On Training-Test (Mis)alignment in Unsupervised Combinatorial Optimization: Observation, Empirical Exploration, and Analysis

Fanchen Bu [1 2] **Kijung Shin** [3 1]

Abstract

In *unsupervised combinatorial optimization* (UCO), during training, one aims to have continuous decisions that are promising in a *probabilistic* sense for each training instance, which enables end-to-end training on initially discrete and non-differentiable problems. At the test time, for each test instance, starting from continuous decisions, *derandomization* is typically applied to obtain the final deterministic decisions. Researchers have developed more and more powerful test-time derandomization schemes to enhance the empirical performance and the theoretical guarantee of UCO methods. However, we notice a misalignment between training and testing in the existing UCO methods. Consequently, lower training losses do not necessarily entail better post-derandomization performance, *even for the training instances without any data distribution shift*. Empirically, we indeed observe such undesirable cases. We explore a preliminary idea to better align training and testing in UCO by including a differentiable version of derandomization into training. Our empirical exploration shows that such an idea indeed improves training-test alignment, but also introduces nontrivial challenges into training.

1. Introduction

Combinatorial optimization (CO) problems are naturally discrete. Typical examples include optimization problems on graphs where we make binary yes-or-no decisions on each node, and the objective is a function of the graph structure and the binary decisions. CO problems have a rich lineage in various research fields, including theoretical computer science (Arumugam et al., 2016) and operations research (Modaresi et al., 2020), with real-world applications from network design (Cheng et al., 2006) to scheduling (Hwang & Cheng, 2001) and bioinformatics (Bauer et al., 2007). However, the discrete nature of CO problems makes it non-trivial to apply typical machine learning methods that are based on differentiable optimization to them.

To overcome these challenges, researchers have explored various strategies to effectively combine machine learning with CO problems (Bengio et al., 2021). Early approaches often involved supervised methods (Li et al., 2018) or reinforcement learning techniques (Kool et al., 2019; Berto et al., 2025). However, these methods require labeled data or extensive interaction, limiting their applicability and generalizability in many real-world scenarios (Karalias & Loukas, 2020). Consequently, the field has seen a growing interest in unsupervised approaches, resulting in the development of *unsupervised combinatorial optimization* (UCO) methods.

The key idea of UCO is to (1) allow continuous decisions and (2) evaluate the expected objective by interpreting the continuous decisions as random variables, which gives a fully differentiable process and enables end-to-end training. At the test time, *derandomization* is typically applied to continuous decisions to transform them into deterministic decisions as the final output (Karalias & Loukas, 2020).

Over the course of time, UCO researchers have developed more and more powerful test-time derandomization schemes, from naive random sampling (Karalias & Loukas, 2020), iterative rounding (Karalias & Loukas, 2020; Wang et al., 2022), to greedy derandomization (Bu et al., 2024). With the development of test-time derandomization, we have witnessed the enhancement of the empirical performance as well as the theoretical quality guarantee of UCO methods.

However, we notice a misalignment between training and testing in the existing UCO methods: the training essentially tries to optimize the expected quality of the output continuous decisions assuming naive random sampling, while rather sophisticated derandomization is actually used at the test time. Therefore, even if we have lower training losses, we cannot guarantee better post-derandomization performance at the test time, *even for the training instances* (i.e., here we

[1] School of Electrical Engineering, KAIST, Daejeon, Republic of Korea [2] AI4CO Open-Source Community [3] Kim Jaechul Graduate School of Artificial Intelligence, KAIST, Seoul, Republic of Korea. Correspondence to: Kijung Shin <kijungs@kaist.ac.kr>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Algorithm 1 Iterative Rounding

```
Input: (1) Continuous decisions $\tilde{D} \in [0,1]^n$,
       (2) Rounding sequence $\pi_n = (v_1, \dots, v_n)$
Output: Final discrete decisions $D \in \{0,1\}^n$
$D \leftarrow \tilde{D}$                                  ▷ Initialization
for $j \leftarrow v_1$ to $v_n$ do                        ▷ Iteration following the sequence
    for $b \in \{0,1\}$ do
        $D' \leftarrow D$                                 ▷ Copy
        $D'_j \leftarrow b$                               ▷ Modify a single entry
        $\Delta_b \leftarrow \tilde{f}(D) - \tilde{f}(D')$  ▷ Evaluation
    end for
    $D_j \leftarrow \arg\max_{b \in \{0,1\}} \Delta_b$      ▷ Rounding
end for
return $D$
```

are not discussing the training-test misalignment regarding data distributions but regarding methodology). We indeed empirically observe such undesirable cases.

We explore a preliminary idea to better align training and testing in UCO by including a differentiable version of derandomization into training. By our empirical exploration, we validate that such an idea indeed can improve training-test alignment. However, we also observe that including such additional soft derandomization schemes into training increases the difficulty of training, i.e., we may not be able to have the training losses stably decrease during training. Our analysis suggests that the future development of UCO methods may need to find a balance between training-test alignment and the ease of training.

Reproducibility. The code and datasets are available in the online appendix (Bu & Shin, 2025).[1]

2. Preliminaries and Background

2.1. Combinatorial Optimization (CO)

We consider CO problems with n binary decisions, where n is a positive integer. A CO problem aims to find the *optimal decisions* D_{opt} inside a *feasible set* $\mathcal{C}$ such that $D_{\text{opt}} \in \mathcal{C} \subseteq \{0,1\}^n$, to minimize an *objective* $f : \mathcal{C} \mapsto \mathbb{R}$ (i.e., $D_{\text{opt}} = \arg\min_{D \in \mathcal{C}} f(D)$).

2.2. Unsupervised Combinatorial Optimization (UCO)

Based on the probabilistic method (Erdős & Spencer, 1974), Karalias & Loukas (2020) proposed the UCO framework with the high-level idea to (1) apply continuous relaxation to the objective f and its domain, to obtain $\tilde{f} : [0,1]^n \to \mathbb{R}$ and enable end-to-end training, and (2) apply derandomization at the test time to obtain the final output decisions $D_{\text{out}} \in \mathcal{C}$.

Algorithm 2 Greedy Rounding

```
Input: Continuous decisions $\tilde{D} \in [0,1]^n$
Output: Final discrete decisions $D \in \{0,1\}^n$
$D \leftarrow \tilde{D}$                                  ▷ Initialization
repeat
    for $j \in \{1, \dots, n\}$ and $b \in \{0,1\}$ do
        $D' \leftarrow D$                                 ▷ Copy
        $D'_j \leftarrow b$                               ▷ Modify a single entry
        $\Delta_{j,b} \leftarrow \tilde{f}(D) - \tilde{f}(D')$  ▷ Evaluation
    end for
    $(j^*, b^*) \leftarrow \arg\max_{j,b \in [n] \times \{0,1\}} \Delta_{j,b}$  ▷ Best choice
    $D_{j^*} \leftarrow b^*$                              ▷ Rounding with the best choice
until $\Delta_{j^*,b^*} \leq 0$                           ▷ Until local minimum
return $D$
```

Continuous relaxation. Each $\tilde{D} \in [0,1]^n$ is interpreted as a *distribution* (typically, an independent multivariate Bernoulli distribution) on $\{0,1\}^n$, and we aim to construct a *differentiable* $\tilde{f}$ such that $\tilde{f}(\tilde{D}) \approx \mathbb{E}_{D \sim \tilde{D}}[f(D)] + \beta \Pr_{D \sim \tilde{D}}[D \notin \mathcal{C}]$ with constraint coefficient $\beta > 0$. The key points are (1) now we are able to conduct end-to-end training with this differentiable surrogate $\tilde{f}$ instead of the originally discrete non-differentiable f, and (2) when $\tilde{f}$ is minimized, it is guaranteed that the optimal $\tilde{D}_{\text{opt}} = \arg\min_{\tilde{D} \in [0,1]^n} \tilde{f}(\tilde{D})$ corresponds to the optimal D_{opt} for the original objective f.

Derandomization. At the test time, for each test instance, *derandomization* is used to obtain the final output discrete decisions. Researchers have considered various derandomization schemes for UCO, including (let $\tilde{D}_{\text{output}} \in [0,1]^n$ be the initial continuous output for the test instance):

- **Naive random sampling** (Karalias & Loukas, 2020): We sample D_{out} from the distribution represented by $\tilde{D}_{\text{output}}$;
- **Iterative rounding** (Karalias & Loukas, 2020; Wang et al., 2022): We fix a sequence $\pi_n = \{v_1, v_2, \dots, v_n\}$, iterate for $i = 1, 2, \dots, n$ while rounding for v_i to the one between 0 and 1 that gives lower $\tilde{f}$, and obtain D_{out} after rounding all the n entries (see Algorithm 1);
- **Greedy rounding** (Bu et al., 2024): Repeatedly, we consider all the $n \times 2$ possible rounding possibilities (first choose an entry $i \in [n]$ and then decide to round it to 0 or 1) and greedily pick the one that gives the lowest value for $\tilde{f}$, until no further rounding can improve $\tilde{f}$, i.e., when reaching a local minimum (see Algorithm 2).

To conclude, over the course of time, more and more powerful derandomization schemes have been proposed, and we have observed that such schemes improve both theoretical guarantees and empirical performance in UCO.

[1] https://github.com/bokveizen/uco_derand

Trial	1	2	3	4	5	Average
Iterative	1968 (39.8%)	1826 (36.9%)	1957 (39.5%)	2094 (42.3%)	1549 (31.3%)	1878.8 (38.0%)
Greedy	1958 (39.6%)	2252 (45.5%)	2299 (46.4%)	1974 (39.9%)	1321 (26.7%)	1960.8 (39.6%)

Table 1: **Empirically, there are many "bad" pairs where the surrogate objective and the final post-derandomization objective give different relative ordering.** For each derandomization scheme (iterative or greedy rounding), we report the number of "bad" pairs among all the 4950 pairs in each random trial, as well as the average number over all trials.

3. Observation: Training-Test Misalignment

Although more powerful derandomization schemes enable better theoretical guarantees and empirical performance, we identify a training-test misalignment issue in existing UCO. In this section, we shall discuss the issue from a methodological perspective and provide empirical evidence.

3.1. Methodological Misalignment

In training, the surrogate objective $\tilde{f} \approx \mathbb{E}_{D\sim\tilde{D}}[f(D)] + \beta \Pr_{D\sim\tilde{D}}[D \notin \mathcal{C}]$ essentially evaluates the expected objective (plus penalty on the probability of violating the constraints) when we obtain D by naive random sampling from $\tilde{D}$. However, at the testing time, existing UCO methods have actually used much more sophisticated derandomization schemes (iterative or greedy rounding).

Although the construction of $\tilde{f}$ guarantees that the optimal $\tilde{D}_{\text{opt}}$ for $\tilde{f}$ corresponds to the optimal D_{opt} for the original objective f, in principle we cannot guarantee the training actually finds the optimum due to the complexity of objective in many CO problems, many of which are even NP-hard.

Therefore, in practice, when the surrogate objective $\tilde{f}$ improves during training, i.e., we obtain new continuous decisions $\tilde{D}_{\text{new}}$ that are better than old ones $\tilde{D}_{\text{old}}$ with $\tilde{f}(\tilde{D}_{\text{new}}) < \tilde{f}(\tilde{D}_{\text{old}})$, the corresponding post-derandomization decisions D_{new} and D_{old} may *not* satisfy $f(D_{\text{new}}) < f(D_{\text{old}})$. That is, a better surrogate objective does not necessarily give better final test-time performance, *even for the training instance without any distribution shift*, resulting in an undesirable training-test misalignment.

3.2. Empirical Evidence: Toy Example

Below, we provide empirical evidence for the methodological misalignment discussed above. We consider a toy example of quadratic functions. Specifically, we consider the objective $f : \{0,1\}^n \mapsto \mathbb{R}$ in the form of $f(D) = \sum_{i,j=1}^{n} \alpha_{ij} d_i d_j$, where $D = (d_1, d_2, \ldots, d_n)$. We consider the simplistic CO problem with objective f and without any constraints (i.e., $\mathcal{C} = \{0,1\}^n$). We can construct the exact expectation $\tilde{f} : [0,1]^n \mapsto \mathbb{R}$ of f, which is $\tilde{f}(\tilde{D}) = \sum_{i,j=1}^{n} \alpha_{ij} \tilde{d}_i \tilde{d}_j$, where $\tilde{D} = (\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_n)$.

Now, we use $n = 50$, sample random α_{ij}'s from i.i.d. normal distributions, and also sample 100 random $\tilde{D}^{(k)}$'s for $k = 1, 2, \ldots, 100$, where each $\tilde{D}^{(k)}$ follows an independent multivariate uniform distribution between 0 and 1 (i.e., each entry in $\tilde{D}^{(k)}$ follows an i.i.d. uniform distribution 0 and 1). For each $\tilde{D}^{(k)}$, we compute its surrogate objective $\tilde{f}^{(k)} := \tilde{f}(\tilde{D}^{(k)})$, its corresponding outputs $D_{\text{iter}}^{(k)}, D_{\text{grd}}^{(k)} \in \{0,1\}^n$ after iterative rounding and greedy rounding respectively, and the corresponding test-time objective $f_{\text{iter}}^{(k)} := f(D_{\text{iter}}^{(k)})$ and $f_{\text{grd}}^{(k)} := f(D_{\text{grd}}^{(k)})$. We have 4950 pairs of $(\tilde{D}^{(k_1)}, \tilde{D}^{(k_2)})$'s in total, and we say a pair $(\tilde{D}^{(k_1)}, \tilde{D}^{(k_2)})$ is "bad" if $(\tilde{f}^{(k_1)} - \tilde{f}^{(k_2)})(f^{(k_1)} - f^{(k_2)})) < 0$, i.e., if the surrogate objective and the final post-derandomization objective give different relatively ordering.

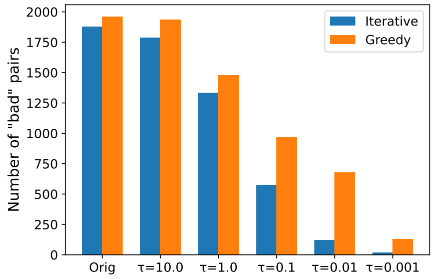
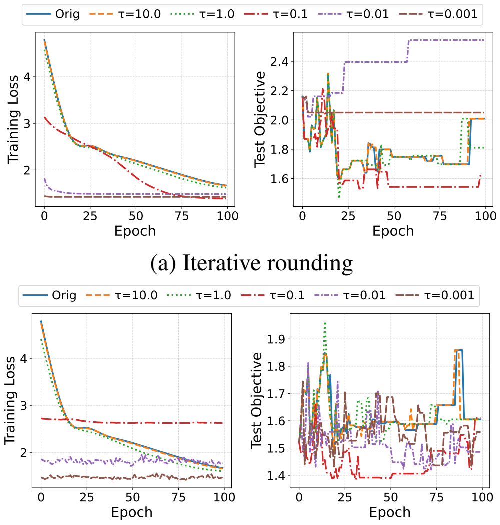


Figure 1: **Our preliminary idea of including soft derandomization improves training-test alignment.** As the soft temperature τ decreases, "bad" pairs reduces.

We repeat the above process in five independent random trials, and report the number of "bad" pairs when we use iterative rounding or greedy rounding. As shown in Table 1, there are many "bad" pairs, even for such a simplistic CO problem where we do not have constraints and we can construct the exact expectation as the surrogate objective, which provides empirical evidence to the methodological misalignment discussed in Section 3.1.

4. Empirical Exploration and Analysis

Now that we have identified and empirically observed this training-test misalignment in UCO, how can we improve UCO methods to have better alignment? Below, we discuss our empirical exploration and analysis, including a preliminary idea to improve training-test alignment in UCO, empirical results, and the challenges we encountered.



(a) Iterative rounding

(b) Greedy rounding

Figure 2: The curves of training loss and test objective (both lower the better) with soft derandomization using different temperatures τ on the facility location problem.

4.1. Preliminary Idea: Soft Derandomization

Recall the methodological misalignment we identified in Section 3.1. The key issue is that the surrogate objective $\tilde{f} \approx \mathbb{E}_{D\sim\tilde{D}}[f(D)] + \beta \Pr_{D\sim\tilde{D}}[D \notin \mathcal{C}]$ essentially assumes naive random sampling for training, while more sophisticated derandomization schemes (iterative or greedy rounding) are actually used at the test time.

Therefore, we propose a preliminary and straightforward idea to improve training-test alignment in UCO, which includes (a soft and differentiable version of) the test-time derandomization scheme also in the training. Specifically, we replace the discrete $\arg\max$ in both iterative rounding (Algorithm 1) and greedy rounding (Algorithm 2) with a soft differentiable $\mathrm{softmax}$. It is easy to see that such soft versions are expected to make training and test phases methodologically more similar, and they approach the original derandomization schemes as the temperature decreases.

4.2. Toy Example

We first revisit the quadratic-function toy example in Section 3.2. Now, for each continuous $\tilde{D}$, we first apply the soft version of iterative or greedy rounding to it, before evaluating it with $\tilde{f}$. We keep all the other settings the same as in Section 3.2, with different softmax temperatures $\tau \in \{10.0, 1.0, 0.1, 0.01, 0.001\}$.

As shown in Figure 1, we see that the soft version of rounding indeed improves the training-test alignment and reduces the "bad" pairs, and lower temperatures give better alignment, validating the correctness of our preliminary idea.

4.3. Typical CO Problem: Facility Location

Although our preliminary idea can indeed improve training-test alignment, that is *not* all we need. Specifically, we need both (1) training-test alignment, i.e., test performance improves as training objective improves, and (2) that training objective actually improves during training. Therefore, now the question is about the second point: With soft derandomization, does the objective improve during training?

We study a typical CO problem used in UCO literature, facility location (Drezner & Hamacher, 2004), where we are given a set of locations and we aim to pick a subset of centers to minimize the total distance from each location to its closest picked center. We follow the experimental settings by (Bu et al., 2024), but only check the training and test *on the same instance*, because we are studying training-test (mis)alignment regarding methodology instead of data distributions, as mentioned in Section 1.

In Figure 2, we show the curves of training surrogate objectives and test performance with soft derandomization using different softmax temperatures. When the temperature τ is too high (e.g., $\tau = 10.0$), soft derandomization is weak and has negligible effects. On the other hand, when τ is too low (e.g., $\tau = 0.01$ or 0.001), the loss surface has higher curvature and less smooth gradients, and the training losses almost do not decrease. On the positive side, we do observe better test performance in some cases with soft derandomization included (see, e.g., the curves for $\tau = 0.1$).

5. Discussion

In this work, we study training-test (mis)alignment in unsupervised combinatorial optimization (UCO). We identify a methodological misalignment issue in existing UCO methods, provide empirical evidence for the issue, and propose a preliminary idea for it. Our idea of including soft derandomization into training appears to be a promising direction for further exploration. Our analysis suggests that the future development of UCO may need to achieve better training-test alignment while maintaining stable training.

Beyond UCO, other machine learning methods for combinatorial optimization have also raised discussions on test-time post-processing (Xia et al., 2024). We believe researchers should be more careful about test-time post-processing in general, especially that they should be aware of the potential danger that too powerful test-time post-processing (compared to the training process) might make the training less relevant. Especially, before addressing generalization regarding data distribution shift (Luo et al., 2023), researchers may need to first address the training-test methodological misalignment to ensure meaningful training.

References

Arumugam, S., Brandstadt, A., Nishizeki, T., and Thulasiraman, K. *Handbook of graph theory, combinatorial optimization, and algorithms*, volume 34. CRC Press New York, 2016.

Bauer, M., Klau, G. W., and Reinert, K. Accurate multiple sequence-structure alignment of rna sequences using combinatorial optimization. *BMC bioinformatics*, 8:1–18, 2007.

Bengio, Y., Lodi, A., and Prouvost, A. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

Berto, F., Hua, C., Park, J., Luttmann, L., Ma, Y., Bu, F., Wang, J., Ye, H., Kim, M., Choi, S., Gast, Z., Hottung, A., Zhou, J., Bi, J., Hu, Y., Liu, F., Kim, H., Son, J., Kim, H., Angioni, D., Kool, W., Cao, Z., Zhang, J., Shin, K., Wu, C., Ahn, S., Song, G., Kwon, C., Xie, L., and Park, J. RL4CO: an extensive reinforcement learning for combinatorial optimization benchmark. In *KDD*, 2025.

Bu, F. and Shin, K. On test-time derandomization in unsupervised combinatorial optimization: Code and datasets. https://github.com/bokveizen/uco_derand, 2025.

Bu, F., Jo, H., Lee, S. Y., Ahn, S., and Shin, K. Tackling prevalent conditions in unsupervised combinatorial optimization: Cardinality, minimum, covering, and more. In *ICML*, 2024.

Cheng, M. X., Li, Y., and Du, D.-Z. *Combinatorial optimization in communication networks*. Springer, 2006.

Drezner, Z. and Hamacher, H. W. *Facility location: applications and theory*. Springer Science & Business Media, 2004.

Erdős, P. and Spencer, J. *Probabilistic methods in combinatorics*. Akadémiai Kindó, 1974.

Hwang, S.-I. and Cheng, S.-T. Combinatorial optimization in real-time scheduling: theory and algorithms. *Journal of combinatorial optimization*, 5:345–375, 2001.

Karalias, N. and Loukas, A. Erdos goes neural: an unsupervised learning framework for combinatorial optimization on graphs. In *NeurIPS*, 2020.

Kool, W., Van Hoof, H., and Welling, M. Attention, learn to solve routing problems! In *ICLR*, 2019.

Li, Z., Chen, Q., and Koltun, V. Combinatorial optimization with graph convolutional networks and guided tree search. In *NeurIPS*, 2018.

Luo, F., Lin, X., Liu, F., Zhang, Q., and Wang, Z. Neural combinatorial optimization with heavy decoder: Toward large scale generalization. In *NeurIPS*, 2023.

Modaresi, S., Sauré, D., and Vielma, J. P. Learning in combinatorial optimization: What and how to explore. *Operations Research*, 68(5):1585–1604, 2020.

Wang, H. P., Wu, N., Yang, H., Hao, C., and Li, P. Unsupervised learning for combinatorial optimization with principled objective relaxation. In *NeurIPS*, 2022.

Xia, Y., Yang, X., Liu, Z., Liu, Z., Song, L., and Bian, J. Position: Rethinking post-hoc search-based neural approaches for solving large-scale traveling salesman problems. In *ICML*, 2024.