Technical Report for 2.4.1 Mode/Destination Choice for Intermediate Stop

Siyu LI

June 2016

Model Description

This model will predict both mode and destination for intermediate stop level.

Choice Set

There are 9 modes and 1092 MTZs. The choice set thus has `9*1092=9828` alternatives. (same as)

- `V_1` to `V_1092`: the mode is Public bus, destination is from 1092 MTZs.
- `V_1093` to `V_2184`: the mode is MRT/LRT, destination is from 1092 MTZs.
- `V_2185` to `V_3276`: the mode is Private bus, destination is from 1092 MTZs.
- `V_3277` to `V_4368`: the mode is Drive alone, destination is from 1092 MTZs.
- `V_4369` to `V_5460`: the mode is Shared ride 2, destination is from 1092 MTZs.
- `V_5461` to `V_6552`: the mode is Shared ride 3+, destination is from 1092 MTZs.
- `V_6553` to `V_7644`: the mode is Motorcycle, destination is from 1092 MTZs.
- `V_7645` to `V_8736`: the mode is Walk, destination is from 1092 MTZs.
- `V_8737` to `V_9828`: the mode is Taxi, destination is from 1092 MTZs.

Model Structure

The mode/destination Model is a MNL model.

```
V_counter=0
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_bus +
    cost_over_income_bus_%s * (1-missing_income_dummy) * beta_cost_bus_mrt_1 +
    cost_bus_%s*missing_income_dummy*beta_cost_bus_mrt_2 +
    tt_bus_%s * beta_tt_bus_mrt +
    beta_central_bus_mrt * central_dummy_%s +
    beta_shop * log(1+shop_%s)*shop_stop_dummy +
    beta_work * log(1+employment_%s)*work_stop_dummy +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_bus_mrt +
    beta_female_bus * female_dummy" % (V_counter,j,j,j,j,j,j,j,j))

#mrt
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_mrt +
    cost_over_income_mrt_%s * (1-missing_income_dummy) * beta_cost_bus_mrt_1 +
    cost_mrt_%s*missing_income_dummy*beta_cost_bus_mrt_2 +
    tt_mrt_%s * beta_tt_bus_mrt +
    beta_central_bus_mrt * central_dummy_%s +
    beta_shop * log(1+shop_%s)*shop_stop_dummy +
    beta_work * log(1+employment_%s)*work_stop_dummy +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_bus_mrt +
    beta_female_mrt * female_dummy" % (V_counter,j,j,j,j,j,j,j,j))

#private_bus
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_private_bus +
    cost_over_income_private_bus_%s * (1-missing_income_dummy)*beta_cost_private_bus_1 +
    cost_private_bus_%s*missing_income_dummy*beta_cost_private_bus_2 +
    tt_private_bus_%s * beta_tt_bus_mrt +
    beta_central_private_bus * central_dummy_%s +
    beta_shop * log(1+shop_%s)*shop_stop_dummy +
    beta_work * log(1+employment_%s)*work_stop_dummy +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_private_bus +
    beta_female_private_bus * female_dummy" % (V_counter,j,j,j,j,j,j,j,j))

#drive1
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_drive1 +
    cost_over_income_drive1_%s * (1-missing_income_dummy) * beta_cost_drive1_1 +
    cost_drive1_%s*missing_income_dummy*beta_cost_drive1_2 +
    tt_drive1_%s * beta_tt_drive1 +
    beta_central_drive1 * central_dummy_%s +
    beta_shop * log(1+shop_%s)*shop_stop_dummy +
    beta_work * log(1+employment_%s)*work_stop_dummy +
    (walk_distance_first_%s +
    walk_distance_second_%s)*beta_distance_drive1 +
    beta_zero_drive1 * zero_car +
    beta_oneplus_drive1 * one_plus_car +
    beta_twoplus_drive1 * two_plus_car +
    beta_threeplus_drive1 * three_plus_car +
    beta_female_drive1 * female_dummy" % (V_counter,j,j,j,j,j,j,j,j))

#share2
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_share2 +
    cost_over_income_share2_%s * (1-missing_income_dummy) * beta_cost_share2_1 +
    cost_share2_%s*missing_income_dummy*beta_cost_share2_2 +
    tt_share2_%s * beta_tt_drive1 +
    beta_central_share2 * central_dummy_%s +
    beta_shop * log(1+shop_%s)*shop_stop_dummy +
    beta_work * log(1+employment_%s)*work_stop_dummy +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_share2 +
    beta_zero_share2 * zero_car +
    beta_oneplus_share2 * one_plus_car +
    beta_twoplus_share2 * two_plus_car +
    beta_threeplus_share2 * three_plus_car +
    beta_female_share2 * female_dummy" % (V_counter,j,j,j,j,j,j,j,j))

#share3
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_share3 +
    cost_over_income_share3_%s * (1-missing_income_dummy) * beta_cost_share3_1 +
    cost_share3_%s*missing_income_dummy*beta_cost_share3_2 +
    tt_share3_%s * beta_tt_drive1 +
    beta_central_share3 * central_dummy_%s +
    beta_shop * log(1+shop_%s)*shop_stop_dummy +
    beta_work * log(1+employment_%s)*work_stop_dummy +
    (walk_distance_first_%s+walk_distance_second_%s) * beta_distance_share3 +
    beta_zero_share3 * zero_car +
    beta_oneplus_share3 * one_plus_car +
    beta_twoplus_share3 * two_plus_car +
    beta_threeplus_share3 * three_plus_car +
    beta_female_share3 * female_dummy" % (V_counter,j,j,j,j,j,j,j,j))

#motor
```

```
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_motor +
    cost_over_income_motor_%s * (1-missing_income_dummy) * beta_cost_motor_1 +
    cost_motor_%s*missing_income_dummy*beta_cost_motor_2 +
    tt_motor_%s * beta_tt_drive1 +
    beta_central_motor * central_dummy_%s +
    beta_shop * log(1+shop_%s)*shop_stop_dummy +
    beta_work * log(1+employment_%s)*work_stop_dummy +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_motor +
    beta_zero_motor * zero_motor +
    beta_oneplus_motor * one_plus_motor +
    beta_twoplus_motor * two_plus_motor +
    beta_threeplus_motor * three_plus_motor +
    beta_female_motor * female_dummy" % (V_counter,j,j,j,j,j,j,j,j))


#walk
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_walk +
    tt_walk_%s*beta_tt_walk +
    beta_central_walk * central_dummy_%s +
    beta_shop * log(1+shop_%s)*shop_stop_dummy +
    beta_work * log(1+employment_%s)*work_stop_dummy +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_walk +
    beta_female_walk * female_dummy" % (V_counter,j,j,j,j,j,j))

#taxi
for j in range(1,1093):
    V_counter=V_counter+1
    exec("V_%s = beta_cons_taxi +
    cost_over_income_taxi_%s *(1-missing_income_dummy)* beta_cost_taxi_1 +
    cost_taxi_%s*missing_income_dummy*beta_cost_taxi_2 +
    tt_taxi_%s * beta_tt_taxi +
    beta_central_taxi * central_dummy_%s +
    beta_shop * log(1+shop_%s)*shop_stop_dummy +
    beta_work * log(1+employment_%s)*work_stop_dummy +
    (walk_distance_first_%s+walk_distance_second_%s)*beta_distance_taxi +
    beta_female_taxi * female_dummy" % (V_counter,j,j,j,j,j,j,j,j))



#Estimated values for all betas
#Notice: the betas that not estimated are fixed to zero.
```

```
beta_cost_bus_mrt_1= Beta('bus/mrt cost over income',0,-10,10,0) = -0.00276
beta_cost_private_bus_1 =Beta('private bus cost over income',0,-10,10,0) = -0.00823
beta_cost_drive1_1= Beta('drive1 cost over income',0,-10,10,0) = -0.00641
beta_cost_share2_1= Beta('shared ride 2 cost over income',0,-10,10,0) = -0.0107
beta_cost_share3_1= Beta('shared ride 3 cost over income',0,-10,10,0) = -0.0120
beta_cost_motor_1 = Beta('motor cost over income',0,-10,10,0) = -0.0106
beta_cost_taxi_1 = Beta('taxi cost over income',0,-10,10,0) = -0.000389

beta_cost_bus_mrt_2= Beta('bus/mrt cost (missing income)',0,-10,10,1) = 0
beta_cost_private_bus_2 =Beta('private bus cost (missing income)',0,-10,10,1) = 0
beta_cost_drive1_2= Beta('drive1 cost (missing income)',0,-10,10,1) = 0
beta_cost_share2_2= Beta('shared ride 2 cost (missing income)',0,-10,10,1) = 0
beta_cost_share3_2= Beta('shared ride 3 cost (missing income)',0,-10,10,1) = 0
beta_cost_motor_2 = Beta('motor cost (missing income)',0,-10,10,1) = 0
beta_cost_taxi_2 = Beta('taxi cost (missing income)',0,-10,10,1) = 0


beta_tt_bus_mrt = Beta('bus/mrt travel time',0,-10,10,0) = - 4.11
beta_tt_private_bus= Beta('private bus travel time',0,-10,10,1) = 0
beta_tt_drive1 = Beta('drive1 travel time',0,-10,10,0) = -4.55
beta_tt_share2= Beta('shared ride 2 travel time',0,-10,10,1) = 0
beta_tt_share3= Beta('shared ride 3 travel time',0,-10,10,1) = 0
beta_tt_motor=Beta('motor travel time',0,-10,10,1) = 0
beta_tt_walk=Beta('walk travel time',0,-10,10,0) = -4.11
beta_tt_taxi= Beta('taxi travel time',0,-10,10,0) = -4.14

beta_work = Beta('work beta',0,-10,10,0) = 0.980
beta_shop = Beta('shop beta',0,-10,10,0) = 0.500


beta_central_bus_mrt = Beta('central dummy beta for bus/mrt',0,-10,10,0) = -0.0988
beta_central_private_bus=Beta('central dummy beta for private bus',0,-10,10,0) = -0.101
beta_central_drive1=Beta('central dummy beta for drive1',0,-10,10,1) = 0
beta_central_share2=Beta('central dummy beta for shared ride 2',0,-10,10,0) = 0.0441
beta_central_share3=Beta('central dummy beta for shared ride 3',0,-10,10,0) = 0.130
beta_central_motor=Beta('central dummy beta for motor',0,-10,10,0) = -0.179
beta_central_walk=Beta('central dummy beta for walk',0,-10,10,0) = -2.85
beta_central_taxi=Beta('central dummy beta for taxi',0,-10,10,0) = 0.911

beta_distance_bus_mrt = Beta('distance beta for bus/mrt',0,-10,10,0) = 0.0190
beta_distance_private_bus=Beta('distance beta for private bus',0,-10,10,0) = -0.101
beta_distance_drive1=Beta('distance beta for drive1',0,-10,10,1) = 0
beta_distance_share2=Beta('distance beta for shared ride 2',0,-10,10,0) = -0.00970
beta_distance_share3=Beta('distance beta for shared ride 3',0,-10,10,0) = -0.0129
beta_distance_motor=Beta('distance beta for motor',0,-10,10,0) = -0.00195
beta_distance_walk=Beta('distance beta for walk',0,-10,10,1) = 0
```

```
beta_distance_taxi=Beta('distance beta for taxi',0,-10,10,0) = 0.0000385


beta_cons_bus = Beta('constant for bus',0,-10,10,0) = 2.71
beta_cons_mrt = Beta('constant for mrt',0,-10,10,0) = 2.72
beta_cons_private_bus = Beta('constant for private bus',0,-10,10,0) = 3.93
beta_cons_drive1 = Beta('constant for drive1',0,-10,10,1) = 0
beta_cons_share2 = Beta('constant for shared ride 2',0,-10,10,0) = 0.597
beta_cons_share3 = Beta('constant for shared ride 3',0,-10,10,0) = 0.895
beta_cons_motor = Beta('constant for motor',0,-10,10,0) = -5.61
beta_cons_walk = Beta('constant for walk',0,-10,10,0) = -4.67
beta_cons_taxi = Beta('constant for taxi',0,-10,10,0) = -2.85

beta_zero_drive1=Beta('zero cars in drive1',0,-10,10,1) = 0
beta_oneplus_drive1=Beta('one plus cars in drive1',0,-10,10,0) = 1.42
beta_twoplus_drive1=Beta('two plus cars in drive1',0,-10,10,0) = 0.669
beta_threeplus_drive1=Beta('three plus cars in drive1',0,-10,30,1) = 0

beta_zero_share2=Beta('zero cars in share2',0,-10,10,1) = 0
beta_oneplus_share2=Beta('one plus cars in share2',0,-10,10,0) = 1.54
beta_twoplus_share2=Beta('two plus cars in share2',0,-10,10,0) = 0.716
beta_threeplus_share2=Beta('three plus cars in share2',0,-10,10,1) = 0

beta_zero_share3=Beta('zero cars in share3 plus',0,-10,10,1) = 0
beta_oneplus_share3=Beta('one plus cars in share3 plus',0,-10,10,0) = 1.89
beta_twoplus_share3=Beta('two plus cars in share3 plus',0,-10,10,0) = 0.711
beta_threeplus_share3=Beta('three plus cars in share3 plus',0,-30,10,1) = 0


beta_zero_motor=Beta('zero motors in motor',0,-10,10,1) = 0
beta_oneplus_motor=Beta('one plus motors in motor',0,-10,10,0) = 3.87
beta_twoplus_motor=Beta('two plus motors in motor',0,-10,10,0) = -1.07
beta_threeplus_motor=Beta('three plus motors in motor',0,-10,10,1) =0

beta_female_bus=Beta('female dummy in bus',0,-10,10,0) = 0.958
beta_female_mrt=Beta('female dummy in mrt',0,-10,10,0) = 1.04
beta_female_private_bus=Beta('female dummy in privatebus',0,-10,10,0) = 0.876
beta_female_drive1=Beta('female dummy in drive1',0,-10,10,1) = 0
beta_female_share2=Beta('female dummy in share2',0,-10,10,0) = 0.681
beta_female_share3=Beta('female dummy in share3 plus',0,-10,10,0) = 0.320
beta_female_motor=Beta('female dummy in motor',0,-10,10,0) = 0.687
beta_female_taxi=Beta('female dummy in taxi',0,-10,10,0) = 1.52
beta_female_walk=Beta('female dummy in walk',0,-50,10,1) = 0
```

Variables

Basically the variables used in mode/destination choice model is the same with variables used in mode choice model: travel time, travel cost, zonal information, personal and household characteristics. The difference is that for mode/destination choice, the destination of a tour is not determined. Thus we need to provide travel time, travel cost, zonal information for each of the 1092 zones.

Previous to generating related variables, data sets (`am`,`pm`,`op` and `zone_employment`) can be stored in dicts for fast querying.

Noticed that there are few differences that distinguish the mode/destination choice model from mode/destination choice model of tour level.

1. The selection of origin and destination: For stops on the first half-tour, the origin is the destination of last modeled stop on the same half tour. In case the the stop is the very first stop being modeled in the half-tour, the origin is the primary activty location of tour. And it is destination that need to be determined in the model. For stops on the second half-tour, the origin is the destination of last modeled stop on the same half tour. In case the stop is the very first stop being modeled in the half-tour, the origin is the primary activity location of the tour.

2. Travel time and cost: In stop mode/destination choice model, what we calculate is marginal travel time and cost: travel time/cost from origin to destinaiton plus from destination to home minus from destination directly to home.

```
# From gen.py
import numpy as np
am=np.genfromtxt(file_dir+'AMcosts.dat',names=True)
pm=np.genfromtxt(file_dir+'PMcosts.dat',names=True)
op=np.genfromtxt(file_dir+'OPcosts.dat',names=True)
zone_employ=np.genfromtxt(file_dir+'zone_employment.txt',names=True)
AM={};PM={}; OP={}; ZONE={}

for row in am:
    AM[(int(row['origin']),int(row['destin']))]=\
    {'distance': row['AM2dis'], \
    'car_ivt' : row['AM2Tim']/60,\
    'pub_ivt':  row['AM2ivt']/60, \
    'pub_out':  (row['AM2aux']+row['AM2wtt'])/60, \
    'pub_wtt': row['AM2wtt']/60,\
    'pub_walkt': row['AM2aux']/60,\
    'car_cost_erp': row['AM2ERP']/100, \
    'avg_transfer': row['AM2trf'],\
    'pub_cost': row['AM2cos']/100}

for row in pm:
    PM[(int(row['origin']),int(row['destin']))]=\
    {'distance': row['PM2dis'], \
    'car_ivt':   row['PM2Tim']/60,\
    'pub_ivt':  row['PM2ivt']/60, \
    'pub_out':  (row['PM2aux']+row['PM2wtt'])/60, \
    'pub_wtt': row['PM2wtt']/60,\
    'pub_walkt': row['PM2aux']/60,\
    'car_cost_erp': row['PM2ERP']/100, \
    'avg_transfer': row['PM2trf'],\
    'pub_cost': row['PM2cos']/100}

for row in op:
    OP[(int(row['origin']),int(row['destin']))]=\
    {'distance': row['OPdis'], \
    'car_ivt':   row['OPTim']/60,\
    'pub_ivt':  row['OPivt']/60, \
    'pub_out':  (row['OPaux']+row['OPwtt'])/60, \
    'pub_wtt': row['OPwtt']/60,\
    'pub_walkt': row['OPaux']/60,\
    'car_cost_erp': row['OPERP']/100, \
    'avg_transfer': row['OPtrf'],\
    'pub_cost': row['OPcos']/100}

for row in zone_employ:
    ZONE[int(row['zone_ID'])]=\
    {'zone_id':int(row['zone_ID']),\
    'zone_code':int(row['zone_code']),\
    'employment':row['employment'],\
    'central_dummy': row['central_dummy'],\
    'parking_rate': row['parking_rate'],\
    'population':row['population'],\
    'area':row['area']}
```

Noted that in the original skims `AMCosts.dat`,`PMCosts.dat`,`OPCosts.dat`, costs are measured in cent and time is measured in minutes. For our variable, we need to first convert them to dollor and hour when generating the dicts (divided by 100 and 60 respectively)

Time/Cost Related Variables

```
#define cost and travel time

for i in range(1,1093):
    exec("cost_bus_%s = cost_public_%s" % (i,i))
    exec("cost_mrt_%s = cost_public_%s" % (i,i))
    exec("cost_private_bus_%s = cost_public_%s" % (i,i))
    exec("cost_drive1_%s=cost_car_ERP_%s+cost_car_OP_%s+cost_car_parking_%s" % (i,i,i,i))
    exec("cost_share2_%s=(cost_car_ERP_%s+cost_car_OP_%s+cost_car_parking_%s)/2.0" % (i,i,i,
    exec("cost_share3_%s=(cost_car_ERP_%s+cost_car_OP_%s+cost_car_parking_%s)/3.0" % (i,i,i,
    exec("cost_motor_%s=0.5*cost_car_ERP_%s+0.5*cost_car_OP_%s+0.65*cost_car_parking_%s" % (
    exec("cost_taxi_%s=(6.8+cost_car_ERP_%s+6*central_dummy_%s+((walk_distance_first_%s*(wal
    exec("cost_over_income_bus_%s=30*cost_bus_%s/(0.5+income_mid)" % (i,i))
    exec("cost_over_income_mrt_%s=30*cost_mrt_%s/(0.5+income_mid)" % (i,i))
    exec("cost_over_income_private_bus_%s=30*cost_private_bus_%s/(0.5+income_mid)" % (i,i))
    exec("cost_over_income_drive1_%s=30*cost_drive1_%s/(0.5+income_mid)" % (i,i))
    exec("cost_over_income_share2_%s=30*cost_share2_%s/(0.5+income_mid)" % (i,i))
    exec("cost_over_income_share3_%s=30*cost_share3_%s/(0.5+income_mid)" % (i,i))
    exec("cost_over_income_motor_%s=30*cost_motor_%s/(0.5+income_mid)" % (i,i))
    exec("cost_over_income_taxi_%s=30*cost_taxi_%s/(0.5+income_mid)" % (i,i))

for i in range(1,1093):
    exec("tt_bus_%s = tt_public_ivt_%s +tt_public_out_%s" % (i,i,i))
    exec("tt_mrt_%s = tt_public_ivt_%s +tt_public_out_%s" % (i,i,i))
    exec("tt_private_bus_%s = tt_car_ivt_%s" % (i,i))
    exec("tt_drive1_%s = tt_car_ivt_%s +1.0/12" % (i,i))
    exec("tt_share2_%s = tt_car_ivt_%s + 1.0/12" % (i,i))
    exec("tt_share3_%s = tt_car_ivt_%s + 1.0/12" % (i,i))
    exec("tt_motor_%s = tt_car_ivt_%s+ 1.0/12" % (i,i))
    exec("tt_walk_%s = (walk_distance_first_%s+walk_distance_second_%s)/5/2" % (i,i,i))
    exec("tt_taxi_%s = tt_car_ivt_%s +1.0/12" % (i,i))
```

see attached `intermediate_mode_destination.py` for how to generate variables.
Other Variables ————————————-

```
employment_%s = ZONE[%s]['employment'] # % destination
shop_%s = ZONE[%s]['shop'] # % destination

# The following dummy variables are generated based on
# household car, motorcycle ownership.

zero_car = 1*(car_own_normal==0)
```

```
one_plus_car = 1*(car_own_normal>=1)
two_plus_car = 1*(car_own_normal>=2)
three_plus_car = 1*(car_own_normal>=3)

zero_motor = 1*(motor_own==0)
one_plus_motor = 1*(motor_own>=1)
two_plus_motor = 1*(motor_own>=2)
three_plus_motor = 1*(motor_own>=3)

work_stop_dummy=1*(stop_type==1)
edu_stop_dummy=1*(stop_type==2)
shop_stop_dummy=1*(stop_type==3)
other_stop_dummy=1*(stop_type==4)

stop type is a dynamic variable from intermediate stop generation.
```

Availability of Alternatives

The availibility of a mode-destination pair is affected by both the mode and destination. If a mode is not available for an agent, all mode-destination pairs have the mode will not be available. If a destination is not available for an agent(e.g., `destination==origin`), all mode-destination pairs have the destination will not be available. Besides, we have cases where both mode and destination need to consider to determine its availability.

1. if the `destination == origin`, the destination is not available.
2. public bus, private bus and MRT/LRT are only available `if AM[(origin,destination)]['pub_ivt']>0 and PM[(destination,origin)]['pub_ivt']>0:`
3. shared2, shared3+, taxi and motorcycle are available to all.
4. Walk is only avaiable `if (AM[(origin,destination)]['distance']<=5 and PM[(destination,origin)]['distance']<=5)`
5. drive alone is available when for the agent, `has_driving_license * one_plus_car == True`

Except the above constrains, the mode of intermediate stop is constrained by tour mode.

```
'''
Mode priority
1. private bus    (id==3)     [1,2,3,4,5,6,7,8,9]
```

```
2. public Bus/MRT (id==1,2)   [1,2,4,5,6,7,8,9]
3. shared3        (id==6)     [4,5,6,7,8,9]
4. shared2        (id==5)     [4,5,7,8,9]
5. drive alone    (id==4)     [4,7,8,9]
6. taxi           (id==9)     [7,8,9]
7. motor          (id==7)     [7,8]
8. walk           (id==8)     [8]
'''
tour_trip_avail={}
tour_trip_avail[3]=[1,2,3,4,5,6,7,8,9]
tour_trip_avail[1]=[1,2,4,5,6,7,8,9]
tour_trip_avail[2]=[1,2,4,5,6,7,8,9]
tour_trip_avail[6]=[4,5,6,7,8,9]
tour_trip_avail[5]=[4,5,7,8,9]
tour_trip_avail[4]=[4,7,8,9]
tour_trip_avail[9]=[7,8,9]
tour_trip_avail[7]=[7,8]
tour_trip_avail[8]=[8]f
```