# Quantum Optimization with QAOA: Breaking Limitations but not Constraints

Brayden Goldstein-Gelb
BU Quantum

# Convergence guarantee for linearly-constrained combinatorial optimization with a quantum alternating operator ansatz

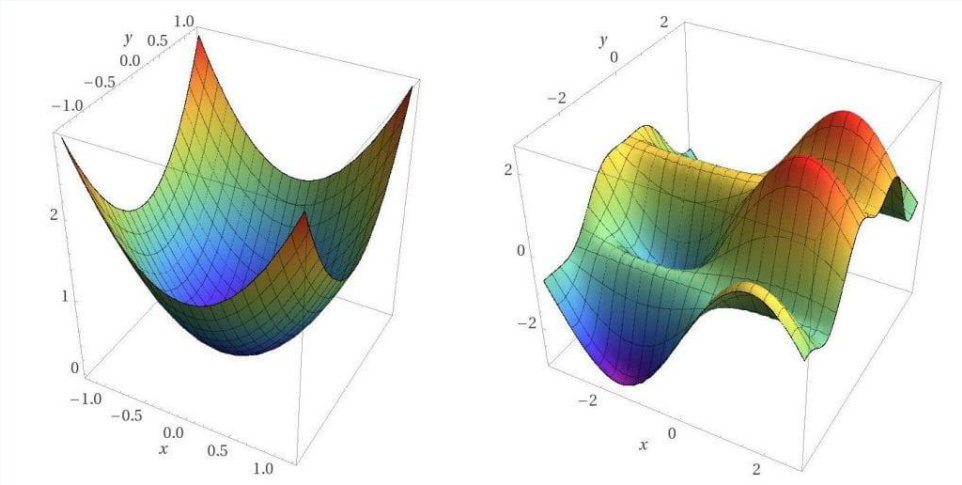
Brayden Goldstein-Gelb[1] and Phillip C. Lotshaw[2]

[1] *Brown University, Providence, RI 02912, USA*

[2] *Quantum Information Science Section, Oak Ridge National Laboratory, Oak Ridge, TN 37381, USA**

# Optimization Problems

- **Find an input that minimizes (or maximizes) some function**



# Quantum Representation

- **Cost function, $C(z)$, can be encoded as a hamiltonian, $C$**
- **Minimum of $C(z)$ $\leftrightarrow$ Ground state of $C$**

$$C\,|z\rangle = C(z)\,|z\rangle$$

# Adiabatic Evolution: high-level overview

- **Idea 1: start with some other Hamiltonian where we do know the ground state**

$$B = -\sum_i X_i \quad \longleftrightarrow \quad \frac{1}{\sqrt{2^n}} \sum_z |z\rangle$$

# Adiabatic Evolution: high-level overview

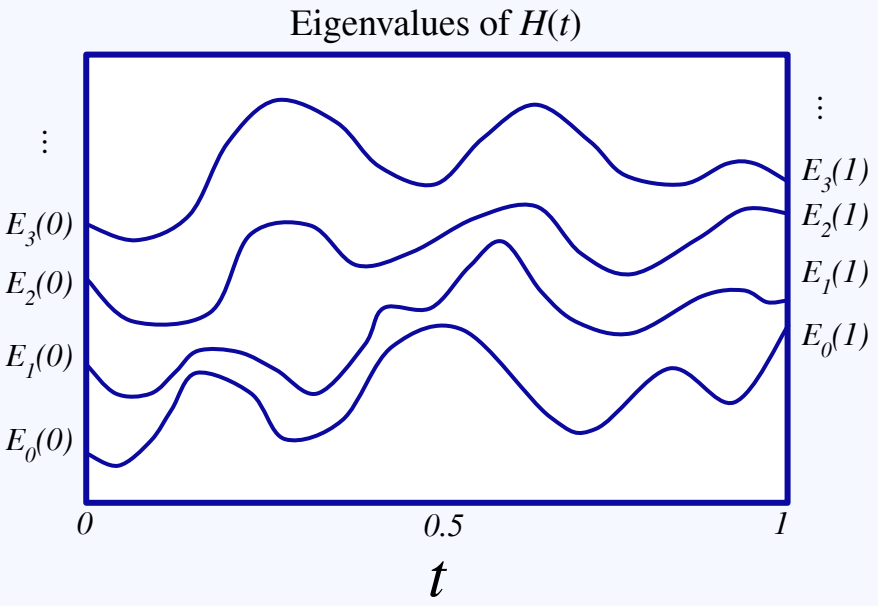
- **Idea 2: interpolate between the two**

$$H(t) = (1-t)B + tC$$

- **Trotter approximation allows us to implement the (approximate) time evolution operator**

$$U(T) \approx e^{-iH(t_{N-1})\Delta t} \cdot \ldots \cdot e^{-iH(t_1)\Delta t} \cdot e^{-iH(t_0)\Delta t}$$

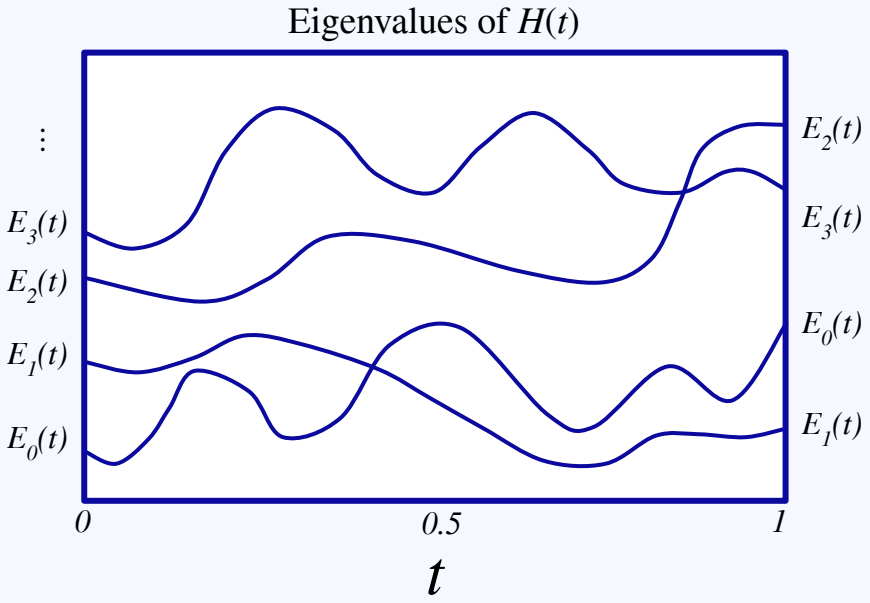
$$H(t) = (1-t)B + tC$$




$$\frac{1}{\sqrt{2^n}} \sum_z |z\rangle$$

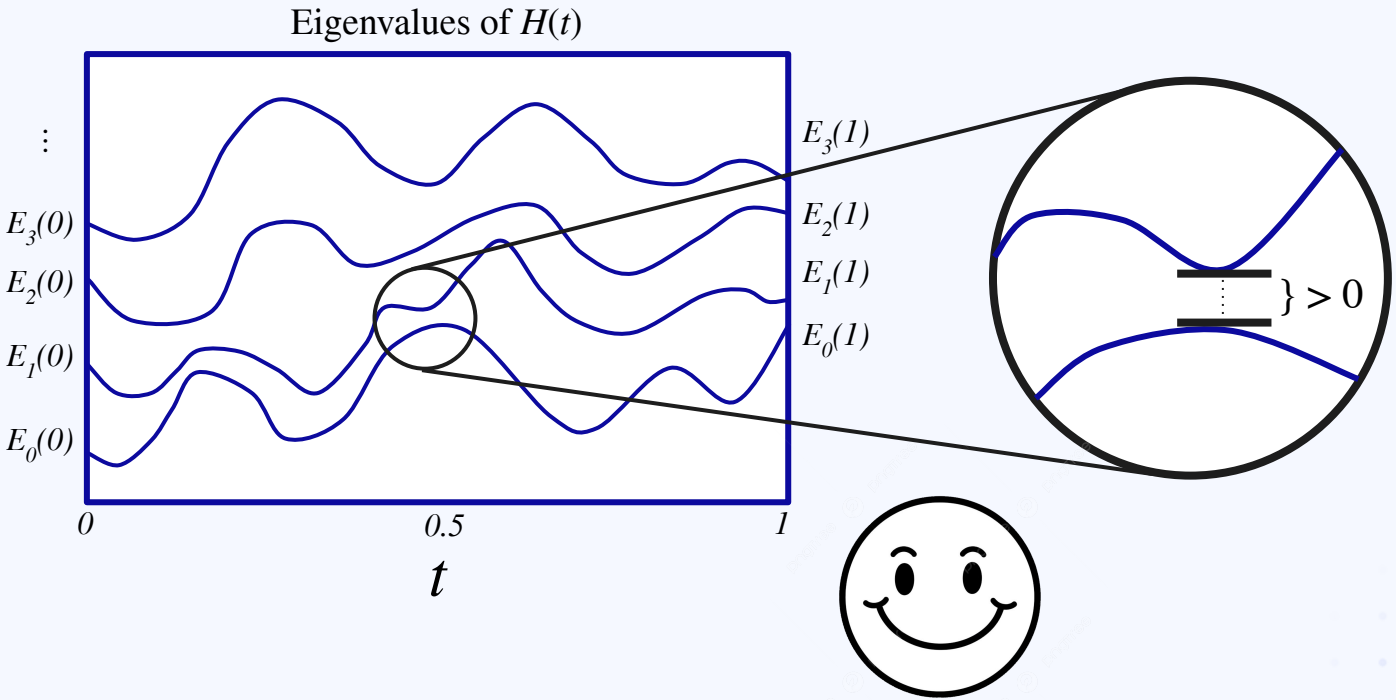
$$|\psi_{\mathrm{OPT}}\rangle$$

**Potential worry: what if there are level crossings?**






## Perron-Frobenius theorem says there are no level crossings for the ground state (under certain conditions)



# QAOA Takes it one step further

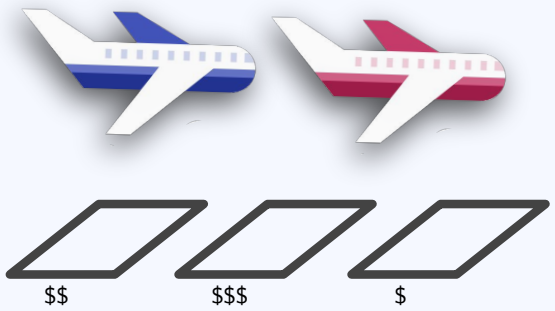
- **Execute the circuit many times and use classical optimization find the best values of β, ɣ**

$$U_B(\beta_l) = e^{-i\beta_l B}, \quad U_C(\gamma_l) = e^{-i\gamma_l C},$$

$$|\psi_p(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \prod_{l=1}^{p} U_B(\beta_l) U_C(\gamma_l) \, |+\rangle^{\otimes n}$$

# Our Focus: Binary Integer Linear Programming

- Cost function is linear in z with integer coefficients
- The $z_i$ take values 0 or 1
- Subject to a linear constraint

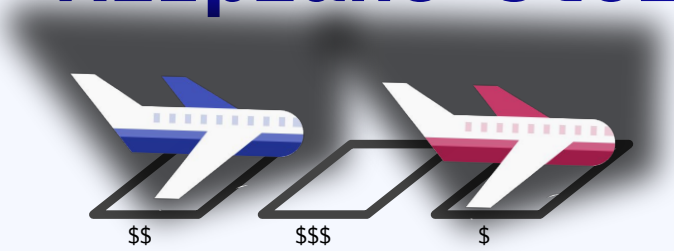



$$z_{\text{opt}} = \arg\min_{z} \sum_{i} c_i z_i$$

$$\text{s.t.} \sum_{i} s_i z_i = b$$

$$z_i \in \{0, 1\} \; \forall i$$

# Example: Airplane Storage




**How do we get QAOA to optimize with respect to a constraint?**

$$C(\boldsymbol{z}) = 2z_1 + 3z_2 + z_3$$

$$\text{s.t } z_1 + z_2 + z_3 = 2$$

# Common Approach: Lagrange Multipliers

**To minimize this**

$$z_{\text{opt}} = \arg\min_{z} \sum_i c_i z_i$$

$$\text{s.t.} \sum_i s_i z_i = b$$

$$z_i \in \{0, 1\} \; \forall i$$

**Instead minimize this**

$$\tilde{C}(z) = C(z) - \sum_\alpha \lambda_\alpha \left( \sum_i s_{\alpha,i} z_i - b_\alpha \right)^2$$

**where $\lambda_a$ are parameters to be optimized classically**

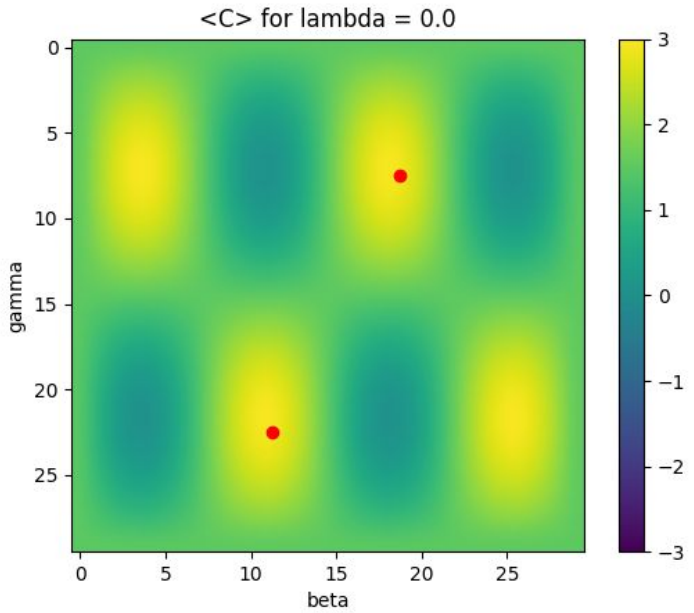
# Common Approach: Lagrange Multipliers

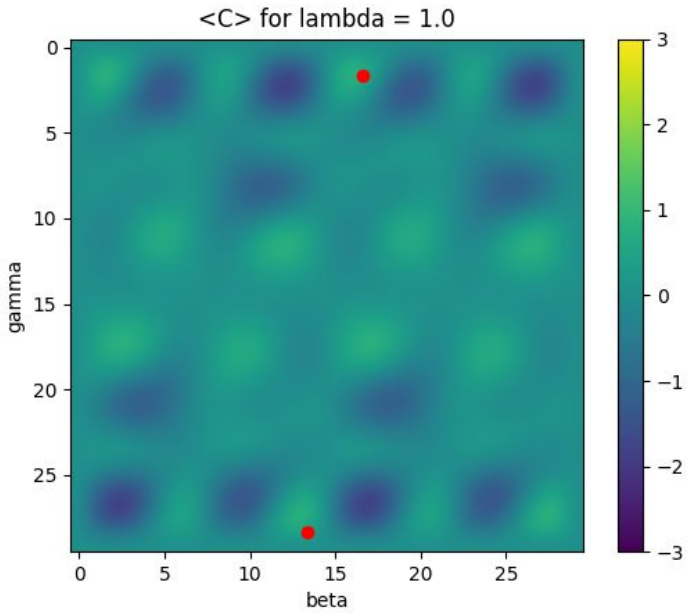
$$\tilde{C}(z) = C(z) - \sum_{\alpha} \lambda_{\alpha} \left( \sum_{i} s_{\alpha,i} z_i - b_{\alpha} \right)^2$$

**Issues:**

- **More complicated cost Hamiltonian**
- **Final outcome not guaranteed to satisfy the constraint**
- **More difficult to optimize β, ɣ**

# Example: 1-layer QAOA for Max-cut






# Alternate Approach: Restrict mixer to a subspace

- In default QAOA, B allows for transitions between all possible bitstrings

**Feasible subspace:**

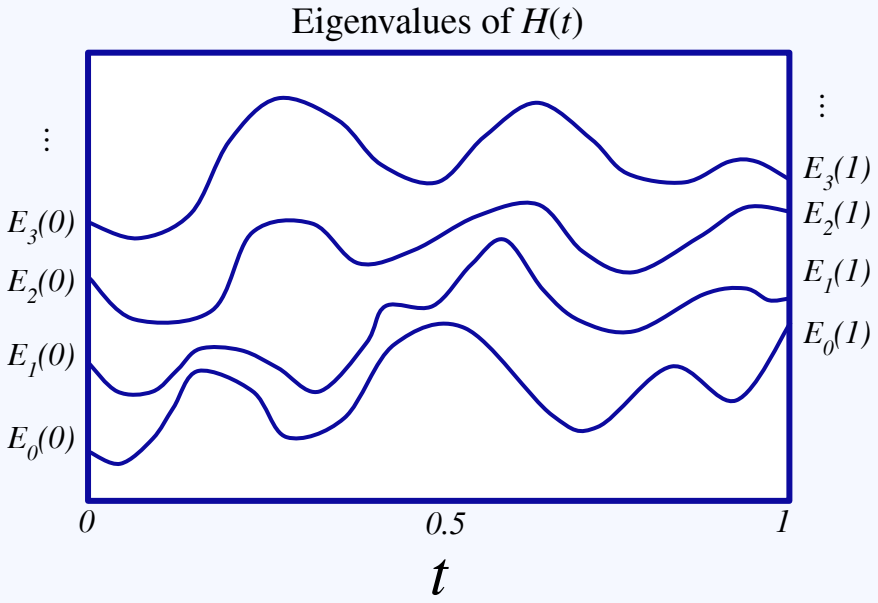
$$\mathcal{S}^{(b)} = \text{span}(\{|\boldsymbol{z}\rangle \mid \boldsymbol{s} \cdot \boldsymbol{z} = \boldsymbol{b}\})$$

**What if B preserved the subspace?**

$$B(\mathcal{S}^{(b)}) \subseteq \mathcal{S}^{(b)}$$

# Alternate Approach: Restrict mixer to a subspace

**Intuition: if you start in the feasible subspace, you end in the feasible subspace**




**Feasible**

$|\psi_{\mathrm{OPT}}\rangle$ **+ Feasible**

# How do we design a more general B?

- Instead, think of B as the sum of terms in a “Mixing Family”
  - If the terms in the mixing family to preserve the subspace, then the sum will as well
- Design goals:
  - Needs to depend only on the constraint
  - Should be easy to implement in terms of gates
  - Still needs to guarantee QAOA convergence

$$B^+ = \sum_{j \in J} B_j$$

# Mixing Family (QAOA → QAOA$^+$)

**Definition II.3.** *Mixing Family (adapted from Ref. [47])*
*Given a constrained optimization problem with a feasible solution space $\mathcal{S}$, a family of Hamiltonians $\mathcal{F} = \{B_j\}_{j \in J}$ is called a mixing family if it satisfies:*

**Three conditions (on next slide)**

**Then, a theorem from Binkowski et al. '23 says that starting in the ground state of $B^+$, running QAOA with $B^+$ as the mixer converges to the optimal solution in the $p \to \infty$ limit**

$$|\psi_p(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \prod_{l=1}^{p} U_{B^+}(\beta_l) U_C(\gamma_l) \, |+\rangle^{\otimes n}$$

$$B^+ = \sum_{j \in J} B_j$$

# So what are the conditions?

(a) *Feasibility preservation: If* $|z\rangle \in \mathcal{S}$ *and* $|z'\rangle \notin \mathcal{S}$, *then* $\langle z| B_j |z'\rangle = 0$ *for each* $B_j$.

**Each $B_j$ maps feasible states to feasible states**

(b) *Nonpositivity: The restriction of each* $B_j$ *to the solution subspace* $\mathcal{S}$, *denoted* $B_j|_{\mathcal{S}}$, *is component-wise nonpositive in the computational basis.*

**All entries in the matrix of $B_j$ must be nonpositive**

(c) *Full mixing of solutions: The basis-state-transition graph with adjacency matrix* $\sum_j B_j|_{\mathcal{S}}$ *is connected.*

**Any two feasible states are connected by a sequence of $B_j$ transitions**

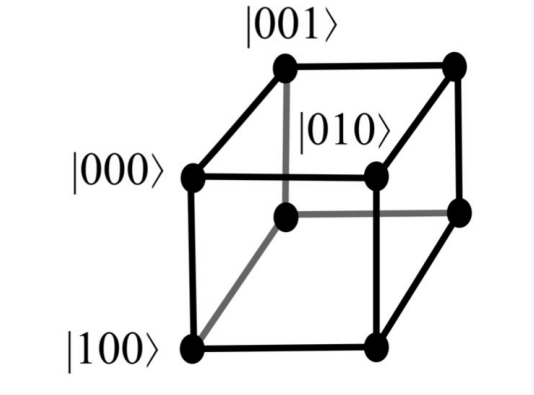
**Binkowski et al. show that this is enough for Perron-Frobenius to kick in!**

# Sanity check for standard mixer:

$$B = -\sum_i X_i$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Basis state interaction graph:




(a) *Feasibility preservation: If* $|z\rangle \in \mathcal{S}$ *and* $|z'\rangle \notin \mathcal{S}$, *then* $\langle z| B_j |z'\rangle = 0$ *for each* $B_j$.

✅

(b) *Nonpositivity: The restriction of each* $B_j$ *to the solution subspace* $\mathcal{S}$, *denoted* $B_j|_{\mathcal{S}}$, *is component-wise nonpositiv in the computational basis.*

✅

(c) *Full mixing of solutions: The basis-state-transition graph with adjacency matrix* $\sum_j B_j|_{\mathcal{S}}$ *is*

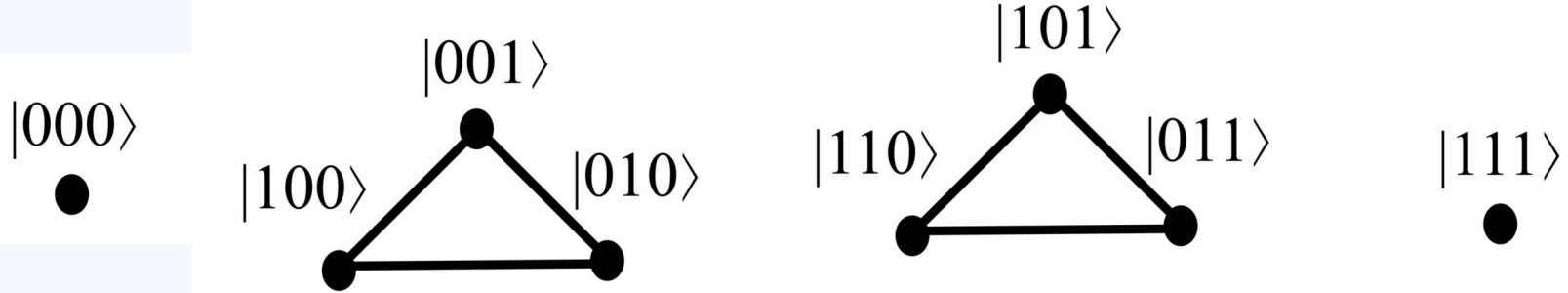
✅

# Previous work: XY-Mixer

**Wang et al. '20 designed the following mixer for QAOA+, which preserves Hamming weight**

$$B^{XY} = -\frac{1}{2}\sum_{i<j} X_i X_j + Y_i Y_j = -\sum_{i<j} |0_i, 1_j\rangle \langle 1_i, 0_j| + |1_i, 0_j\rangle \langle 0_i, 1_j|$$

Basis state interaction graph:




**Can we generalize this?**

# Table Notation

- Can use a table to represent a constraint
- Rows are coefficients, columns are number of occurrences of a variable
- Note: each row does not need to have the same number of columns
- Index coefficient by the row/column where they appear in the table

$$\sum_{\kappa=1}^{k} \kappa \sum_{l=1}^{l_\kappa} z_{[\kappa,l]} \quad \leftrightarrow \quad \begin{array}{c} 1 \\ \cdots \\ \kappa \leq k \\ \cdots \\ k \end{array} \quad \begin{array}{|c|c|c|} \hline z_{[1,1]} & \cdots & z_{[1,l_1]} \\ \hline & & \\ \hline z_{[\kappa,1]} & \cdots & z_{[\kappa,l_\kappa]} \\ \hline & & \\ \hline z_{[k,1]} & \cdots & z_{[k,l_k]} \\ \hline \end{array} .$$

# Table Notation Examples

**“Base Constraint” - all constraints we consider contain it**

$$z_{[1,1]} + 2z_{[2,1]} + 3z_{[3,1]} + 4z_{[4,1]} + 5z_{[5,1]} + z_{[1,2]} \leftrightarrow$$

| | | |
|---|---|---|
| 1 | $z_{[1,1]}$ | $z_{[1,2]}$ |
| 2 | $z_{[2,1]}$ | |
| 3 | $z_{[3,1]}$ | |
| 4 | $z_{[4,1]}$ | |
| 5 | $z_{[5,1]}$ | |

.

$$\left|1_{[1,1]}, 1_{[2,1]}, 0_{[3,1]}, 1_{[4,1]}, 0_{[5,1]}, 0_{[1,2]}\right\rangle \leftrightarrow$$

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 2 | 1 | |
| 3 | 0 | |
| 4 | 1 | |
| 5 | 0 | |

.

# Table Notation Examples

**“Repeated Sequential Constraint”**

$$\sum_{\kappa=1}^{3} \sum_{l=1}^{3} \kappa z_{[\kappa,l]} \quad \leftrightarrow$$

| | | | |
|---|---|---|---|
| 1 | $z_{[1,1]}$ | $z_{[1,2]}$ | $z_{[1,3]}$ |
| 2 | $z_{[2,1]}$ | $z_{[2,2]}$ | $z_{[2,3]}$ |
| 3 | $z_{[3,1]}$ | $z_{[3,2]}$ | $z_{[3,3]}$ |

# Table Notation Examples

**“Unstructured Constraint” - constraints can have many shapes and sizes**

$$z_{[1,1]} + z_{[1,2]} + 2z_{[2,1]} + 3z_{[3,1]} + 3z_{[3,2]} + 3z_{[3,3]} + 4z_{[4,1]} + 5z_{[5,1]} + 6z_{[6,1]} + 6z_{[6,2]} \leftrightarrow$$

| | | | |
|---|---|---|---|
| 1 | $z_{[1,1]}$ | $z_{[1,2]}$ | |
| 2 | $z_{[2,1]}$ | | |
| 3 | $z_{[3,1]}$ | $z_{[3,2]}$ | $z_{[3,3]}$ |
| 4 | $z_{[4,1]}$ | | |
| 5 | $z_{[5,1]}$ | | |
| 6 | $z_{[6,1]}$ | $z_{[6,2]}$ | |

# Building a Mixing Family

**We will build the mixing family out of the following "Merge operators":**

$$M^{(m)}_{I,i^*} = |0_{i_1}, 0_{i_2}, \ldots 0_{i_m}, 1_{i^*}\rangle\langle 1_{i_1}, 1_{i_2}, \ldots 1_{i_m}, 0_{i^*}| + |1_{i_1}, 1_{i_2}, \ldots 1_{i_m}, 0_{i^*}\rangle\langle 0_{i_1}, 0_{i_2}, \ldots 0_{i_m}, 1_{i^*}|$$

where $I = \{i_1, i_2, \ldots i_m\}$ is a set of qubits

**Note: the $|I| = 1$, case is the same as the XY-Mixer**

$$\frac{1}{2}(X_i X_j + Y_i Y_j) = |0_i, 1_j\rangle \langle 1_i, 0_j|$$

| operator $M_{I,i^*}$ | Pauli representation |
|---|---|
| $M_{a,b}$ | $2^{-1}(X_aX_b + Y_aY_b)$ |
| $M_{\{a,b\},c}$ | $2^{-2}(X_aX_bX_c - Y_aY_bX_c + Y_aX_bY_c + X_aY_bY_c)$ |
| $M_{\{a,b,c\},d}$ | $2^{-3}(X_aX_bX_cX_d - Y_aY_bX_cX_d - Y_aX_bY_cX_d - X_aY_bY_cX_d + Y_aX_bX_cY_d + X_aY_bX_cY_d + X_aX_bY_cY_d - Y_aY_bY_cY_d)$ |

# Problem Restriction

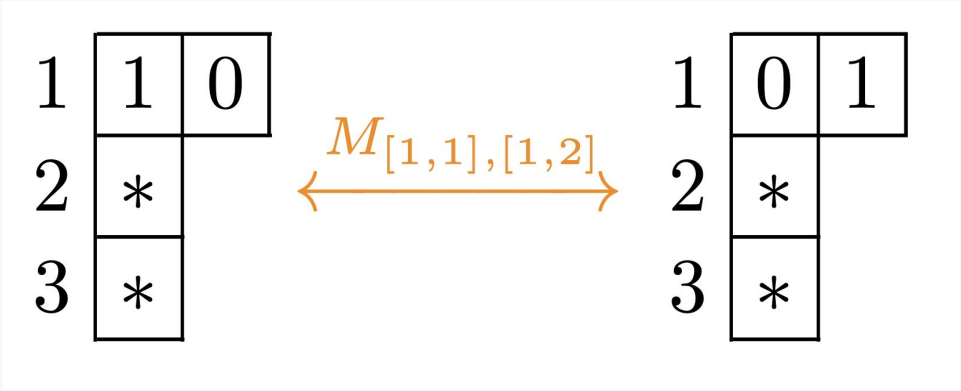
- In this work, we only consider constraints where
  - Each coefficient $1, \ldots, k$ appears at least once
  - Coefficient $1$ appears at least twice

Will see later that this is necessary for our design...

# Constraint-Preserving Operations (i)

$$z_{[1,1]} + z_{[1,2]} + 2z_{[2,1]} + 3z_{[3,1]}$$

- **Swapping two cells in the same row (think 1 = 1)**

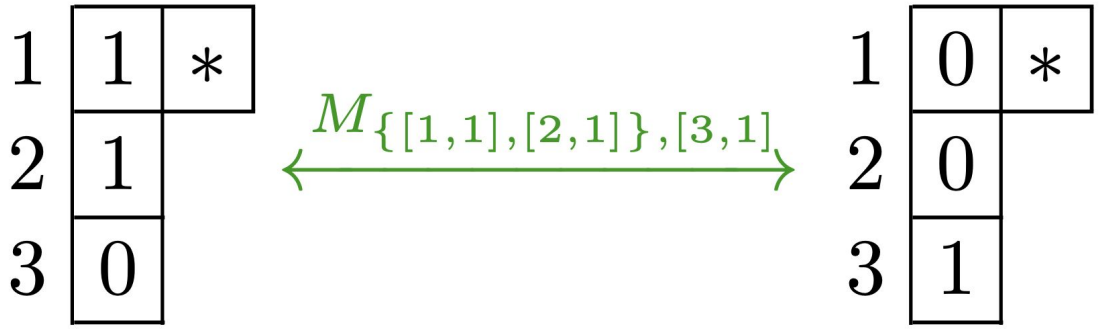



$$1(1) + 1(0) + 2(*) + 3(*) = 1(0) + 1(1) + 2(*) + 3(*)$$

# Constraint-Preserving Operations (ii)

$$z_{[1,1]} + z_{[1,2]} + 2z_{[2,1]} + 3z_{[3,1]}$$

- **Shifting an element down one row at the cost of setting the [1,1] cell to zero (think 1 + 2 = 3)**

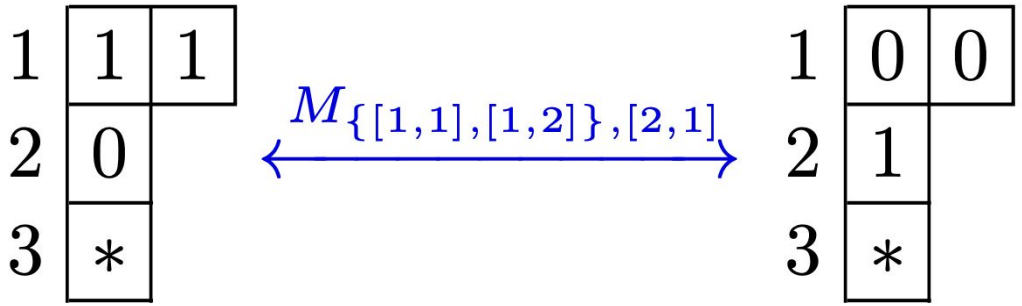



$$1(1) + 1(*) + 2(1) + 3(0) = 1(0) + 1(*) + 2(0) + 3(1)$$

# Constraint-Preserving Operations (iii)

$$z_{[1,1]} + z_{[1,2]} + 2z_{[2,1]} + 3z_{[3,1]}$$

- **Set two ones to zero in exchange for a two (think 1 + 1 = 2)**




$$1(1) + 1(1) + 2(0) + 3(*) = 1(0) + 1(0) + 2(1) + 3(*)$$

# Our mixing families

$$\mathcal{F}_{\min} = \{-M_{[\kappa,l],[\kappa,l+1]} : \kappa \in \{1,\dots,k\}, l \in \{1,\dots l_\kappa - 1\}\} \quad \textbf{(i)}$$

$$\cup \{-M_{\{[1,1],[\kappa,1]\},[\kappa+1,1]} : \kappa \in \{2,\dots,k-1\}\} \quad \textbf{(ii)}$$

$$\cup \{-M_{\{[1,1],[1,2]\},[2,1]}\} \quad \textbf{(iii)}$$

**We call this the "minimal mixing family" because this is (we will show) enough**

**Can also make a "maximal mixing family" by taking every possible term:**

$$\mathcal{F}_{\max} = \{-M^{(m)}_{I,i^*} : i^* \in \{1,\dots,N\}\ \forall m, I = \{i_1,\dots i_m\},\ \text{s.t. } i_j \neq i^*,\ \sum_{i\in I} s_i = s_{i^*}\}$$

# Theorem (main result)

**Theorem III.1.**

Let $\mathcal{F}$ be a set of Hamiltonians such that $\mathcal{F}_{min} \subseteq \mathcal{F} \subseteq \mathcal{F}_{max}$ .
Then, $\mathcal{F}$ is a mixing family.

**Idea is that a user can choose how big to make their mixing family based on use case**

# Things to prove:

(a) *Feasibility preservation: If* $|z\rangle \in \mathcal{S}$ *and* $|z'\rangle \notin \mathcal{S}$, *then* $\langle z| B_j |z'\rangle = 0$ *for each* $B_j$.

✅

(b) *Nonpositivity: The restriction of each* $B_j$ *to the solution subspace* $\mathcal{S}$, *denoted* $B_j|_{\mathcal{S}}$, *is component-wise nonpositive in the computational basis.*

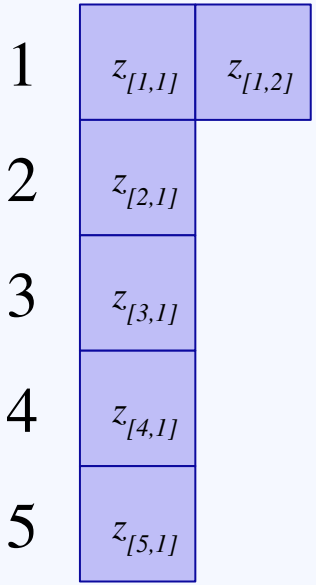
✅

(c) *Full mixing of solutions: The basis-state-transition graph with adjacency matrix* $\sum_j B_j|_{\mathcal{S}}$ *is connected.*

We only need to show this for minimal mixing family:
adding more mixers adds more edges!

# Warmup

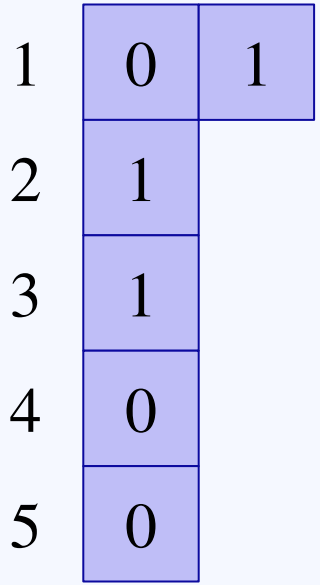
$$z_{[1,1]} + z_{[1,2]} + 2z_{[2,1]} + 3z_{[3,1]} + 4z_{[4,1]} + 5z_{[5,1]} = 6$$




## 6 Feasible Solutions:

1. $0 + 1 + 2 + 3 + 0 + 0 = 6$
2. $1 + 0 + 2 + 3 + 0 + 0 = 6$
3. $0 + 0 + 2 + 0 + 4 + 0 = 6$
4. $1 + 1 + 0 + 0 + 4 + 0 = 6$
5. $0 + 1 + 0 + 0 + 0 + 5 = 6$
6. $1 + 0 + 0 + 0 + 0 + 5 = 6$

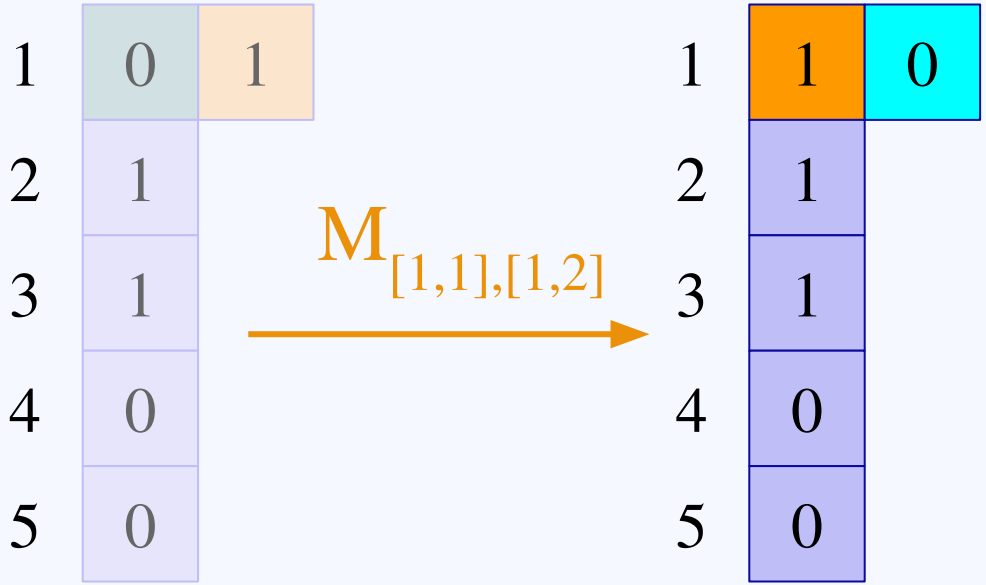
# Warmup




**1.**

$$0 + 1 + 2 + 3 + 0 + 0 = 6$$

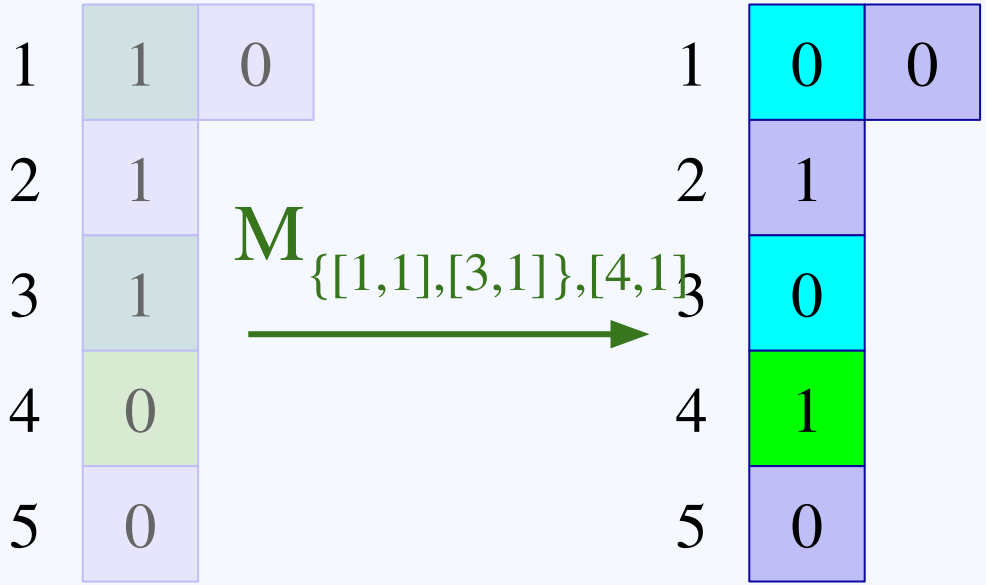
# Warmup




**2.**

$1 + 0 + 2 + 3 + 0 + 0 = 6$

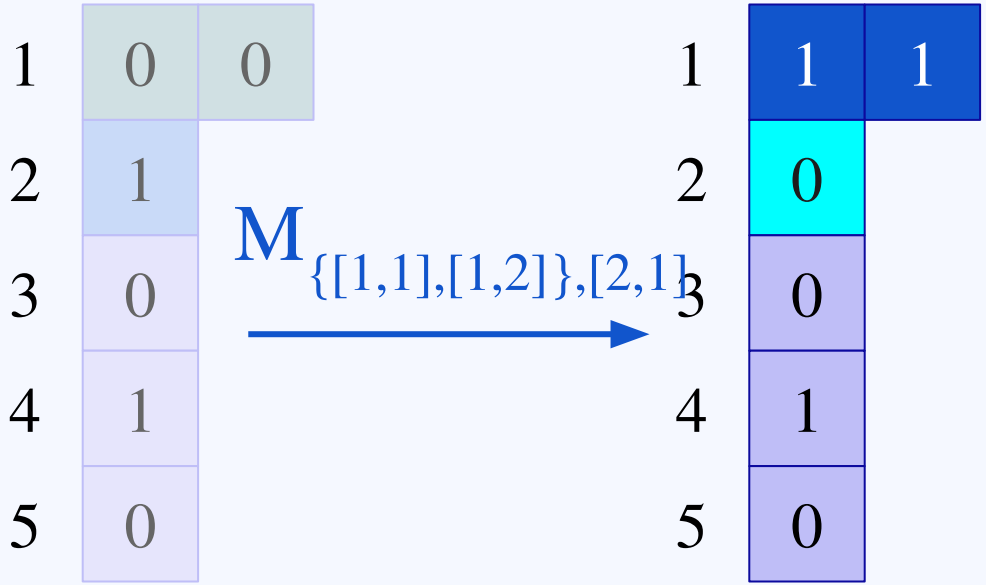
# Warmup




**3.**

$0 + 0 + 2 + 0 + 4 + 0 = 6$

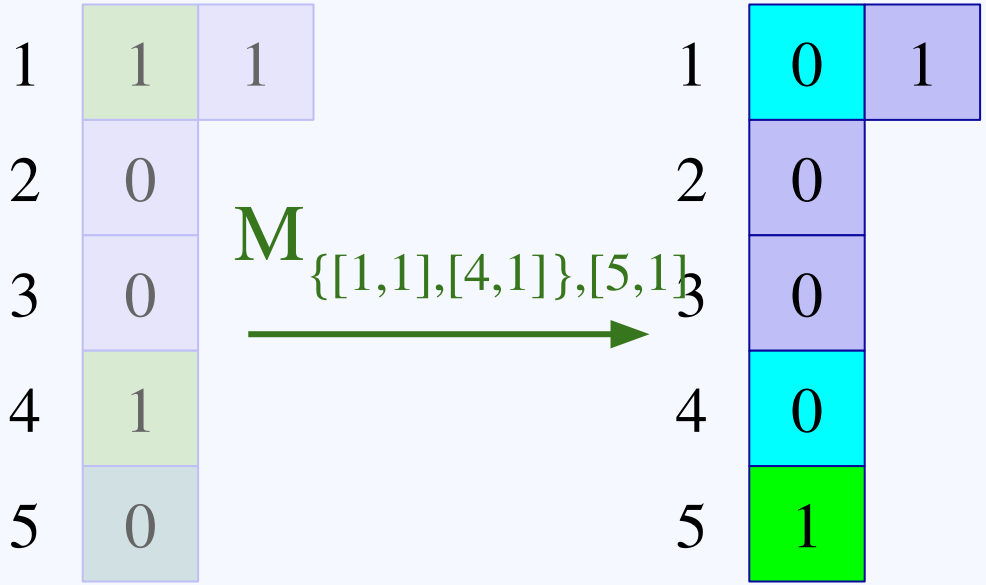
# Warmup




**4.**

$1 + 1 + 0 + 0 + 4 + 0 = 6$

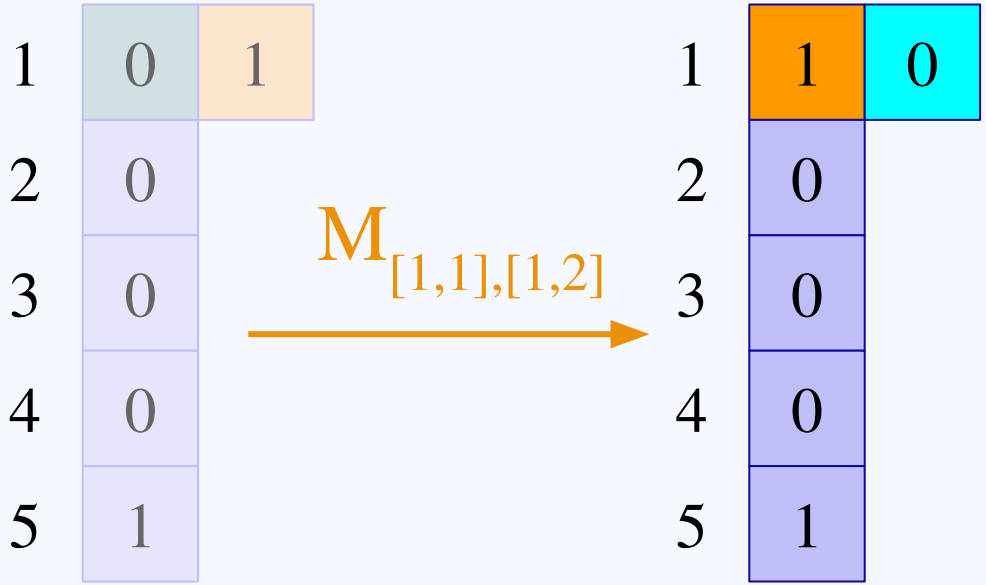
# Warmup




**5.**

$0 + 1 + 0 + 0 + 0 + 5 = 6$

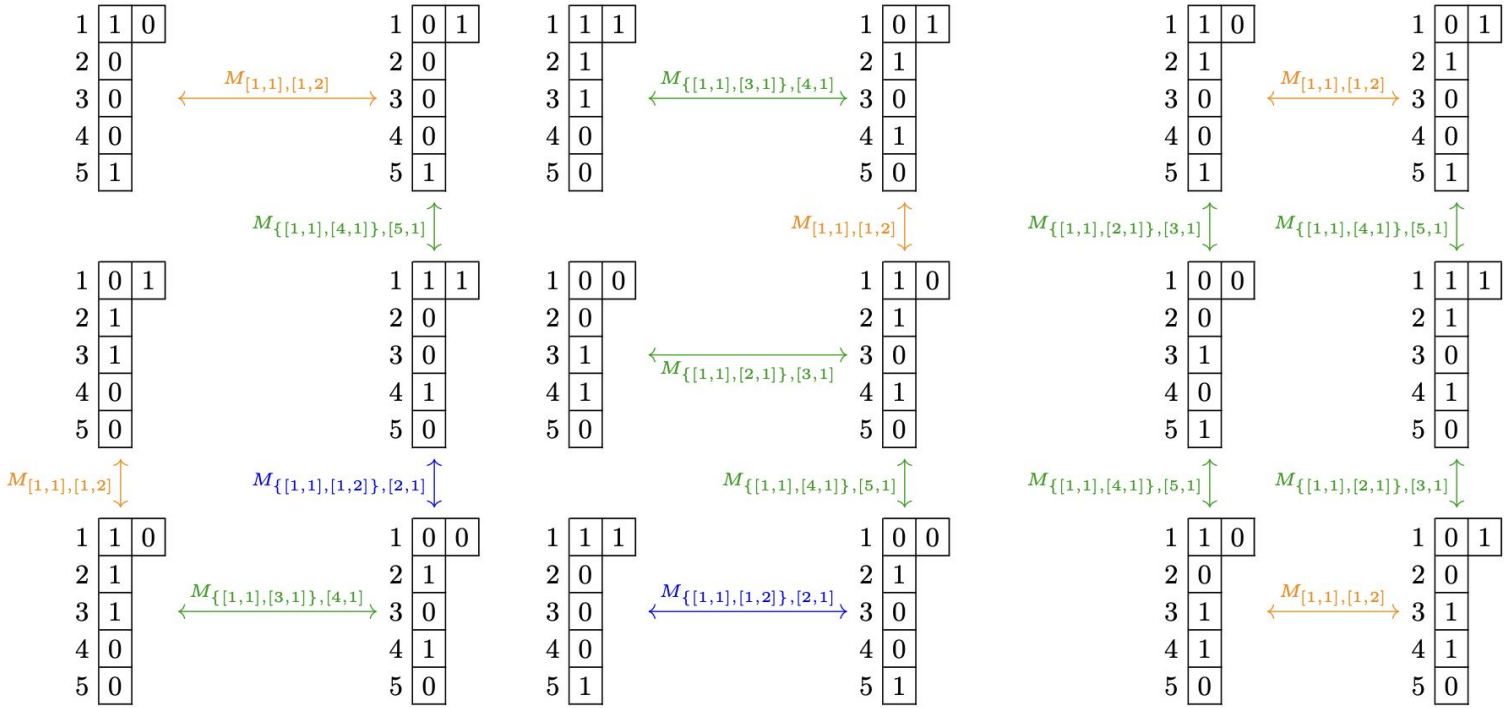
# Warmup



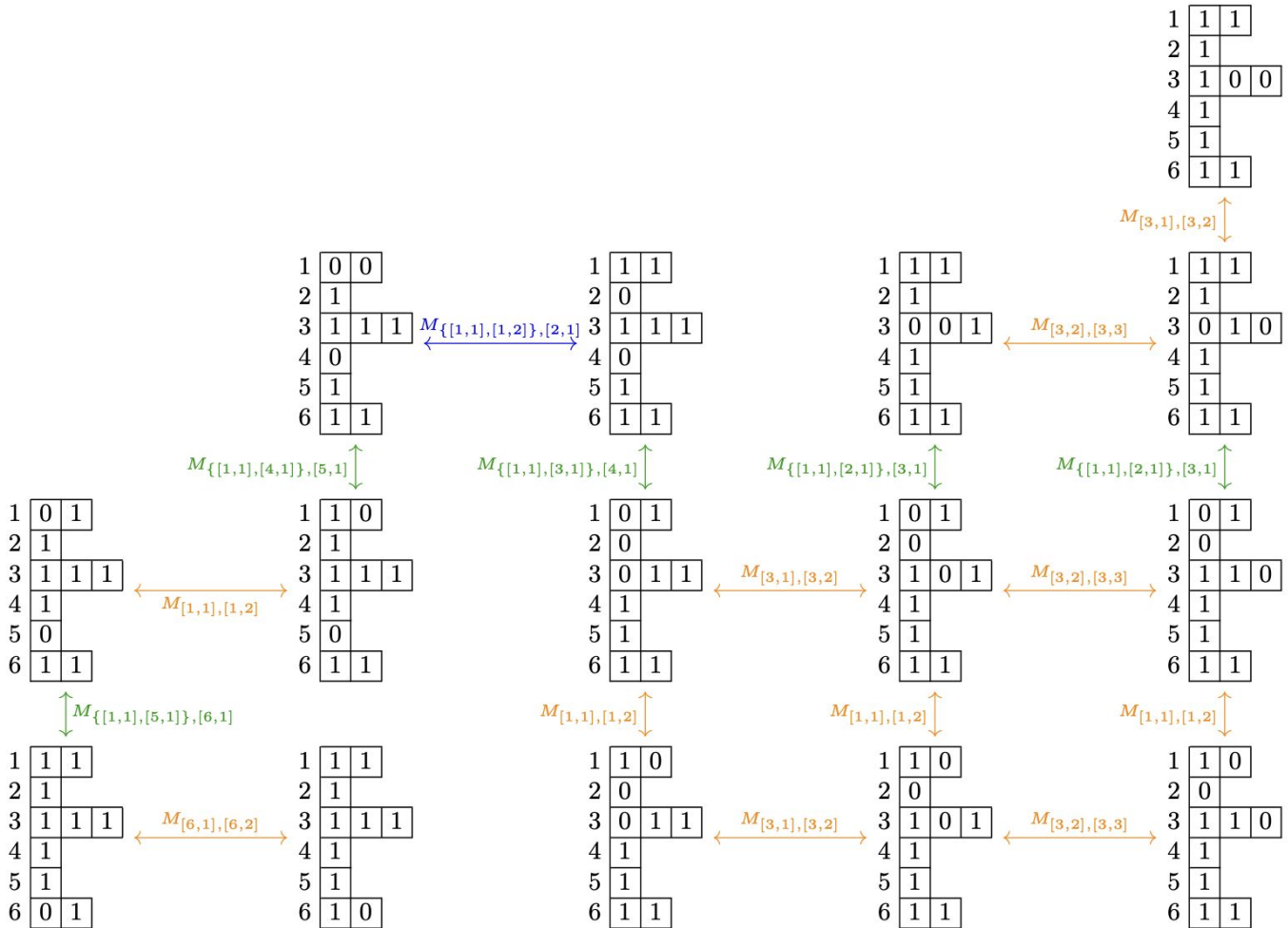

**6.**

$1 + 0 + 0 + 0 + 0 + 5 = 6$

# Lots of graphs!




(f) $b = 6$ (g) $b = 7$ (h) $b = 8$

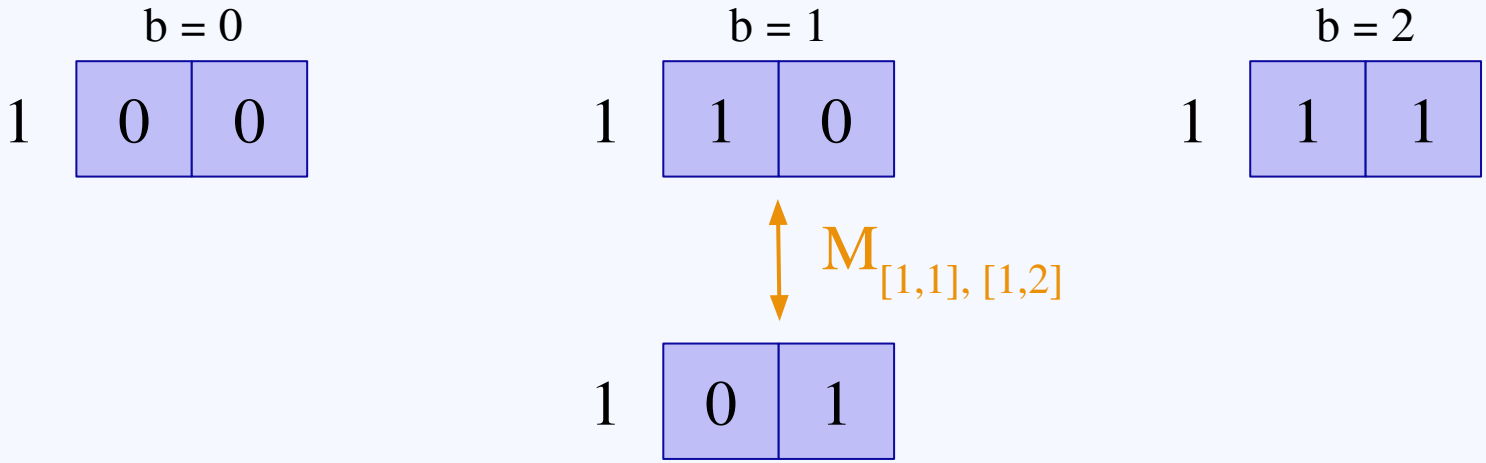

$M_{\{[1,1],[1,2]\},[2,1]}$
$M_{\{[1,1],[4,1]\},[5,1]}$
$M_{\{[1,1],[3,1]\},[4,1]}$
$M_{\{[1,1],[2,1]\},[3,1]}$
$M_{[3,1],[3,2]}$
$M_{[3,2],[3,3]}$
$M_{[1,1],[1,2]}$
$M_{\{[1,1],[5,1]\},[6,1]}$
$M_{[6,1],[6,2]}$

Let’s prove it’s always connected!

Proof: by induction on *N*, number of variables

**Base Case:** *N = 2* (coefficient 1 must appear twice)

Possible (non-empty) graphs, $G^{(2,\,b)}$ :



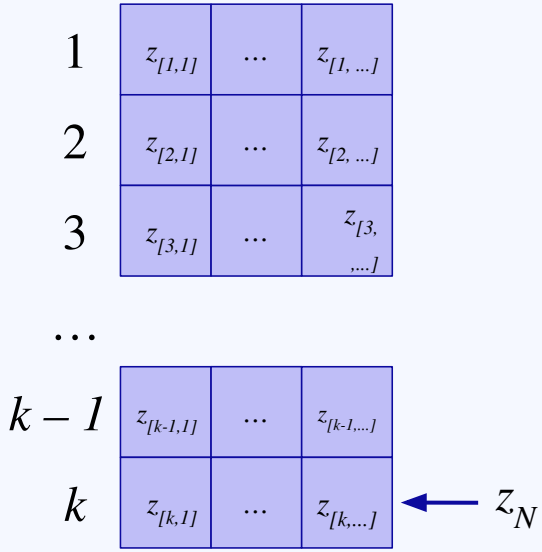

$$z_{[1,1]} + z_{[1,1]} = b$$

✅

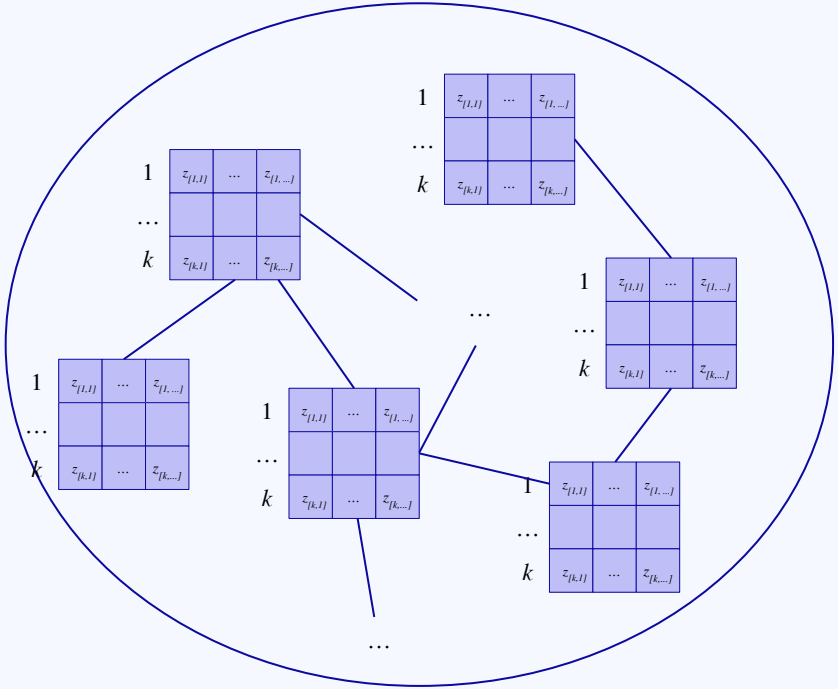
Proof: by induction on *N*, number of variables

**Inductive Case:** Suppose holds for *N* - *1* variable problems

Now consider some constraint with *N* variables:

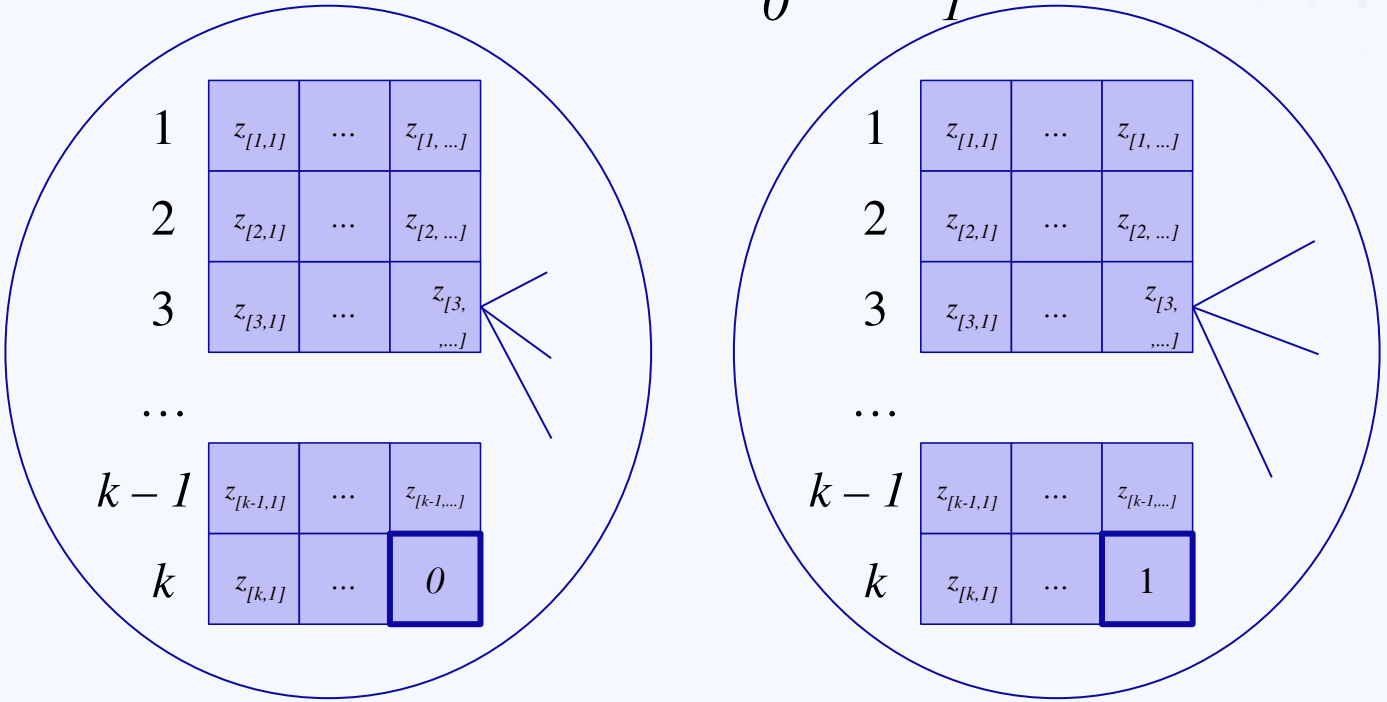


$$G^{(N,\,b)} = (V^{(N,\,b)},\, E^{(N,\,b)})$$



Partition vertices based on the value of $z_N$

$$V^{(N,\,b)} = V_0 \square V_1$$



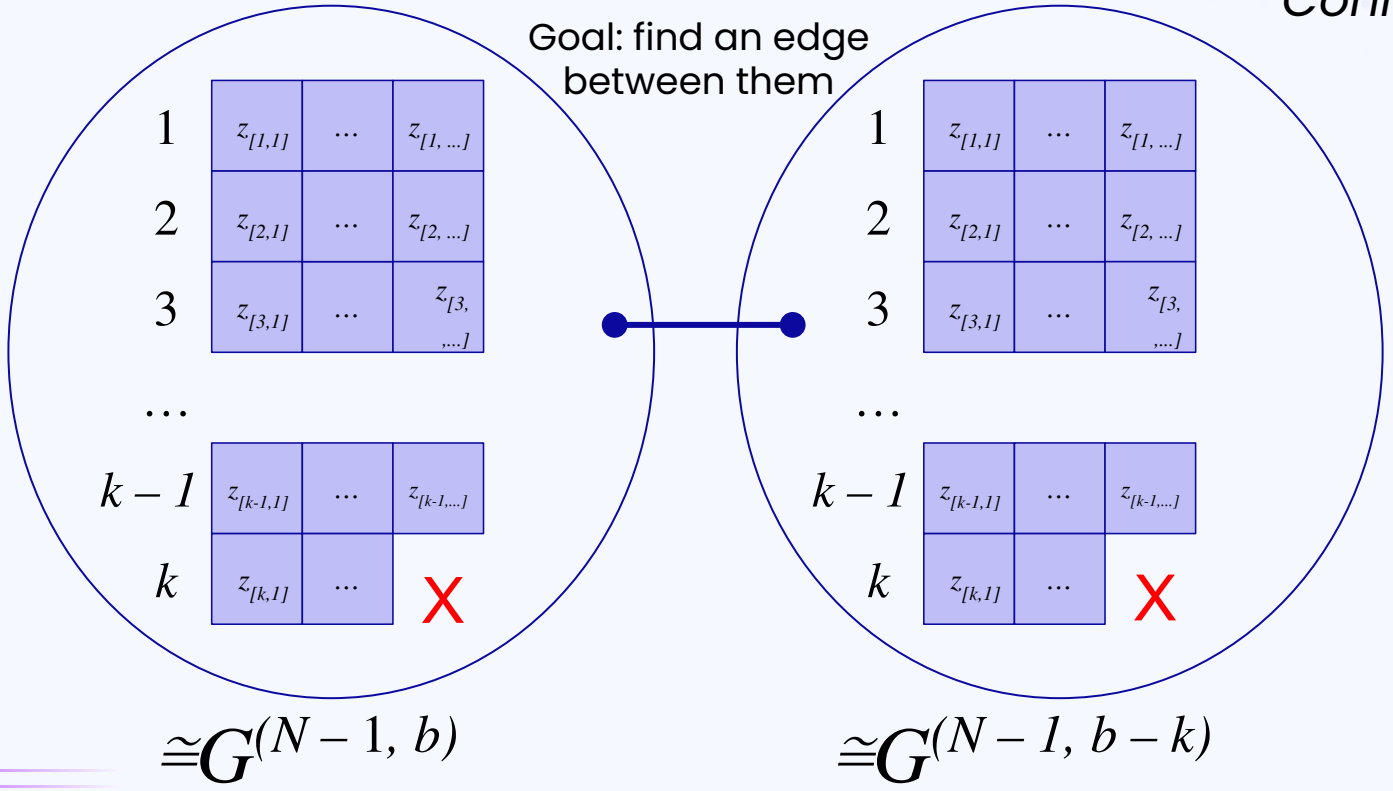
$$\sum_{i=1}^{N-1} s_i z_i = b$$

$$\sum_{i=1}^{N-1} s_i z_i + k = b$$

*Connected!*

*Connected!*

Goal: find an edge between them

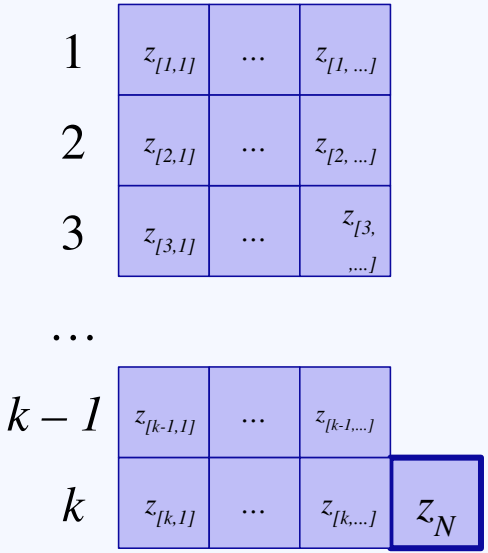



$\cong G(N-1,\ b)$

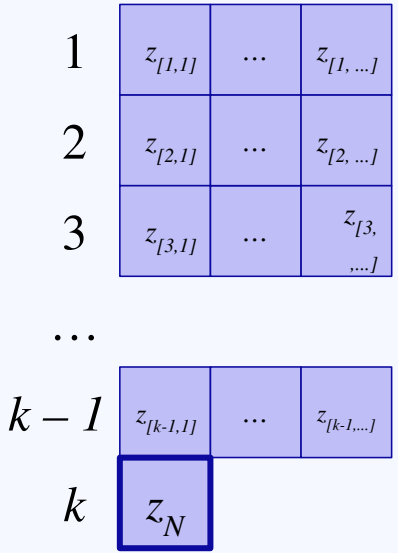
$\cong G(N-1,\ b-k)$

# Two cases:

(1) Added to existing row:




(2) Added to new row:



Case (1)

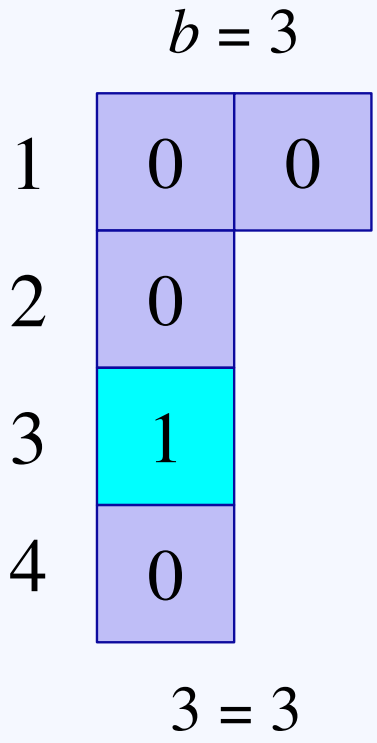
**Lemma** *(informal)*: For any reasonable value of $b$, I can choose any cell in the table and guarantee that there is a feasible state where that cell is equal to one

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 2 | 0 | 1 |
| 3 | 1 | 0 |

**Lemma C.4.** *Suppose we have a linearly constrained problem specified by (17). Suppose there are $l_\kappa$ variables with coefficient $\kappa$, and denote the $l$th one of these variables as $z_{[\kappa,l]}$. Consider a specific variable $z_{[\kappa*,l*]}$. Then, $\forall b$ with $\kappa* \leq b \leq \sum_{\kappa=1}^{k} \kappa l_\kappa$, $\exists |\boldsymbol{z}\rangle \in V^{(b)}$ such that $z_{[\kappa*,l*]} = 1$.*

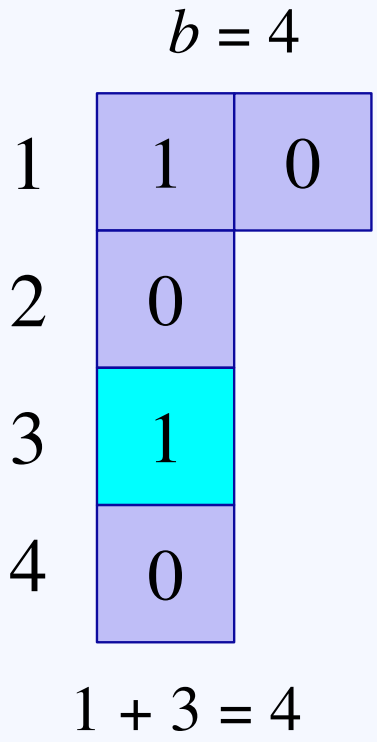
Case (1)

**Proof:** by animation



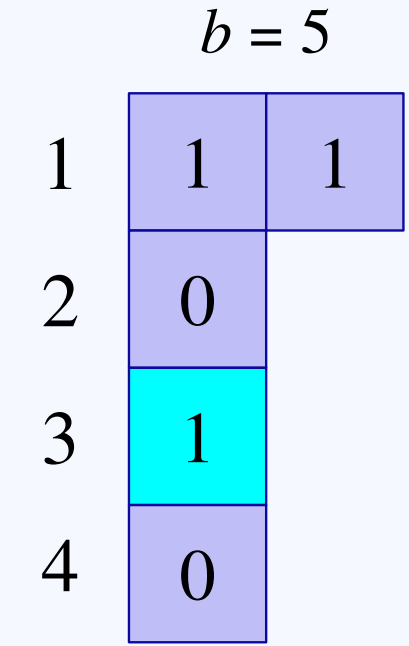
Case (1)

**Proof:** by animation



Case (1)

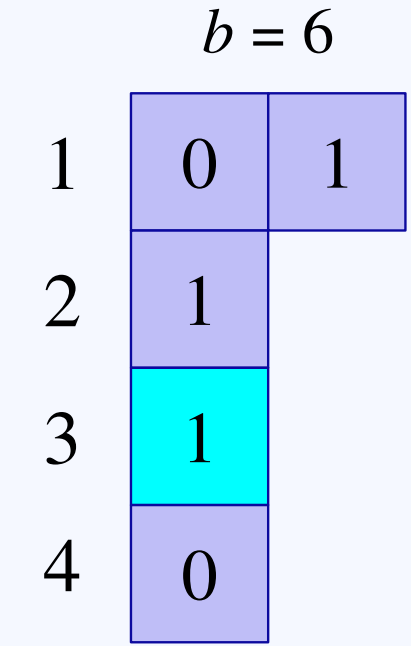
**Proof:** by animation




$1 + 1 + 3 = 5$

Case (1)

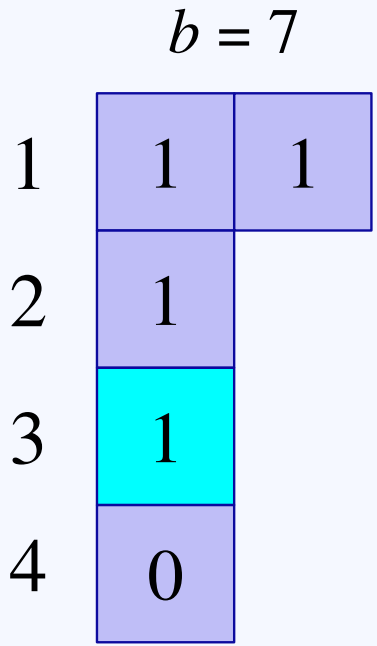
**Proof:** by animation




$1 + 2 + 3 = 6$

Case (1)

**Proof:** by animation

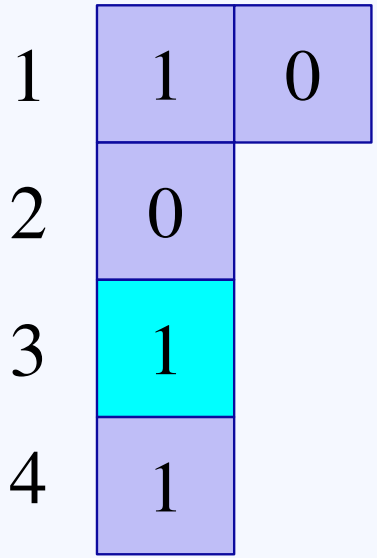



$$1 + 1 + 2 + 3 = 7$$

Case (1)

**Proof:** by animation

$b = 8$

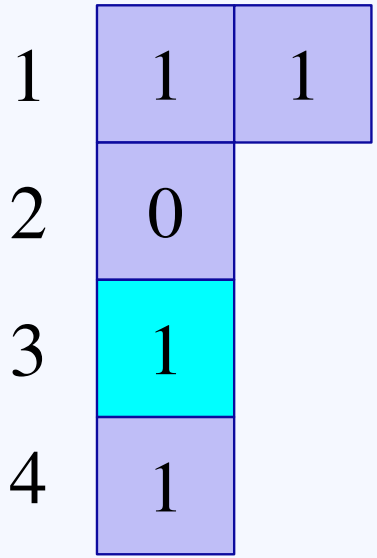



$1 + 3 + 4 = 8$

Case (1)

**Proof:** by animation

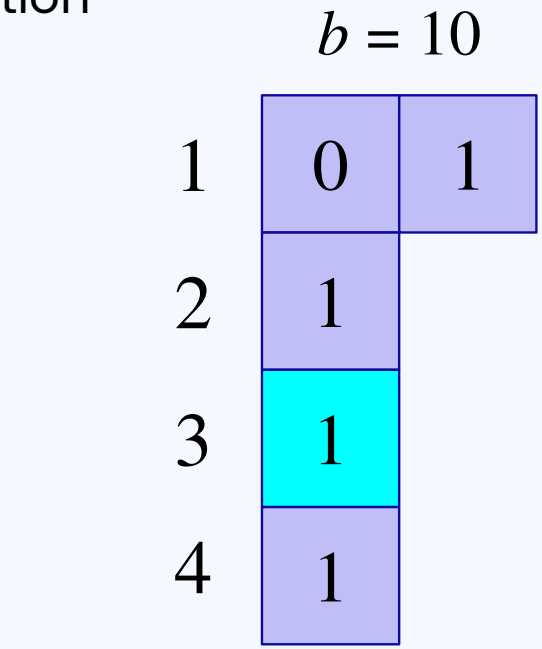
$b = 9$




$1 + 1 + 3 + 4 = 9$

Case (1)

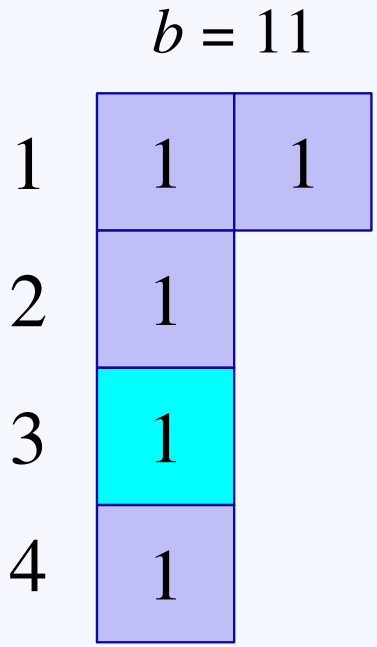
**Proof:** by animation




$$1 + 2 + 3 + 4 = 10$$

Case (1)

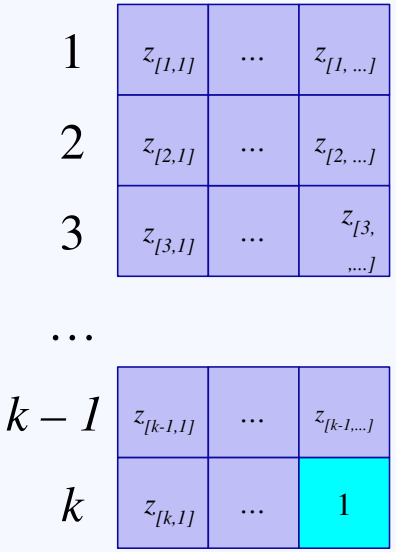
**Proof:** by animation




$$1 + 1 + 2 + 3 + 4 = 11$$

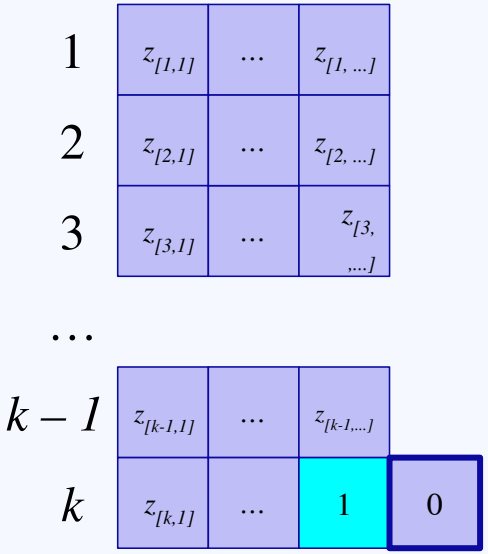
Case (1)

By the lemma, there is a vertex in $G^{(N-1,\ b)}$ where the blue cell = 1

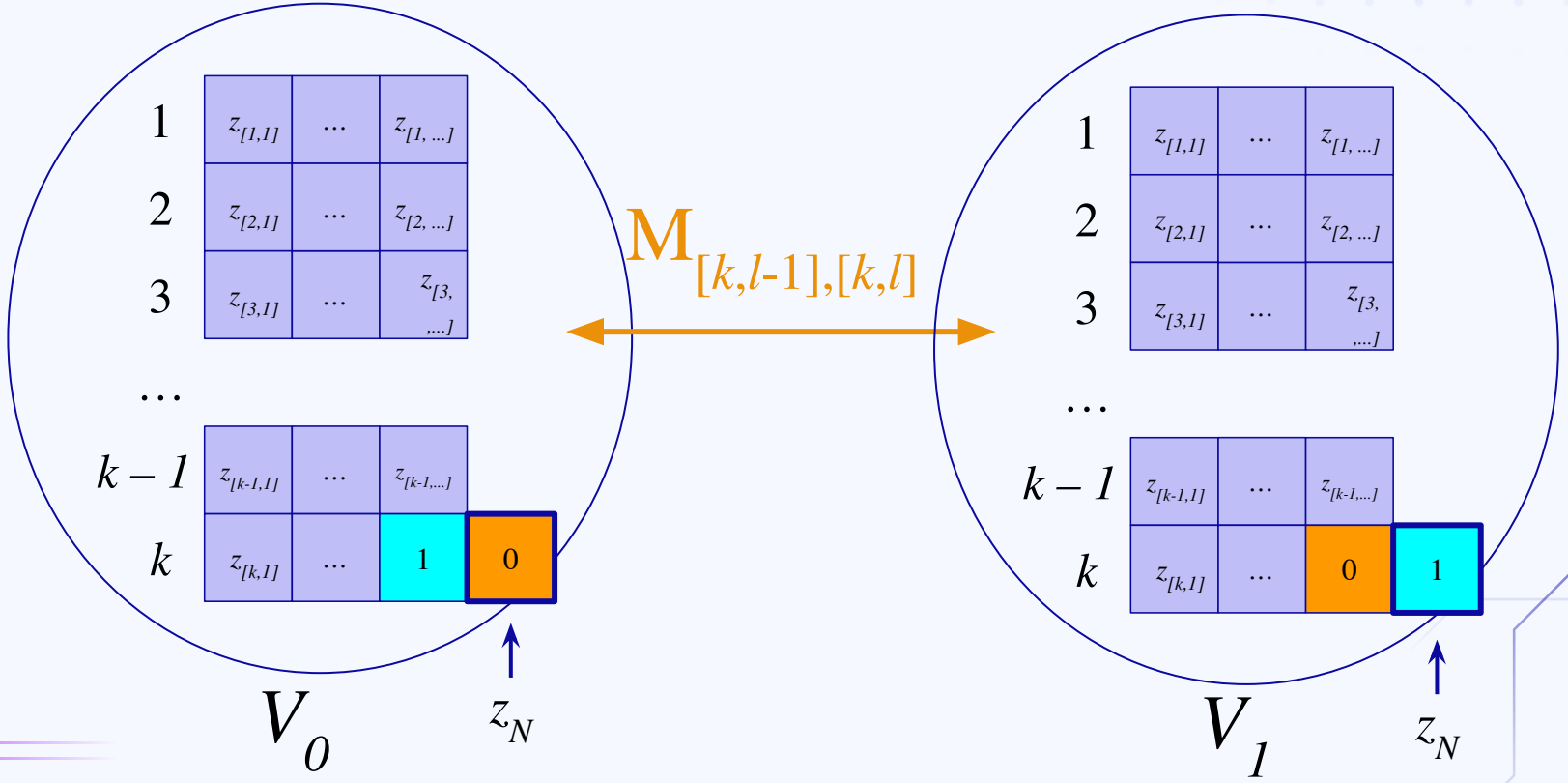


Case (1)

This corresponds to a vertex in $V_0$ where the blue cell = 1 (the image under the isomorphism $G^{(N-1,\,b)} \rightarrow V_0$ )

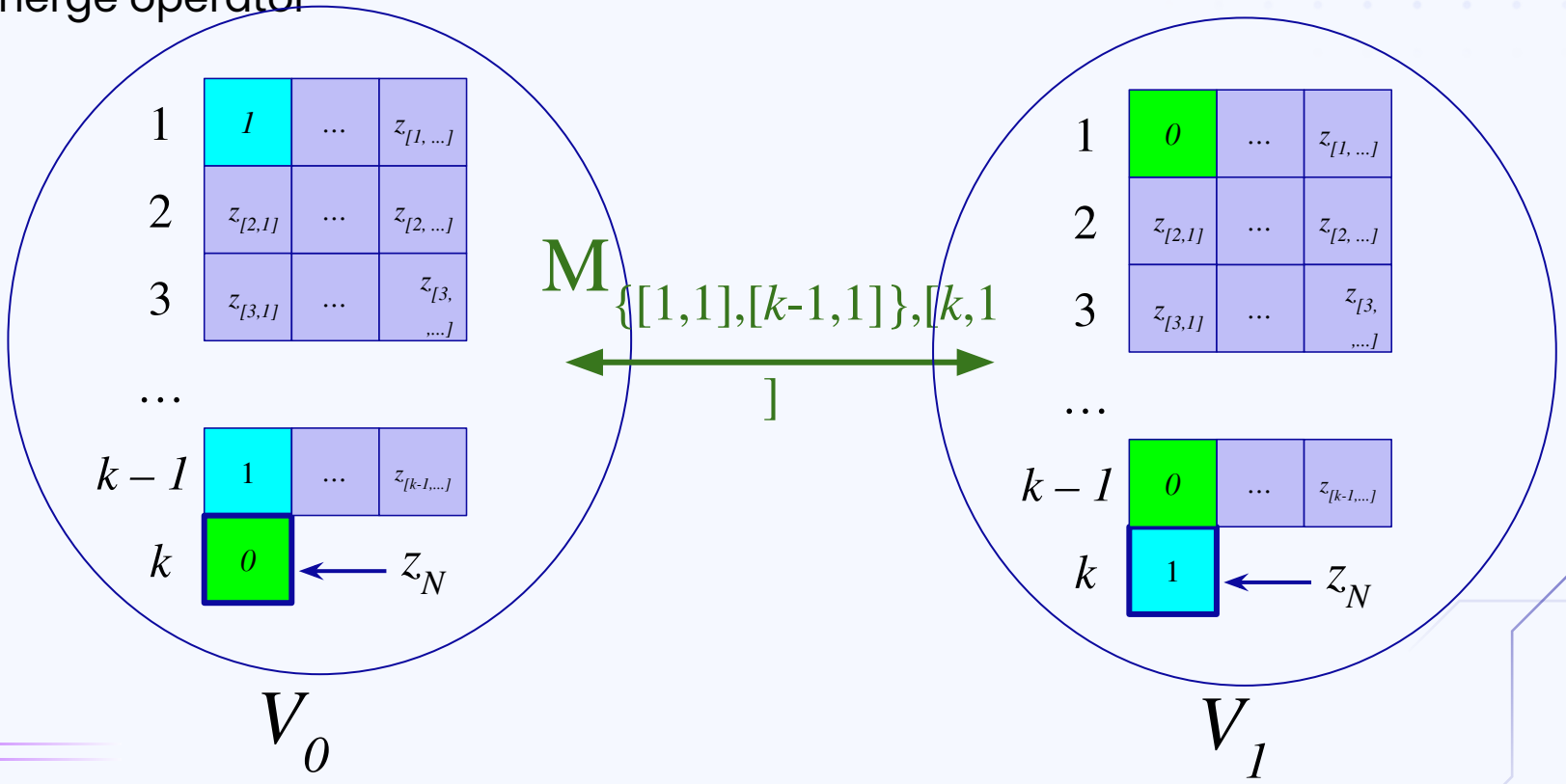


Case (1)
Apply a swap operation



Case (2) is basically the same:

Another lemma gives us a vertex in $G^{(N-1,\,b)} \cong V_0$ and this time we apply a merge operator



# Remains to prove:

(a) *Feasibility preservation: If* $|z\rangle \in \mathcal{S}$ *and* $|z'\rangle \notin \mathcal{S}$, *then* $\langle z| B_j |z'\rangle = 0$ *for each* $B_j$.

✅

(b) *Nonpositivity: The restriction of each* $B_j$ *to the solution subspace* $\mathcal{S}$, *denoted* $B_j|_{\mathcal{S}}$, *is component-wise nonpositive in the computational basis.*

✅

(c) *Full mixing of solutions: The basis-state-transition graph with adjacency matrix* $\sum_j B_j|_{\mathcal{S}}$ *is connected.*

✅

Future directions:

- More general constraints?
- Multiple constraints at once?
- Initial state preparation
- Parameter selection

# Thank You!

# Efficient Circuits:

$$U_B(\beta_l) = e^{-i\beta_l B}$$

| operator $M_{I,i*}$ | Pauli representation |
|---|---|
| $M_{a,b}$ | $2^{-1}(X_aX_b + Y_aY_b)$ |
| $M_{\{a,b\},c}$ | $2^{-2}(X_aX_bX_c - Y_aY_bX_c + Y_aX_bY_c + X_aY_bY_c)$ |
| $M_{\{a,b,c\},d}$ | $2^{-3}(X_aX_bX_cX_d - Y_aY_bX_cX_d - Y_aX_bY_cX_d - X_aY_bY_cX_d + Y_aX_bX_cY_d + X_aY_bX_cY_d + X_aX_bY_cY_d - Y_aY_bY_cY_d)$ |

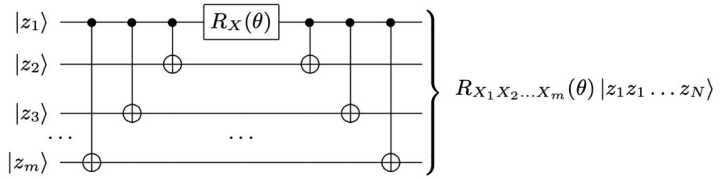



Figure 6: Quantum circuit diagram for an $m$-qubit simultaneous rotation gate about Pauli-$X$.

