**Least Cost Method**

def leastCostIBFS(fact, ware,weights):

  n = 0

  w = 0

  ibfs = 0

  temp\_weights = weights.copy()

  arr = np.array([[0 for i in range(len(ware))] for i in range(len(fact))])

  while sum(fact) > 0:

    n = np.argwhere(weights == np.min(weights))[0][0]

    w = np.argwhere(weights == np.min(weights))[0][1]

    if fact[n] > ware[w]:

      arr[n][w] = ware[w]

      fact[n] -= ware[w]

      ware[w] = 0

    elif fact[n] < ware[w]:

      arr[n][w] = fact[n]

      ware[w] -= fact[n]

      fact[n] = 0

    else:

      arr[n][w] = fact[n]

      fact[n] = 0

      ware[w] = 0

    weights[n][w] = sys.maxsize

  for i in range(len(arr)):

    for j in range(len(arr[0])):

      ibfs += arr[i][j] \* temp\_weights[i][j]

  return arr,ibfs

fact = [7,9,18]

ware = [5,8,7,14]

weights = np.array([[19,30,50,10],

                    [70,30,40,60],

                    [40,8,70,20]])

leastCostIBFS(fact, ware,weights)

OUTPUT:

(array([[0, 0, 0, 7],

[2, 0, 7, 0],

[3, 8, 0, 7]]), 814)

fact = [70,

        30,

        50]

ware = [65,42,43]

weights = np.array([[5,7,8],

                    [4,4,6],

                    [6,7,7]])

leastCostIBFS(fact, ware,weights)

OUTPUT:

(array([[35, 35, 0],

[30, 0, 0],

[ 0, 7, 43]]), 890)