Learning-Augmented Streaming Algorithms for Approximating Boolean Max-CSPs

Yinhao Dong* Pan Peng* Ali Vakilian† Santhoshini Velusamy‡ Xiaoyang Xu§

*University of Science and Technology of China †Virginia Tech ‡University of Waterloo §University of California, Santa Barbara

Boolean Maximum Constraint Satisfaction Problems

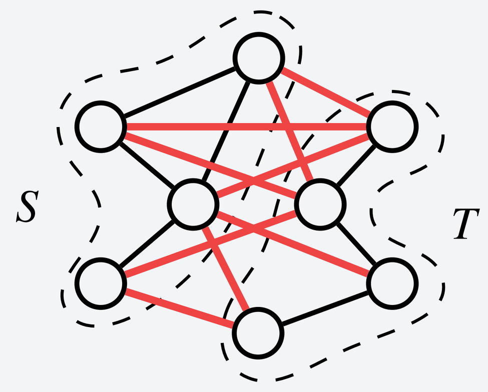
- **Variables:** $x_1, x_2, \ldots, x_n$ taking values in $\{-1, 1\}$.
- **Constraints:** $C = (f, \mathbf{j})$ where $f : \{-1,1\}^k \to \{0,1\}$ and $\mathbf{j} = (j_1, \ldots, j_k) \in \{1, \ldots, n\}^k$.
  Example: $f(a,b) = \neg a \wedge b$ and $\mathbf{j} = (3,8)$, read as $\neg x_3 \wedge x_8$.
- **Value:** Given a Boolean Max-CSP instance $\Psi = \{(f_i, \mathbf{j}_i)\}_{i=1}^m$, its value is the largest number of satisfied constraints over all assignments. Formally,

$$\mathsf{val}_\Psi := \max_{\boldsymbol{\sigma} \in \{-1,1\}^n} |\{(f, \mathbf{j}) \in \Psi : f(\sigma_{j_1}, \ldots, \sigma_{j_k}) = 1\}|.$$

- We are interested in designing algorithms to **estimate** val_Ψ.
- **α-approximation** $(0 < \alpha \le 1)$**:** $\alpha \cdot \mathsf{val}_\Psi \le \widetilde{\mathsf{val}}_\Psi \le \mathsf{val}_\Psi$.

Example 1: Max-CUT as a CSP

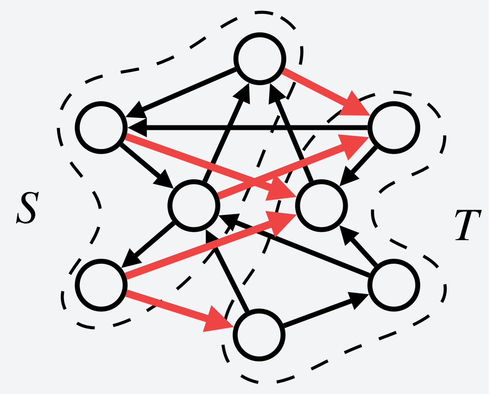
Partition the vertices of an undirected graph into two sides to maximize # of crossing edges.



- **Variables:** $i \in S \Leftrightarrow x_i = 1$, $i \in T \Leftrightarrow x_i = -1$
- **Constraints:** $(i,j) \in E \Leftrightarrow x_i \oplus x_j \in \Psi$

Example 2: Max-DICUT as a CSP

Partition the vertices of a directed graph into two sides to maximize # of edges from S to T.



- **Variables:** $i \in S \Leftrightarrow x_i = 1$, $i \in T \Leftrightarrow x_i = -1$
- **Constraints:** $(i,j) \in E \Leftrightarrow x_i \wedge \neg x_j \in \Psi$

CSPs in the Streaming Model

- **Motivation:** Available memory $\ll$ input size, so the entire input cannot be stored.
- **Stream:** The constraints of Ψ arrive one by one as $\langle C_1, \ldots, C_m \rangle$.
- **Goal:** Estimate val_Ψ in one pass using small space.

Known Results for Streaming Max-CUT and Max-DICUT

- **Max-CUT:** $1/2$-approx, $O(\log n)$ bits of space (trivial).
  Lower Bound: $(1/2 + \eta)$-approx, $\Omega_\eta(n)$ bits of space [KK19].
- **Max-DICUT:** $(1/2 - \eta)$-approx, $n^{1-\Omega_\eta(1)}$ bits of space [ABFS26].
  Lower Bounds: $(4/9 + \eta)$-approx, $\Omega_\eta(\sqrt{n})$ bits of space [CGV20]; $(1/2 + \eta)$-approx, $\Omega_\eta(n)$ bits of space [KK19].

Learning-Augmented Algorithms

- Algorithm has access to a **learned oracle** 🧙, providing a certain type of predictions about the input instance.
- **ε-accurate predictions** $(0 < \varepsilon \le 1/2)$ [CdG+24, GMM25]
  - $\boldsymbol{\sigma}^* \in \{-1,1\}^n$ is some fixed but unknown optimal assignment.
  - Oracle access to a prediction vector $\mathbf{Y} \in \{-1,1\}^n$.
  - Y_i is independently correct with probability slightly better than random guessing: $\Pr[Y_i = \sigma_i^*] = 1/2 + \varepsilon$ for each $i \in \{1, \ldots, n\}$.

References

[ABFS26] Amir Azarmehr, Soheil Behnezhad, Shane Ferrante, and Mohammad Saneian. Half-Approximating Maximum Dicut in the Streaming Setting. In *STOC'26*.

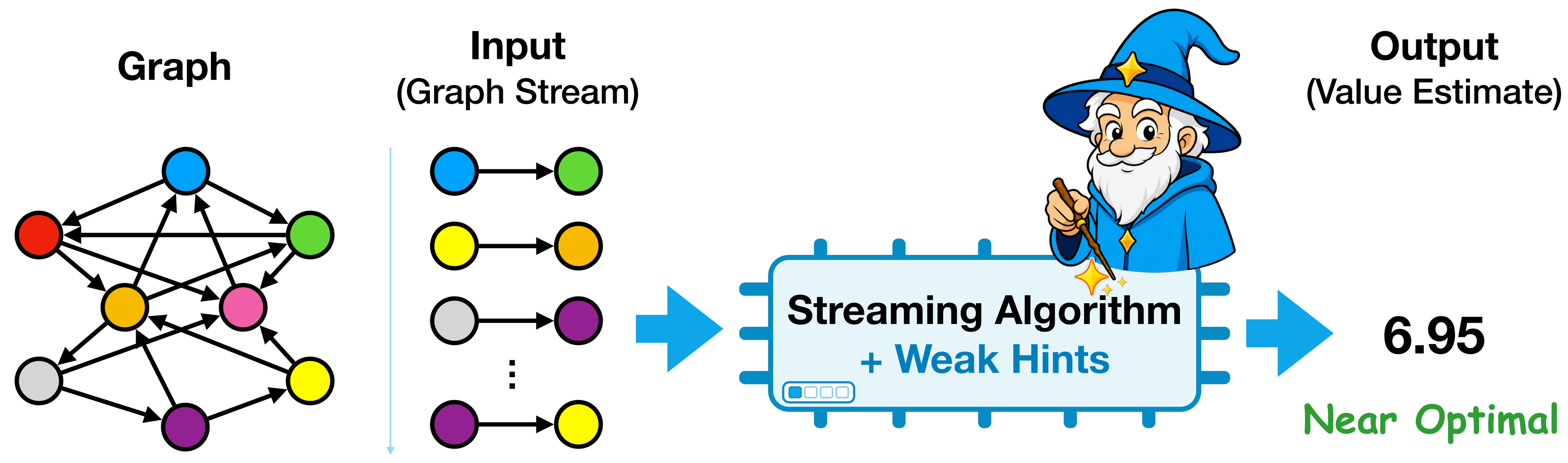
[CdG+24] Vincent Cohen-Addad, Tommaso d'Orsi, Anupam Gupta, Euiwoong Lee, and Debmalya Panigrahi. Learning-Augmented Approximation Algorithms for Maximum Cut and Related Problems. In *NeurIPS'24*.

[CGV20] Chi-Ning Chou, Alexander Golovnev, and Santhoshini Velusamy. Optimal Streaming Approximations for all Boolean Max-2CSPs and Max-kSAT. In *FOCS'20*.

[DPV25] Yinhao Dong, Pan Peng, and Ali Vakilian. Learning-Augmented Streaming Algorithms for Approximating MAX-CUT. In *ITCS'25*.

[GMM25] Suprovat Ghoshal, Konstantin Markarychev, and Yury Markarychev. Constraint Satisfaction Problems with Advice. In *SODA'25*.

[KK19] Michael Kapralov and Dmitry Krachun. An Optimal Space Lower Bound for Approximating MAX-CUT. In *STOC'19*.



Takeaway:
Weak Hints Enable Near-Optimal Value Estimation in Small Space!

What improves over prior work?

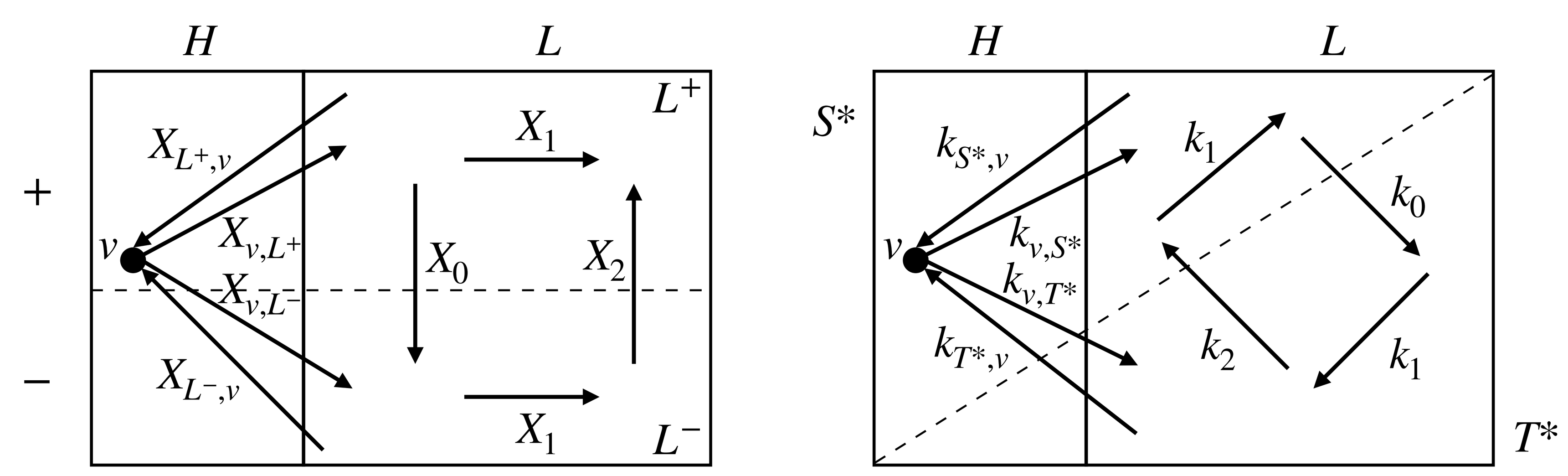
	Prior Work [DPV25]	**This Work**
Problem Class	Max-CUT only	Any Boolean Max-CSP
Approximation	$1/2 + \Omega(\varepsilon^2)$	$1 - \eta$ (arbitrarily close to 1)
Space (words)	$\mathrm{poly}(1/\varepsilon)$	$\mathrm{poly}(1/\varepsilon, 1/\eta)$ (for $k = 2$)

Trade-off: The stronger approximation guarantee introduces an additional $\mathrm{poly}(1/\eta)$ space dependence.

Key Idea: Decoding Noisy Predictions

- Let L and H denote the low- and high-degree vertex sets.
- For Max-CUT, [DPV25] simply follows the predicted labels on vertices in L and handles vertices in H greedily.
- For Max-DICUT, this is not enough... 😑
- But predictions actually reveal richer information! 😄

- $X_0, X_1, X_2 =$ # edges with $(+,-)$, same-sign, $(-,+)$ predictions.
- $k_0, k_1, k_2 =$ # edges from S^* to T^*, inside S^* or T^*, from T^* to S^*.
- **Key Observation:**
  $(\mathbb{E}[X_0], \mathbb{E}[X_1], \mathbb{E}[X_2])$ is a **linear transformation** of (k_0, k_1, k_2).
- **Decoding:** Invert the linear transformation to obtain estimator
  $\widetilde{k}_0 := f_0(\varepsilon) \cdot X_0 + f_1(\varepsilon) \cdot X_1 + f_2(\varepsilon) \cdot X_2$.



- **Low-degree part:** Apply the decoder to $G[L]$.
- **High-degree part:** For each $v \in H$, similarly decode its $S^* \to v$ and $v \to T^*$ contributions, and greedily take the larger one.
- **Extension to Max-CSPs:** Reduce to Max-AND and apply a similar decoding method.
- **Streaming:** Use standard sampling and sketching tools.

Open Problems

- Can a similar theorem be proved in other prediction models?
- Generalize the prediction model to q-ary alphabets?

Meet Us at ICBS 2026

- Yinhao Dong: `yhdong@mail.ustc.edu.cn`
- Xiaoyang Xu: `xiaoyangxu@cs.ucsb.edu`