1. Describe the machine you are running your tests on :

CPU: Intel 3740QM @2.7GHz

OS: Windows10 64bit

Memory: 16GB DDR3 @2133MHz

#### 2. Minimum Requirements (80%):

2.1 Display the effort put in for the optimized trips; for items without weight information, simply ignore that item & let the user know that the effort output is missing some information;

* 1. Provide menu/program option to allow user to set whether or not weight/effort is calculated for trip/order list
  2. Compare existing routes (from previous project parts) with new weight information to see if there are differences in effort vs. routes.

The running results are shown in the end of this documents.

2-opt (distance) is the 2-optimization algorithm to minimize total distance.

2-opt (effort) is the 2-optimization algorithm to minimize total effort.

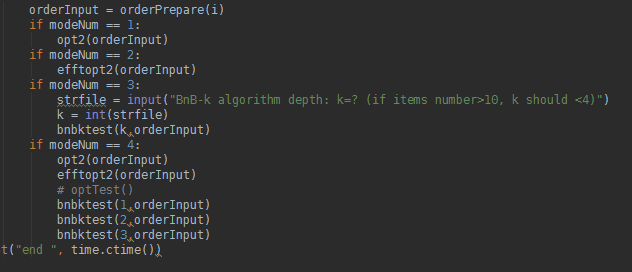
BnB-k is the branch and bond algorithm with limited k depth. Large k will give you better results, but the running time will increase exponentially.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Order#12 (21 items) | | | |
| algorithm | Distance | Effort | Effort (reverse) | Running Time |
| 2-opt (distance opt) | 39.0 | 374.57 | 347.476 | 6.83 |
| 2-opt (effort opt) | 39.0 | 293.45 | | 8.27 |
| BnB-1 | 41.0 | 501.44 | 324.12 | 0.10 |
| BnB-2 | 39.4 | 448.73 | 327.87 | 0.11 |
| BnB-3 | 35.0 | 355.25 | 350.44 | 1.01 |

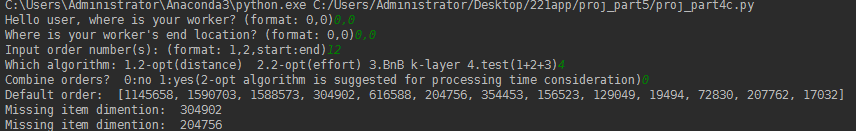
#### 3. Preferred Requirements (90%)

User should be able to select from menu option which algorithm they want to use (from previous project part):

Function: in main()



Running screen shot:



#### 4. Ideal Requirements (100%)

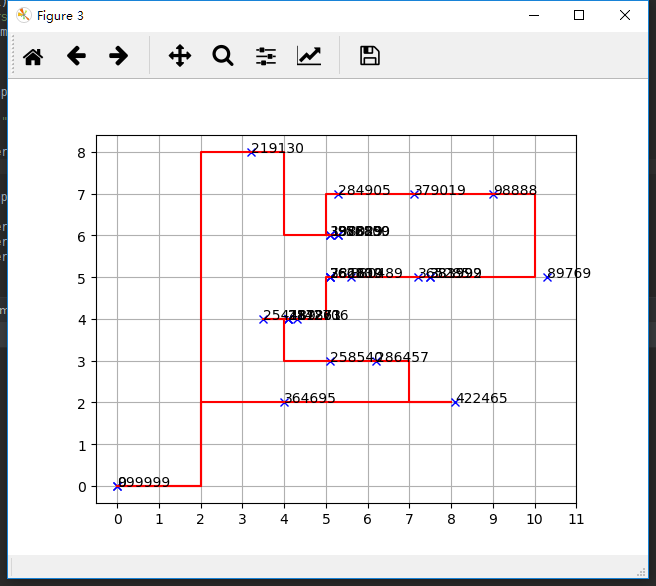
#### Batch processing of orders:

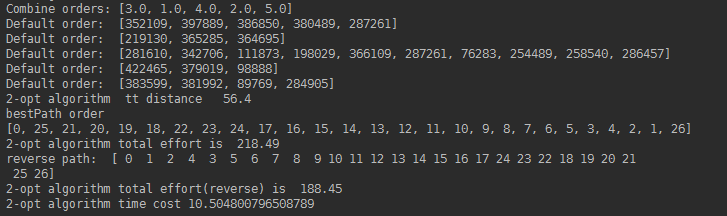
#### Function: (in main())

#### 

#### Running screen shot: batch order 1-5

#### 





#### 5. Bonus (+10%)

Using greedy algorithm:

1. sort the order by total weight;
2. choose the order with largest total weight (if exceed max capacity of cart, split the order by running a sub-greedy algorithm on each items);
3. filling the cart until no space or no fit order;
4. run optimization algorithm on batched order.

Select order 1-10,

Using 2-opt algorithm,

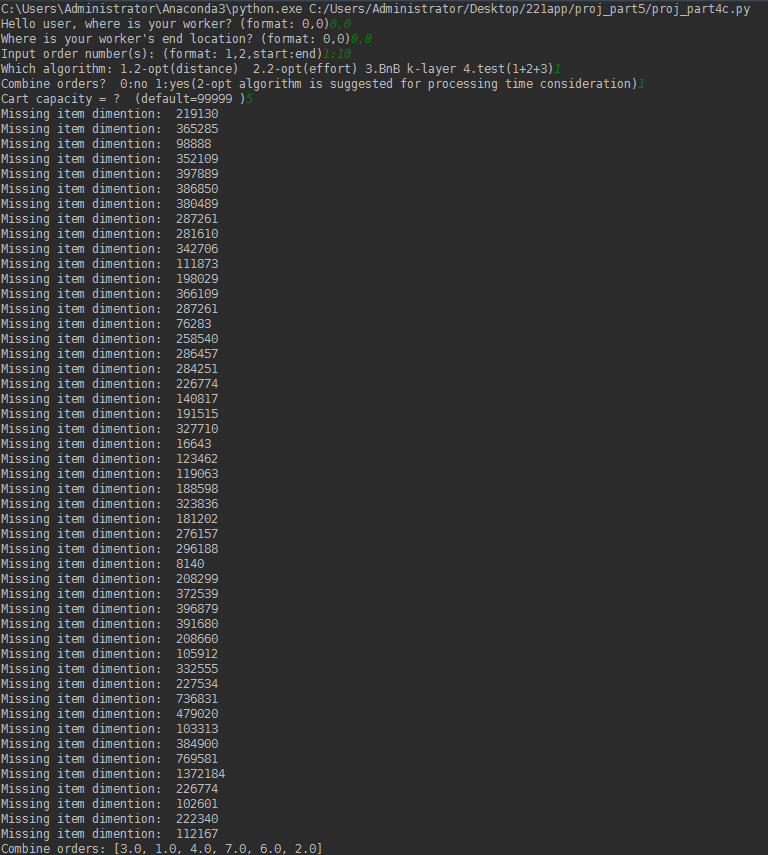
Set the maximum cart capacity to 5:

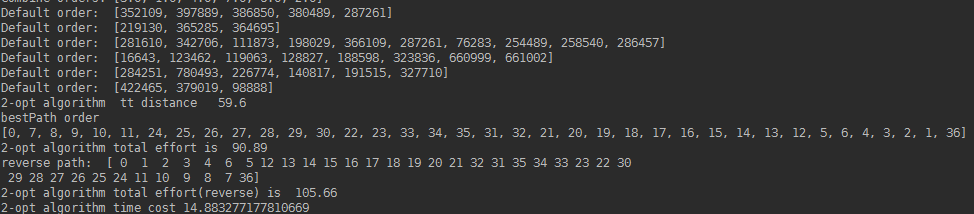
We can see that the program split the order into 2 order batch:

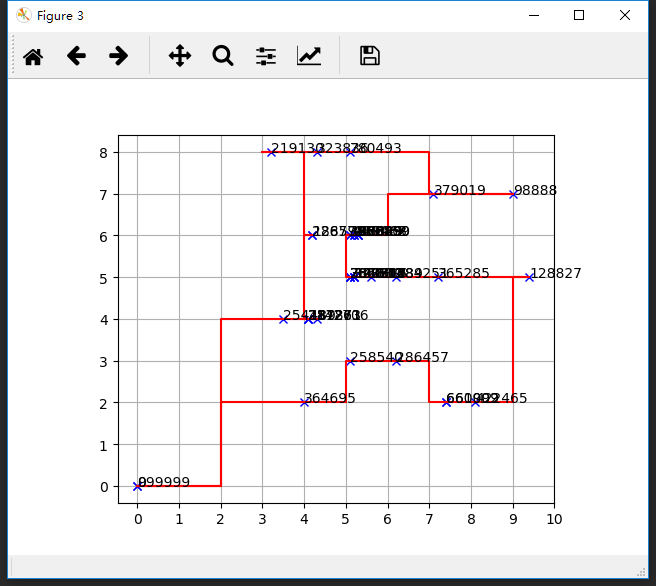
1: 

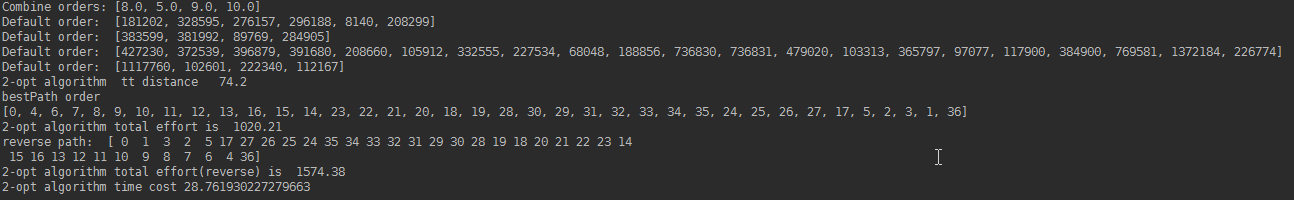
2: 

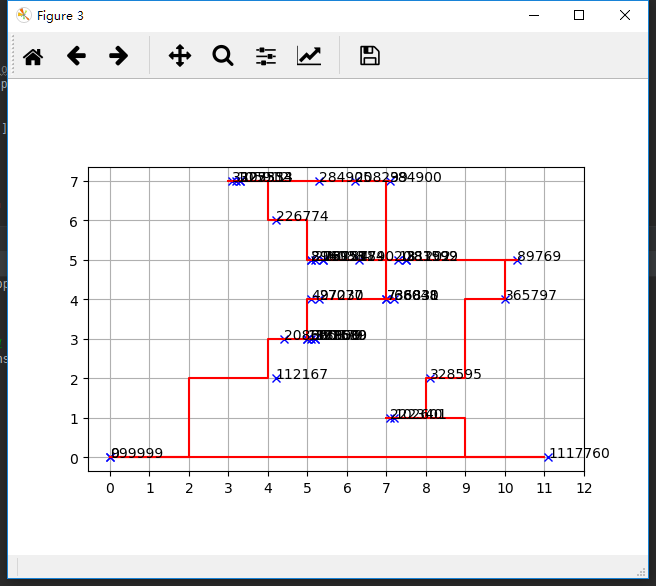
Running screen shot:











Hello user, where is your worker? (format: 0,0)0,0

Where is your worker's end location? (format: 0,0)0,0

Input order number(s): (format: 1,2,start:end)12

Which algorithm: 1.2-opt(distance) 2.2-opt(effort) 3.BnB k-layer 4.test(1+2+3)4

Combine orders? 0:no 1:yes(2-opt algorithm is suggested for processing time consideration)0

Default order: [1145658, 1590703, 1588573, 304902, 616588, 204756, 354453, 156523, 129049, 19494, 72830, 207762, 17032]

Missing item dimention: 304902

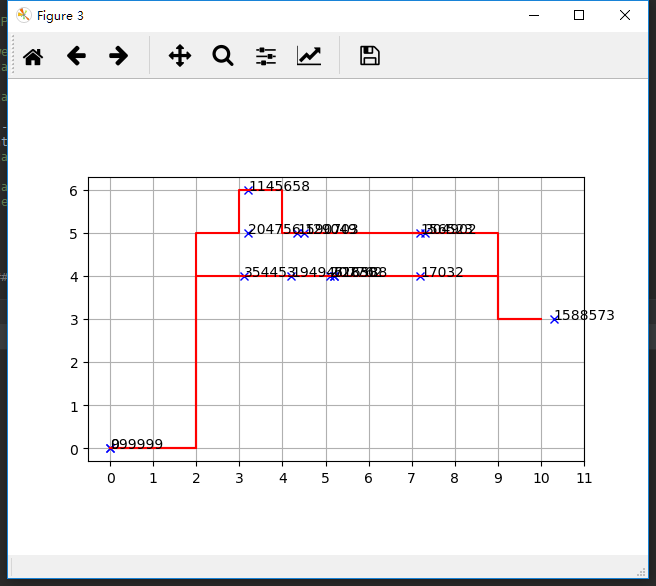
Missing item dimention: 204756

Missing item dimention: 354453

Missing item dimention: 156523

Missing item dimention: 129049

Missing item dimention: 19494



2-opt algorithm tt distance 39.0

bestPath order

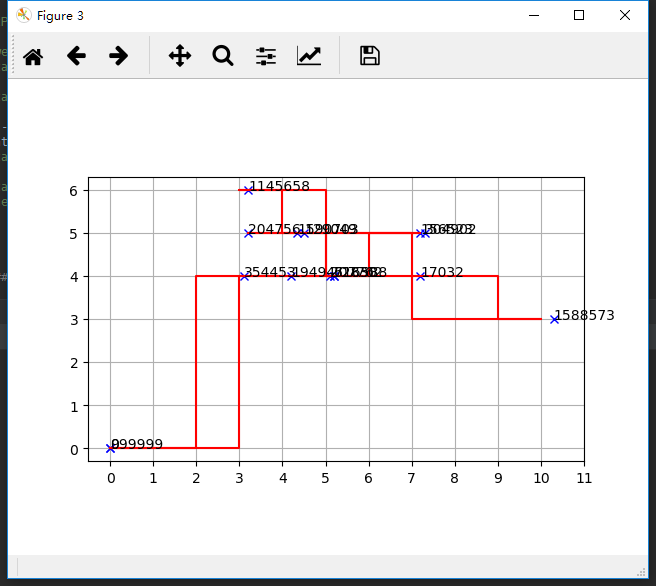
[0, 2, 3, 4, 5, 6, 7, 1, 12, 11, 10, 9, 13, 8, 14]

2-opt algorithm total effort is 374.57

reverse path: [ 0 8 13 9 10 11 12 1 7 6 5 4 3 2 14]

2-opt algorithm total effort(reverse) is 347.476

2-opt algorithm time cost 6.834900617599487



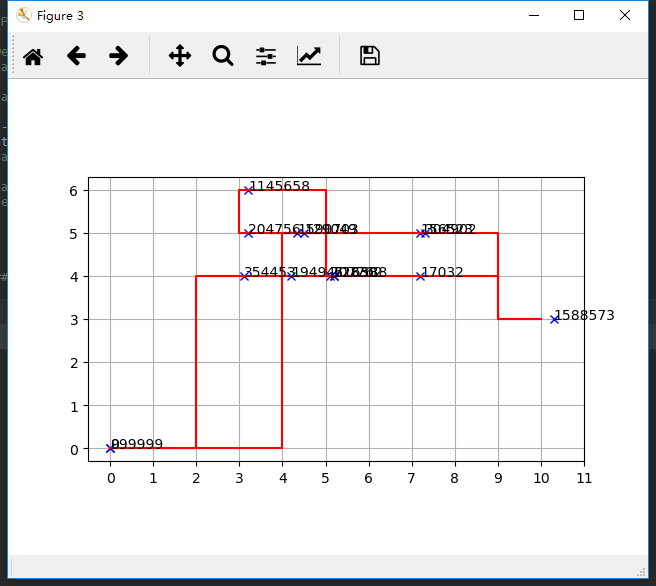
Effort 2-opt algorithm: total distance 49.0

Effort 2-opt algorithm: total effort 293.45

bestPath order

[0, 2, 3, 12, 11, 8, 10, 1, 7, 13, 9, 4, 5, 6, 14]

2-opt algorithm time cost 8.27332353591919



bnb 1 -layer optPath

[0, 2, 3, 4, 5, 6, 10, 9, 8, 13, 7, 1, 12, 11, 14]

bnb 1 -layer lower bound is [41.0]

bnb 1 -layer total distance is 41.0

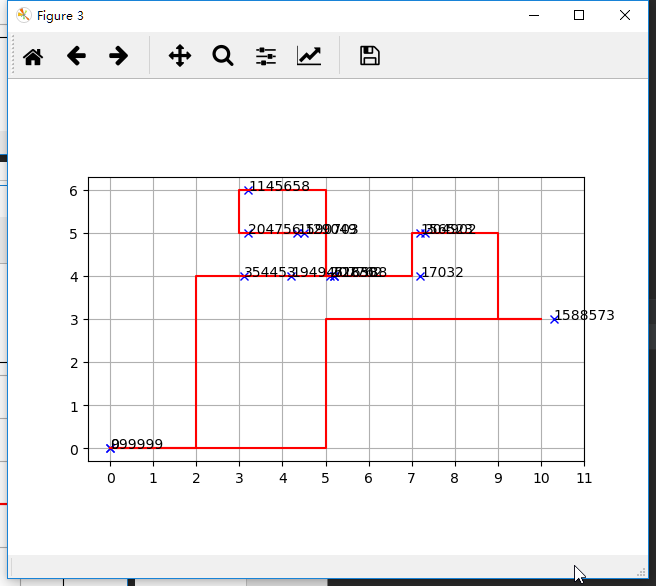
bnb 1 -layer total effort is 501.44

reverse path: [0, 11, 12, 1, 7, 13, 8, 9, 10, 6, 5, 4, 3, 2, 14]

bnb 1 -layer total effort(reverse) is 324.12

bnb 1 -layer total effort is 501.44

bnb 1 -layer time cost 0.10232257843017578



bnb 2 -layer optPath

[0, 2, 3, 4, 5, 6, 10, 9, 8, 13, 7, 11, 12, 1, 14]

bnb 2 -layer lower bound is [39.400000000000006]

bnb 2 -layer total distance is 39.4

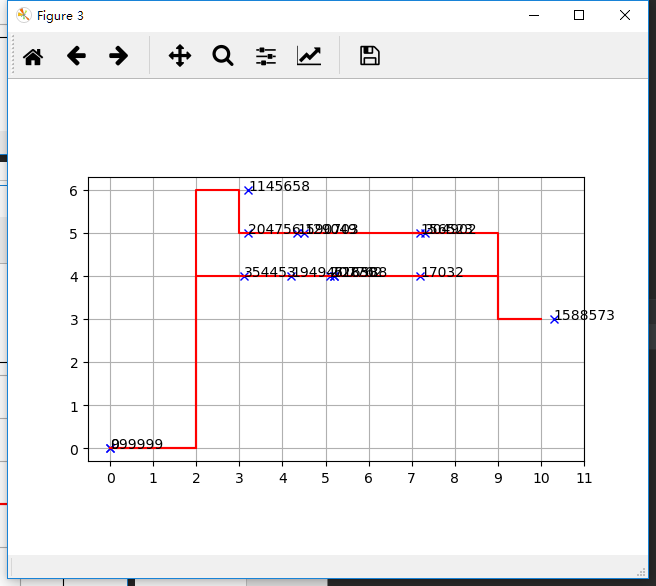
bnb 2 -layer total effort is 448.73

reverse path: [0, 1, 12, 11, 7, 13, 8, 9, 10, 6, 5, 4, 3, 2, 14]

bnb 2 -layer total effort(reverse) is 327.87

bnb 2 -layer total effort is 448.73

bnb 2 -layer time cost 0.11338663101196289



bnb 3 -layer optPath

[0, 2, 3, 4, 5, 6, 7, 1, 12, 11, 10, 9, 8, 13, 14]

bnb 3 -layer lower bound is [35.0]

bnb 3 -layer total distance is 35.0

bnb 3 -layer total effort is 355.25

reverse path: [0, 13, 8, 9, 10, 11, 12, 1, 7, 6, 5, 4, 3, 2, 14]

bnb 3 -layer total effort(reverse) is 350.44

bnb 3 -layer total effort is 355.25

bnb 3 -layer time cost 1.013364315032959

end Mon May 21 05:45:50 2018

Process finished with exit code 0