Ethan Howe
16.412/6.834J Problem Set #1

Part A:

1) Neural Network Design: I know the basic ideas and methods of neural networks but I am interested in how the layout of such networks effects their performance. There has been research into actively restructuring neural networks during learning to achieve better characterization of the goal. I would like to learn more about research into understanding neural structures in animals applied to mimicking particular functions in artificial networks. This would also involve learning about recurrent nets. These methods would be useful to robotics in performing motor skills, reflexes, and recognition. For these tasks, hardware solutions for neural networks would also be useful as a part of intercommunicating robotic system.
2) Dynamic Bayesian Networks: I am interested in DBNs for representing hidden Markov model problems and their usefulness in exploiting sparseness. I would like to learn methods for utilizing DBNs as general constructions. Applications of DBNs to predicting the likelihood of different outcomes from a robot's actions. It would also be interesting to understand how adequate networks are constructed and automated techniques. These topics would be useful in complicated production robots were the same models can be reused many times.
3) Sub-optimal planning methods: I would like to learn about advanced planning methods such as FF and LPG. These tools are in many systems and would be useful to build upon. In the process of learning these methods, we could also summarize basic traits of state-action planning. Planning applications are becoming widespread in commercial systems and so would be useful.

Part B:

I would like to design a large-scale system for robotic building construction. I would design a team of robots to complete construction tasks either assisting human construction workers or independently. Different robots would be needed for operating large scale machinery such as bulldozers and cranes, organizing building materials for others to use, and doing intricate small scale tasks. The system would also most likely have a foreman host to oversee planning, execution, and negotiation. Robots would have to communicate with each other locally to avoid danger, indicate the location of resources, and cooperatively complete tasks. The system would be handed detailed blueprints including descriptions and images of materials. Robots would share an accurate up-to-date map of the site over a reliable network and could localize through both GPS and visual means.

Three components important to this system would first be establishing a top-level plan for each component of the robot team to achieve intermediate goals. The planning system would establish landmarks and sub-goals in order to break down the search space of the construction into manageable, logical pieces. Tasks performed by each robot could be broken down and described by the developers so that the system has a model of what possible operations each robot can perform, how long it will take, and materials they will need. These parameters could be stored in Markov decision processes or dynamic decision networks so that they could actively take account of differences in environment and individual circumstances (such as a board slipping slightly). The robots would also need advanced visual skills to make sure that tasks were completed within very small error of the plans. Stereoscopic vision, 3-D models, pose estimation, and feature recognition would all be used to pick up, position, and assess the correct use of a wide variety of building materials. When two robots are required to complete a task such as one holding a board while the other nails it in, they would have to share data and negotiate methods and timing. The distributive architecture would optimize the cost for each in pursuit of the common goal.

Part C:

Specialized robotic systems are already in use in production environments. Manufacturing robots handle materials and complicated assembly. Storage facilities use robotic assistants to transport goods in a controlled setting. Medical robots have sensitivity and accuracy greater than their human operators. My proposed system would have to integrate the skills in a novel way but most of the pieces are already there. It is the planning over distributed agents and on a project with nearly infinite degrees of freedom that is most important in making the system capable. The overall foreman computer would have to know what objects to put where and in what order using what tools. It would have to break down tasks and distribute them efficiently. Establishing landmarks of the process will be critical and hopefully easier in the space of construction jobs.

Part D:

Hoffmann, J., Porteous, J. and Sebastia, L. (2004) "Ordered Landmarks in Planning", JAIR, Volume 22, pages 215-278.

I selected this paper because as I suggested above breaking down the building a structure into logical steps seems very natural and would undoubtedly simplify the planning of such a large scale objective. The paper builds on Hoffmann's earlier work on reasonable orders for top level goals and extends and adapts it to the task of reasonable orders for landmarks. The paper goes through all the steps of identifying candidates and verifying landmarks, assessing (obedient) reasonable orders, and using them break down the search space into sub-tasks for planners to handle. The results are then compared among ordering and non-ordering of each search.

The paper was easily readable with a limited amount of jargon, good step-by-step definitions, and examples of challenging concepts. Not only does it give a good background and theory for their addition to path planning, it goes through implementation, effectiveness assessment, and future work to be done. The paper's weaknesses include the substantial length needed to go through everything and the fact that the method does not stand on its own and the reader needs other advanced material to utilize it. Another topic only lightly treated in the paper is that of how best to resolve cycles in the ordering graph. Test were alluded to but none were detailed for this step in the algorithm. An almost excessive number of tests are detailed such that you really feel the authors surveyed a broad cross-section of planning problems. While this is good for understanding the improvements made, it can also be deceptive as to what types of planning are left out and being over confident in the methods positive effects.

Hoffmann, J. and Nebel, B. (2001) "The FF Planning System: Fast Plan Generation Through Heuristic Search", JAIR, Volume 14, pages 253-302.

I selected this paper to compliment the previous paper and get a good basis for the actually planning techniques needed to solve the proposed problem. The paper introduces a method for planning that puts together known pieces in a novel way and adds certain performance tweaks. FF planning is a heuristic search planner, HSP, variant that uses GraphPlan to generate a heuristic based on relaxed paths to the goal and enforced hill-climbing search along with a couple specific pruning methods. The GraphPlan algorithm was shown to run in polynomial time for relaxed search and heuristics of No-Ops first, minimal difficulty, and action set linearization for the backtracking phase.

The paper has many of the strengths of the above since the main author is the same but it is a bit less theoretical and has more of an applications motivated feel. For each portion of the algorithm, the authors looked at efficient implementation and while this added many small details, the total was an very effective search. One of the weaknesses in the FF method described is its over reliance on benchmarks to motivate design choices. While this makes it effective in solving these sets of problems efficiently, in the future there may arise an important set for which FF performs horribly. To address this concern the authors say they are working on classifying what is the underlying trait FF works well on. I also question the authors' decision that upon failure of enforced best-first search all progress should be discarded rather than backtrack a bit and use another full heuristic search. They also described some methods for finding and avoiding dead-end states said they did not wish to use but could have considered using upon failure.

Ahmadabadi, M., and M. Asadpour. "Expertness Based Cooperative Q-Learning." *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* 32, no. 1 (February 2002).

This paper would be useful for taking account of unexpected circumstances arising in the real world context. The system would have to learn and adjust for individual and environmental differences. The text was meant to continue previous work on weighted strategy sharing (WSS) by the same authors. It reviewed the previous method and added more realistic conditions (communication uncertainty, false data) and more testing. This paper complements the other two by looking at the layer below action planning. It deals with effective action execution and learning. The robots experiences need to feed back to the way actions are carried out for instance if one of them is working slower than others it should figure into our heuristic for actions and this could be accomplished through expertness criteria.

The subjects were described in an easy to understand fashion with a minimum of implementation details. The paper assumed a background in such systems including knowledge of Q-Tables and how a learning system would be implemented in the first place. The methods described were for addition to such a multiple learning robot platform to take better advantage of cooperative learning. I found that the paper did not have much to offer in impressive results and the method was tested on only one benchmark. The authors also could have presented their results more understandably and also put them in the abstract. The organization and explanation of relative effectiveness could have been clearer emphasizing the most striking cases out of the large tests. Overall the results could be useful in that where positive reenforcement dominated taking the positive weight signals worked best and the same held for negative reenforcement and signals.

Part E:

For my project, I could design a system to extract actions and states from a full blueprint including all materials and positions for a portion of a building for a group of robots to constructed. The planner would break down sequences of events into necessary landmarks in the process and assign efficient tasks to sets of robots. The system could use some kind of knowledge set to construct the state network and then use the FF planning and reasonable orders detailed above. Then if time allows another layer could be added to the planning to take account of feedback and learning from the robots as the structure progresses. The robots actions would be simulated and have random discrepancies added.