QAOA Tutorial Outline

- Quantum Approximate Optimization Algorithm: review and status
- The «Quantum Alternating Operator Ansatz»
  - Mixing Operators
  - Examples
- Compiling and Executing
  - The gate synthesis problem
  - Review of compilation methods
  - Compiling framework in nearest-neighbor architectures

Davide Venturelli, Ph.D.
USRA Research Institute for Advanced Computer Science
Quantum Artificial Intelligence Laboratory
NASA Ames Research Center
davide.venturelli@nasa.gov

READING LIST

1. Review on QAOA Research
1. Design a binary optimization classical Hamiltonian ("phase separation")
2. Design a unitary operator that can connect and allow jumps between different states ("mixing")
3. Prepare a QAOA state for some parameters
   \[ |\beta, \gamma\rangle = Q_p(\beta, \gamma) |s\rangle \]
   \[ Q_p(\beta, \gamma) = U_M(\beta_p) U_P(\gamma_p) \cdots U_M(\beta_1) U_P(\gamma_1) \]
4. Measure the state in the computational value and compute the exp. value of \( C(z) \)
   \[ F_p(\gamma, \beta) = \langle \gamma, \beta | C | \gamma, \beta\rangle \]
   \[ M_p \geq M_{p-1} \]
   \[ M_p = \max_{\gamma, \beta} F_p(\gamma, \beta) \]
   \[ \lim_{p \to \infty} M_p = \max_z C(z) \]
5. Change the parameters if they are not proven optimal and repeat 3-4
1. UNDERSTANDING

2. PRACTICAL ASPECTS
   b. Hybridization: Guerreschi (2016)

3. GENERALIZATIONS/APPLICATIONS

4. EXPERIMENTS
1. UNDERSTANDING

2. PRACTICAL ASPECTS
   b. **Hybridization**: Guerreschi (2016)

3. GENERALIZATIONS/APPLICATIONS
   a. **Combinatorial Optimization**: Hadfield (2016)
   b. **Factoring**: Anschuetz (2018)

4. EXPERIMENTS
   a. **Superconducting**: Otterbach (2017)
   b. **Neutral Atoms**: Pichler (2018)
1. UNDERSTANDING

2. PRACTICAL ASPECTS
   b. Hybridization: Guerreschi (2016)

3. GENERALIZATIONS/APPLICATIONS

4. EXPERIMENTS
2. QAOA for Constrained Optimization Problems
QAOA Applications

- Maximum Cut
- Max-SAT, Min-SAT, NAE-SAT
- Set Splitting
- MaxE3LIN2
- Max-ColorableSubgraph
- Graph Partitioning
- Maximum Bisection
- Max Vertex k-Cover
- MaxIndependentSet
- MaxClique
- MinVertexCover
- MaxSetPacking
- MinSetCover
- TSP
- SMS with various metrics and constraints
- ...

Objective Function: Soft Constraints
Feasible States: Hard Constraints
Alternating Operator Ansatz

Some unitary respecting:
- Preserve the feasible subspace
- Provide all-to-all nonzero transitions between all feasible states
- Non-necessarily time evolution of a local Hamiltonian

Some initial state respecting:
- It is a superposition of several solutions in the feasible subspace
- It can be prepared efficiently

\[ Q_p(\beta, \gamma) = U_M(\beta_p)U_P(\gamma_p) \cdots U_M(\beta_1)U_P(\gamma_1) \]

\[ |\beta, \gamma\rangle = Q_p(\beta, \gamma) |s\rangle \]

\[ e^{-i\beta H_M} \quad e^{-i\gamma H_P} \]

Some unitary respecting:
- Is diagonal in the computational basis
- The spectrum of HP encodes the objective function

\[ H_f |x\rangle = f(x) |x\rangle \]
Node \( u \) is colored by \( c \)

\[ X_{u,c} = 1 \]

Phase Separator (QUBO objective function)

\[ H'_P = \frac{4 - \kappa}{4} m I + \frac{1}{4} \sum_{\{u,v\} \in E} \sum_{a=1}^{\kappa} (Z_{u,a} + Z_{v,a} - Z_{u,a}Z_{v,a}) \]

Initial state:

\[ |W\rangle_v = \frac{1}{\sqrt{k}} (|100 \cdots 0\rangle + |010 \cdots 0\rangle + |0 \cdots 01\rangle) \]

- Babbush (2017)
- Verstraete (2009)
- Wang (2009)
- Childs (2002)
- ...
Exp(iH_{ring}) is difficult to implement

$$H_{ring}^{(enc)} = \sum_a^d (X_a X_{a+1} + Y_a Y_{a+1})$$

Respects the Hamming Weight constraint
Design Freedom and Tradeoffs

\[ H^{(\text{enc})}_\text{ring} = \sum_a^d (X_a X_{a+1} + Y_a Y_{a+1}) \]

Exp(iH\text{ring}) is difficult to implement

\[ U_M = \prod_{v=1}^n U_{v,\text{parity}}^{(\text{enc})} \prod_{a} \text{Exp}(iX_a X_{a+1} + Y_a Y_{a+1}) \]

\[ U_M = [U_1 U_3 U_5 U_7] [U_2 U_4 U_6 U_8] [U_1 U_3 U_5 U_7] [U_2 U_4 U_6 U_8] \ldots \]

This couples only distance 2; has to be repeated k/2 times

All these 2-qubit k^2/2 gates need to be scheduled

Respects the Hamming Weight constraint
Finding the largest induced subgraph colorable by k colors

Node u is colored by c or uncolored (c=0)

\[ X_{u,c} = 1 \]

\[ C = m - \sum_v x_{v,0} \]

\[ H_C = \frac{1}{2} \sum_v Z_{v,0} \]

Still needs to be compiled to 2 qubit gates

All these gates need to be scheduled
In **traveling salesman** encoding

\[ X_{v,j} = 1 \text{ if city } v \text{ is visited as } j^{th} \]

\[ \sum_{(u,v) \in E} d_{u,v} \sum_{j=1}^{n} (x_{u,j} x_{v,j+1} + x_{v,j} x_{u,j+1}) \]

\[ H_{PS,i,j,\{u,v\}}^{(enc)} = S_{u,i}^{+} S_{v,j}^{+} S_{u,j}^{-} S_{v,i}^{-} + S_{u,i}^{-} S_{v,j}^{-} S_{u,j}^{+} S_{v,i}^{+} \]

(partitioned using edge coloring and parity

\( \approx (n-1)n^2/4 \) mixers)

(needs to be repeated \( n(n-1)/2 \) times for all-to-all)

In **single machine scheduling**

\[ X_{j,t} = 1 \text{ if job } j \text{ starts at time } t \]

\[ C = \sum_j w_j \sum_{t} x_{j,t}(t + p_j - d_j) \]

\[ H_{TS,i,j,\{t\}}^{(enc)} = S_{i,t+p_j}^{+} S_{j,t}^{+} S_{i,t}^{-} S_{j,t+p_j}^{-} + S_{i,t}^{+} S_{j,t+p_j}^{+} S_{i,t+p_j}^{-} S_{j,t}^{-} \]

(But if we add release dates then we need controls on the no-overlap constraint)
Zoology of Ansatze

Bitflip mixers
• Maximum Cut
• Max-SAT, Min-SAT, NAE-SAT
• Set Splitting
• MaxE3LIN2

Controlled Bitflip mixers
• MaxIndependentSet
• MaxClique
• MinVertexCover
• MaxSetPacking
• MinSetCover

XY mixers
• Max-ColorableSubgraph
• Graph Partitioning
• Maximum Bisection
• Max Vertex k-Cover

Controlled XY mixers
• Max-k-ColorableInducedSubgraph
• MinGraphColoring
• MinCliqueCover

Permutation mixers
• TSP
• SMS with various metrics and constraints

From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz


* Department of Computer Science, Columbia University, New York, NY
† Quantum Artificial Intelligence Lab., NASA Ames Research Center, Moffett Field, CA
‡ USRA Research Institute for Advanced Computer Science (RIACS), Mountain View, CA
§ Stinger Ghaffarian Technologies, Inc., Greenbelt, MD
∥ Berkeley Quantum Information and Computation Center and Departments of Chemistry and Computer Science, University of California, Berkeley, CA

September 12, 2017

The next few years will be exciting as prototype universal quantum processors emerge, enabling implementation of a wider variety of algorithms. Of particular interest are quantum heuristics, which require experimentation on quantum hardware for their evaluation, and which have the potential to significantly expand the breadth of applications for which quantum computers have an established advantage. A leading candidate is Farhi et al.’s Quantum Approximate Optimization Algorithm, which alternates between applying a cost-function-based Hamiltonian and a mixing Hamiltonian. Here, we extend this framework to allow alternation between more general families of operators. The essence of this extension, the Quantum Alternating Operator Ansatz, is the consideration of general parameterized families of unitaries rather than only those corresponding to the time-evolution under a fixed local Hamiltonian for a time specified by the parameter. This ansatz supports the representation of a larger, and potentially more useful, set of states than the original formulation, with potential long-term impact on a broad array of application areas.
3. QAOA in the “Real World”
Quantum Circuits can be composed by single and two-qubit gates of universal set*

**CNOT, $R_y(\theta)$ and $R_z(\alpha)$**

Each single qubit gate can be decomposed by single qubit rotations.

$U_1 = R_z(\alpha) \, R_y(\beta) \, R_z(\gamma) \, e^{i\phi}$

Each two qubit gate is reversible and it is representable by a Unitary Matrix.

$R_z$ gates can be «virtually» compiled. (McKay 2017 and refs)

* active research to natively support multi-qubit gates

---

Barenco et al. (1995)
Kraus, Cirac (2001)
Vatan, Williams (2003)

Maximum number of elementary 1-qubit gates: **15**
Maximum number of CNOTs: **3**
Maximum depth assuming $R_y$, $R_z$ and simplifications: **11**
Performance of algorithms in NISQ will depend on aspects such as gate fidelities, parallelization, idle time, crosstalks..

Different Metrics to optimize correlate to final performance:
- Total Quantum Factor
- Quantum Volume
- Number of Two-Qubit Gates
- Makespan


...
Example: MaxCut

Interaction graph obtained from quadratic objective function (MAXCUT)

\[ U = \frac{1}{2} \sum_{(i,j) \in E} (1 - s_i s_j) \]

- Defines the cut
- Counts the edges in the cut
- Mixes the two partitions

\[ \sum_i X_i \]

\[ s_i = \pm 1 \]

- Every edge is a gate that needs to be executed (in arbitrary order)
- The same graph has to be executed multiple times \((p \text{ rounds})\).
- Every qubit has to complete all the gates of round \(p\) before being involved in \(p+1\)

\[ U_{PS} = \prod_{jk} \text{Exp}(ibZ_j Z_k) \]

\[ U_M = \prod_j \text{Exp}(igX_j) \]
Interaction graph obtained from quadratic objective function (MAXCUT)

Initial assignment $q_i \rightarrow n_i$

- Every edge is a gate that needs to be executed (in arbitrary order).
- The same graph has to be executed multiple times ($p$ rounds).
- Every qubit has to complete all the gates of round $p$ before being involved in $p+1$. 
Circuit Execution Schedule

Exp(iθZ_1Z_4)  ZZ-Evolution Gate

Duration 2τ_2 + 4τ_1
Circuit Execution Schedule

From Unidirectional CNOTs to SWAP

\[ \text{SWAP } |\Psi\rangle_3 \leftrightarrow |0\rangle \]

Duration \(3\tau_2+4\tau_1\)
SWAPS can also be inserted as part of the UZZ interaction without the need to be sequential.
Circuit Execution Schedule

Objective: finding the makespan-minimizing Gantt Schedule for p=1, p=2, N=8, N=21

Benchmark presented at ICAPS17
Circuit Execution Schedule
Circuit Execution Schedule
Circuit Execution Schedule

[Diagram of quantum circuit execution schedule]
Circuit Execution Schedule

q1 q2 q3 q4 q5 q6 q7

Q14 Q13 Q15

The diagram above represents a quantum circuit with qubits q1 to q7 and operations Q14, Q13, and Q15. The schedule below shows the execution order of these operations.
All actions of round 1 are completed – qubit can be mixed. Qubit 1 can start participating to round 2.
P-S(3,7) is fast on $n_1, n_4$
P-S(3,7) is slow on $n_4, n_6$

How to obtain these schedules efficiently?
Temporal Planning

Planning Domain Specification Language (PDDL)

(:durative-action swap_1_2
 :parameters (?q1 - qstate ?q2 - qstate)
 :duration (= ?duration 2)
 :condition (and (at start (located_at_1 ?q1))
 (at start (located_at_2 ?q2)))
 :effect (and (at start (not (located_at_1 ?q1)))
 (at start (not (located_at_2 ?q2)))
 (at end (located_at_1 ?q2))
 (at end (located_at_2 ?q1))))
LPG (Gerevini 2003): *Local Search*

SGPlan (Wah 2004): *Subproblem planning decomposition*

CPT (Vidal 2006): *Partial-Order-Causal Link (POCL)*

TFD (Eyerich 2009): *Fast-Forward Search (FFS) + postprocessing*

POPF (Coles 2010): *POCL+FFS+Simple Temporal Network*

Plan quality relative comparison

<table>
<thead>
<tr>
<th></th>
<th>p=1, N8</th>
<th>p=1, N21</th>
<th>p=2, N8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Utilization u</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>1.0</td>
<td>0.9</td>
<td>1.0</td>
</tr>
<tr>
<td>0.9</td>
<td>1.0</td>
<td>0.9</td>
<td>1.0</td>
</tr>
<tr>
<td>0.9</td>
<td>1.0</td>
<td>0.9</td>
<td>1.0</td>
</tr>
<tr>
<td>0.9</td>
<td>1.0</td>
<td>0.9</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>p=1, N8</th>
<th>p=1, N21</th>
<th>p=2, N8</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPG</td>
<td><strong>0.88</strong></td>
<td>0.89</td>
<td>0.91</td>
</tr>
<tr>
<td>TFD</td>
<td>0.87</td>
<td>0.87</td>
<td><strong>0.95</strong></td>
</tr>
<tr>
<td>SGPlan</td>
<td>0.67</td>
<td>0.68</td>
<td>0.64</td>
</tr>
<tr>
<td>POPF</td>
<td>0.81</td>
<td>0.81</td>
<td><strong>0.94</strong></td>
</tr>
</tbody>
</table>

(CPT is provably optimal but timing scales badly)
CIRQ Toolchain

Quantum user-assisting Software for applied research (QuaSar)

QuaSar is a unified framework to run algorithms and simulations on both classical and quantum devices using a unified and user-friendly interface.

- QuaSAR Server
- Generic GUI
- Target Application GUI/UI
- Quantum Compiler
- Driver
- API
- Results
- QuaSAR Hardware
  - Google HW
  - Rigetti HW
  - IBM HW
  - D-Wave HW
- Supercomputer
  - QuEST
  - QHIPSTER
  - qTorch
  - UFO@NASA
- Benchmark Tools
  - OpenFermion
  - QAOA
  - ...

User's Environment
- Generic GUI
- Target Application GUI/UI
- OpenFermion
- QAOA
- ...

NASA (Advanced Exploration Systems AES, Convergent Aeronautics Solutions CAS)
Contacts

Davide Venturelli, Ph.D.
USRA Research Institute for Advanced Computer Science
Quantum Artificial Intelligence Laboratory
NASA Ames Research Center
davide.venturelli@nasa.gov