Due to the OpenReview max file size limitation, we upload a representative subset of the dataset. All the dataset will be provided if the paper is accepted.
To run all the three benchmarks, please enter "code&data" folder.

"MSPPO_ProdSales" Folder

Code and data for the production and sales problem.

1. To run Baseline method:
    1. Enter "Baseline" folder
    2. Train NN using Baseline: run *"python3 train_Baseline.py <low or high profit product> <simulation_startmark> <simulation_endmark>"*
    E.g. To train on low-profit product groups on 12 stages using simulation 0 training and testing instances, run: *"python3 train_Baseline.py 0 0 1"*
2. To run SCD method:
    1. Enter "SCD" folder
    2. Train NNs using SCD: run *"python3 train_SCD.py <low or high profit product> <simulation_startmark> <simulation_endmark>"*
    E.g. To train on low-profit product groups on 12 stages using simulation 0 training and testing cases, run: *"python3 train_SCD.py 0 0 1"*
3. To run PCD method:
    1. Enter "PCD" folder
    2. Get true optimal solutions to run iteration 0: run *"python3 get_TOV_train.py <low or high profit product> <simulation_startmark> <simulation_endmark>"*
    E.g. To get true optimal solutions of low-profit product groups on 12 stages with simulation 0 training instances, run: *"python3 get_TOV_train.py 0 0 1"*
    3. Train NNs using PCD: run *"python3 train_PCD.py <low or high profit product> <simulation_startmark> <simulation_endmark>"*
    E.g. To train on low-profit product groups on 12 stages using simulation 0 training and testing cases, run: *"python3 train_PCD.py 0 0 1"*
4. To run classical regression methods and testing:
    1. Enter "test" folder
    2. Run classical regression methods: run *"python3 trad_model.py <simulation_startmark> <simulation_endmark>"*
    E.g. To train on 12 stages with simulation 0 training and testing instances, run: *"python3 trad_model.py 0 1"*
    3. Run testing: run *"python3 test.py <low or high profit product> <simulation_startmark> <simulation_endmark>"*
    E.g. To run the testing on low-profit product groups on 12 stages using simulation 0 training and test cases, run: *"python3 test.py 0 0 1"*

"MSPPO_Investment" Folder

Code and data for the investment problem.

1. To run Baseline method:
    1. Enter "Baseline" folder
    2. Train NN using Baseline: run *"python3 train_Baseline.py <capital> <transaction factor> <simulation_startmark> <simulation_endmark>"*
    E.g. To train on capital=25, transaction factor=0.01 on 4 stages using simulation 0 training and testing instances, run: *"python3 train_Baseline.py 25 0.01 0 1"*
2. To run SCD method:
    1. Enter "SCD" folder
    2. Train NNs using SCD: run *"python3 train_SCD.py <capital> <transaction factor> <simulation_startmark> <simulation_endmark>"*
    E.g. To train on capital=25, transaction factor=0.01 on 4 stages using simulation 0 training and testing instances, run: *"python3 train_SCD.py 25 0.01 0 1"*
3. To run PCD method:
    1. Enter "PCD" folder
    2. Get true optimal solutions to run iteration 0: run *"python3 get_TOV_train.py <capital> <transaction factor> <simulation_startmark> <simulation_endmark>"*
    E.g. To get true optimal solutions of capital=25, transaction factor=0.01 on 4 stages with simulation 0 training instances, run: *"python3 get_TOV_train.py 25 0.01 0 1"*
    3. Train NNs using PCD: run *"python3 train_PCD.py <capital> <transaction factor> <simulation_startmark> <simulation_endmark>"*
    E.g. To train on capital=25, transaction factor=0.01 on 4 stages using simulation 0 training and testing instances, run: *"python3 train_PCD.py 25 0.01 0 1"*
4. To run classical regression methods and testing:
    1. Enter "test" folder
    2. Run classical regression methods: run *"python3 trad_model.py <simulation_startmark> <simulation_endmark>"*
    E.g. To train on 4 stages with simulation 0 training and testing instances, run: *"python3 trad_model.py 0 1"*
    3. Run testing: run *"python3 test.py <capital> <transaction factor> <simulation_startmark> <simulation_endmark>"*
    E.g. To run the testing on capital=25, transaction factor=0.01 on 4 stages using simulation 0 training and test cases, run: *"python3 test.py 25 0.01 0 1"*

“MSPPO_NRP” Folder

Code and data for the nurse rostering problem.

1. To run Baseline method:
    1. Enter “Baseline” folder
    2. Train NN using Baseline: run *“python3 train_Baseline.py <extra nurse payment> <simulation_startmark> <simulation_endmark>”*
       E.g. To train on extra nurse payment is 15 on 7 stages using simulation 0 training and testing instances, run: *“python3 train_Baseline.py 15 0 1”*
2. To run SCD method:
    1. Enter “SCD” folder
    2. Train NNs using SCD: run *“python3 train_SCD.py <extra nurse payment> <simulation_startmark> <simulation_endmark>”*
       E.g. To train on extra nurse payment is 15 on 7 stages using simulation 0 training and testing instances, run: *“python3 train_SCD.py 15 0 1”*
3. To run PCD method:
    1. Enter “PCD” folder
    2. Get true optimal solutions to run iteration 0: run *“python3 get_TOV_train.py <extra nurse payment> <simulation_startmark> <simulation_endmark>”*
       E.g. To get true optimal solutions of extra nurse payment is 15 on 7 stages with simulation 0 training instances, run: *“python3 get_TOV_train.py 15 0 1”*
    3. Train NNs using PCD: run *“python3 train_PCD.py <extra nurse payment> <simulation_startmark> <simulation_endmark>”*
       E.g. To train on extra nurse payment is 15 on 7 stages using simulation 0 training and testing instances, run: *“python3 train_PCD.py 15 0 1”*
4. To run classical regression methods and testing:
    1. Enter “test” folder
    2. Run classical regression methods: run *“python3 trad_model.py <simulation_startmark> <simulation_endmark>”*
       E.g. To train on 7 stages with simulation 0 training and testing instances, run: *“python3 trad_model.py 0 1”*
    3. Run testing: run *“python3 test.py <extra nurse payment> <simulation_startmark> <simulation_endmark>”*
       E.g. To run the testing on extra nurse payment is 15 on 7 stages using simulation 0 training and test cases, run: *“python3 test.py 15 0 1”*