אוניברסיטת בן גוריון **הפקולטה למדעי ההנדסה**

המחלקה להנדסת מערכות מידע

372-2-5212 תאוריות ויישומים של שיתוף פעולה בין מכונות

תשע"ז - סמסטר אביב

Lab Assignment 1 - MAPF

Submission deadline: 11.04.17

Introduction

In this assignment we will focus on executing the code of several solvers on various MAPF domains.

We will use 3 main solvers:

- A* variation called EPEA* (Enhanced partial expansion A*). To learn more about this variation you can read this paper.
- Conflict-based search (CBS). Details are available in this paper.
- MA-CBS(5), a variation of CBS from the same paper as CBS.

These solvers can run on several domains, we will focus on two domains:

- Grid world – a basic n*n grid with possible obstacles and changing number of agents.
- Dragon Age maps – maps taken from the RPG game (Playing the game is not part of the course syllabus and will not be accounted for extra points).

Preliminaries

- The code is available to download from: https://www.dropbox.com/sh/twmks6amoit7jsw/AACaNxF5xSEpdShUkE5ZEkYWa?dl=0
- The code is written in on C#, so you are requested to download Visual Studio. As students in our department, you can download it for free from here using your BGU email and password. Another option is to work in the labs.

Running Instructions

When Visual Studio is installed, double-click on the file “CPF-experiment.csproj” will open the project. Important places in the code:

Program.cs:

contains the domain definitions. Look at the function Main() -

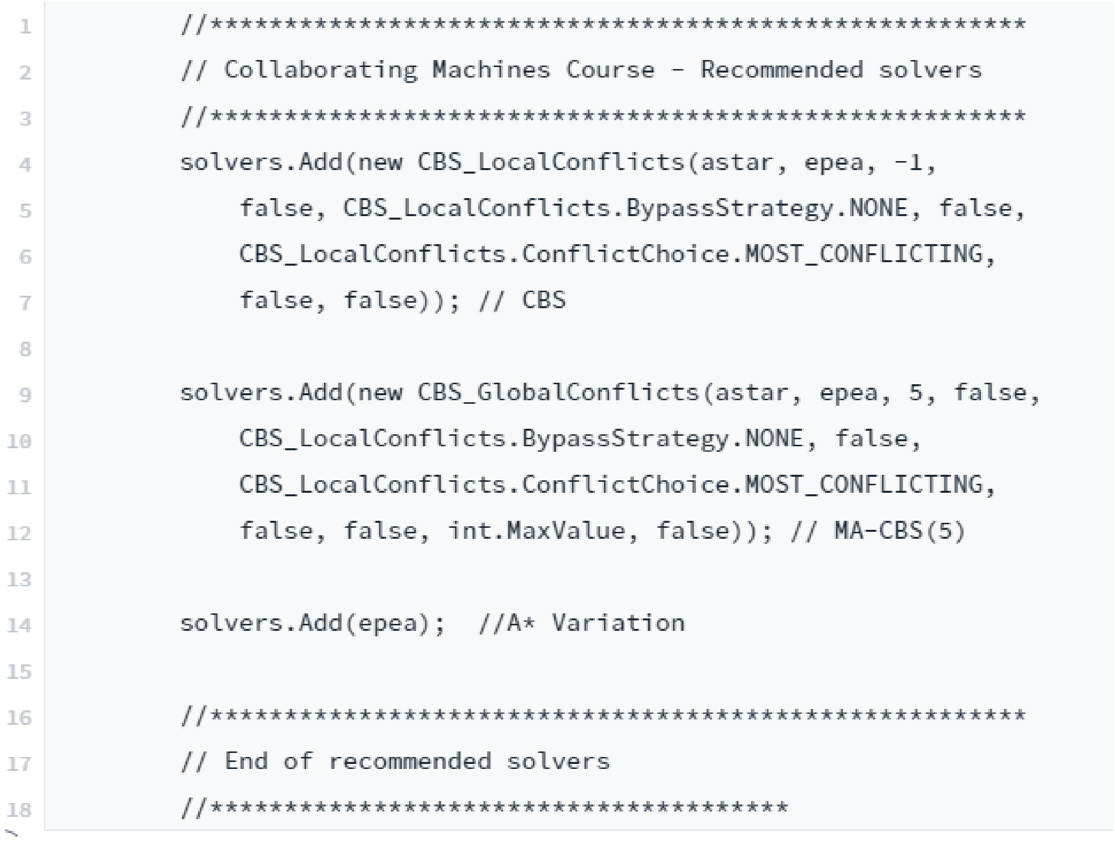
```
        bool runGrids = false;
        bool runDragonAge = false;
        bool runMazesWidth1 = false;
        bool runSpecific = false;
```

these are the lines where you decide which domain types to run. We recommend to work with *Grids* and *DragonAge*. For each domain, you can set the different parameters in the relevant if clause.

```
        if (runGrids == true)
        {
      // ...
        }
        else if (runDragonAge == true)
            me.RunDragonAgeExperimentSet(instances, Program.daoMapFilenames);
            // Obstacle percents and grid sizes built-in to the maps.
```

Run.cs:

Specifically, the function Run() is the function that is responsible for choosing the solvers that will run. Again, choosing X solvers will multiply the number of possible runs by X. Each time you see a line “solvers.Add(…)”, it means a new solver is added to the list of solvers that will be used. We recommend on using three solvers, which are the commented ones starting from line 153:



```
        //*****************************************************
        // Collaborating Machines Course - Recommended solvers
        //*****************************************************
        solvers.Add(new CBS_LocalConflicts(astar, epea, -1,
            false, CBS_LocalConflicts.BypassStrategy.NONE, false,
            CBS_LocalConflicts.ConflictChoice.MOST_CONFLICTING,
            false, false)); // CBS

        solvers.Add(new CBS_GlobalConflicts(astar, epea, 5, false,
            CBS_LocalConflicts.BypassStrategy.NONE, false,
            CBS_LocalConflicts.ConflictChoice.MOST_CONFLICTING,
            false, false, int.MaxValue, false)); // MA-CBS(5)

        solvers.Add(epea);  //A* Variation

        //*****************************************************
        // End of recommended solvers
        //*************************************
```

After you set your domains and solvers, run the program.

For each instance that was finished, the output will be printed to a console called d0.exe.

Important Notice:

1. In all domain definitions, if you state more than one option, it will *run all possible combinations*. For example, if you chose Grid with {2, 3, 4} agents and grid sizes of {3, 4, 5}, you will have 9 runs in total in this batch: 2 agents on a 3*3 grid, 2 agents on a 4*4 grid, etc…

2. If a problem was already solved, the solver *will not run again.* In order to make a solver run on a problem that was set before, go to:

   <projectFolder>\bin\Debug\Instances

   and delete the relevant instance from there.

Task list

For each of the configurations defined below, run both EPEA* and CBS:

1. 8*8 grid with 3 agents, 0% obstacles.
2. 8*8 grid with 5 agents and 5% obstacles.
3. 8*8 grid with 10 agents and 5% obstacles.
4. 8*8 grid with 15 agents and 5% obstacles.
5. dragonage map no. den502d with 5 agents.
6. dragonage map no. den502d with 10 agents.

For each configuration, write down:

7. Runtime in milliseconds
8. Cost
9. Number of Generated and Expanded nodes
10. Number of steps in the solution path.
11. For CBS only – what is the maximal group size?

Submission Instructions

Submission is individual, joint submissions will not be graded.

Submit a single file, named by your ID.

The file should contain two tables, one for each algorithm, using the format below or a similar format in excel or google sheets (ignore the values in the example).

Questions regarding the assignment and result files should be submitted to reuth.ise@gmail.com.

Good Luck!

Problem description	**Runtime**	**Cost**	**Generated/Expanded Nodes**	**Solution Path**	**Maximal Group Size (CBS only)**
8*8 grid with 3 agents, 0% obstacles	0.01	20	3/11	9	2