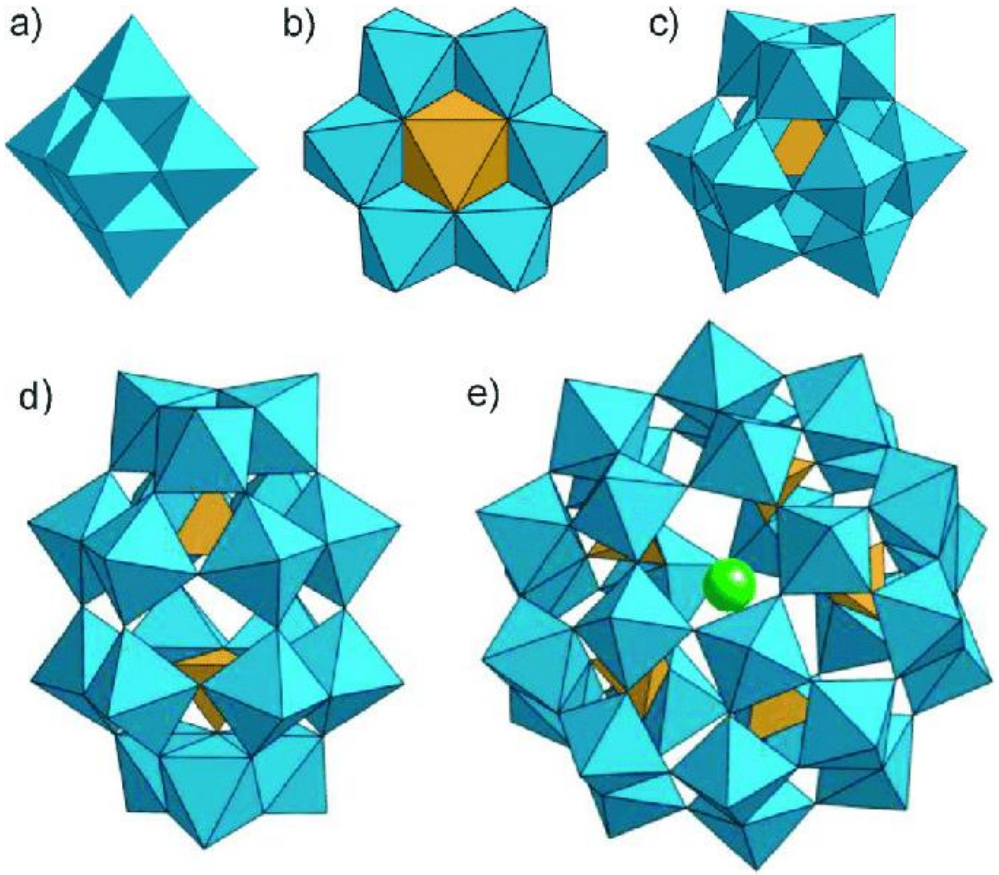
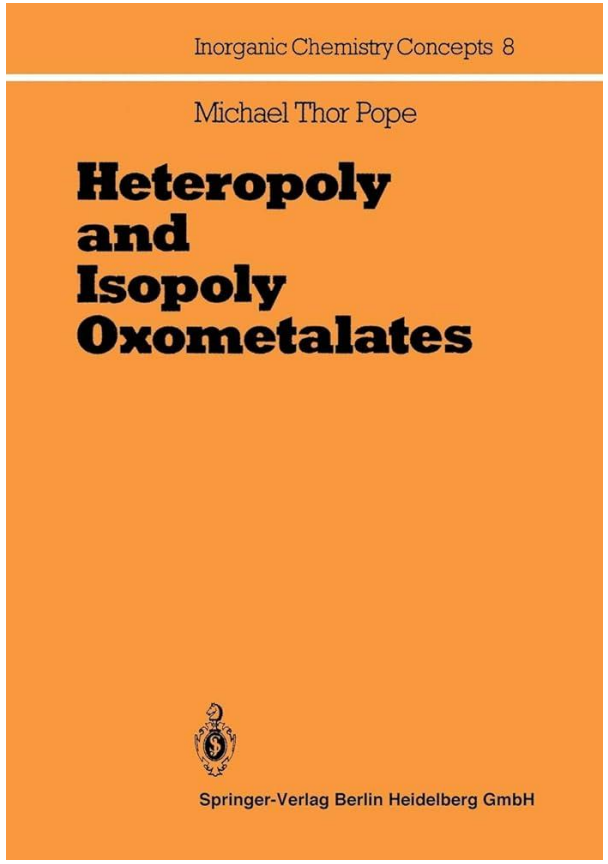
THE UNIVERSITY of TENNESSEE KNOXVILLE

Lecture 17: Decisions and Bandits

Instructor: Sergei V. Kalinin

Synthesis of polyoxometallates



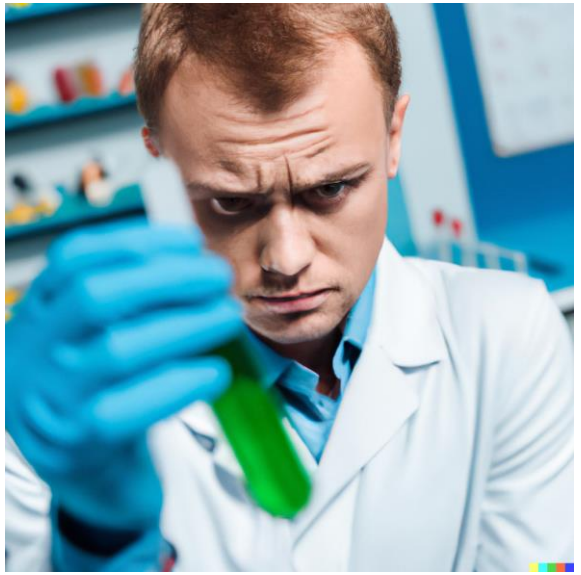


X. López, J. Carbó, C. Bo, J. Poblet, *Structure, properties and reactivity of polyoxometalates: A theoretical perspective*, Chemical Society reviews 71, 7537 (2012)

Synthesis of polyoxometallates

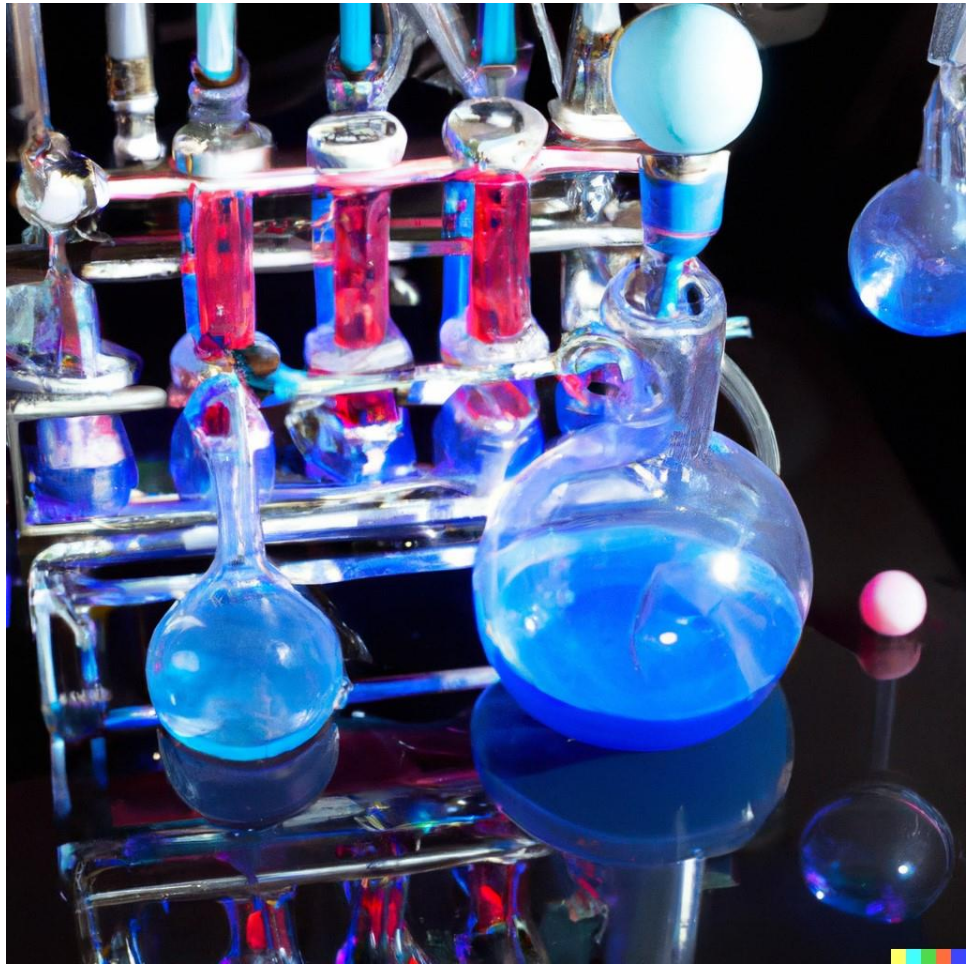
Good day

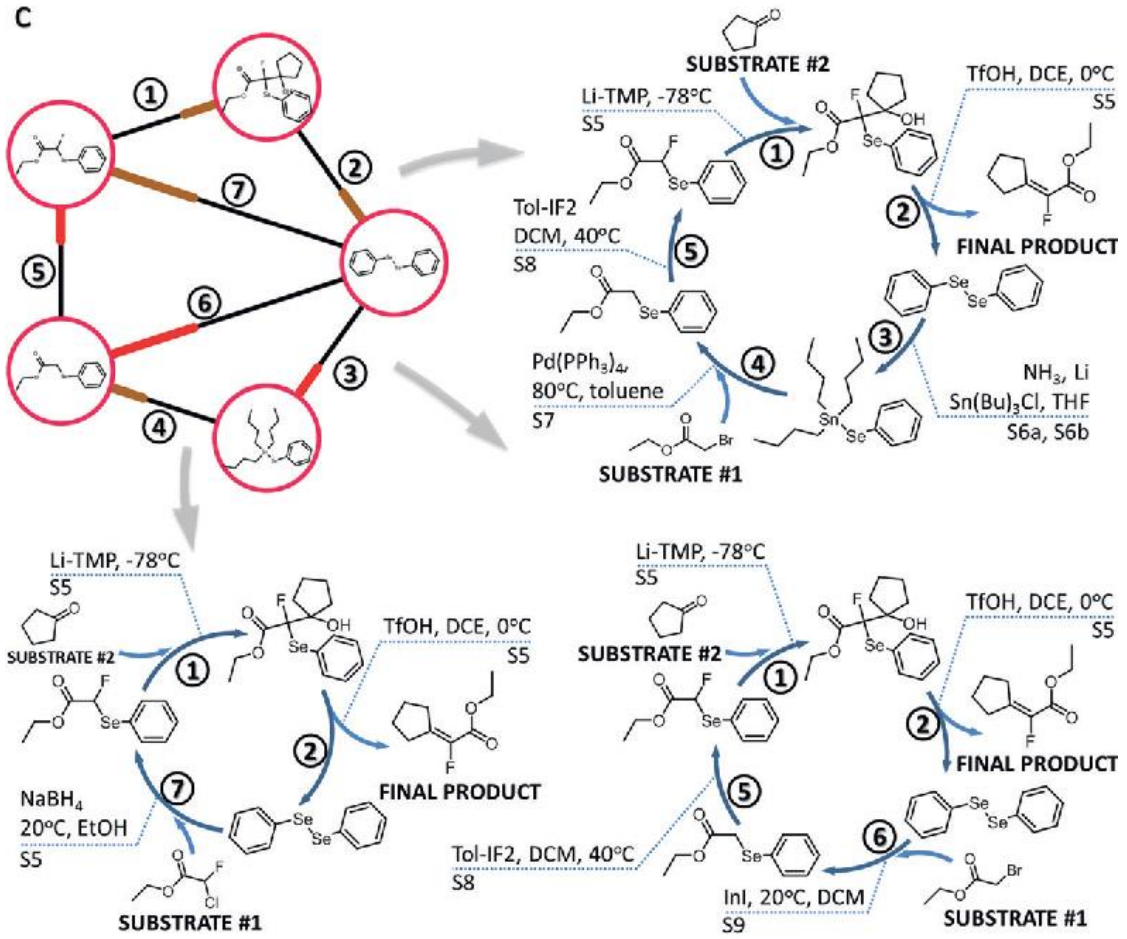
Bad day



- Synthesis may or may not work!
- Depending on difficult to quantify factors
- It’s a coin toss, a.k.a. Bernoulli distribution

Organic synthesis





M. Bajczyk, P. Dittwald, A. Wołoś, S. Szymkuć, B. Grzybowski, *Discovery and Enumeration of Organic-Chemical and Biomimetic Reaction Cycles within the Network of Chemistry*, DOI:10.1002/anie.201712052

Definitions:

- **Objective:** overall goal that we aim to achieve. Not available during or immediately after experiment.

- **Reward:** the measure of success available at the end of experiment

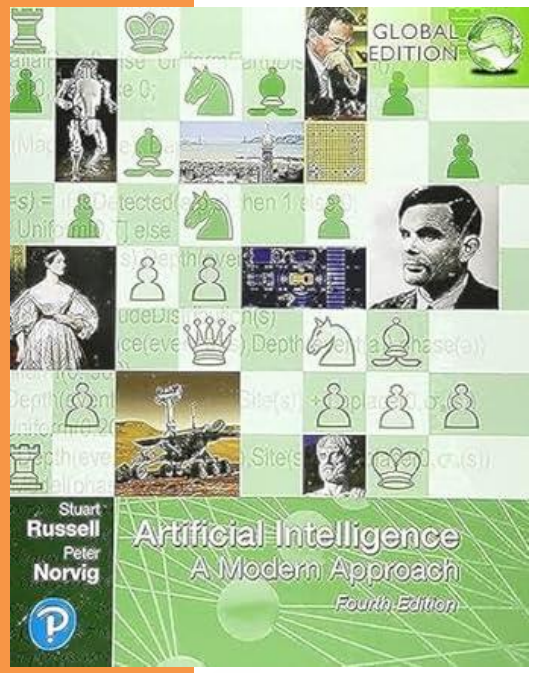
- **Value:** expected reward. Difference between reward and value is a feedback signal for multiple types of active learning

- **Action:** how can ML agent interact with the system

- **State:** information about the system available to ML agent

- **Policy:** rulebook that defines actions given the observed state

Where do reward come from?



Somewhat remarkably, almost all AI research until very recently has assumed that the performance measure can be exactly and correctly specified in the form of utility or reward function

Predictive power
Parsimony

Physics
- Analytical theory
- Computational experiments

Microscope optimization

- Structured GPs
- PINNs
- Neural operators

Learning structure-property relationships

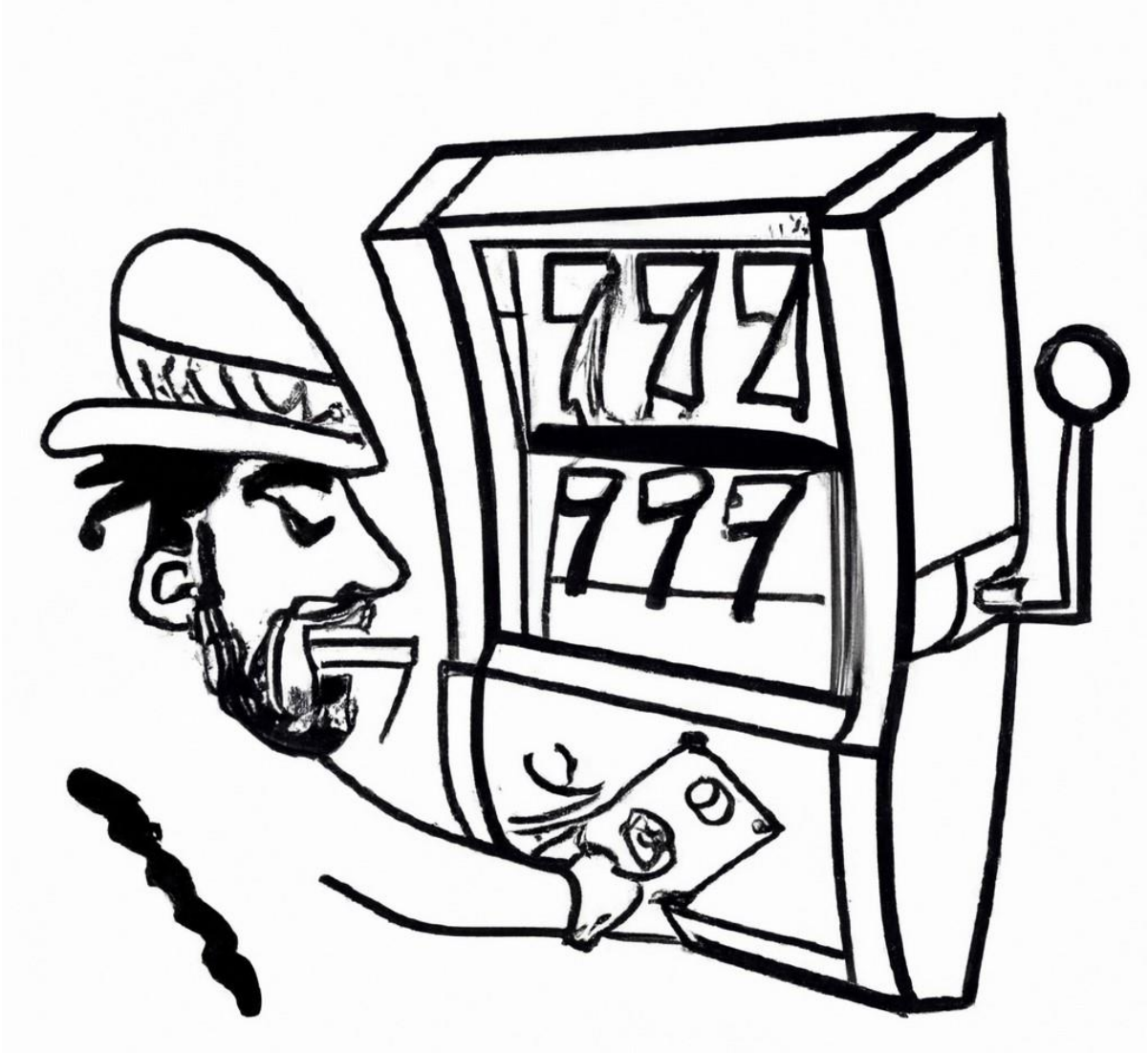
Materials discovery

Loss functions

Rewards
- Global warming
- Better storage materials
- …

Data:
- DCNNs
- LLMs
- And so on

Bandit problem



- Imagine that we have a number of slot machines with different probabilities of win…
- Or different web-sites to places ads on…
- Or groups of patients for specific medical protocol….
- Or team members to synthesize certain material…
- Or reaction pathways to choose

How do we optimize this problem and maximize our reward?

Bandits

- **Objective:** get rich!
- **Reward:** pay-off from specific hand/click-rate of ad/effectiveness of drug
- **Value:** expected reward
- **Action:** playing a hand/placing ad/administering drug
- **State:** no state
- **Policy:** gameplan given the values of specific actions

A/B Testing

- The most common exploration strategy is **A/B testing**, a method to determine which one of the two alternatives (of online products, pages, ads etc.) performs better.
- The users are randomly split into two groups to try different alternatives. At the end of the testing period, the results are compared to choose the best alternative, which is then used in production for the rest of the problem horizon.
- This approach can be applied for more then two alternatives - **A/B/n testing**.

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Definitions: reward and value

$$Q_n \triangleq \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$$

- First, we denote the reward (i.e. 1 for click, 0 for no click) received after selecting the action a for the n^{th} time by R_i.

- Q_n estimates the expected value of the reward that this action yields, R, after n-1 observations.

- Q_n is also called the action value of a. Here, Q_n estimates of the action value after selecting this action n - 1 times.

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Updating value

$$Q_{n+1} = \frac{R_1 + R_2 + \cdots + R_n}{n} = Q_n + \frac{1}{n} \cdot (R_n - Q_n)$$

- Q_n is the estimate for the action value of a before we take it for the n^{th} time.
- When we observe the reward R_n, it gives us another signal for the action value.
- We adjust our current estimate, Q_n, in the direction of the **error** that we calculate based on the latest observed reward, R_n, with a **step size** $1/n$ and obtain a new estimate Q_{n+1}
- For convenience, $Q_0 = 0$ (But - Human heuristics!)

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Updating value: generalization

$$Q_{n+1}(a) = Q_n(a) + \alpha(R_n(\alpha) - Q_n(a))$$

- The rate at which we adjust our estimate will get smaller as we make more observations
- The step size must be smaller than 1 for the estimate to converge (and larger than 0 for a proper update).
- Using a fixed α will make the weights of the older observations to decrease exponentially as we take action a more and more.

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Limitations of A/B testing

- **A/B/n testing is inefficient as it does not modify the experiment dynamically by learning from the observations.** It fails to benefit from the early observations in the test by writing off/promoting an alternative even if it is obviously under- or outperforming the others.

- **It is unable to correct a decision once it's made.** There is no way to correct the decision for the rest of the deployment horizon.

- **It is unable to adapt to the changes in a dynamic environment.** If the underlying reward distributions change over time, plain A/B/n testing has no way of detecting such changes after the selection is fixed.

- **The length of the test period is a hyperparameter to tune, affecting the efficiency of the test.** If this period is chosen to be shorter than needed, an incorrect alternative could be declared the best because of the noise in the observations. If the test period is chosen to be too long, too much money gets wasted in exploration.

- **A/B/n testing is simple.** Despite all these shortcomings, it is intuitive and easy to implement, therefore widely used in practice

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020