SAND2020-0585PE

A Hybrid Quantum Optimization Algorithm Incorporating Classical Heuristics

Jaimie S. Stephens, William Bolden, and Ojas Parekh

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

First an Appetizer: Logical Cooling

J.S.S., Robin Blume-Kohout, Kevin Young, Craig W. Hogle, and Susan Clark

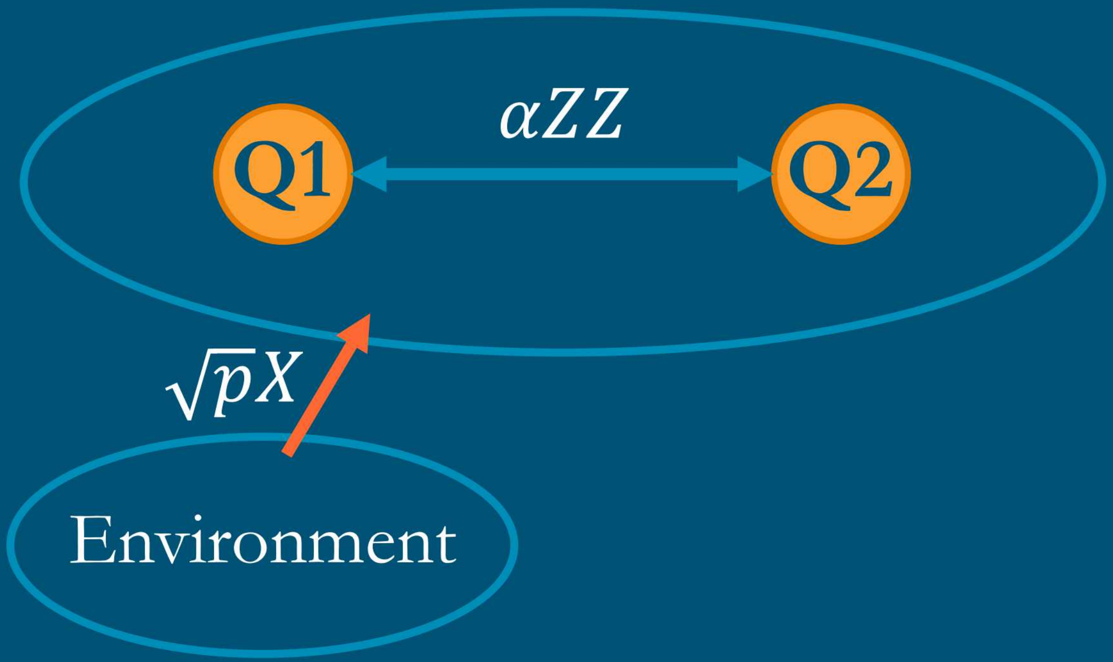
Motivation

- Analogue quantum simulations can teach us about fundamental quantum physics in regimes that are difficult to simulate classically such as quantum phase transitions near the critical point.

- Can't use fault-tolerant error correction due to continuous operations.

- Calculation length is limited by decoherence time, but error mitigation can lengthen the decoherence time.

Logical Cooling



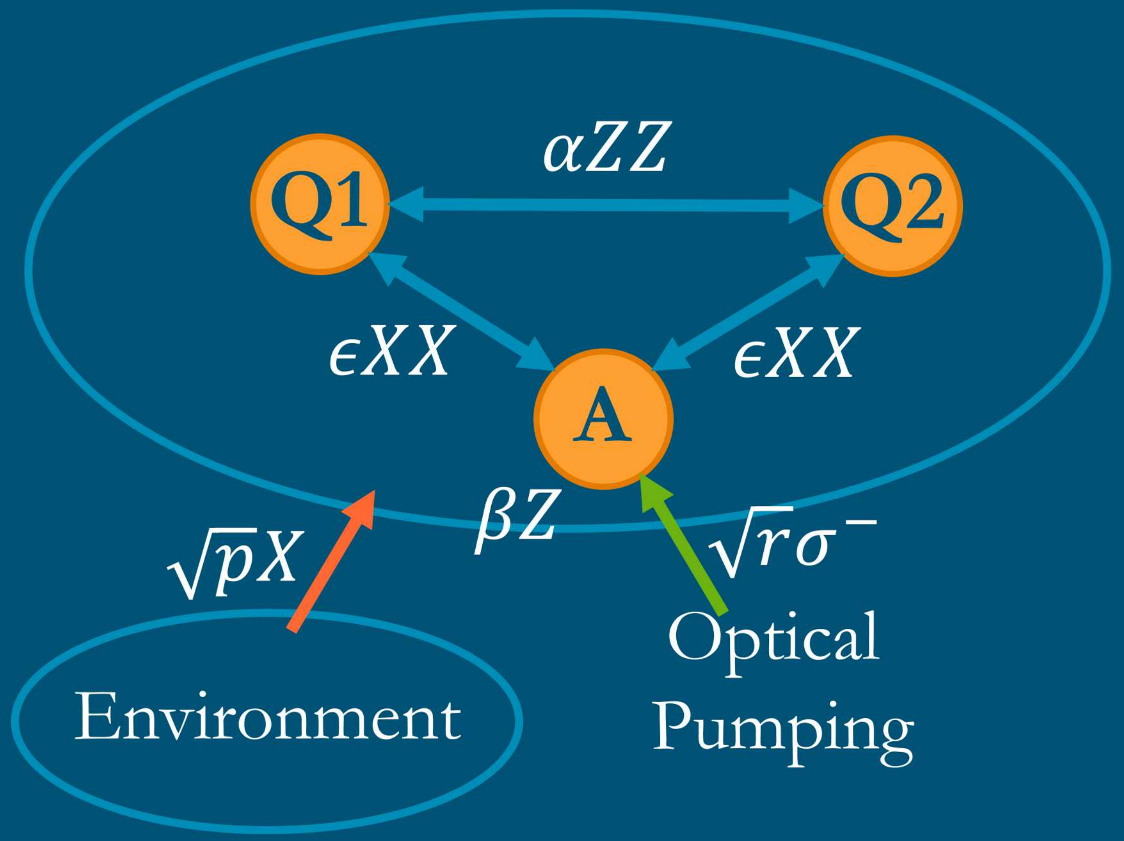
- Find ground state of

$$H = \alpha Z_1 Z_2.$$

- Decoherence from environment

$$L_\mu \in \{\sqrt{p}X_1, \sqrt{p}X_2\}$$

Logical Cooling



- Evolve under:

$$H = \alpha Z_1 Z_2 + \beta Z_A + \epsilon(X_1 + X_2)X_A$$

- 1st term: Hamiltonian in question
- 2nd term: ancilla energy splitting
- 3rd term: transfer heat from system to the ancilla
- Decoherence:

$$L_\mu \in \{\sqrt{p}X_1, \sqrt{p}X_2, \sqrt{p}X_A\}$$

- Cooling the ancilla:

$$L_\mu \in \{\sqrt{\mathrm{r}}\sigma_{\mathrm{A}}^{-}\}$$

Master Equation

$$\frac{d\rho}{dt} = -i[(\alpha Z_1 Z_2 - \beta Z_A), \rho]$$
$$-i\epsilon[(X_1 + X_2)X_A, \rho\ ]$$
$$-3p\rho + p(X_1 \rho X_1\ + X_2 \rho X_2 + X_A \rho X_A)$$
$$-\frac{r}{4}(\sigma_A^- \sigma_A^+ \rho + \rho \sigma_A^- \sigma_A^+ + \sigma_A^- \rho \sigma_A^+)$$

Energy splittings

Transfer heat to ancilla

Decoherence

Optical pumping

Why Should Logical Cooling Work?

Interaction picture:

$$X_S X_A \to \left(e^{i\alpha t}\sigma_S^+ + e^{-i\alpha t}\sigma_S^-\right)\left(e^{i\beta t}\sigma_A^+ + e^{-i\beta t}\sigma_A^-\right)$$

$$= e^{it(\alpha+\beta)}\sigma_S^+\sigma_A^+ + e^{it(\alpha-\beta)}\sigma_S^+\sigma_A^- + e^{-it(\alpha-\beta)}\sigma_S^-\sigma_A^+ + e^{-it(\alpha+\beta)}\sigma_S^-\sigma_A^-$$

$$= e^{it2\alpha}\sigma_S^+\sigma_A^+ + \sigma_S^+\sigma_A^- + \sigma_S^-\sigma_A^+ + e^{-it2\alpha}\sigma_S^-\sigma_A^-$$

$\alpha = \beta$

Dominating term swaps ancilla and system qubit

Rapidly rotating terms approximate to nothing

Now the Main Course: Hybrid QAOA Incorporating Classical Heuristics

J.S.S., William Bolden, and Ojas Parekh

Motivation

- QAOA can approximate the solution to NP-Hard problems.
- Can QAOA yield a better **approximation ratio** than classical algorithms with equal time complexities?
- QAOA circuits are low in depth. → QAOA could be a good NISQ application.
- We know that QAOA even for $p = 1$ prepares a state that is classically hard to sample from [1].

[1] E. Farhi and A. W. Harrow (2016)

Approximation Ratio

$$\Gamma \equiv \frac{C(x^*)}{\min_x C(x)}, \text{ or } \Gamma \equiv \frac{C(x^*)}{\max_x C(x)}$$

If the objective function $C(x)$ is being minimized, then $\Gamma \geq 1$.

If the objective function $C(x)$ is being maximized, then $\Gamma \leq 1$.

The Quantum Approximate Optimization Algorithm (QAOA)

[1] E. Farhi, *et. al.* (2014)

QAOA

- Given $C: \{0,1\}^n \to \mathbb{R}$, find $x \in \{0,1\}^n$ s.t. $C(x)$ is a minimum.

- We can represent $C(x)$ as an objective Hamiltonian,
$$H_C|x\rangle = C(x)|x\rangle.$$

- Let the mixing Hamiltonian be
$$H_M = \sum_{j=1}^{n} X_j\,.$$

QAOA

Let $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_p)$ be free parameters.

Then, the Q_p operator is defined as

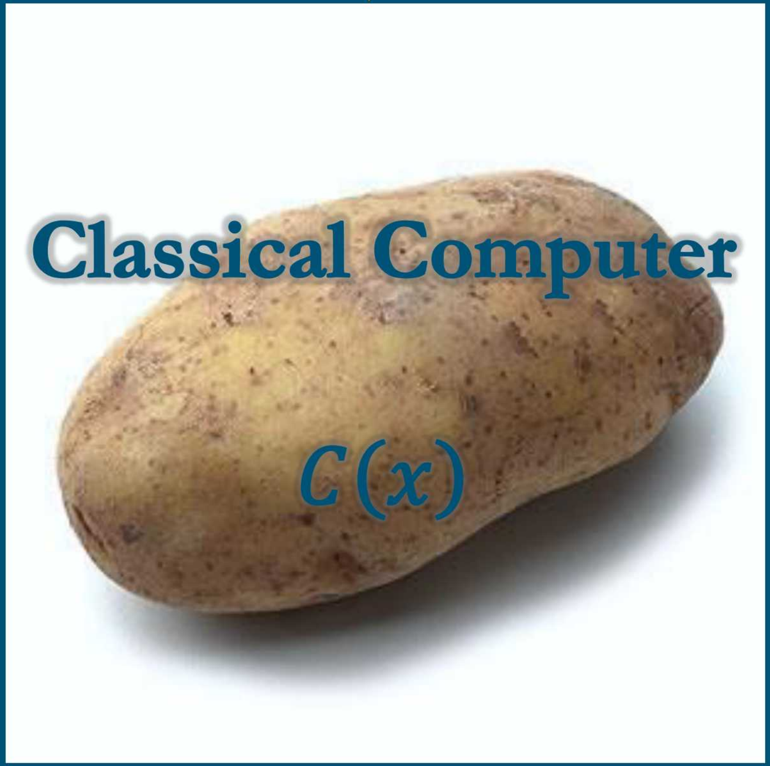
$$Q_p(\boldsymbol{\beta}, \boldsymbol{\gamma}) \equiv \prod_{k=1}^{p} e^{-i\beta_k H_M} e^{-i\gamma_k H_C}.$$

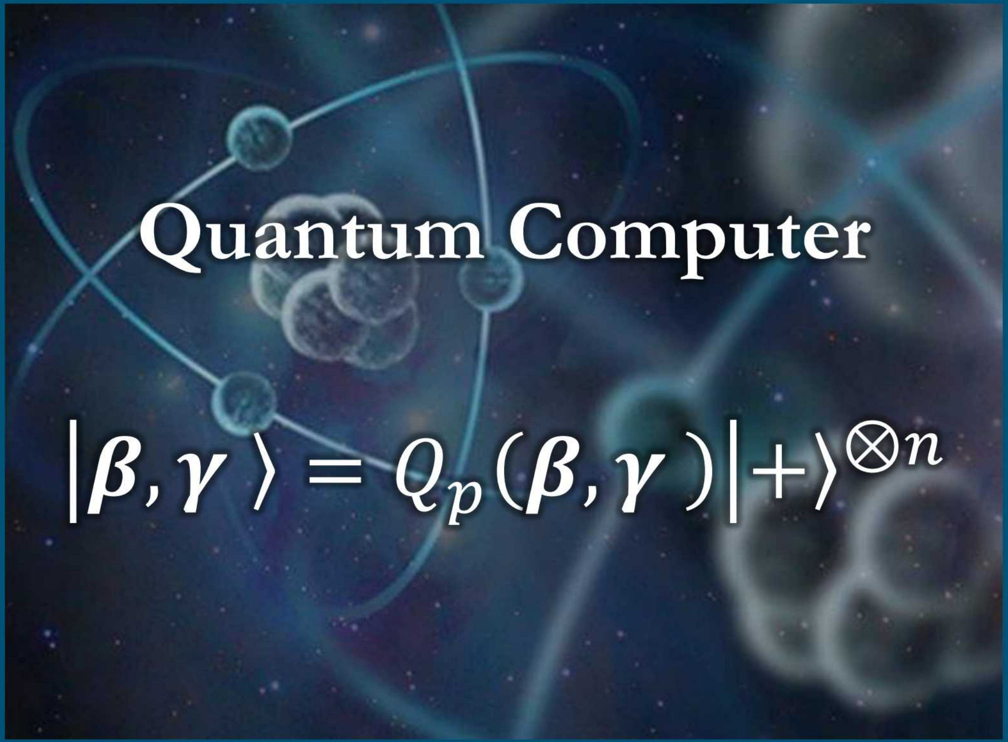
QAOA is a Variational Algorithm

C

H_C

$\boldsymbol{\beta}, \boldsymbol{\gamma}$





x

Goal

- Find good $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ for some number of iterations p such that the expectation value of the problem Hamiltonian is close to the minimum.
- i.e. The approximation ratio is close to 1.

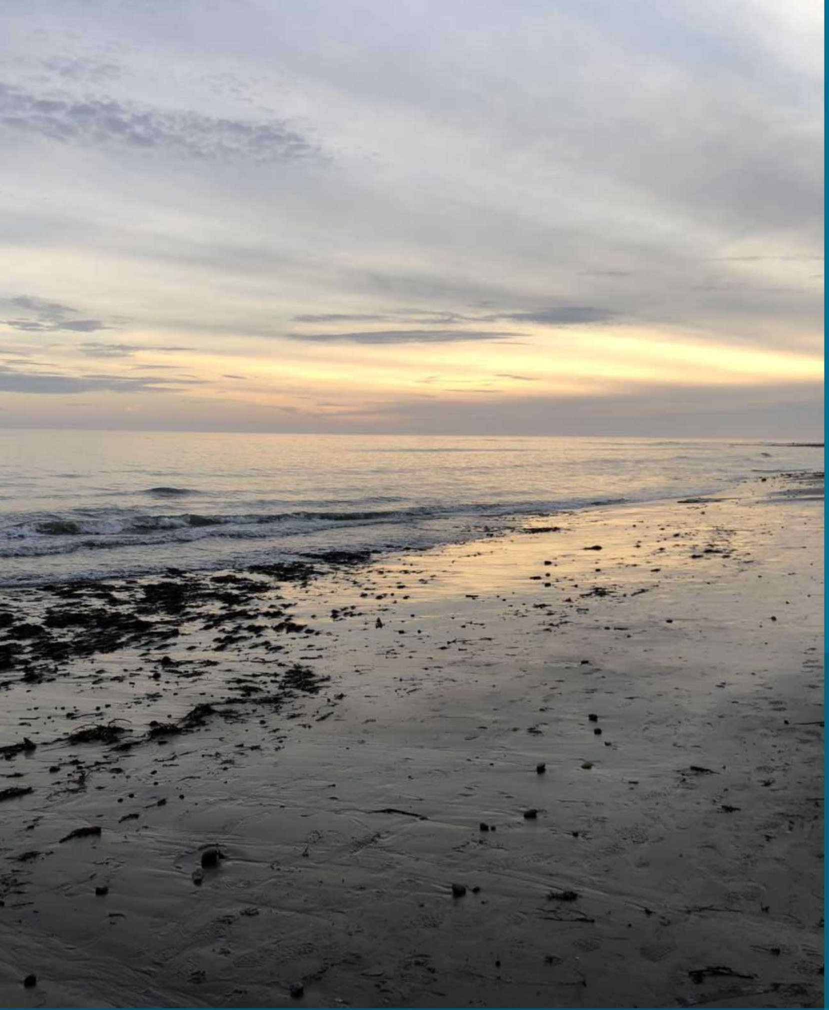
$$\Gamma = \frac{\min_{\beta,\gamma} \langle \boldsymbol{\beta}, \boldsymbol{\gamma} | H_C | \boldsymbol{\beta}, \boldsymbol{\gamma} \rangle}{\min_x C(x)}$$

Does The QAOA Work?

- From Trotterization, it follows that we obtain the optimal solution when $p \to \infty$. But, we want to run on NISQ devices. $\Longrightarrow$ Circuit depth $\propto p$ should be small.

- Historically, it is a pattern to show the QAOA does better on a problem only for a better classical algorithm to be discovered [1], [2], [3], and [4].
  - **We look at a way to improve QAOA's performance.**

[1] E. Farhi *et. al.* (2014), [2] Barak et. al. (2015), [3] Ryan-Anderson (2018) Ch. 3, [4] Z. Wang *et. al.* (2017)

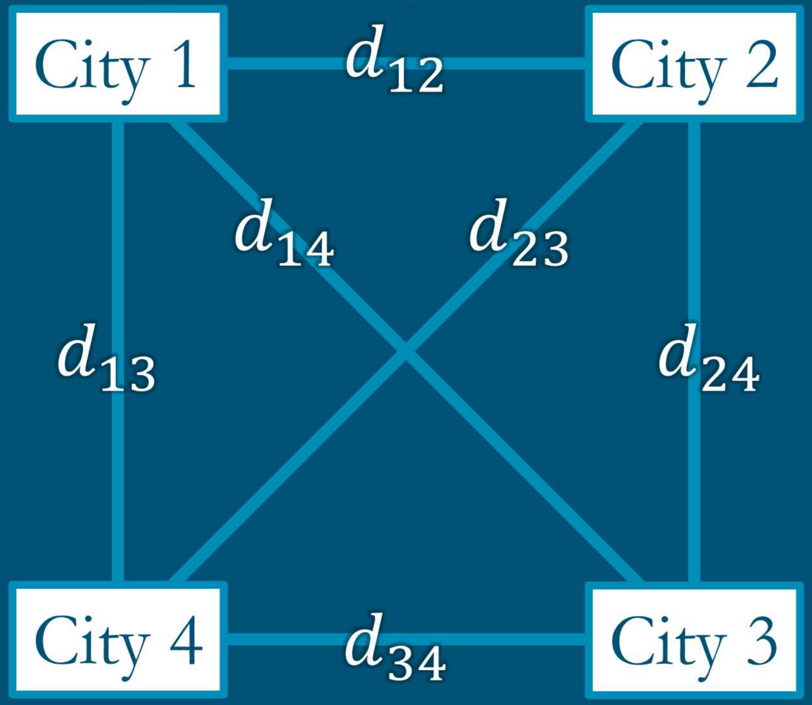


New: QAOA with Classical Heuristics

Example Classical Heuristic: Local Search

- In general, local search makes small changes to the input solution to see if the objective can be improved.

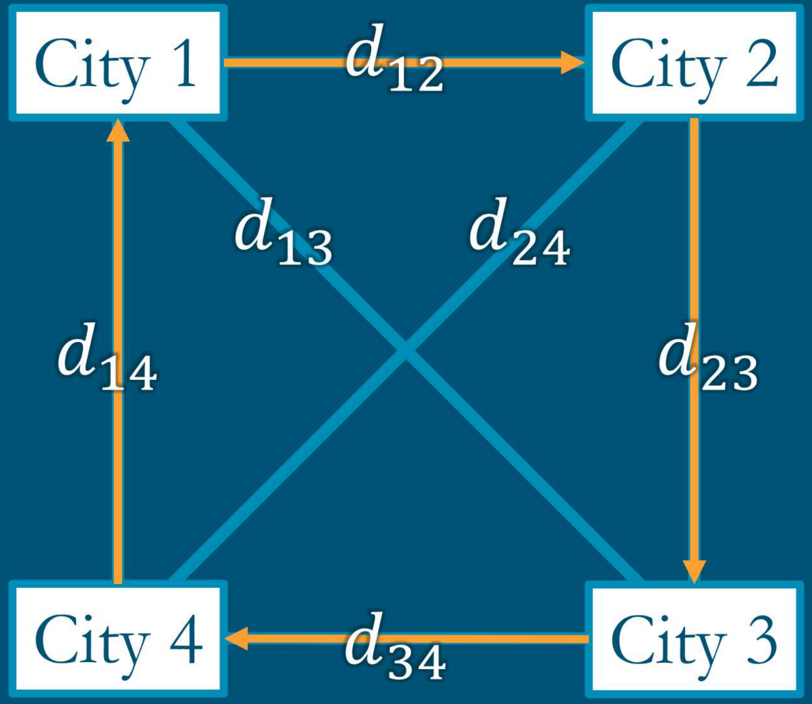
- Traveling Sales Person (TSP) example: Find the shortest distance one can travel to visit each city exactly once.



Example Classical Heuristic: Local Search

- In general, local search makes small changes to the input solution to see if the objective can be improved.

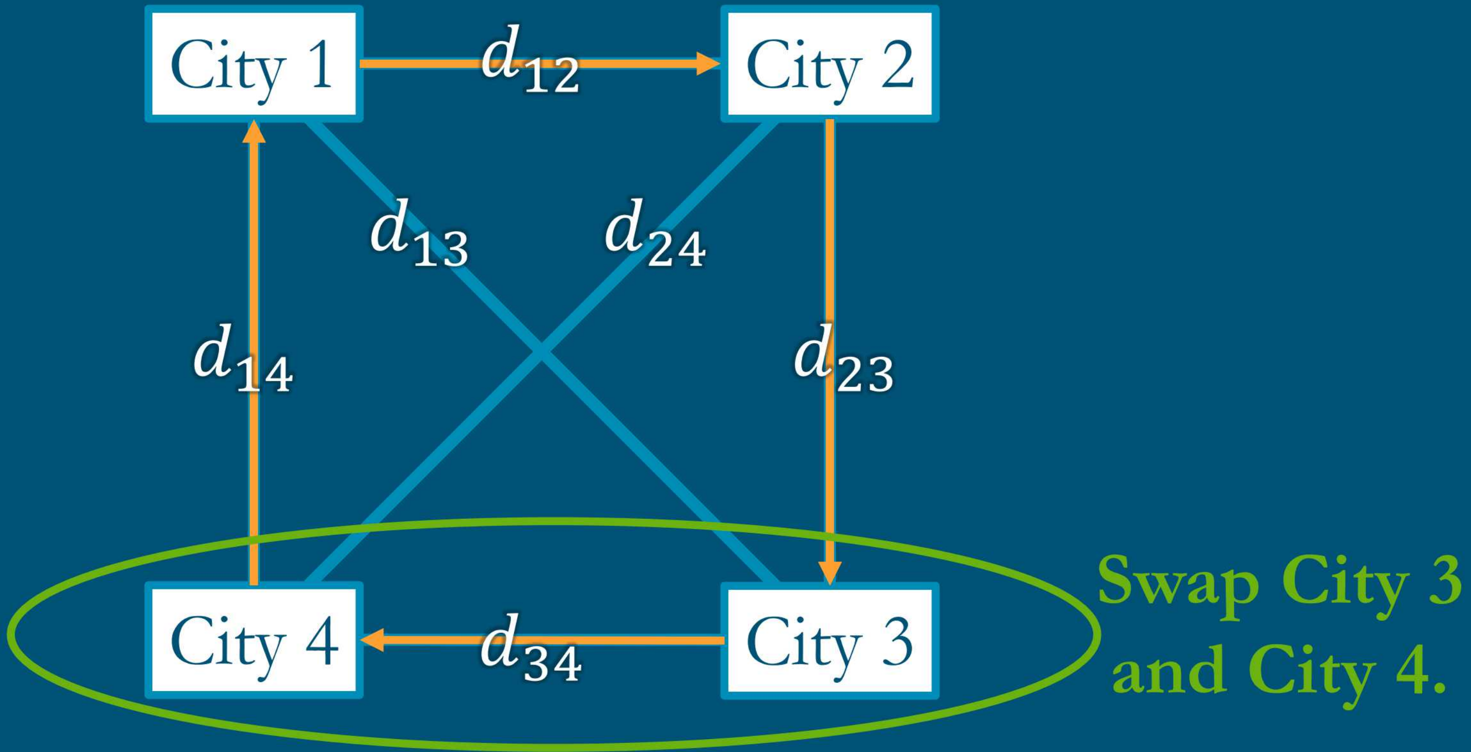
- Traveling Sales Person (TSP) example: $D = d_{12} + d_{23} + d_{34} + d_{14}$



Example Classical Heuristic: Local Search

- In general, local search makes small changes to the input solution to see if the objective can be improved.

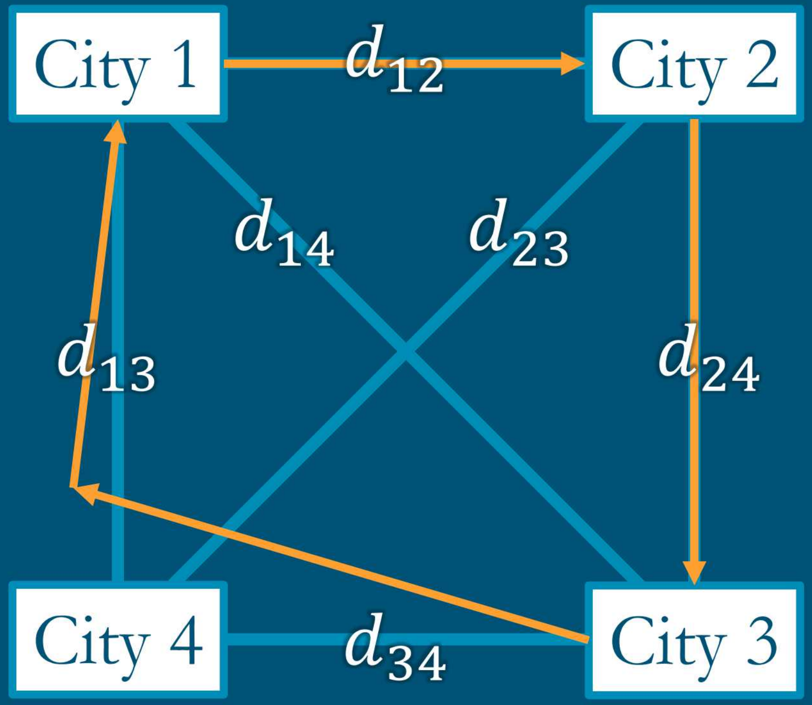
- Traveling Sales Person (TSP) example: $D = d_{12} + d_{23} + d_{34} + d_{14}$



Example Classical Heuristic: Local Search

- In general, local search makes small changes to the input solution to see if the objective can be improved.

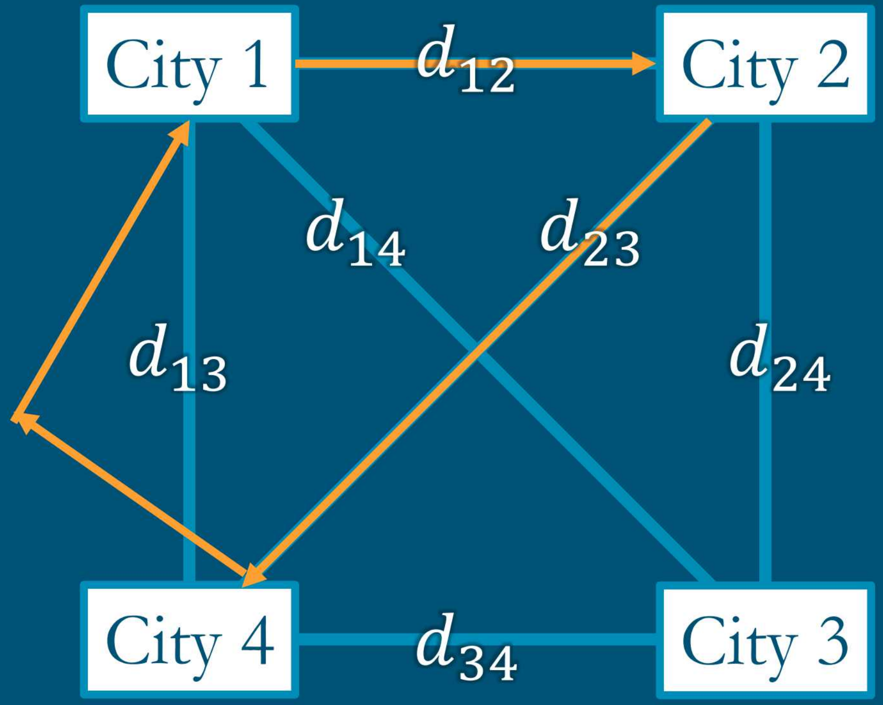
- Traveling Sales Person (TSP) example:



Example Classical Heuristic: Local Search

- In general, local search makes small changes to the input solution to see if the objective can be improved.

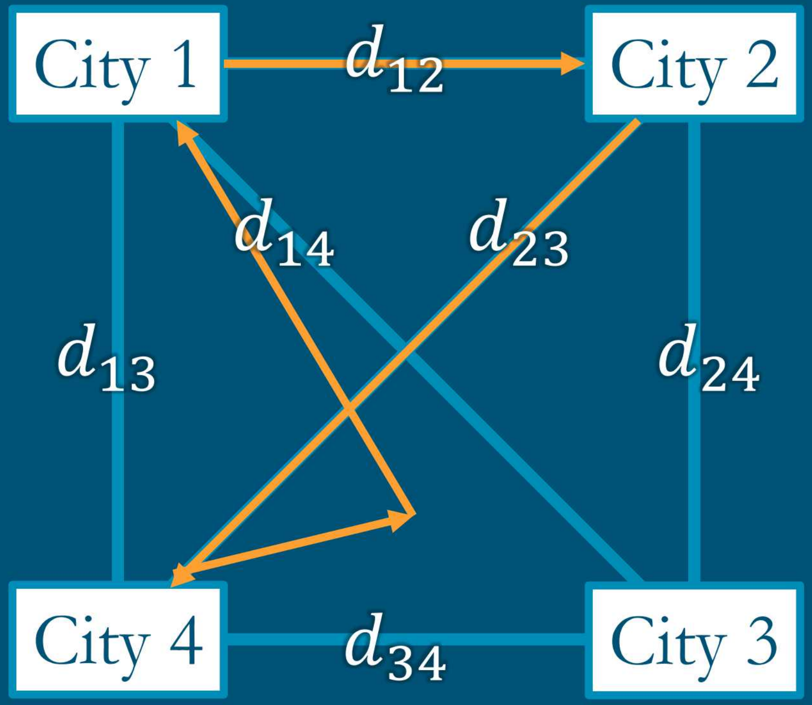
- Traveling Sales Person (TSP) example:



Example Classical Heuristic: Local Search

- In general, local search makes small changes to the input solution to see if the objective can be improved.

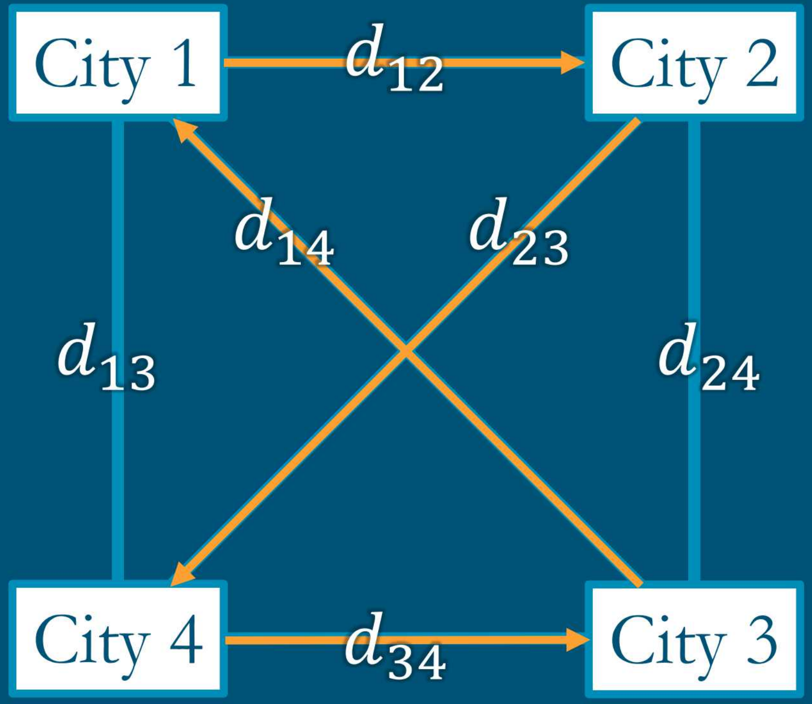
- Traveling Sales Person (TSP) example:



Example Classical Heuristic: Local Search

- In general, local search makes small changes to the input solution to see if the objective can be improved.

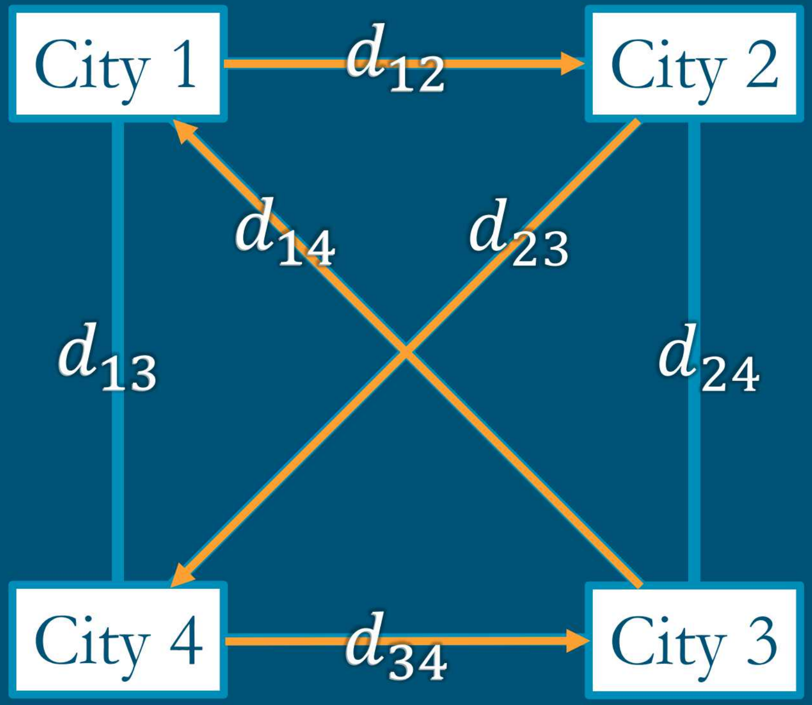
- Traveling Sales Person (TSP) example: $D_2 = d_{12} + d_{23} + d_{34} + d_{14}$



Example Classical Heuristic: Local Search

- In general, local search makes small changes to the input solution to see if the objective can be improved.

- Traveling Sales Person (TSP) example: If $D_2 < D_1$, change solution.

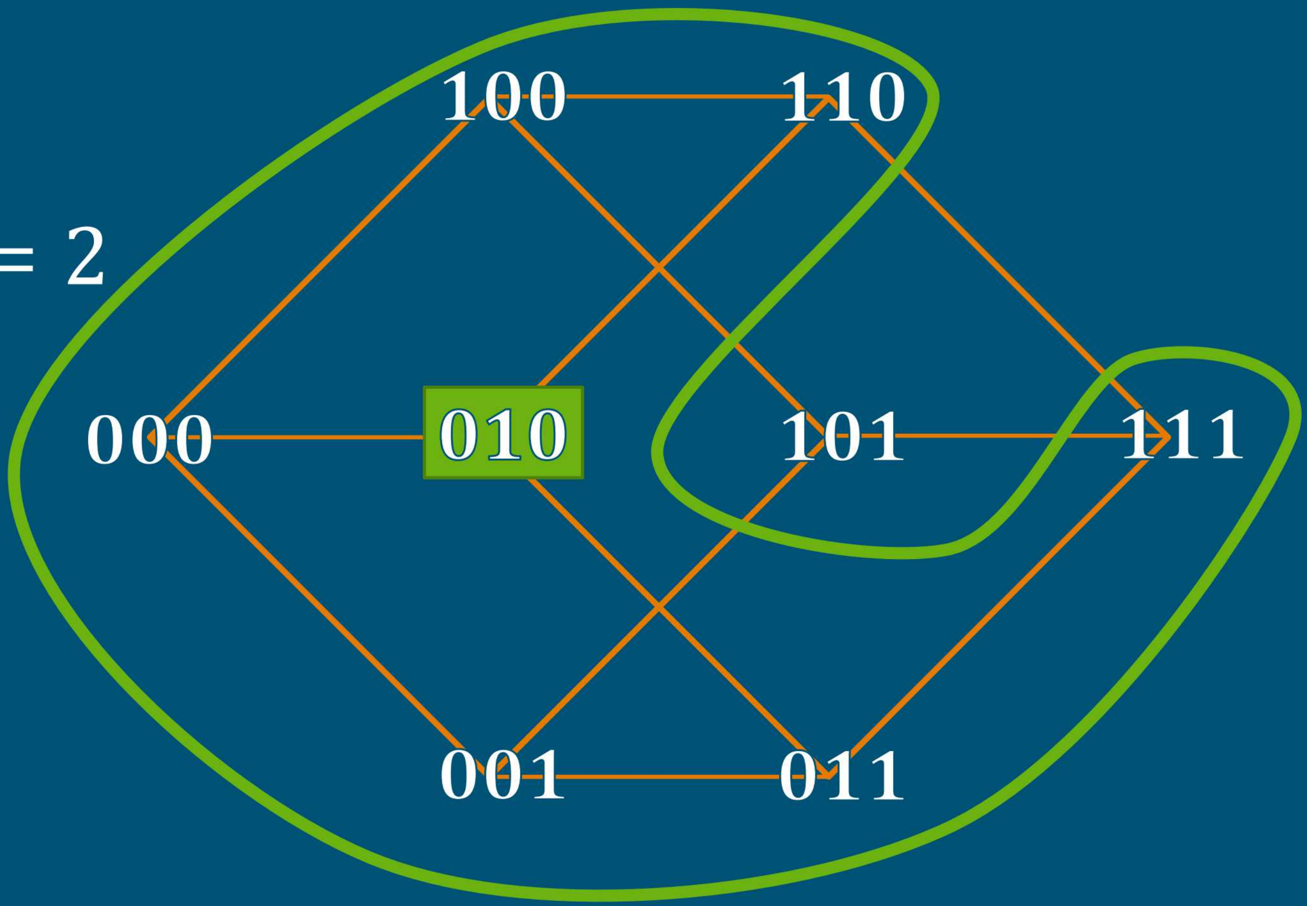


Local Search on Bit Strings

Local search of depth k searches all bit strings within hamming distance k of the input string.

Input: $x = 010$

Search depth: $k = 2$



Why Heuristics Instead of Approximation Algorithms?

Heuristics

- Intuitive
- Perform well on practical instances
- Difficult to prove bounds on approximation ratio

Approximation Algorithms

- May perform poorly compared to heuristics
- Rigorous bound on worst-case performance
- Designed by performance proof → Can be less intuitive

Hybrid QAOA Classical Heuristic

- Let $h: \{0,1\}^{\otimes n} \rightarrow \{0,1\}^{\otimes n}$ be a classical heuristic.
- For **unconstrained** problems, we choose a heuristic that **improves the output of objective function.**
- For **constrained** problems, we choose a heuristic that **maps infeasible states to feasible states.**

Hybrid QAOA Classical Heuristic

Now define our new objective Hamiltonian,

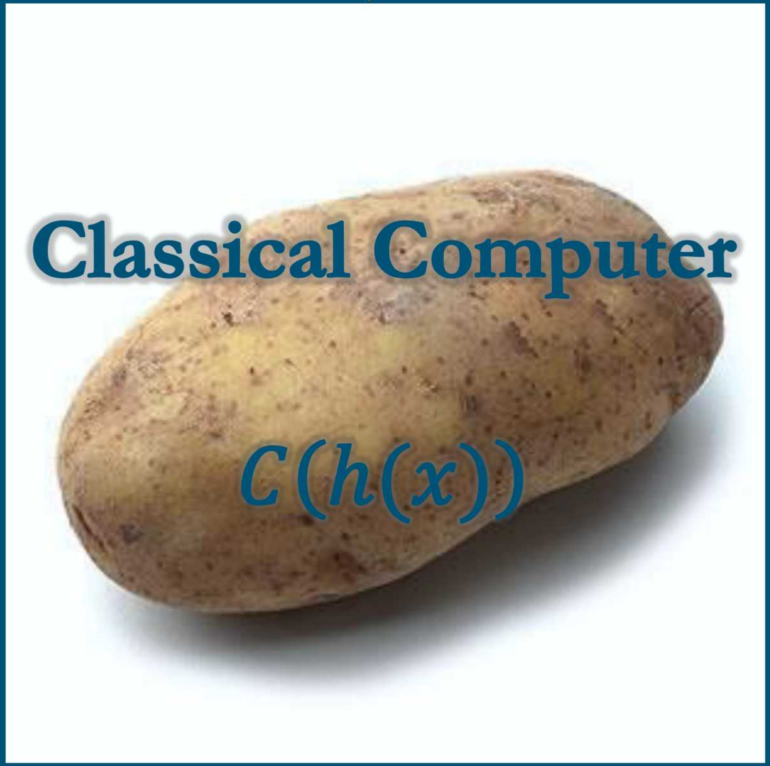
$$H_C' = \sum_{x=0}^{2^n-1} C\big(h(x)\big)|x\rangle\langle x|\,.$$

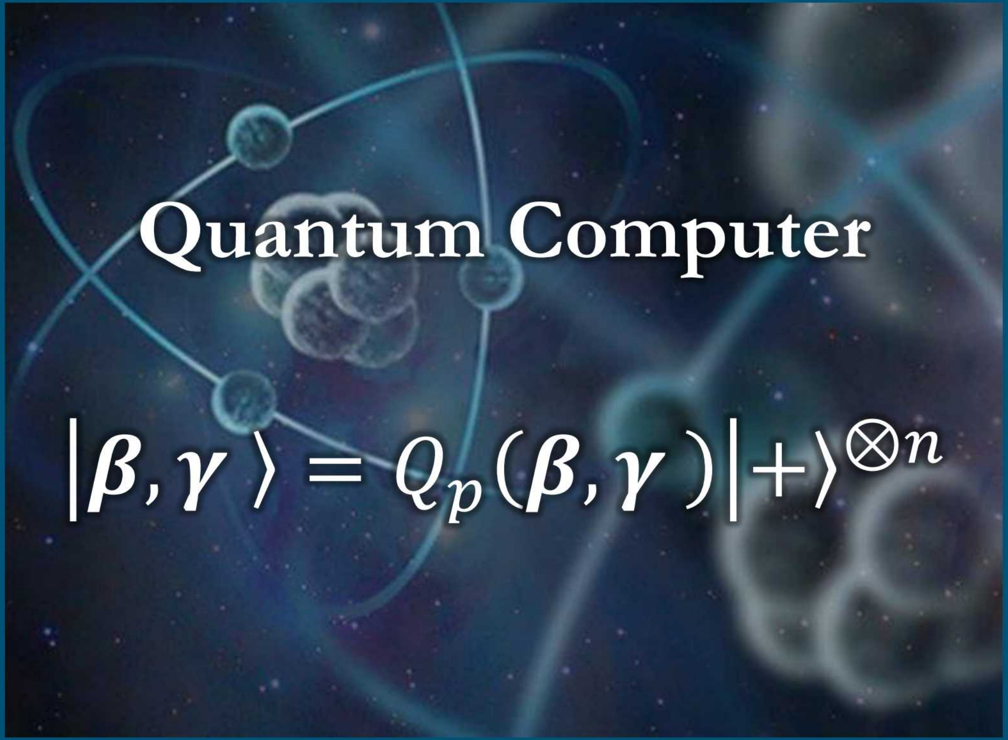
Hybrid QAOA Classical Heuristics

C, h

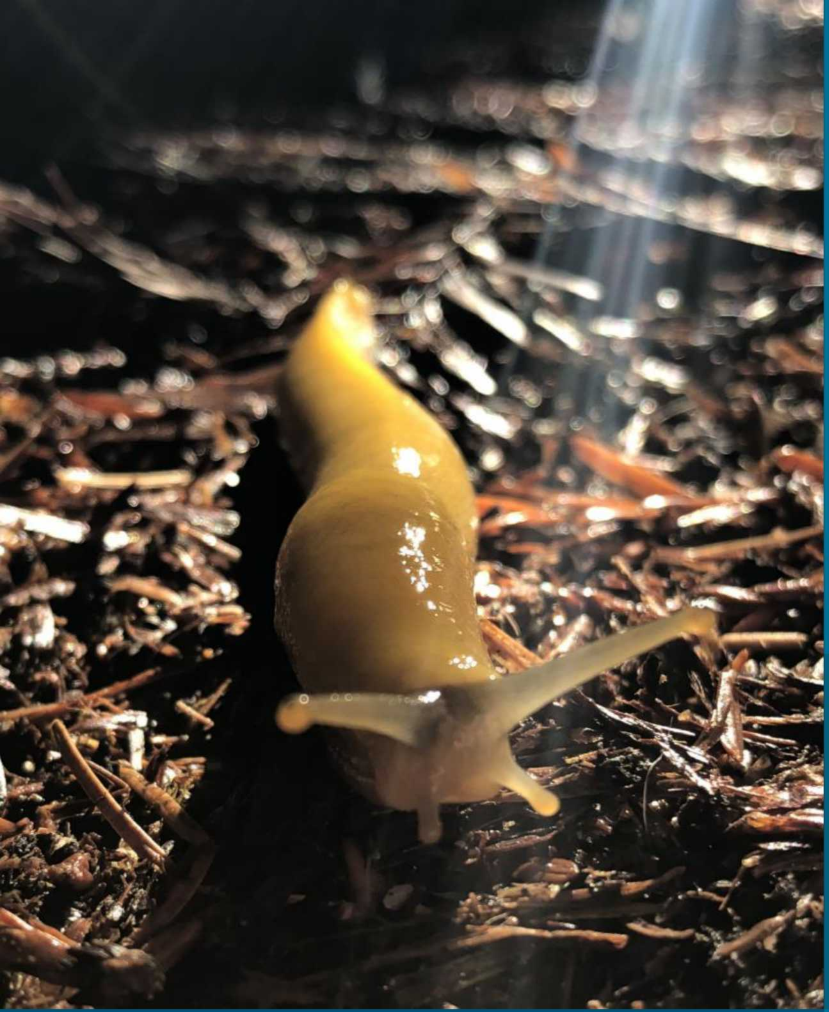
$\boldsymbol{\beta}, \boldsymbol{\gamma}$

H_C'





x

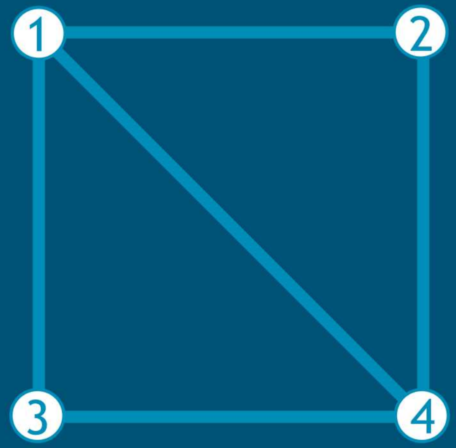


MaxCut Example

MAXCUT

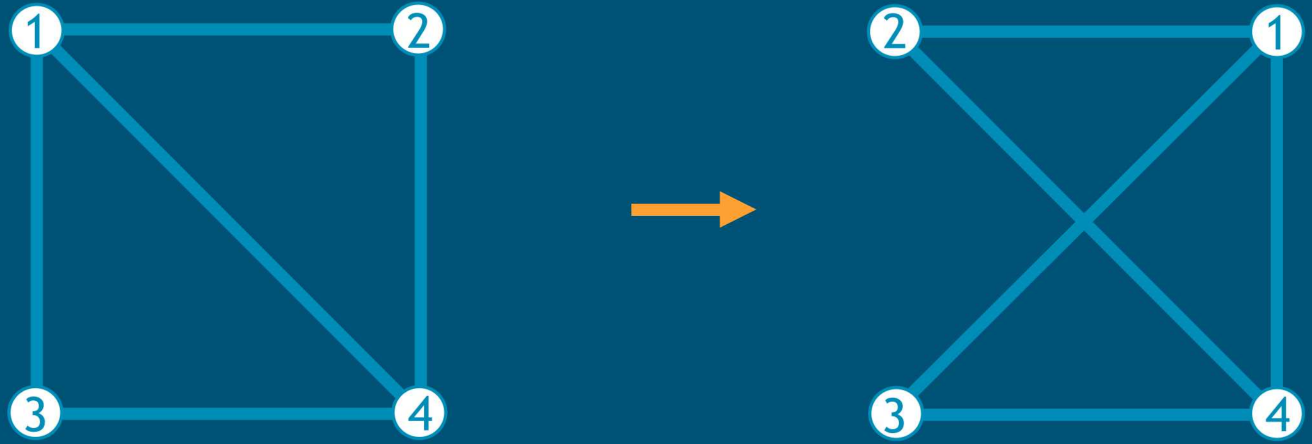
- Given a graph $G = (V, E)$, we want to partition the vertices into two sets $V' \subseteq V$ and $V \backslash V'$, s.t. the number of edges **cut** is maximized.
- An edge, $\{u, v\} \in E$, is **cut** if $u \in V'$ and $v \in V \backslash V'$ or vice versa.

- Example: MAXCUT = ?



MaxCut

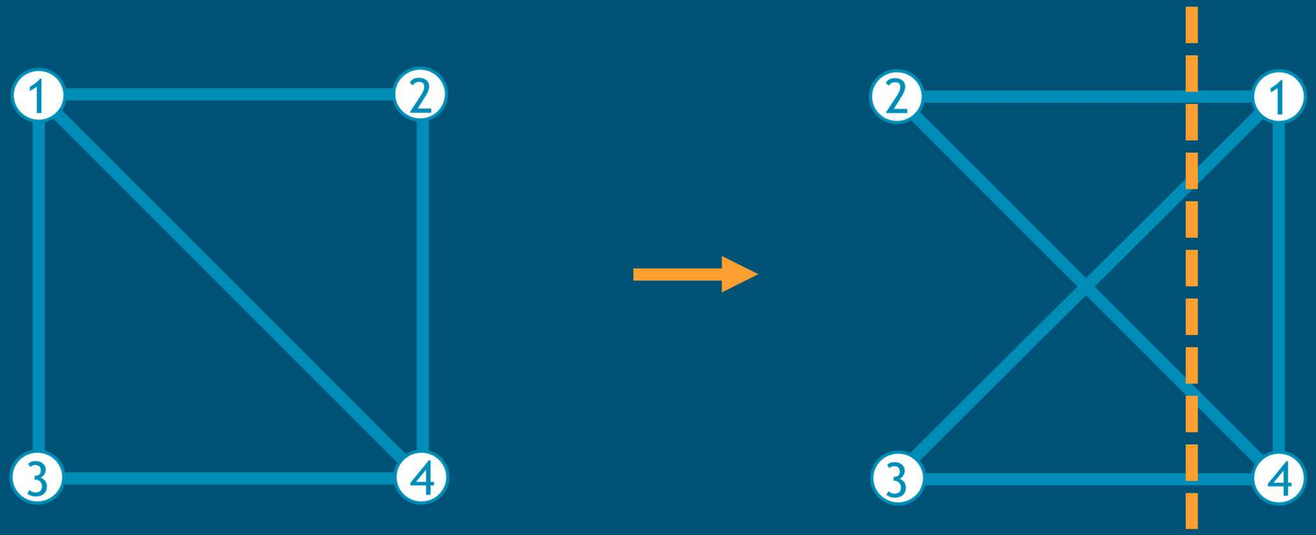
- Given a graph $G = (V, E)$, we want to partition the vertices into two sets $V' \subseteq V$ and $V \backslash V'$, s.t. the number of edges **cut** is maximized.
- An edge, $\{u, v\} \in E$, is **cut** if $u \in V'$ and $v \in V \backslash V'$ or vice versa.
- Example: MAXCUT = ?



MAXCUT

- Given a graph $G = (V, E)$, we want to partition the vertices into two sets $V' \subseteq V$ and $V \backslash V'$, s.t. the number of edges **cut** is maximized.
- An edge, $\{u, v\} \in E$, is **cut** if $u \in V'$ and $v \in V \backslash V'$ or vice versa.

- Example: MAXCUT = ?

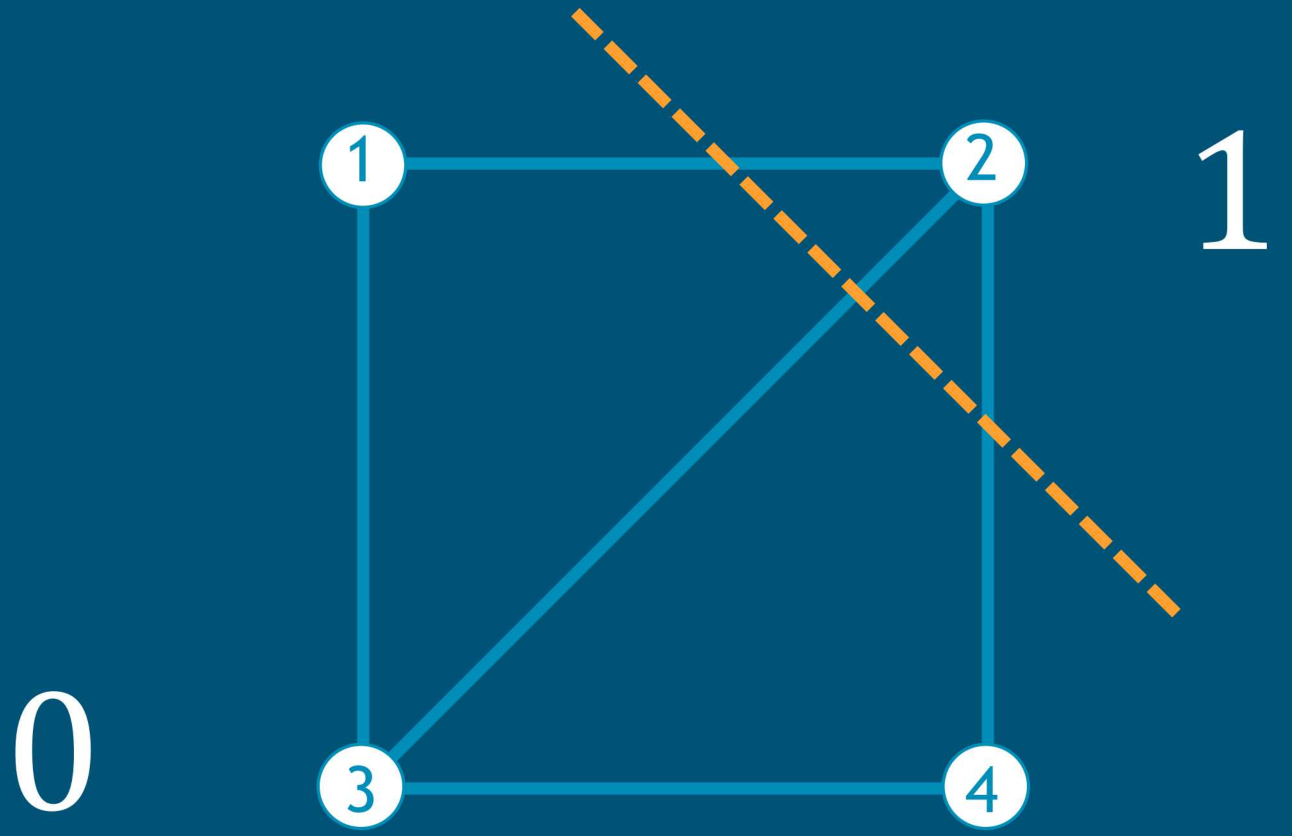


MAXCUT Facts

- Decision problem related to MAXCUT is NP-complete. [1]
- MAXCUT is APX-complete. It can be approximated in polynomial time to within a constant factor. [2]
- It is NP-Hard to approximate MAXCUT with an approximation ratio $\Gamma > {}^{16}/_{17} \approx 0.9412$. [3]

[1] M. R. Gareyand D. S. Johnson (1979), [2] C. Papadimitriou and M. Yannakakis (1991), [3] J.Håstad (2001)

Mapping to Qubits

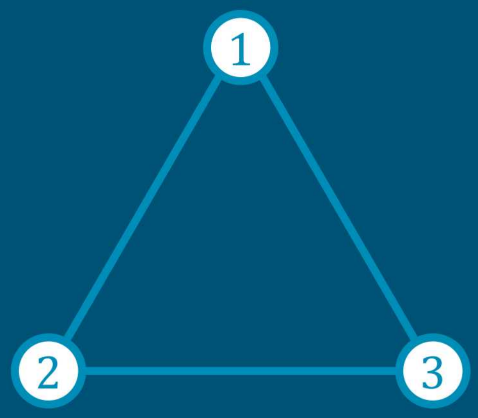


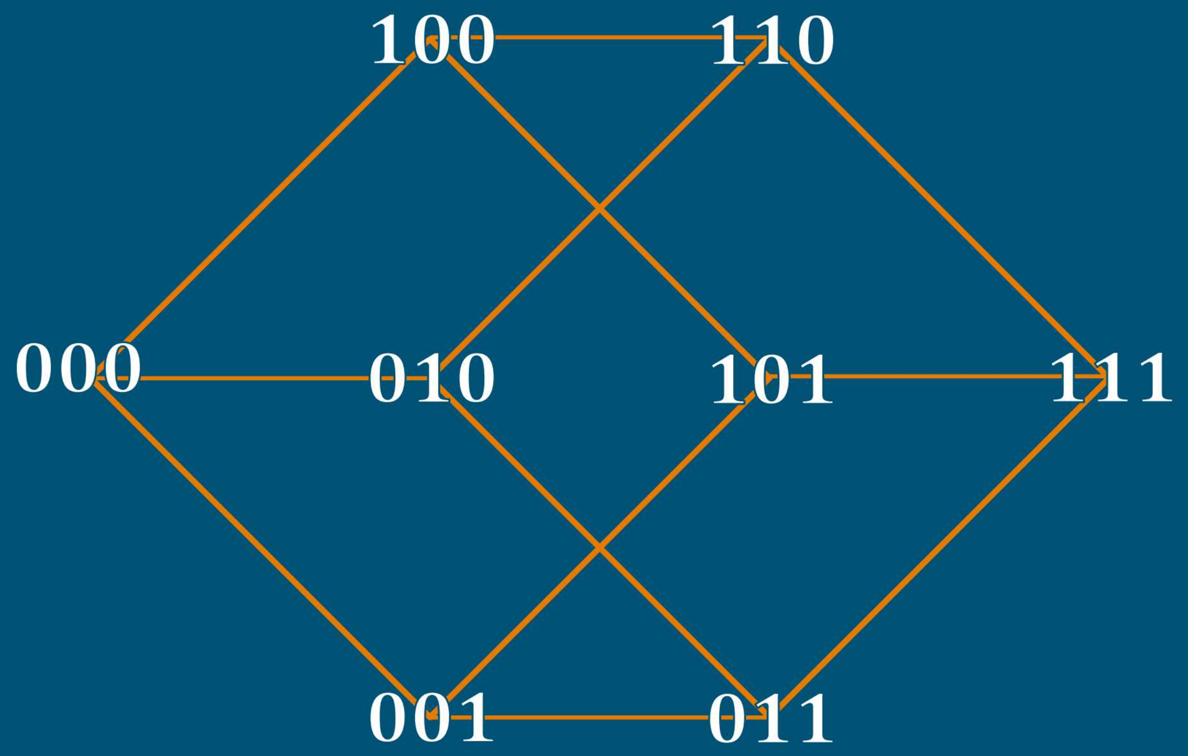
$$|0100\rangle$$

1 2 3 4

Local Search Heuristic for MAXCUT

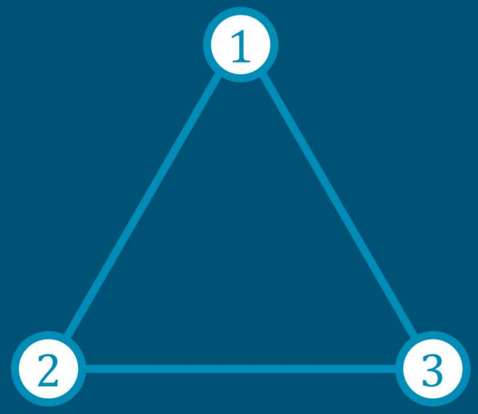
- $C(x) = \sum_{(\mu,\nu)\in E}(1 - x_\mu x_\nu)$
- $k = 1$

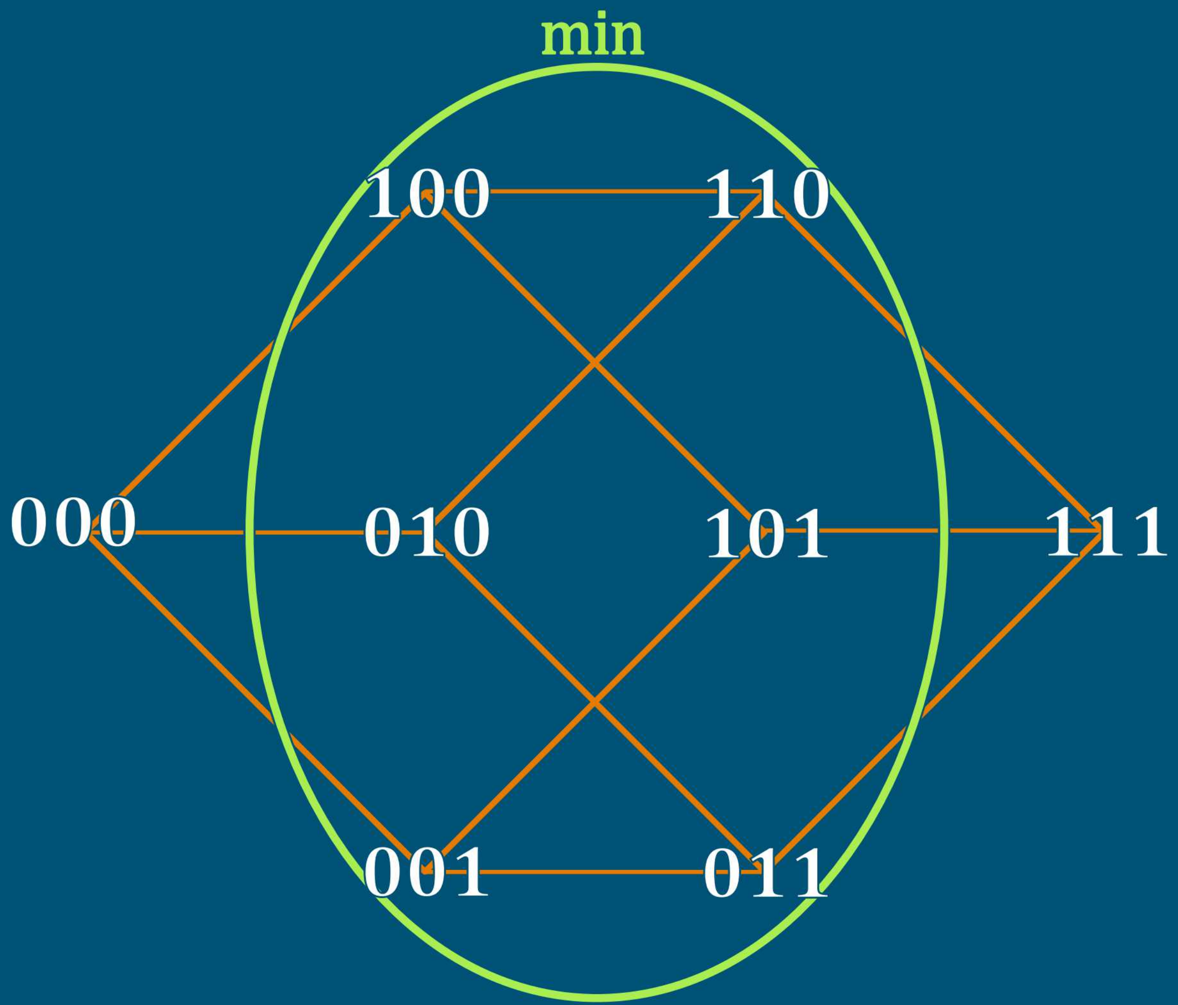




Local Search Heuristic for MAXCUT

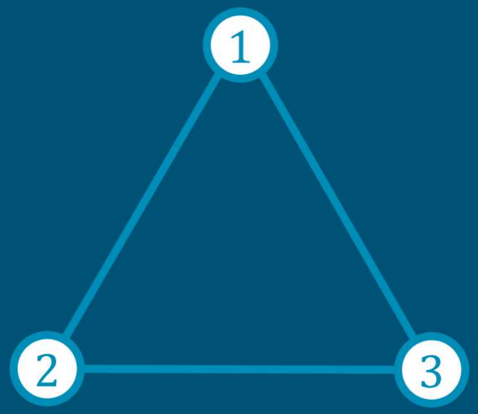
- $C(x) = \sum_{(\mu,\nu)\in E}(1 - x_\mu x_\nu)$
- $k = 1$

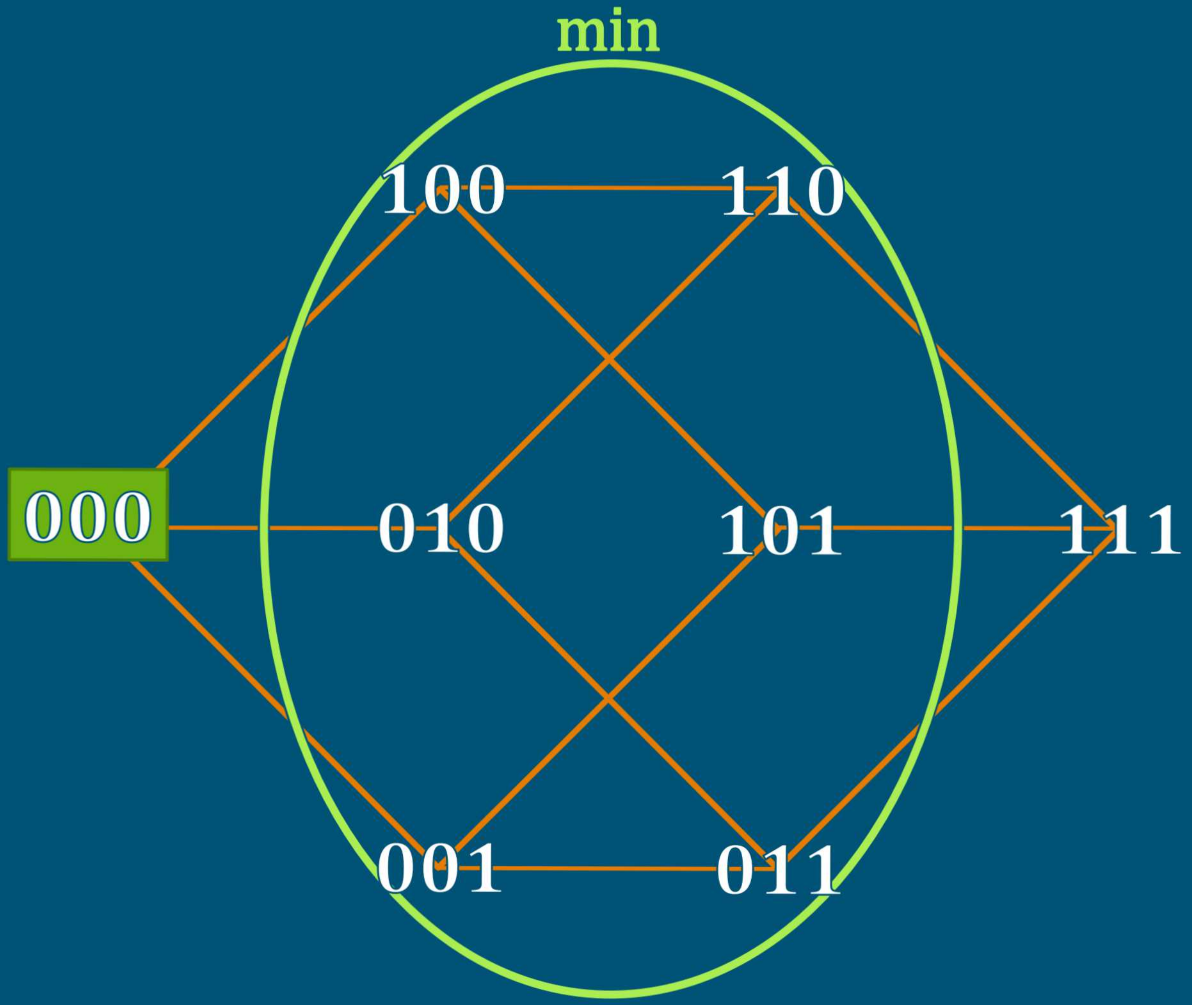




Local Search Heuristic for MAXCUT

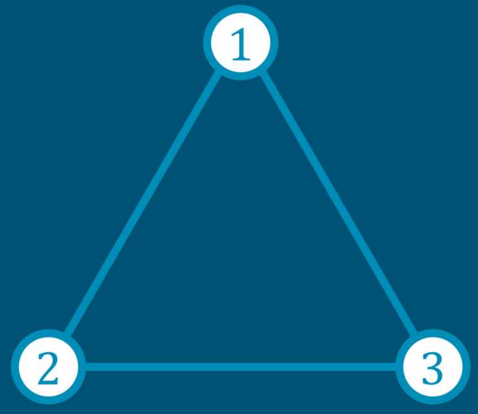
- $C(x) = \sum_{(\mu,\nu)\in E}(1 - x_\mu x_\nu)$
- $k = 1$
- $x = 000$
- $C(000) = 0$

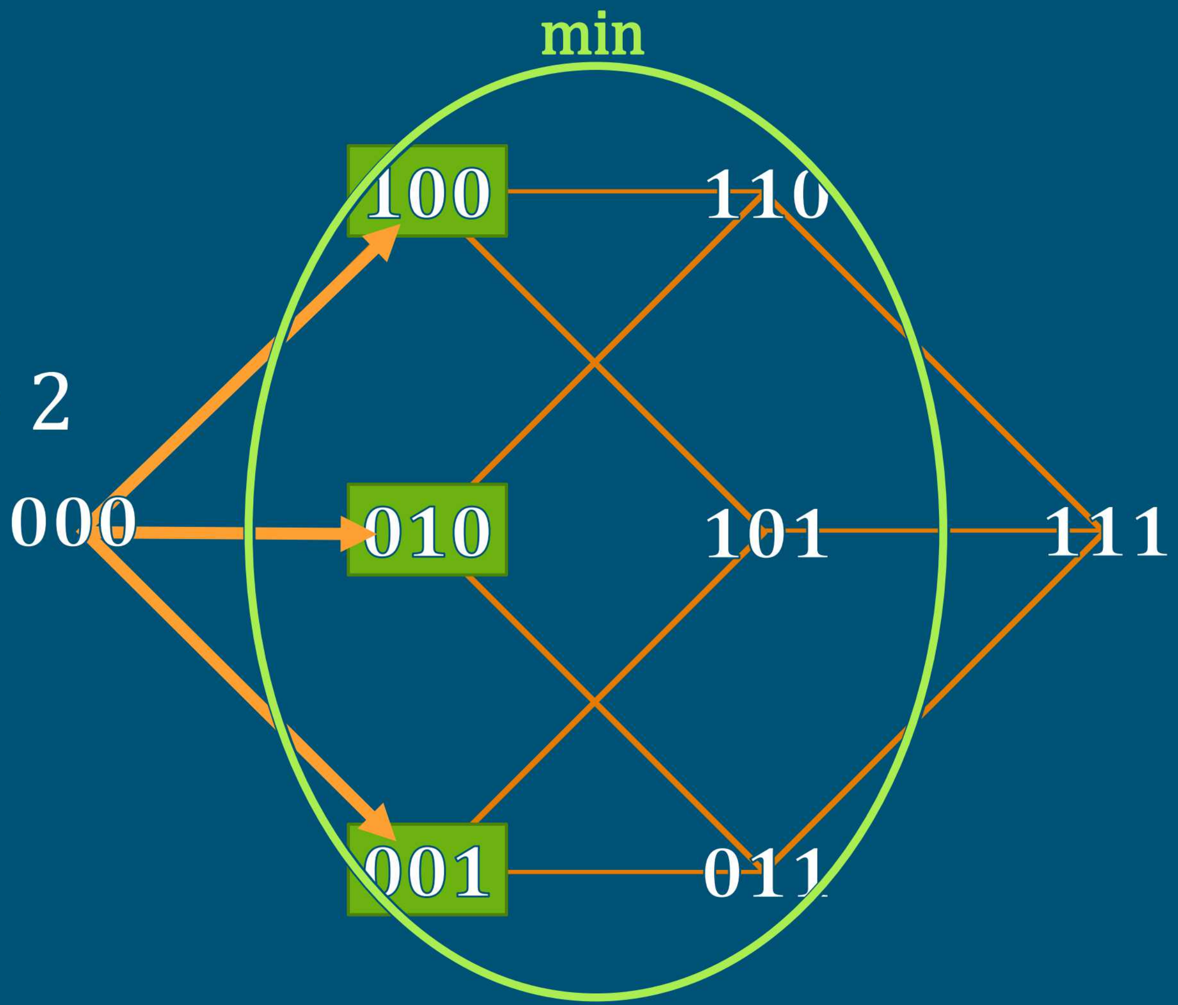




Local Search Heuristic for MAXCUT

- $C(x) = \sum_{(\mu,\nu)\in E}(1 - x_\mu x_\nu)$
- $k = 1$
- $x = 000$
- $C(100) = C(010) = C(001) = 2$

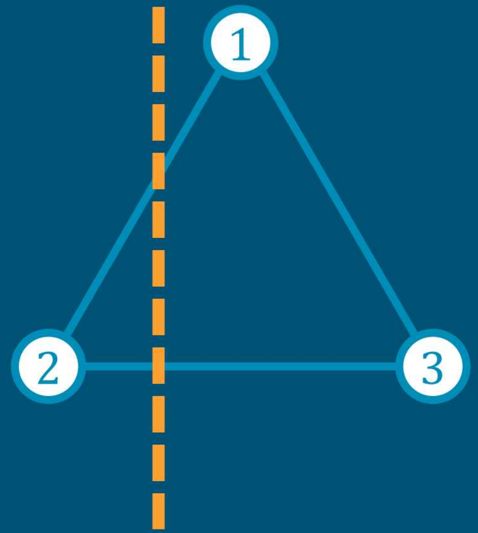


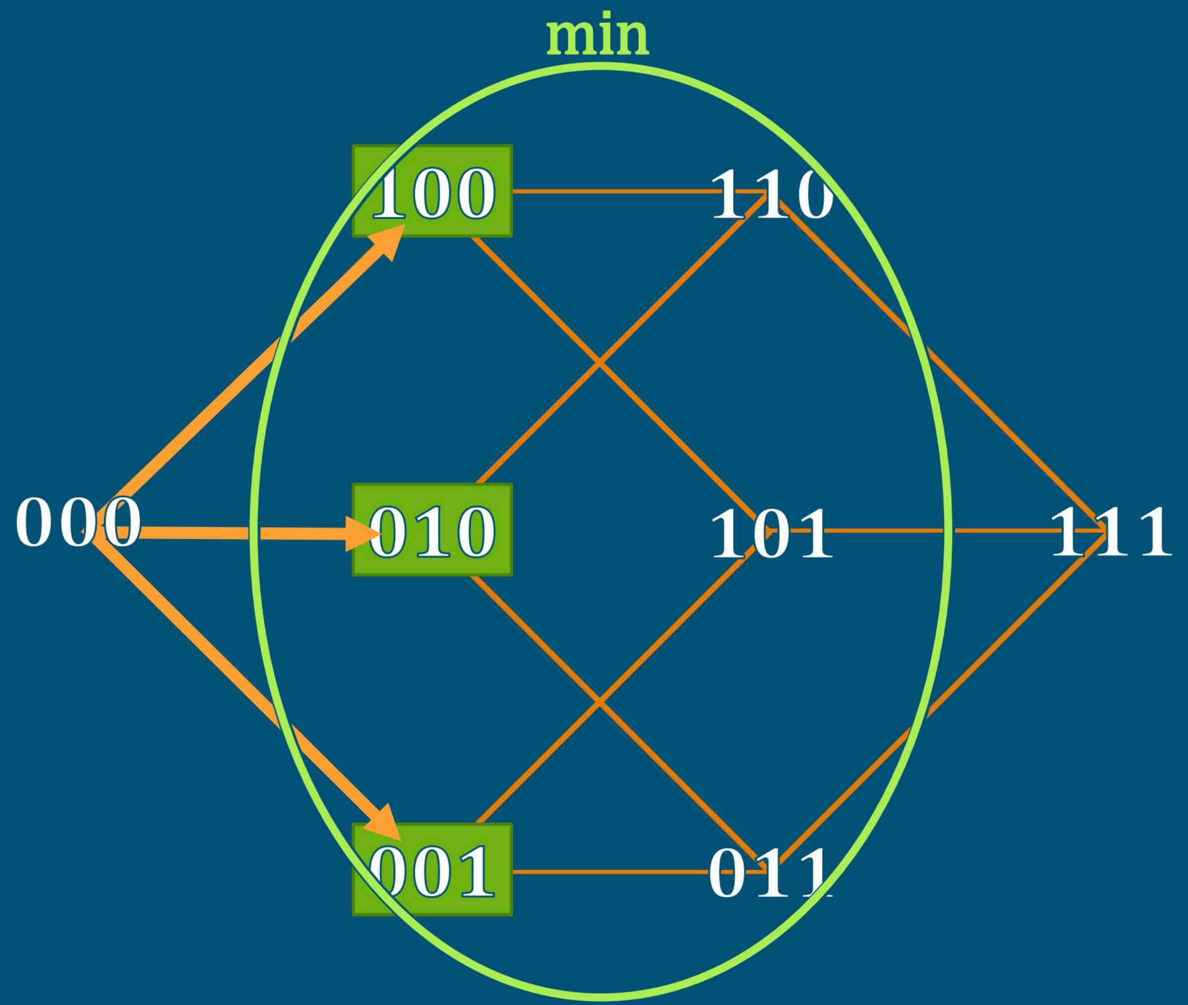


Local Search Heuristic for MAXCUT

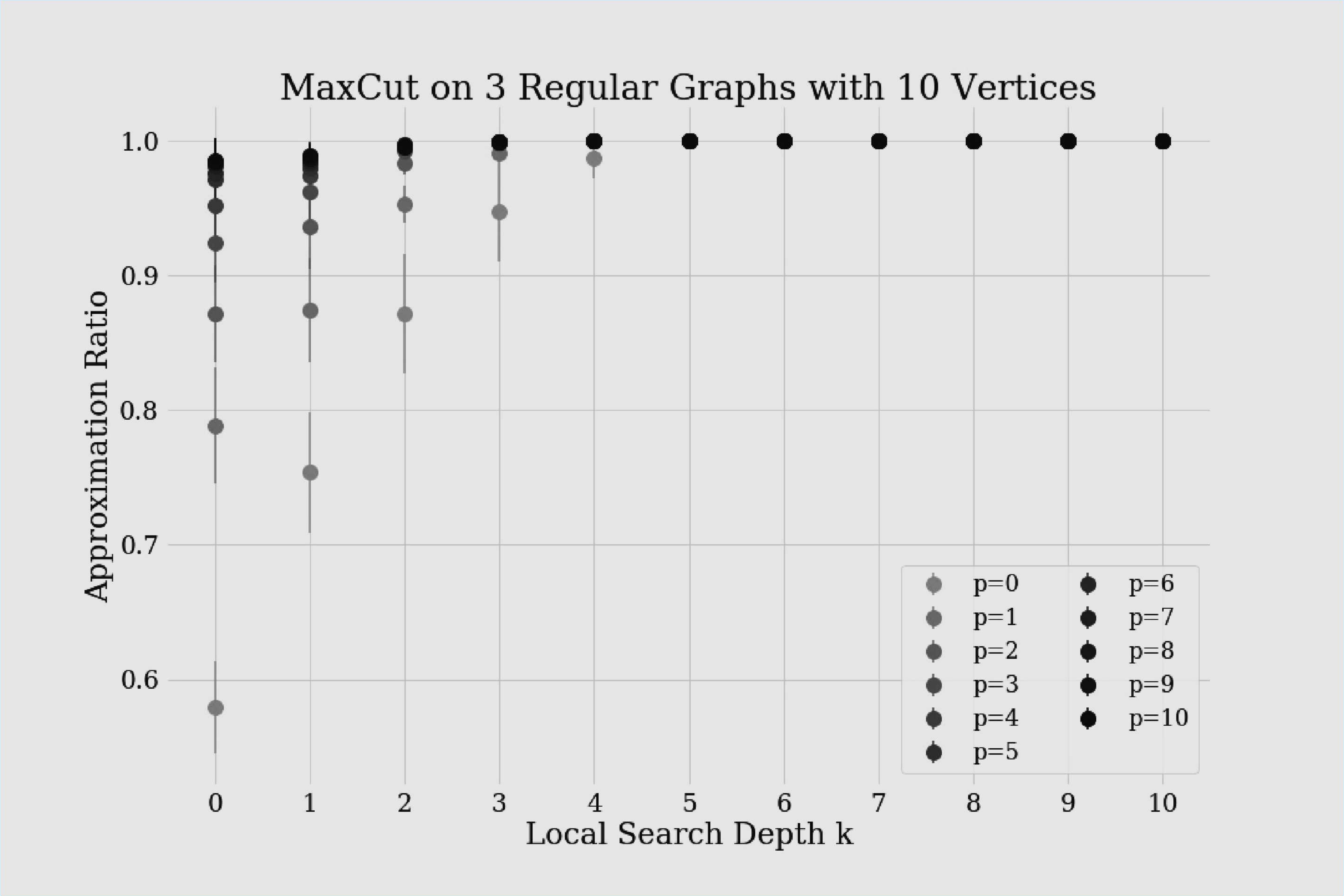
- $C(x) = \sum_{(\mu,\nu)\in E}(1 - x_\mu x_\nu)$
- $k = 1$
- $x = 000$

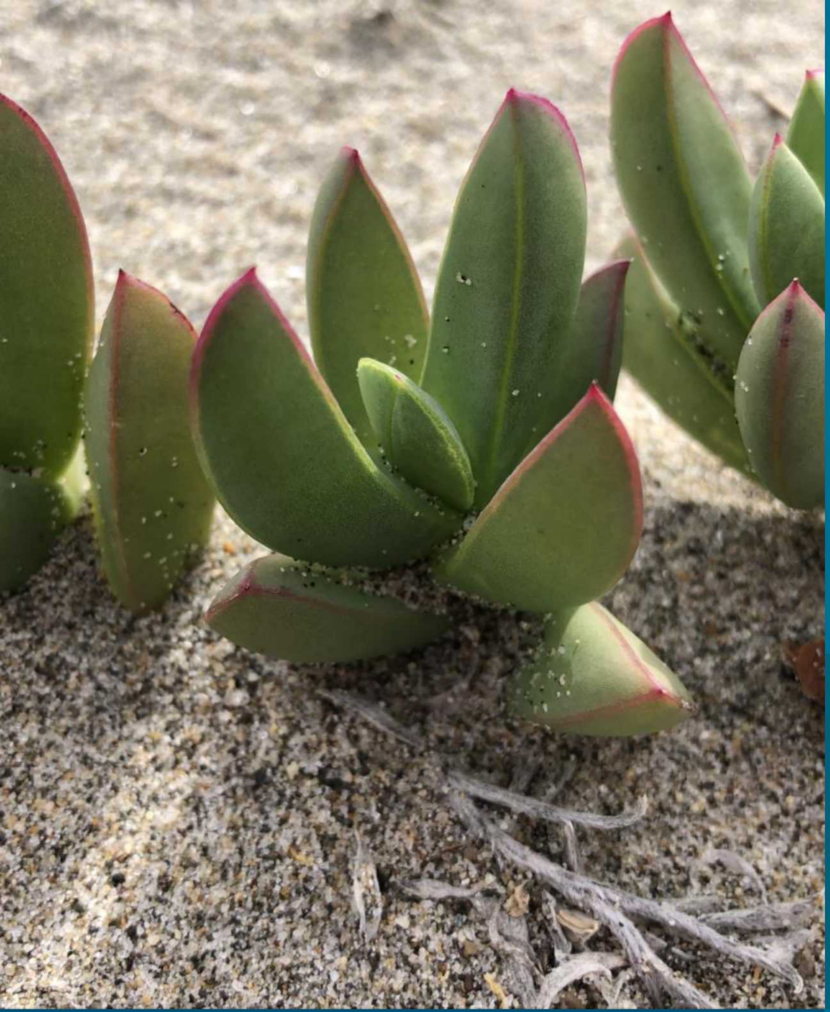
$\Rightarrow \max_{\boldsymbol{x}} \boldsymbol{C}(\boldsymbol{x}) = \mathbf{2}$





Random Regular Graphs

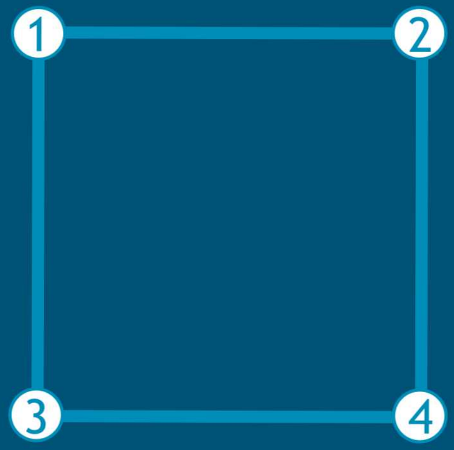




MinVertexCover Example

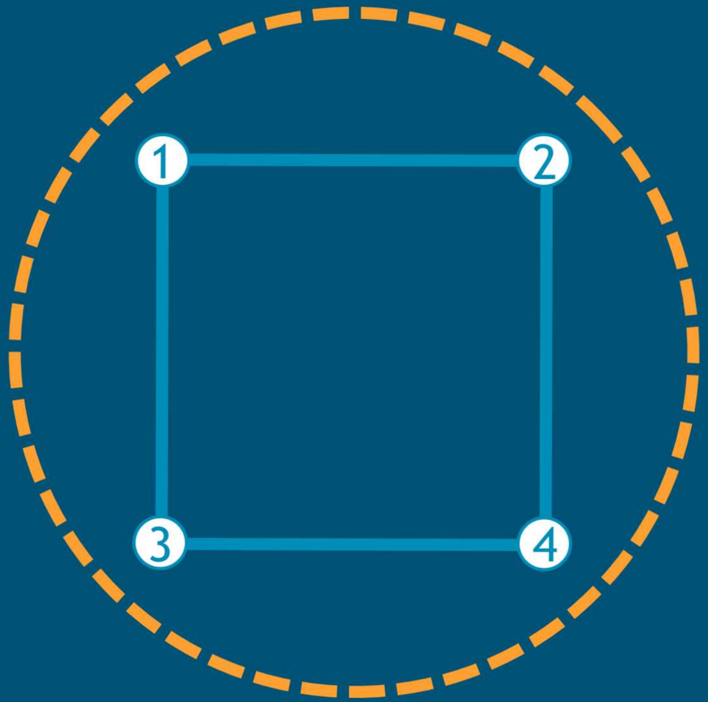
VERTEXCOVER

- Given a graph $G = (V, E)$, we want to find a subset of vertices $V' \subseteq V$, s.t. $\forall$ edges are **covered**.
- An edge, $\{u, v\} \in E$, is **covered** if $u \in V'$ or $v \in V'$.
- Example:



VERTEXCOVER

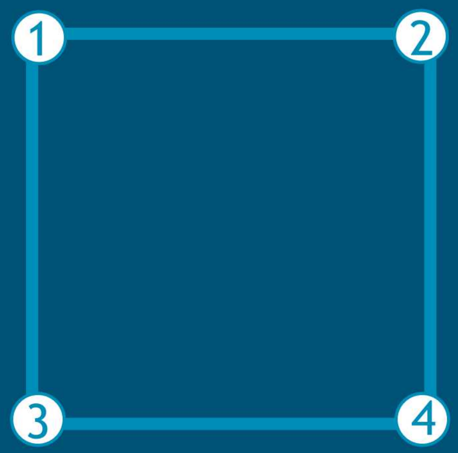
- Given a graph $G = (V, E)$, we want to find a subset of vertices $V' \subseteq V$, s.t. $\forall$ edges are **covered**.
- An edge, $\{u, v\} \in E$, is **covered** if $u \in V'$ or $v \in V'$.
- Example: Trivial VC



MinVertexCover

- Given a graph $G = (V, E)$, we want to find a minimum subset of vertices $V' \subseteq V$, s.t. $\forall$ edges $\{u, v\} \in E$ are covered.

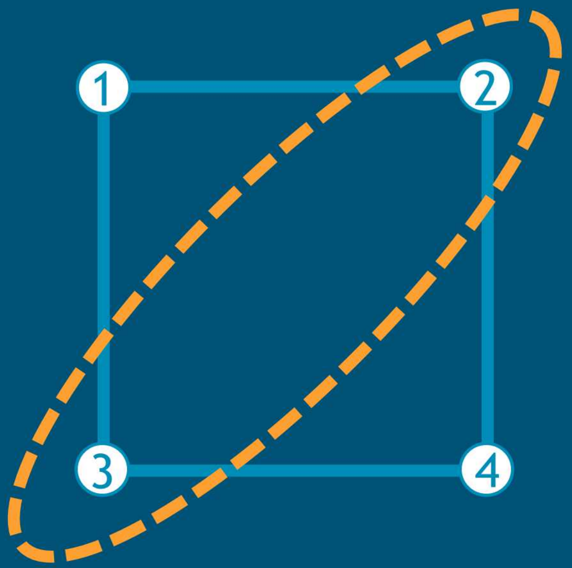
- Example: MinVertexCover = ?



MinVertexCover

- Given a graph $G = (V, E)$, we want to find a minimum subset of vertices $V' \subseteq V$, s.t. $\forall$ edges $\{u, v\} \in E$ are covered.

- Example: MinVertexCover = 2



MINVERTEXCOVER Facts

- Decision problem related to MINVERTEXCOVER is NP-complete. [1]

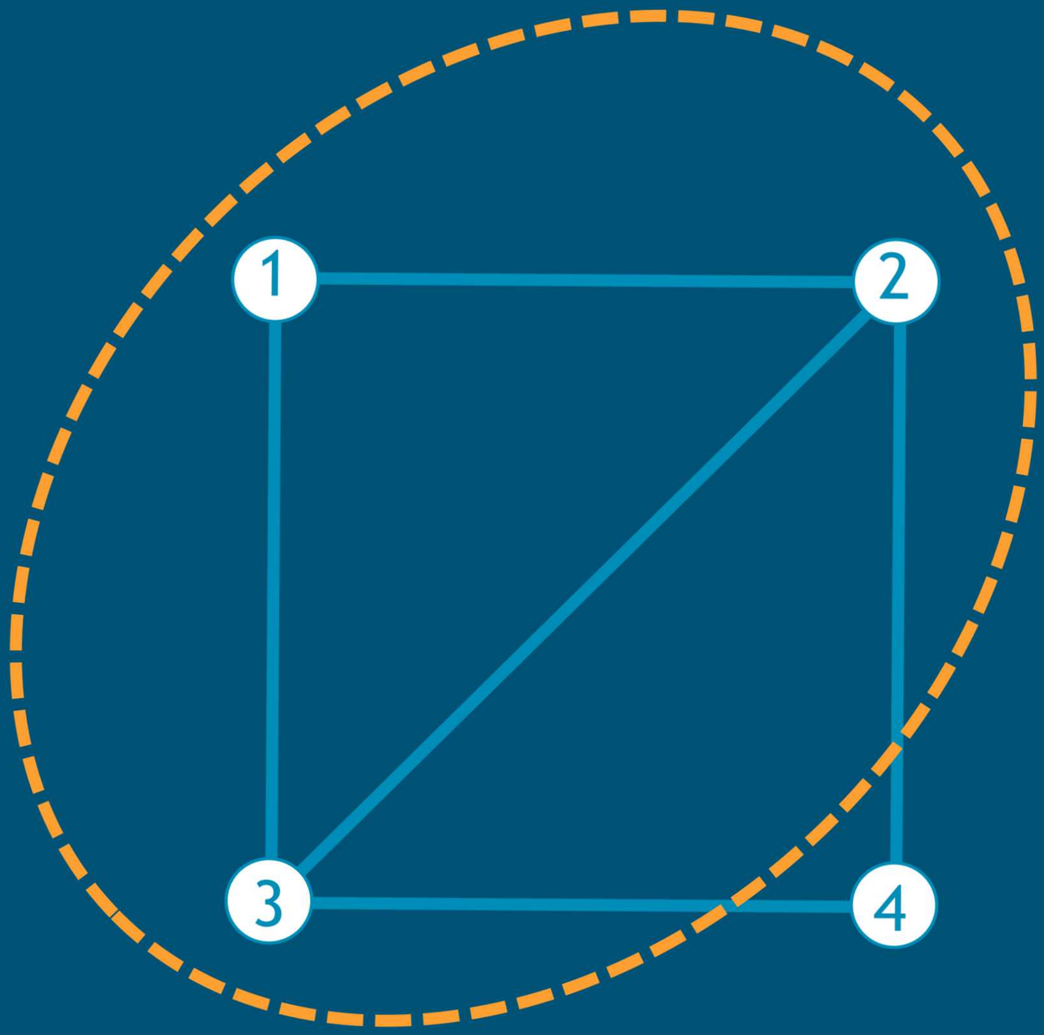
- MINVERTEXCOVER is APX-complete. It can be approximated in polynomial time to within a constant factor. [2]

- MINVERTEXCOVER has a $(2 - O(1/\log n))$–approximation. [3]

- It is NP-Hard to approximate MINVERTEXCOVER with an approximation ratio $\Gamma > 1.3606$. [4]

[1] T. H. Cormen *et. al.* (2009). Gareyand D. S. Johnson, [2] C. Papadimitriou and M. Yannakakis (1991),
[3] G. Karakostas (2009) [4] I. Dinur and S. Safra (2005)

Mapping to Qubits

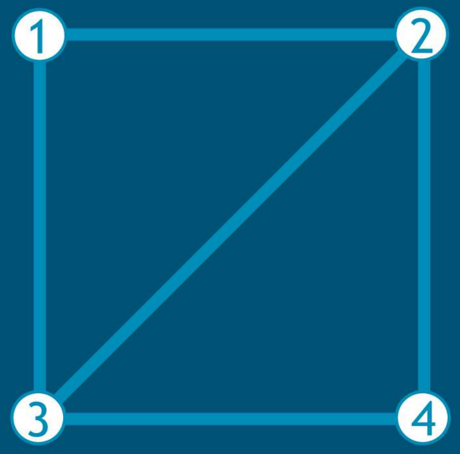


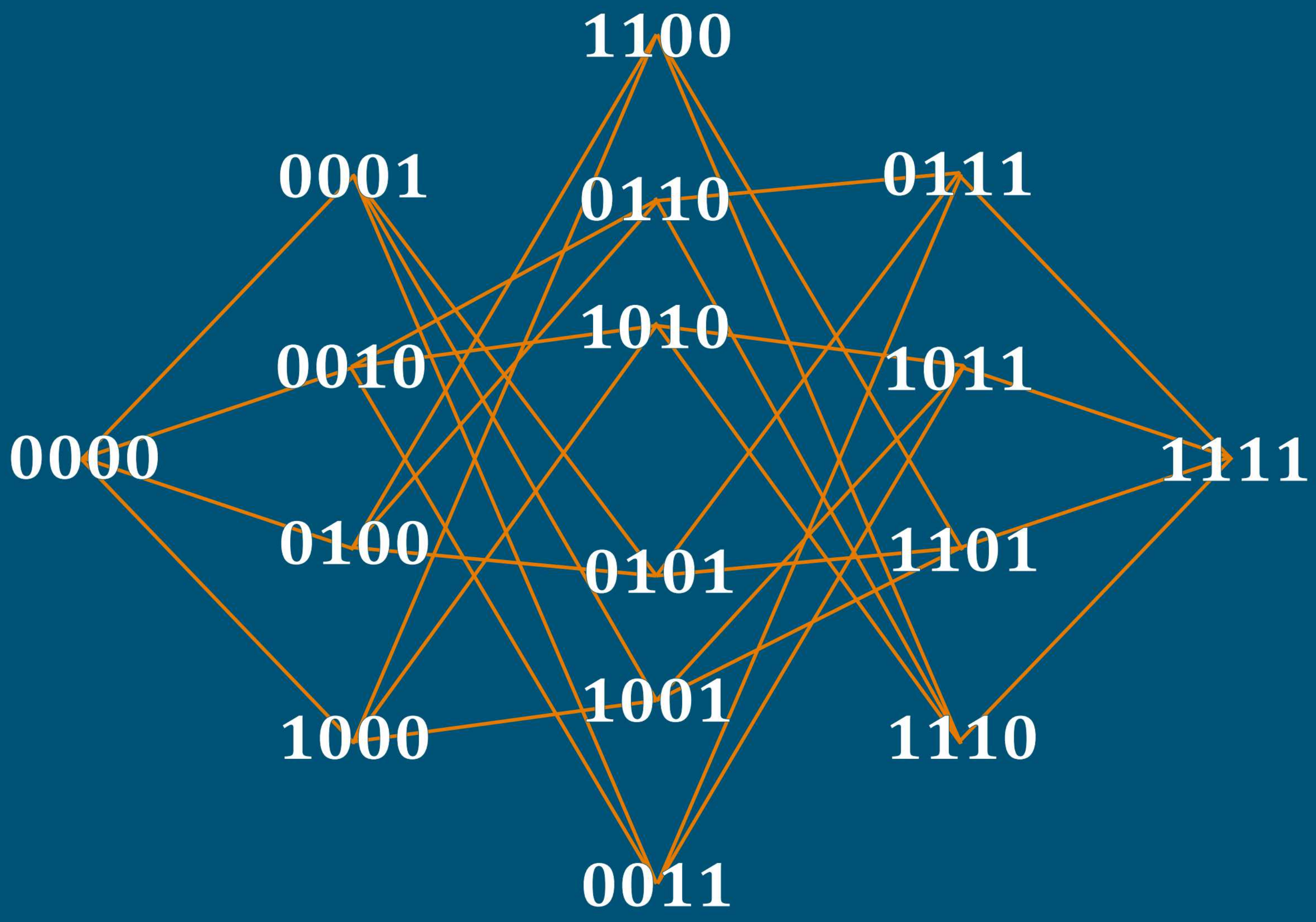
$$|1110\rangle$$

1 2 3 4

Local Search Heuristic for MINVERTEXCOVER

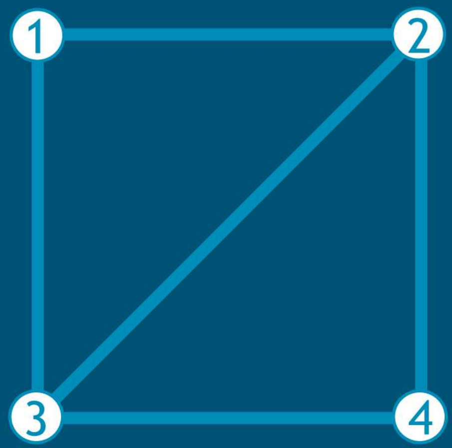
- $C(x) = \sum_{\mu \in V} x_\mu$
- $k = 3$

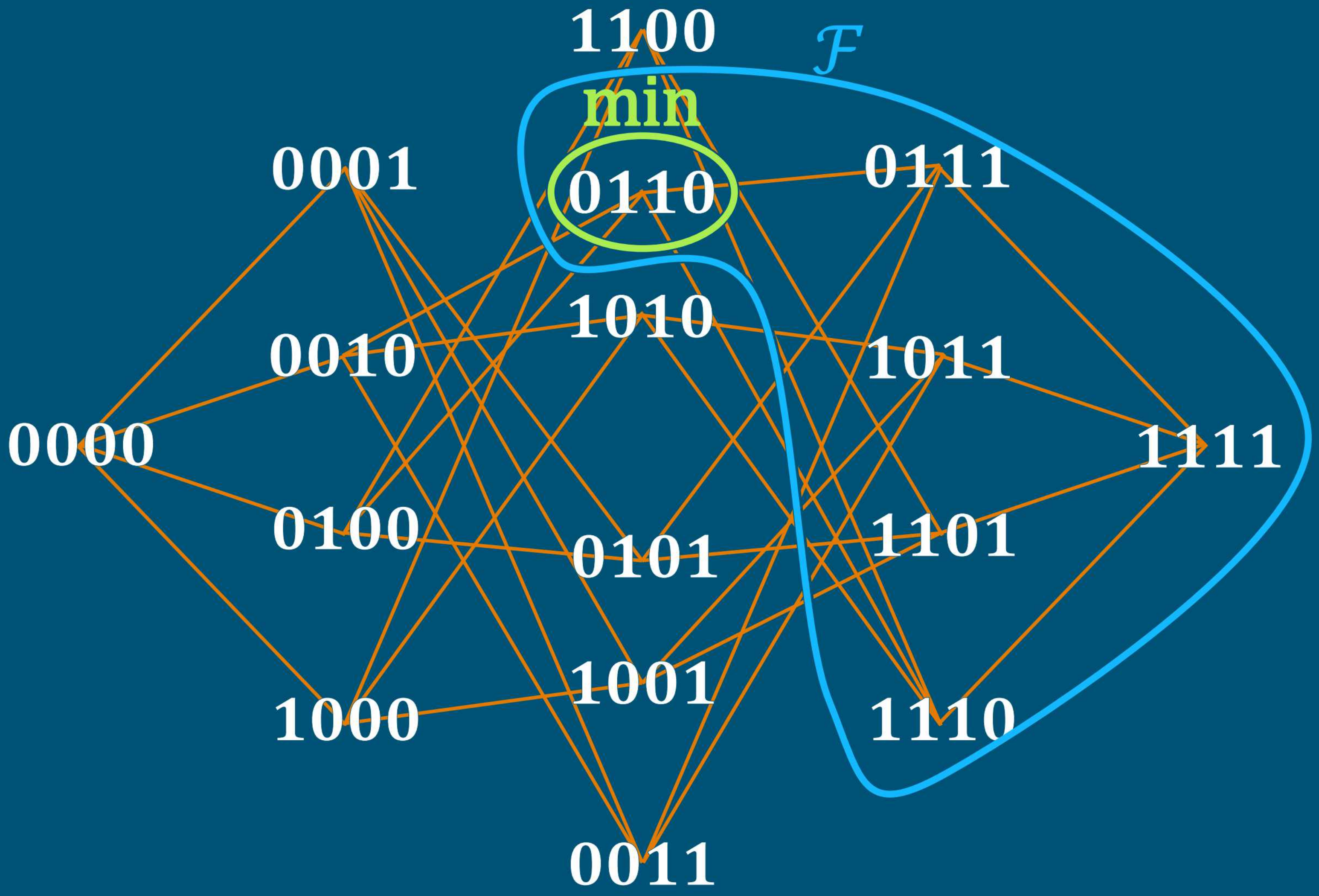




Local Search Heuristic for MinVertexCover

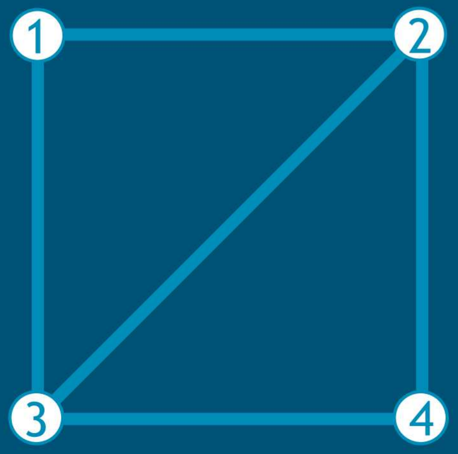
- $C(x) = \sum_{\mu \in V} x_\mu$
- $k = 3$

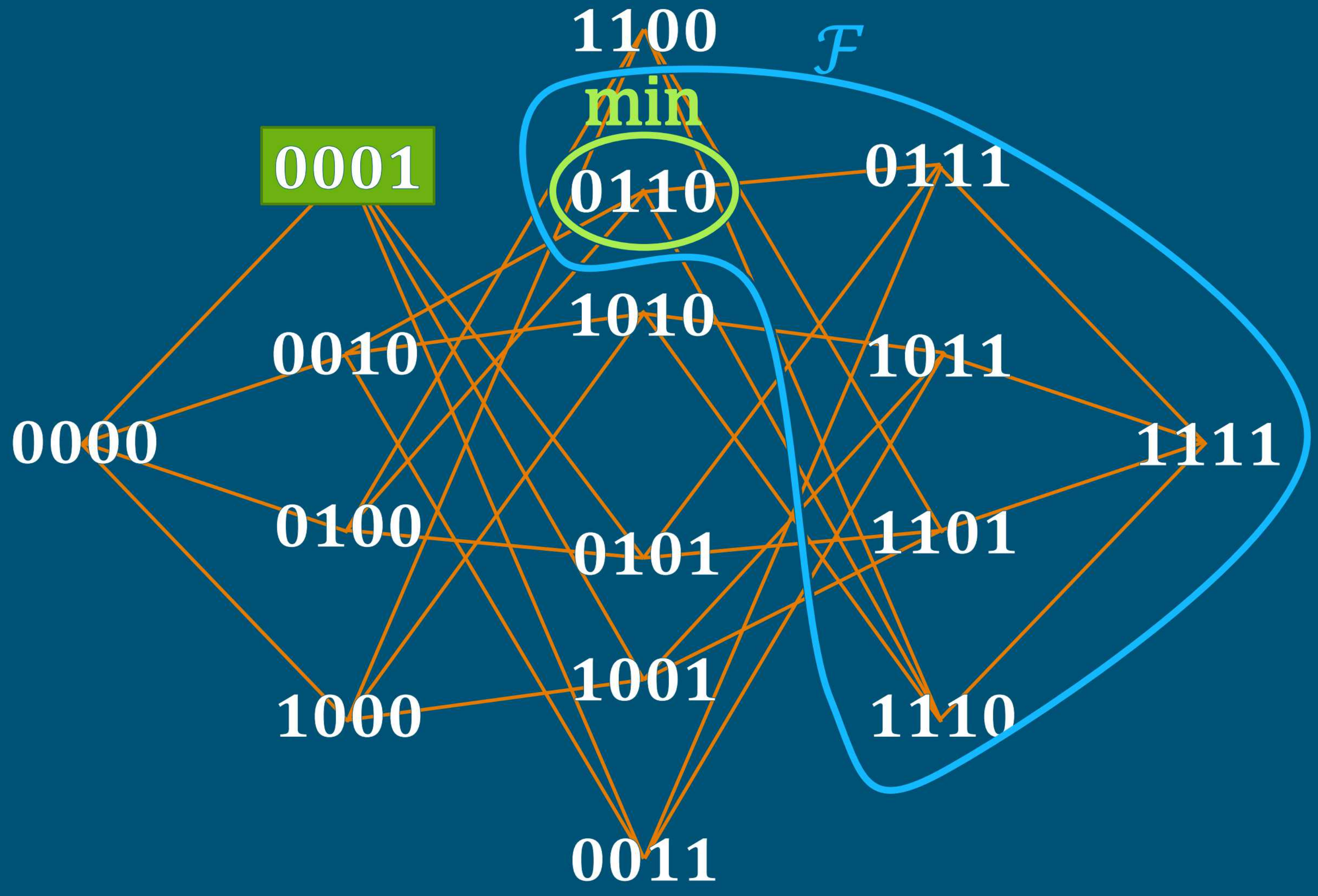




Local Search Heuristic for MINVERTEXCOVER

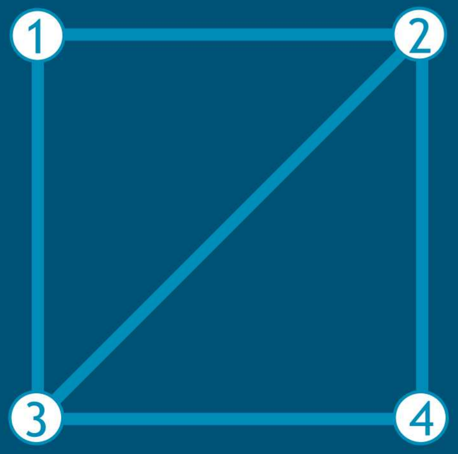
- $C(x) = \sum_{\mu \in V} x_\mu$
- $k = 3$
- $x = 0001$
- $0001 \notin \mathcal{F}$

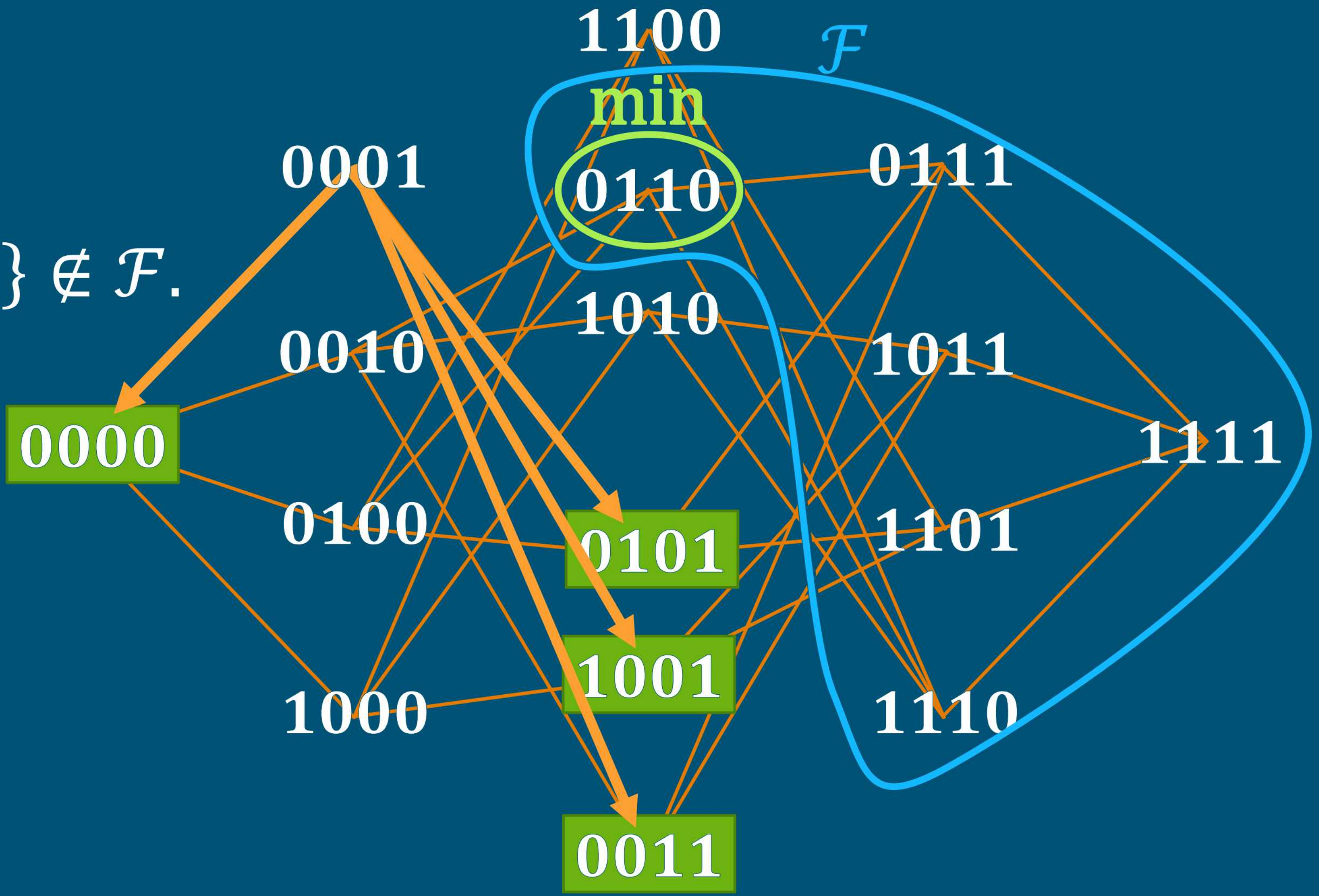




Local Search Heuristic for MINVERTEXCOVER

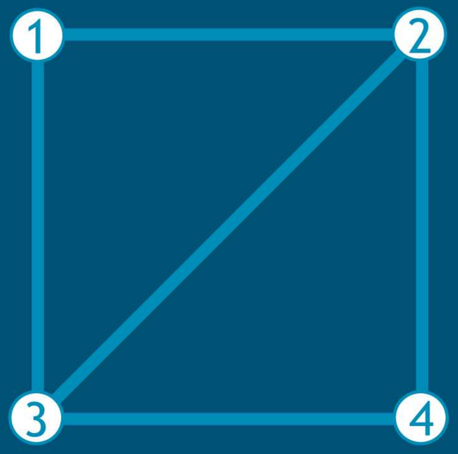
- $C(x) = \sum_{\mu \in V} x_\mu$
- $k = 3$
- $x = 0001$
- $\{0000, 0011, 1001, 0101\} \notin \mathcal{F}$.

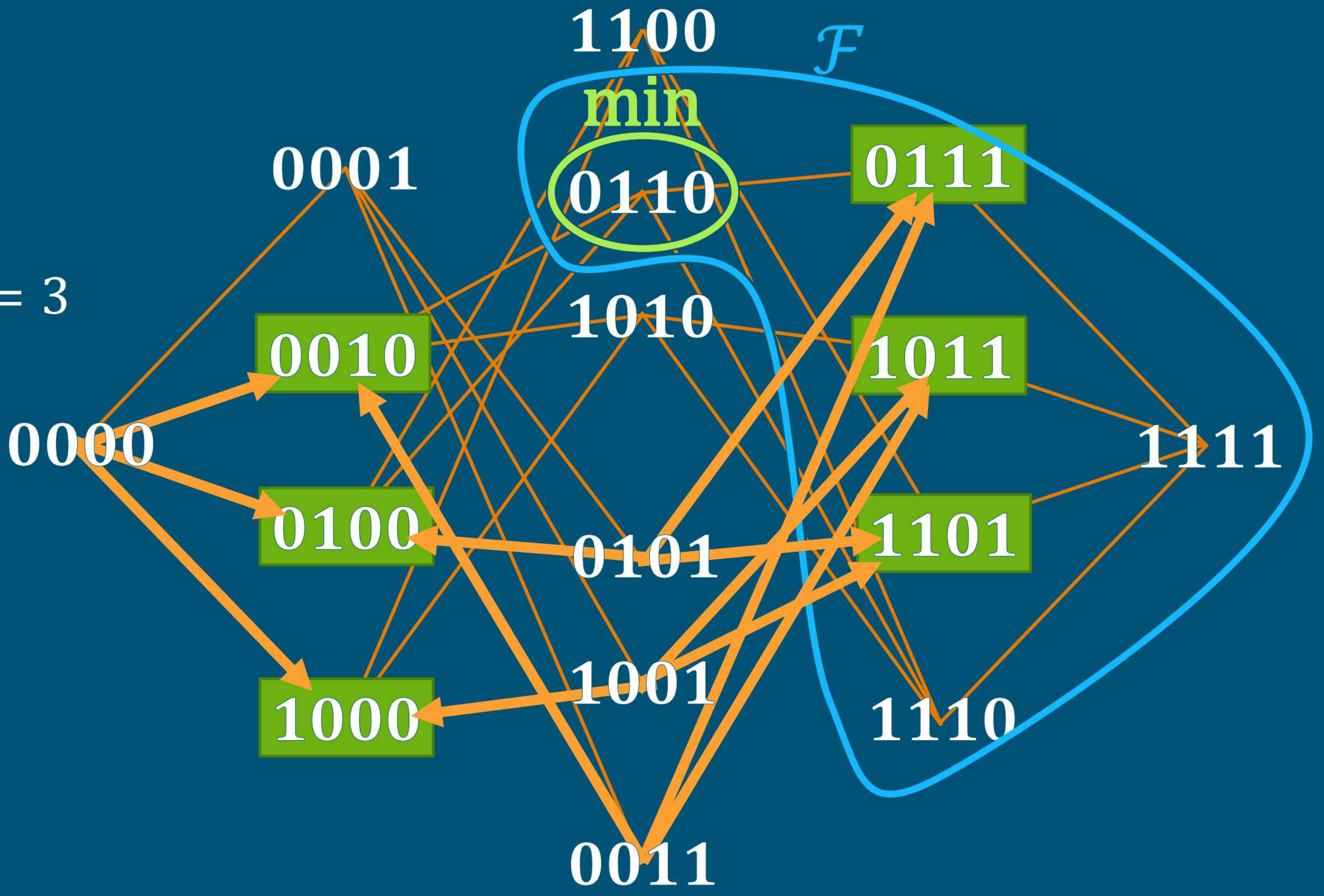




Local Search Heuristic for MINVERTEXCOVER

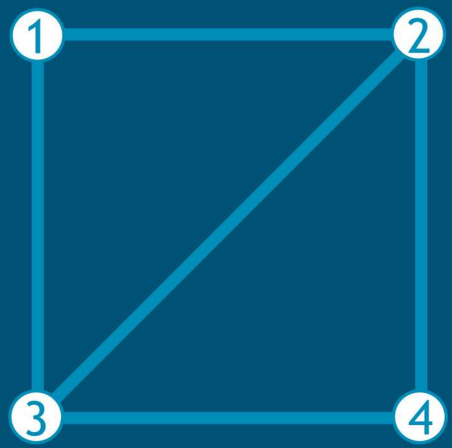
- $C(x) = \sum_{\mu \in V} x_\mu$
- $k = 3$
- $x = 0001$
- $\{0010, 0100, 1000\} \notin \mathcal{F}$
- $C(0111) = C(1011) = C(1101) = 3$

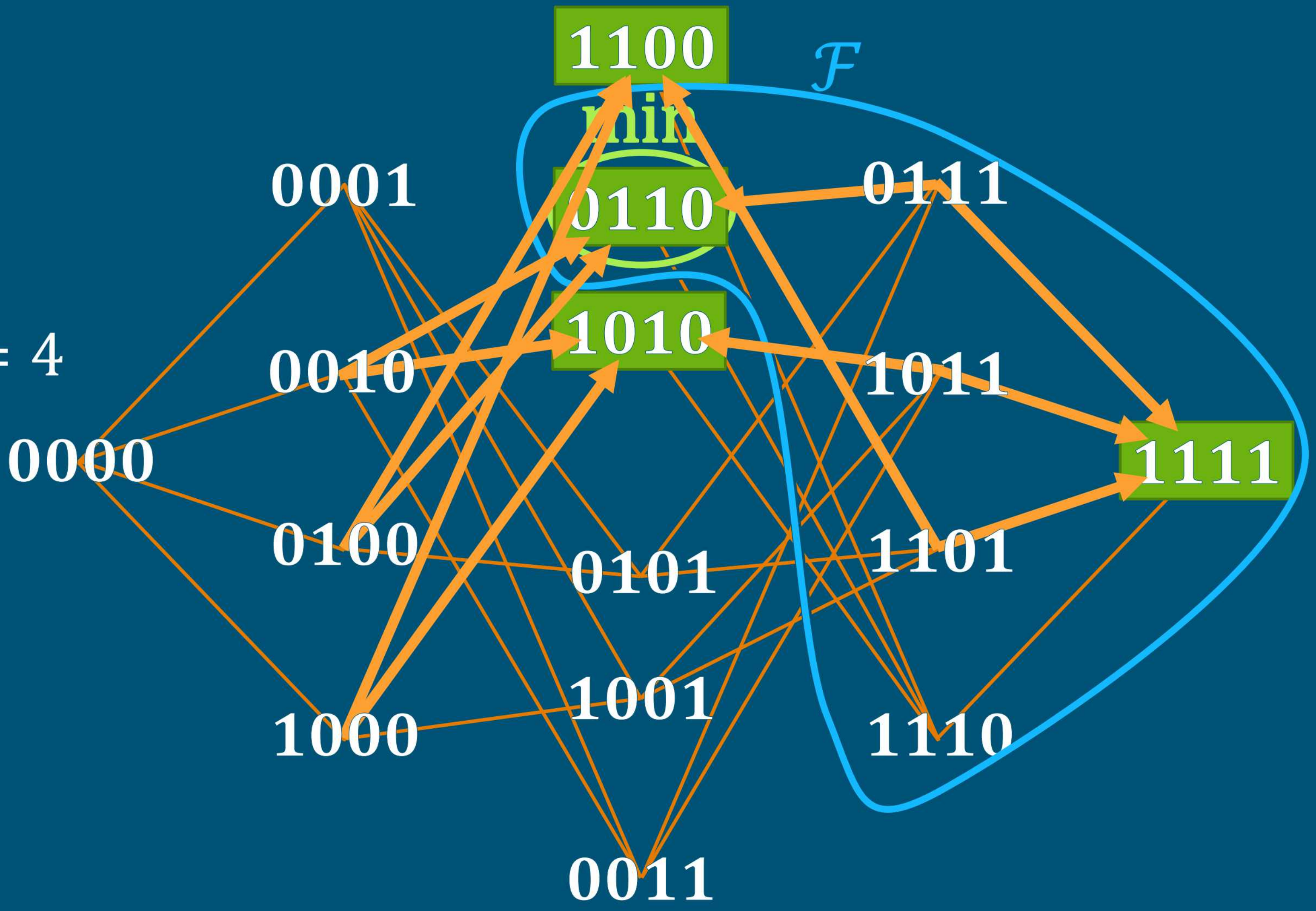




Local Search Heuristic for MINVERTEXCOVER

- $C(x) = \sum_{\mu \in V} x_\mu$
- $k = 3$
- $x = 0001$
- $\{0010, 0100\} \notin \mathcal{F}$
- $C(0110) = 2$ and $C(1111) = 4$

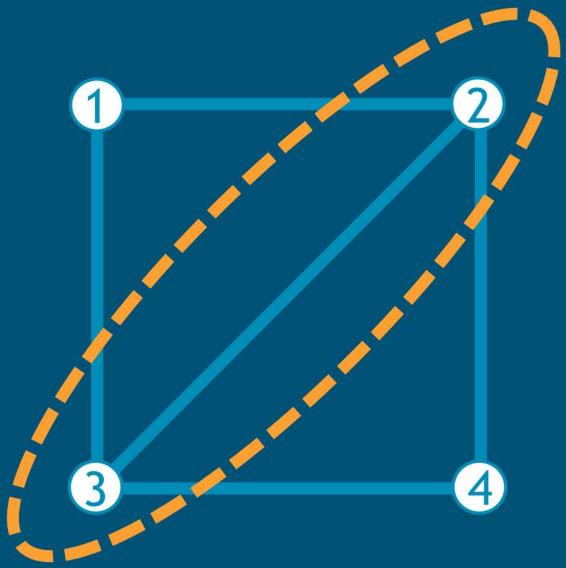


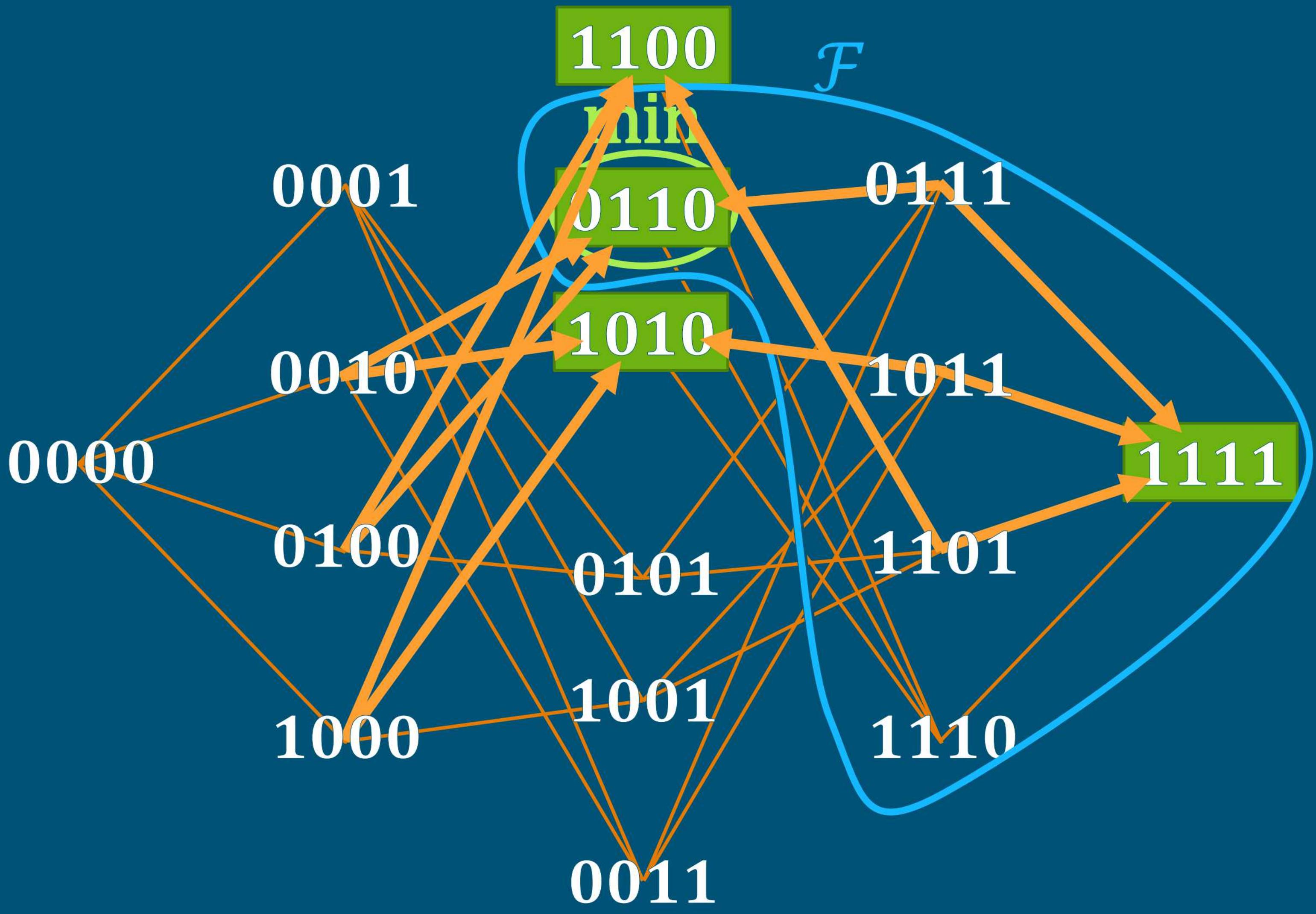


Local Search Heuristic for MINVERTEXCOVER

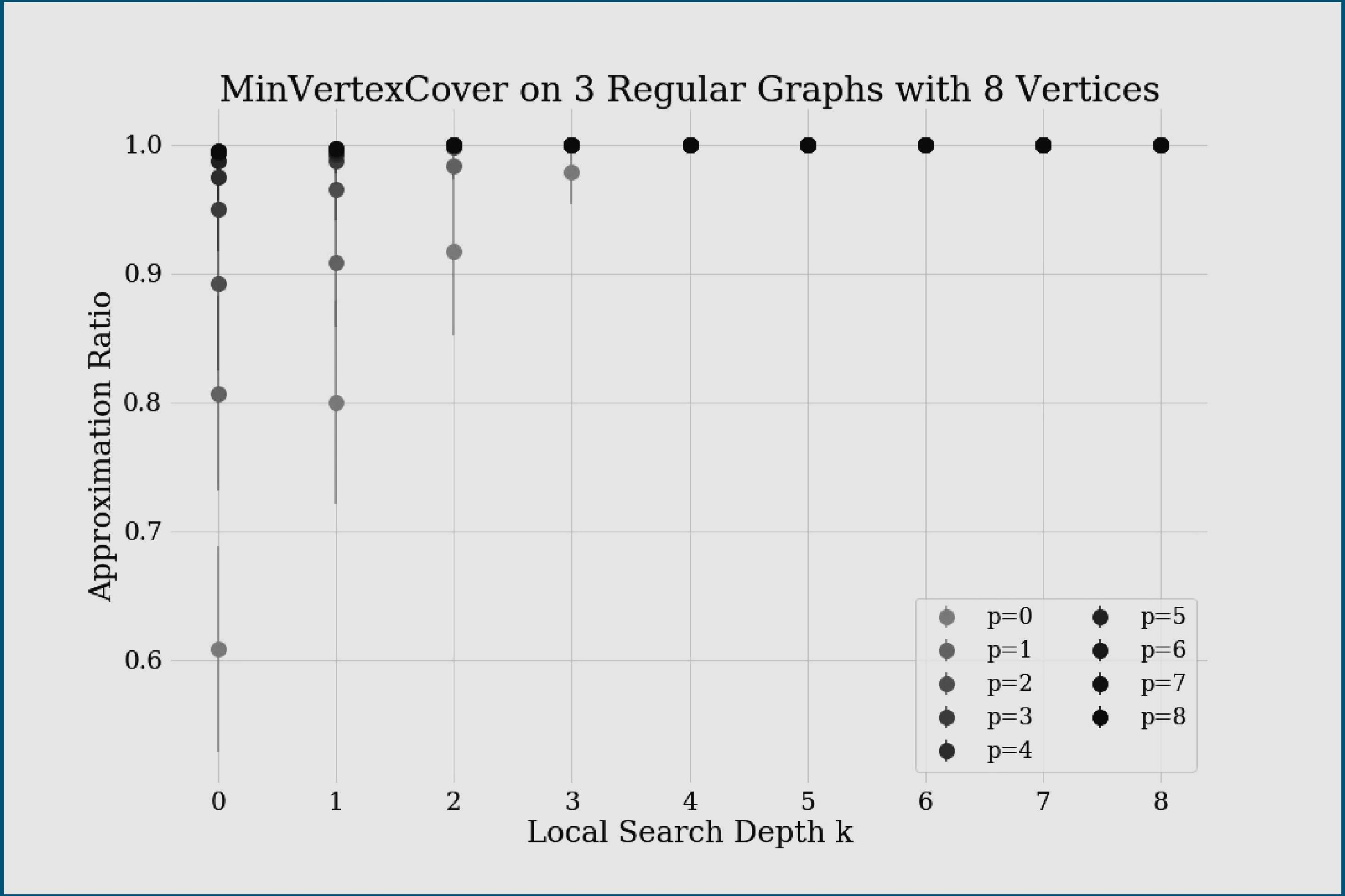
- $C(x) = \sum_{\mu \in V} x_\mu$
- $k = 3$
- $x = 0001$

$$\Rightarrow \min_{x \in \mathcal{F}} C(x) = 2$$





Random Regular Graphs



Summary

Conclusions

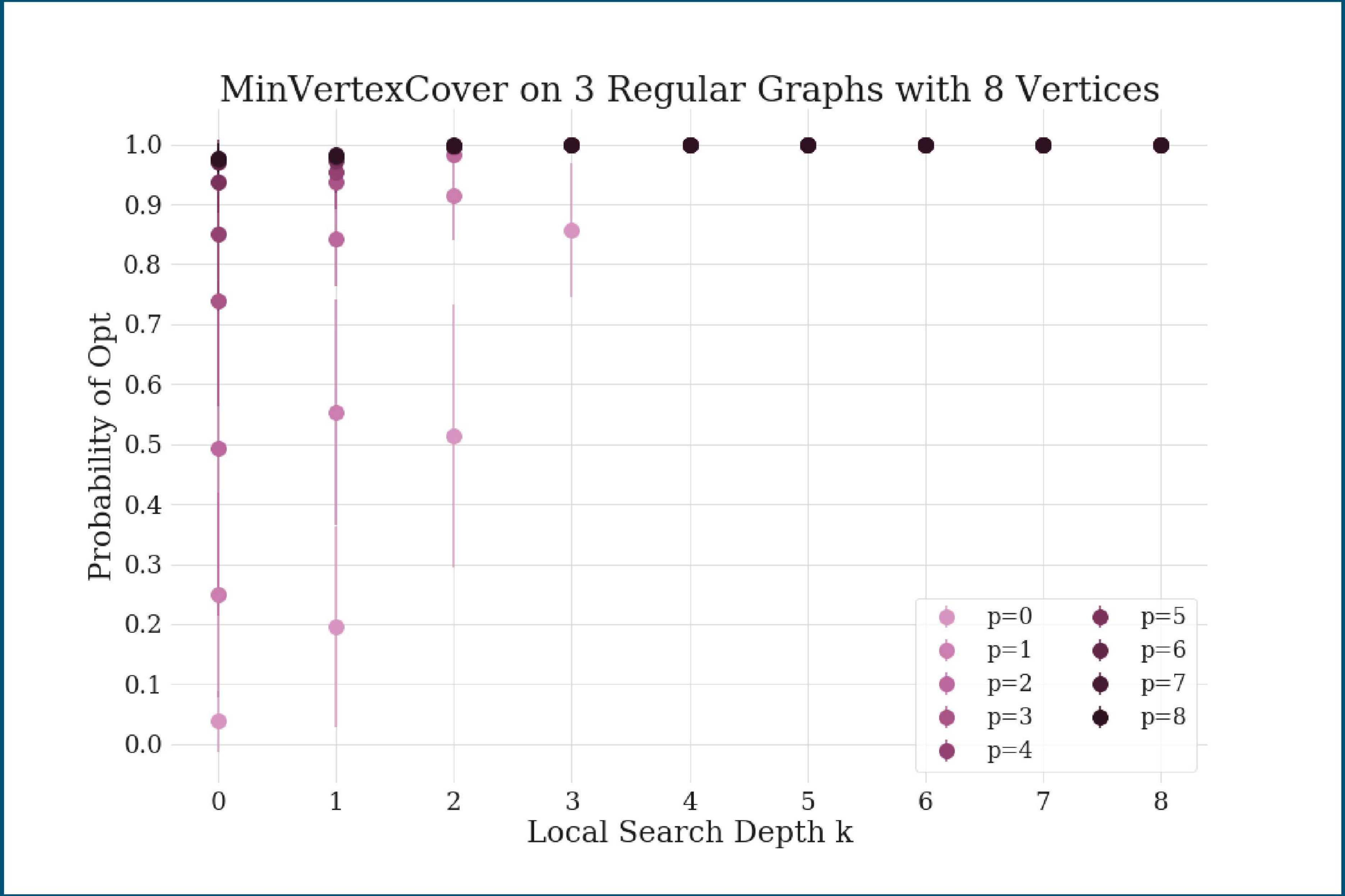
- The approximation ratio can be improved by increasing the local search depth or the QAOA circuit depth.
- Combining QAOA with classical heuristics allows us to get a good approximation ratio with a shorter circuit depth.
- This allows QAOA to be ran on NISQ devices.

Future Work

- Look at how our hybrid QAOA with local search performs on TSP (Traveling Sales Person).
- Analyze QAOA's performance with other classical heuristics.
- Run our hybrid QAOA on Rigetti.


Thanks!

Random Regular Graphs



Random Regular Graphs