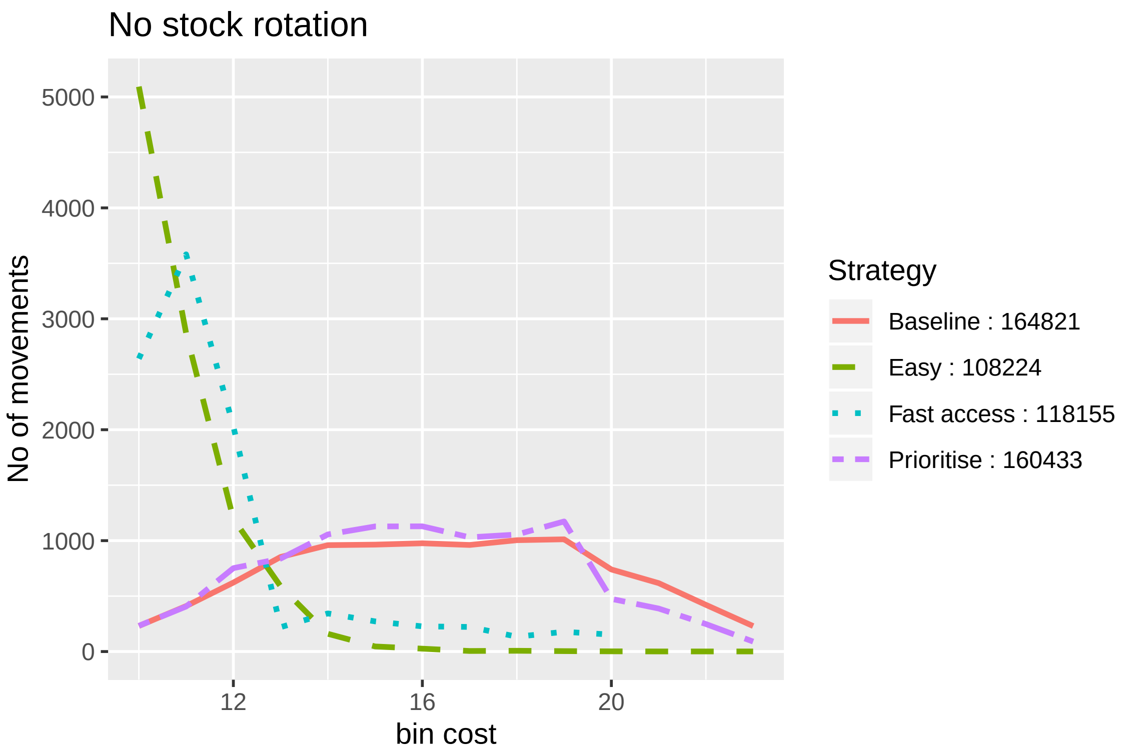
With the modified FIFO strategy with an age limit A, the first A cycles will draw from the zero cost bins, then the next n_1 cycles will clear the high cost bins. Then the whole sequence is repeated. The net average cost will therefore be $c \times n_1/(A + n_1)$. By making A considerably larger than n_0 it is clear that we can cut down the cost of stock rotation.

RESULTS

A range of strategies was simulated over 10,000 cycles with randomised product demand distributions and product movements. Repeat runs of a given strategy with different starting points showed little variation. The results shown here are therefore for single runs.

No stock rotation

The key goal is to maximize the number of transactions taking place in low cost bins (and therefore minimize the number of transactions from high cost bins). The distributions achieved under four strategies was as follows:



The baseline strategy (shown as a red solid line) carries a cost of 164,821 over the 10,000 cycles. By adopting the strategy of always accessing the cheapest (=nearest) bin we can reduce this cost dramatically to 108,224 (shown in green dashes). By contrast the "clever" srategy of prioritising low cost bins for high demand products actually works terribly - the cost is scarely better than the baseline: 160,433 (shown in purple dot-dash), while the "fast access" strategy of reserving 5 low cost bins for every product comes close to the "easy" strategy 118,155 (shown in blue dots), but is still not as good.

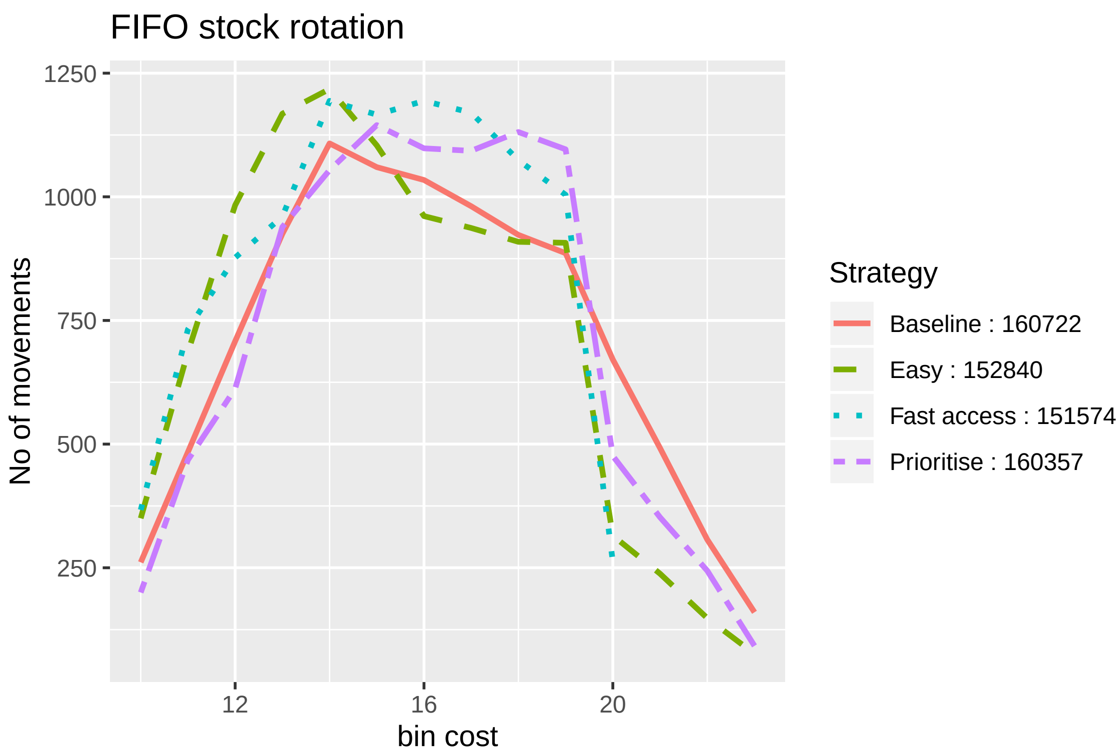
Why did the prioritisation strategy work so badly! A little reflection will explain why our intuition here was wrong - when we reserve bins for high demand product, we are reserving a high number of bins. Under this strategy a large number of low cost bins will be filled with product that never needs to move, since the high demand can be met by repeatedly filling a small number of bins.The lower demand product, on the other hand, will always have to move from more expensive bins.

The fast access strategy was designed to correct this error, but the cost is still significantly higher than that of the "easy" strategy, for a much more complex system.

Based on these results we can save 34% on baseline costs by using the "easy" strategy.

Including FIFO stock rotation

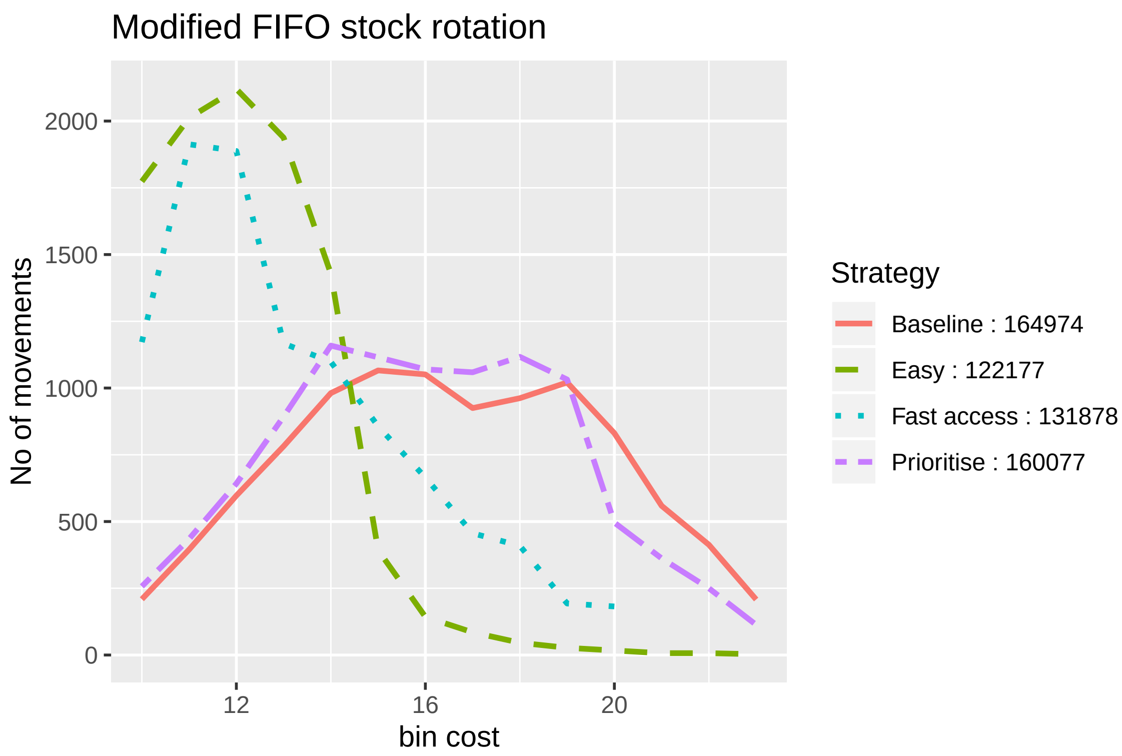
With stock rotation the results looked markedly different:



FIFO ruins everything! The baseline cost actually decreases marginally from 164,821 to 160,722 (although this is probably not significant), but our best options ("easy" and "fast access") increase dramatically to 152,840 and 151,574 respectively. The potential gain from implementing the "easy" strategy has been cut to 5%. Can we help by implementing the modified FIFO strategy descibed above?

Including modified FIFO stock rotation

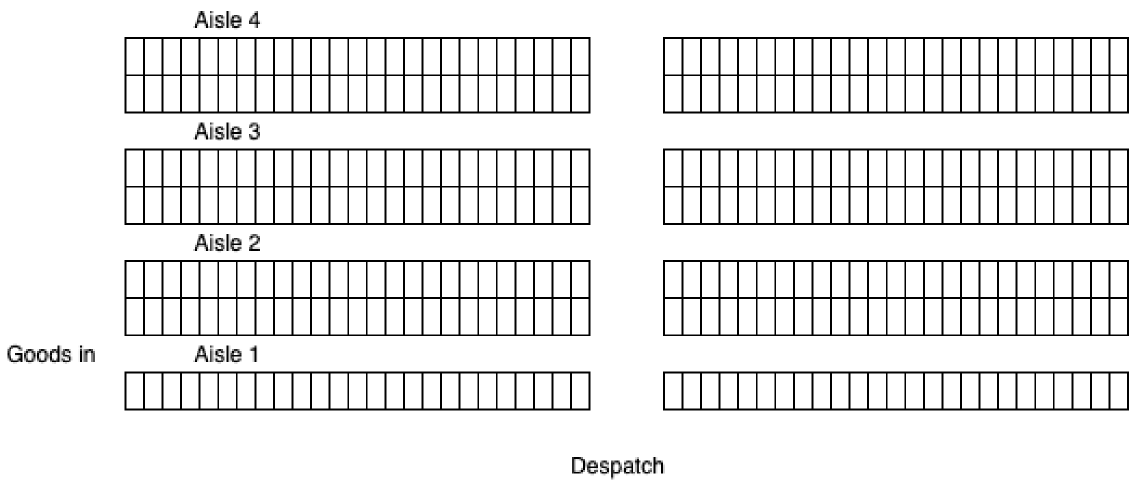
We do indeed see some improvement:



The “prioritise” strategy does not improve significantly, but “Easy” and “Fast access” do. “Easy” is still the best strategy and offers a potential gain of 26% over the baseline.

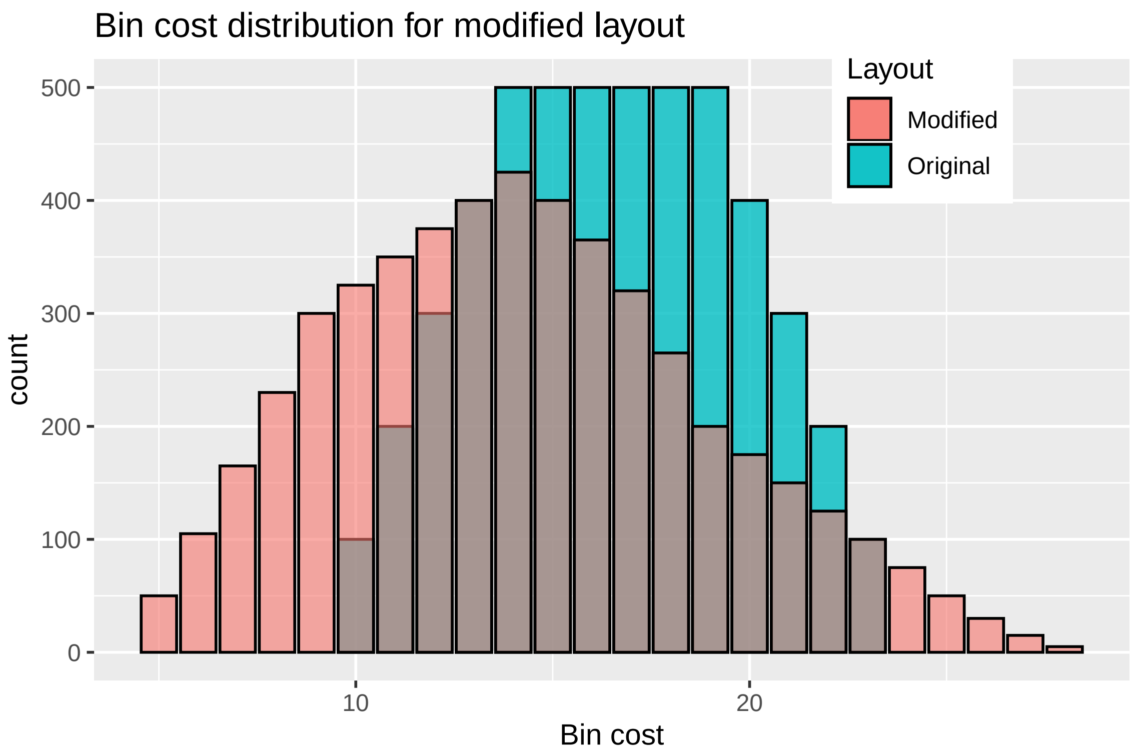
Modifying the geometry

Is it possible that the results we have obtained are peculiar to the specifric geometry we have used? To test this we adopted a different warehouse layout:

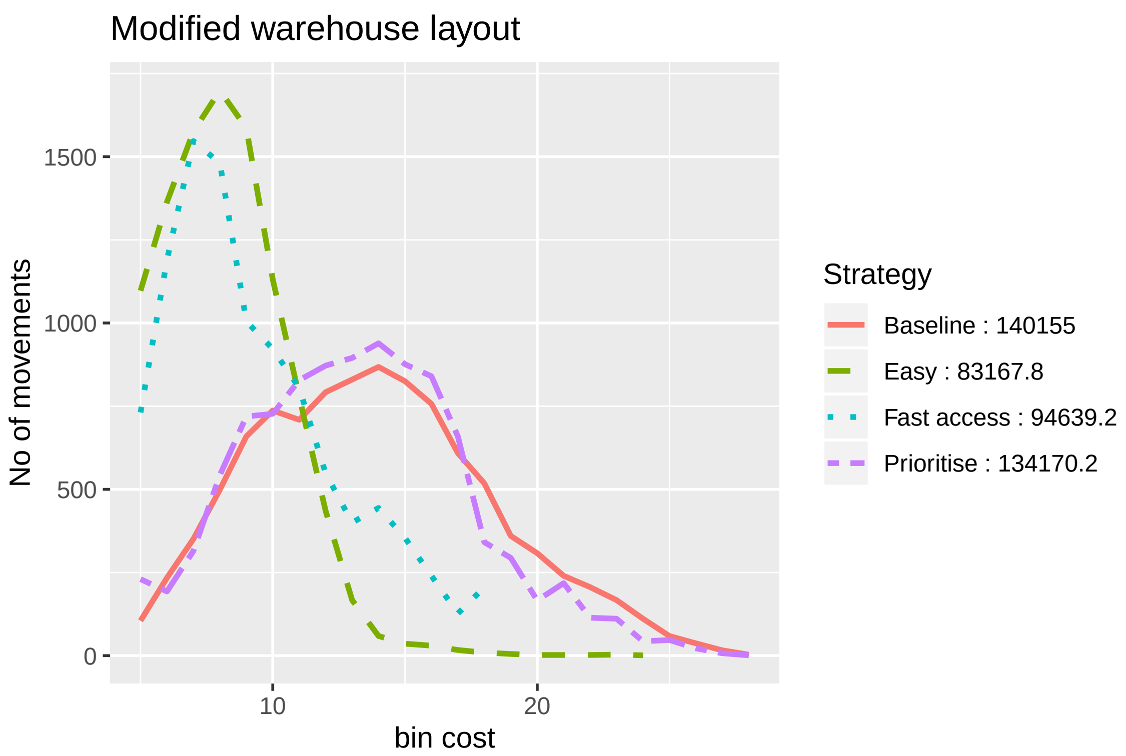


2. Complex warehouse

This modified arrangement changes the cost function, since bins *beyond* the central aisle now incur an incremental charge deoending on location. The distribution of bins costs shifts lower and spreads:



How does this affect our results? Rerunning the last trial with modified FIFO yields the following results:



Overall the pattern has not changed. The costs have come down a bit, since the warehouse layout is now more efficient. The cost benefit of using the "easy" strategy has now increased to 41%

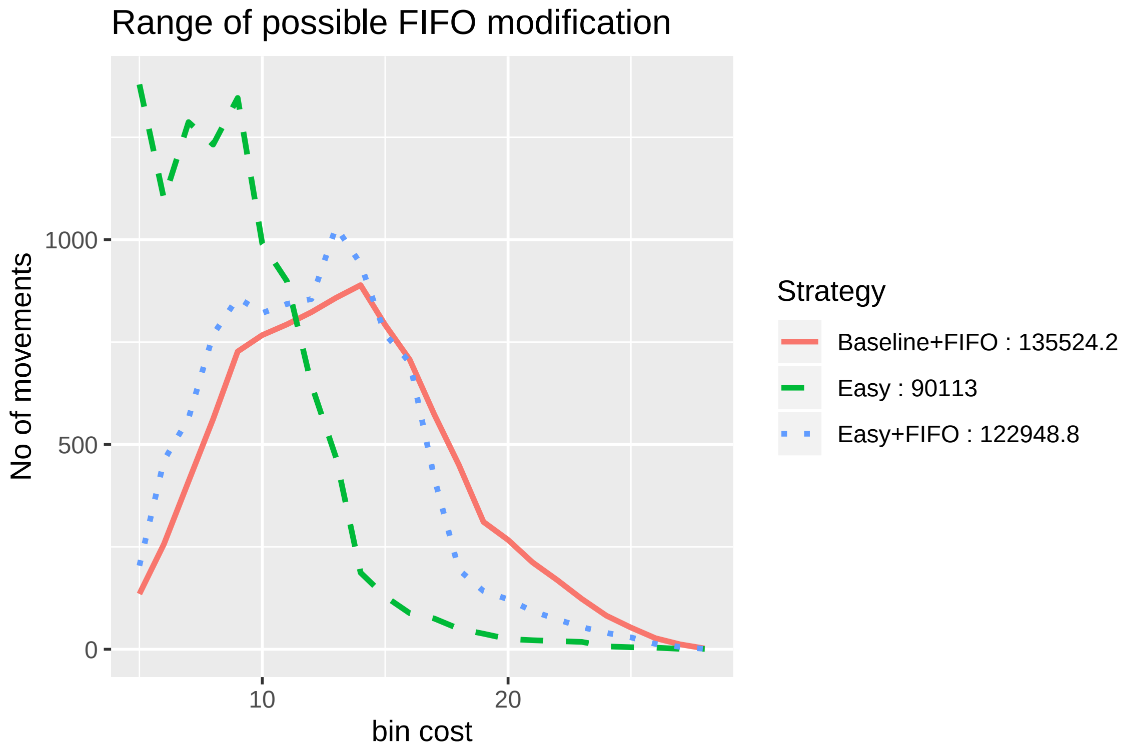
Bulk arrivals

An artificiality with our model so far is that product is presumed to arrive in single units. In fact delivery to the warehouse will almost certainly be in container-loads. How is our model affected if product arrives in, say, lots of 20 units?

The pattern is similar, but our advantage has been reduced. The "Easy" strategy now yields a cost saving of 28%, which is still considerable.

Tuning the age limit parameter

The winning formula appears to be easy placement combined with FIFO modified by allowing stock to age to a certain extent. Can we tune this parameter? In fact the results are self-evident. If we set the parameter to zero (i.e. the stock may not age at all), we effectively have pure FIFO. On the other hand if we allow the parameter to become very big (i.e. the stock can age without restriction) then we effectively have no FIFO. So these two extremes will define the limit of our tuning range. Let's see how these compare with a baseline + FIFO model:



Easy stock placement with no FIFO yields a cost saving of 34%, while adding in full FIFO redices this to 9%. By tuning the age limit parameter we could expect to achieve cost savings between these two numbers.

CONCLUSIONS

The conclusions are then twofold:

1.) The simple policy of using the cheapest (= easiest access) bin location cannot be improved by any more sophisticated strategy

2.) Stock rotation is undoubtedly necessary, but a strict FIFO strategy carries a very high cost. A better plan is to allow some ageing of stock.

How far this limitation of FIFO can go is obviously product dependent. A warehouse implementing this policy would probably need to have age limits for individual products, but could expect savings up to 34%.
R script is available at https://github.com/rtlawton/warehousing