TauRieL: Targeting Traveling Salesman Problem with deep reinforcement learning

Gorker Alp Malazgirt, Osman Unsal, Adrian Cristal
Barcelona Supercomputing Center, Barcelona, Spain
E-mail: {gorker.alp.malazgirt, osman.unsal, adrian.cristal}@bsc.es

Keywords—TSP, deep reinforcement learning, algorithms

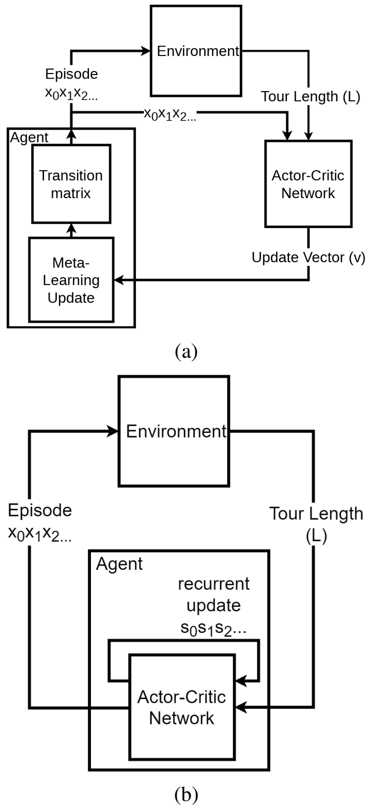
I. Extended Abstract

We propose TauRieL [1], a novel deep reinforcement learning (DRL) based method and target Traveling Salesman Problem (TSP) since it has broad applicability in theoretical and applied sciences. TauRieL utilizes an actor-critic inspired DRL architecture that adopts ordinary feedforward nets to obtain an update vector v. Then, we use v to improve the state transition matrix from which we generate the policy. Also, the state transition matrix allows the solver to initialize from precomputed solutions such as nearest neighbors. In an online learning setting, TauRieL unifies the training and the search where it can generate near-optimal results in seconds. The input to the neural nets in the actor-critic architecture are raw 2-D inputs, and the design idea behind this decision is to keep neural nets relatively smaller than the architectures with wide embeddings with the tradeoff of omitting any distributed representations of the embeddings.

TSP has been formulated and studied widely as a combinatorial optimization problem, and approximate heuristics based methods have been proposed. However, the design of heuristics requires detailed domain-specific knowledge and tuning. In this paper, we attack this aspect and develop a general and neural net (NN)-based TSP method that can produce feasible results two orders of magnitude faster than current NN-based methods with competitive solution quality. Current methods that target TSP use recurrent neural net (RNN) based methods. Figure 1(b) presents the RNN method. The RNN based algorithms stochastically explore the design space and learn the latent space using the RNN architectures. The work proposed by Bello et al. [1] is an example of RNN based method.

A. TauRieL

We represent the problem as a Markov Decision Process (MDP). Given a graph G that consists of cities $G = \{x\}_i^n$, the objective is to find shortest tour by visiting each city. We represent the environment as a Markov Decision Process (MDP), which is a tuple $\langle S, A, P, R \rangle$. S defines a state space where each state is a city in a given TSP instance. P is a state transition probability matrix such as the probability of reaching state s_j at time $t+1$ from s_i at time t is $P_{i,j}$. We define R as reward when agent transition from state s_i to s_j with action a. We are look for finding tours that generate higher rewards (where tour length is minimal) and assign them higher probabilities.



(a)

(b)

Fig. 1. High level schemas of TauRieL (a) and a state-of-the-art Actor-Critic based TSP solver using RNN [1](b)

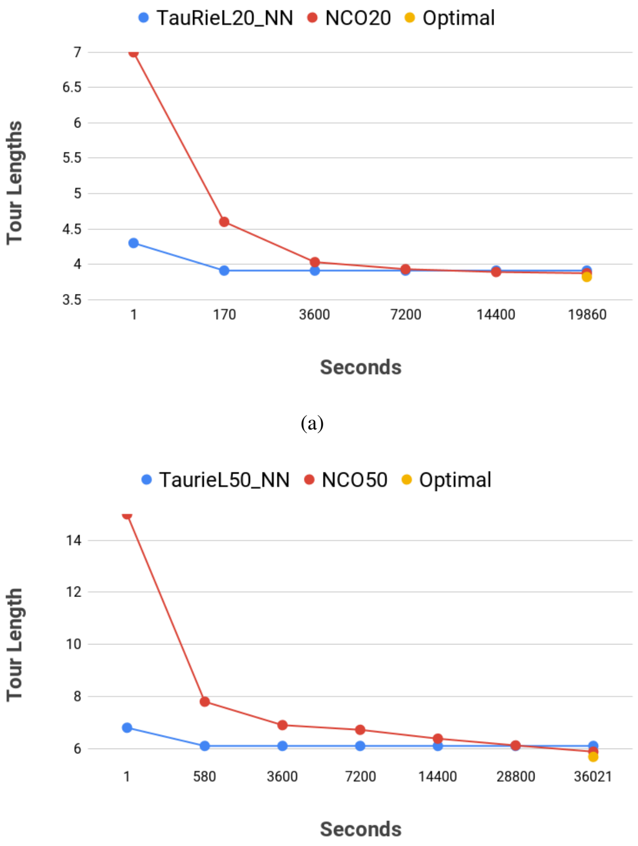
Our goal is to optimize the parameters $\theta_{act,cri}$ of the Actor-Critic neural nets that yield the best update vector v. We optimize the parameters of the neural net with respect to the objective i.e. the expected tour length:

$$J(\theta_{act}|G) = \mathbb{E}_{\phi \sim p(.|s)} L(\phi \mid G) \tag{1}$$

$L(\phi \mid G)$ is defined as the tour length of a given tour. The gradient $\nabla_{\theta_{act}}$ of the expected tour length is calculated using the REINFORCE algorithm [2] that we tailored for TSP and it is estimated by stochastic batch gradient method as:

$$\nabla_{\theta_{act}} J(\theta_{act}|G) = \frac{1}{B} \sum_{i=1}^{B} [(L(\phi \mid G) - b(G)) \\ \nabla_{\theta_{act}} log\ p(\phi \mid G)] \tag{2}$$

[1] A wood-elf character from Hobbit the Movie who possesses *superior pathfinding abilities*



(a)

(b)

Fig. 2. The average tour length vs training duration for 20 and 50-city instances of TauRieL and NCO [1]

TABLE I. Comparison of average tour lengths using the datasets provided by NCO [1], Sinkhorn Policy Gradient (SPG) [4] and A3 algorithm [5] obtained from [3]

N	Optimal	A3	NCO	SPG	TauRieL
10	2.87	3.07	2.88	NA	2.88
20	3.83	4.24	3.88	4.62	3.91
50	N/A	6.46	6.09	N/A	6.37

The $b(G)$ is called the baseline, which is a parametric metric that all generated tour lengths can be compared against. Hence, we use $b(G)$ to reduce large fluctuations of the tour lengths observed by the agent during the search. In this work, we select the baseline as the estimated tour length value obtained from the second neural network which is called the critic.

We update the transition matrix P with the update vector v using the update in Equation 3:

$$P_{i,j} = P_{i,j} + \epsilon\,(v_i - P_{i,j}) \qquad \forall i\ [i \in 1, \ldots, n]\ and\ \exists j\ [j \in 1, \ldots, n] \tag{3}$$

The actor net in the actor-critic architecture in Figure 1(a) is responsible for producing the update vector v and after each K episode the transition matrix is updated with v which aims to redirect the exploration towards shorter tour lengths.

B. Experimental Environment

We compare our results with the tour lengths obtained from the Google OR-Tools, Ptr-Net, NCO and Sinkhorn Policy Gradient [3], [1], [4]. We obtain 2.4% and 6.1% from the optimal compared to 1.4% and 3.5% which are reported by NCO[1] for 20-city and 50-city instances respectively. TauRieL outperforms A3 [5] for both 20 and 50 city instances and obtains tour lengths within 0.007 % and 2% of Ptr-Net for 20 and 50-city instances. Moreover, TauRieL outperforms an actor-critic based Sinkhorn Policy Gradient [4] in 20-cities case which is the only reported TSP size by the authors.

In Figures 2(a) and 2(b), we introduce the improvements in tour lengths with respect to the training. Specifically, we measure the training times for 20-city and 50-city instances of NCO as 19860 and 36021 seconds respectively. On the other hand, without needing any data sets, TauRieL runs in 170 seconds for 20-city and 580 seconds for 50-city instances when sample and episode step size is 250. For 20-city TSP, NCO [1] necessitates more than two hours of training in order to outperform TauRieL which can solve a 20-city instance in less than three minutes. Similarly, for 50-city, NCO needs to train at least eight hours to reach TauRieL's performance whereas TauRieL can obtain a solution in less than ten minutes. Hence, on average, training NCO below the specified duration yields poorer results than TauRieL.

C. Conclusion

In this study, we show that TauRieL generates TSP solutions two orders of magnitude faster per TSP instance as compared to state-of-the-art offline techniques with a performance impact of 3% in the worst case.

References

[1] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv preprint arXiv:1611.09940*, 2016.

[2] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[3] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2692–2700.

[4] P. Emami and S. Ranka, "Learning permutations with sinkhorn policy gradient," *arXiv preprint arXiv:1805.07010*, 2018.

[5] "C++ implementation of traveling salesman problem using christofides and 2-opt," https://github.com/beckysag/traveling-salesman, 2017.

Gorker Alp Malazgirt received his B.S. in Microelectronics Engineering from Sabanci University, Turkey, his M.S. in Electrical Engineering from Lund University, Sweden and his Ph.D. from Universitat Politècnica de Catalunya, Spain. He worked at Ericsson and ST-Ericsson in Lund, Sweden as an R&D engineer. His research interests are algorithms, computer architectures and reconfigurable computing.