Accelerating Diffusion-based Combinatorial Optimization Solvers by Progressive Distillation

Junwei Huang
Carnegie Mellon University
`junweih@andrew.cmu.edu`

Zhiqing Sun
Carnegie Mellon University
`zhiqings@cs.cmu.edu`

Yiming Yang
Carnegie Mellon University
`yiming@cs.cmu.edu`

Abstract

Graph-based diffusion models have shown promising results in terms of generating high-quality solutions to NP-complete (NPC) combinatorial optimization (CO) problems. However, those models are often inefficient in inference, due to the iterative evaluation nature of the denoising diffusion process. This paper proposes to use *progressive* distillation to speed up the inference by taking fewer steps (e.g., forecasting two steps ahead within a single step) during the denoising process. Our experimental results show that the progressively distilled model can perform inference **16** times faster with only **0.019%** degradation in performance on the TSP-50 dataset.

1 Introduction

The realm of Combinatorial Optimization (CO) is intricately tied to optimization within discrete spaces. A significant portion of CO problems falls into the category of NP-complete (NPC) problems, for which efficient algorithms for exact solutions are nearly impossible to find. In the past, researchers have often turned to traditional approximation algorithms as a means to tackle these NPC problems (Arora, 1998; Lawler et al., 1980). Unfortunately, these traditional methods demand an extensive degree of mathematical understanding specific to each individual problem, resulting in heuristics that lack transferability across diverse problem types.

Modern approaches utilize the pattern recognition power of machine learning techniques to generate high-quality solutions. Representative methods include the deep reinforcement learning (DRL) models (Khalil et al., 2017; Barrett et al., 2019; Yao et al., 2021; Bello et al., 2016) and neural diffusion models (Graikos et al., 2022; Sun and Yang, 2023). DRL approaches focus on training an agent to learn the heuristics for solving a CO problem, by allowing the agent to interact with its environment, aided by a graph representation of the problem. In the unsupervised setting for a DRL solver, the system learns the search strategy without requiring annotated optimal solutions for training-set graphs (Karalias and Loukas, 2020; Wang et al., 2022), but DRL CO solvers are usually hard to train, due to the sparse reward problem (Wu et al., 2021). In the supervised settings, on the other hand, neural network solvers learn a generative process to directly predict the high-quality solutions to CO problems (Joshi et al., 2019; Sun and Yang, 2023). They are usually more stable in training and efficient in inference, but require near-optimal annotations generated by traditional solvers.

Recent progress in diffusion models for image generation (Ho et al., 2020; Song et al., 2020; Song and Ermon, 2020) has sparked interest in applying diffusion models to CO problem with the hope of exploiting its expressiveness. Graikos et al. (2022) show by encoding the traveling salesman problem (TSP) as 64×64 greyscale images, image diffusion models (Ho et al., 2020) are able to generate heatmaps from which the solutions could be extracted. Sun and Yang (2023) proposed the DIFUSCO framework, which works by explicitly modeling the graph structure of CO problems and denoising

ICML 2023 Workshop: Sampling and Optimization in Discrete Space (SODS 2023).

the variable indicator vector. However, these diffusion-based methods are inefficient in inference because of the non-trivial number of inference steps required to produce satisfactory results.

To remedy the efficiency problem encountered by these diffusion-based solvers, we propose to use progressive distillation (Rezagholizadeh et al., 2022; Salimans and Ho, 2022; Meng et al., 2023) to address the inference efficiency issue of DIFUSCO, accelerating its inference while preserving its solution quality. We train the student to compress two denoising steps from the teacher into one in a single training iteration and we iteratively perform this compression to achieve further distilled student. We report a 0.019% performance degradation on our 4x distilled student, which inferences 16 times faster than the teacher.

2 Method

2.1 Problem Definition

In this paper, our primary focus is on the Travelling Salesman Problem (TSP) as a subject of investigation. However, we posit that both the algorithm we have developed and our conclusions have broader applicability, potentially extending to other types of Combinatorial Optimization (CO) problems. This perspective aligns with the findings presented in DIFUSCO (Sun and Yang, 2023).

Formally, we define the possible solution space of TSP instance $\boldsymbol{s}$ to be $\mathcal{X}_{\boldsymbol{s}} = \{0, 1\}^N$, where N is the number of edges in the problem instance. For each possible solution vector $\mathbf{x} \in \mathcal{X}_{\boldsymbol{s}}$, $\mathbf{x}_i$ is the indicator variable for whether edge i is chosen in the solution vector $\mathbf{x}$. We define the objective function $\boldsymbol{J}_{\boldsymbol{s}} : \mathcal{X}_{\boldsymbol{s}} \to \mathbb{R}$ such that it incorporates both the cost and the validity of the solution:

$$\boldsymbol{J}_{\boldsymbol{s}}(\mathbf{x}) = \mathbf{cost}_{\boldsymbol{s}}(\mathbf{x}) + \mathbf{valid}_{\boldsymbol{s}}(\mathbf{x}) \tag{1}$$

where $\mathbf{cost}_{\boldsymbol{s}}(\mathbf{x}) = \mathbf{x}^T\mathbf{d}, \mathbf{d}_i$ being the i^{th} edge's weight in $\boldsymbol{s}$, and $\mathbf{valid}_{\boldsymbol{s}}(\mathbf{x})$ is 0 for a feasible TSP solution, $+\infty$ for infeasible solutions. The optimal solution to $\boldsymbol{s}$ is

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}_{\boldsymbol{s}}}{\arg\min}\, \boldsymbol{J}_{\boldsymbol{s}}(\mathbf{x}) \tag{2}$$

2.2 Diffusion Solvers for Combinatorial Optimization

DIFUSCO (Sun and Yang, 2023) is a framework for directly modeling the graph representation of CO problems and solving CO problems via diffusion models. By first re-scaling (Chen and Tian, 2019) $\{0, 1\}$ valued variables to $\{-1, 1\}$ and treating them as real-valued variables, it becomes viable to apply continuous diffusion (Ho et al., 2020; Song et al., 2020) to the problem setup in 2.1. Re-scaling for binary valued variable at time step $t \in \{1, \dots T\}$ is given as:

$$\hat{\mathbf{x}}_t = 2\mathbf{x}_t - 1 \tag{3}$$

After the re-scaling, we then use the vanilla forward process proposed by Ho et al. (2020)

$$q_\theta(\hat{\mathbf{x}}_t|\hat{\mathbf{x}}_{t-1}) = \mathcal{N}(\hat{\mathbf{x}}_t; \sqrt{1-\beta_t}\hat{\mathbf{x}}_{t-1}, \beta_t\mathbf{I}) \tag{4}$$

where β_t is the ratio of corruption at timestep t. The t step marginal distribution of $\hat{\mathbf{x}}_t$ is therefore:

$$q_\theta(\hat{\mathbf{x}}_t|\hat{\mathbf{x}}_0) = \mathcal{N}(\hat{\mathbf{x}}_t; \sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{5}$$

We train a neural network $\epsilon_\theta(\cdot, \cdot)$ to predict the Gaussian noise ϵ and given the corrupted $\hat{\mathbf{x}}_t$ and timestep t.

$$\tilde{\epsilon_t} = \epsilon_\theta(\hat{\mathbf{x}}_t, t) \tag{6}$$

We apply the Mean Squared Error (MSE) Loss to $\tilde{\epsilon}_t$ and ϵ_t and optimize ϵ_θ with respect to this loss, producing a performant predictor of noise that we could use to iteratively denoise and produce the predicted solution $\hat{\mathbf{x}}_0$. The predicted solution $\hat{\mathbf{x}}_0$ is obtained with the Gaussian Posterior from Denoising Diffusion Implicit Model (DDIM) (Song et al., 2020).

$$\text{GuassianPosterior}(\tilde{\epsilon}_t, \hat{\mathbf{x}}_t) = \sqrt{\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}}\left(\hat{\mathbf{x}}_t - \sqrt{1-\bar{\alpha}_t}\tilde{\epsilon}_t\right) + \sqrt{1-\bar{\alpha}_{t-1}}\tilde{\epsilon}_t \tag{7}$$

We finally perform quantization to obtain $\mathbf{x}_0$ from $\hat{\mathbf{x}}_0$.

Algorithm 1 Progressive Distillation on DIFUSCO - training

```
Require: Trained teacher model $\hat{\epsilon_\eta}(\cdot, \cdot)$
Require: Data Set $\mathcal{D}$, Student sampling steps $N$, Distillation factor $K$
  for $K$ iterations do
    $\theta \leftarrow \eta$                                          ▷ Init student from teacher
    while not converged do
      Sample { $\mathbf{x} \sim \mathcal{D}$,   $t = i/N$,   $i \sim$ Categorical $[1, 2, \dots, N]$,   $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ }
      $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
      $t' = t - 0.5/N, t'' = t - 1/N$
      #2 steps of teacher denoising
      $\mathbf{x}_{t'} =$ GaussianPosterior$_\eta$ $[\hat{\epsilon_\eta}(\mathbf{x}_t, t), \mathbf{x}_t]$
      $\mathbf{x}_{t''} =$ GaussianPosterior$_\eta$ $[\hat{\epsilon_\eta}(\mathbf{x}_{t'}, t'), \mathbf{x}_{t'}]$
      #1 step of student denoising
      $\tilde{\mathbf{x}} =$ GaussianPosterior$_\theta$ $[\hat{\epsilon_\theta}(\mathbf{x}_t, t), \mathbf{x}_t]$
      $\mathcal{L} = ||\mathbf{x}_{t''} - \tilde{\mathbf{x}}||_2^2$                               ▷ MSE loss
      $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}$          ▷ Gradient Update on student model parameters
    end while
    $\eta \leftarrow \theta$                                  ▷ Student becomes the new teacher
    $N \leftarrow N/2$                                    ▷ Halve number of sampling steps
  end for
```

2.3 Progressive Distillation

The DIFUSCO framework is capable of producing high-quality solutions for CO problems like TSP-50, TSP-100, etc. However, this framework is often inefficient in inference due to the need of a considerable number of inference diffusion steps to generate reasonable solutions. Aiming to accelerate this method while preserving solution quality, we propose to use progressive distillation to accelerate its inference.

Progressive Distillation (Rezagholizadeh et al., 2022; Salimans and Ho, 2022) is an algorithm designed to accelerate the inference speed of diffusion models. For one iteration of progressive distillation, we update the student model to make its one denoising step match two denoising steps from the teacher model. We then iteratively train a series of student models in this manner, where the teacher model in each iteration is initialized from the converged student model in the previous iteration. The objective is to progressively distill knowledge from one model to the next, gradually improving the performance of the student model with fewer inference steps.

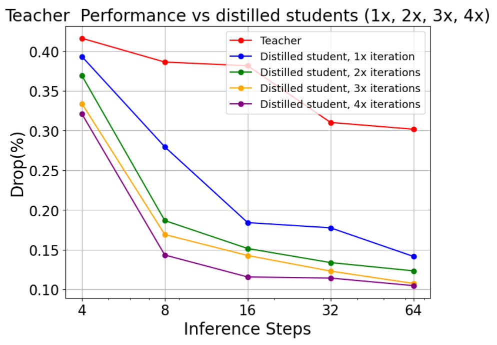
By the end of multiple distillation iterations, it becomes viable for the yielded progressively distilled student model to compress the diffusion steps required to achieve the same level of quality of solution by multiple times. We describe the progressive distillation training algorithm for DIFUSCO in 1.

3 Experiments

We benchmark our results against the TSP-50 dataset, where each training sample $\mathbf{x}_0$ is the solution to a TSP instance with exactly 50 vertices in the problem graph. We fixed the diffusion steps to 1024 in all experiments. For the teacher model and each student model, we vary the number of inference steps as powers of 2, from 4 to 64 steps. The result of the distilled students after $1\times$, $2\times$, $3\times$, and $4\times$ progressive distillation iterations are shown in Figure 1. We also perform $4\times$ parallel sampling for all experiments in Figure 1 and report results in Figure 2. $1\times$, $2\times$, $3\times$, and $4\times$ distilled students effectively have 512, 256, 128, and 64 diffusion steps during training.

Model Setup Following Sun and Yang (2023), we set the noise predictor neural network $\hat{\epsilon}_\theta(\cdot, \cdot)$ to be a 12-layer anisotropic Graph Neural Network (Bresson and Laurent, 2018) with a width of 256. We fix the diffusion scheduler $\{\beta_i\}_{i=1}^T$ to be a linear denoising schedule from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$. We use DDIM Song et al. (2020) for fast inference and use a linear skipping strategy for all experiments (Sun and Yang, 2023).

Performance Metrics Per convention, we use the solution cost drop percentage on the test set as the metric of performance. Solution cost drop percentage is calculated as the the difference between yielded solution cost and ground truth cost divided by the ground truth cost.



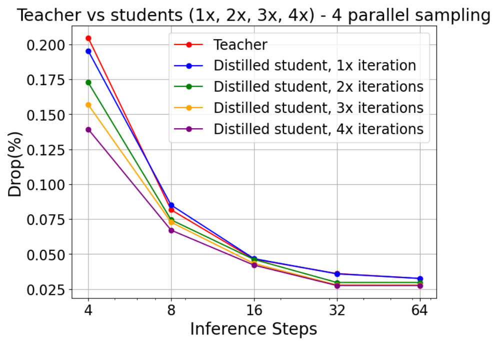


Figure 1: TSP-50 results for $1\times$, $2\times$, $3\times$, and $4\times$ distilled student model performance against an optimized teacher model performance. We vary the number of inference steps and tabulate performance drop on each choice of inference step. Figure 2: We enable $4\times$ parallel sampling where we decode 4 samples in parallel and pick the best candidate.

Same inference steps We first examine how much better our distilled student performs when the teacher and the student are allowed the same number of inference steps. As shown in Figure 1, we observed that our distilled student performs considerably better than the teacher when we allow the same number of inference steps for the student and teacher. The student is expected to compress multiple steps of the teacher into one step of its own. Therefore the student inferencing with a distill factor of k is theoretically equivalent to teacher inferencing with k times the student inference steps. When parallel sampling is enabled, we observe that performance gap between teacher and distilled students shrinks but does not vanish.

Inference with $1/k$ of steps We also examine how minimal the performance degradation is when we restrict the student to inference with $1/k$ of teacher inference steps We observed that distilled students are able to retain minimal performance degradation compared to our already optimized teacher. Our 4x distilled student was able to compress 64 inference steps of teacher model into 4 steps. Comparing the result of student inferencing with 4 steps and teacher inferencing with 64 steps, we observed a performance degradation of 0.019%.

4 Related Work

Prior work by Graikos et al. (2022) used image diffusion model to introduce visual inductive bias to aid solving CO problems. We would like to point out progressive distillation on image generation models has been explored by Salimans and Ho (2022). We explored upon the prediction target mentioned and decided to adopt unscaled clean solution as prediction target due to its stability during experimentation.

5 Discussion & Conclusion

In this paper, we have demonstrated that the adoption of a progressive distillation scheme can significantly accelerate the inference process of diffusion-based CO solvers. Crucially, this enhancement in speed does not compromise the quality of the solutions obtained, resulting in only a negligible performance degradation. While our current experimentation is focused on TSP-50, we are optimistic about the potential of extending our methodology to other CO problems in future studies. This area of research provides a promising direction for future exploration and analysis.

Aside from continuous Gaussian noises, DIFUSCO (Sun and Yang, 2023) also mentioned injecting discrete Bernoulli noise which yielded better experimental results. We will extend our distillation scheme to support Bernoulli noise in future work, taking advantage of its inherent consistency with the discrete nature of CO problems. We also consider exploring other performant backbone networks, such as transformers (Peebles and Xie, 2022).

References

S. Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, sep 1998. ISSN 0004-5411. doi: 10.1145/290179.290180. URL `https://doi.org/10.1145/290179.290180`.

T. D. Barrett, W. R. Clements, J. N. Foerster, and A. I. Lvovsky. Exploratory combinatorial optimization with reinforcement learning. *ArXiv*, abs/1909.04063, 2019.

I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio. Neural combinatorial optimization with reinforcement learning. *ArXiv*, abs/1611.09940, 2016.

X. Bresson and T. Laurent. An experimental study of neural networks for variable graphs. In *International Conference on Learning Representations Workshop*, 2018.

X. Chen and Y. Tian. Learning to perform local rewriting for combinatorial optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/131f383b434fdf48079bff1e44e2d9a5-Paper.pdf`.

A. Graikos, N. Malkin, N. Jojic, and D. Samaras. Diffusion models as plug-and-play priors. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. URL `https://arxiv.org/pdf/2206.09012.pdf`.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

C. K. Joshi, T. Laurent, and X. Bresson. An efficient graph convolutional network technique for the travelling salesman problem. *arXiv preprint arXiv:1906.01227*, 2019.

N. Karalias and A. Loukas. Erdos goes neural: an unsupervised learning framework for combinatorial optimization on graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6659–6672. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/49f85a9ed090b20c8bed85a5923c669f-Paper.pdf`.

E. B. Khalil, H. Dai, Y. Zhang, B. N. Dilkina, and L. Song. Learning combinatorial optimization algorithms over graphs. In *NIPS*, 2017.

E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Generating all maximal independent sets: Np-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9(3):558–565, 1980. doi: 10.1137/0209042. URL `https://doi.org/10.1137/0209042`.

C. Meng, R. Rombach, R. Gao, D. Kingma, S. Ermon, J. Ho, and T. Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023.

W. S. Peebles and S. Xie. Scalable diffusion models with transformers. *ArXiv*, abs/2212.09748, 2022.

M. Rezagholizadeh, A. Jafari, P. S. Saladi, P. Sharma, A. S. Pasand, and A. Ghodsi. Pro-KD: Progressive distillation by following the footsteps of the teacher. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4714–4727, Gyeongju, Republic of Korea, Oct. 2022. International Committee on Computational Linguistics. URL `https://aclanthology.org/2022.coling-1.418`.

T. Salimans and J. Ho. Progressive distillation for fast sampling of diffusion models. *ArXiv*, abs/2202.00512, 2022.

J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, October 2020. URL `https://arxiv.org/abs/2010.02502`.

Y. Song and S. Ermon. Improved techniques for training score-based generative models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12438–12448. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92c3b916311a5517d9290576e3ea37ad-Paper.pdf.

Z. Sun and Y. Yang. Difusco: Graph-based diffusion solvers for combinatorial optimization, 2023.

H. Wang, N. Wu, H. Yang, C. Hao, and P. Li. Unsupervised learning for combinatorial optimization with principled objective relaxation. *ArXiv*, abs/2207.05984, 2022.

Y. Wu, W. Song, Z. Cao, J. Zhang, and A. Lim. Learning improvement heuristics for solving routing problems. *IEEE transactions on neural networks and learning systems*, 33(9):5057–5069, 2021.

F. Yao, R. Cai, and H. Wang. Reversible action design for combinatorial optimization with reinforcement learning. *ArXiv*, abs/2102.07210, 2021.