Wishlist: Interesting Problems
Mark Velednitsky
September 10, 2017

Abstract
These are problems I would be interested in working on and/or knowing more about. If you want to work together, contact me! I enjoy collaboration. My email address is marvel@berkeley.edu.

1 Explainable Learning: RL for Hanabi
A classic challenge in RL is creating an agent which acts in an explainable way. The idea of this project would be to try to create a training regime where the interaction between agents forces them to play a simple, common strategy.

Hanabi is a great example of a game where two RL agents would end up developing an esoteric system that is totally incomprehensible. To break this, you could train two agents together (A and B) and another two agents (C and D), then have them play together across the pairs (AC, AD, BC, BD). This would force the pairs to learn a strategy that generalizes well.

2 TSP with Time Windows
This is a practical version of the TSP which incorporates scheduling and is relevant to sales representatives. In short, the Traveling Salesperson would like to find the shortest path between destinations such that their time spent in each destination falls into some time window (in practical applications, when their customer is available).

3 Subgraph Removal Game
Given a graph $G$ given a collection of graphs $H$, we play the following game. On their turn, a player may remove any subgraph from $G$ if it is in $H$ (meaning, remove all the nodes corresponding to the subgraph). The first player unable to do so loses.

Interesting cases:

- It is hard to even compute if a player has lost (subgraph isomorphism).
• Special case: remove any \( k \) nodes at a time. Easy.

• Special case: remove any \( k \) connected nodes at a time. Difficult to compute the winning strategy even when \( G \) is a line graph and \( k \) is 2. Can we formalize this?

• How about trees and bipartite graphs? Anything special about those?

• Are there connections to automorphisms?

4 Neural Recommendation Systems

Think about the way you recommend movies to someone. You typically have a sense of their tastes, especially which popular items they most enjoy. You make deep conclusions based on a few popular items. Maybe the key to good recommendations is to train NNs to predict the user ratings based on their opinion of a small set of popular movies.

5 Others

• Neural Combinatorial Optimization
• The Quadratic Assignment Problem (\( \min c_{ij}x_{ij}^2 \) subject to assignment polytope)
• Inverse Shortest Path
• Inverse Minimum Spanning Tree
• Maximum Cut
• Streaming Medians Problem