CSE 571 Spring 2020 Team Project Description

General guideline:

- Team projects are open-minded—you have a lot more freedom than individual projects here; at the same time, this means more researching: more reading, exploration, potential for errors, and debugging of course. **Deadlines is May 1st. You have slightly more than a month to complete the project.**
- You may consult online resources but do NOT copy.
- For your report, use the IEEE transactions template (Transactions articles) [https://journals.ieeeauthorcenter.ieee.org/create-your-ieee-journal-article/authoring-tools-and-templates/ieee-article-templates/templates-for-transactions/]. Report should not exceed 4 pages, ***not*** counting references. Recommended organization: 1) abstract and introduction: ***0.75*** page; 2) Technical approach: ***1*** page; 3) Results, analyses and discussions: ***2*** pages; 4) Conclusions and discussions: ***0.25*** page.
- Project evaluation will be done on ***a comparative basis (except for self-proposed ideas)***: *meaning that you will be compared with others choosing the same project topic.* This is meant to encourage exploration.
- Each team must select a topic by ***March 29 by sending an email to me AND your TAs with the title <CSE 571 S20: Project enrollment: Team (ABC) - topic X>, where ABC is a team name you choose and X is the topic number. Please include the names of your team members in the email. You will NOT be able to change your topic later so take time to decide.***
- Team size is ***3 or 4***. If you could not find more people to join your team by the deadline, please go ahead sending your project enrollment email. In which case, we may combine teams with 1-2 people to form larger teams. You then need to negotiate among yourselves to choose a topic to work on (you have a second chance to choose from any topic if you are combined) and send us a new email once it is decided.
- If you feel like proposing your own ideas, see instructions below.

Project topic 1. Bi-directional search

a) Implement bi-directional search describe in the following paper: "*Bidirectional Search That Is Guaranteed to Meet in the Middle*", Robert C. Holte, Ariel Felner, Guni Sharon, Nathan R. Sturtevant, AAAI 2016, and ***integrate*** into the Pacman domain for path-finding problems (from start to a fixed goal location) in your individual project 1 (http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/download/12320/12109)

b) Compare the algorithm's performance with other search methods (BFS, DFS and A*) that you have already implemented in environments *of* ***different sizes and complexities***. Provide ***statistical analyses*** for your results.

c) Submit a written report with your findings, which should include *at least the following*: 0. Abstract; 1. Introduction (motivation and your achievements in this project); 2) Technical approach (a brief technical discussion of the bi-direction search method you implemented); 3) Results (***Results that CLEARLY illustrate the strengths of your approach compared to others in a statistically meaningful way***, for example, you could compare the number of nodes expanded, computational time, etc.). 4) Conclusions (any observations and discussions); 5) Team effectiveness (notes about team member contributions).

d) Submit your code with ***comments, and with instructions*** (as a README file) to run it.

Project topic 2. Life-long planning

a) Implement lifelong A* search described in the following paper: "*D* Lite*", Sven Koenig and Maxim Likhachev, AAAI 2002, and ***integrate*** into the Pacman domain for path-finding problems (from start to a fixed goal location) in your individual project 1 (http://www.aaai.org/Papers/AAAI/2002/AAAI02-072.pdf). 1) Assume that Pacman only knows about the size of the (grid-world) environment initially, and can ***only observe local environment*** surrounding itself (you may define what local environment is). 2) Also, assume that Pacman always ***knows where it is*** in the environment (i.e., it can localize). 3) Once the Pacman observes something, it is able to ***keep it in its mind*** (i.e., it maintains the knowledge that there is an obstacle in a given location once that is observed).

b) Compare the algorithm's performance with *an A* baseline that simply replans every time when new obstacles are observed* in environments *of* ***different sizes and complexities***. Provide ***statistical analyses*** for your results.

c) Submit a written report with your findings, which should include: 0. Abstract; 1. Introduction (motivation and your achievements in this project); 2) Technical approach (a brief technical discussion of the life-long planning method you implemented); 3) Results (***Results that CLEARLY illustrate the strengths of your approach compared to others in a statistically meaningful way***, for example, you could compare the number of nodes expanded, computational time, etc.). 4) Conclusions (any observations and discussions); 5) Team effectiveness (notes about team member contributions).

d) Submit your code with ***comments, and with instructions*** (as a README file) to run it.

Project topic 3: Games & theory of mind

a) The expectimax game was discussed in class, including how an adversary can be modeled as an expectation node, instead of a min node in minimax. *Based on your individual project 2*, ***verify that the optimal reasoning level is one level beyond your opponent in the Pacman domain (with one Pacman and one ghost)***. For simplicity, you may assume that the agents are always using minimax up to a specific level. Furthermore, you may temporarily assume that the agents A1) with ***0.8*** probability choose actions according to the minimax solution, and with ***0.2*** probability choose a random action. A2) Also, you can assume that the two agents ***share the same evaluation function***.

b) ***Reevaluate the conclusion*** when the two assumptions (A1 & A2) made above no longer hold. ***In both cases***, you must compare the performances of your agent when the ***adversary is reasoning at different levels, and when your agent is overthinking or underthinking***. You must provide ***statistical analyses*** for your results.

c) Submit a written report with your findings, which should include: 0. Abstract; 1. Introduction (motivation and your achievements in this project); 2) Technical approach (a brief technical discussion of the different agent settings you implemented to evaluate the conclusions); 3) Results (***Results that CLEARLY support your conclusions in a statistically meaningful way***.). 4) Conclusions (any observations and discussions); 5) Team effectiveness (notes about team member contributions).

d) Submit your code with ***comments, and with instructions*** (as a README file) to run it.

Project topic 4: Turning logic agent to probabilistic reasoning agent

a) Enable your agent to ***reason about uncertainty using Bayesian networks*** via the pgmpy library [https://pgmpy.org]. This assumes that your agent has noisy sensors (you may assume and implement whatever noisy sensor models that are ***reasonable***). ***Update your hybrid agent to reason probabilistically and plan accordingly***: your agent will no longer be relying on logic reasoning. **Explain how your agent plans in the report.**

b) To complete this project, the ***prerequisite*** is that you ***know the basics of Bayesian network*** [Textbook Chapter 14], and how to convert the Wumpus domain into a probabilistic domain. ***Compare the performance of your hybrid logic agent and hybrid probabilistic reasoning agent*** with noisy sensor readings (*with the logic agent assumes that the readings are always accurate*). You must provide ***statistical analyses*** for your results.

c) Submit a written report with your findings, which should include: 0. Abstract; 1. Introduction (motivation and your achievements in this project); 2) Technical approach (a brief technical discussion of the probabilistic domain and Bayesian networks that you used); 3) Results (***Results that CLEARLY support your conclusions in a statistically meaningful way***.). 4) Conclusions (any observations and discussions); 5) Team effectiveness (notes about team member contributions).

d) Submit your code with ***comments, and with instructions*** (as a README file) to run it.

Project topic 5. Reinforcement Learning agent in Wumpus world

a) ***Implement and integrate any chosen reinforcement learning*** method to control your agent in the Wumpus world with noisy action model (you may assume and implement whatever noisy action model that is ***reasonable***). This turns your logic agent into a learning agent.

b) To complete this project, the ***prerequisite*** is that you ***know the basics of reinforcement learning*** [Textbook Chapter 21]. Try varying the noisy action model to generate different behaviors. ***Compare the performance of your hybrid logic agent with reinforcement reasoning agent,*** under a noisy action model (*with the logic agent assumes that there is no noise in the action model*). You must provide ***statistical analyses*** for your results. You must also provide ***an analysis of convergence*** for your learning agent (i.e., how fast it converges to the optimal policy).

c) Submit a written report with your findings, which should include: 0. Abstract; 1. Introduction (motivation and your achievements in this project); 2) Technical approach (a brief technical discussion of how you formulate the problem into a reinforcement learning problem); 3) Results (***Results that CLEARLY support your conclusions in a statistically meaningful way***.). 4) Conclusions (any observations and discussions); 5) Team effectiveness (notes about team member contributions).

d) Submit your code with ***comments, and with instructions*** (as a README file) to run it.

Self-proposing topics: Let me explore myself!

a) If you would like to propose your own project idea, **please schedule an appointment with me for your team**. It would work better if you can prepare a 1-page proposal to be discussed and send it to me at least 24-hours before the appointment. All such appointments must be scheduled before ***March 27*** *(in a week)*. ***All proposed ideas must be connected to our class materials.***

b) Submit a written report with your findings, which should include: 0. Abstract; 1. Introduction (motivation and your achievements in this project); 2) Technical approach (a brief technical discussion of your method); 3) Results (***Results that CLEARLY support your conclusions in a statistically meaningful way***.). 4) Conclusions (any observations and discussions); 5) Team effectiveness (notes about team member contributions).

c) Submit your code with ***comments, and with instructions*** (as a README file) to run it.