Problem with "split-table"

Donglai Wei

2010.4.21

I)Corrected Pseudocode for split-table

0) **Settings:**
K nonempty dishes
Pick the Rth Restaurant with m tables.
Pick the Tth($T \leq m$) tables with n($n \geq 2$) customers.

1) **2-means++**

(a) First 2 points:

Randomly pick a customer C1 from table T to form a new table (m+1)
Randomly sample a dish(can be new) for the new table (m+1) according to the cost
Randomly sample another customer C2 from table t according to the cost
Randomly sample a dish(can be new) for the new table (m+2) according to the cost

(b) Initialization:

For ww=randperm(customers from table T except C1,C2)
Randomly sample the table assignment zz$\in \{m+1, m+2\}$ for customer ww according to the cost
Randomly sample a dish(can be new) for table zz according to the cost
End

(c) Iteration(2-means):

While no more changes of table config and dish config can increase P:
b=rand([0,1])
Switch (ceil(b*4)):
case 1: Randomly pick a customer from table (m+1), assign it to table (m+2) if the change increase P
case 2: Randomly pick a customer from table (m+2), assign it to table (m+1) if the change increase P
case 3: pick table (m+1), assign it the dish which increase P mostly
case 4: pick table (m+2), assign it the dish which increase P mostly
End

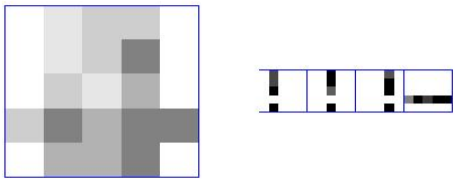

Figure 1: Table config for Restaurant 1 that gives lowest Free Energy

II)Controlled Experiment

a)Settings

1) 40 Restaurants, each of which is a mixture of bars.
2) Erase table and dish config for the 1st Restaurant while keep the ground truth config for the rest.

b)Goal

Recover the table and dish config for the 1st Restaurant, conditioning on the rest.

c.1)Initialization–bottom up

1) Initialization: In the 1st Restaurant, every customer has his own table and every table has its own dish.
2) Methods:

1. Local Search table(LS-t)+Local Search dish(LS-d):
   The one in the middle is the Restaurant, around are table config results for different runs.
   Clearly seen from below, though the local search works as well as fast, it can be hard to figure out the "row bar"

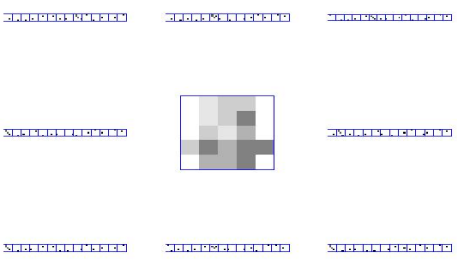

Figure 2: LS-t once

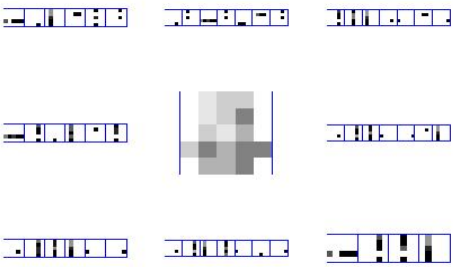

Figure 3: LS-t+LS-d+LS-t

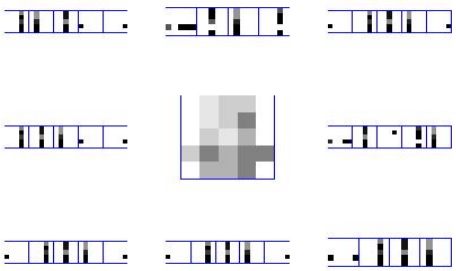

Figure 4: LS-t+LS-d+LS-t+LS-d+LS-t

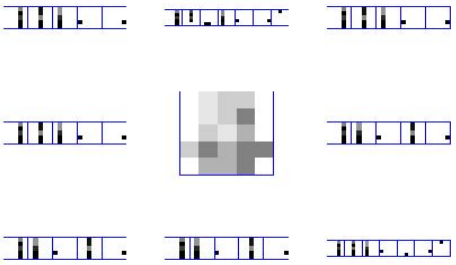

Figure 5: LS-d+LS-t

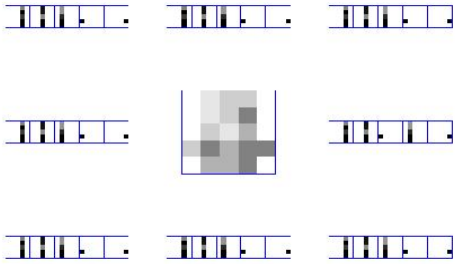

Figure 6: LS-d+LS-t+LS-d+LS-t

2. Merge table(M-t)+Merge dish(M-d):

3. Comparisons of Random Combinations

c.1)Initialization–bottom up

1) Initialization: In the 1st Restaurant, all customers have the same table which has a new dish.
2) Methods:

1. Local Search Move: As expected, the restaurant refuses to change.

2. Split Move:There is an "annealing" parameterα(tuning weights) during sampling. So long α is not too small or too big, the split moves give similar results.

3. Comparisons of Random Combinations

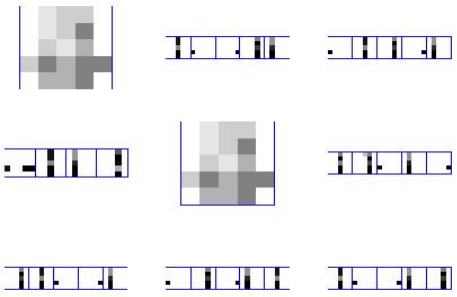

Figure 7: Merge table once

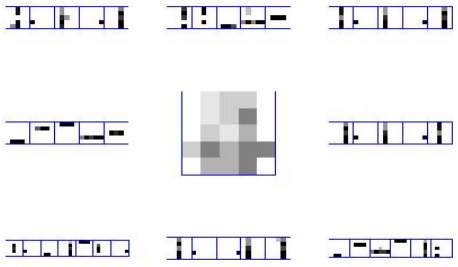

Figure 8: Merge dish+Merge table

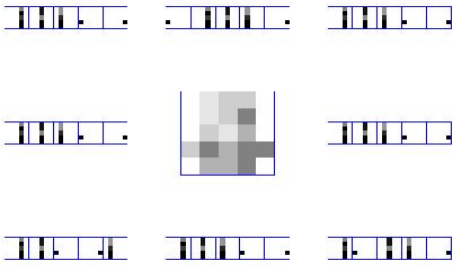

Figure 9: Random Combo of Local moves

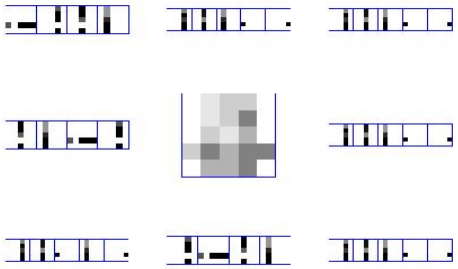

Figure 10: Random Combo of Merge moves

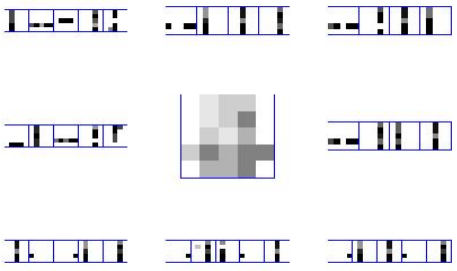

Figure 11: Random Combo of Merge,Split moves

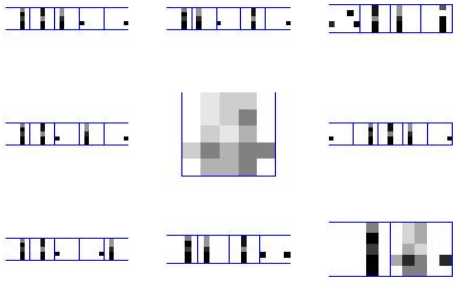

Figure 12: Random Combo of Local,Merge,Split moves

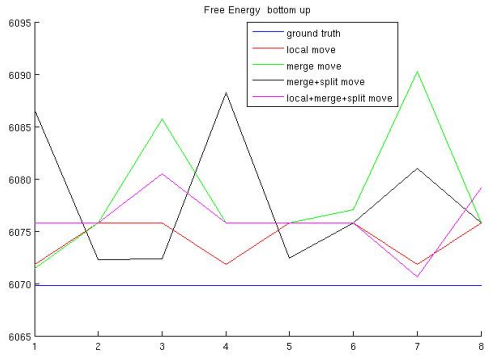

Figure 13: Free Energy corresponds to above config

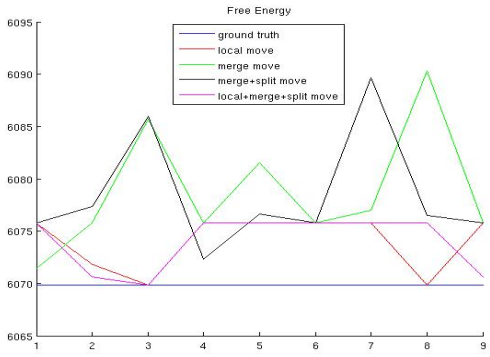

Figure 14: Another run

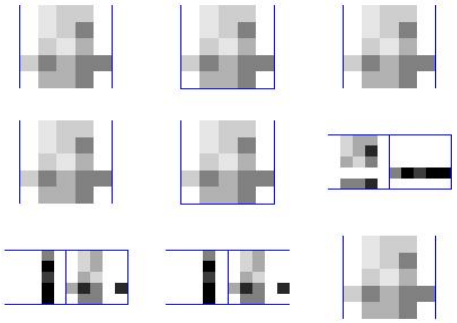

Figure 15: Split table once

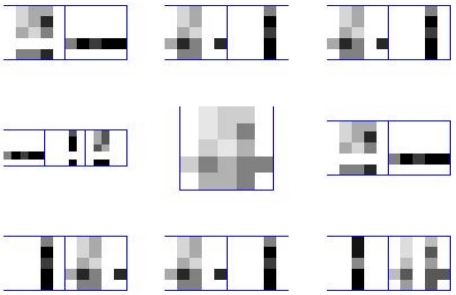

Figure 16: Split table twice

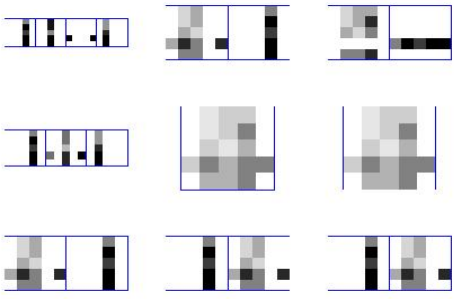

Figure 17: Split table three times

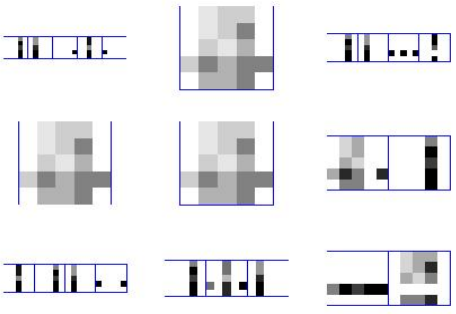

Figure 18: Split table four times

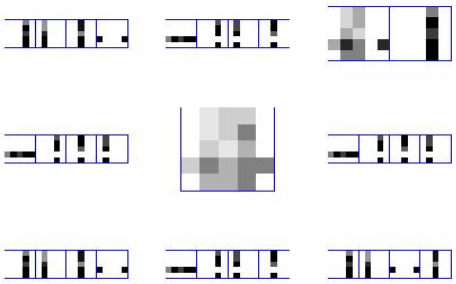

Figure 19: Random Combo of Merge,Split moves

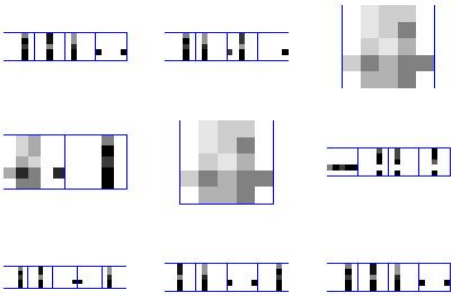

Figure 20: Random Combo of Local,Merge,Split moves

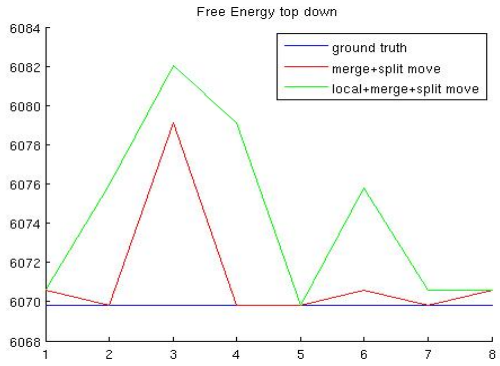

Figure 21: Free Energy corresponds to above config

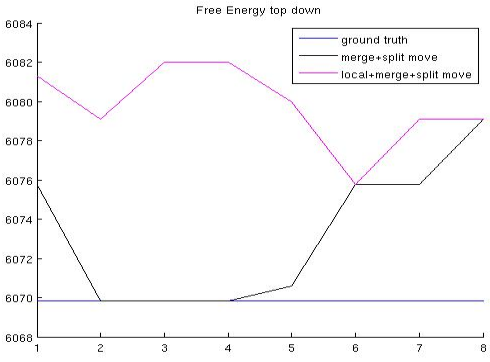

Figure 22: Another run