A Reinforcement Learning Approach to the Labeled Maximum Matching Problem

Maximilian Moll$^{1[0000-0002-0270-3039]}$ and Leonhard Kunczik$^{1[0000-0003-0907-7250]}$

Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85579 Neubiberg
{maximilian.moll,leonhard.kunczik}@unibw.de

Abstract. The application of reinforcement learning to combinatorial optimization is studied by considering the labeled maximum matching problem. To this end, a suitable environment, graph embedding, and training approach are being discussed.

Keywords: Reinforcement Learning · Combinatorial Optimization · Graph Embedding.

1 Introduction

After the wave of interest in neural networks, the field of Reinforcement Learning (RL) gains more and more attention. Lately, after breakthroughs such as AlphaZero [5] or Alpha Star [6] this data-based approach to optimal control has become widely recognized. However, despite the many successes in playing games, real applications - at least outside of robotics - are quite rare.

Recently, attempts have been made to apply RL to combinatorial optimization problems [4]. The benefit is obvious: combinatorial optimization has many important industrial applications, like the traveling salesman problem. However, there are also many reasonable classical solutions to these problems. While an RL approach might lack the theoretical guarantees usually associated with traditional solutions, it provides much better scalability.

2 Problem Description

In this work, we study the labeled maximum matching problem (LMMP), which was first described in [2]. Let $G = (V, E)$ be a graph with vertex set G and edge set $E \subset V \times V$. A matching $M \subset E$ is a set of non-adjacent edges. If the size of M is the largest among all possible matchings, it is a maximum matching. For the labeled problem we introduce a label function $L \colon E \to \{c_1, \dots, c_m\}$, which assigns a label to each edge. The problem is now to find a maximum matching of a given graph G using a minimum number of labels, i.e.

$$\underset{M \subset E}{\operatorname{argmin}} |\{L(e)|e \in M : \text{M a maximum matching}\}|. \tag{1}$$

To translate this problem into an RL setting, several choices have to be made. We decided to choose a state consisting of a graph and an indicator list of already used labels. The starting state is then the graph under consideration and no used labels. At each time step, the action consists of selecting one of the edges, which will be added to the matching. The label of that edge is added to the list of already used labels. Moreover, the nodes of the selected edge, along with all their edges, are removed from the state graph to ensure that the chosen edges actually form a matching. The reward given at any time step is simply 1. Once arriving in the final state, i.e., an empty state graph, there is a penalizing reward for the number of labels being used. Moreover, should the number of time steps be below the size of a maximum matching, a further large penalty is being given. This prioritizes finding a maximum matching over selecting a small number of labels.

3 Approach

Next, a suitable RL set-up to solve the above task has to be chosen. To obtain more stable and sample efficient training, then is usually associated with deep RL, Graph Neural Networks were avoided here. Moreover, such an approach would fix at least an upper bound for the graph size used in the training and test sets.

Instead, we chose to use a Graph-to-Vector embedding [3], which will be end-to-end trainable. However, since in the problem under consideration, the emphasis is clearly on the edges, the embedding was reformulated to work on those. A crucial point here is the chosen size of the embedding, which needs to be determined experimentally.

To construct a suitable Q function, we followed a similar approach to [4], adapting it to the problem at hand. For the training of this set-up, we decided to use a Covariance-Matrix-Adaption Evolutionary Strategy [1]. Training is conducted on random graphs of varying sizes up to certain bounds.

4 Outlook

While the performance needs to be evaluated experimentally, there is no reason why the described approach should not work. However, the question remains, why such a computationally expensive approach should be preferred to a traditional approach, e.g., a MILP formulation. One goal of the work described above is to fine-tune the system to the point that training on small networks will give good performance for significantly larger networks.

References

1. Auger, A., Hansen, N.: Tutorial cma-es: evolution strategies and covariance matrix adaptation. In: Proceedings of the 14th annual conference companion on Genetic and evolutionary computation. pp. 827–848 (2012)

2. Carrabs, F., Cerulli, R., Gentili, M.: The labeled maximum matching problem. Computers & Operations Research **36**(6), 1859–1871 (2009)
3. Dai, H., Dai, B., Song, L.: Discriminative embeddings of latent variable models for structured data. In: International conference on machine learning. pp. 2702–2711 (2016)
4. Khalil, E., Dai, H., Zhang, Y., Dilkina, B., Song, L.: Learning combinatorial optimization algorithms over graphs. In: Advances in Neural Information Processing Systems. pp. 6348–6358 (2017)
5. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of go without human knowledge. Nature **550**(7676), 354–359 (2017)
6. Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al.: Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature **575**(7782), 350–354 (2019)