Discrete mapd simulation app documentation

petr.smid

March 2023

Contents

1 Purpose of the app

This app allows users to plan collision free paths in both warehouse and random environment. The plans can be further visualised and validated, such that the user can easily verify non-collision paths.
Each plan contains cost and time info, various algorithms can be compared in different environment and eventually user can implement own algorithm and test it against already implemented performance.
Two modes are available: Scenario with given number of orders and stress test. With given number of orders, output is simply plan solving the problem with selected algorithm.
Stress test allows user to test limit of their algorithms. Solves problem for k orders (starting at 1) and then repeat with k+1, until either given time limit is reached or plan for all orders in given scenario is created.

2 Step-by-step

First, load map file using Upload map button. Then load scenario for given map using Upload scenario. (For correct file formats, see Input).
Select algorithm using "Choose Algorithm" button. If no algorithm is selected, default "mapd greedy" is selected.
Optionally, selects different output file (Default is {*mapname*}.plan), modify scenario using Scenario Controlls)
Choose mode (regular or stress test) and set parametres. See)
Run the algorithm. Plan is then exported into outputfile.plan . If visualisation option is selected, user may view the simulation and validate plan accoding to here).

3 Main Window

3.1 Problem definition

- Upload map
  Loads .map (environment). For correct format see "input".
  In case file is corrupted, warning will show up and the abort loading.
  This function is available only if no other map is active.

- Upload Scenario
  Loads .scen (scenario). For correct format see "input".
  In case file is corrupted, warning will show up and the abort loading.
  This function is available only if no other scenario/plan is active.

- Load Plan
  Loads .plan (Plan). For correct format see "input".
  In case file is corrupted, warning will show up and the abort loading.
  This function is available only if no other scenario/plan is active.

- Choose Algorithm
  Promps window where user can select algorithm to solve the problem. Upon advanced algorithm selection, additional algorithm settings will open.

- Output File
  User selects output file/directory for .plan, and PlanValidation.pdf files.

3.2 Problem settings

- stress test checkbox:
  If checked, stress test mode is selected. User can choose to output plans for every iteration of the test via "Output all planns" checkbox.

- Show plan / Show plan for best solution
  Upon creating plan, visualisation is created and can be viewed in the map window.

- Time limit in seconds
  For stress test only, user inputs maximum number in seconds, for one iteration (IE when iteration for k orders fails to find plan within time limit, stress test terminates with best solution being k-1 orders)

- Number of orders
  For regular mode only, user select target number n of orders for the problem. First n orders from scenario is then loaded into the problem. For loading all orders within scenario file, enter arbitrary large number.

- Save and Create Plan
  Saves current setting above,checks its validity and assigns orders to the agents. If valid, user can proceed to solve by hitting Create Plan button. Create Plan button is disabled before validation of settings.

3.3 Scenario Controlls

On the top of this section, info about map and scenario is displayed.

- Add new Agent
  User can add new Agent with id and location via new Agent Window. Valid location and id must be selected.

- Add new Order
  User can add new Order with id and location via new Order Window. Valid initial,target location and id must be selected.

- Clear orders
  Clears current loaded orders from scenario and their assignments. New problem (different number of orders or stress test) can be run on same scenario

- Clear scenario
  Clear current scenario. New scenario needs to be loaded from file via Upload scenario.

- Clear map
  Clears current map and scenario. To proceed, user needs to load a new map via Upload map and new scenario via Upload scenario buttons.

4 Input Format

The input is divided into two parts. Map (the environment) and scenario (order locations, targer locations, time, agent locations etc). Both are compatible with MAPF benchmarks viz: https://movingai.com/benchmarks/mapf/index.html

4.1 .map

Map format remain unchanged from MAPF benchmark. For correct format, please see "Example.map" or benchmark documentation
For best performance, use maps with dimensions 64x64 and lower. Bigger maps work the same way, however the visualisation experience will decrease with maps with dimensions over 128x128.

4.2 .scen

Example from mapf benchmark:
23 room-64-64-16.map 64 64 45 7 4 56 92.04163055
28 room-64-64-16.map 64 64 54 36 3 55 113.84062042
In scenario, instead of searching path from A to B (normal MAPF problem) we translate this data to orders. In above example, we obtain 2 orders with it initial and target location, map type co verify correct usage and Manhattan distance.
Example with time data:
21 room-64-64-16.map 64 64 3 43 47 20 84.69848480 10
14 room-64-64-16.map 64 64 28 47 23 12 59.69848480 25
In above example, orders appear availible in time 10 and 25 respectively. One time unit is one step in algorithm, or the distance of neighbour tiles. If no time is given, default value of 0 is used
Agents need to be added to plan file manually, example:
A room-64-64-16.map 64 64 1 18

or generated in app itself.

4.3 .plan

File *.plan is generated by the app and can be later loaded via main window. User can view visualisation of the plan, but validation must be done upon creation of the plane.
If the file is corrupted, warning is thrown and the visualisation stops.

5 Simulation Window

Upon solving given problem and creating visualisation, user can view the plan in Simulation Window. Map is visualised such that empty spaces are white tiles,

walls and obsticles are black tiles.
Before executing plan visualisation, all orders are colored according to their assigned agent. To view info about agent/order, hover mouse over the colored square. In right column, info is displayed.
To start a plan, click Resume button, to pause plan visualisation, click Pause button.
Info such as current cost, time and state of a plann is displayed in the right corner.
If user wants to export validation, click Validate button. Current visualisation will be paused and validation process will begin.
In settings, user may switch between mono colors (one color for agents, one for orders) or different colors for each agent and its orders. User may also hide or highlight chosen agents.

6 Plan validation

Goal of validation is to create segments of pictures corresponding to the paths of the agents in given time frame in such way, that any human controller may easily verify that the plan is non colliding.
Validation starts after user click Validate Plan button and confirms. Please do not interrupt the process. Non conflict path segments are extracted one by one, screenshot and exported to pdf. Interrupting during this process will lead to corrupting the final pdf.
The plan is exported to the PlanValidation.pdf file.
Once the validation process is completed, the Visualisation window is shown again and user may finish viewing the plan.