Tackling Prevalent Conditions in Unsupervised Combinatorial Optimization: Cardinality, Minimum, Covering, and More

Fanchen Bu[1], Hyeonsoo Jo[1], Soo Yong Lee[1], Sungsoo Ahn[2], Kijung Shin[1]

[1]KAIST: {boqvezen97, hsjo, syleetolow, kijungs}@kaist.ac.kr, [2]POSTECH: {sungsoo.ahn}@postech.ac.kr

Code and Data: *github.com/ai4co/unsupervised-CO-ucom2* [You can also scan the QR code]

Background: UL4CO

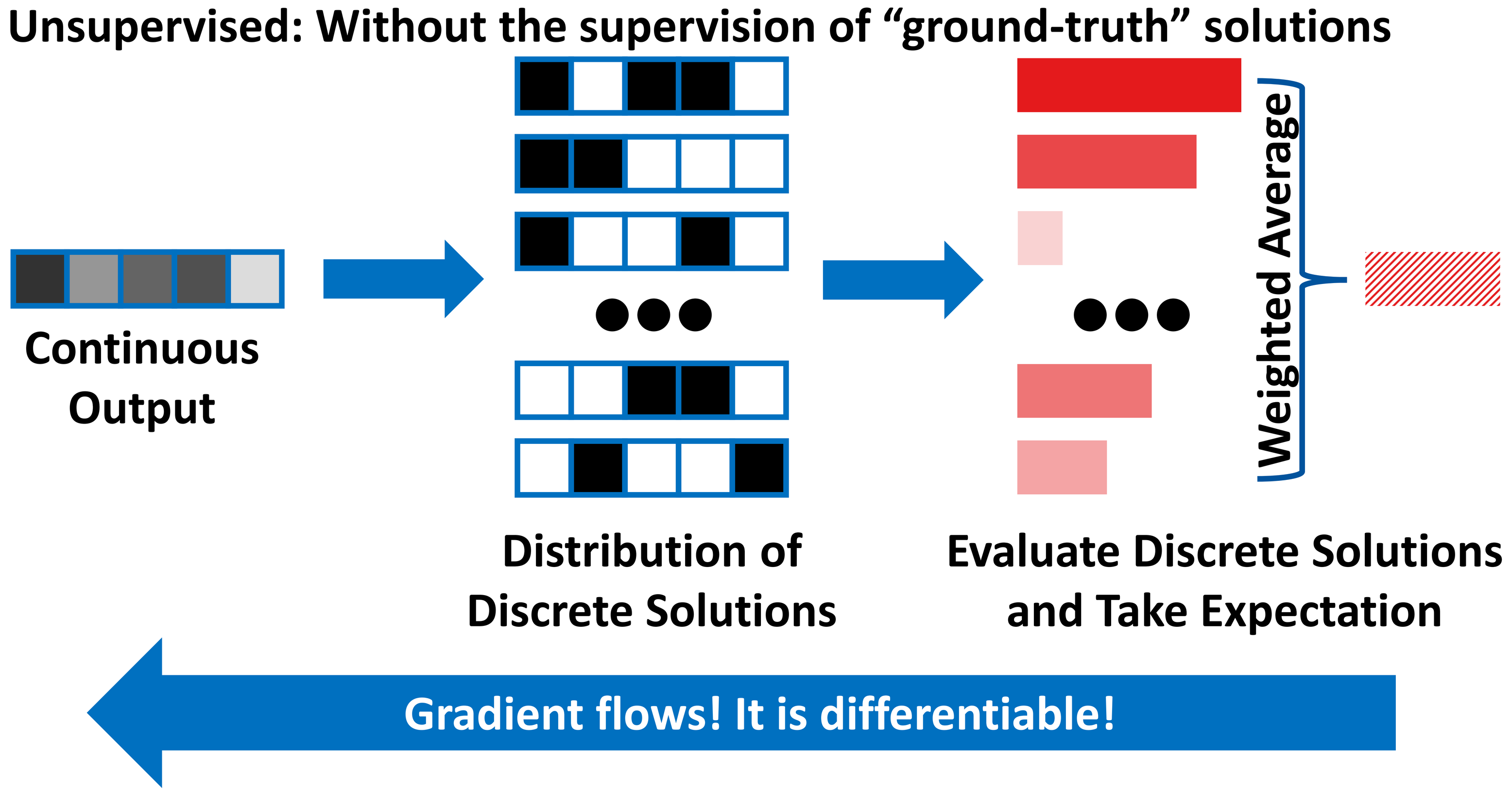
- Combinatorial optimization (CO) is naturally *discrete*
- The output of neural networks is typically *continuous*

Q: How to *differentiably* evaluate continuous outputs for discrete questions?

- Supervision learning is an option, but obtaining "ground-truth" optimal solutions is impractical for NP-hard CO problems or large instances

UL4CO: Unsupervised Learning for Combinatorial Optimization

- (1) Seeing each continuous output as a *distribution* of discrete solutions
- (2) Evaluating the *expectation* of objective on the distribution
- Unsupervised: Without the supervision of "ground-truth" solutions



Two Major Technical Steps and Corresponding Challenges of UL4CO

- (1) Expectation evaluation: Naively evaluating each possible discrete solutions is computationally prohibitive (specifically, exponential)

Q: How to *efficiently* and *accurately* evaluate expected objectives?

- (2) Solution derandomization: What we get is still continuous outputs (i.e., good solution distributions), not directly discrete solutions

Q: How to *efficiently* obtain *good* discrete solutions from learnt outputs?

Concretizing Targets: What Do We Need?

Q: What *mathematical properties* do we need to achieve for both steps?

- (1) Expectation evaluation: There are known good properties, but existing works do not tell us how to satisfy them
  - Good properties: Differentiability, entry-wise concavity, etc.

Our Theorem: All You Need Is the *Expectation of a Tight Upper Bound*

- Tight: Have the same optimum with the original objective

- (2) Solution derandomization: Existing derandomization schemes are either random sampling or iterative rounding
  - Random sampling: Sampling from the learnt distribution; it needs many samples and good luck 🍀
  - Iterative rounding: Determining sub-solutions one by one; performance may highly depend on the order of rounding

Our Proposed Scheme: *Incremental Greedy* Derandomization

- Greedy: Find the best sub-solution to derandomize at each step
- Incremental: Based on the incremental differences after each possible derandomization step, instead of evaluating the whole objective

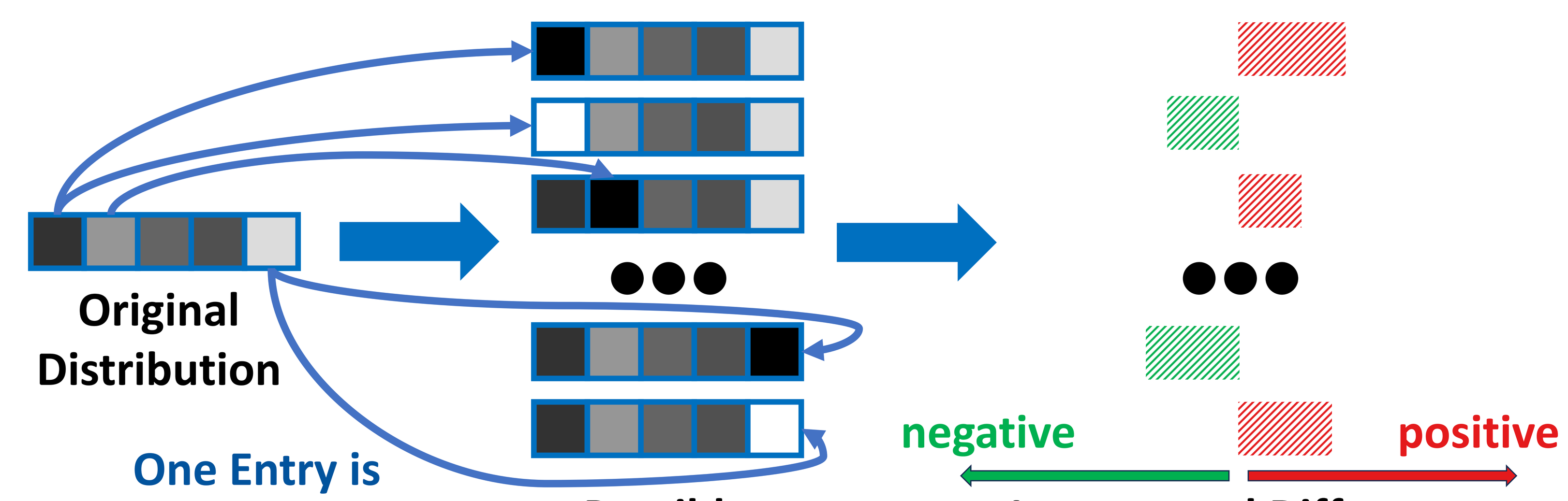
Deriving Formulae for Conditions to Meet the Targets

Roadmap: Targets -> Conditions -> Problems

- With the concrete targets, we (1) *derive detailed formulae* for various prevalent (i.e., commonly-involved) conditions, and (2) *combine these formulae* for different CO problems involving such conditions

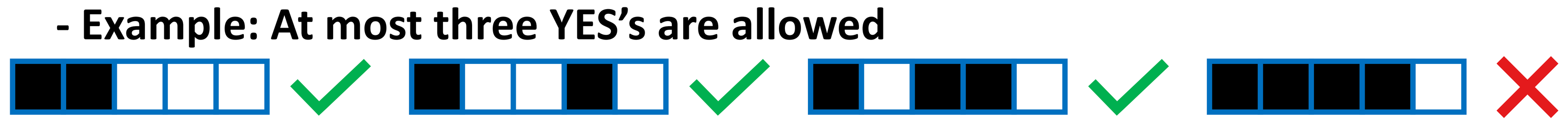
Target -> Conditions: What We Need to Do for Each Condition

- (1) Expectation evaluation: We first find a tight upper bound of the objective, and then derive the expectation of the tight upper bound
- (2) Solution derandomization: We derive its incremental differences, i.e., the change in the objective after each possible derandomization step



Condition: Cardinality Constraints

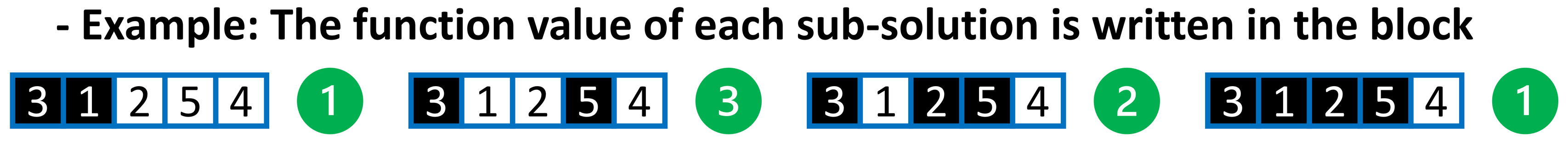
- Definition: The number of YES's (i.e., 1's) should be in a given range
  - Example: At most three YES's are allowed



- Expectation: The cardinality of a given distribution follows a *Poisson binomial* distribution, and we adopt a method based on the *discrete Fourier transform*
- Incremental differences: We use the *recursive formula* of Poisson binomial

Condition: Minimum Within a Subset

- Definition: The minimum of a given function on positive sub-solutions
  - Example: The function value of each sub-solution is written in the block



- Expectation: The minimum is achieved on a sub-solution if and only if
  - (1) the sub-solution is a YES and
  - (2) any other sub-solution with a smaller function value is a NO
- Incremental differences: The change of a sub-solution only affects the sub-solutions with higher function values

See the Main Paper for the Details of Other Conditions

- Covering, cliques, non-binary decisions, and uncertainty

Combine the Conditions for Different Problems

Conditions -> Problems: What We Need to Do for Each Problem

- We first analyze what conditions are involved in the problem
- We then combine the formulae for those conditions

Problem: Facility Location

- Definition: Given a group of locations and a number k, we aim to find k locations as "facilities" so that the summation of distances from each location to its closest facility is minimized
- Involved conditions: (1) Cardinality constraints (k locations should be chosen) and (2) minimum within a subset (the distance to the closest facility is counted for each location)
  - Hence, we just combine the derivations for (1) cardinality constraints and (2) minimum within a subset for the facility location problem

See the Main Paper for the Details of Other Problems

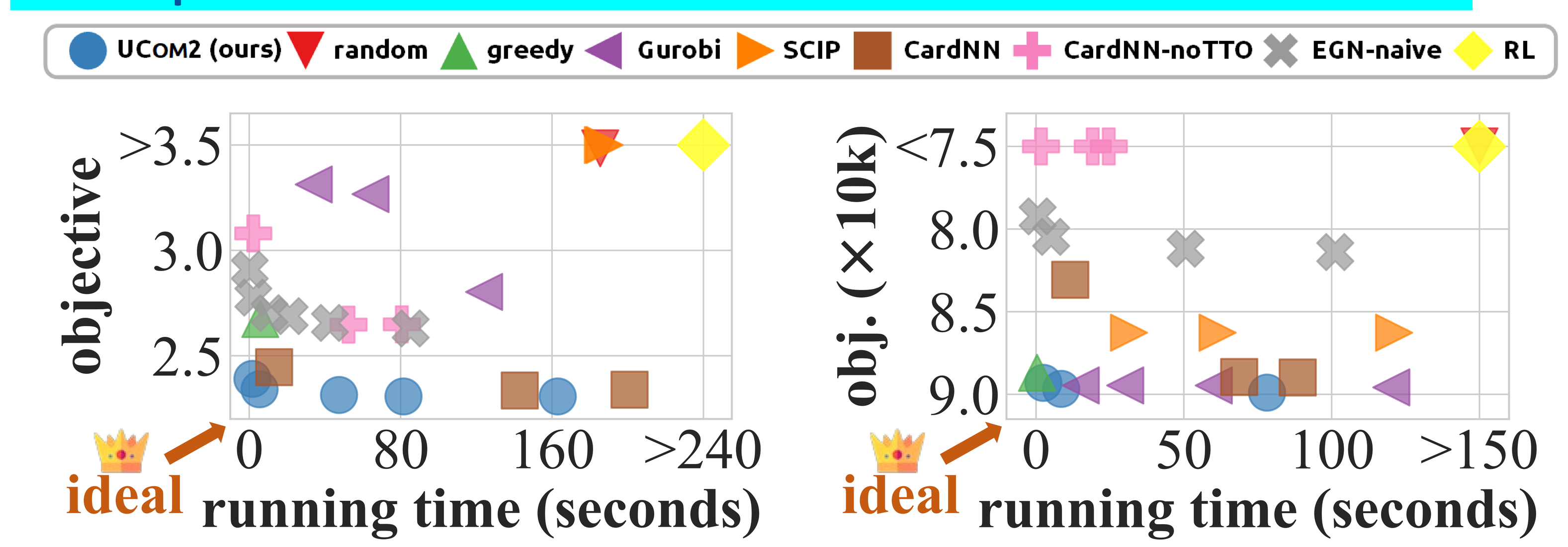
- Maximum coverage: cardinality constraints + covering
- Robust coloring: cliques + uncertainty + non-binary decisions

Experimental Results

We Show the Empirical Usefulness of Our Derivations

- Datasets: Both synthetic datasets and real-world datasets
  - We consider inductive settings, i.e., the models are always trained on synthetic datasets even when tested on real-world datasets
- Baselines: Simple baselines (random and greedy), integer-programming solvers (Gurobi and SCIP), ML methods (CardNN [ICLR'23] and RL)

The Proposed Method Achieves the Best Time-Performance Trade-off



Facility Location: 800 locations | Maximum Coverage: 1000 sets

See the Main Paper for the Full Results

- Results on real-world datasets
- Results on the robust coloring problem

Conclusion and Discussion

Our Main Contributions in This Work

- We mathematically formulated and concretized the targets for UL4CO
- We derived formulae for various conditions to meet the targets
- We applied our derivations to different CO problems

Discussion

- Our targets can be used for guiding the derivations of other conditions
- Our formulae can be used for other problems involving such conditions
- We will also explore such possibilities in the future