DNN-based Branch-and-bound for the Quadratic Assignment Problem

*Koichi Fujii, Naoki Ito, Yuji Shinano

NTT DATA Mathematical Systems Inc., , FAST RETAILING CO., LTD., Zuse Institute Berlin

2019/03/29

Introduction of NTT Data Mathematical Systems Inc.

Will be introduced at next talk : Takahito Tanabe "Implementation issues of Interior-Point Method for real-world NLP problems"

Summary : DNN-based Branch-and-bound for the Quadratic Assignment Problem

Motivation

- Quadratic assignment problems still remain as one of the most difficult combinatorial problems
- Recent conic relaxation technique DNN updates the lower bounds of quadratic assignment problem

Goal

Improve branch-and-bound method for quadratic assignment problems

Our Results

First implementation of DNN-based branch-and-bound

Agenda

Quadratic Assignment Problem

$$\min \left\{ \boldsymbol{x}^T (\boldsymbol{B} \otimes \boldsymbol{A}) \boldsymbol{x} \mid \boldsymbol{x} \in \{0,1\}^n, (\boldsymbol{I} \otimes \boldsymbol{e}^T)\boldsymbol{x} = (\boldsymbol{e}^T \otimes \boldsymbol{I})\boldsymbol{x} = e, x_i x_j = 0 \right\},$$

where $\boldsymbol{B} \otimes \boldsymbol{A}$ denotes the kronecker product of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

Known as having week LP/QP relaxation.

Quadratic Assignment Problem as Polynomial Optimization Problem

Relax linear constraints by Lagrangian multiplier λ.

$$\min \left\{ \boldsymbol{x}^T (\boldsymbol{B} \otimes \boldsymbol{A}) \boldsymbol{x} + \lambda \frac{\|\boldsymbol{B} \otimes \boldsymbol{A}\|}{\|\boldsymbol{D}\|} \tilde{\boldsymbol{x}}^T \boldsymbol{D} \tilde{\boldsymbol{x}} \middle| \boldsymbol{x} \in [0,1]^n, x_i x_j = 0, \tilde{\boldsymbol{x}} = [1; \boldsymbol{x}] \right\},$$

where

$$\boldsymbol{D} := \begin{pmatrix} \boldsymbol{d}^T \boldsymbol{d} & -\boldsymbol{d}^T \boldsymbol{C} \\ -\boldsymbol{C}^T \boldsymbol{d} & \boldsymbol{C}^T \boldsymbol{C} \end{pmatrix} \tag{3}$$

$$\boldsymbol{C} := \boldsymbol{I} \otimes \boldsymbol{e}^T \tag{4}$$

$$\boldsymbol{d} := [\boldsymbol{e}; \boldsymbol{e}] \tag{5}$$

Quadratic Assignment Problem and DNN relaxation

Polynomial optimization problem (POP) with non-negative variables

$\min_x\{f_0(x) \mid f_i(x) = 0 \ (i = 1, 2, \ldots, m), x \geq 0\}$

- 0-1 binary quadratic optimization problem
- Optimal power flow, sensor network localization, ...

Doubly non-negative (DNN) relaxation

SDP relaxation + non-negative constraints

- better lower bounds than SDP
- very large $O(n^2)$ non-negative constraints

$\xrightarrow{\text{POP}}$ **DNN relaxation** $\longrightarrow$ **BP method** $\xrightarrow{\text{lower bound}}$

BBCPOP improved lower bounds for QAPLIB instances.

Quadratic Assignment Problem and DNN relaxation

DNN optimization problem

$$\min_{Z} \{\langle F_0,\ Z\rangle \mid \langle H_0,\ Z\rangle = 1, Z \in \mathbb{K}_1 \cap \mathbb{K}_2\}$$

where

- $F_0 \in \mathbb{S}^n$ and $H_0 \in \mathbb{S}^n_+$ $(i = 1, 2, \ldots, m)$
- $\mathbb{K}_1 = \mathbb{S}^n_+$ and $\mathbb{K}_2 \subseteq \mathbb{S}^n_{\geq 0}$ are convex cones

- $\mathbb{S}^n$: space of symmetric matrices
- $\mathbb{S}^n_+$: space of symmetric positive semidefinite matrices
- $\mathbb{S}^n_{\geq 0}$: space of symmetric nonnegative matrices

Quadratic Assignment Problem and DNN relaxation

$$\min \left\{ \boldsymbol{x}^t \boldsymbol{Q} \tilde{\boldsymbol{x}} \middle| \boldsymbol{x} \in [0,1]^n, x_i x_j = 0 \ ((i,j) \in \Gamma), \tilde{\boldsymbol{x}} = [1; \boldsymbol{x}] \right\}, \qquad (6)$$

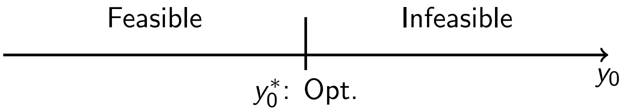
$\Downarrow$ DNN relaxation

$$\{\langle \boldsymbol{Q}, \ \boldsymbol{Z} \rangle \mid \boldsymbol{Z}_{00} = 1, \boldsymbol{Z} \in \mathbb{K}_1 \cap \mathbb{K}_2\} \qquad (7)$$

$$\mathbb{K}_2 \quad := \quad \left\{ \boldsymbol{Z} \in \mathbb{S}^{n+1} \middle| \begin{array}{ll} \boldsymbol{Z}_{\alpha\beta} \geq 0 & \text{nonnegativity} \\ \boldsymbol{Z}_{0\alpha} = \boldsymbol{Z}_{\alpha 0} \geq \boldsymbol{Z}_{\alpha\alpha} & \\ \boldsymbol{Z}_{\alpha\beta} = 0 & \text{if } (\alpha, \beta) \in \Gamma \end{array} \right\} (8)$$

DNN Optimization : BP method [Kim, Kojima, & Toh, '16]

$\min_Z \{\langle F_0,\ Z\rangle \mid \langle H_0,\ Z\rangle = 1, Z \in \mathbb{K}_1 \cap \mathbb{K}_2\}$
$\Updownarrow$ Strong duality
$\max_{y_0} \{y_0 \mid \underbrace{\boldsymbol{F}_0 - y_0 \boldsymbol{H}_0}_{\boldsymbol{G}(y_0)} \in \mathbb{K}_1^* + \mathbb{K}_2^*\}$



BP method : Bisection method to judge the feasibility of a point y_0

DNN Optimization : BP Method [Kim, Kojima, & Toh, '16]

How to judge if $\boldsymbol{G}(y_0) \in \mathbb{K}_1^* + \mathbb{K}_2^*$? $\Rightarrow$ solve regression model

$$
\begin{aligned}
f^* &= \min_{\boldsymbol{Y}_1, \boldsymbol{Y}_2} \{ \|\boldsymbol{G} - (\boldsymbol{Y}_1 + \boldsymbol{Y}_2)\|^2 \mid \boldsymbol{Y}_1 \in \mathbb{K}_1^*, \ \ \boldsymbol{Y}_2 \in \mathbb{K}_2^* \} \\
&= \min_{\boldsymbol{Y}_1} \{ \min_{\boldsymbol{Y}_2} \{ \|(\boldsymbol{G} - \boldsymbol{Y}_1) - \boldsymbol{Y}_2\|^2 \mid \boldsymbol{Y}_2 \in \mathbb{K}_2^* \} \mid \boldsymbol{Y}_1 \in \mathbb{K}_1^* \} \\
&= \min_{\boldsymbol{Y}_1} \{ \|(\boldsymbol{G} - \boldsymbol{Y}_1) - \Pi_{\mathbb{K}_2^*}(\boldsymbol{G} - \boldsymbol{Y}_1))\|^2 \mid \boldsymbol{Y}_1 \in \mathbb{K}_1^* \} \\
&= \min_{\boldsymbol{Y}_1} \{ \|\Pi_{\mathbb{K}_2}(\boldsymbol{Y}_1 - \boldsymbol{G})\|^2 \mid \boldsymbol{Y}_1 \in \mathbb{K}_1^* \} \ \ (\text{where } \boldsymbol{Y}_2 = \Pi_{\mathbb{K}_2^*}(\boldsymbol{G} - \boldsymbol{Y}_1))
\end{aligned}
$$

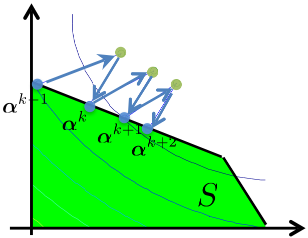
- Obviously, $f^* = 0 \Leftrightarrow \boldsymbol{G} \in \mathbb{K}_1^* + \mathbb{K}_2^*$.
- Apply accelerated proximal gradient (APG) to check if $f^* = 0$.
  $\rightarrow$ [Assumption 1] $\Pi_{\mathbb{K}_2}, \Pi_{\mathbb{K}_1}$ can be computed efficiently.

DNN Optimization : APG method

Constrained optimization: $\min_{\boldsymbol{\alpha} \in S} f(\boldsymbol{\alpha})$

Gradient projection method (e.g., [Goldstein, '64])

Step 1: $\boldsymbol{\alpha}^{k+1} = \Pi_S\left(\boldsymbol{\alpha}^k - \frac{1}{L_k}\nabla f(\boldsymbol{\alpha}^k)\right)$

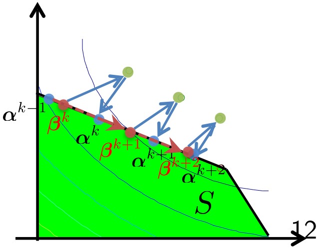


APG method [Beck and Teboulle, '09]

Step 1: $\boldsymbol{\alpha}^k = \Pi_S\left(\boldsymbol{\beta}^k - \frac{1}{L_k}\nabla f(\boldsymbol{\beta}^k)\right)$

Step 2: $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$

Step 3: $\boldsymbol{\beta}^{k+1} = \boldsymbol{\alpha}^k + \underbrace{\frac{t_k - 1}{t_{k+1}}(\boldsymbol{\alpha}^k - \boldsymbol{\alpha}^{k-1})}_{\text{momentum}}$



DNN Optimization : APG method

Gradient projection method

$$\underbrace{f(\boldsymbol{\alpha}^k)}_{\text{current}} - \underbrace{f(\boldsymbol{\alpha}^*)}_{\text{opt.}} \leq O(1/k)$$

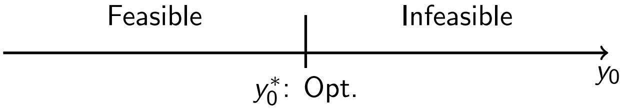
Accelerated proximal gradient (APG) method

$$f(\boldsymbol{\alpha}^k) - f(\boldsymbol{\alpha}^*) \leq O(1/k^2)$$

e.g., [Beck and Teboulle, '09] [Nesterov, '03]

DNN Optimization : Computing a Valid Lower Bound

- BP method may output an UB of the opt. val. (infeasible sol.), because APG can fail to judge feasibility due to numerical error.
- Can we compute a valid lower bound y_0^ℓ of DNN?



CDNN Optimization : Computing a Valid Lower Bound

[Arima, Kim, Kojima & Toh, '17]

$$\min_{\boldsymbol{Z}} \left\{ \langle \boldsymbol{F}_0, \ \boldsymbol{Z} \rangle \ \middle| \ \langle \boldsymbol{H}_0, \ \boldsymbol{Z} \rangle = 1, \boldsymbol{Z} \in \mathbb{K}_1 \cap \mathbb{K}_2 \ \right\}.$$

$\Updownarrow$ with $\boldsymbol{I} \in \mathbb{K}_1$ and large enough $\rho \geq 0$

$$\min_{\boldsymbol{Z}} \{ \langle \boldsymbol{F}_0, \ \boldsymbol{Z} \rangle \mid \langle \boldsymbol{H}_0, \ \boldsymbol{Z} \rangle = 1, \ \langle \boldsymbol{I}, \ \boldsymbol{Z} \rangle \leq \rho, \ \boldsymbol{Z} \in \mathbb{K}_1 \cap \mathbb{K}_2 \}$$

$\Updownarrow$ Strong duality

$$\max_{y_0, \mu} \{ y_0 + \rho\mu \mid \underbrace{\boldsymbol{F}_0 - y_0 \boldsymbol{H}_0}_{\boldsymbol{G}(y_0)} \mu \boldsymbol{I} \in \mathbb{K}_1^* + \mathbb{K}_2^*, \ \mu \geq 0 \}$$

$\Updownarrow$

$$\max_{y_0, \mu, \boldsymbol{Y}_2} \{ y_0 + \rho\mu \mid \boldsymbol{G}(y_0) - \boldsymbol{Y}_2 - \mu \boldsymbol{I} \in \mathbb{K}_1^* (= \mathbb{S}_+^n), \ \boldsymbol{Y}_2 \in \mathbb{K}_2^*, \ \mu \geq 0 \}$$

DNN Optimization : Summary

$$\min_{\boldsymbol{Z}} \{\langle \boldsymbol{F}_0,\ \boldsymbol{Z}\rangle \mid \langle \boldsymbol{H}_0,\ \boldsymbol{Z}\rangle = 1,\ (i = 1, 2, \ldots, m), \boldsymbol{Z} \in \mathbb{K}_1 \cap \mathbb{K}_2\}.$$

Dual of Lagrangian relaxation with parameter $\rho \geq 0$

$$\max_{y_0, \mu, \boldsymbol{Y}_2} \{y_0 + \rho\mu \mid \boldsymbol{G}(y_0) - \boldsymbol{Y}_2 - \mu \boldsymbol{I} \in \mathbb{K}_1^*,\ \boldsymbol{Y}_2 \in \mathbb{K}_2^*,\ \mu \geq 0\}$$

We searches

- y_0 by bisection method
- $\boldsymbol{Y}_1 \in \mathbb{K}_1^*$ and $\boldsymbol{Y}_2 \in \mathbb{K}_2^*$ by APG (to judge feasibility of y_0)
- μ : minimal eigenvalue of $\boldsymbol{G}(y_0) - \boldsymbol{Y}_2 \rightarrow$ always gives a valid LB

[Assumption 1] $\Pi_{\mathbb{K}_2}, \Pi_{\mathbb{K}_1^*}$ can be computed efficiently.

[Assumption 2] We have a tight $\rho \geq 0$.

BBCPOP : Matlab implementation [Naoki Ito, Kim, Kojima, Takeda and Toh, 2018]

2018/5/11

Improved QAPLIB lower bounds using BBCPOP

Hans D Mittelmann

The following lower bounds for problems from QAPLIB were computed using the code BBCPOP.
Full logfiles are available here.

problem	new bound	old bound	upper bound
Tai35b	2.695324e+08	242172800	283315445
Tai40b	6.088084e+08	564428353	637250948
Tai50b	4.310907e+08	395543467	458821517
Tai60a	6.325978e+06	5578356	7505962
Tai60b	5.923718e+08	542376603	608215054
Tai80a	1.165701e+07	10501941	13499184
Tai80b	7.862988e+08	717907288	818415043
Tai100a	1.785384e+07	15844731	21052466
Tai100b	1.151591e+09	1058131796	1185996137
Sko42	1.533264e+04	14934	15812
Sko49	2.265021e+04	22004	23386
Sko56	3.338503e+04	32610	34458
Sko64	4.701738e+04	45736	48498
Sko72	6.445510e+04	62691	66256
Sko81	8.835922e+04	86072	90998
Sko90	1.124237e+05	109030	115534

DNN Optimization : the case of Quadratic Assignment Problem

DNN formulation

$$\min\{\langle \boldsymbol{F}_0, \boldsymbol{Z}\rangle \mid \langle \boldsymbol{H}_0, \boldsymbol{Z}\rangle = 1, \boldsymbol{Z} \in \mathbb{K}_1 \cap \mathbb{K}_2\}. \tag{9}$$

$$\mathbb{K}_1 := \left\{\boldsymbol{Z} \in \mathbb{S}^{n+1}_{+}\right\} \tag{10}$$

$$\mathbb{K}_2 := \left\{ \boldsymbol{Z} \in \mathbb{S}^{n+1} \;\middle|\; \begin{array}{ll} \boldsymbol{Z}_{\alpha\beta} \geq 0 & \text{nonnegativity} \\ \boldsymbol{Z}_{0\alpha} = \boldsymbol{Z}_{\alpha 0} \geq \boldsymbol{Z}_{\alpha\alpha} & \\ \boldsymbol{Z}_{\alpha\beta} = 0 & \text{if } (\alpha, \beta) \in \Gamma \end{array} \right\} \tag{11}$$

DNN Optimization : the case of Quadratic Assignment Problem

[Assumption 1] $\Pi_{\mathbb{K}_2}, \Pi_{\mathbb{K}_1^*}$ can be computed efficiently.

- $\Pi_{\mathbb{K}_1^*}$ is the projection on to symmetric cones
- $\Pi_{\mathbb{K}_2}$ is defined as:

$$
\begin{array}{ll}
\Pi_{\mathbb{K}_2}(\boldsymbol{Z}_{\alpha\beta}) := max(0, \boldsymbol{Z}_{\alpha,\beta}) & \text{if } (\alpha, \beta) \in \gamma \\
\Pi_{\mathbb{K}_2}(\boldsymbol{Z}_{\alpha\alpha}) := avg(\boldsymbol{Z}_{\alpha\alpha}, \boldsymbol{Z}_{\alpha 0}, \boldsymbol{Z}_{0\alpha}) & \text{if } \boldsymbol{Z}_{\alpha 0} < \boldsymbol{Z}_{\alpha\alpha} \\
\Pi_{\mathbb{K}_2}(\boldsymbol{Z}_{\alpha\beta}) := \boldsymbol{Z}_{\alpha\beta} & \text{otherwise}
\end{array}
\tag{12}
$$

DNN Optimization : the case of Quadratic Assignment Problem

Example:

$$Z = \begin{pmatrix} 9.51 & 4.10 & 5.23 & 5.30 & 3.96 \\ 4.10 & 1.76 & 2.25 & 2.28 & 1.70 \\ 5.23 & 2.25 & 2.88 & 2.91 & 2.18 \\ 5.30 & 2.28 & 2.91 & 2.95 & 2.20 \\ 3.96 & 1.70 & 2.18 & 2.20 & 8.65 \end{pmatrix} \tag{13}$$

$$\Pi_{\mathbb{K}_2}(Z) = \begin{pmatrix} 9.51 & 4.10 & 5.23 & 5.30 & 5.52 \\ 4.10 & 1.76 & 0 & 0 & 1.70 \\ 5.23 & 0 & 2.88 & 2.91 & 0 \\ 5.30 & 0 & 2.91 & 2.95 & 0 \\ 5.52 & 1.70 & 0 & 0 & 5.52 \end{pmatrix} \tag{14}$$

DNN Optimization : the case of Quadratic Assignment Problem

[Assumption 2] We have a tight $\rho \geq 0$.

$$\min_{\boldsymbol{Z}} \{ \langle \boldsymbol{F}_0, \ \boldsymbol{Z} \rangle \mid \langle \boldsymbol{H}_0, \ \boldsymbol{Z} \rangle = 1, \ \langle \boldsymbol{I}, \ \boldsymbol{Z} \rangle \leq \rho, \ \boldsymbol{Z} \in \mathbb{K}_1 \cap \mathbb{K}_2 \}$$

In QAP case, we can set $\rho := n + 1$, where n is a dimension of $\boldsymbol{A}$, $\boldsymbol{B}$

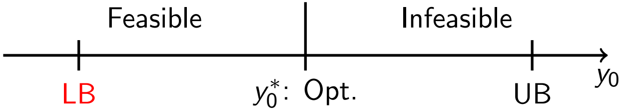
DNN-based Branch-and-Bound

DNN-based branch and bound

- Solve DNN relaxation at each node
- Prune the nodes if valid lower bound is greater than incumbent value.
- Implemented by C++ (c.f. BBCPOP is implemented by MATLAB)

The advantages of BP method / DNN relaxation:

- Warm start for BP-search
- Valid lower bound is useful to avoid numerical issues



DNN-based Branch-and-Bound

What is challenging?

- **No** primal solution of relaxation problem

Branch-and-bound without primal solution in relaxation problem

- How to branch? (no information of fractional solution)
    - Use ∇f to branch.
- How to get primal integer solution
    - Implement taboo search [Taillard, 1991]

DNN-based Branch-and-Bound: Parallelization

DNN-based branch-and-bound parallelized with UG

What is UG?

- General parallel branch-and-bound framework
- Many academic/commercial solvers are parallelized (ParaNUOPT, ParaSCIP, ParaXpress) and solve many open MIPLIB problems
- UG Best Practice : Use UG with Yuji! (implementation took three days $\approx$ 30 hours)

DNN-based Branch-and-Bound: Computational Results (SDP(ADMM)-based VS DNN-based)

problem	ADMM([1])		DNN			
	#node	time(s)	#node	1thread	3thread	6thread
nug12	23	43.74	35	31.39	26.21	24.77
nug14	14	49.56	28	73.64	63.34	62.74
nug15	15	147.85	72	138.59	91.71	75.51
nug16a	16	144.84	46	194.35	161.67	154.89
nug16b	31	419.06	197	384.59	186.83	139.53
nug17	188	1151.46	191	541.90	274.46	209.96
nug18	805	5071.32	355	1532.24	744.25	557.02

Table: Computational Results

[1] Liao, Z. (2016). Branch and bound via the alternating direction method of multipliers for the quadratic assignment problem.

DNN-based Branch-and-Bound: Computational Results

- nug24 is solved in 48652.65(s) with 3 threads, 30598.3369(s) with 6 threads, 7773 nodes

Table 4. Computational results for QAPLIB instances

Prob.	Time(sec)	# of subproblems	Improvements
nug21	13287	593656913	none
nug22	147378	6712276783	none
nug24	1269218	44317904109	none

Yuji Shinano, Tetsuya Fujie (1999). Parallel Branch-and-Bound Algorithms on a PC Cluster using PUBB.

Summary

- First implementation of DNN-based branch and bound
- Promising to solve difficult QAP
- Future work:
  - better way of branching?
  - Will dual solution help primal heuristics?
  - Explore symmetry in QAP
  - Parallelization on distributed memory environment (should be easy within UG-framework)