Smoothing Binary Optimization: A Primal-Dual Perspective

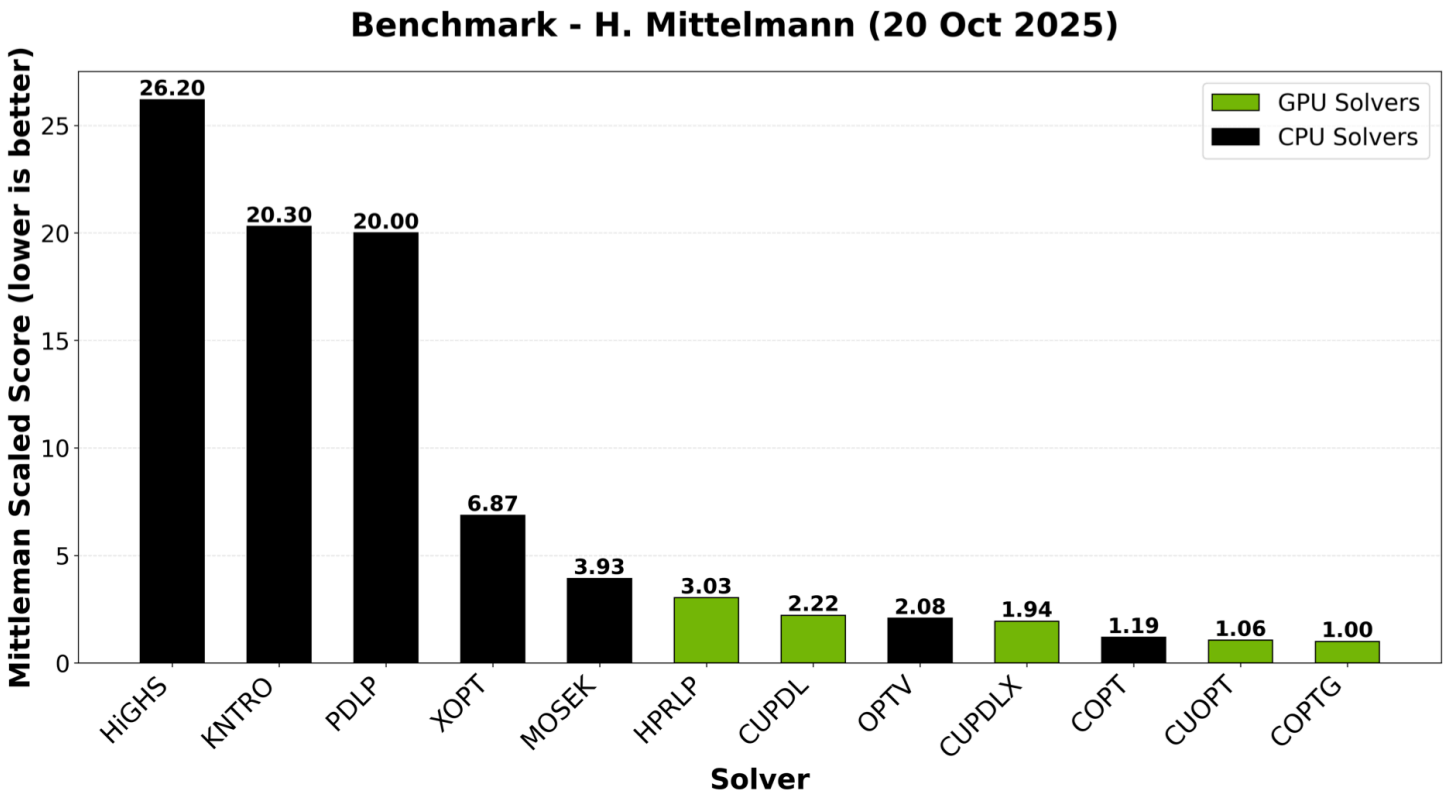
王阿康

wangakang@sribd.cn
深圳市大数据研究院
香港中文大学（深圳）

中国运筹学会数学规划分会青年学者论坛
2026 年 5 月 16 日

Table of Contents

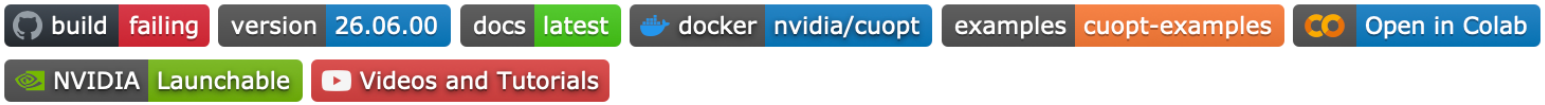
Background



Source: `https://plato.asu.edu/ftp/informs_talks_2025/fender.pdf`

GPU-based first-order methods are HOT!

cuOpt - GPU-accelerated Optimization



NVIDIA® cuOpt™ is a GPU-accelerated optimization engine that excels in mixed integer linear programming (MILP), linear programming (LP), quadratic programming (QP), and vehicle routing problems (VRP). It enables near real-time solutions for large-scale LPs with millions of variables and constraints, and MIPs with hundreds of thousands of variables. cuOpt offers easy integration into existing modeling languages and seamless deployment across hybrid and multi-cloud environments.

The core engine is written in C++ and wrapped with a C API, Python API and Server API.

For the latest version, ensure you are on the `main` branch.

Source: `https://github.com/NVIDIA/cuopt`

Background

来源：中国运筹学会算法与软件分会第三届算法软件与应用大会

Motivation

Can we develop GPU-based algorithms for discrete optimization?

Table of Contents

QUBO

Consider the Quadratic Unconstrained Binary Optimization (QUBO) problem:

$$\min_{x \in \{0,1\}^n} f(x) := x^\top Q x + c^\top x, \tag{1}$$

where $Q_{ii} = 0 \ \forall i$.

QUBO

Consider the Quadratic Unconstrained Binary Optimization (QUBO) problem:

$$\min_{x \in \{0,1\}^n} f(x) := x^\top Q x + c^\top x, \tag{1}$$

where $Q_{ii} = 0 \ \forall i$.

Remarks

- $\mathcal{NP}$-complete
- Integer Linear Programs (approximate)
- Combinatorial Optimization (Max Cut, Maximum Independent Set)

Literature Review

- Meta-heuristics: ABS2 (Nakano et al., 2023)[local search+genetic]
- GNNs:
  - PI-GNN (Schuetz et al., 2022) [*Nature Machine Learning*]
  - ANYCSP (Tönshoff et al., 2023)
  - CRA (Ichikawa, 2024)
  - ROS (Qiu et al., 2025)
- Probabilistic Model:
  - Free Energy Machine (Shen et al., 2025)[Annealing-based, *Nature Computational Science*]
  - MCPG (Chen et al., 2025)[Sampling-based]
- Quantum Annealing (Berwald, 2019)

Smoothing

We reformulate Problem (1) as a constrained continuous optimization problem:

$$\begin{aligned} \min_{x \in [0,1]^n} \quad & f(x) \\ \text{s.t.} \quad & g(x_i) = 0 \quad \forall i \in [n], \end{aligned} \tag{2}$$

where $g : [0, 1] \to \mathbb{R}$ enforces binarity of each variable x_i.

Smoothing

We reformulate Problem (1) as a constrained continuous optimization problem:

$$\begin{aligned} \min_{x\in[0,1]^n} \quad & f(x) \\ \text{s.t.} \quad & g(x_i) = 0 \quad \forall i \in [n], \end{aligned} \tag{2}$$

where $g : [0, 1] \to \mathbb{R}$ enforces binarity of each variable x_i.

Function $g(\cdot)$

- g is strictly convex and continuous on $[0, 1]$, with $g(0) = g(1) = 0$
- g is differentiable on $(0, 1)$

Smoothing

We reformulate Problem (1) as a constrained continuous optimization problem:

$$\begin{aligned} \min_{x\in[0,1]^n} \quad & f(x) \\ \text{s.t.} \quad & g(x_i) = 0 \quad \forall i \in [n], \end{aligned} \tag{2}$$

where $g : [0,1] \to \mathbb{R}$ enforces binarity of each variable x_i.

Function $g(\cdot)$

- g is strictly convex and continuous on $[0,1]$, with $g(0) = g(1) = 0$
- g is differentiable on $(0,1)$

Example

(i) $g(x_i) = x_i^2 - x_i$;

(ii) $g(x_i) = x_i \log(x_i) + (1 - x_i) \log(1 - x_i)$.

Lagrangian Duality

Let $y \in \mathbb{R}^n$ denote the dual variables, then the Lagrangian function is:

$$L(x, y) := f(x) + \sum_{i=1}^{n} y_i g(x_i).$$

Lagrangian Duality

Let $y \in \mathbb{R}^n$ denote the dual variables, then the Lagrangian function is:

$$L(x, y) := f(x) + \sum_{i=1}^{n} y_i g(x_i).$$

Theorem (Strong Max-Min)

$$\inf_{x \in [0,1]^n} \sup_{y \in \mathbb{R}^n} L(x, y) = \sup_{y \in \mathbb{R}^n} \inf_{x \in [0,1]^n} L(x, y).$$

Lagrangian Duality

Let $y \in \mathbb{R}^n$ denote the dual variables, then the Lagrangian function is:

$$L(x, y) := f(x) + \sum_{i=1}^{n} y_i g(x_i).$$

Theorem (Strong Max-Min)

$$\inf_{x \in [0,1]^n} \sup_{y \in \mathbb{R}^n} L(x, y) = \sup_{y \in \mathbb{R}^n} \inf_{x \in [0,1]^n} L(x, y).$$

Problem (2) can be reformulated as a minimax problem with strong max-min property.

$$\min_{x \in [0,1]^n} \max_{y \in \mathbb{R}^n} L(x, y) \tag{3}$$

Gradient Descent-Ascent (GDA)

A classical approach for minimax problems is the GDA method:

$$x^{t+1} \leftarrow \Pi_{[0,1]^n}\left(x^t - \alpha \cdot \nabla_x L(x^t, y^t)\right),$$
$$y^{t+1} \leftarrow y^t + \beta \cdot \nabla_y L(x^t, y^t),$$

Gradient Descent-Ascent (GDA)

A classical approach for minimax problems is the GDA method:

$$x^{t+1} \leftarrow \Pi_{[0,1]^n}\left(x^t - \alpha \cdot \nabla_x L(x^t, y^t)\right),$$
$$y^{t+1} \leftarrow y^t + \beta \cdot \nabla_y L(x^t, y^t),$$

Example (Max-Cut)

Let $f(x) := x^\top \boldsymbol{W} x - \mathbf{1}^\top \boldsymbol{W} x$ and $g(x_i) := x_i^2 - x_i$. Then the fractional solution $x^t := (\frac{1}{2}, \ldots, \frac{1}{2}) \in \mathbb{R}^n$ yields $\nabla_x L(x^t, y^t) = 0$ for any $y^t \in \mathbb{R}^n$.

The example shows that GDA may stabilize at a fractional point.

Algorithm GDA

INPUT: Inital solution $(x^0, y^0) \in [0,1]^n \times \mathbb{R}^n_{++}$, stepsizes $(\alpha, \beta) \in \mathbb{R}^2_{++}$, tolerance $\delta > 0$, number of iterations $T_{\max}$

for $t = 0, 1, 2, \ldots, T_{\max} - 1$ **do**
  for $i = 1, 2, \ldots, n$ **do**
    if $|x_i^t - \frac{1}{2}| \leq \delta$ **and** $|\frac{\partial}{\partial x_i} L(x^t, y^t)| \leq 2\delta$ **and** $y_i^t \leq 0$ **then**

$$x_i^{t+1} \leftarrow \begin{cases} \frac{1}{2} - \delta & \text{if } x_i^t \leq \frac{1}{2}, \\ \frac{1}{2} + \delta & \text{otherwise.} \end{cases}$$

    else

$$x_i^{t+1} \leftarrow \Pi_{[0,1]} \left(x_i^t - \alpha \cdot \frac{\partial}{\partial x_i} L(x^t, y^t) \right)$$

$$y_i^{t+1} \leftarrow y_i^t + \beta \cdot g(x_i^t)$$

Convergence to Binary Points

Proposition (Lower bound of y)

Define $\Theta := \max_{x\in[0,1]^n} ||\nabla_x f(x)||_1$ *and* $b := \frac{1+\Theta}{g'(\frac{1}{2}-\delta)} + \left(2 + \lceil \frac{1}{2\alpha} \rceil\right) \cdot \beta \cdot g(\frac{1}{2})$.

Then for each $i \in [n]$ *and any* $t \geq 0$, *we have* $y_i^t \geq b$.

Note that $y_i^{t+1} - y_i^t = \beta g(x_i^t) \leq 0$, $\{y_i^t\}_{t\geq 0}$ is monotone non-increasing.

Convergence to Binary Points

Proposition (Lower bound of y)

Define $\Theta := \max_{x\in[0,1]^n} ||\nabla_x f(x)||_1$ *and* $b := \frac{1+\Theta}{g'(\frac{1}{2}-\delta)} + \left(2 + \lceil \frac{1}{2\alpha} \rceil\right) \cdot \beta \cdot g(\frac{1}{2})$.

Then for each $i \in [n]$ *and any* $t \geq 0$, *we have* $y_i^t \geq b$.

Note that $y_i^{t+1} - y_i^t = \beta g(x_i^t) \leq 0$, $\{y_i^t\}_{t\geq 0}$ is monotone non-increasing.

Corollary (Convergence of y)

The sequence $\{y^t\}_{t\geq 0}$ *converges to some* $y^* \in \mathbb{R}^n$ *with* $y_i^* \geq b$, $\forall i \in [n]$.

Convergence to Binary Points

Proposition (Lower bound of y)

Define $\Theta := \max_{x\in[0,1]^n} ||\nabla_x f(x)||_1$ *and* $b := \frac{1+\Theta}{g'(\frac{1}{2}-\delta)} + \left(2 + \lceil \frac{1}{2\alpha} \rceil\right) \cdot \beta \cdot g(\frac{1}{2})$.

Then for each $i \in [n]$ *and any* $t \geq 0$, *we have* $y_i^t \geq b$.

Note that $y_i^{t+1} - y_i^t = \beta g(x_i^t) \leq 0$, $\{y_i^t\}_{t\geq 0}$ is monotone non-increasing.

Corollary (Convergence of y)

The sequence $\{y^t\}_{t\geq 0}$ *converges to some* $y^* \in \mathbb{R}^n$ *with* $y_i^* \geq b$, $\forall i \in [n]$.

Note that $g(x_i^t) = \frac{y_i^{t+1} - y_i^t}{\beta} \to 0$

Corollary (Convergence of x to integer points)

For any $i \in [n]$, *the sequence* $\{g(x_i^t)\}_{t\geq 0}$ *converges to* 0.

Convergence Rate

Definition

Given Problem (2), a point $x \in [0,1]^n$ is called an ϵ-binary point if $-\sum_{i=1}^{n} g(x_i) \leq \epsilon$.

Convergence Rate

Definition

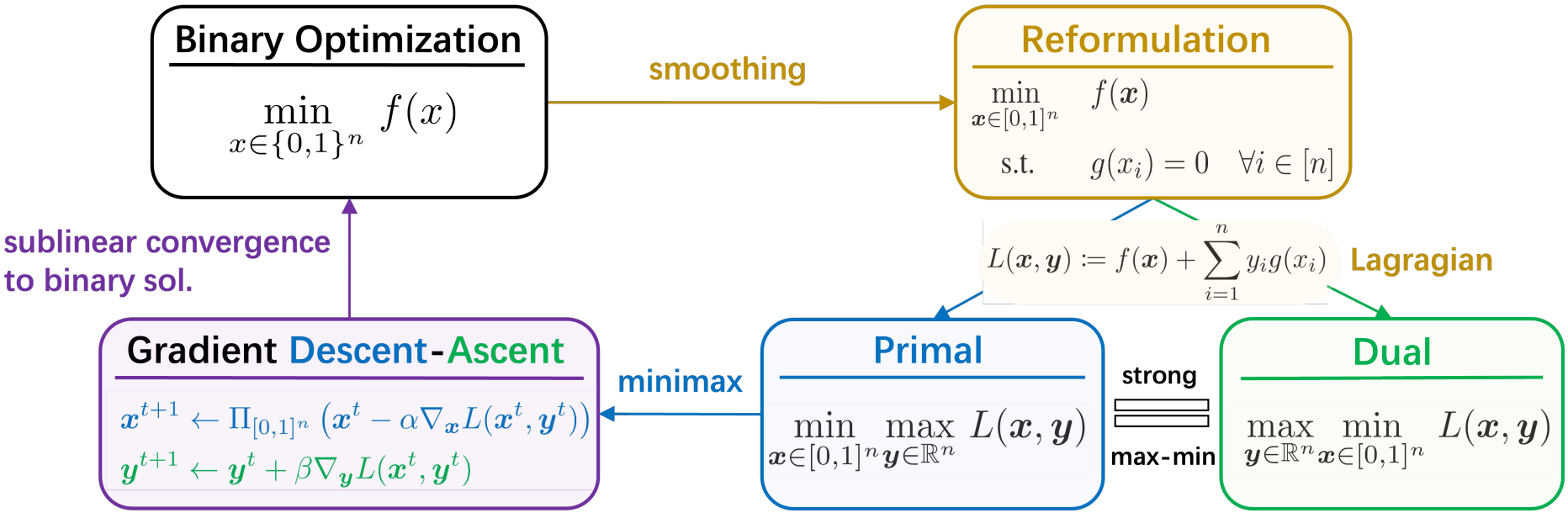
Given Problem (2), a point $x \in [0,1]^n$ is called an ϵ-binary point if $-\sum_{i=1}^{n} g(x_i) \leq \epsilon$.

Theorem

The iteration complexity for Algorithm 1 to return an ϵ-binary point is bounded by

$$\mathcal{O}\left(\frac{\|y^0 - y^*\|_1}{\beta\epsilon}\right)$$

PDBO



We refer to this approach as "a Primal-Dual approach for Binary Optimization" (denoted as PDBO)

Landscape Analysis

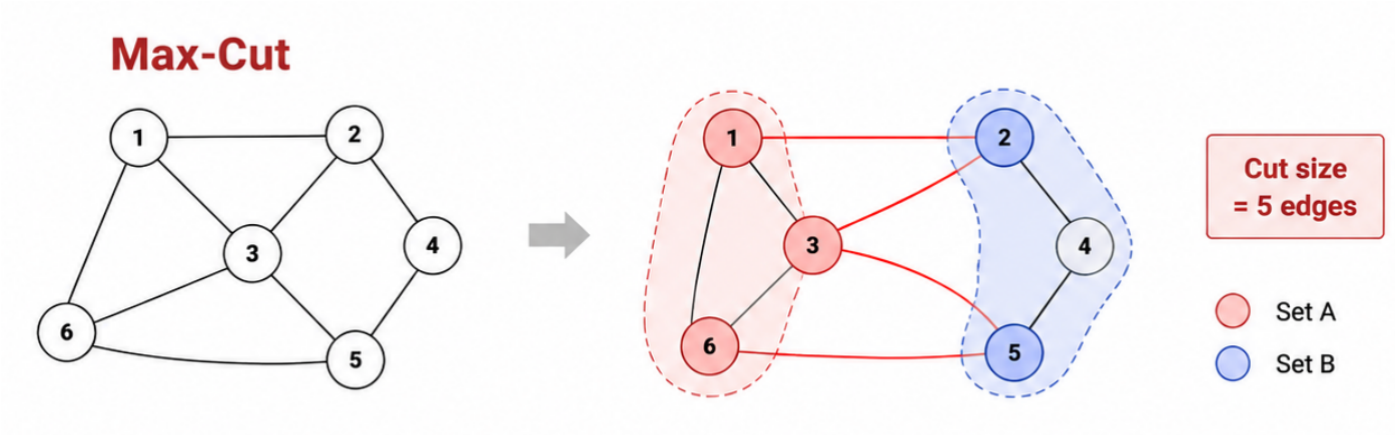
$$\min_{x\in\{0,1\}^n} \; -x^\top Wx + \mathbf{1}^\top Wx \qquad \text{(Max Cut)}$$

$$\nabla_x^2 L(x,y) = 2(W + \operatorname{diag}(y))$$

- **Phase 1. Convex initialization**
  $y^0 = \bar{y}\cdot\mathbf{1}$.Choosing $\bar{y} \geq -\lambda_{\min}(W)$ ensures $\nabla_x^2 L(x,y^0) \succeq 0$
- **Phase 2. Gradual non-convexification**
  $y_i^{t+1} - y_i^t = \beta\cdot\left((x_i^t)^2 - x_i^t\right) \leq 0,\ y_i^t \downarrow$
- **Phase 3. Binary convergence** $y_i(x_i^2 - x_i) = -y_i(x_i - x_i^2)$
  $x_i - x_i^2 \geq 0$ penalizes fractionality, with $-y_i$ increasing.

Numerical Experiments

We evaluate PDBO on Max-Cut instances:



Configuration:

- 180 seconds
- intel i9-12900K CPU + NVIDIA GeForce RTX 3090 GPU
- PDBO is implemented in JAX 0.6.1

Max-Cut

- Obj. denotes the best objective value achieved.
- Time is the time spent for seeking the corresponding solution.

Table: Results on Gset Instances for Max-Cut

Method	G67 (n=10k)		G70 (n=10k)		G72 (n=10k)		G77 (n=14k)		G81 (n=20k)	
	Obj ↑	Time ↓	Obj ↑	Time ↓	Obj ↑	Time ↓	Obj ↑	Time ↓	Obj ↑	Time ↓
PIGNN (Schuetz et al., 2022)	5538	23.7	8534	25.2	5588	44.5	7896	42.1	11078	157.6
CRA (Ichikawa, 2024)	5948	53.7	9240	51.7	6058	53.9	8720	75.9	12450	120.4
ROS (Qiu et al., 2025)	6144	1.5	8872	1.8	6148	1.2	8746	2.2	12320	5.2
ANYCSP (Tönshoff et al., 2023)	6772	39.9	9379	35.7	6816	36.1	9686	53.5	13670	73.5
FEM (Shen et al., 2025)	6782	2.4	5120	0.2	6824	2.6	9688	4.0	13684	7.5
ABS2 (Nakano et al., 2023)	6880	156.3	9510	175.5	6932	172.2	9824	171.1	13850	177.1
Gurobi	**6944**	51.9	9514	133.3	**6990**	171.6	**9882**	175.1	13848	179.8
PDBO	6872	2.5	**9537**	1.9	6906	2.0	9812	2.1	**13852**	3.3

Max-Cut

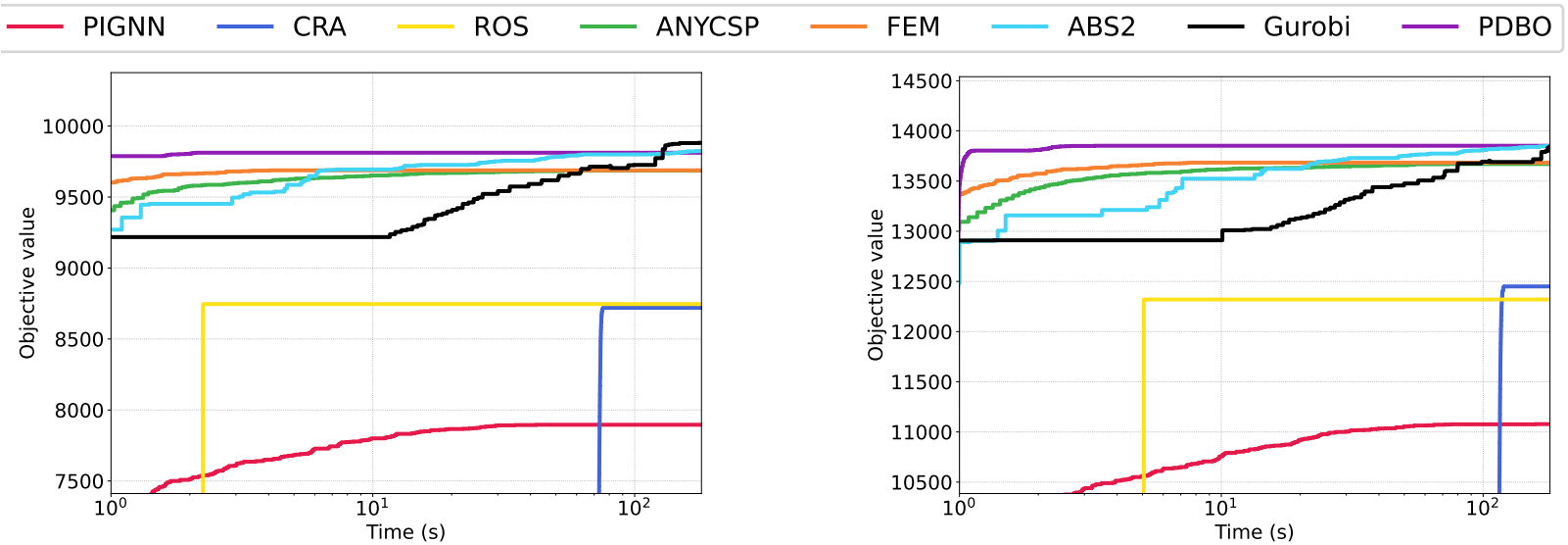


Figure: The objective value of Max-Cut, as a function of runtime.

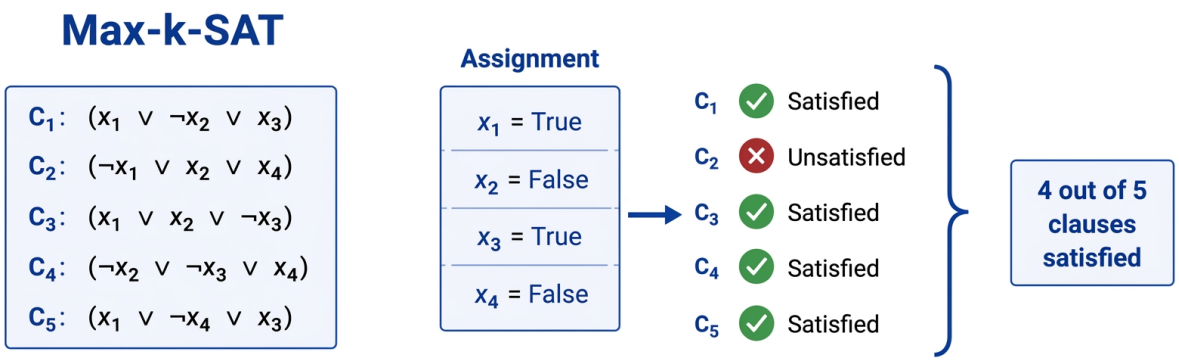
Table of Contents

Extension

Can this framework be extended to other problems?

Extension #1: Max-k-SAT

Given a list of Conjunctive Normal Form (CNF) $\{C_1, C_2, ..., C_5\}$:



$$\max_{x \in \{0,1\}^n} \sum_{j=1}^{m} \prod_{l_i \in C_j} p(x_i),$$

where $\prod_{l_i \in C_j} p(x_i)$ is a multi-linear function, e.g., $1 - (1 - x_1) \cdot x_2 \cdot (1 - x_3)$.

Applications:

- EDA

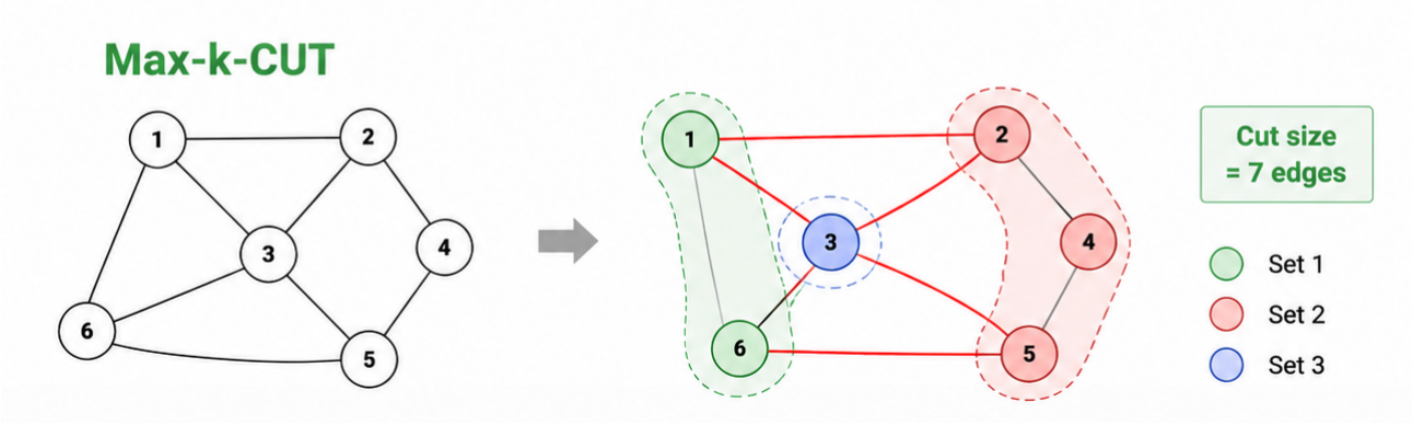
Extension #1: Max-k-SAT

Table: Results on CNF Instances for Max-k-SAT

Method	3CNF		4CNF		5CNF	
	Obj ↓	Time ↓	Obj ↓	Time ↓	Obj ↓	Time ↓
FEM	$1885.2_{\pm 23.9}$	$0.8_{\pm 0.1}$	–	–	–	–
ANYCSP	1583.3 ± 17.5	123.8 ± 40.7	1210.9 ± 12.6	141.6 ± 25.9	1213.7 ± 11.4	141.0 ± 31.9
Gurobi	9322.6 ± 64.2	0.0 ± 0.1	9329.2 ± 69.7	0.0 ± 0.0	9310.7 ± 75.4	0.1 ± 0.0
PDBO	**1582.7** ± 12.0	0.9 ± 0.1	**1147.4** ± 0.8	1.2 ± 0.1	**976.0** ± 9.3	2.2 ± 0.3

- PDBO outperforms other methods across all datasets
- PDBO identifies high-quality solutions within seconds.

Extension #2: Max-k-Cut



$$\begin{aligned} \max_{X \in \{0,1\}^{k \times n}} \quad & \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{i,j} \left(1 - X_{:,i}^{\top} X_{:,j}\right) \\ \text{s.t.} \quad & \sum_{j=1}^{k} X_{j,i} = 1, \quad \forall i \in [n] \end{aligned} \tag{4}$$

Applications:

- PCI
- VLSI

Extension #2: Max-k-Cut

Lagrangian function:

$$L(X, y) := f(X) + \sum_{i=1}^{n} y_i \sum_{j=1}^{k} g(X_{ji}).$$

Theorem (Strong Max-Min)

$$\inf_{X \in \Delta_k^n} \sup_{y \in \mathbb{R}^n} L(X, y) = \sup_{y \in \mathbb{R}^n} \inf_{X \in \Delta_k^n} L(X, y).$$

Problem (4) can be reformulated as a minimax problem.

$$\min_{X \in \Delta_k^n} \max_{y \in \mathbb{R}^n} L(X, y). \tag{5}$$

Extension #2: Max-k-Cut

The strong max-min property enables simultaneous updates as follows:

$$X^{t+1} \leftarrow \Pi_{\Delta_k^n}\left(X^t - \alpha \cdot \nabla_X L(X^t, y^t)\right)$$
$$y^{t+1} \leftarrow y^t + \beta \cdot \nabla_y L(X^t, y^t)$$

Proposition (Convergence to feasible solutions.)

The sequence of iterates generated by the update rules converges to a feasible solution.

$\Pi_{\mathcal{S}}(\cdot)$: Projection

- Euclidean Projection (ℓ_2-norm):
  - Alternating Projection (Boyd and Vandenberghe, 2004)
  - Dykstra's projection algorithm (Dykstra, 1983)

$\Pi_{\mathcal{S}}(\cdot)$: Projection

- Euclidean Projection (ℓ_2-norm):
  - Alternating Projection (Boyd and Vandenberghe, 2004)
  - Dykstra's projection algorithm (Dykstra, 1983)

Issues:

- The trajectory oscillates along the faces of the polytope.

Softmax Reparameterization

$$\min_{X \in \Delta_k^n} \max_{y \in \mathbb{R}^n} L(X, y)$$

Softmax Reparameterization

$$\min_{X \in \Delta_k^n} \max_{y \in \mathbb{R}^n} L(X, y)$$

Given $Z \in \mathbb{R}^{k \times n}$, we adopt the softmax operator:

$$X_{:,i} = \mathrm{SoftMax}(Z_{:,i}) = \left(\frac{\exp(Z_{j,i})}{\sum_{j=1}^{k} \exp(Z_{j,i})} \right)_{j=1,\ldots,k}.$$

Softmax Reparameterization

$$\min_{X \in \Delta_k^n} \max_{y \in \mathbb{R}^n} L(X, y)$$

Given $Z \in \mathbb{R}^{k \times n}$, we adopt the softmax operator:

$$X_{:,i} = \mathrm{SoftMax}(Z_{:,i}) = \left(\frac{\exp(Z_{j,i})}{\sum_{j=1}^{k} \exp(Z_{j,i})} \right)_{j=1,\ldots,k}.$$

Consequently,

$$\min_{Z \in \mathbb{R}^{k \times n}} \max_{y \in \mathbb{R}^n} L(\mathrm{SoftMax}(Z), y).$$

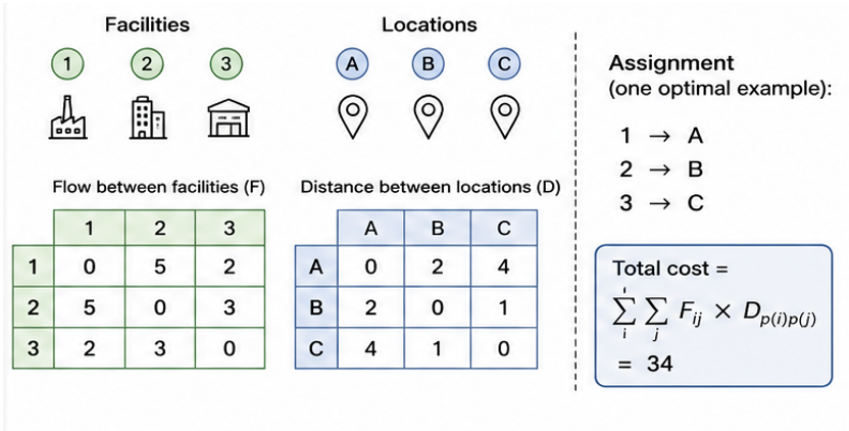
Extension #2: Max-k-Cut

Table: Results on Gset Instances for Max-3-Cut

Method	G67 (n=10k)		G70 (n=10k)		G72 (n=10k)		G77 (n=14k)		G81 (n=20k)	
	Obj ↑	Time ↓	Obj ↑	Time ↓	Obj ↑	Time ↓	Obj ↑	Time ↓	Obj ↑	Time ↓
ROS	7364	3.0	9983	1.9	7435	2.9	10559	5.3	14907	9.7
ANYCSP	7797	55.9	9909	16.2	7906	58.9	11158	84.0	15727	115.0
FEM	7748	3.3	**9999**	1.4	7835	0.7	11102	4.4	15683	6.1
PDBO	**8015**	6.0	**9999**	3.3	**8111**	4.7	**11467**	5.0	**16191**	7.9

- PDBO finds better solutions than the other baselines.

Extension #3: Quadratic Assignment Problem (QAP)



$$\min_{X \in \mathcal{P}_n} f(X) := \mathrm{tr}(FXDX^\top) + \mathrm{tr}(C^\top X), \tag{6}$$

where $\mathcal{P}_n := \left\{ X \in \{0,1\}^{n \times n} \mid X\mathbf{1} = X^\top \mathbf{1} = \mathbf{1} \right\}$.

Applications:

- Facility location
- Graph matching

Extension #3: QAP

$$\min_{X \in \mathcal{D}_n} \max_{Y \in \mathbb{R}^{n \times n}} L(X, Y),$$

where $\mathcal{D}_n := \left\{ X \in \mathbb{R}_+^{n \times n} \mid X\mathbf{1} = X^\top \mathbf{1} = \mathbf{1} \right\}$ represents a Birkhoff polytope.

Sinkhorn Operator $\mathcal{S}(\cdot)$

Given $Z \in \mathbb{R}^{n \times n}$, consider the Sinkhorn operator $\mathcal{S}(Z)$:

$$\begin{aligned} \mathcal{S}^0 &= \exp(Z) \\ \mathcal{S}^{k+1} &:= \mathcal{T}_c(\mathcal{T}_r(\mathcal{S}^k)) \\ \mathcal{S}(Z) &:= \lim_{k \to \infty} \mathcal{S}^k(Z) \end{aligned}$$

where

$$[\mathcal{T}_r(M)]_{ij} := \frac{M_{ij}}{\sum_{l=1}^n M_{il}}, \quad [\mathcal{T}_c(M)]_{ij} := \frac{M_{ij}}{\sum_{l=1}^n M_{lj}}.$$

Sinkhorn Operator $\mathcal{S}(\cdot)$

Given $Z \in \mathbb{R}^{n\times n}$, consider the Sinkhorn operator $\mathcal{S}(Z)$:

$$\begin{aligned}\mathcal{S}^0 &= \exp(Z)\\ \mathcal{S}^{k+1} &:= \mathcal{T}_c(\mathcal{T}_r(\mathcal{S}^k))\\ \mathcal{S}(Z) &:= \lim_{k\to\infty} \mathcal{S}^k(Z)\end{aligned}$$

where

$$[\mathcal{T}_r(M)]_{ij} := \frac{M_{ij}}{\sum_{l=1}^n M_{il}}, \quad [\mathcal{T}_c(M)]_{ij} := \frac{M_{ij}}{\sum_{l=1}^n M_{lj}}.$$

Lemma (Sinkhorn's Theorem (Sinkhorn, 1964))

For any $Z \in \mathbb{R}^{n\times n}$, $\mathcal{S}(Z) \in \mathcal{D}_n$.

Sinkhorn Reparameterization

Problem (6) is transformed into an unconstrained minimax problem:

$$\min_{Z \in \mathbb{R}^{n \times n}} \max_{Y \in \mathbb{R}^{n \times n}} L(\mathcal{S}(Z), Y). \tag{7}$$

Sinkhorn Reparameterization

Problem (6) is transformed into an unconstrained minimax problem:

$$\min_{Z \in \mathbb{R}^{n \times n}} \max_{Y \in \mathbb{R}^{n \times n}} L(\mathcal{S}(Z), Y). \tag{7}$$

In practice, we perform K iterations within the Sinkhorn operator, $\mathcal{S}_K(\cdot)$:

$$Z^{t+1} \leftarrow Z^t - \alpha \cdot \nabla_Z L(\mathcal{S}_K(Z^t), Y^t)$$
$$Y^{t+1} \leftarrow Y^t + \beta \cdot \nabla_Y L(\mathcal{S}_K(Z^t), Y^t)$$

QAPLIB

- n changes from 12 to 256
- time limit: 100 sec

Instance	Gurobi	SM	IPFP	RRWM	NGM	SAWT	PDBO
bur	0.5%	22.4%	9.6%	23%	8.4%	4.0%	**0.0%**
chr	**1.5%**	443.7%	377.4%	600.3%	360.7%	147.5%	2.3%
els	**0.0%**	96.4%	47.4%	355.8%	88.4%	47.4%	0.6%
esc	8.6%	284%	188.2%	58.1%	214.8%	43.4%	**0.0%**
had	0.2%	17.4%	12.1%	25.2%	14.5%	5.2%	**0.0%**
kra	6.1%	68%	43.1%	55.8%	45.3%	32.9%	**0.0%**
lip	6.9%	17.6%	1.9%	18.5%	9.6%	1.4%	**0.3%**
nug	2.2%	44.5%	33.9%	66.4%	35.3%	19.3%	**0.0%**
rou	1.3%	35.8%	24.1%	50.9%	26.6%	15.1%	**0.0%**
scr	**0.0%**	107.8%	63.9%	143.7%	68.6%	33.9%	**0.0%**
sko	14.0%	24.5%	20.3%	32.9%	20.7%	16.2%	**0.4%**
ste	7.4%	417.7%	131.1%	548.6%	121.2%	108%	**0.3%**
tai	10.5%	109%	105.9%	135.8%	108.1%	34.7%	**0.8%**
tho	10.9%	47.6%	34%	58.4%	34.8%	24.1%	**0.3%**
wil	7.5%	14.2%	12%	17.7%	12.8%	9.5%	**0.2%**
Avg. Gap	5.2%	116.7%	73.7%	146.1%	78.0%	36.2%	**0.3%**
Avg. Time (s)	82.7	0.2	**0.1**	0.3	0.4	14.7	9.8

PDBO consistently achieves the smallest gap across most instance families

- Time limit: 1800 sec
- Baselines:
  - Ro-TS (Taillard, 1991): A trajectory-based heuristic that enhances standard tabu search by implementing a dynamic tabu tenure mechanism
  - BMA (Benlic and Hao, 2015): A hybrid evolutionary approach that synergizes global population-based search with aggressive local improvement procedures

Instance	**BKS**	Ro-TS		BMA		PDBO	
		Obj. ↓	Time (s) ↓	Obj. ↓	Time (s) ↓	Obj. ↓	Time (s) ↓
tai125e01	35,426	39,746	320	37,992	1,013	**36,788**	70
tai125e02	36,178	39,354	592	**36,830**	1,000	38,286	72
tai125e03	30,498	35,138	133	33,500	1,553	**32,220**	70
tai175e01	57,540	65,420	355	**60,648**	1,800	63,272	161
tai175e02	50,110	66,142	118	54,220	1,756	**53,000**	154
tai175e03	53,900	218,360	829	59,108	1,740	**55,832**	137
tai343e01	141,048	177,228	4	161,526	1,714	**148,714**	403
tai343e02	148,584	183,594	12	170,858	1,800	**153,116**	423
tai343e03	142,092	176,884	3	163,758	1,768	**145,888**	399
tai729e01	416,260	883,310	473	499,166	809	**445,612**	615
tai729e02	422,570	913,870	129	505,600	823	**450,528**	618
tai729e03	405,004	497,770	76	480,546	1,596	**422,500**	636

Conclusions

- We introduce a primal-dual framework that reformulates QUBO/Max-k-SAT as a continuous minimax problem, enabling efficient and highly parallelizable gradient-based solutions on GPUs
- We extend this primal-dual framework to address Max-k-Cut/QAPs via reparameterization
- We demonstrate the superior scalability and robustness of our proposed framework, achieving significant improvements over state-of-the-art baselines.

Future work:

– Extend this framework to other variants in discrete optimization.

Acknowledgements

Collaborators:

- UCAS: Wenbo Liu, Dun Ma, Wenguo Yang
- CUHKSZ: Jinxin Xiong, Rui Chen, Ruoyu Sun
- CityUHK: Hongyi Jiang
- SRIBD: Jianghua Wu, Xiaodong Luo

Thank You!

References I

Una Benlic and Jin-Kao Hao. Memetic search for the quadratic assignment problem. *Expert Systems with Applications*, 42(1): 584–595, 2015.

Jesse J Berwald. The mathematics of quantum-enabled applications on the d-wave quantum computer. *Not. Am. Math. Soc*, 66 (832):55, 2019.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Cheng Chen, Ruitao Chen, Tianyou Li, Ruicheng Ao, and Zaiwen Wen. A monte carlo policy gradient method with local search for binary optimization. *Mathematical Programming*, pages 1–57, 2025.

Richard L Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78 (384):837–842, 1983.

Yuma Ichikawa. Controlling continuous relaxation for combinatorial optimization. *Advances in Neural Information Processing Systems*, 37:47189–47216, 2024.

Koji Nakano, Daisuke Takafuji, Yasuaki Ito, Takashi Yazane, Junko Yano, Shiro Ozaki, Ryota Katsuki, and Rie Mori. Diverse adaptive bulk search: a framework for solving qubo problems on multiple gpus. In *2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 314–325. IEEE, 2023.

Yeqing Qiu, Xue Ye, Akang Wang, Yiheng Wang, Qingjiang Shi, and Zhi-Quan Luo. Ros: A gnn-based relax-optimize-and-sample framework for max-k-cut problems. In *Forty-second International Conference on Machine Learning*, 2025.

Martin JA Schuetz, J Kyle Brubaker, and Helmut G Katzgraber. Combinatorial optimization with physics-inspired graph neural networks. *Nature Machine Intelligence*, 4(4):367–377, 2022.

Zi-Song Shen, Feng Pan, Yao Wang, Yi-Ding Men, Wen-Biao Xu, Man-Hong Yung, and Pan Zhang. Free-energy machine for combinatorial optimization. *Nature Computational Science*, pages 1–11, 2025.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

Éric Taillard. Robust taboo search for the quadratic assignment problem. *Parallel computing*, 17(4-5):443–455, 1991.

Jan Tönshoff, Berke Kisin, Jakob Lindner, and Martin Grohe. One model, any csp: Graph neural networks as fast global search heuristics for constraint satisfaction. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 4280–4288. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track.