|  |
| --- |
| if \_\_name\_\_=='\_\_main\_\_':  iteration\_number=10  current\_number = 2  # obtain the initial feasble solution  # generate the initial solution of sub-problems  //output: output\_agent.csv as the initial solution/extreme point.  os.system(r'"C:\Users\jliu215\Desktop\DW-Interaction\2\_Small\_base\_test\_DW\_Interaction\Initial\_Sub-problems\_VRP.exe"')  g\_output\_initial\_agent\_VRP\_solution()  # generate the marginal values  //output: GAMS\_arc\_cap\_marg.csv; GAMS\_vehicle\_weight\_marg.csv; GAMS\_passenger\_pickup\_marg.csv;  ws\_INI = GamsWorkspace()  master\_problem\_INI = ws\_INI.add\_job\_from\_file("C:\\Users\\jliu215\\Desktop\\DW-Interaction\\2\_Small\_base\_test\_DW\_Interaction\\Vehicle assignment\_master.gms")  master\_problem\_INI.run()    # iteration by itetation to imporve the solution  for i in range(2,iteration\_number+1):  // Py\_output\_num\_VRP.csv:output the iteration number to VRP.exe to record the kth paths/iterations  g\_output\_ite\_num\_VRP(i,iteration\_number)  print("iteration\_number:%d" % i)  # generate the new solution of sub-problems  // output\_agent\_internal.csv:to record the output agent solution each iteration. Sub\_GAMS\_input\_arc\_cap.csv, Sub\_GAMS\_input\_incid\_veh\_path\_arc.csv, Sub\_GAMS\_input\_incid\_veh\_path\_pax.csv, Sub\_GAMS\_input\_set.csv, Sub\_GAMS\_input\_veh\_path\_cost.csv, Sub\_GAMS\_input\_veh\_path\_selection.csv, those files are from VRP.exe based on the path solution of each iteration.  os.system(r'"C:\Users\jliu215\Desktop\DW-Interaction\2\_Small\_base\_test\_DW\_Interaction\Sub-problems\_VRP.exe"')  //Internal\_GAMS\_input\_incid\_veh\_path\_arc.csv, Internal\_GAMS\_input\_arc\_cap.csv, Internal\_GAMS\_input\_incid\_veh\_path\_pax.csv, Internal\_GAMS\_input\_set.csv, Internal\_GAMS\_input\_veh\_path\_cost.csv, Internal\_GAMS\_input\_veh\_path\_selection.csv, those are the current new network based on the previous path solution and current path solution, to update the network as input of GAMS.  g\_UpdateNetworkData()  //based on the latest network, the required txt formats for GAMS is regenerated.  g\_output\_GAMS\_inputs()  // output the agent path solution based on the current result and previous results.  g\_output\_agent\_VRP\_solution()  # generate the marginal values of master problem  //output: GAMS\_arc\_cap\_marg.csv; GAMS\_vehicle\_weight\_marg.csv; GAMS\_passenger\_pickup\_marg.csv;  ws = GamsWorkspace()  master\_problem = ws.add\_job\_from\_file("C:\\Users\\jliu215\\Desktop\\DW-Interaction\\2\_Small\_base\_test\_DW\_Interaction\\Vehicle assignment\_master.gms")  master\_problem.run()  if(i==iteration\_number):  // at the last iteration, we output the finally identifiable solution.(1)output: GAMS\_vehicle\_path\_solution.csv(final solution from GAMS:lamda(v,k)); (2)output\_GAMS\_solution\_agent.csv(final solution with path details: the path node/time/state sequence of path k of vehicle v);  g\_output\_GAMS\_solution()  print("well done!")  // output\_agent\_path\_arc\_incidence.csv is useless, and was used to output the path arc incidence for GAMS previously. |