Full Algorithm diagrams:

1. Creation of Environment:

123

123

1. Execution of Simulator:

Pass Environment to Simulator in its constructor

Setup number of time-steps

simulate() called

Yes

Yes

No

Finished

saveResults()

currentTimeStep < totalTimeSteps

Environment->calculateAllProfitMagins()

Environment->calculateAllMarketShares()

Environment->executeAllMoves()

No

i < StrategicPlayers.size()

i++

StrategicPlayers[i]->decideNextMove()

int i = 0

Int currentTimeStep = 0

Yes

1. Execute DecisionMaker:

Set as active the node with least effort from all open nodes

Evaluate total effort of new nodes

Connect them to active node

Generate new nodes in reach

Calculate optimal through HEB optim.

decideNextMove () called

Finish

Yes

No

Total final effort is lower than open nodes’ effort

Calculate total effort

Connect active node to optimal

! optimal.isInReach()

Yes

No