

Yuan Qin

yqin7@uci.edu

May.31st

Final report:

1. Batch process all orders:

As shown in the following snippet, we can choose not to visualize in GUI and manually input the starting and end point. Then input and output .csv file could be entered. At last the program starts to process after you put in the last argument to choose whether you calculate the effort or not.

The optimized result and its corresponding shortest path distance is stored in the optimized2500_1.csv

```
/usr/local/bin/python3.6 /Users/YvetteQ/PycharmProjects/eecs221/graphplot.py
Hello User, whether you want to visualize an sample order: 1 for yes? 2 for no?2
Total goods num: 25525
Max rack number in row, col 18.0 10.0
Total item number: 19305
Hello User, where is your worker? please enter: x,y: if exceeds,default 0,0
> 0,0
What is your worker's end location? please enter x,y: if exceeds, default 0,20
> 0,20
Hello User, input manually: yes? no?no
Please list order file name:
>warehouse-orders-v02-tabbed.csv
Please list output file name:
>optimized2500_1.csv
Whether compute effort? 1 for yes, 2 for no1
The order ready to pick: [108335, 391825, 340367, 286457, 661741]

Distance for one order without optimization 76
Computing greedily shortest distance to travel .....
after shift: [391825, 340367, 286457, 661741, 108335]
mindist: 70
one optimized: [340367, 661741, 286457, 391825, 108335]
after shift: [340367, 286457, 661741, 108335, 391825]
mindist: 70
one optimized: [340367, 661741, 286457, 391825, 108335]
after shift: [286457, 661741, 108335, 391825, 340367]
mindist: 70
one optimized: [661741, 340367, 286457, 391825, 108335]
after shift: [661741, 108335, 391825, 340367, 286457]
mindist: 70
one optimized: [661741, 340367, 286457, 391825, 108335]
Minimum travel distance: 70 ,in order of: start from (0, 0) [340367, 661741, 286457, 391825, 108335] , end at (0, 20)
go to shelf: [5, 3] on location: [11, 7] pick up item: 340367 , then
go to shelf: [5, 3] on location: [11, 7] pick up item: 661741 , then
go to shelf: [6, 3] on location: [13, 7] pick up item: 286457 , then
go to shelf: [5, 7] on location: [11, 15] pick up item: 391825 , then
go to shelf: [10, 5] on location: [21, 11] pick up item: 108335 , then
```

```
one optimized: [1400573, 1400573, 1145768, 392015]
Minimum travel distance: 56 ,in order of: start from (0, 0) [1400573, 1400573, 392015, 1145768] , end at (0, 20)
go to shelf: [9, 0] on location: [19, 1] pick up item: 1400573 , then
go to shelf: [9, 0] on location: [19, 1] pick up item: 1400573 , then
go to shelf: [5, 5] on location: [11, 11] pick up item: 392015 , then
go to shelf: [6, 6] on location: [13, 13] pick up item: 1145768 , then
drop off at: [0, 20]
Nearest neighbor cost: 0.16811180114746094
total effort 30.4
number processed: 2500
write into file.....

Process finished with exit code 0
```

The end of output.

—Regarding the left and right issue:

In nearest neighbor:

Every time when computing the next destination, it would compare whether the current location on coordinate x is larger than the next one or not, because of the intuition of nearest neighbor, if the current x is smaller than the next x of the shelf

location, than choosing the left side of the rack would be intuitively shortest since it saves at least a distance of the length along x coordinate of a shelf, so would be the larger case.

In branch and bound:

lower bound would be computed according to the relative distance between the two consecutive nodes to be traversed. Then shortest path would be computed afterwards along the optimized order of the order.

- Regarding the issue while there's no item info for certain item, :
default weight is set to 0.1 lb. Effort is calculated under this assumption.

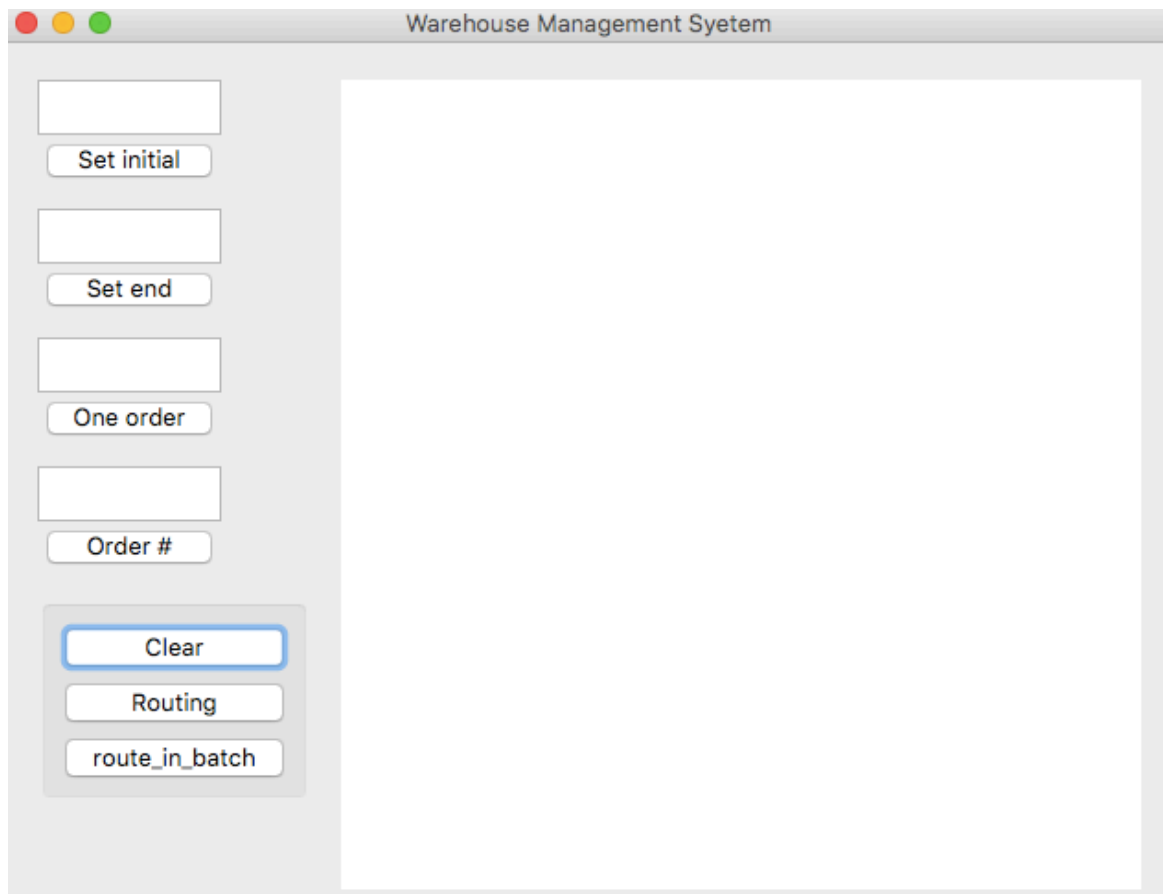
2. GUI interface

1) User-put-in order list:

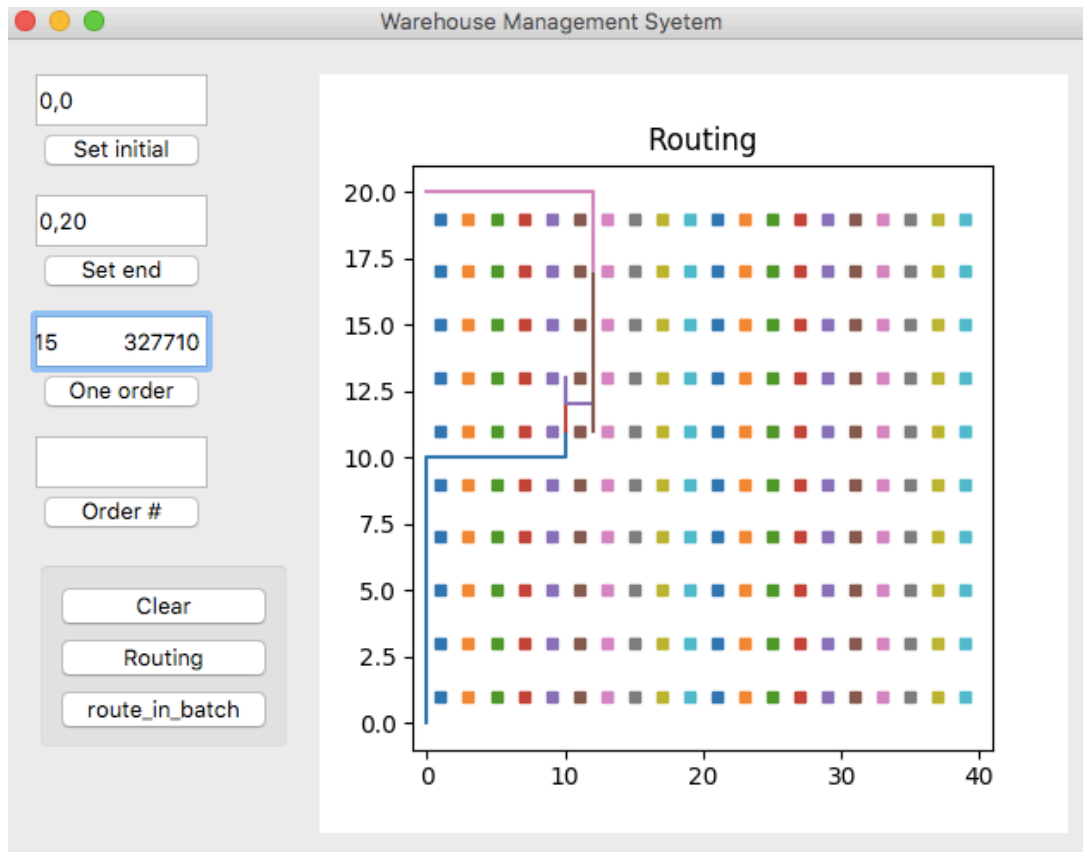
- As shown in the following snapshot, initial point and end point could be entered, and one order could be entered with item numbers separated by tabs.
- Don't forget to click the set buttons each time new values entered, otherwise the initial or end value won't be updated or it would use the default value.(when nothing ever has been entered)
- To show the routing, click the *clear* button to either clear or display the rack/shelf position, click the *routing* button to paint out the routing lines. Command line input to select algorithm used.

2) Select from batch file:

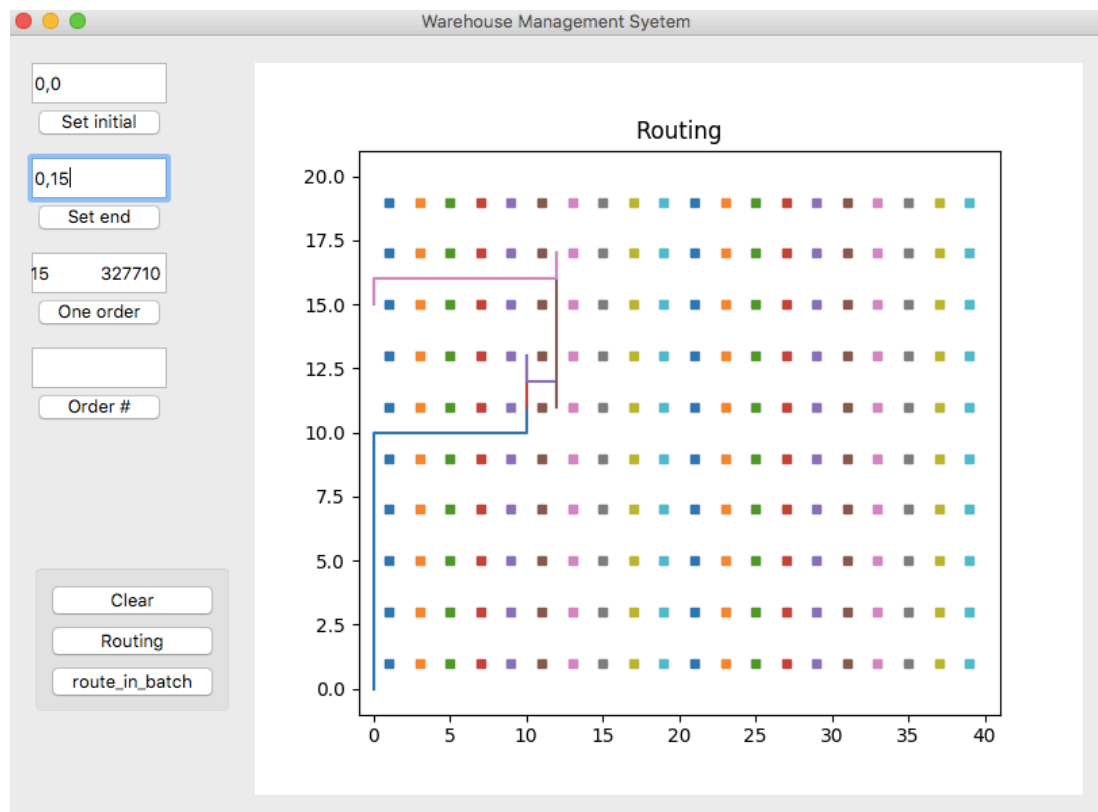
- Clear* the window and then input the order number specified in the batch file. No use to input starting and ending point in this scenario since it is set to (0,0) and (0,20) which is precomputed in the .csv file.
- Click *route_in_batch* to visualize route for this certain optimized order number.



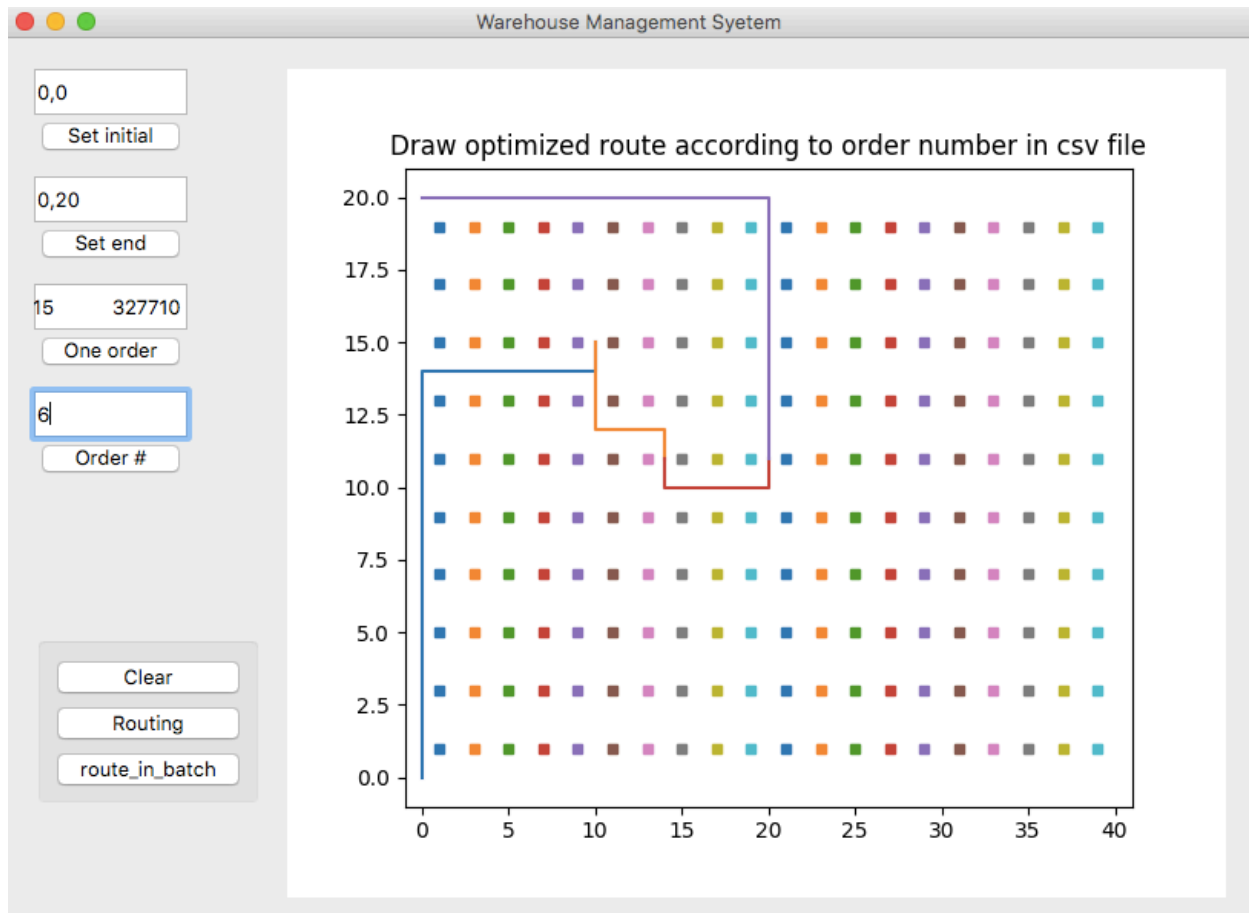
welcome, click *clear* to initialize grid



Set initial and end point, input order by items



Reset end points and routing again



Read optimized order # from csv file and click Order# then route_in_batch button to visualize

3.maximum weight limit

- Implementation: choose whether split/rearrange orderlist or not, in which stored the total information of orders and items.
- In my implementation, I combine orders if total weight does not reach 50
- if total item number is larger than 9, split order for quicker routing
- if total item weight is larger than 50 lb, split
- if single item weight is larger than 50, set 1 item per order

```
Rearranged order list: [[108335, 391825, 340367, 286457, 661741, 219130, 365285, 364695, 422465], [379019]
Please list output file name:
>optimized100_rearranged.csv
Whether compute effort? 1 for yes, 2 for no
The order ready to pick: [108335, 391825, 340367, 286457, 661741, 219130, 365285, 364695, 422465]

Distance for one order without optimization 126
Please choose algorithm, 1 for nearest neighbor, 2 for branch and bound:
```

Reordered order are processed and saved in the *optimized100_rearrange.csv* file

4. Instructions

Please run from command line:

- `python3 graphplot.py`
- or (`python3 warehouseapp.py`) without GUI implementation

GUI input:

- Initial and end point input: separated by comma
- One order should be entered separated by tab, please copy from the .csv order file
- Order # is according to the order optimized and stored in optimized2500_1.csv file, the initial and end point is (0,0) and (0,20)
- Each run:
 1. Clear
 2. One order —>Routing
realtime computing routing, you have to input method(b&b or nearest neighbor in run time) onto command line after you click Routing button.
 3. Order # —>route_in_batch file

With these library installed:

Package	Version	Latest
PyQt5	5.10.1	5.10.1
cycler	0.10.0	0.10.0
decorator	4.3.0	4.3.0
kiwisolver	1.0.1	1.0.1
matplotlib	2.2.2	2.2.2
memory-profiler	0.52.0	
networkx	2.1	2.1
numpy	1.14.2	➡ 1.14.3
pip	9.0.3	➡ 10.0.1
psutil	5.4.5	5.4.5
pyparsing	2.2.0	2.2.0
python-dateutil	2.7.2	➡ 2.7.3
pytz	2018.4	2018.4
setuptools	39.0.1	➡ 39.1.0
sip	4.19.8	4.19.8
six	1.11.0	1.11.0