QuantumGEP: Gene Expression Programming for Quantum Computing

Gonzalo Alvarez,
Ryan Bennink,
Stephan Irle,
Jacek Jakowski

March Meeting of the APS. March 21st, 2023


Credit: MIT

Motivation: Find Quantum Circuits

Which quantum circuits do we want?

Those that when applied to an initial state yield the ground state of a Hamiltonian

Motivation: Find Quantum Circuits

Credit: MIT

- **Which** quantum circuits do we want?
Those that when applied to an initial state yield the ground state of a Hamiltonian

- **Why?** In order to use **quantum computing** in Condensed Matter, Graph Theory, and Quantum Chemistry

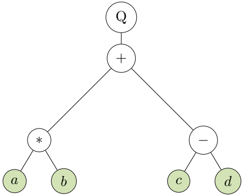
Motivation: Find Quantum Circuits

- Which quantum circuits do we want? Those that when applied to an initial state yield the ground state of a Hamiltonian

- Why? In order to use quantum computing in Condensed Matter, Graph Theory, and Quantum Chemistry

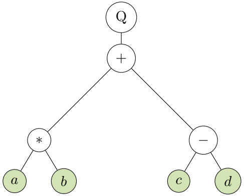
- QuantumGEP can produce thousands of circuits, with hundreds correct; it produces non-parametric circuits

Gene Expression Programming Overview

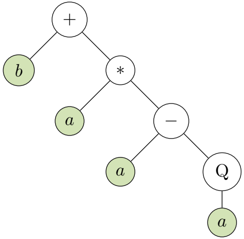


Abstract Syntax Tree (AST) for $\sqrt{a * b + (c - d)}$, with string representation Q+*-abcd, operator (or primitives) in $\{\mathrm{Q}, +, -, *\}$, and leaves (shaded in the figure) in $\{a,\ b,\ c,\ d\}$.
Order is breadth first. No parenthesis!

Gene Expression Programming Overview



Abstract Syntax Tree (AST) for $\sqrt{a*b+(c-d)}$, with string representation Q+*-abcd, operator (or primitives) in $\{Q,+,-,*\}$, and leaves (shaded in the figure) in $\{a,\ b,\ c,\ d\}$. Order is breadth first. No parenthesis!

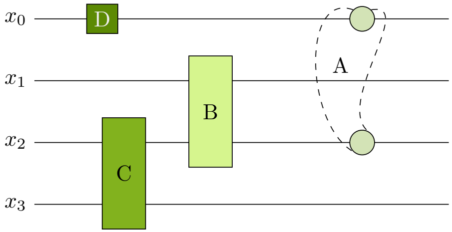


Operators here as before, inputs a and b, expression $b+a(a-\sqrt{a})$. The 20 character AST string **+b*a-aQa**babbabbbabab has a coding region (the first 8) shown pictorially as an AST. The last 12 are non coding. [Ferreira, 2006]

Quantum Circuits in GEP

Example of a 4-bit quantum circuit [Alvarez et al., 2023]

QC Praticitioner's representation



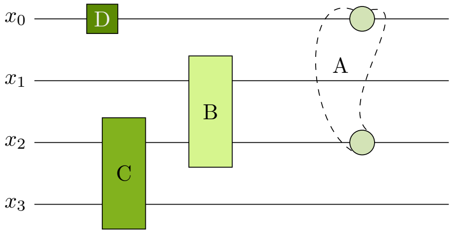
Every gate takes one input and produces one output

The final output is $A_{0,2}B_{1,2}C_{2,3}D_0\psi_0$

Quantum Circuits in GEP

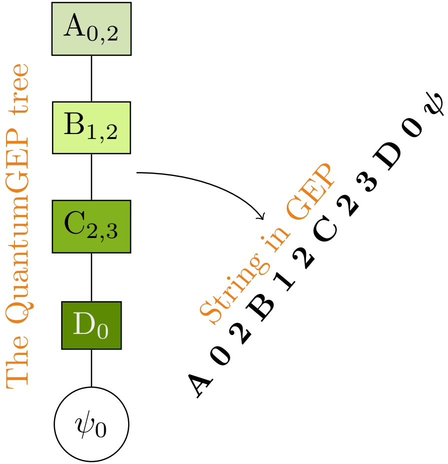
Example of a 4-bit quantum circuit [Alvarez et al., 2023]

QC Praticioner's representation



Every gate takes one input and produces one output

The final output is $A_{0,2}B_{1,2}C_{2,3}D_0\psi_0$



QuantumGEP: The Main Algorithm

1. Start with an original population of M circuits generated randomly
2. By gene expression programming mutate and combine the existing size M population, to generate $M' = 2M$ new circuits $\{\mathcal{C}(\varphi)\}_{0\leq j<M'}$ [1]

[1] These circuits may depend on K continuous variables $\varphi \equiv \{\varphi_k\}_{0\leq k<K}$. For example, consider that the rotation gate may depend on the angle of rotation, and, in general, gates may depend on arbitrary parameters collectively called φ. [Alvarez et al., 2023]

QuantumGEP: The Main Algorithm

1. Start with an original population of M circuits generated randomly
2. By gene expression programming mutate and combine the existing size M population, to generate $M' = 2M$ new circuits $\{\mathcal{C}(\varphi)\}_{0\leq j<M'}$[1]
3. For every circuit j, define the $\varphi-$dependent state $|\psi_j(\varphi)\rangle \equiv \mathcal{C}_j(\varphi)|\psi_{in}\rangle$.

[1]These circuits may depend on K continuous variables $\varphi \equiv \{\varphi_k\}_{0\leq k<K}$. For example, consider that the rotation gate may depend on the angle of rotation, and, in general, gates may depend on arbitrary parameters collectively called φ. [Alvarez et al., 2023]

QuantumGEP: The Main Algorithm

1. Start with an original population of M circuits generated randomly
2. By gene expression programming mutate and combine the existing size M population, to generate $M' = 2M$ new circuits $\{\mathcal{C}(\varphi)\}_{0\leq j<M'}$[1]
3. For every circuit j, define the $\varphi-$dependent state $|\psi_j(\varphi)\rangle \equiv \mathcal{C}_j(\varphi)|\psi_{in}\rangle$.
4. For every circuit j, compute the pre-fitness function $P_j(\varphi) = -\langle\psi_j(\varphi)|H|\psi_j(\varphi)\rangle$ and find the φ where the maximum occurs; let's call it $\varphi_{\max}$.
5. For every circuit j, calculate its fitness $F_j \equiv P_j(\varphi_{\max})$.

[1] These circuits may depend on K continuous variables $\varphi \equiv \{\varphi_k\}_{0\leq k<K}$. For example, consider that the rotation gate may depend on the angle of rotation, and, in general, gates may depend on arbitrary parameters collectively called φ. [Alvarez et al., 2023]

QuantumGEP: The Main Algorithm

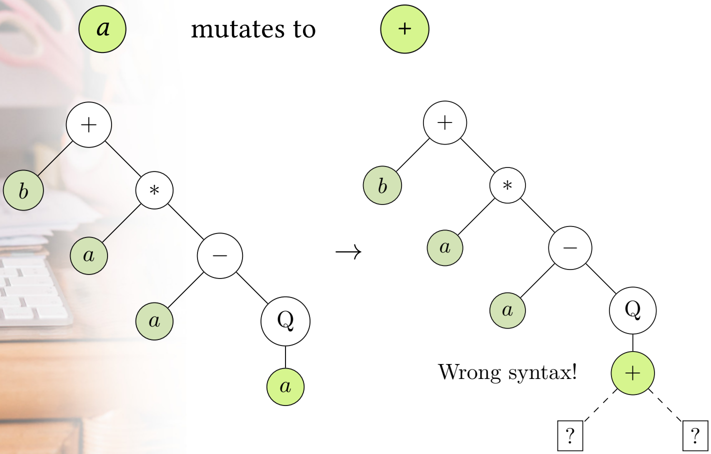
1. Start with an original population of M circuits generated randomly
2. By gene expression programming mutate and combine the existing size M population, to generate $M' = 2M$ new circuits $\{\mathcal{C}(\varphi)\}_{0\leq j<M'}$ [1]
3. For every circuit j, define the $\varphi-$dependent state $|\psi_j(\varphi)\rangle \equiv \mathcal{C}_j(\varphi)|\psi_{in}\rangle$.
4. For every circuit j, compute the pre-fitness function $P_j(\varphi) = -\langle\psi_j(\varphi)|H|\psi_j(\varphi)\rangle$ and find the φ where the maximum occurs; let's call it $\varphi_{\max}$.
5. For every circuit j, calculate its fitness $F_j \equiv P_j(\varphi_{\max})$.
6. Eliminate the M circuits with least fitness and keep the remaing M for the next step.
7. Go to step 2.

[1] These circuits may depend on K continuous variables $\varphi \equiv \{\varphi_k\}_{0\leq k<K}$. For example, consider that the rotation gate may depend on the angle of rotation, and, in general, gates may depend on arbitrary parameters collectively called φ. [Alvarez et al., 2023]

Mutations may create wrong trees

Photo by Sincerely Media
https://unsplash.com/photos/O9K5vvTurCY

Here the string is +b*a-aQa and



GEP: Unrestricted Mutations

String has fixed size, with a maximum size head. The tail contains only leaves. [Ferreira, 2001]

Yet the coding section size varies.

Example. Total Size = 20, Head Max Size=15, and **+b*a-aQa**babbabbbabab has 8 coding characters. **+b*a-aQ+ba**bbabbbabab has 10 coding characters.

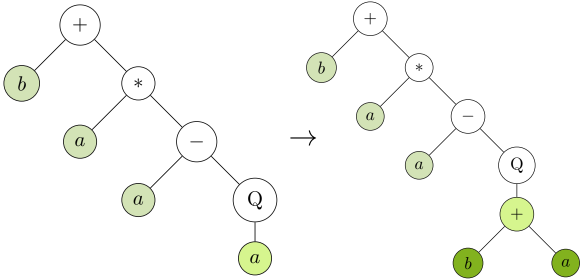
Mutations preserve the head and tail structure

Here *a* mutates to +

The syntax is still correct thanks to "junk DNA."

+b*a-aQ**a**babbabbbabab →

+b*a-aQ**+**babbabbbabab



Results in Condensed Matter

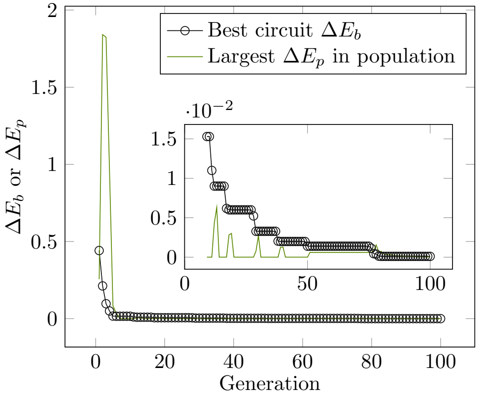
$H = J_x \sum_i \sigma_i^x \sigma_{i+1}^x$, σ^x is the Pauli x matrix

Aim: Finding the quantum circuit that produces the ground state of H from $|0000\rangle$

After a few generations we get many perfect quantum circuits

For example, with fitness 3.9971(6) we find:

$\mathrm{Ry}_0{:}3\pi/2\ \mathrm{Ry}_1{:}\pi/2\ \mathrm{Ry}_2{:}3\pi/2\ \mathrm{Ry}_3{:}\pi/2$



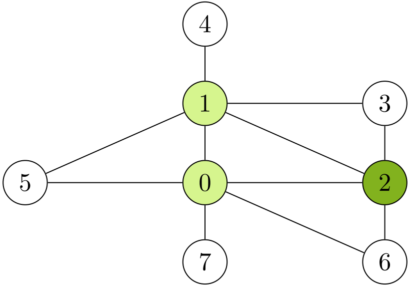
[1] Green line: energy difference between the ground state and that yielded by the best individual at that generation; open circles: largest energy difference within the population at that generation. Inset: magnification of the figure starting at generation eight. [Alvarez et al., 2023]

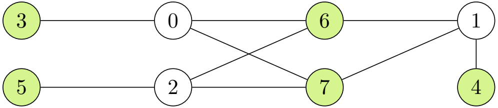
Input And Output

```
##Ainur1.0
#This tests GroundState for the XX model on a chain
HeadSize=4;
Population=60;
Generations=20;
NumberOfBits=4;
MinimizerTolerance=0.1;
Primitives="Ry,P";
MinimizerAlgorithm="Simplex";
RunType="GroundState";
Hamiltonian="xx";
HamiltonianCoupling=1;
#HamiltonianIsPeriodic=1;
InVectorFile="../TestSuite/inputs/vector10.txt";
```

```
./quantumGep -f ../TestSuite/inputs/input10.ain
Ground State Energy=-3
Ry2:4.445622188 Ry0:0.4704532758 Ry1:1.806751358 Ry0:6.017431122 0  fit 1.43069 0 1 2 3 4 5 6 7  #= 5
Ry0:3.224076586 Ry0:0.2626101643 Ry2:1.900779445 Ry3:4.425218811 0  fit 0.907306 0 1 4 5 8 9 12 13  #= 5
P0 Ry2:2.040787545 Ry1:4.424636695 0 0  fit 0.854915 0 2 4 6  #= 4
Ry2:4.453122188 Ry3:2.070684695 P2 P1 0  fit 0.848304 0 4 8 12  #= 5
...
Ry0:4.871224564 Ry1:1.642748165 Ry2:4.728122188 Ry3:1.706156449 0  fit 2.97288 0 1 2 3 4 5 6 7 8 9 10
11 12 13 14 15  #= 5
Ry0:4.871224564 Ry1:1.642748165 Ry2:4.728122188 Ry3:1.706156449 0  fit 2.97288 0 1 2 3 4 5 6 7 8 9 10
11 12 13 14 15  #= 5
```

Results for the Max-Cut Problem





A solution is given by 248, (binary) 11111000: sites 3 to 7 have down spins and form the maximal cut; energy equals -9, fitness 9.

A solution is given by 3 or 7. The number, 3, is binary 00000011: sites 0, 1, have down spins and form the maximal cut; The other solution is 7, (binary) 00000111: sites 0, 1, and 2 form the maximal cut. Vertices 0, 1, and 2 have been shaded, with vertex 2 shaded more to indicate that it appears only in the 2nd solution. In all cases energy is -5, fitness 5.[2]

[2]Both graphs in [Alvarez et al., 2023] appear also in [Lotshaw et al., 2021]

Advanced Features in GEP

(1) Multiple genes

(2) Numerical constants

[2] Adapted from Figures 5.2 and 3.13 in [Ferreira, 2006].

Advanced Features in GEP

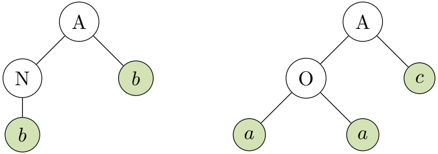
(1) Multiple genes

Logical primitives: {And, Not, Or}

Leaves: {a, b, c}; a,b,c$\in$ {true, false}

0123456 0123456

ANbbabcAOcaabc



First Gene Second Gene

(2) Numerical constants

[2] Adapted from Figures 5.2 and 3.13 in [Ferreira, 2006].

Advanced Features in GEP

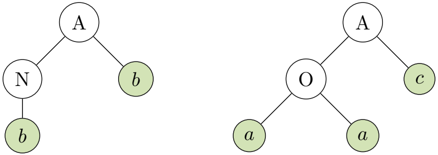
(1) Multiple genes

Logical primitives: {And, Not, Or}

Leaves: {a, b, c}; a,b,c$\in$ {true, false}

0123456 0123456

ANbbabcAOcaabc



First Gene Second Gene

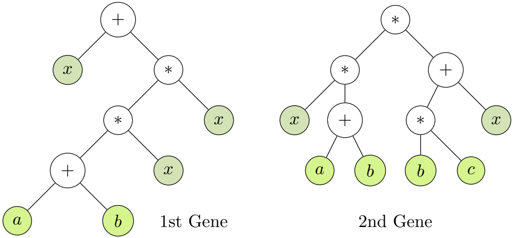
(2) Numerical constants

Primitives: arithm. set; leaf: x

Constants: a=0.298, b=1.083, c=1.466

0123456789012 0123456789012

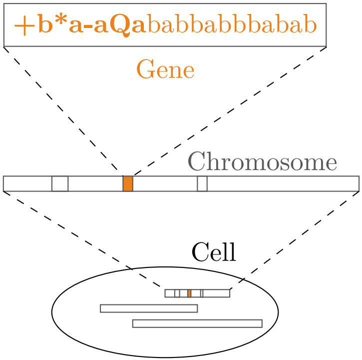
+x**x+xabbcae**+x+*xabbcae



[2] Adapted from Figures 5.2 and 3.13 in [Ferreira, 2006].

Automatically Defined Functions

- Multiple genes form a chromosome
  And multiple chromosomes form a cell
- A cell may produce multiple outputs,
  Or may be controlled by a single output
- ADFs then create a hierarchy



Summary and Outlook

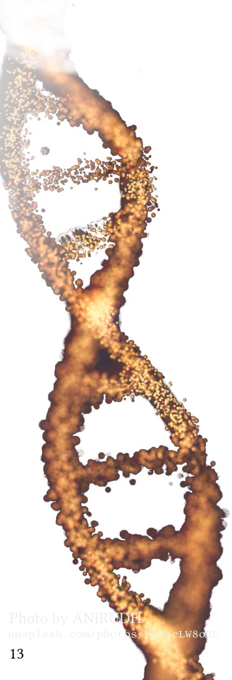


Photo by ANIRUDH
unsplash.com/photos/YQYacLW8o2U

QuantumGEP...

- yields thousands of quantum circuits for each problem
  And they aren't parametric
- shows success in condensed matter and graph theory
  And may work in other domain sciences; arxiv.2303.08203
- has a free and open source license
  And development happens in the open

code.ornl.gov/gonzalo_3/evendim
github.com/g1257/evendim

Credits

This work was performed at Oak Ridge National Laboratory, operated by UT-Battelle, LLC under contract DE-AC05-00OR22725 for the US Department of Energy (DOE). Support for the work came from the DOE Advanced Scientific Computing Research (ASCR) Accelerated Research in Quantum Computing (ARQC) Program under field work proposal ERKJ354.

Produced with LaTeX and the Beamer package
Tikz was used for some figures.

Photo by Jake Belcher; MIT; `https://news.mit.edu/2022/alex-greene-quantum-computers-1013`.

Photo by Sincerely Media; `unsplash.com/photos/O9K5vvTurCY`. Photo modified from ANIRUDH `unsplash.com/photos/YQYacLW8o2U`.

References

Alvarez, G., Bennink, R., Irle, S., and Jakowski, J. (2023). **Gene expression programming for quantum computing. arxiv.2303.08203.**

Ferreira, C. (2001). Gene expression programming: A new adaptive algorithm for solving problems. *Complex Systems*, 13:87–129.

Ferreira, C. (2006). *Gene Expression Programming, Mathematical Modeling by an Artificial Intelligence*. Springer Verlag, Berlin, Heidelberg, 2nd ed. edition.

Lotshaw, P. C., Humble, T. S., Herrman, R., Ostrowski, J., and Siopsis, G. (2021). Empirical performance bounds for quantum approximate optimization. *Quantum Information Processing*, 20(12).