Investigating the use of the discrete quantum walk to solve combinatorial optimization problems

JIQ 2026

Rachel Roux[1], Edouard Debry[1], Davide Boschetto[2], Hamza Jeffali[3], Henri De Boutray[3]

[1]MBDA, [2]ENSTA, [3]ColibriTD

January 19, 2026

MBDA

Introduction

We tackle optimization problems by developping quantum algorithm for the Fault-Tolerant Quantum Computing (FTQC) era.

We aim for at least **quadratic speedup** to offset the absorption of the acceleration by error correction.

We present current investigation on using the discrete quantum random walk (MNRS version) to address our particular optimization problems.

1st problem: Multi-objective Path finding

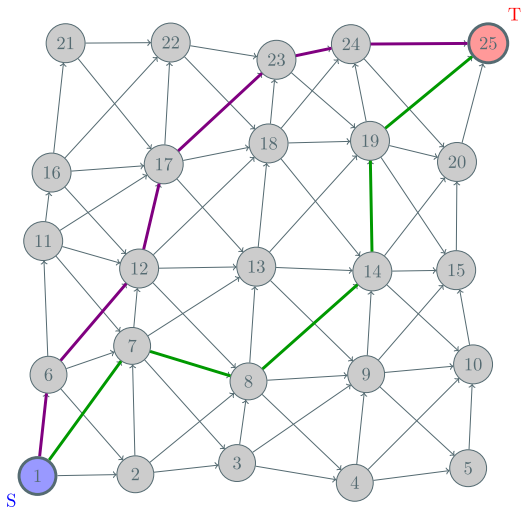


Figure: Two paths respecting several requirement

$$\min_{\{x_{ij}^k\}\in\{0,1\}^{|E|\times n}} \sum_{(ij)\in E} a_{ij}(x_{ij}^1 + \cdots + x_{ij}^n) \tag{1}$$

$$\min_{\{x_{ij}^k\}\in\{0,1\}^{|E|\times n}} \sum_{(i,j)\in E} \sum_{k\neq l} c_{ij}^{kl} x_{ij}^k x_{ij}^l \tag{2}$$

$$\min_{\{x_{ij}^k\}\in\{0,1\}^{|E|\times n}} \frac{1}{n}\sum_k (t_k)^2 - \left(\frac{1}{n}\sum_k t_k\right)^2, \quad t_k = \sum_{(i,j)\in E} t_{ij} x_{ij}^k \tag{3}$$

- $x_{ij}^k \in \{0,1\}$ is the part of the k-th path from i to j
- The formula (1) is the minimisation of risk for all paths
- The formula (2) permits to minimize the number of path going through the same direction
- The formula (3) minimises the travel time variance : so as to ensure isotiming

1st problem: Multi-objective Path finding

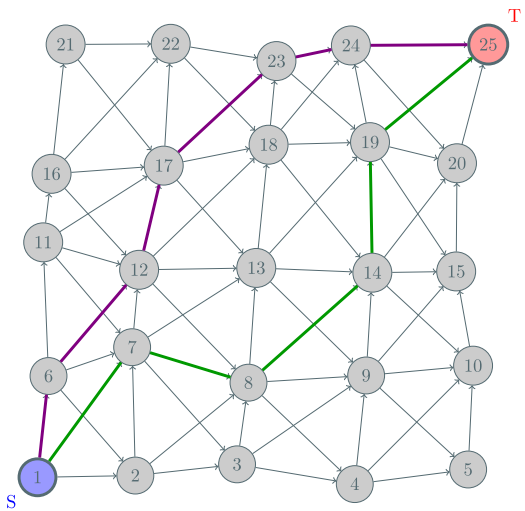


Figure: Two paths respecting several requirement

$$\min_{\{x_{ij}^k\}\in\{0,1\}^{|E|\times n}} \sum_{(ij)\in E} a_{ij}(x_{ij}^1 + \cdots + x_{ij}^n) \tag{1}$$

$$\min_{\{x_{ij}^k\}\in\{0,1\}^{|E|\times n}} \sum_{(i,j)\in E}\sum_{k\neq l} c_{ij}^{kl} x_{ij}^k x_{ij}^l \tag{2}$$

$$\min_{\{x_{ij}^k\}\in\{0,1\}^{|E|\times n}} \frac{1}{n}\sum_k (t_k)^2 - \left(\frac{1}{n}\sum_k t_k\right)^2, \quad t_k = \sum_{(i,j)\in E} t_{ij} x_{ij}^k \tag{3}$$

subject to :

$$\forall\, k \in [\![1,n]\!], \forall i \in \mathcal{N}, \sum_{(i,j)\in E} x_{ij}^k - \sum_{(j,i)\in E} x_{ji}^k = \begin{cases} 1 & \text{, if } i = s \\ -1 & \text{, if } i = t \\ 0 & \text{, if } i \notin \{s,t\} \end{cases} \tag{4}$$

$$\forall\, k \in [\![1,n]\!], \quad \sum_{(ij)\in E} f_{ij} x_{ij}^k \leq f_{\max} \tag{5}$$

$$\forall\, k \in [\![1,n]\!], \quad \sum_{(ij)\in E} \sigma_{ij} x_{ij}^k \leq \sigma_{\max} \tag{6}$$

Assignment problem

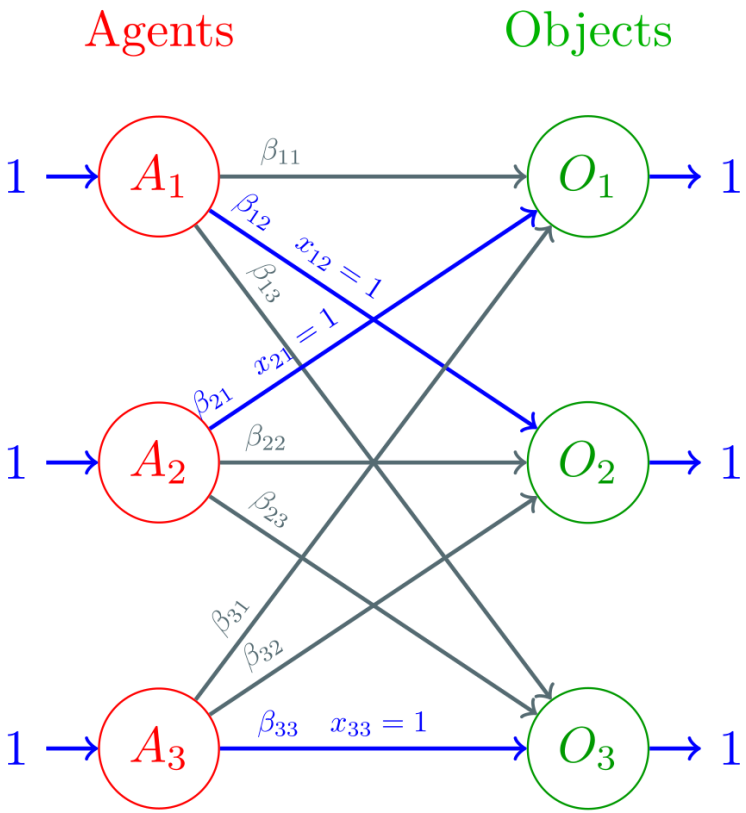


Figure: Symmetric assignment

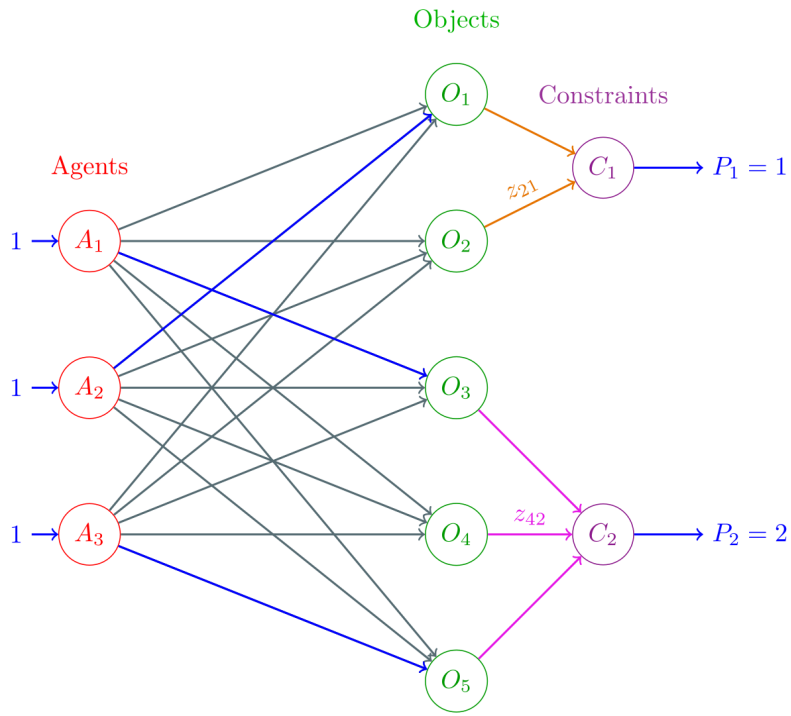
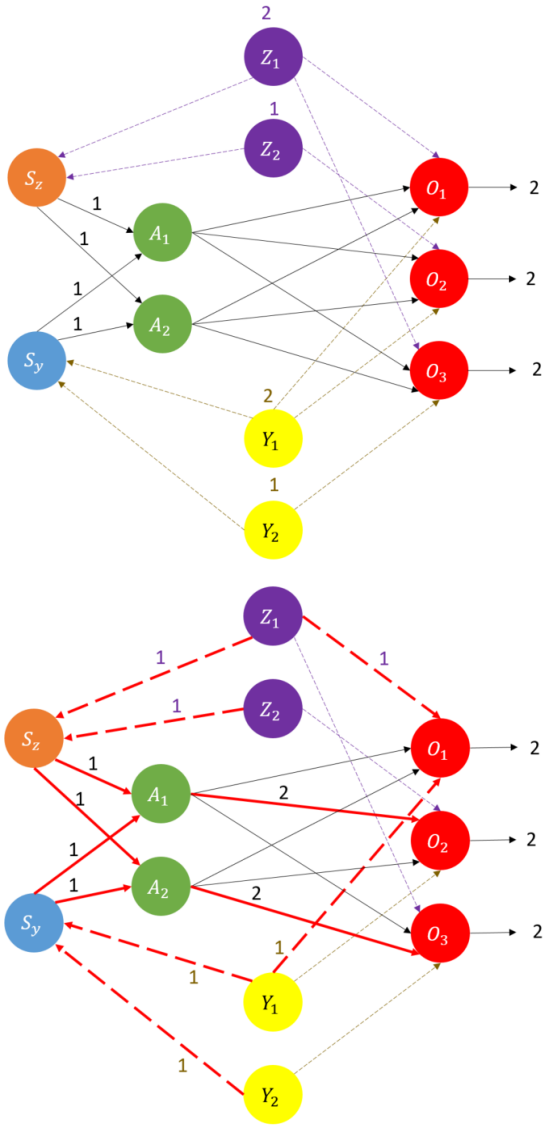


Figure: Asymmetric assignment with M agents and N objects and 2 constraints

2nd problem: Optimal assignment



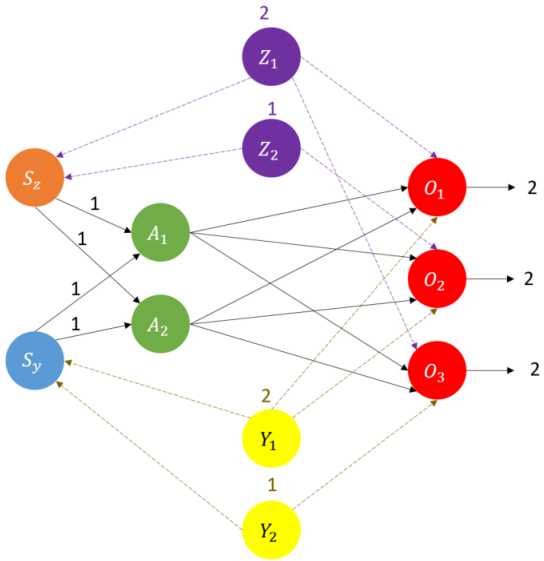
$$\max_{\{x_{ij}\}\in\{0,1\}^{M\times N}} \sum_{(ij)\in E} \beta_{ij}(x_{ij}) - \sum_{(ij)\in E^2} c_{ijkl}(x_{ij}x_{kl}) \tag{7}$$

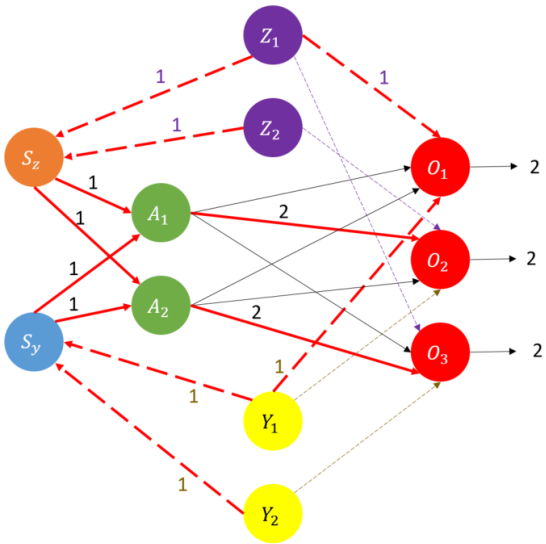
$$\forall\, i \in [\![1, M]\!], \quad \sum_{j=1}^{N} x_{ij} = M_i \tag{8}$$

$$\forall\, j \in [\![1, N]\!], \quad \sum_{i=1}^{N} x_{ij} = 1 - y_{k_j j} = 1 - z_{l_j j} \tag{9}$$

- $x_{ij} \in 0, 1$ is the assignment between an agent A to an objet O
- The objective function is a maximisation from benefit assigment minus conflict between two assigment
- The formula (8) express that the agent i has M_i assignment into it.
- The formula (9) express that objet can be assign to at most 1 agent

Optimal assignment





$$\max_{\{x_{ij}\}\in\{0,1\}^{M\times N}} \sum_{(ij)\in E} \beta_{ij}(x_{ij}) - \sum_{(ij)\in E^2} c_{ijkl}(x_{ij}x_{kl}) \tag{7}$$

$$\forall\, i \in [\![1, M]\!], \quad \sum_{j=1}^{N} x_{ij} = M_i \tag{8}$$

$$\forall\, j \in [\![1, N]\!], \quad \sum_{i=1}^{N} x_{ij} = 1 - y_{k_j j} = 1 - z_{l_j j} \tag{9}$$

$$\forall\, k \in K \quad Y_k = \sum_{j\in\mathcal{J}_k} y_{kj} + y_{ks}, \qquad y_{ks} \in \{0, 1, \ldots, P_k^y\} \tag{10}$$

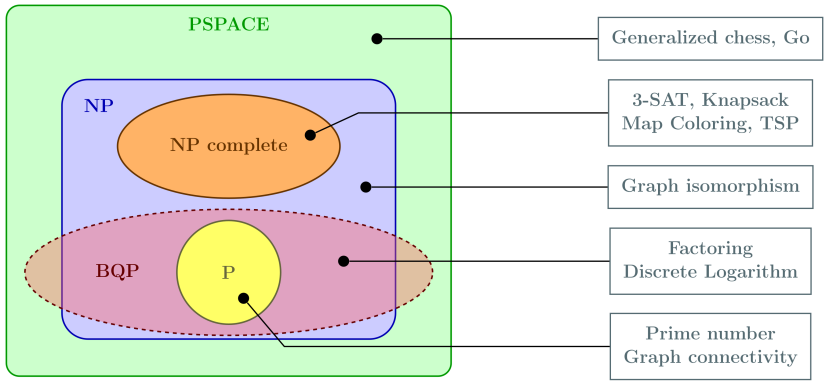
$$\forall\, l \in L \quad Z_l = \sum_{j\in\mathcal{J}_k} z_{lj} + z_{ls}, \qquad z_{ls} \in \{0, 1, \ldots, P_l^z\} \tag{11}$$

$$\sum_{k\in K} y_{ks} = N \tag{12}$$

$$\sum_{l\in L} z_{ls} = N \tag{13}$$

Complexity

The complexity of these problems lies in their **NP-Completeness**, stemming from the multi-objective pathfinding and from the multiple assignment conflicts.



Quantum Walks take advantage of our problems' graph structures.

While existing implementations of quantum walks for solving optimization problems have focused on continuous-time quantum walks, we investigate the potential of **discrete-time quantum walks**, addressing several key challenges:

- The modeling of the coin operator.
- The efficient encoding of the graph structure in a quantum register.

[0] "U. Nzongani et al, Sampled-Based Guided Quantum Walk, 2025"

Random Walk as a Markov Chain

Markov Chain

Sequence of events

Process with no memory

p_{ij}: transition probability from i to j

P : Transition probability matrix

Sum of all row's elements $= 1$

$p_i^T P = p_{i+1}$

Aperiodic and Irreducible

Random walk

Undirected graph search

From one node to its neighbors

$p_n(i)$: probability to be at i at time n

A : Adjacency matrix

Sum of all column's elements $= 1$

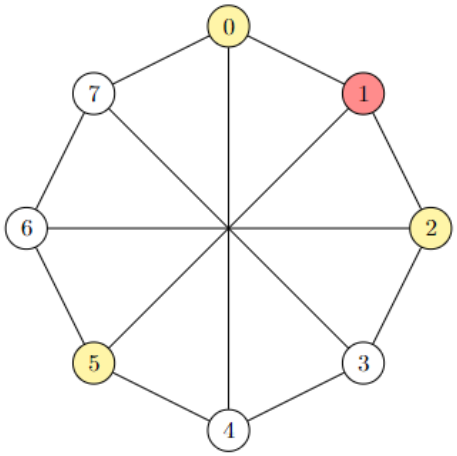
$Ap_i = p_{i+1}$

Connected and Irreducible

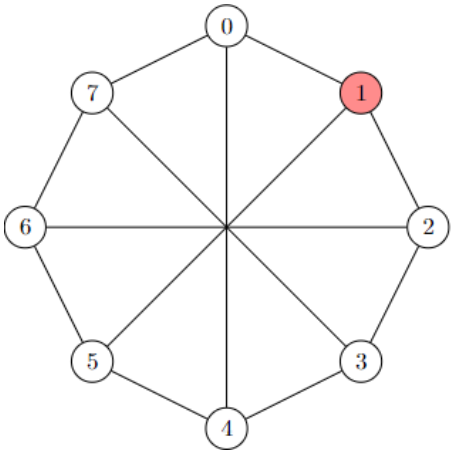
Random Walk Transition Matrix

Consider a graph G where the initial state, denoted as p_0, is associated with the red node.
A the adjacency matrix of graph G.

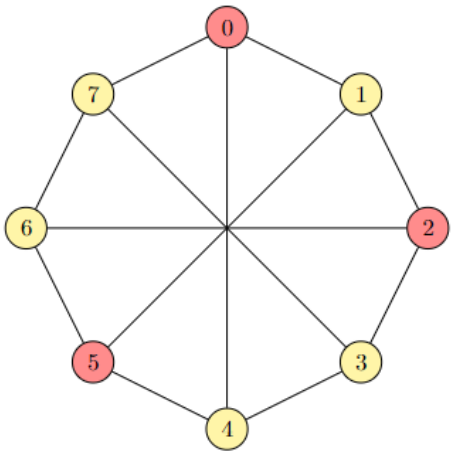
$$A = \frac{1}{3}\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$



$$p_1 = Ap_0 = \frac{1}{3}(10100100)^T$$



$$p_0 = (01000000)^T$$



$$p_2 = Ap_1$$

Convergence into the stationary distribution π

After n iterations, for sufficiently large n, the state vector p_n approximates the stationary distribution π. Specifically, $A^{\infty} p_0 = A\pi = \pi$ where $A\pi = \pi$.

Perron-Frobenius Theorem

- If the adjacency matrix A is irreducible then the second-largest eigenvalue in magnitude, λ_2, is strictly less than 1.
- If the graph is non-bipartite, then all eigenvalues λ_m $(\forall m \in \{1, \ldots, \text{ord}(A)\})$ satify $\lambda_m > -1$

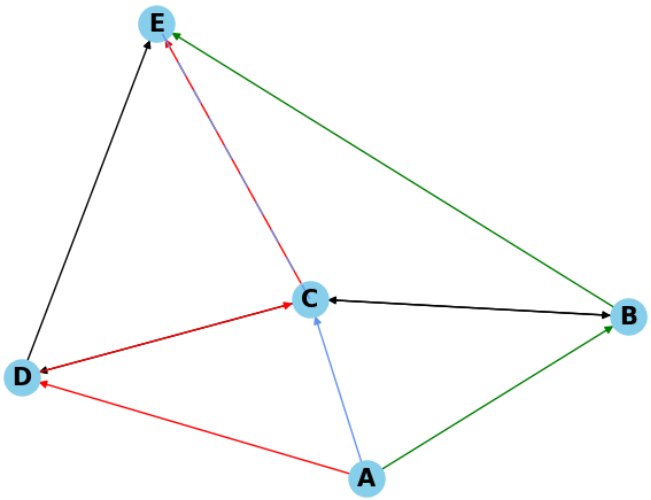
Given these conditions:

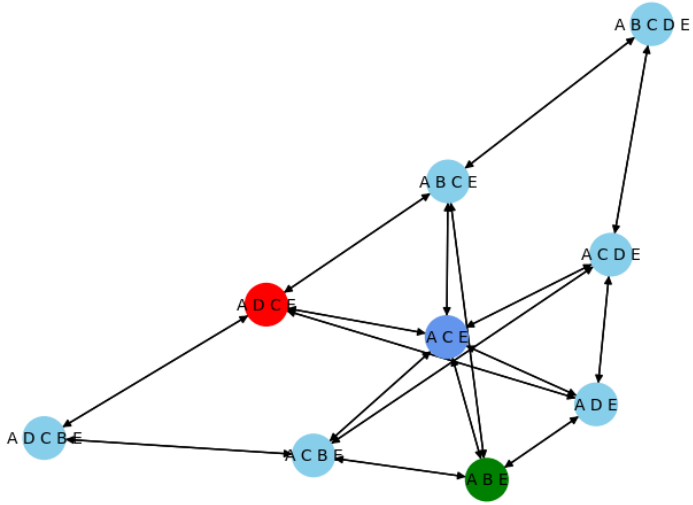
- $\lambda_1 = 1$ is the highest eigenvalue, associated with the stationary distribution π.
- $|\lambda_2|$: the second-largest eigenvalue in magnitude.

The spectral gap δ is defined as: $\delta = \lambda_1 - |\lambda_2|$
The $n(\delta)$ iterations required to converge to the stationary distribution is $n(\delta) = O(\frac{1}{\delta})$

Solution graph for our optimization problems

- The pathfinding problem graph is transformed into an assignment problem, allowing the enumeration of total distinct paths through assignment counting.
- The problems are reformulated into a "solution graph" that encapsulates all possible solution, each one represented by a node.
- Two nodes in this graph are connected if the corresponding solutions are neighbors: differing by one local change (addition, deletion, or substitution) in their paths.





Stationary distribution of the solution graph

Probability of transition from ADCE to ABCE:

$$P_{ADCE \to ABCE} = \frac{1}{2} + \frac{1}{2} \frac{a_{AD}+a_{DC}-(a_{AB}+a_{BC})}{a_{AD}+a_{DC}+(a_{AB}+a_{BC})}$$

If the local change reduces the path length, specifically when $a_{AD} + a_{DC} - (a_{AB} + a_{BC}) < 0$, the probability favors that transition, implying $P(ADCE \to ABCE) > \frac{1}{2}$.
Following the computation of transition probabilities, the resultant probabilities must be normalized to ensure the adjacency matrix's row sums total to 1, maintaining a valid probability distribution across all states.

The stationary state of a random walk on this solution graph reveals that nodes representing the optimal paths stand out.

From adjacency matrix to coin operator with MNRS

- Inspired by the MNRS algorithm, we translate the adjacency matrix A of a graph into a unitary operator U.
- The quantum walk is defined over the edges of a graph, with the state space spanned by states of the form $|x, y\rangle$ where $(x, y) \in E$. The "current" state x is in the first register, and the "next" state y is in the second register.
- The states $|\psi_x\rangle$ for $x \in V$ are defined as: $|\psi_x\rangle = \frac{1}{d} \sum_{x,y \in E} |x, y\rangle$ where d is the degree of vertex x.
- The coin operator $C(P)$ is constructed to mix the next state by implementing a reflection around the star states. It is defined as: $C(P) = 2 \sum_{x \in V} |\psi_x\rangle \langle\psi_x| - I$.

Quantum algorithm for generating solution graph

The quantum algorithm requires four registers:

- The first register indicates the beginning node of each segment in the path.
- The second one indicates the ending node of each segment in the path.
- The third and fourth are unique index for each possible path created.

Leverage superposition to generate all potential paths by mapping S. Each state in the superposition, such as $|0,1\rangle$, $|1,2\rangle$, and $|2,3\rangle$ denotes individual edges in a path.

$$|\psi_S\rangle = \frac{1}{\sqrt{|S|}} \sum_{(i,j)\in S} |i\rangle\,|j\rangle \tag{1}$$

$$|\psi_S\rangle = \frac{1}{\sqrt{|S|}} \sum_{(i,j)\in S} |i\rangle\,|S(i)\rangle \tag{2}$$

$$U_S\,|i\rangle\,|0\rangle^{\otimes n} = |i\rangle\,|S(i)\rangle \tag{3}$$

[0]F. Magniez et al, Search via quantum walk, 2007

Quantum algorithm for generating solution graph

The quantum algorithm requires four registers:

- The first register indicates the beginning node of each segment in the path.
- The second one indicates the ending node of each segment in the path.
- The third and fourth are unique index for each possible path created.

Leverage superposition to generate all potential paths by mapping S. Each state in the superposition, such as $|0,1\rangle$, $|1,2\rangle$, and $|2,3\rangle$ denotes individual edges in a path.

$$|\psi_S\rangle = \frac{1}{\sqrt{|S|}} \sum_{(i,j)\in S} |i\rangle\,|j\rangle \tag{1}$$

$$|\psi_S\rangle = \frac{1}{\sqrt{|S|}} \sum_{(i,j)\in S} |i\rangle\,|S(i)\rangle \tag{2}$$

$$U_S\,|i\rangle\,|0\rangle^{\otimes n} = |i\rangle\,|S(i)\rangle \tag{3}$$

An oracle is employed to evaluate pairs of solutions (nodes). We need to index each pairs of solutions. It returns the transition probability between these pairs for navigating the solution graph.

$$O\,|k\rangle\,|\psi_{S_k}\rangle\,|l\rangle\,|\psi_{S_l}\rangle \tag{4}$$

With ψ_{S_k} the path indexed by k and ψ_{S_l} the path indexed by l.

[0]F. Magniez et al, Search via quantum walk, 2007

Extraction of the Stationary State via Spectral Decomposition

- Determining the stationary state therefore reduces to identifying the eigenspace of the coin operator associated with eigenvalue 1.
- QPE is applied to the operator C with an arbitrary input state : $|\psi\rangle = \sum_i \alpha_i |v_i\rangle$
- the QPE produces the entangled state : $\sum_i \alpha_i |v_i\rangle |\lambda_i\rangle$
- The probability of observing the eigenvalue λ_1 is $|\alpha_1|^2$
- QAA can be employed to increase $|\alpha_1|$
- So, the stationary state is obtained through spectral projection onto the eigenvalue 1

Conclusion

- We directly used the quantum walk property to compute the stationary state.
- We need to find efficient implementation of Quantum Walk Operator based on local path differences.
- Further investigations: Continuous-time quantum walk which avoids some difficulties of the discrete one (coin operation implementation)