Machine Learning and Combinatorial Optimization

For combinatorial optimization, exact methods and heuristics are two main approaches to make decisions.

The COP can be modeled as sequential decision making problems. such as deciding an order to visit cities in TSP. The approaches can be divided into three classes.

- Solving COP based on deep learning (seq-seq\ Encoder-Decoder Ar chitecture), the esscence of this kind of methods is finding a mapping from sequence to sequence.

  $p_\theta(\pi|s) = \prod_{t=1}^{n} p_\theta(\pi_t|s, \pi_{1:t-1})$

paper	author	abstract	major contribution	similarities and difference	Comments
(2015Nips)pointer network	Vinyals O, Fortunato M, Jaitly N	Pointer Net (Ptr-Net) is based the attention mechanism and it uses attention as a **pointer** to select a member of the input sequence as the output.	（1）seq-seq:（2）attention mechanism(the output is probability not the element compared to the original attention mechanism)	the common works of encoder-decoder Architecture without RL: the objective is to maximize the conditional probability.	They didn't use RL to solve TSP, but proposed a network instructure to solve TSP. This is the cornerstone of using machine learning to solve COP. (2)performance relied on the quality of supervised lablels and getting a high-quality labels is expensive.
(2016)Neural Combinatorial Optimization with Reinforcement Learning	Bello I, Pham H, Le Q V, et al	using GAN(Graph embedding and node embedding)	defines $s = \{x_i\}_{i=1}^n$ as a sequence of n cities in 2-dim space, then calcula ting $L(\pi\|s)$. The stocahnistic policy $p(\pi\|s)$ is parameterized .The traing objective is $J(\theta\|s) = E_{\pi\ p_\theta} L(\pi\|s)$ training with the **Actor-Critic** algorithm. They also adopt search strategies: sampling and active search.	the common works of encoder-decoder Architecture with RL: the objective is to minimize the expectation of the tour length. **Using moel-free policy gradient method to optimize the parameters in RNN(PN), the objective is the total length of a tour.**	1)Advantages of using RL to train the pointer network: (1) do **not need optimal labels**, **only reward feedbacks**; (2)generalization ability(3) sampling solutions(policy gradient): Unbiased Monte-Carlo estimate.
(ICLR2019)ATTENTION, LEARN TO SOLVE ROUTING PROBLEMS!	Kool W, Van Hoof H, Welling M	They proposed a powerful model based on attention and train this model using REINFORCE with a simple but effective greedy rollout baseline.	training with the **REINFORCE** algorithm(PG method) **with deterministic Greedy Rollout Baseline**.	the training objective and policy is same to NCORL(2016);$L(\pi_\theta)$, policy:$p_\theta(\pi\|s)$	This article using greedy method as a baseline to guide Pointer Network to solve COP. Compared to the traditional action value function in RL，greedy method is more effective.
(2018Nips)Reinforcement learning for solving the vehicle routing problem	Nazari M, Oroojlooy A, Snyder L, et al	compared to Pointer Network, customers are dependent and there is no sequential information in inputs, just a set of unordered locations. So they change the RNN by the embedding layer, and using decoder and attention mechanism as PN.	compared to TSP，VRP takes account of the need of each customer. Training with the **REINFORCE** algorithm(PG method) without rollout baseline. They also adopt search strategies: sampling and active search.	the common works of encoder-decoder Architecture with RL: the objective is to minimize the expectation of the tour length or the waiting time of customers while satisfying the damand of customers.	This article is different from articles above for solving TSP. It cost too much for VRP to update the encoder for the varying demands of customers. This article is based on Ptr-Net.

Articles above all neglected the structure of the problem. Using RL to train the parameters in NN(RNN/GNN/GAN). And the objective is to minimize the expectation of tour length. the action is to find next city which will be visited next and get an ordered sequence of cities.

Articles above predicted the solution directively, a number of nodes were input to the Encoder-Decoder Architecture, then got an ordered list(the solution). But this kind of methods didn't take into account of the structure of the solution.

- The second class is to integrate the property of the COP and machine learning methods. These articles solved TSP/VRP based on Reinforcement learning(using reward feedbacks and without the encoder-decoder architecture)

paper	author	abstract	creativity	major contribution	Comment
(2018Nips)Learning to Perform Local Rewriting for Combinatorial Optimization	Xinyun Chen;Yuandong Tian(Facebook AI Research)	They proposed NeuRewriter that learns a policy to pick heuristics and rewrite the local components of the current solution to iterationly improve it until convergence.	Motivated by the "gradient descent" method, starts from a feasiable solution, iteratively converges to a good solution. They use RL algorithms to train two policy to iteratively modify the solution. Similar to heuristic methods, iteratively modifying the structure of the solution until it convergence.	The policy factorizes into a **region-picking and a rule-picking** component, each parameterized by a neural network(two NN) trained with **actor-critic methods** in RL. The state is the solution, the action is to find the region w_t(node for VRP) first to re-route the tour, then using the rule-picking policy to a moving actionu_t for the region w_t , the reward is $r(s_t,(w_t,u_t))=c(s_t)-c(s_{t+1})$ and the objective are the loss function($L_\omega(\theta)$) between the cumulative reward and $Q(s_t,w_t,\theta)$, and the the advantage function ($L_\mu(\phi)$). The total loss is $L_\mu(\phi)+\alpha L_\omega(\theta)$The training methods are Q-Actor-Critic algorithm and Advantage-Actor-Critic algorithm respectively.	NeuRewriter captures the general structure of combinatorial problems and shows strong performance in three versatile tasks: expression simplification, online job scheduling and vehicle routing problems.(I made the notes on the thesis and t he slides of yuxi Li.)
(Iclr2020)A LEARNING-BASED ITERATIVE METHOD FOR SOLVING VEHICLE ROUTING PROBLEMS	Hao Lu, Xingwen Zhang, Shuang Yang	starting from a initial solution, then iteratively refines the solution with improvement operator selected by an RL based controller and perturbation operator chosen by rule-based controller.	(1) meta-controller: a threhold to decide which operator should be use: improvement or perturbation. (2)actions: intra-route(to reduce the cost of current solution-consists of sub-routes) and inter-route(to reduce the cost of the total routes)	For VRP, the state is the features(solution-specific, e.g. stationary features like location and demand of each customer, and problem-specific-features(dynamic features) are based on current traveling plan.), the action are intra-route and inter-route, the first reward function is 1 if the operator improves current solution, -1 otherwise. And the second reward function is advantage-based which equals to the difference between the subsequent distance and the first iteration's result.	This article takes previous actions as well as their effect into consideration. The policy network is based on MLP and generate action probabilities. Embedding the features together at first, then using attention network and combined with previous actions.(the RL-based controller and the rule-based controller are not clearly)
			This article presents a **variable**		

(ICLR2020-declined, now ICJAI) Targeted sampling of enlarged neighborhood via Monte Carlo tree search for TSP	Zhang-hua Fu, Kai-Bin Qiu, MQ, Hongyuan Zha	A survey paper(Author sumarized these articles into two kind of methods: one is Encoder-Decoder, the other is the combination of CO and ML)	**neighbourhood search strategy** combined with machine learning in solving TSP. The search process of the optimal route is considered as a Markov decision process (MDP). And a 2-opt local search is used to search within a small neighborhood, while a Monte Carlo tree search(**MCTS)** method is used to **sample** a number of **targeted actions** within an enlarged neighborhood.	This kind of article is the same as Tian's article, they adopted 2-opt local search and MCTS to get a better solution. The state is the solution(an ordered list), the action can be viewed as an k-opt transformation which converts a given state s to a new state $s^{\star}$.	BUT this article is based on ML, not RL. The experiments result is better than other existing methods. The openreview given by editors is not innovative.
(2018)Machine Learning for Combinatorial Optimization: a Methodological Tour d'Horizon	Y Bengio, A Prouvos	(a survey)They divided these articles above into three classes: end to end learning, learning meaningful properties of optimization problems, and machine learning alongside optimization algorithms.		They summarize these three kind of articles according to this classification.	It's a good summary about using ML methods to solve Combinatorial Optimization problems.

These kinds of methods took the whole ordered sequence(solution) as the state.

- Solving cop based on graph theory

These articles are always the combination of graph theory and Encoder-Decoder Architecture with/without RL