Rey’s Confetti Shop

By: Rey Dialarmsaw

I run a confetti shop. I have three confetti machine options; X, Y, and Z. X takes up 5 sq ft, Y takes up 7 sq ft, and Z takes up 6, and I have 150 total sq ft. X costs 1.3K, Y costs 1.7K, and Z costs 1.6K, and I have 30K total.

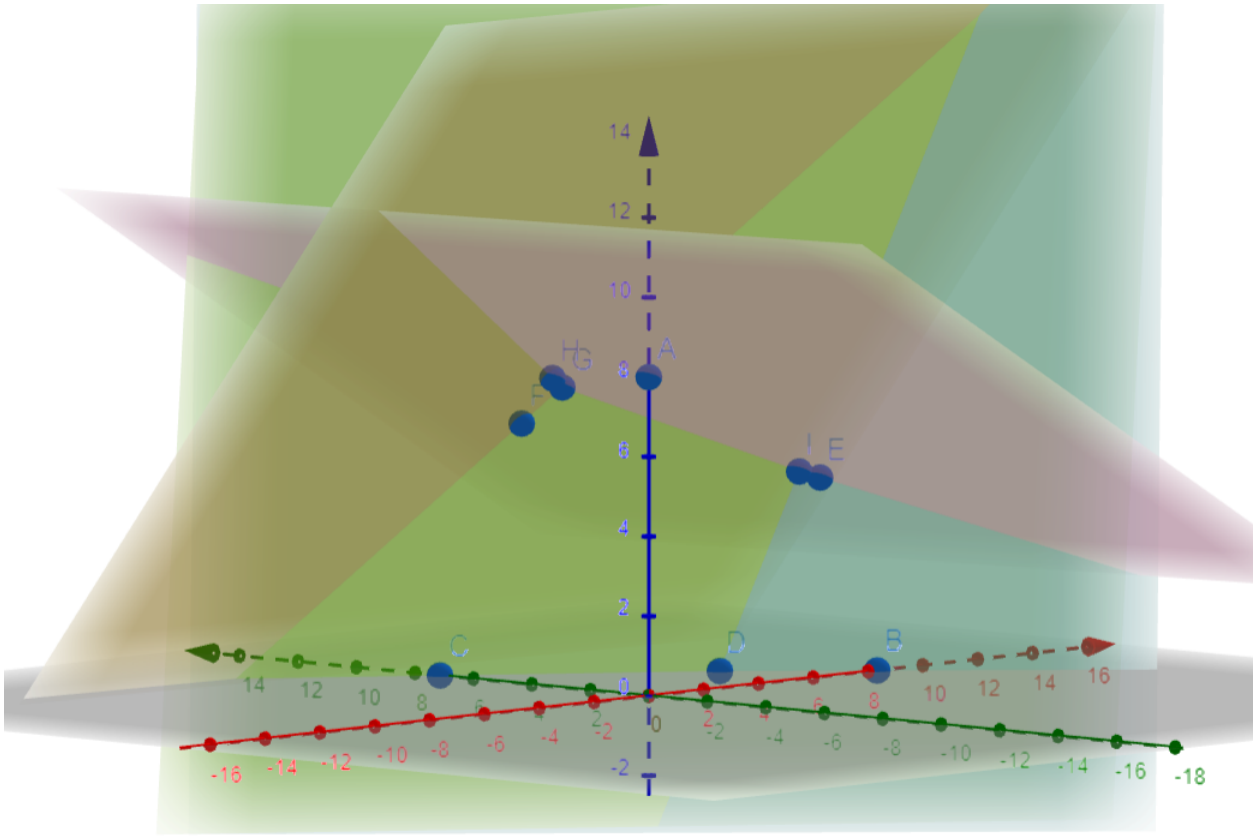
X needs 12 materials to run, Y needs 13, and Z needs 11, I have 200 total per day. X takes 120 watts, Y takes 140 watts, and Z takes 110 watts, and I want to use less than 1K watts per day.

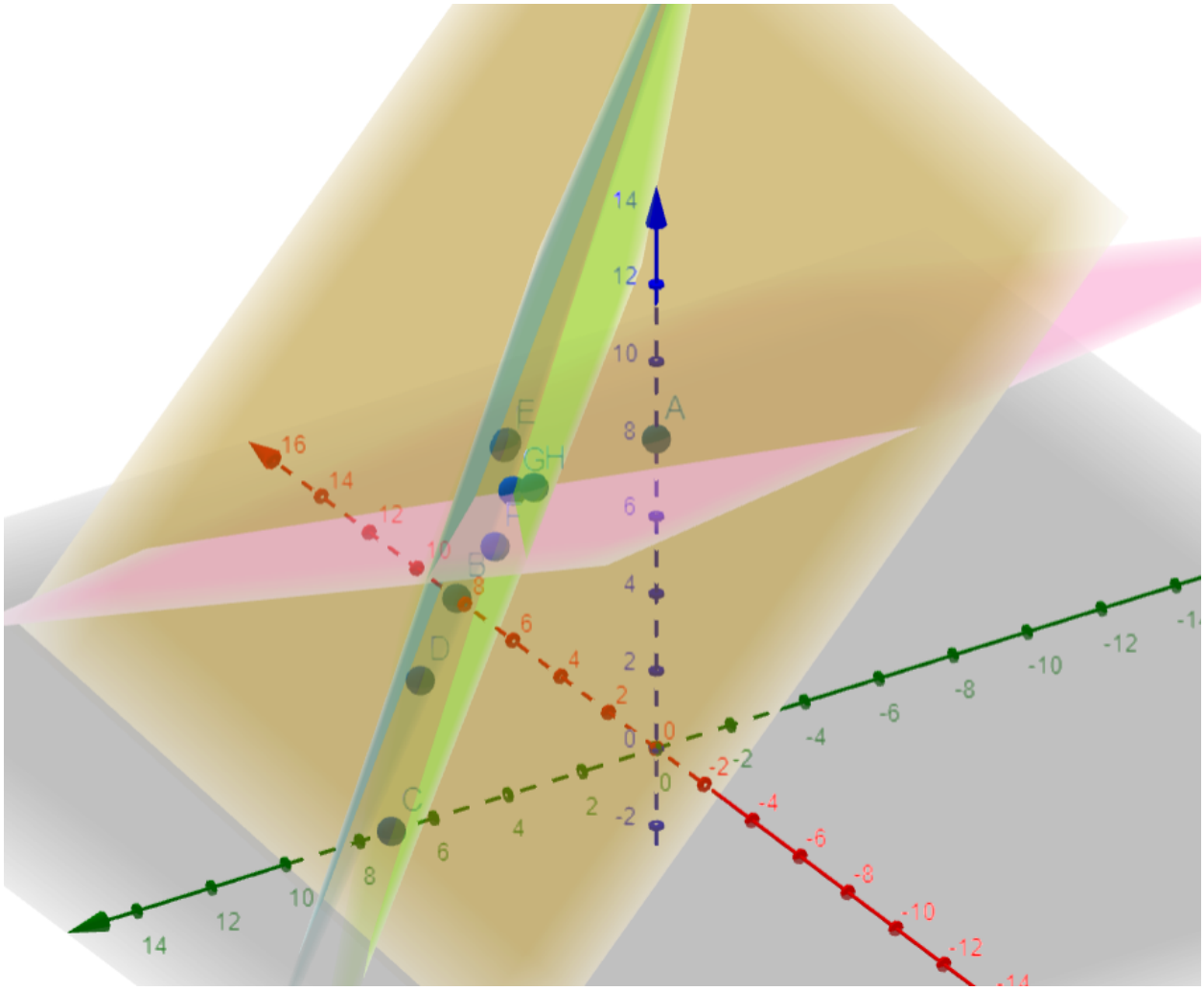
X makes $12, Y makes $15, and Z makes $25. What is the optimal configuration of machines to maximize output? X equals the number of X machines, Y equals the number of Y machines, and Z equals the number of Z machines.

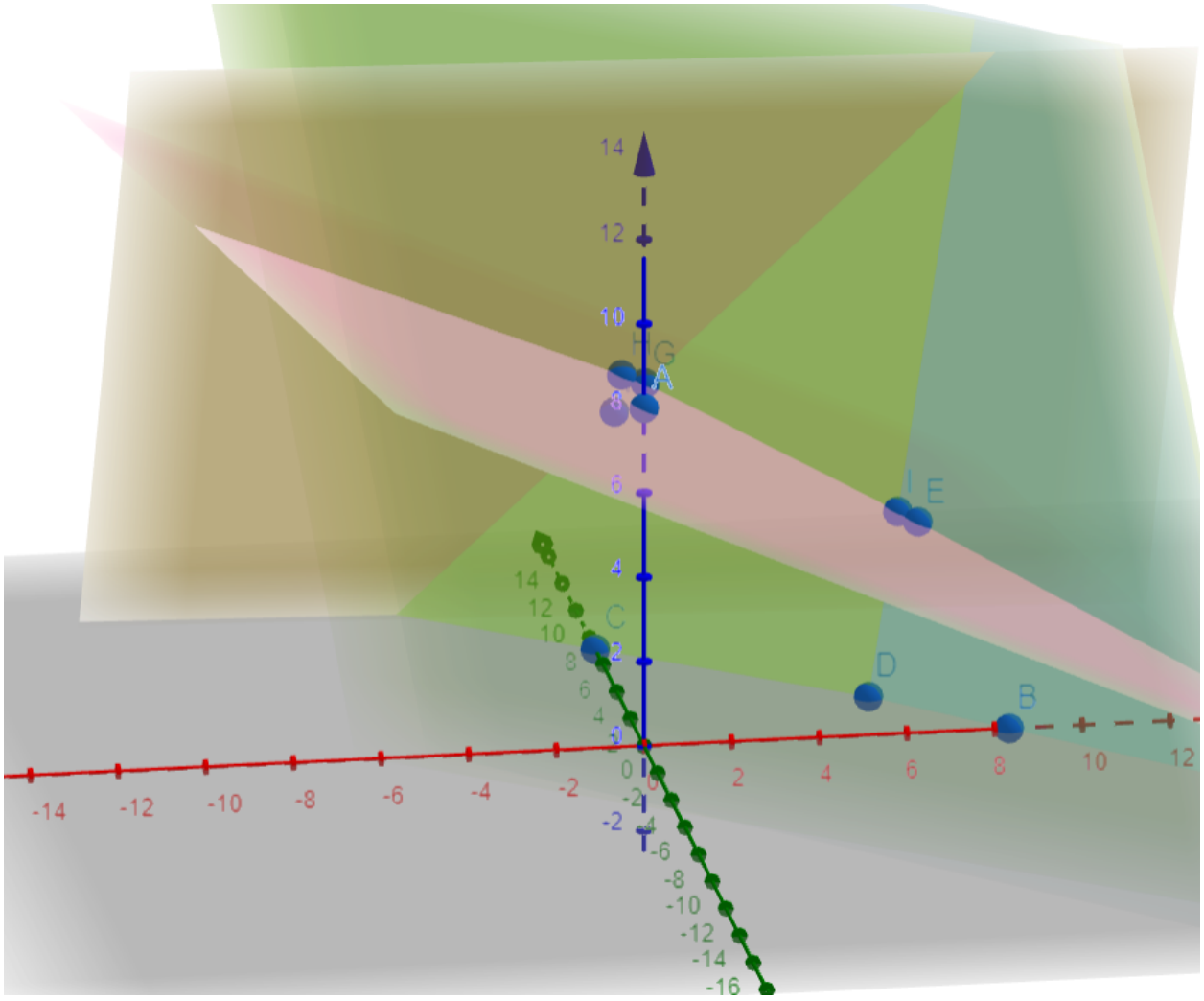
Equations:
5x+7y+6z<=150
1.3Kx+1.7Ky+1.6Kz<=30K
12x+13y+11z<=200
120x+140y+110z<=1K

Objective equation:
12x+15y+25z=Ans

X machines	Y machines	Z machines	Output:
0	0	8	200
8.33	0	0	99.96
0	7.14	0	107.1
5.56	2.78	0	108.42
6.25	0	5	200
0	4.35	6.52	228.25
0	3.3	7.74	243
0.57	3.5	7.44	245.34
5.84	0.33	5.16	204.03
X machines	Y machines	Z machines	Total Output:
0	0	8	200
8	0	0	96
0	7	0	105
5	2	0	90
6	0	5	197
0	4	6	210
0	3	7	220
0	3	7	220


14
12
10
8
6
4
2
0
-2
A
G
H
F
E
I
C
D
B
14 12 10 8 6 4 2 -2 -4 -6 -8 -10 -12 -14 -16 -18
-16 -14 -12 -10 -8 -6 -4 -2 0 2 4 6 8 10 12 14 16


A
B
C
D
E
F
GH
14
12
10
8
6
4
2
0
-2
16
14
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
14
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12


14
12
10
H
G
A
6
I
E
4
14
12
10
C
8
2
D
6
B
4
0
0
2
4
6
8
10
12
-14
-12
-10
-8
-6
-4
-2
-2
-4
-6
-8
-10
-12
-14
-16

```
import numpy as np # imports the numpy file

print("Plane 1:") # Gets the plane coordinates
x1 = int(input("x >>> "))
y1 = int(input("y >>> "))
z1 = int(input("z >>> "))
c1 = int(input("c >>> "))

print("Plane 2:")
x2 = int(input("x >>> "))
y2 = int(input("y >>> "))
z2 = int(input("z >>> "))
c2 = int(input("c >>> "))

print("Plane 3:")
x3 = int(input("x >>> "))
y3 = int(input("y >>> "))
z3 = int(input("z >>> "))
c3 = int(input("c >>> "))

a = np.array([[x1, y1, z1], [x2, y2, z2], [x3, y3, z3]]) # Makes a matrix out of the coordinates

const = np.array([[c1],[c2],[c3]]) # Makes a matrix out of the constants

ainv = np.linalg.inv(a) # finds the invert of the matrix

result = np.matmul(ainv, const) # creates a matrix multiplying ainv by const

res_list = result.tolist() # makex the x y z values into a list

x = res_list[0]

y = res_list[1]

z = res_list[2]
print("The point that the meet: X =",x," Y =",y," Z =",z) # prints values
```

12x+2y+25z+200 (pink)
120x+120y+50z=1000 (cyan)
1.2x+15y+13z=150 (brown)
120x+140y+110z=1000

120x+120y+50z=1000
-120x+140y+110=1000
_____
-20y-60z=0

-20y=60z

-y=-3x

120x+1500y+1300z=15000
-120x+140y+110z=1000
_____
1360y+1190z

1360(3z)+1190z=14000

4080z+1190z=14000

5270z=14000

z=14000/5270

When I made my choice, I had an objective function: 12x+15y+25z=k, and I want to get k as big as possible. In the spreadsheet, I put every point into a spreadsheet and used the equation, and the best one is 0, 3, 7. The one on the top has decimals, but realistically you can't have half of a machine, so I rounded down.