Factored MDP Solver using Cutting Plane Method

Description

This C++ program package is designed to solve Factored Markov Decision Processes (MDPs) using the cutting plane method. The core of the program, `main.cpp`, orchestrates the process by calling various functions that collectively aim to find the optimal policy for given Factored MDPs.

Key Components

- `main.cpp`: The main driver of the program that ties together all the components and functionalities.
- `topology.cpp`: Contains functions for creating the network topology, defining transition probabilities, and establishing basis functions.
- `parameter.h`: Houses the definitions of various parameters that dictate the problem space and configurations.
- `cutting_plane.cpp`: Implements the cutting plane method for scope-one, scope-two and scope-three formulations, interacting with other components for iterative solution refinement.
- `master_prob.cpp`: Constructs and solves the master problem, utilizing a warm start approach for efficiency.
- `local_search.cpp`: Heuristically solves the subproblem through a coordinate-wise descent method, aimed at finding local optima.
- `sub_problem.cpp`: Employs Mixed Integer Linear Programming (MILP) techniques to solve subproblems accurately.
- `naïve_approaches.cpp:` Includes three naïve approaches and their Monte Carlo simulation code: random, priority and level.
- `monte_carlo.cpp`: Simulates real-world scenarios using the Monte Carlo method to estimate the expected returns of policies.
- `Variables_elimination.cpp`: Implements the Variable Elimination method proposed by Guestrin et al. (2003), a foundational technique for tackling Factored MDPs.
- `Auxiliary.cpp:` Other auxiliary functions.

Dependencies

- This program requires a C++11 compliant compiler. Specific components, such as `sub_problem.cpp`, might require external libraries or solvers for MILP (e.g., CPLEX and Gurobi) which should be installed and properly configured on your system.