**Using Backtracking Algorithm For Solving Mutiple Container Loading 2D Mininize Cost**

1. Strategy.

By using backtracking we can ‘draw’ all possible pattern of loading each item into container. Suppose that the number of container is always enough, so the feasible solution is always exist. Then we look at all solutions which is found, we choose the best one

2. Constraint of loading

* All item must be loaded exactly one time
* Not overlap

3. Algorithm

N= number of items

K= number of containers

X=[]

def Try(p): # find the possible position to load item p

for c in range(K):

candy=find\_candy(c)

if candy:

for x,y in candy:

for r in range(2): # rotation

if check\_freespace():

X.append([c,x,y,t])

Update\_freespace()

If p=n-1:

Solution()

Else:

#heuristic\_should\_be\_here

Try(p+1)

X.pop() #return value of X

a. Dealing with overlap, and free\_space.

We introduce a matrix Am.n corresponding to size of container. We use this matrix to check its possibility of putting item i . Each element A[i,j] represent to a point with condinate i,j at the container. If A[i,j]=1, we cant put item on this point anymore.

At initial state the matrix A have form :

* A[i,j]=0, for element at center
* A[i,j]=0.75 for element at corner
* A[i,j]=0.5 for element at edge

Eg: A=

+ Secondly, we introduce a matrix Bm’.n’ corresponding to size of each item. Like matrix A but have a few defferences

* B[i,j]=1, for element at center
* B[i,j]=0.25 for element at corner
* B[i,j]=0.5 for element at edge

Eg: B=

+ When we want to put an item into container, we just search A[i,j] !=1, and sum of A[x:x+m’+1,y:x+n’+1] and B not make any element of matrix A >1 then (x,y) is a feasible solution of load item i in to container j.

4. Branching and bound

+ Instead of select order container randomly, we sort container following the value of usage ( cost/area\_of\_container)



+ We use f\_min ( cost of using container) which is the best temporary solution to cut every branches have cost greater or equal f\_min. At the initial state the value of f\_min should be total cost of all container

