Featured Prediction Competition

Santa's Workshop Tour 2019

In the notebook we can build a model, and pretend that it will optimize...

$25,000
Prize Money

Kaggle · 1,042 teams · a month to go (a month to go until merger deadline)

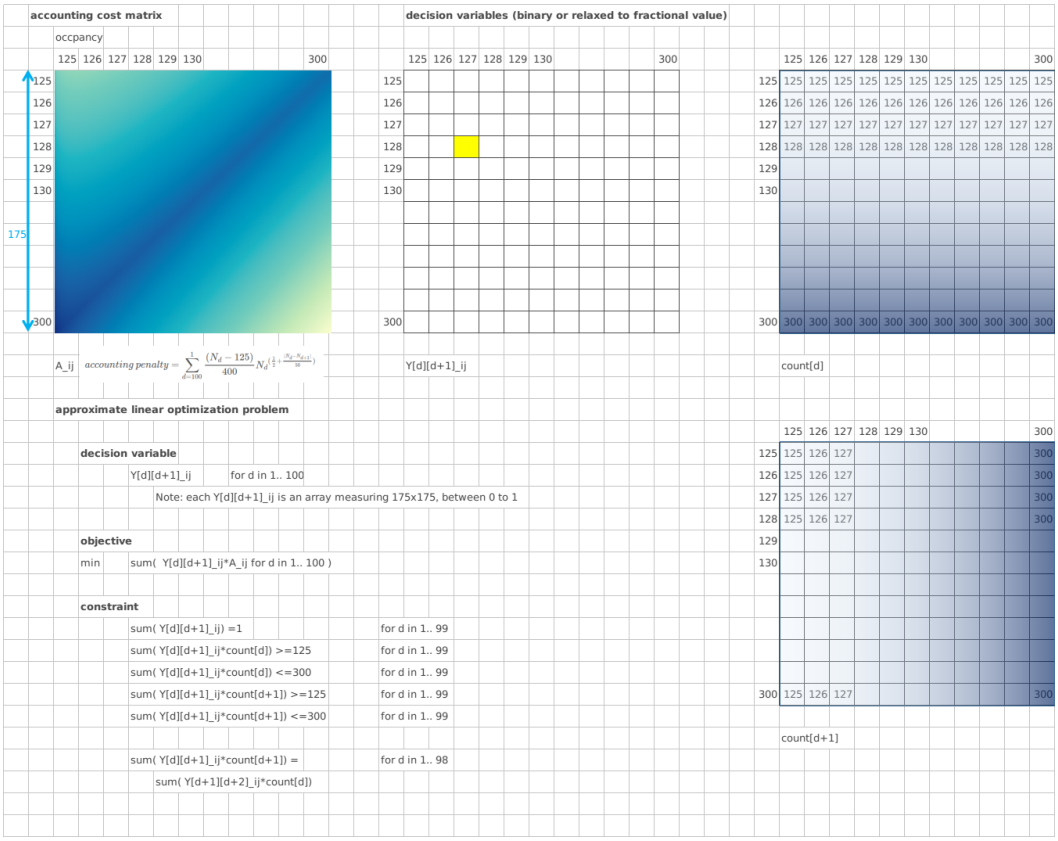
Heng CherKeng
97th place

lower bound 67309.48: global solution with relaxed integer programming

22

posted in Santa's Workshop Tour 2019 3 days ago

how to linearize the accounting penalty :



Options

Comments (22)

Sort by
Oldest

Click here to comment...

related: https://www.kaggle.com/c/santa-workshop-tour-2019/discussion/120103#latest-687237

Heng CherKeng Topic Author • (97th in this Competition) • 3 days ago • Options • Reply

3

seems to work ... but i need a good solver for large scale santa problem

Dummy example

```
if 1: dummy data for code development
    NUM_DAY = 5
    NUM_FAMILY = 10
    FAMILY_SIZE = np.random.choice(4,NUM_FAMILY)+1

    NUM_FAMILY_MEMBER = FAMILY_SIZE.sum()
    MIN_OCCUPANCY,MAX_OCCUPANCY=2,8
    NUM_OCCUPANCY = MAX_OCCUPANCY-MIN_OCCUPANCY+1
    ACCOUNTING = np.random.uniform(0,10,(NUM_OCCUPANCY, NUM_OCCUPANCY))
    PREFERENCE = np.random.uniform(0,10,(NUM_FAMILY, NUM_DAY))

    FIRST_DAY_OCCUPANCY=4
    LAST_DAY_OCCUPANCY =MIN_OCCUPANCY #not used
```

precomputed cost

using pyomo for modeling:

```
model = ConcreteModel()
def make_model():

    # variables
    model.x = Var(range(NUM_FAMILY),range(NUM_DAY), domain=Binary, initialize=0) #PercentFraction Binary
    model.y = Var(range(NUM_DAY),range(NUM_OCCUPANCY),range(NUM_OCCUPANCY), domain=Binary, initialize=0) #PercentFraction Binary

    # objective
    preference = [ PREFERENCE[i, d] * model.x[i,d] for i in range(NUM_FAMILY)for d in range(NUM_DAY) ]
    preference = sum(preference)
    accounting = [ACCOUNTING[u,v] * model.y[d,u,v] for u in range(NUM_OCCUPANCY) for v in range(NUM_OCCUPANCY) for d in range(NUM_DAY) ]
    accounting = sum(accounting)
    model.objective = Objective(expr=preference+accounting , sense=minimize)
```

decision variables

x : to assign family

y : to assign 'target occpancy'

constraint:

```
# constraints
model.constraint = ConstraintList()

occupancy = {}
for d in range(NUM_DAY):
    occupancy[d] = sum([model.x[i,d]* FAMILY_SIZE[i] for i in range(NUM_FAMILY)])
occupancy[d+1] = occupancy[d]

for i in range(NUM_FAMILY):
    model.constraint.add(sum([model.x[i, d] for d in range(NUM_DAY)]) == 1)

for d in range(NUM_DAY):
    model.constraint.add(occupancy[d] >= MIN_OCCUPANCY)
    model.constraint.add(occupancy[d] <= MAX_OCCUPANCY)

# ---
count = np.arange(MIN_OCCUPANCY,MAX_OCCUPANCY+1)

for d in range(NUM_DAY):
    y_sum_u = sum( model.y[d,u,v]*count[u] for v in range(NUM_OCCUPANCY) for u in range(NUM_OCCUPANCY) )
    y_sum_v = sum( model.y[d,u,v]*count[v] for v in range(NUM_OCCUPANCY) for u in range(NUM_OCCUPANCY) )
    # model.constraint.add(y_sum_u - occupancy[d]<=2) #relax equality
    # model.constraint.add(occupancy[d] - y_sum_u<=2)
    model.constraint.add(y_sum_u == occupancy[d])
    model.constraint.add(y_sum_v == occupancy[d+1])

    y_sum = sum( model.y[d,u,v] for v in range(NUM_OCCUPANCY) for u in range(NUM_OCCUPANCY) )
    model.constraint.add(y_sum == 1)


for d in range(NUM_DAY-1):
    for t in range(NUM_OCCUPANCY):
        y_sum_u = sum( model.y[d  ,u,t] for u in range(NUM_OCCUPANCY) )
        y_sum_v = sum( model.y[d+1,t,v] for v in range(NUM_OCCUPANCY) )
        model.constraint.add(y_sum_u == y_sum_v)
#
```

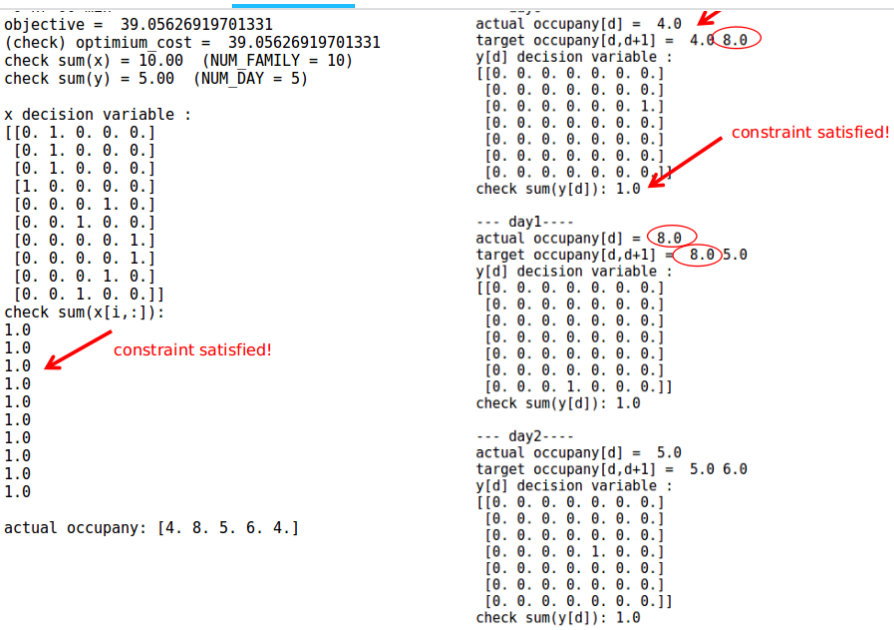
real occpancy

only one day assigment for each family

each real occpancy is limited

constaint 'real occpancy' equal to 'target occupancy'

only one target occpancy assignment for each day



Heng CherKeng Topic Author • (97th in this Competition) • 2 days ago • Options • Reply

5

solving using 'glpk' solver (58 min) with real variables (from 0 to 1) gives objective = 67309.47862737744.
this is the lower bound for integer programming

```
solver objective =  67309.4786351117  #

optimium_cost =  67309.4786351117  #
optimium_preference_cost =  61383.472062839195
optimium_accouting_cost  =  5926.006572272518


number of variables
= 5000*100 + 176*176*100
= 3,597,600

number of constraints
= 5000+100*2 + 100*3 + 100*176
= 23,100
```

Heng CherKeng Topic Author • (97th in this Competition) • 2 days ago • Options • Reply

0

we compared with https://www.kaggle.com/vipito/santa-ip, which is a lp (real variable) solution without considering accounting cost. Instead, it uses:

```
    for d in range(NUM_DAY-1):
        model.constraint.add(occupancy[d  ]-occupancy[d+1]<= 25)
        model.constraint.add(occupancy[d+1]-occupancy[d  ]<= 25)
```

the results are solving using 'glpk' solver (2 min) with real variables (from 0 to 1) :

```
solver objective =  72926.80102040817  #

optimium_cost =  73243.48583560437
```

CPMP • (19th in this Competition) • 2 days ago • Options • Reply

0

Heng, this is interesting. I am a bit confused however. your title says linear programming while the comment I respond to says integer programming. The difference is that the former does not handle integrality constraints hence produces a lower lower bound.

I think your lower bound uses integer programming, based on the lower bounds we are getting.

fsguzi • (85th in this Competition) • 2 days ago • Options • Reply

0

I was doing something similar but didn't get any meaningful reasult yet. One thing I noticed is the number of constraints in your formulation. Mine has 100 * 2 + 99 * 176 occupancy related constraints(assuming it's the whole and initial search space), wonder what those 100 * 3 might be.

Heng CherKeng Topic Author • (97th in this Competition) • 2 days ago • Options • Reply

1

@cpmpml

you are correct! what i have implemented is relaxed integer programming. i have corrected the title. thanks!

67309.48 is the lower bound of the integer programming problem. My next step is to think of a way to get exact solution from this lower bound solution but i have no ideas for that yet

CPMP • (19th in this Competition) • 2 days ago • Options • Reply

2

Thanks, I feel better. Was afraid of a major blunder on our side ;) FYI, we get a lower bound above 68000 with cplex at root node for a model similar to this one. Maybe there should be a dual competition about lower bounds here ;) I'll suggest it to Kaggle for next year :D

mrlzla • (683rd in this Competition) • 2 days ago • Options • Reply

0

Have you already found a good solver? It seems max dimension of constraint matrix in glpk for ILP has to be less than or equal to INT_MAX(2**31 - 1)

CPMP • (19th in this Competition) • 2 days ago • Options • Reply

0

2**31 - 1 is 2 Billions, should be large enough ;)

mrlzla • (683rd in this Competition) • 2 days ago • Options • Reply

0

Yes. But if we want to solve ILP problem we have 5000 * 100 variables only for preference and constraint that describes we have to choose only one day for each family is a matrix with shape [5000, 5000 * 100]. Number of elements of the matrix is 5000 * 5000 * 100 > 2**31 - 1

CPMP • (19th in this Competition) • a day ago • Options • Reply

0

You can have an IP model with way less variables than that.

@mrlzla

a few solutions:

decision parameters reduction

- preference cost
  - consider only first 4 choice, e.g. 5000x4 variables
  - use relaxed integer programming as first solution. e.g. only 75 families are found to have fractional values. apply integer programming on those 75+ family (you will have find a way to generate sub problem)
- accounting cost
  - fix occupancy for some days (from relaxed solutions, you might be able to guess the optimum values for some days)?
  - consider accounting cost matrix in step of 2 people, then binary variables reduce from 176x176 to 88x88
  - if you already have relaxed solution you need only to consider changes made in exact solution in integer programming. The variables will be less than 176x176. E,g we expect change to be within 30, variables will be 30

In summary, fixed some variables and solved the rest. Repeat and iterate

mrlzla • (683rd in this Competition) • a day ago • Options • Reply 0

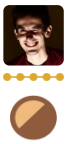
Thank you guys.
IMHO the easiest thing you can try is Gomory cutting plane.

Heng CherKeng Topic Author • (97th in this Competition) • 16 hours ago • Options • Reply 0

@mrlzla

it seems that Gomory cutting plane is automatically used by CBC solver

```
Cbc0012I Integer solution of 63485 found by DiveCoefficient after 94 iterations and
0 nodes (0.61 seconds)
Cbc0031I 29 added rows had average density of 44.37931
Cbc0013I At root node, 29 cuts changed objective from 63173.155 to 63485 in 5
passes
Cbc0014I Cut generator 0 (Probing) - 6 row cuts average 2.0 elements, 56 column
cuts (56 active)  in 0.036 seconds - new frequency is 1
Cbc0014I Cut generator 1 (Gomory) - 18 row cuts average 53.5 elements, 0 column
cuts (0 active)  in 0.008 seconds - new frequency is 1
Cbc0014I Cut generator 2 (Knapsack) - 33 row cuts average 9.7 elements, 0 column
cuts (0 active)  in 0.036 seconds - new frequency is 1
Cbc0014I Cut generator 3 (Clique) - 0 row cuts average 0.0 elements, 0 column cuts
(0 active)  in 0.000 seconds - new frequency is -100
Cbc0014I Cut generator 4 (MixedIntegerRounding2) - 15 row cuts average 11.7
elements, 0 column cuts (0 active)  in 0.012 seconds - new frequency is 1
Cbc0014I Cut generator 5 (FlowCover) - 0 row cuts average 0.0 elements, 0 column
cuts (0 active)  in 0.000 seconds - new frequency is -100
Cbc0014I Cut generator 6 (TwoMirCuts) - 119 row cuts average 96.0 elements, 0
column cuts (0 active)  in 0.008 seconds - new frequency is -100
Cbc0014I Cut generator 7 (ZeroHalf) - 13 row cuts average 37.2 elements, 0 column
cuts (0 active)  in 0.008 seconds - new frequency is -100
Cbc0001I Search completed - best objective 63485, took 94 iterations and 0 nodes
(0.61 seconds)
Cbc0035I Maximum depth 0, 1058 variables fixed on reduced cost
Cuts at root node changed objective from 63173.2 to 63485
```



Vlado Boza • (6th in this Competition) • 2 days ago • Options • Reply

2

Used very similar formulation got something around 65303.

Vlado Boza • (6th in this Competition) • 2 days ago • Options • Reply

1

Fixed, got similar bound like you.

Heng CherKeng Topic Author • (97th in this Competition) • 10 hours ago • Options • Reply

0

dimkadimon • (73rd in this Competition) • 34 minutes ago • Options • Reply

0

I think Heng's bound helped @usamec to find a bug in his code and hence find the optimal. Great work!

Heng CherKeng Topic Author • (97th in this Competition) • 21 hours ago • Options • Reply

5

reduce the number y decision variables for 100x176x176 to 100x176:

```
let y[d] = 176x1 binary variables:
then, the y[d][d+1] be be obtained  matrix multiplication:
y[d][d+1] =y[d] * transpose(y[d+1])
```

CPMP • (19th in this Competition) • 20 hours ago • Options • Reply

4

Heng, It is very interesting to see how you, not an optimization specialist, quickly explore things that a seasoned optimization practitioner would try. What you just found leads to an IP with quadratic constraints or quadratic objective, depending on the rest of the model.

Ole Kröger • (67th in this Competition) • an hour ago • Options • Reply

0

You're my personal hero of this challenge :) Fascinating to see how you tackle this problem especially as it seems like you're more like the AI guy.
Would enjoy to read a post about your thought process after the challenge! Keep up your wonderful work!