CSCE 52803: Graph and Combinatorial Algorithms

Fall 2026

Course Overview

- Course Description: Introduction to graph and combinatorial optimization problems, P and NP, approximation algorithms for solving NP hard problems as well as machine learning approaches for solving these problems.

- Goal: The goal of this course is for students to understand what are graph and combinatorial optimization problems, the difficulty in solving these problems, traditional and machine learning approaches, and finally equip the students with the skills to read the state-of-the-art research papers that use machine learning for solving these problems.

Class Hours

- Class hours MoWeFr 10:45AM - 11:35AM.
  - Location: JBHT 239
- Instructor – Lu Zhang
  - Email: lz006@uark.edu
  - Office: JBHT 522
  - Webpage: https://zh0007lu.github.io/
- Course Website
  - https://zh0007lu.github.io/course/2026fall/52803/52803.html
- Blackboard

Course Materials

- No required textbook.
- Reference materials:
  - Guichard, D. (2017). An introduction to combinatorics and graph theory. Whitman College-Creative Commons.
  - Vazirani, V. V. (2001). Approximation algorithms (Vol. 1). Berlin: springer.
  - Dive into Deep Learning, by Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola (2020)
    - Available online: https://d2l.ai/
  - L. Wu, P. Cui, J. Pei, and L. Zhao. (2022). Graph Neural Networks: Foundations, Frontiers, and Applications. Springer, Singapore.
    - Available online: https://graph-neural-networks.github.io/index.html
  - Richard S. Sutton and Andrew G. Barto (2018). Reinforcement learning: An introduction. A Bradford Book, Cambridge, MA, USA.

Course Prerequisite

- CSCE graduate standing
- Expect students should know
  - Algorithms
  - Linear algebra
  - Calculus
  - Probability and statistics
  - Machine learning background (preferred)

Grading

- Composition
  - Assignments 45%
  - Paper reading and discussion 25%
  - Final exam 30%

- The final class grade will be assigned according to the 10-point scale shown below. The grades may or may not be curved.
  - A 90 – 100%
  - B 80 – 89.9%
  - C 70 – 79.9%
  - D 60 – 69.9%
  - F < 60%

Assignments

- There will be 4 assignments designed to enhance understanding of the material taught in the course:
  - Exercise questions on graph theory.
  - Exercise questions on proving NP problems using polynomial reducibility.
  - Exercise questions on designing greedy algorithms and computing approximation ratios.
  - Exercise questions on graph neural networks, reinforcement learning, and learning-augmented algorithms.
- Late policy
  - 10% penalty for each day after the due date for up to 3 days late.
  - Assignment more than 3 days late should be submitted together with an explanation.
  - Weekends count as 1 day.

Paper Reading and Discussions

- A list of state-of-the-art research papers on solving intractable graph problems using machine learning will be provided.
- Each student will pick a paper to present and lead the discussion.
- Other students are also required to read the papers and participate in the discussion.

Final Exam

- Closed book: students ARE allowed one 8.5x11 page of notes and a calculator, but they are NOT allowed any other materials or other electric devices such as cell phones, smart watches, tablets, or computers.

Office Hours

- MoWe 2:00 – 3:00 PM.
  - Location: JBHT 522
- Or, other times by appointment.

Academic Dishonesty Policy

- As a core part of its mission, the University of Arkansas provides students with the opportunity to further their educational goals through programs of study and research in an environment that promotes freedom of inquiry and academic responsibility. Accomplishing this mission is only possible when intellectual honesty and individual integrity prevail. Each University of Arkansas student is required to be familiar with and abide by the University's ‘Academic Integrity Policy’ at honesty.uark.edu. Students with questions about how these policies apply to a particular course or assignment should immediately contact their instructor.

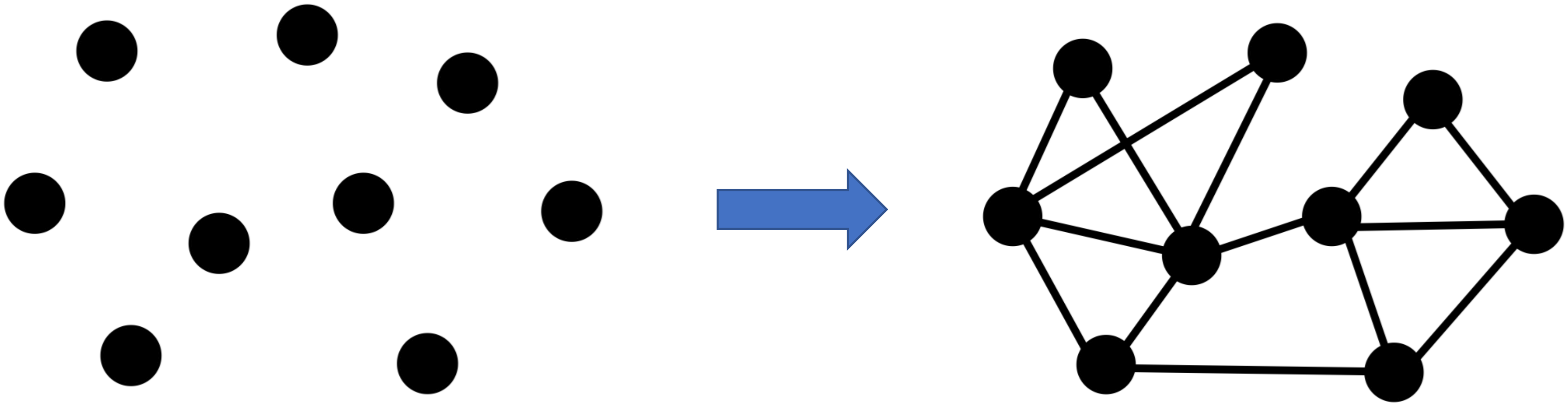
University Policies

- Academic Integrity
  - Refer to https://honesty.uark.edu/policy/
- Emergency Preparedness
  - Refer to http://emergency.uark.edu/
- Inclement Weather
  - Refer to http://safety.uark.edu/inclement-weather/
- RazALERT
  - Refer to http://safety.uark.edu/emergency-preparedness/emergency-notification-system/
- Academic Support
  - Refer to http://www.uark.edu/academics/academic-support.php

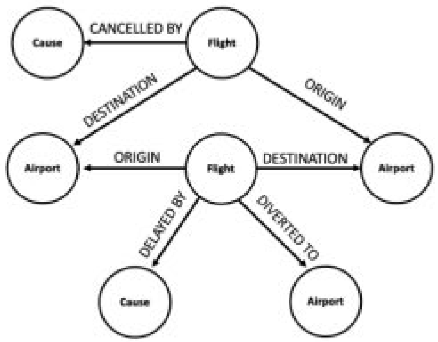
Introduction

What Are Graphs?

- Graphs are a general language for describing and analyzing entities with relations/interactions



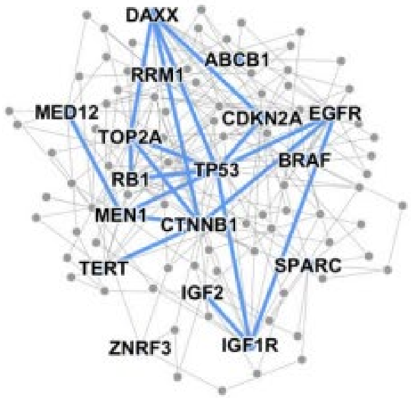
Many Types of Data Are Graphs (1)



Event Graphs

Image credit: SalientNetworks

Computer Networks



Disease Pathways

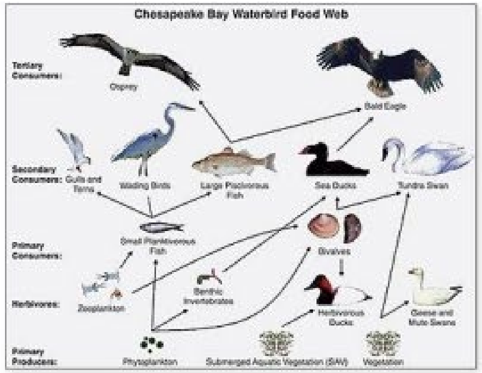


Image credit: Wikipedia

Food Webs

Image credit: Pinterest

Particle Networks

Image credit: visitlondon.com

Underground Networks

Many Types of Data Are Graphs (2)

Image credit: Medium

Social Networks

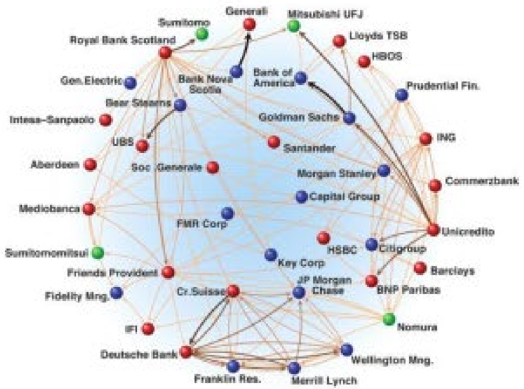
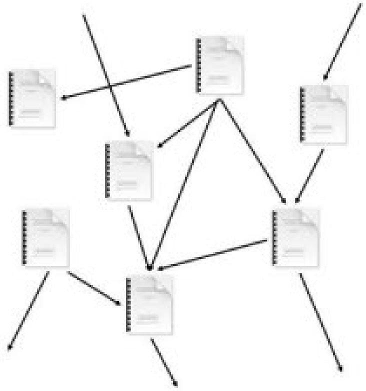


Image credit: Science

Economic Networks

Image credit: Lumen Learning

Communication Networks



Citation Networks

Image credit: Missoula Current News

Internet

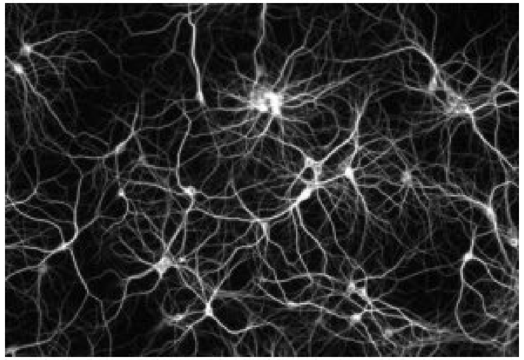


Image credit: The Conversation

Networks of Neurons

Many Types of Data Are Graphs (3)

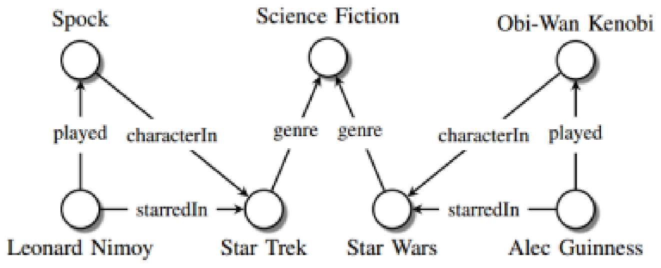


Image credit: Maximilian Nickel et al

Knowledge Graphs

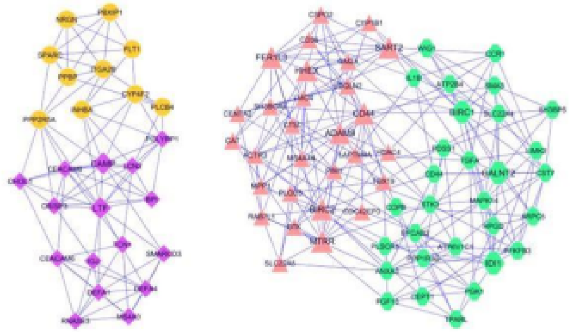


Image credit: ese.wustl.edu

Regulatory Networks

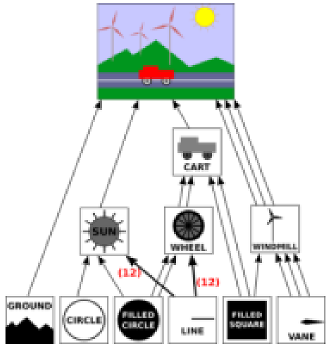


Image credit: math.hws.edu

Scene Graphs

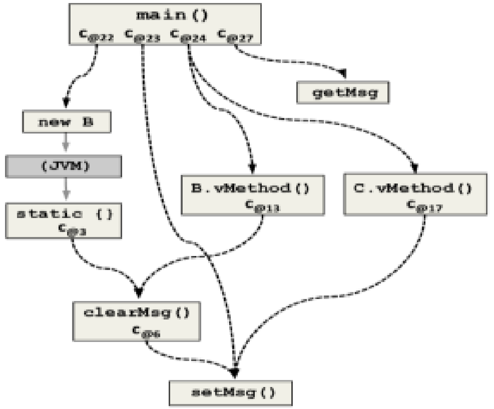


Image credit: ResearchGate

Code Graphs

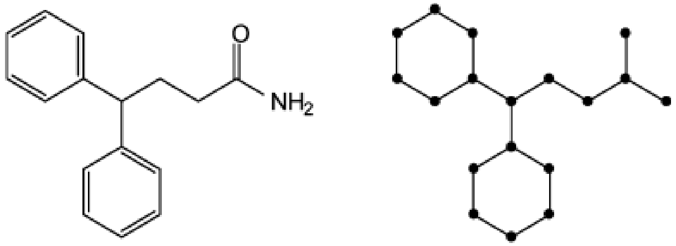


Image credit: MDPI

Molecules

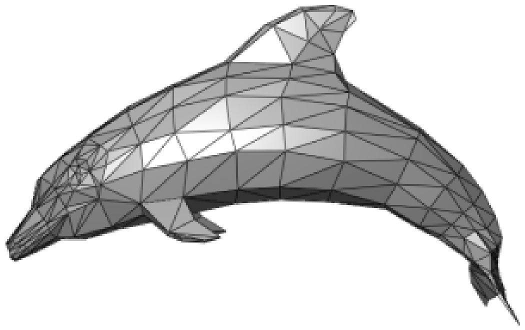


Image credit: Wikipedia

3D Shapes

Graph Theory

- A subfield in mathematics that studies graphs, which are mathematical structures used to model pairwise relations between objects.

What Is Combinatorial Optimization?

- Combinatorial optimization is the process of searching for maximum (or minimum) of an objective function F whose domain is a discrete but large configuration space (as opposed to an N-dimensional continuous space).

- Has many applications in operations research and computer science, with many critical real-world applications such as vehicle routing or scheduling.

Combinatorial Optimization Is Difficult

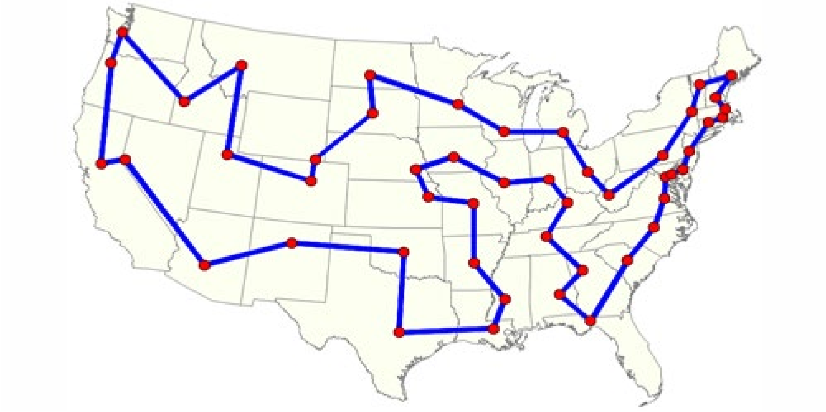
- CO problems are generally hard from a complexity theory standpoint due to their discrete and non-convex nature.

- Difficult to find optimal solutions and usually seeks for heuristics/approximate solutions.

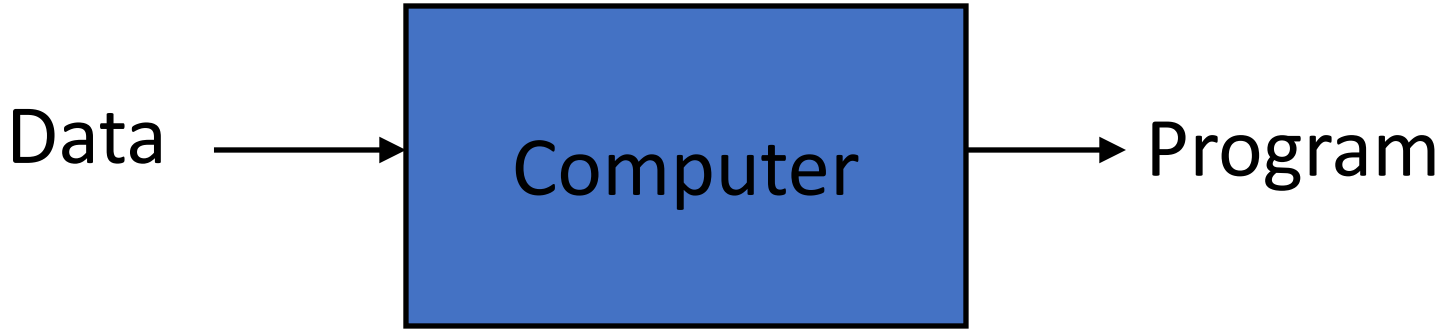
- Not all problems are equally difficult.

Graph and Combinatorial Optimization

- Due to the discrete nature of CO problems and the prevalence of network data in the real world, graphs are a central object of study in the CO field.
  - Among Karp's 21 NP-complete problems, ten of them are graph problems.
- Example: Traveling Salesman Problem (TSP)
  - Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?



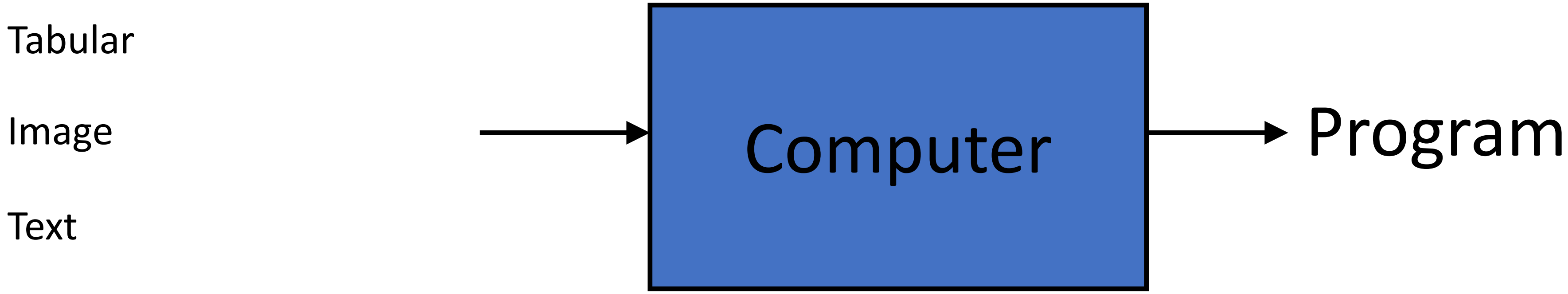
What Is Machine Learning?



- Can we use machine learning to generate programs for solving graph and CO problems?
  - Yes, but still underdeveloped

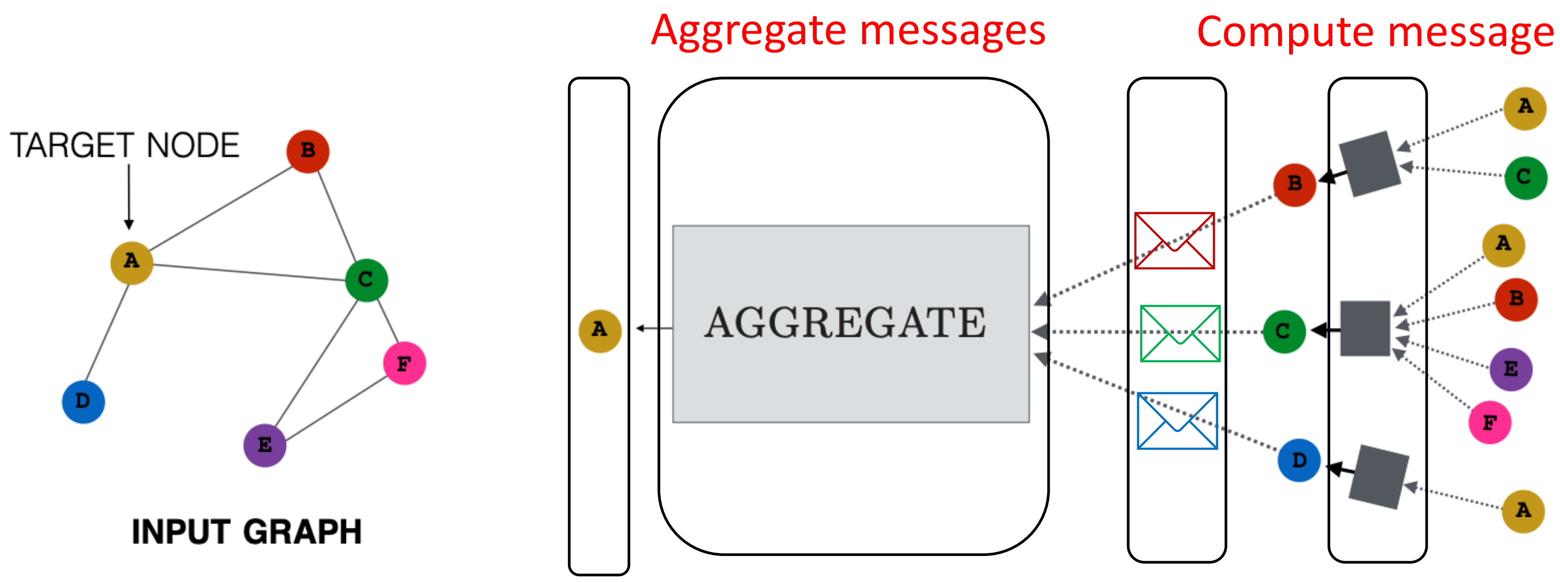
What Do We Need for ML?

- Explicit modeling of graphs



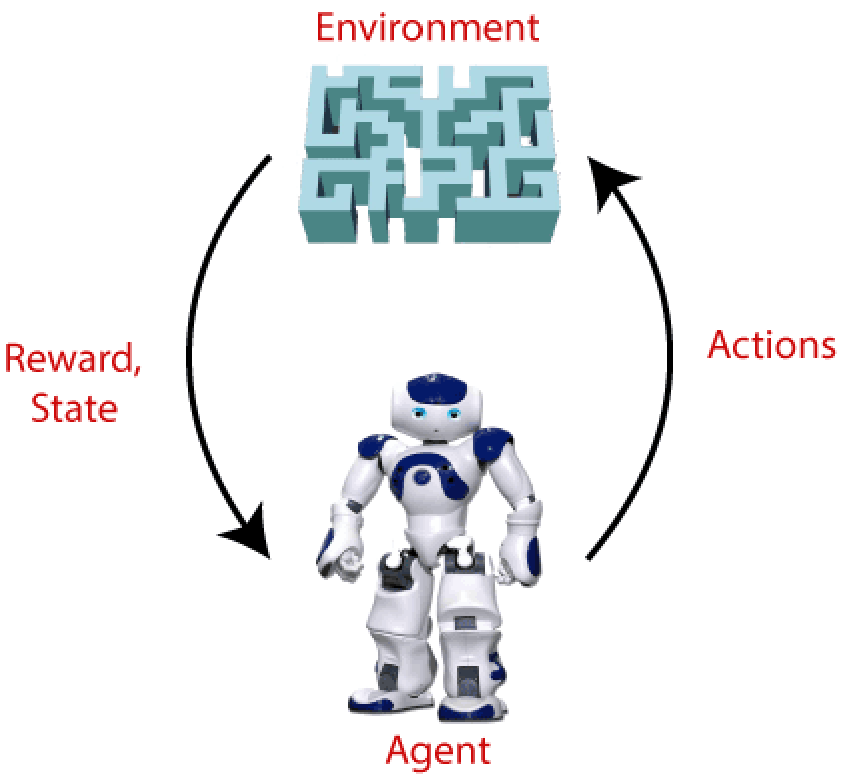
- How about graphs?
    - Graph neural networks

GNN Message Passing Paradigm



What Do We Need for ML?

- Modeling of heuristics/approximation algorithms
    - Reinforcement learning



Learning Augmented Algorithms

- An algorithm that can make use of a prediction to improve its performance.
  - Prediction of an optimal solution
  - Prediction of the input
  - Prediction of algorithmic actions
- Consistency, Smoothness, Robustness
- Has been recently applied to classic graph and combinatorial optimization problems, e.g., set cover, maximum cut, vertex cover

Topics Covered

- Introduction and preliminaries
- Tractable graph problems (shortest path, bipartite matching, spanning tree, Eular circuit)
- P and NP
- Approximation algorithms (greedy algorithm, LP relaxation and rounding, approximation ratio)
- Intractable graph problems (traveling salesman problem)
- Revisit of machine learning and neural network basics
- Learning augmented algorithms
- Graph neural networks
- Reinforcement learning basics
- Machine learning for graph problems