Robust Factored MDP Solver using Cutting Plane Method

Description

This C++ program package implements a cutting plane solution approach for Robust Factored Markov Decision Processes (MDPs). It incorporates a comparison framework that evaluates the performance of policies generated by clairvoyant, nominal (non-robust), and robust formulations based on historical data. The program is designed to investigate the effectiveness of different strategic formulations under uncertainty.

Key Components

- `historical.cpp`: Generates historical observations using a simple strategy to provide a basis for comparison among different formulations.
- `main.cpp`: The main driver program that orchestrates the cutting plane method and the comparative analysis of generated policies.
- `cutting_plane.cpp`: Implements the cutting plane method for clairvoyant, nominal (non-robust), and robust formulations, interacting with other components for iterative solution refinement.
- `topology.cpp`: Contains functions to create network topology, define transition probabilities, and establish basis functions.
- `parameter.h`: Defines the problem parameters, setting the stage for the robust and non-robust problem formulations.
- `master_prob.cpp`: Constructs and solves the master problem, employing a warm start approach to enhance computational efficiency.
- `local_search.cpp`: Heuristically solves the subproblem using a coordinate-wise descent method, aimed at optimizing local solution quality.
- `sub_problem.cpp`: Utilizes Mixed Integer Linear Programming (MILP) to accurately solve subproblems, critical for the robust formulation's integrity.
- `monte_carlo.cpp`: Employs the Monte Carlo method to simulate real-world scenarios, estimating the expected returns of different policies.

Dependencies

- C++11 compliant compiler.
- External libraries or solvers for MILP, such as CPLEX or Gurobi, are required for specific components like `sub_problem.cpp`.