Homework 8

CS 325

Daniel Kim

Problem 1: *(20 pts)* In the bin packing problem, items of different weights (or sizes) must be packed into a finite number of bins each with the capacity C in a way that minimizes the number of bins used. The decision version of the bin packing problem (deciding if objects will fit into <= k bins) is NP-complete. There is no known polynomial time algorithm to solve the optimization version of the bin packing problem. In this homework you will be examining three greedy approximation algorithms to solve the bin packing problem.

- First-Fit: Put each item as you come to it into the first (earliest opened) bin into which it fits. If there is no available bin then open a new bin.
- First-Fit-Decreasing: First sort the items in decreasing order by size, then use First-Fit on the resulting list.
- Best Fit: Place the items in the order in which they arrive. Place the next item into the bin which will leave the least room left over after the item is placed in the bin. If it does not fit in any bin, start a new bin.

a) Give pseudo code and the running time for each of the approximation algorithms.

b) Implement the algorithms in Python, C++ or C. Your program named binpack should read in a text file named bin.txt with multiple test cases as explained below and output to the terminal the number of bins each algorithm calculated for each test case. Submit a README file and your program to TEACH.

Example bin.txt: The first line is the number of test cases, followed by the capacity of bins for that test case, the number of items and then the weight of each item. You can assume that the weight of an item does not exceed the capacity of a bin for that problem.

```
3
10
6
5 10 2 5 4 4
10
20
4 4 4 4 4 4 4 4 4 4 6 6 6 6 6 6 6 6 6 6
10
4
3 8 2 7
```

Sample output:
Test Case 1 First Fit: 4, First Fit Decreasing: 3, Best Fit: 4
Test Case 2 First Fit: 15, First Fit Decreasing: 10, Best Fit: 15
Test Case 2 First Fit: 3, First Fit Decreasing: 2, Best Fit: 2

c) Randomly generate at least 20 bin packing instances. Summarize the results for each algorithm. Which algorithm performs better? How often? *Note: Submit a description of how the inputs were generated not the code used to produce the random inputs.*

a) First-Fit

```
For all items i = 1, 2,3,.., n do
    For all bins j=1,2,...do
        if item i fits bin j then
        Put item i in bin j .
        Break and move to next item
    If item does not fit in any current bins,
        create a new bin
        Put item in new bin
```

Time complexity **for First-Fit = O(n^2)**

First-Fit-Decreasing

Sort items i=1,...,n in decreasing order

Apply **First-Fit** algorithm (above)

Time complexity for First-Fit-Decreasing = **O(n^2)**

Best-Fit

for all items i = 1, 2, . . . , n do

  for All bins j = 1, 2, . . . do

    if item i fits in bin j then

      Add item and calculate remaining capacity

  Pack item i in the bin j with minimum calculated remaining capacity if item i were packed

  If no bin exists

      Create a new bin

      Add item to new bin

Time complexity for **Best-Fit** = **O(n^2)**

b) Code Submitted to Teach! Thank you!

c) Random Numbers Generated
   1. Cases (iterations) = 300
   2. Bin Capacity Weight (integer) from 10 to 50
   3. Item Weight (integer) from 2 to Bin Capacity
   4. Number of Items (integer) from 10 to 30

   <u>Results</u>

   - All methods (First-fit, First-fit-decreasing, and Best-fit) had equal performance: 67.33% of 300 iterations
   - First-fit-decreasing had the best outcome 32.33% of 300 iterations
   - First-fit resulted in the best outcome 0.33% of 300 iterations

Problem 2: *(10 pts)* An exact solution to the bin packing optimization problem can be found using 0-1 integer programming (IP) see the format on the Wikipedia page.

Write an integer program for each of the following instances of bin packing and solve with the software of your choice. Submit a copy of the code and interpret the results.

a) Six items S = { 4, 4, 4, 6, 6, 6} and bin capacity of 10

b) Five items S = { 20, 10, 15, 10, 5} and bin capacity of 20

Note: The version of LINDO that you have access to on the OSU server has a limit of 50 integer variables. Therefore, LINDO will only be able to solve problems with at most 6 items.

a) MIN Y1+Y2+Y3+Y4+Y5+Y6

ST

X11 + X12 + X13 + X14 + X15 + X16 = 1

X21 + X22 + X23 + X24 + X25 + X26 = 1

X31 + X32 + X33 + X34 + X35 + X36 = 1

X41 + X42 + X43 + X44 + X45 + X46 = 1

X51 + X52 + X53 + X54 + X55 + X56 = 1

X61 + X62 + X63 + X64 + X65 + X66 = 1

4X11 + 4X21 + 4X31 + 6X41 + 6X51 + 6X61 - 10Y1 <= 0

4X12 + 4X22 + 4X32 + 6X42 + 6X52 + 6X62 - 10Y2 <= 0

4X13 + 4X23 + 4X33 + 6X43 + 6X53 + 6X63 - 10Y3 <= 0

4X14 + 4X24 + 4X34 + 6X44 + 6X54 + 6X64 - 10Y4 <= 0

4X15 + 4X25 + 4X35 + 6X45 + 6X55 + 6X65 - 10Y5 <= 0

4X16 + 4X26 + 4X36 + 6X46 + 6X56 + 6X66 - 10Y6 <= 0

END

INT X11

INT X12

INT X13

INT X14

INT X15

INT X16

INT Y2

INT X21

INT X22

INT X23

INT X24

INT X25

INT X26

INT Y3

INT X31

INT X32

INT X33

INT X34

INT X35

INT X36

INT Y4

INT X41

INT X42

INT X43

INT X44

INT X45

INT X46

INT X51

INT X52

INT X53

INT X54

INT X55

INT X56

INT Y6

INT X61

INT X62

INT X63

INT X64

INT X65

INT X66

<untitled>

```
MIN Y1+Y2+Y3+Y4+Y5+Y6
ST
X11 + X12 + X13 + X14 + X15 + X16 = 1
X21 + X22 + X23 + X24 + X25 + X26 = 1
X31 + X32 + X33 + X34 + X35 + X36 = 1
X41 + X42 + X43 + X44 + X45 + X46 = 1
X51 + X52 + X53 + X54 + X55 + X56 = 1
X61 + X62 + X63 + X64 + X65 + X66 = 1

4X11 + 4X21 + 4X31 + 6X41 + 6X51 + 6X61 - 10Y1 <= 0
4X12 + 4X22 + 4X32 + 6X42 + 6X52 + 6X62 - 10Y2 <= 0
4X13 + 4X23 + 4X33 + 6X43 + 6X53 + 6X63 - 10Y3 <= 0
4X14 + 4X24 + 4X34 + 6X44 + 6X54 + 6X64 - 10Y4 <= 0
4X15 + 4X25 + 4X35 + 6X45 + 6X55 + 6X65 - 10Y5 <= 0
4X16 + 4X26 + 4X36 + 6X46 + 6X56 + 6X66 - 10Y6 <= 0
END

INT X11
INT X12
INT X13
INT X14
INT X15
INT X16
INT Y2
INT X21
INT X22
INT X23
INT X24
INT X25
INT X26
INT Y3
INT X31
INT X32
INT X33
INT X34
INT X35
INT X36
INT Y4
INT X41
INT X42
INT X43
INT X44
INT X45
INT X46
INT X51
INT X52
INT X53
INT X54
INT X55
INT X56
INT Y6
INT X61
INT X62
INT X63
INT X64
INT X65
INT X66
```

Reports Window

```
       8)         0.000000          0.100000
       9)         0.000000          0.000000
      10)         0.000000          0.000000
      11)         0.000000          0.000000
      12)         0.000000          0.100000
      13)         0.000000          0.000000

 NO. ITERATIONS=      21
 BRANCHES=    0 DETERM.=  1.000E    0

 LP OPTIMUM FOUND AT STEP     40
 OBJECTIVE VALUE =    3.00000000


 NEW INTEGER SOLUTION OF    3.00000000     AT BRANCH      0 PIVOT      42
 BOUND ON OPTIMUM:  3.000000
 ENUMERATION COMPLETE. BRANCHES=     0 PIVOTS=      42

 LAST INTEGER SOLUTION IS THE BEST FOUND
 RE-INSTALLING BEST SOLUTION...

        OBJECTIVE FUNCTION VALUE

        1)      3.000000

  VARIABLE        VALUE          REDUCED COST
       X11         0.000000          0.400000
       X12         1.000000          0.000000
       X13         0.000000          0.000000
       X14         0.000000          0.000000
       X15         0.000000          0.400000
       X16         0.000000          0.000000
        Y2         1.000000          1.000000
       X21         0.000000          0.400000
       X22         0.000000          0.000000
       X23         0.000000          0.000000
       X24         0.000000          0.000000
       X25         1.000000          0.400000
       X26         0.000000          0.000000
        Y3         0.000000          1.000000
       X31         1.000000          0.400000
       X32         0.000000          0.000000
       X33         0.000000          0.000000
       X34         0.000000          0.000000
       X35         0.000000          0.400000
       X36         0.000000          0.000000
        Y4         0.000000          1.000000
       X41         0.000000          0.600000
       X42         0.000000          0.000000
       X43         0.000000          0.000000
       X44         0.000000          0.000000
       X45         1.000000          0.600000
       X46         0.000000          0.000000
       X51         0.000000          0.600000
       X52         1.000000          0.000000
       X53         0.000000          0.000000
       X54         0.000000          0.000000
       X55         0.000000          0.600000
       X56         0.000000          0.000000
        Y6         0.000000          1.000000
       X61         0.000000          0.600000
       X62         0.000000          0.000000
       X63         0.000000          0.000000
       X64         0.000000          0.000000
       X65         1.000000          0.600000
       X66         0.000000          0.000000
        Y1         0.400000          0.000000
        Y5         1.600000          0.000000


       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         0.000000          0.000000
        3)         0.000000          0.000000
        4)         0.000000          0.000000
        5)         0.000000          0.000000
        6)         0.000000          0.000000
        7)         0.000000          0.000000
        8)         0.000000          0.100000
        9)         0.000000          0.000000
       10)         0.000000          0.000000
       11)         0.000000          0.000000
       12)         0.000000          0.100000
       13)         0.000000          0.000000

 NO. ITERATIONS=      42
 BRANCHES=    0 DETERM.=  1.000E    0
```

b)

Reports Window

```
       X41          1.000000          0.000000
       X42          0.000000          0.000000
       X43          0.000000          0.000000
       X44          0.000000          0.000000
       X45          0.000000          0.000000
        Y5          1.000000          1.000000
       X51          0.000000          0.000000
       X52          0.000000          0.000000
       X53          0.000000          0.000000
       X54          0.000000          0.000000
       X55          1.000000          0.000000


       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)          0.000000          0.000000
        3)          0.000000          0.000000
        4)          0.000000          0.000000
        5)          0.000000          0.000000
        6)          0.000000          0.000000
        7)          0.000000          0.000000
        8)          0.000000          0.000000
        9)          0.000000          0.000000
       10)          0.000000          0.000000
       11)          0.000000          0.000000

 NO. ITERATIONS=      48
 BRANCHES=    0 DETERM.=  1.000E    0

 LP OPTIMUM FOUND AT STEP     19
 OBJECTIVE VALUE =   3.00000000


 NEW INTEGER SOLUTION OF    3.00000000     AT BRANCH      0 PIVOT      19
 RE-INSTALLING BEST SOLUTION...

        OBJECTIVE FUNCTION VALUE

        1)      3.000000

  VARIABLE        VALUE          REDUCED COST
        Y1          1.000000          1.000000
        Y2          1.000000          1.000000
        Y3          1.000000          1.000000
        Y4          0.000000          1.000000
        Y5          0.000000          1.000000
       X11          0.000000          0.000000
       X12          1.000000          0.000000
       X13          0.000000          0.000000
       X14          0.000000          0.000000
       X15          0.000000          0.000000
       X21          1.000000          0.000000
       X22          0.000000          0.000000
       X23          0.000000          0.000000
       X24          0.000000          0.000000
       X25          0.000000          0.000000
       X31          0.000000          0.000000
       X32          0.000000          0.000000
       X33          1.000000          0.000000
       X34          0.000000          0.000000
       X35          0.000000          0.000000
       X41          1.000000          0.000000
       X42          0.000000          0.000000
       X43          0.000000          0.000000
       X44          0.000000          0.000000
       X45          0.000000          0.000000
       X51          0.000000          0.000000
       X52          0.000000          0.000000
       X53          1.000000          0.000000
       X54          0.000000          0.000000
       X55          0.000000          0.000000


       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)          0.000000          0.000000
        3)          0.000000          0.000000
        4)          0.000000          0.000000
        5)          0.000000          0.000000
        6)          0.000000          0.000000
        7)          0.000000          0.000000
        8)          0.000000          0.000000
        9)          0.000000          0.000000
       10)          0.000000          0.000000
       11)          0.000000          0.000000

 NO. ITERATIONS=      19
 BRANCHES=    0 DETERM.=  1.000E    0
```

```
MIN Y1+Y2+Y3+Y4+Y5
ST
X11+    X12+    X13+    X14+    X15=    1
X21+    X22+    X23+    X24+    X25=    1
X31+    X32+    X33+    X34+    X35=    1
X41+    X42+    X43+    X44+    X45=    1
X51+    X52+    X53+    X54+    X55=    1


20X11 + 10X21 + 15X31 + 10X41 + 5X51 -20Y1 <=0
20X12 + 10X22 + 15X32 + 10X42 + 5X52 -20Y2 <=0
20X13 + 10X23 + 15X33 + 10X43 + 5X53 -20Y3 <=0
20X14 + 10X24 + 15X34 + 10X44 + 5X54 -20Y4 <=0
20X15 + 10X25 + 15X35 + 10X45 + 5X55 -20Y5 <=0

END

INT Y1
INT Y2
INT Y3
INT Y4
INT Y5


INT X11
INT X12
INT X13
INT X14
INT X15


INT X21
INT X22
INT X23
INT X24
INT X25


INT X31
INT X32
INT X33
INT X34
INT X35


INT X41
INT X42
INT X43
INT X44
INT X45

INT X51
INT X52
INT X53
INT X54
INT X55
```

MIN Y1+Y2+Y3+Y4+Y5

ST

X11+X12+X13+X14+X15=   1

X21+X22+X23+X24+X25=   1

X31+X32+X33+X34+X35=   1

X41+ X42+ X43+ X44+X45=   1

X51+ X52+ X53+ X54+X55=   1

20X11 + 10X21 + 15X31 + 10X41 + 5X51 -20Y1 <=0

20X12 + 10X22 + 15X32 + 10X42 + 5X52 -20Y2 <=0

20X13 + 10X23 + 15X33 + 10X43 + 5X53 -20Y3 <=0

20X14 + 10X24 + 15X34 + 10X44 + 5X54 -20Y4 <=0

20X15 + 10X25 + 15X35 + 10X45 + 5X55 -20Y5 <=0

END

INT Y1

INT Y2

INT Y3

INT Y4

INT Y5

INT X11

INT X12

INT X13

INT X14

INT X15

INT X21

INT X22

INT X23

INT X24

INT X25

INT X31

INT X32

INT X33

INT X34

INT X35

INT X41

INT X42

INT X43

INT X44

INT X45

INT X51

INT X52

INT X53

INT X54

INT X55