Experimental report for COM1005 Assignment 2: A Production System for Bagging

Ruiqing Xu

May 11, 2020

1 Proposed improved system

The main 4 classes of the improved systems are: ImprovedStartBagging.java, ImprovedGetNextItem.java, ImprovedPutInCurrentBag.java, ImprovedChooseCorrectBag.java.

- ImprovedStartBagging.java: This class used to start bagging and make sure the first bag space is 100%.
- ImprovedGetNextItem.java: This class used to get items in the trolley and "write down" the space and name of each items.
- ImprovedPutInCurrentBag.java: This class used to make sure the current item space is less than the bag remaining space, and if so put the item into the right bag.
- ImprovedChooseCorrectBag.java: This class used to find and "write down" the remaining spaces of all the bags.

1.1 Description of improved production rules

The best part of the Improved algorithm is to change the original START NEW BAG step. For example, in trolley, one cherry (80% space), one apple (30% space), one pear (60% space), and one mango (20% space) were brought out in order. The Basic Algorithm will take out three bags to pack the three items: bag1 contains cherry, bag2 contains apple and pear, and bag3 contains mango. The Improved algorithm requires only two bags: bag1 contains cherry and mango, bag2 contains apple and pear. This is because it will compare the current item space with all the rest of the previous bags space, so that the mango will be placed in the first bag. Thus it can be seen the Improved Algorithm is more efficiency than Basic Algorithm.

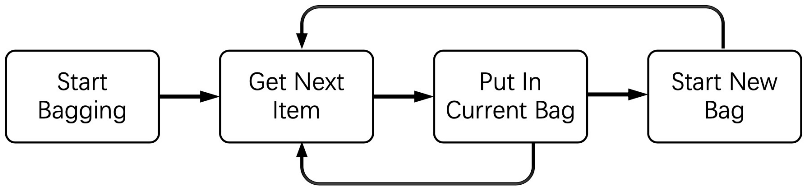

Figure 1: Production rule

1.2 Pseudo-code for BasicBagger

Algorithm 1 Basic algorithm

• Start Bagging (Get the 1st Bag i.e., B_1 $W_j = \sum_{i \in B_j} w_i$;
if $W_j < C$ **then**
  • Put Item i in Current Bag B_j
else
  • Start a New Bag B_{j+1};
  • Put Item i in Current Bag B_{j+1};
end if
i = i + 1;

1.3 Pseudo-code for ImprovedBagger

2 Experimental Results

2.1 Evaluation Metrics

$$\mathcal{L} = \frac{\sum_i w_i}{j * C} \tag{1}$$

2.2 Description of tests

In the system, the code was tested in the following:

- This study used 20 different items which have different space and colour in order to test the comparison of efficiency.

Algorithm 2 Improved algorithm

• Start Bagging (Get the 1st Bag i.e., B_1);
while true **do**
  • Get Next Item i with corresponding weight w_i;
  $W_j = \sum_{i \in B_j} w_i$;
  if $W_j < C$ **then**
    • Put Item i in Current Bag B_j;
  else
    booleam ifStartBagFLag = true;
    for Bag k, k = 1 to j **do**
      if $W_k + w_i <= C$ **then**
        ifStartBagFLag = false;
        • Put Item i in previous Bag B_k;
        Break;
      end if
    end for
    if ifStartBagFLag == true **then**
      • Start a New Bag B_{j+1};
      • Put Item i in Current Bag B_{j+1};
    end if
  end if
  i = i + 1;
end while

- This study also divided items into 3 group which depends on its space, and compare the two algorithm's bag utility.

2.3 Results from running tests

- The improved system works better than basic system, the improved algorithm got 11 bags, and the basic algorithm got 14 bags.
- According to each item group the performance all improved.

2.3.1 Case study

- The trolley contains 20 different items which have different space and colour, and these items will be bagged in order: 1. Cherry 2.Filbert 3. Mushroom 4. Corn 5. Peach 6. Mango 7. Milk 8. Pitaya 9. Apple 10. Cabbage 11. Kiwi 12. Grape 13. Blueberry 14. Tomato 15. Lemon 16. Eggplant 17. Carrot 18. Cucumber 19. Kumquat 20. Potato.
- In terms of improved algorithm, there are 11 bags and 7 of them are fully occupied. In terms of basic algorithm, there are 14 bags and 1 of them are fully occupied, there are still a lot of bags that remaining space.

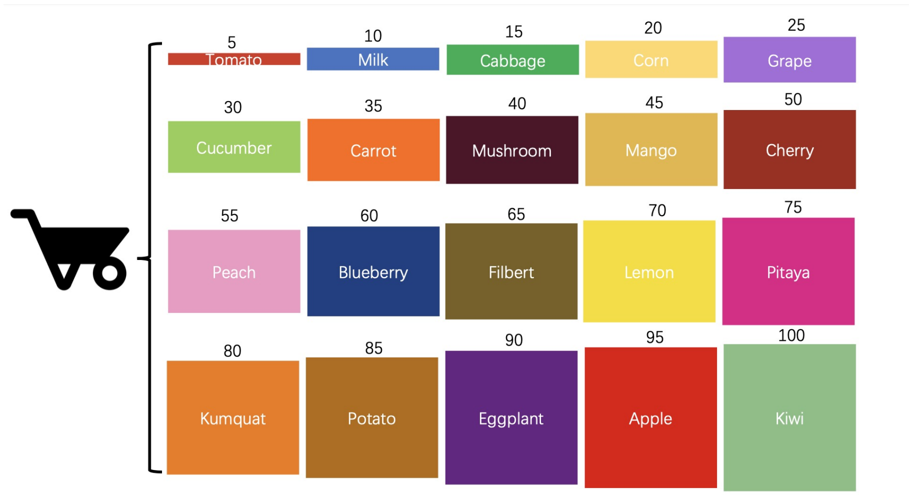

Figure 2: Trolley

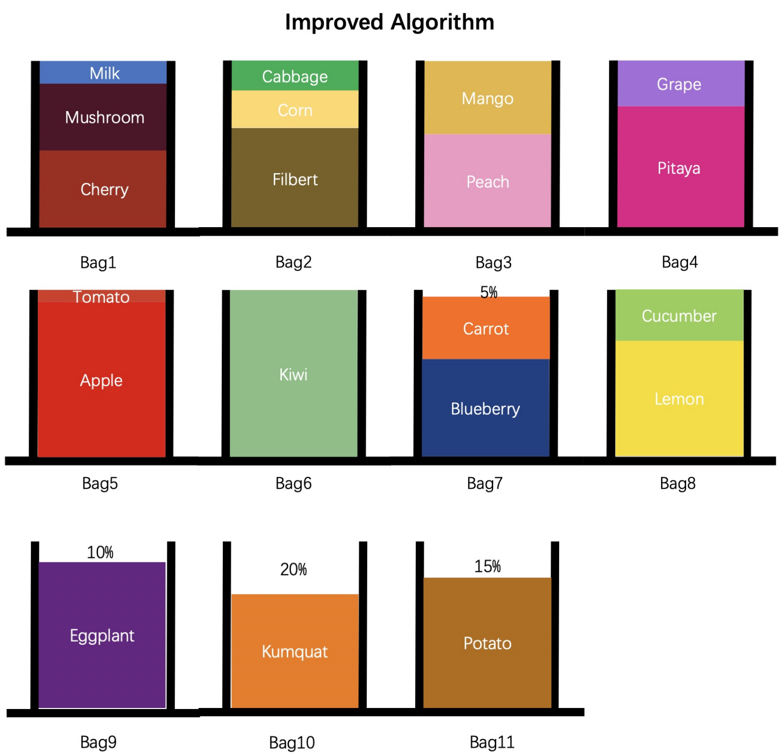

Figure 3: Improved Algorithm

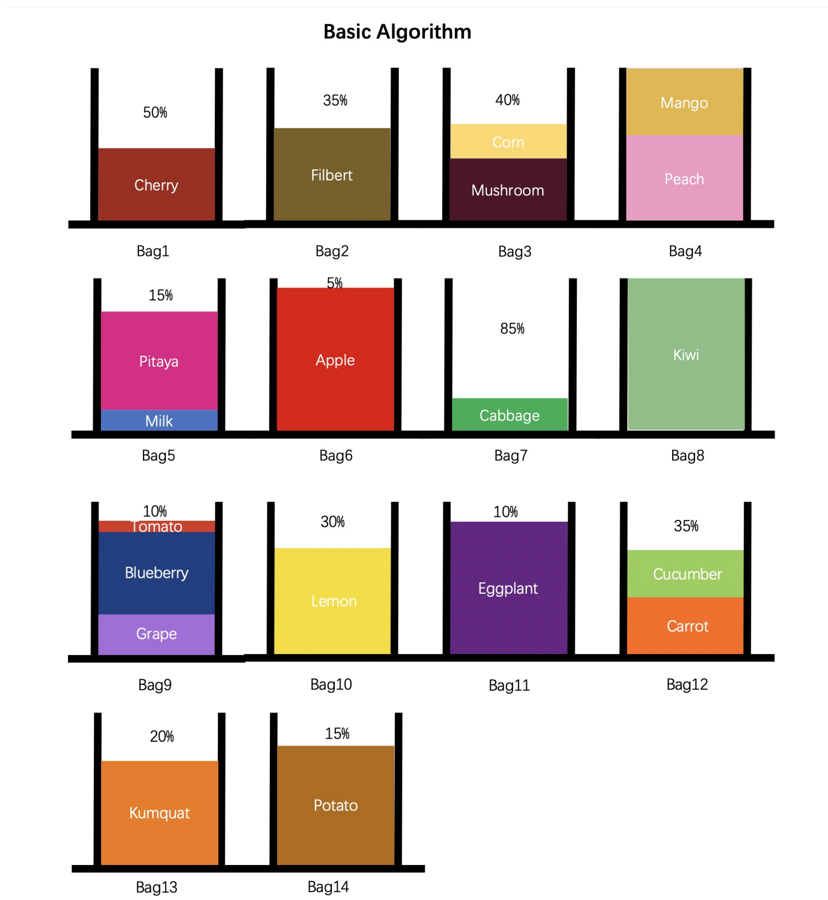

Figure 4: Basic Algorithm

2.3.2 Comparison between basic and improved algorithms

- This study divides items into three types, the large item group(all the items$\geq$50%) , medium item group(25%$\leq$all the items$<$50%), and small item group(all the items$\leq$25%). Each group contains 20 items.
- There is the table which shows the result of each group's bag utility.

	Bag Utility		
	Large Item Group	**Medium Item Group**	**Small Item Group**
Basic Algorithm	41.30%	80.00%	75.00%
Improved Algorithm	45.80%	90.00%	87.50%
Performance Improvement	4.50%	10.00%	12.50%

Table 1: Comparison of the bag utility among different item groups.

3 Conclusions

In conclusion, some code were modified based on the basic algorithm, and space of each bag was reasonably utilized. This study used different 20 items to tests the efficiently and also compared the bag utility of different item groups, and it all shows that improved algorithm is better.