Combinatorial Optimization Challenge 2021

Yang Xikun

ZTE

2021.06.09

Table Of Contents

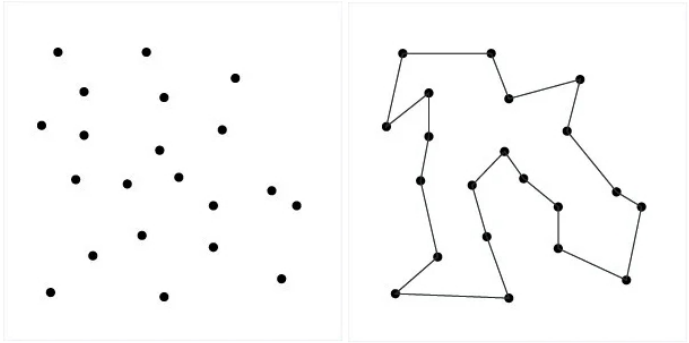
Combinatorial Optimization: An introduction

What is combinatorial optimization?

- Combinatorial optimization (CO) is:
  - A topic that consists of finding an optimal solution from a finite discrete set of solutions
  - A subfield of mathematical optimization
- It has important applications in real life.
  - Engineering and manufacturing
  - Transportation
  - Telecommunication
- A challenge for AI/ML and software engineering.

What is combinatorial optimization?

- Classical CO problems in real world:



Travelling Salesman Problem

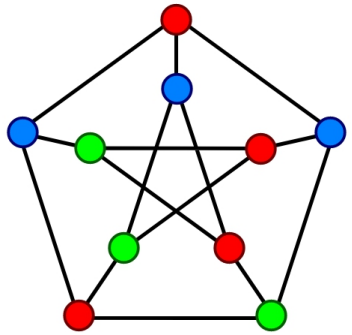
⬇

Circuit design

Bin Packing Problem

⬇

transportation



Graph Coloring Problem

⬇

Mobile Radio
Frequency Assignment

Difficulty of solving CO problems

Most problems are hard to solve.

- When the solution space is small, we can use some classical algorithms to get optimal solution. eg. Dynamic Planning.
- As the scale of the problem increases, the number of solutions will increase rapidly. It is almost impossible to traverse the entire solution space.

Scale of Travelling Salesman Problem

nodes (n)	5	7	10	100	...
Possible routes ((n-1)!)	120	5040	3628800	9.3e157	...

- We can only use heuristic algorithms to get an "approximate optimal solution". eg. Genetic Algorithm, Simulated Annealing Algorithm.
- Shortage: time consuming, low efficiency.

Deep Learning for Combinatorial Optimization

Why we use DL in Combinatorial Optimization?

- A high-dimensional problem solver
  - Combinatorial optimization problems are always company with many variables and constraints
  - DL model gives a approximate function with a large number of parameters, which can be optimized and get a good solution on most case of the dataset
  - Focus on inputs, outputs and modeling
- A learner not a searcher
  - Get 'knowledge' from data(supervised learning) or from reward(reinforcement learning)
  - Can be used in, or transferred to a series of similar scenes
  - Fast to get a solution

DL solutions for Classical Problems

- The solution to travelling salesman problem

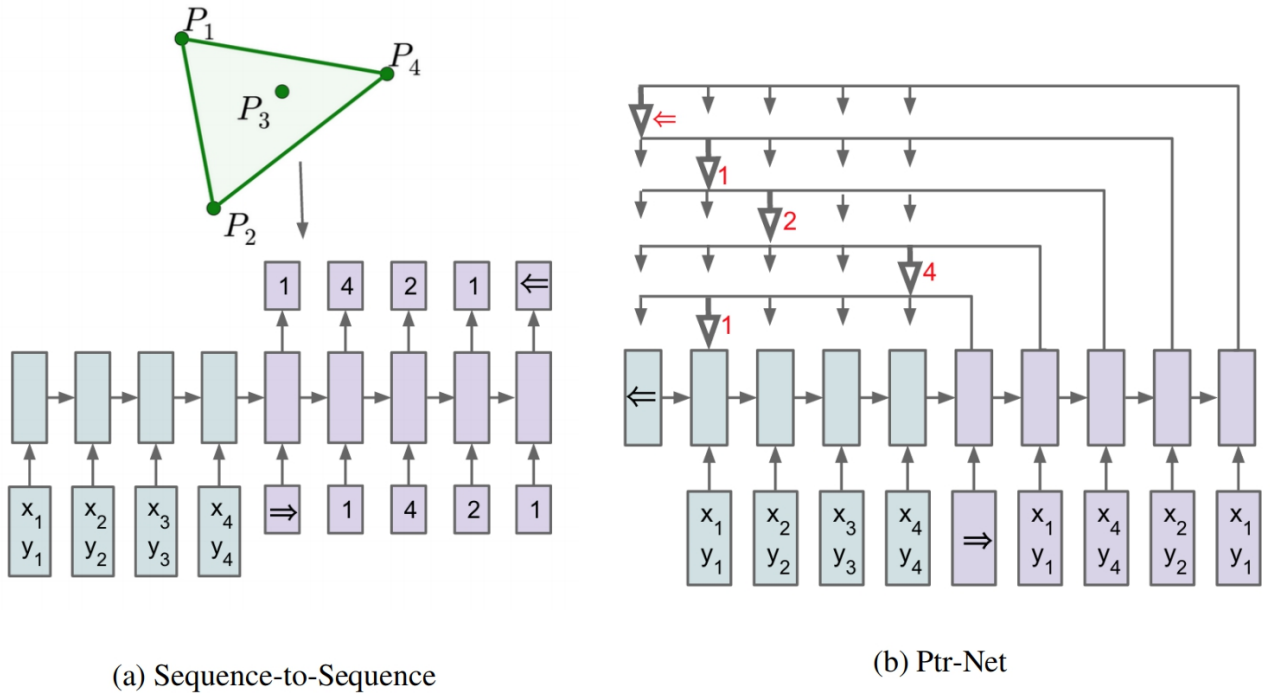
Pointer Network, which modifies seq2seq+attention model

Inputs: coordinates of city

Outputs: the shortest possible route(in order of cities) that visits each city exactly once and returns to the starting point

Supervised learning with a generated dataset

Problem scale: number of city <= 50



(a) Sequence-to-Sequence

(b) Ptr-Net

Oriol Vinyals, et al. Pointer networks. In Advances in neural information processing systems, pages 2692–2700, 2015.

DL solutions for Classical Problems

- The solution to 3D bin packing problem

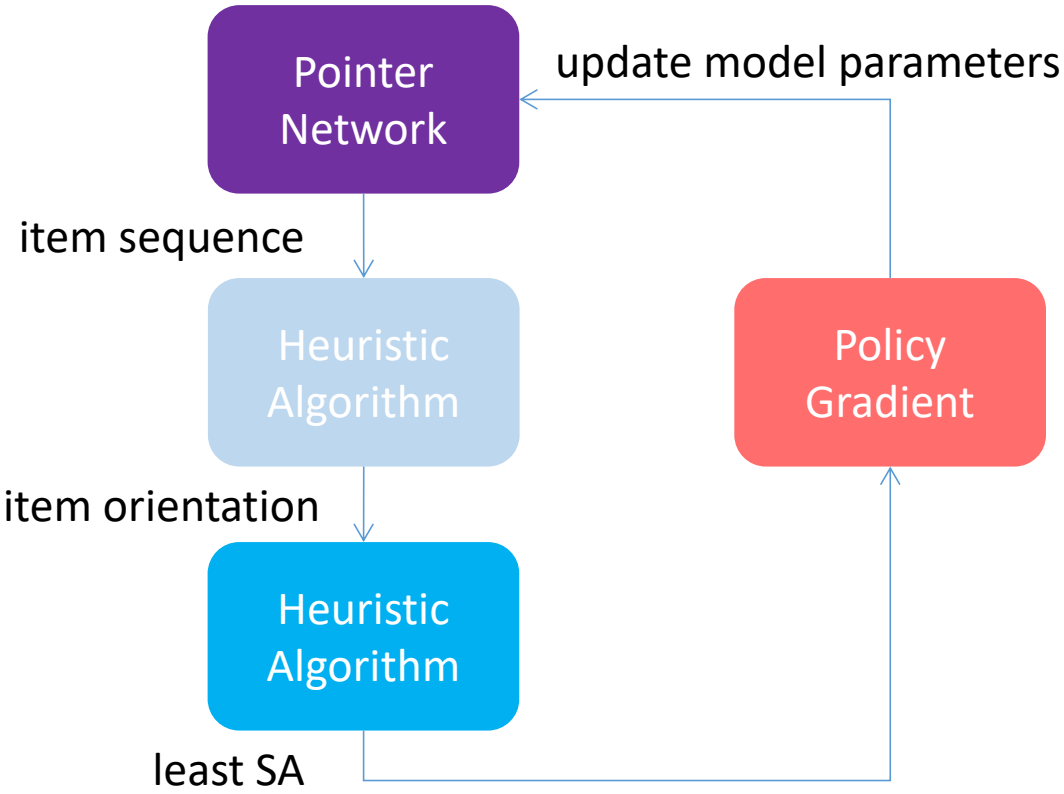
Difficulties:

- complex decision (order, orientation and position of items)
- many constraints (can not exceed the size of bin)

Target: minimize the surface area(SA) of bin

Method: reinforcement learning + heuristic algorithm

Item number: 8, 10, 12 (based on the number of items in one customer order)



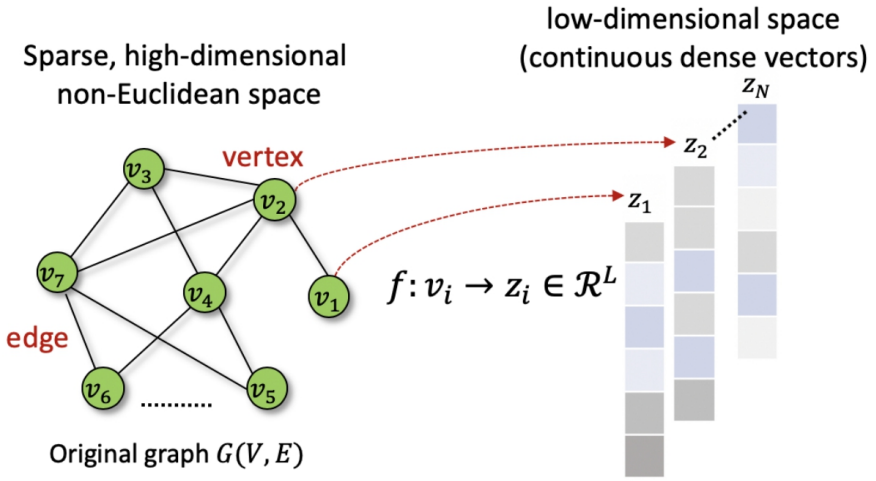
Haoyuan Hu, et al. Solving a new 3d bin packing problem with deep reinforcement learning method. arXiv preprint arXiv:1708.05930, 2017.

DL solutions for Classical Problems

- Useful tools for graph feature extracting

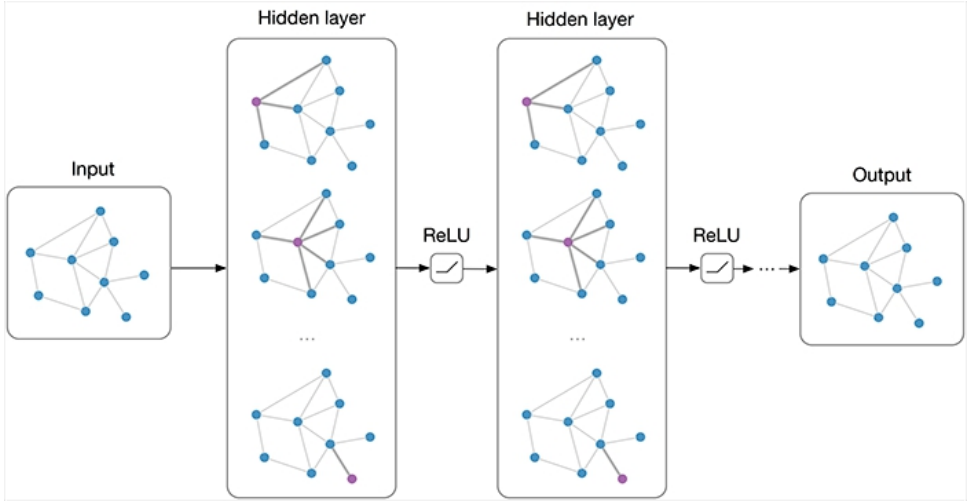
Graph Embedding:

- converting high-dimensional sparse graphs into low-dimensional, dense and continuous vector spaces, preserving maximally the graph structure properties
- DeepWalk, node2vec, struct2vec ...



Graph Convolution Network(GCN):

- can be used directly as a classification model or as a part of model to get graph embedding



Xu, M. (2020). Understanding graph embedding methods and their applications. arXiv preprint arXiv:2012.08019.

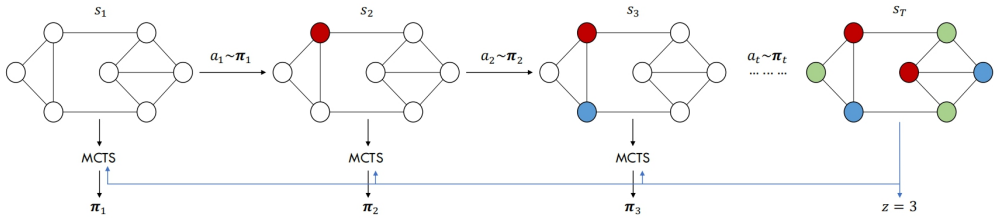
https://tkipf.github.io/graph-convolutional-networks/

DL solutions for Classical Problems

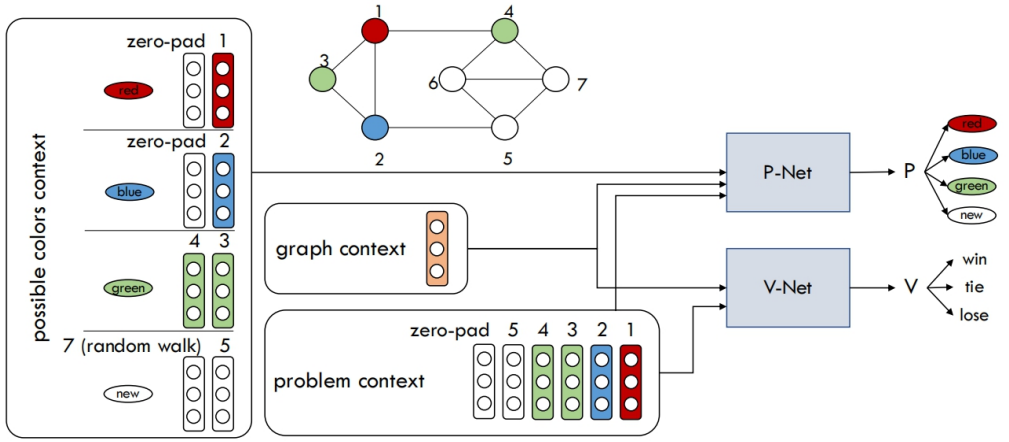
- The solution to graph coloring problem

FastColorNet (AlphaGoZero Framework):

- Graph embedding
- Monte Carlo Tree Search(MCTS)



- Deep reinforcement learning model



- Self-play

Jiayi Huang, et al. Coloring big graphs with alphagozero. arXiv preprint arXiv:1902.10162, 2019.

Combinatorial Optimization In Networking

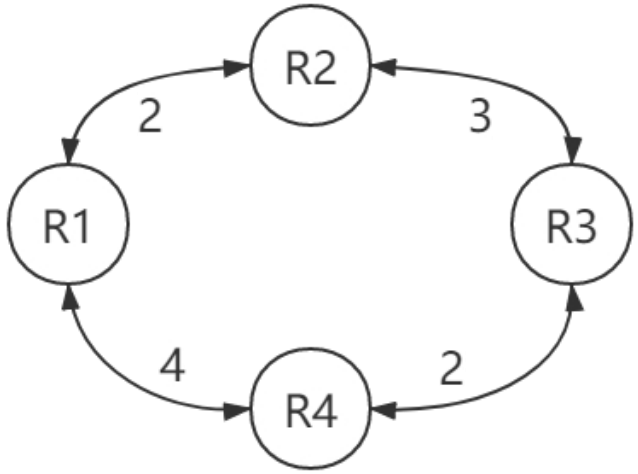
Application: Network Planning

- Benefits:
  - The 5G network brings many innovations, including higher speed, higher bandwidth, lower delay and so on.
  - At the same time, it also brings an explosion in data traffic, with more users and services.
  - A good network planning can not only bring high-quality network services to users, but also save bandwidth resources, which will reduce telecom operators' costs greatly.
- Typical scenario:
  - User services planning and deploying

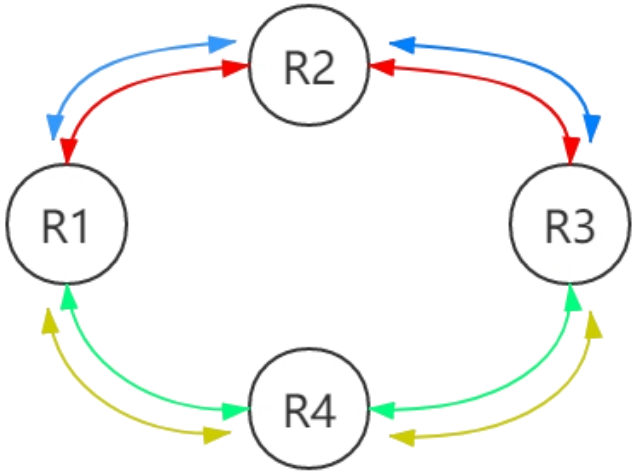
Details Of Our Challenge

- Goal:

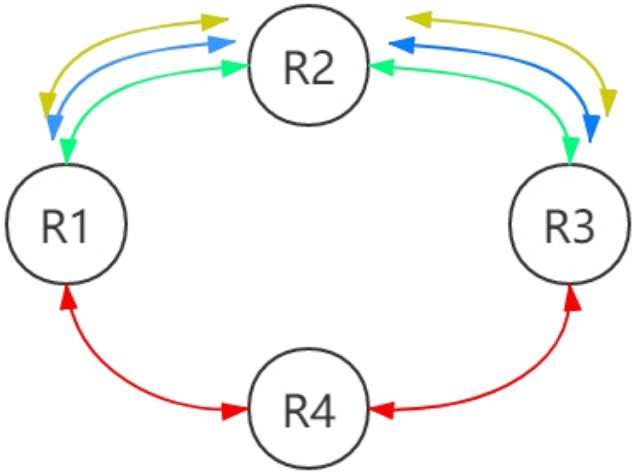
  Deploy all services to minimize the overall delay under the network resource constraints



link bandwidth	5
service:	s1 s2 s3 s4
route	R1 <-> R3
bandwidth	1 4 3 1



total delay:
5 + 5 + 6 + 6 = 22



total delay:
5 + 6 + 5 + 5 = 21

Details Of Our Challenge

- Challenge Description:
  - Please see statement from

    https://wiki.lfaidata.foundation/display/ADLIK/2021+Combinatorial+Optimization+Challenge:+Delivery+route+optimization
  - We have omitted the complicated telecommunication technology and abstracted it as a delivery route planning problem.

- Solution:
  - **Not limited to deep learning, any other algorithms which have excellent performance can be used**

Details Of Our Challenge

- Dataset:
  - Can be download from problem statement website
  - **Format:**

```
"""
graph format: a json object
"""
{
    nodes:[0,1,2] , # id list of the transportation nodes
    edges:[
        (0,1):{ # (starting point, end point)
            # id list of the lanes, the length of the array is the number of the lanes between the two adjacent nodes
            lanes: [0,1,2],
            lane_weights:[10, 10, 20] # The weight that each lane can carry
        }
    ]
}

"""
delivery format: an array of tuple
"""
[(starting point, endpoint, weight), (starting point, endpoint, weight)]
```

- **Tips:**

  1. We have public dataset for challengers and private dataset for testing

  2. You can generate your own data according to the problem description

Details Of Our Challenge

- Submitting:
  - Create a private Github repository to contain their work.
  - Add ITU-AIChallenge-ZTE as a collaborator. The repository must be made public before submission deadline and should be accessible till the end of the final event of the ITU challenge.

  - The repository should contains:
    1. Runnable Source codes
    2. result and running time of all the samples in dataset
    3. A description document describing how to verify the result with the source code, hardware configuration, explanation of the algorithm, and any other thing you want to say about the Challenge.

Details Of Our Challenge

- Evaluation Criteria:
  - Performance of algorithm (60%):
    - Both total cost and running time will be taken into account.
  - Solution advantage (30%):
    - Whether the solution is reasonable and whether the solution has enough practicability, innovation and universality.
  - Completeness (10%):
    - Whether the requirements of the challenge are fulfilled according to the proposed scheme and design.

Details Of Our Challenge

- Tips:
  - **Registration:** Register for the problem statement, you can enroll as a team with 1-4 members. The registration is open from June 9th, 2021 to August 31th, 2021.
  - **Submission:** We will begin to accept submissions from July 1st, 2021 and the submission deadline is October 25th, 2021, please submit by adding ITU-AIChallenge-ZTE as a collaborator of your github repo containing your solution.
  - **Evaluation:** All the submissions will be evaluated before November 10th, 2021.
  - **Global round:** Winners of the problem statement will take part in the Grand Challenge Finale of ITU AI/ML in 5G challenge and compete for the final awards.

Thanks