QAOA versus Quantum Annealing on Traveling Salesman Problem

...,[1,2] Yongcheng Ding,[1,2,*] and ...

[1] *International Center of Quantum Artificial Intelligence for Science and Technology (QuArtist) and Department of Physics, Shanghai University, 200444 Shanghai, China*

[2] *Department of Physical Chemistry, University of the Basque Country UPV/EHU, Apartado 644, 48080 Bilbao, Spain*

(Dated: September 17, 2019)

to be written at the end

I. INTRODUCTION

to be written at the end

II. MODEL

Here we introduce Traveling Salesman Problem (TSP), as the benchmark problem in this paper. TSP sets the goal of answering the following question: what is the most economical route for a traveling salesman that visits each sites only once and returns to the starting point. This combinatorial optimization problem is proved to be NP-hard, which is notorious for its difficulty to be solved exactly. Even the verification version of TSP, i.e. , checking if there exists any better solution than a given one, is NP-complete. The simplest TSP would be an undirected complete graph, where each edge is weighed by the distance between its nodes. This model could be extended to more complicated cases, e.g., the nodes are not fully connected, or the graph is oriented. Suppose that there are n sites in a d-dimensional space $\mathcal{R}^d$, the distance d_{ij} between site i and j is defined by a certain metric. A weight coefficient w_{ij} represents the cost for a unit distance of d_{ij}, which allows asymmetry of the graph. One can encode the configuration of an arbitrary solution by $n \times n$ binary variables $x_{i,s}$, where $x_{i,s} = 1$ or 0 means "site i is/isnot visited at step s" . The solution is constrained in these ways: (i) only one site can be visited in a single step; (ii) all sites should be visited for only one time in the route.

Accordingly, TSP can be formulated as a 0-1 programming problem for minimizing the total cost

$$\text{cost}(x_{i,s}) = \sum_{i,j,s}^{n} d_{ij} w_{ij} x_{i,s} x_{k,s+1}, \tag{1}$$

$$\text{s.t.} \sum_{i}^{n} x_{i,s} = 1 \text{ for all } s, \tag{2}$$

$$\sum_{s}^{n} x_{i,s} = 1 \text{ for all } i, \tag{3}$$

$$x_{i,1} = x_{i,n+1} \text{ for all } i. \tag{4}$$

A periodic boundary condition Eq. (4) takes the cost for traveling from the last site to the starting point into consideration without introducing extra binary variables. Since the problem is NP-hard, the computing time of any classical algorithm for the worst scenario increases exponentially with the number of sites.

III. QUANTUM OPTIMIZATION ALGORITHM

For problems that cannot be solved by classical algorithms efficiently, quantum computing is applied to speed up the calculation by the principle of quantum mechanics. Among all quantum algorithms, there are quantum optimization algorithms that minimize a given cost function, where is modeled by a combinational optimization problem in this paper. The core idea is that, one may substitute an arbitrary classical binary variable x by qubit operator $\hat{q}$ with eigenvalues of 0 and 1, i.e., $\hat{q}|0\rangle = 0$ and $\hat{q}|1\rangle = 1$ and accelerate the calculation by superpositions and coherence. In this way, the cost function is transformed to a problem Hamiltonian, where its ground state gives the global minimum.

A. Problem Hamiltonian

As we mentioned in Sec. II, the configuration of a certain solution can be encoded by $n \times n$ binary variables, which requires at least the same amount of qubits for the implementation in a quantum computing platform. Much more computational resource can be included as ancilla qubits for achieving error-correction or qubits are not connected according to the graph structure of the problem Hamiltonian.

The problem Hamiltonian can be constructed by additional quadric penalty terms with adequate penalty strengths that represent constraint conditions. For a TSP, the total cost can be translated to a sub-Hamiltonian with the combination of Eq. (1) and (4)

$$H_{\text{cost}} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{s=1}^{n-1} d_{ij} w_{ij} \hat{q}_{i,s}\hat{q}_{j,s+1} + \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} w_{ij} \hat{q}_{i,1}\hat{q}_{j,n}, \tag{5}$$

where qubit operator $\hat{q}_{i,s}$ denotes the status of site i in step s. Constraint conditions Eq. (2) and (3) can be

* jonzen.ding@gmail.com

expressed by penalty terms

$$H_{\text{penalty}} = \sum_{s=1}^{n} \lambda_s (\sum_{i=1}^{n} \hat{q}_{i,s} - 1)^2 + \sum_{i=1}^{n} \mu_i (\sum_{s=1}^{n} \hat{q}_{i,s} - 1)^2, \quad (6)$$

where penalty strength λ_s and μ_i should ensure that any action that violates the constraint conditions will lead to a penalty larger than the reward from the total cost. Hence, the problem Hamiltonian can be given by $H_{\text{P}} = H_{\text{cost}} + H_{\text{penalty}}$, with its ground state to be the global minimum we want to obtain by quantum optimization algorithms.

B. Quantum Approximated Optimization Algortithm

We briefly review Farhi's original protocol of QAOA before our mapping. QAOA is an variational hybrid algorithm based on quantum circuit for approximated solution of combinatorial optimization problem, e.g., MAX-CUT, MAX-2-SAT, etc. To be more specific, this algorithm maximize a cost function $C(z)$, where $z = z_1 z_2 \cdots z_n$ is a bit string with $z_i = \pm 1$ approximately via a sequence of unitary operators. Once we choose the computational basis vector $|z\rangle$, two unitary operators can be defined by $U(B, \beta) = \exp(-i\beta B)$ and $U(C, \gamma) = \exp(-i\gamma C)$, where $B = \sum_i^n \sigma_x$ is the sum of n Pauli-X operators. Angle β and γ as independent parameters in $[0, 2\pi]$ and $[0, pi]$, respectively. The initial state is prepared to be a uniform superposition $|s\rangle = |+_i +_2 \cdots +_n\rangle$, which is the ground state of B. An angle dependent quantum state can be evolved according to $2p$ unitary operators

$$|\beta, \gamma\rangle = U(B, \beta_p) U(C, \gamma_p) \cdots U(B, \beta_1) U(C, \gamma_1) |s\rangle, \quad (7)$$

where the expectation value of C can be measured by $F_p = \langle \beta, \gamma | C | \beta, \gamma \rangle$. For a fixed accuracy p, the set of angles (β, γ) can be altered to obtain a better F_p. The maximum value of F_p is always larger than that of F_{p-1}, and when $p \to \infty$, this algorithm gives an exact global maximum.

For solving TSP by QAOA, the problem Hamiltonian should be reformulated to fit this protocol. However, following the same problem Hamiltonians setting given above will not guaranteed a valid solution satisfied with the constraints, since the mixer Hamiltonian doesn't preserve the constrained solution space. To deal with this issue, we adapted a different ansatz scheme, the Quantum Alternating Operator Ansatz [], in which the initial state $|s\rangle$ is set to a valid state and mixer Hamiltonians are carefully desigened to ensure that $|s\rangle \in \Omega$ and $U(B, \beta_p)|s\rangle \in \Omega$, where Ω is the solution space of the formulated problem. We presented more details of the method in the appendix part.

C. Quantum Annealing

Quantum annealing is another approach to accelerate computation of specific problems by the principle of quantum mechanics. A more precise description of the algorithm would be adiabatic quantum computation (AQC), where a quantum system is initially prepared to the ground state of a simple Hamiltonian and then adiabatically evolved to the problem Hamiltonian. The system is supposed to be in the ground state of the problem Hamiltonian at the end, which is guaranteed by the adiabatic theorem. The estimation of evolving time $t_f = O(1/\text{gap}^2_{\text{min}})$ is governed by the minimal energy gap of the problem Hamiltonian. The Hamiltonian of the quantum system can be written as

$$H(t) = -A(\frac{t}{t_f}) \sum_i^n \hat{\sigma}_i^x + B(\frac{t}{t_f}) H_{\text{spin}}, \quad (8)$$

where $A(\frac{t}{t_f})$ smoothly evolves from $A(0)$ to negligible and $B(\frac{t}{t_f})$ grows reversely to $B(1)$.

However, AQC cannot be achieved easily for the following reasons: (i) one need to find gap_{min} of the problem Hamiltonian for ensuring an adiabatic evolution, which is equivalent to find the ground state of it; (ii) even if the evolution of time can be given in advance, decoherence and noise can affect the performance of the algorithm massively. Thus, quantum annealing accelerate the evolution by tunneling the traverse magnetic field faster, which reduces the computing time. The quantum system can be excited to a higher energy state during the process, but still remains a relative high possibility of obtaining a ground state of the problem Hamiltonian once the initial traverse magnetic field is switched off. Following this idea, quantum annealer as computing platform has been built by concluding the global minimum with statistical methods.

IV. EXPERIMENTS

A. Quantum Approximate Optimization Algorithm

Generally, exponential speedup is considered to be the most important advantage of quantum computing. However, for hybrid variational quantum algorithms, rigorous analysis on complexity is generally difficult. Since the performance of optimization process varies with many relative factors such as problem type and instance, optimizer algorithm and parameter settings, etc.

Up to now, it's still a open problem wether variational hybrid quantum algorithms can present quantum advantages versus classical computer or quantum annealer. A numerical experiment may help to address this problem.

We implemented quantum circuits for QAOA based on ETH ProjectQ quantum framework[]. and invoke a classical optimizer for the variational optimization process. The simulation is performed on Huawei HiQ cloud platform[] with a enhanced high performance, distributed quantum simulatior as the backend.

For travelling salesman problem, the circuit is enhanced with operator ansatz scheme. We deployed benchmarking with problem scale ranging from 4*4(9 qubits) to 7*7 (36 qubits), and 100 randonm samples in each batch.

The choice of optimizer and parameter setting also affect the performance of hybrid algorithms such as QAOA. Previous numerical research has discussed the effect of various optimizers[]. Here we pick the Constrained Optimization by Linear Approximation (COBYLA) algorithm as the optimizer, with auto-stop threshold set to 0.2.

The code and data for thie benchmarking is available online[].

1. *Space Complexity*

We evaluate the space complexity of QAOA by counting the resource used by the quantum circuit[]. For gate based computing model, such as ion trap and superconducting circuit[], the resource is countting by number of gates used and circuit scale, more specificlly, maximum depth and width of circuit.

The main issue here is counting the resources used by time evolution operator. In gate model, time evolution unitary with a fermionic Hamiltonian expressed in form of Pauli operators can be precisely simulated[]. The mixer and target Hamiltonians addressed in QAOA can firstly be decomposed into commute sub-hamiltonians and then compiled into H, $CNOT$ and Rz gates, after which a resource counting is possible.

2. *Time Complexity*

For hybrid variational algorithms, the time consumption comes from both the quantum and classical parts. Employing a gradient based optimizer, the variational optimization process is an iterative process invoking gradient calculation and quantum based cost evaluation repeatedly, where the time complexity depends on the number of iterations and the execution time of quantum circuit.

While the iterations can vary with instances, the time consumption for each evaluation of quantum circuits is decisive, propositional to circuit depth depending on problem size and Hamiltonian setting adopted for specific problem type. Denoting the time complexity of the optimization iterations to be $O(f(n))$ and the complexity of quantum evaluation to be $O(g(n))$, then the time complexity of variational algorithm can be expressed by $O(f(n)g(n))$.

Based on the experimental data of superconducting circuit, we can even roughly evaluated the time comsuption of QAOA. More details is presented in the appendix.

3. *Approximation Ratio*

Here we pick the approximation ratio, quotient between the QAOA-optimized cost and global optimal cost given by classical algorithm. Previously numerical work has implied that the approximation ratio stabley increase with the layer number p.

B. Quantum annealer simulator

Before we test the the performance of quantum annealing on quantum computing platform, we simulate the process by a classical device for a better understanding of the algorithm. Even though D-Wave provides local solver in the software, it does not simulate the evolution of a quantum system but solves the effective 0-1 programming problem with classical algorithm instead, e.g., tabu search for the local server in *qbsolv*. Since the problem Hamiltonian for TSP problem is stoquastic, i.e., free from sign problem, we apply Path-Integral Quantum Monte Carlo Method (PIQMC) [1] to simulate the D-Wave quantum annealer. The algorithm allows finite but small temperature for simulating quantum tunneling, and gives a coarse population for each eigenstate without exactly simulation of its dynamics. In Appendix. B, we introduce the algorithm amply by providing its principle and technical details for implementing.

C. D-Wave quantum annealer

D-Wave 2000Q

V. DISCUSSION

Our numerical results indicates that the default ansatz of QAOA is only suitable for optimization problems with free solution space, such as MAXCUT or 3-SAT, etc. Even though the setting of penalty terms will minimize the violation of constraints, current QAOA design on a shallow circuit NISQ device can't guarantee a valid solution. To tackle constrained problems, we must adapted new schemes such as operator ansatz.

Potential enhancements

VI. CONCLUSION

to be written at the end

VII. ACKNOWLEDGMENTS

We acknowledge funding from projects QMiCS (820505) and OpenSuperQ (820363) of the EU

Flagship on Quantum Technologies, Spanish Government PGC2018-095113-B-I00 (MCIU/AEI/FEDER, UE), Basque Government IT986-16, Shanghai Municipal Science and Technology Commission (18010500400 and 18ZR1415500), and the Shanghai Program for Eastern Scholar, as well as the and EU FET Open Grant Quromorphic. This work is supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR) quantum algorithm teams program, under field work proposal number ERKJ333.

[1] R. Martoňák, G. E. Santoro, and E. Tosatti, Phys. Rev. B **66**, 094203 (2002).
[2] Our GitHuB repo, still private now :)

Appendix A: Path Integral Monte Carlo Algorithm for Simulating Quantum Annealing

Here we review the path integral formalism (based on Suzuki-Trotter theorem) of the spin-1/2 Hamiltonian (Eq. (8)) that enables us to sample onr an effective classical system of higher-dimension before we introduce how to simulate quantum annealing. We denote the tunneling Hamiltonian and the problem Hamiltonian by H_A and H_B, respectively, which are not commutable. The partition function $Z = \mathrm{Tr}\exp(-\beta H)$can be written as

$$Z = \mathrm{Tr}(\exp(-\beta(H_\mathrm{A} + H_\mathrm{B})/m)^m \quad \text{(A1)}$$

$$= \sum_{z^k}\prod_k \langle z^k|\exp(-\beta(H_\mathrm{A} + H_\mathrm{B})/m)|z^{k+1}\rangle, \quad \text{(A2)}$$

with Trotter number m, spin configuration of each Trotter slice z^k, and periodic boundary condition $z^{m+1} = z^1$. Once we ignore the incommutability of H_A and H_B, an approximated partition function Z_approx can be accepted by the substitution $\langle z^k|\exp(-\beta(H_\mathrm{A} + H_\mathrm{B})/m)|z^{k+1}\rangle \rightarrow \langle z^k|\exp(-\beta H_\mathrm{A}/m)\exp(-\beta H_\mathrm{B}/m)|z^{k+1}\rangle$, where

$$\langle z^k|\exp(-\beta H_\mathrm{A}/m)\exp(-\beta H_\mathrm{B}/m)|z^{k+1}\rangle \quad \text{(A3)}$$

$$= \langle z^k|\exp(-\beta H_\mathrm{A}/m)|z^{k+1}\rangle\exp(-\beta H_\mathrm{B}(z^{k+1})/m) \quad \text{(A4)}$$

The nontrivial term can be rewritten as

$$\langle z^k|\exp(-\beta H_\mathrm{A}/m)|z^{k+1}\rangle \quad \text{(A5)}$$

$$= \prod_{i,j}\langle z^k_{i,j}|\exp(\frac{\beta A}{m}\hat{\sigma}^x_{i,j})|z^{k+1}_{i,j}\rangle \quad \text{(A6)}$$

After calculating the contributions of all terms included, the approximated partition function, $(2+1)$-dimensional Hamiltonian and coupling strength between Trotter slices can be given as

$$Z_\mathrm{approx} = (\frac{1}{2}\sinh\frac{2A}{mT})^{\frac{mn}{2}}\sum_{z^k_{i,j}}\exp(-\frac{H_{2+1}}{mT}), \quad \text{(A7)}$$

$$H_{2+1} = \sum_{k=1}^{m}(H_\mathrm{B} - J_\perp\sum_{i,j} z^k_{i,j}z^{k+1}_{i,j}), \quad \text{(A8)}$$

$$J_\perp = -\frac{mT}{2}\ln\tanh\frac{A}{mT} \quad \text{(A9)}$$

where $z^k_{i,j}$ is the classical spin variable that denotes the state of site (i,j) in Trotter slice k.

After obtaining the Hamiltonian that describes the equivalent $(2+1)$-d classical system, we implement PIQMC to simulate quantum annealing with this algorithm: (i) input $N = n \times n$, m, T, for the number of the spins in H_spin for 2-d quantum system, Trotter number, and system temperature, respectively; (ii) design the quantum annealing schedule as a list $\{(A_0, B_0), \cdots, (A_l, B_l)\}$ with $l+1$ elements that contains discretized weight of H_A and H_B; (iii) randomize a spin configuration of H_spin and make $m-1$ copies to generate the initial configuration for H_{2+1}; (iv) start with (A_0, B_0), generate the new $(2+1)-$d Hamiltonian, flip each spin and accept the configuration by Metropolis acceptance criterion, and go for the next pair of weights; (v) decode the solution by finding the minimum energy of H_spin with spin configurations among all m Trotter slices.

To define a path integral, all paths should be restricted to be the same state at the beginning and the end, but are allowed to vary in between. Step (iii) guarantees the boundary condition at the beginning, and paths are very likely to end at the same state since the coupling strength between Trotter layers grows larger when A_l is negligible. Different from simulated annealing algorithm, we adjust weights of Hamiltonians instead of system temperature T, which means T is fixed and could be understood as the probability of quantum tunneling. As Ref. [1] suggests, mT should be larger than the characteristic coupling strength J between sites in each layer for obtaining a thermal equilibrium. Meanwhile, T should be set to an adequate value that the acceptance possibility of a positive energy shift is reasonable. This avoids the error after decoding, e.g., constraint conditions Eqs. (2), (3) are violated because of an accepted spin flip that cost more due to the penalty. Results that breach constraint conditions are excluded before data processing. The accuracy can be enhanced by introducing more annealing steps, trotter slices, and Monte Carlo steps for each spin.

The algorithm can be time-consuming if the classical Hamiltonian H_{2+1} contains too many spins, which significantly affect the efficiency for finding a global minimum of a large TSP problem. The most trivial trick for speed up would be calculating the energy shift instead of evaluating the system energy twice. Once a spin is flipped, coupled spins are influenced which means we only need to calculate $3n-1$ spins in the same layer and another two contributed by its neighbor located in the nearest

Trotter slices. The algorithm can also be parallelized for GPU acceleration, if adequate state updating policy is applied. Codes for implementing this algorithm on TSP is uploaded to an open access repository [2].

Appendix B: Improved anstaz schme for QAOA: Quantum Alternating Operator Ansatz on TSP

Quantum Alternating Operator Ansatz[] (Operator Ansatz) is an enhanced ansatz scheme for Quantum Approximation Optimization Algorithm (QAOA).

In Operator Ansatz, any optimization problem can be formalized in a quantum-feasible form composed by three parts:

1. Solution space of problem, Ω
2. Phase-separator (Target Hamiltonian) $U_{\gamma_p,C} = e^{-i\gamma_p C}$
3. Mixer (Mixer/Driver Hamiltonian) $U_{\beta_p,B} = e^{-i\beta_p B}$

The span of all possible solutions (state configuration) of the problem forms the solution space. $F = span(\{|\psi_i\rangle\})$ where $|\psi_i\rangle$ represents one possible state configuration of the problem.

The basic idea of Operator Ansatz is that the driver Hamiltonian should be a operator that preserve the solution space. $H_m : \Omega \mapsto \Omega$. And for any two states in solution space, $\langle\psi_i|U_{\beta,p}|\psi_j\rangle \neq 0$. That means we can transfer between any two configurations in solution space.

The setting of Operator Ansatz ensures that if the state ansatz is a solution of the problem, all the unitary transformations performed in QAOA will preserve the solution space and avoid searching invalid state configurations.

1. Solution Space of Traveling Salesman Problem

We need to find the order of traveling for the given n cities in TSP. In fact, TSP is a permutation problem, with $n!$ possible configurations in solution space. Since both the label and the order of a city should be determined, at least n^2 qubits are needed for one solution of TSP using one-zero encoding.

$$x_{u,i} = \begin{cases} 1, & \text{if city } u \text{ is visited in step } i \\ 0, & \text{otherwise} \end{cases}$$

For example, $|\psi_1\rangle = |0010\rangle$ means for a 4-city TSP, city 3 is visited at the first step. The complete solution is something like

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes |\psi_3\rangle \otimes |\psi_4\rangle = |1000\rangle|0100\rangle|0010\rangle|0001\rangle$$

This state can be represented by a sparse matrix , since each city in TSP must be visited only once, which means the sum of each column and row of the matrix must be one.

2. Target Hamiltonian of TSP

Since the solution of TSP is a loop, we only care the relative visiting order of cities. In other words, the energy levels of TSP are n-degenerate. By fixing the first city in travel, we can eliminate the degeneracy and reduce the qubits needed in search. The cost Hamiltonian of TSP with the first city fixed is

$$\hat{H}_{cost} = \sum_{u=1}^{n} d_{0,u}\hat{q}_{u,1} + \sum_{i=1}^{n-1}\sum_{u=1}^{n}\sum_{v=1}^{n} d_{u,v}\,\hat{q}_{u,i}\,\hat{q}_{v,i+1} + \sum_{v=1}^{n} d_{v,0}\,\hat{q}_{v,n}$$

in which $\hat{q}_{u,i}$ denotes the state of city u in step i. and $d_{u,v}$ denotes the distance between the city u and the city v. The $n+1$ cities are labeled by 0,1,2...n, and the first city 0 is fixed as the start point of TSP.

Here, the $\hat{q}_{u,i}$ is a 0/1 variable. Translate it into spin operator form $\hat{q}_{u,i} = \frac{1}{2}(I - Z_{u,i})$, in which $Z_{u,i}$ denotes the Pauli Z operator applying on the u, i th qubit.

3. Mixer Hamiltonian of TSP

To satisfy the conditions of Operator Ansatz, the mixer Hamiltonian must be carefully constructed. We must ensure that the mixer could preserve the solution as well as is able to iterate all the possible states in solution space. The swap operator that transfers one TSP solution into another one is a good choice for the partial mixer.

First define

$$\begin{cases} S^+ = X + i\,Y = |1\rangle\langle 0|, \text{ operator that flips 0 to 1} \\ S^- = X - i\,Y = |0\rangle\langle 1|, \text{ operator that flips 1 to 0} \end{cases}$$

Then the swap operator can be wrote as

$$\hat{H}_{ps\{i,j\}\{u,v\}} = S^+_{i,u} S^+_{j,v} S^-_{i,v} S^-_{j,u}$$

If city v is visited at i step and city u visited at j step, the operator will swap the visiting order and produce another configuration of TSP. Since this operator is not an Hermitian, we need to add its inverse to get an unitary.

$$\begin{aligned} \hat{H}_{U\{i,j\}\{u,v\}} &= \hat{H}_{ps\{i,j\}\{u,v\}} + \hat{H}^\dagger_{ps\{i,j\}\{u,v\}} \\ &= S^+_{i,u} S^+_{j,v} S^-_{i,v} S^-_{j,u} + S^+_{i,v} S^+_{j,u} S^-_{i,u} S^-_{j,v} \end{aligned}$$

To iterate every possible state in solution space, the full mixer must contains all the partial swap mixer.

$$\hat{H}_{mixer} = \sum_{i,j}\sum_{u,v} \hat{H}_{U\{i,j\}\{u,v\}}$$

4. Circuit Implementation of Quantum Alternating Operator Ansatz

In QAOA, we must implement the time evolution of the Hamiltonian. Noticed that the Hamiltonian is actually a sum of many Pauli terms and the order of operator does not matter, it's possible to divide these terms into non-commute groups and reduce the circuit depth with some techniques []. The basic idea is grouping non-commute operations and putting them in one layer to parallelize the computation.

For the TSP mixer, the basic unit of our mixer is 4-bit partial-swap operator determined by step i and city u indices. No doubt that the dual indices make this problem more complicated. As the figure shows, we can first group these operators by parity (even or odd) for the step index. Then one parity into subgroups by finding non-commute city index. Formal statement of finding non-commute city index equals least edge-coloring problem in graph theory, where minimum number of colors are allocated to disjoint edges.

The number partial swap operator used in mixer is $C_n^2 = n(n-1)/2$. The circuit depth is the same with a natural order. After optimization, the circuit depth equals the number of groups of non-commute partial operators, which is $\pi\kappa p$, in which p is the number of parities, κ is the number of colors and p is the depth of Trotterization. Obviously this result is much better that a quadratic depth increase.