README

This README file serves as a guide for executing the code related to Section 4.3.1 of the paper. Specifically, it explains how to solve the CIOP for the WM game.

1 Summary of the Functions

1. *function [output_args] = StartFunctionWM (n, option)*

This function is the main function, including four options, each of which calls a different function.

2. *function [edge] = GenerateWM (m)*

This function is used to randomly generate four groups of WM instances with different sizes of the graph (N, E) as described in Section 4.3.1 of the paper. Our 100 WM instances are generated using this function. Since random instances have already been generated, you can directly access them in the data folder.

3. *function [xvector,optimalvalue] = SolveWM (m, edge)*

This function constructs a network for a WM problem and calculates its optimal solution, denoted by x^*, and its optimal value, denoted by v^*. A sub-function of *SolveWM (m, edge)* is defined by *function outmate = PairIndex (inedges, inmaxcardinality)*, and it is used to compute the optimal pairs of matching of a WM problem.

4. *function [adjpernum,adjpervalue,finaladj,edgefinal,xinverse] = InvCostAdjWM (m, dataread)*

This function is used to solve the CIOP for the WM game, where $[l, u] = [v^*, v^*]$. For each group of WM instances with the same size, we solve the CIOP for the WM game to obtain the optimal cost adjustment using the solution algorithm illustrated in Section 4.1 of the paper. As

detailed in Section 1.2 of the paper, the inverse optimization problem is an *inverse optimal solution problem.*

5. *function [adjpernum, adjpervalue, finaladj, edgefinal, xinverse] = InvCostAdjWMRange (m, dataread)*

This function is used to solve the CIOP for the WM game, where $[l, u] = [0.95v^*, 1.05v^*]$. As detailed in Section 1.2 of the paper, the inverse optimization problem encompasses both an *inverse optimal solution problem* and an *inverse optimal value problem.*

6. *function [table2, table3] = FinalResultWM*

This function is used to compute statistical metrics for the results of 100 instances, including column U, column $DV\%$, column $DN\%$, and column T, which are the results presented in Table 2 and Table 3 of the paper.

2 Code Instructions

To run the code smoothly and correctly, please set the variables in the file "WMG \src\ StartfunctionWM.m" as described below. Matlab 2023a is recommended.

1. set parameter: n.

n is the number of nodes.

2. set parameter: option.

option = 1 is to generate and save random instances and compute the optimal solution;

option = 2 is to generate instances to compute the CIOP for WM game;

option = 3 is to generate instances to compute the CIOP for WM game with range;

option = 4 is to compute statistical metrics for the results of 100 instances (set n=0).