Jack O'Neill
AE 5222
Homework 6

Method

For this problem needed to generate a graph which would describe all the possible paths with their respective costs. I could then use Dijkstra's algorithm (the code I borrowed from Joseph Kirk which can be found at https://bit.ly/2Zzu1uG) to calculate the minimal cost (distance) path which would put all the pallets in the correct bins.

For this problem I defined the "6th" bin as the bin defined by b_T. Nodes in this problem are defined as a single bin and pallet configuration, of which there are 720 combinations. To define adjacency I reasoned that each configuration would have a number of adjacent nodes. They were defined by swapping each pallet to b_T. Numerically this looked like arrays, each with the next array index swapping with the last index. With these adjacent nodes I could then search the 720 combinations to find which node they were defined as. Doing this for all nodes allowed me to generate a 720x720 adjacency matrix. The code snippet below builds up an edge matrix E which gives the respective distances between all adjacent nodes.

```
for n = 1:length(Nodes)
    for swap_idx = 1:6
        swapped_config = swap_with_bt(Nodes(n).config,swap_idx);
        for i = 1:length(Nodes)
            if isequal(Nodes(i).config, swapped_config) &&...
                    (n ~= i) && (swap_idx ~= swapped_config(end))
                E(end+1,:) = [n i dist_mat(swap_idx,swapped_config(end))];
                Nodes(n).adjacent_nodes(end+1) = i;
            end
        end
    end
end
```

Figure 1: Defining the edge matrix

For full context the full code has been attached to the end of this report.

To define the distance of each edge, I used the distance matrix shown in figure 2 to find the distance between the two bins which were being swapped in that edge.

	b_1	b_2	b_3	b_4	b_5	b^{T}
b_1	–	66	94	64	132	124
b_2	66	–	36	54	70	145
b_3	94	36	–	52	72	140
b_4	64	54	52	–	118	92
b_5	132	70	72	118	–	209
b^{T}	124	145	140	92	209	–

Figure 2: Distance matrix

For example, say the initial configuration is: {1 2 3 4 5 6}. The cost to transition to the adjacent matrix {6 2 3 4 5 1} would be equal to the distance from b_1 to b_T which is 124. The edge matrix is defined this way for the rest of the edges.

<u>Results</u>

My Matlab code generated the following optimal sequence of pallet ordering given the initial configuration and an end configuration equal to {1, 2, 3, 4, 5}:

```
------ Calculated Minimum Cost Path ------

Path cost = 492

     p1 |  p2 |  p3 |  p4 |  p5 |
     2     5     1     3     4     6

     2     5     1     6     4     3

     2     5     3     6     4     1

     1     5     3     6     4     2

     1     2     3     6     4     5

     1     2     3     6     5     4

     1     2     3     4     5     6
```

Figure 2: Calculated optimal pallet ordering

Where the cost is 492, and each row represents a configuration of bins (1-5 plus 6 as b_T).

Discussion

The logic I followed to complete this problem seems to make sense to me, but overall it was pretty confusing to determine the adjacency conditions and I very well could have messed up at some point along the way. However the path generated by Dijkstra's algorithm shows that it is satisfying the condition that the forklift can only hold one pallet at a time so I am happy with this result.