Yury Chernyak

Supervisor:
Martin Plesch

Resource Efficient Non-Gradient Optimization Methods for Variational Quantum Eigensolver

September 26th, 2024

Minimal Thesis

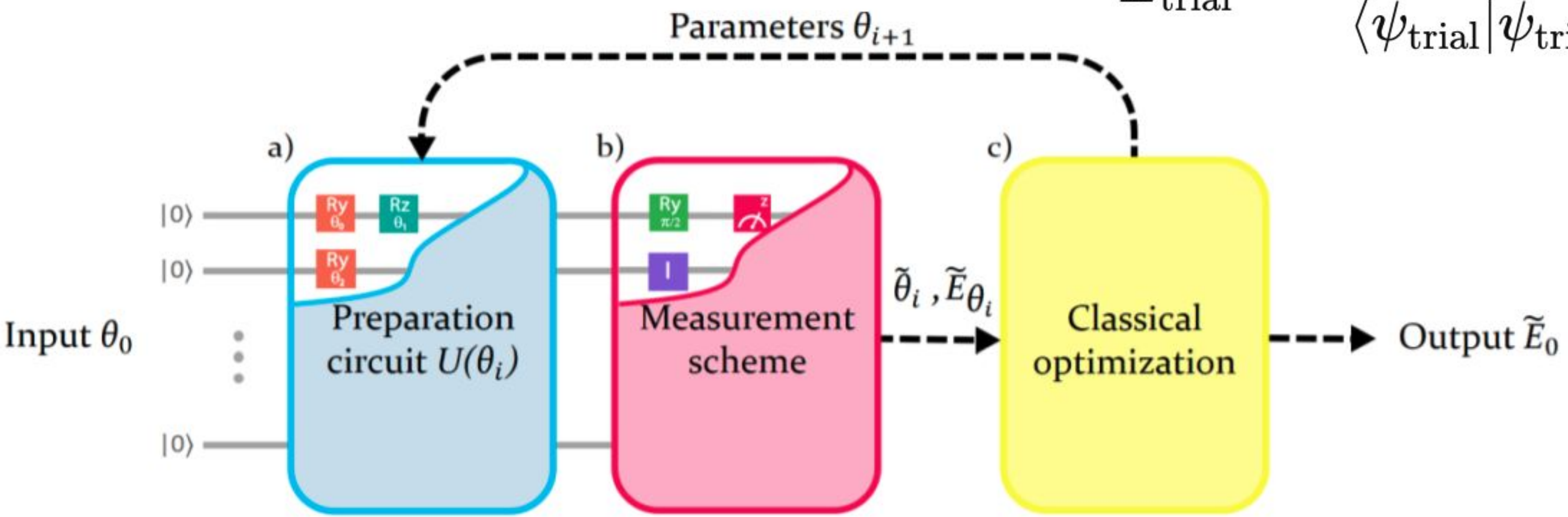
PART 1: Research

- VQE
- Classical Optimization method studied: PSO
- Newly developed Optimization method: HOPSO
- Results & Outcome
- Plans & Future Outlook

VQE (1/2)

Variational Principle

$$E_{\text{trial}} = \frac{\langle \psi_{\text{trial}} | \hat{H} | \psi_{\text{trial}} \rangle}{\langle \psi_{\text{trial}} | \psi_{\text{trial}} \rangle} \geq E_0$$



$$E_0 \leq \langle \psi_{\text{trial}} | \hat{H} | \psi_{\text{trial}} \rangle$$

$$f(\vec{\theta}) = \langle \psi(\vec{\theta}) | H | \psi(\vec{\theta}) \rangle$$

The scheme for

finding the ground state energy of a system (lowest possible value),

→ by updating trial wave functions

→ by updating their parameters

$$\vec{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{pmatrix} \qquad U(\theta)|\psi_0\rangle = |\psi(\theta)\rangle$$

Steps of the VQE I

1. **Hamiltonian Construction**
   a. Hamiltonian = quantum operator representing total energy of a system
   b. Molecular Hamiltonian = total energy of molecular system defined by its atomic configuration
      – *geometry of system* gives the *electronic wavefunction* & ***electronic hamiltonian***

Molecular Hamiltonian

$$H = T_e + V_{ne} + V_{ee} + T_n + V_{nn}$$

First Order Quantization (Electronic Hamiltonian) →

$$H_{\text{electronic}} = T_e + V_{ne} + V_{ee}$$

$$\hat{T}_e = -\sum_i \frac{\hbar^2}{2M_i}\nabla_i^2, \quad \hat{V}_{ne} = -\sum_{i,k} \frac{e^2}{4\pi\epsilon_0}\frac{Z_k}{|\mathbf{r}_i - \mathbf{R}_k|}, \quad \hat{V}_{ee} = \sum_{i<j} \frac{e^2}{4\pi\epsilon_0}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}.$$

Second Order Quantization (Electronic Hamiltonian)

$$\hat{H} = \sum_{ij} t_{ij} a_i^\dagger a_j + \frac{1}{2}\sum_{ijkl} V_{ijkl} a_i^\dagger a_j^\dagger a_k a_l$$

Jordan-Wigner Transformation → **Bravyi-Kitaev Transformation**

Qubit Hamiltoni

$$\hat{H}_{\text{qubit}} = \sum_k \alpha_k P_k$$

creates an electron in orbital *i* — $a_i = \frac{1}{2}\left(\prod_{j=1}^{i-1} Z_j\right)(X_i - iY_i)$

annihilates an electron in orbital *i* — $a_i^\dagger = \frac{1}{2}\left(\prod_{j=1}^{i-1} Z_j\right)(X_i + iY_i)$

2. **Operator Encoding**
   a. **JW or BK Transformation for Qubit-Hamiltonian**

X_i, Y_i, Z_i Pauli operators acting on *i-th* qubit

Steps of the VQE II

3. Measurement Strategy

   a. Group Pauli terms in Hamiltonian (by exploiting commutative properties)
   To reduce the number of quantum circuit measurements (shots) (save computational resources)

4. Ansatz Implementation

   b. Hardware Efficient Ansatz

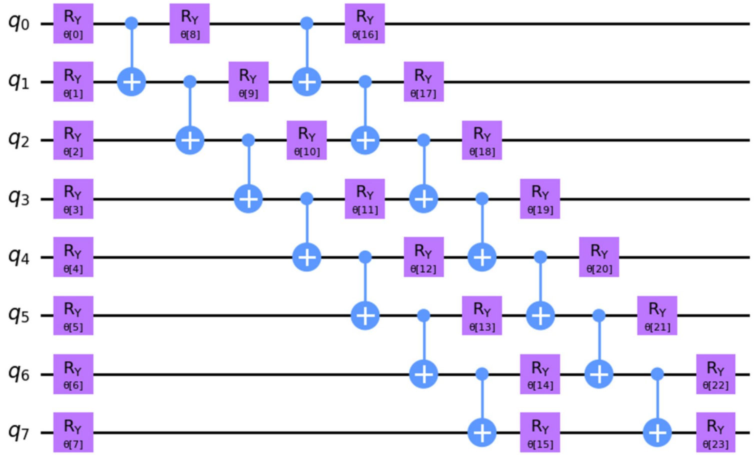
5. Parameter Optimization

   c. Classical Optimizer

      i. *Particle Swarm Optimization*

      ii. *Harmonic Oscillator Particle Swarm Optimization*

6. Error Mitigation



$$|\psi(\Theta)\rangle = \left[\prod_{i=1}^{d} U_{\text{rota}}(\theta_i) \times U_{\text{ent}}\right] \times U_{\text{rota}}(\theta_{d+1})|\psi_{\text{init}}\rangle$$

$$U_{\text{rota}}(\theta_i) = \prod_{q,p} R_p^q(\theta_p^q)$$

$p \in \{X, Y, Z\}$

q is qubit

Details

Ansatz:

- Hardware Efficient Ansatz
- 4 layers, 40 terms
- Linearly Entangled

Number of qubits: 8

Hamiltonian: LiH

Classical Optimizer:

- PSO, HOPSO

Transformation:
Jordan Wigner mapping

D-dimensional parameter search space bounded by [−π,π].

VQE *(Example)*

0. Hamiltonian

$$\hat{H} = \sum_i c_i \hat{P}_i$$

Product of Pauli Matrices $(\hat{I}, \hat{X}, \hat{Y}, \hat{Z})$

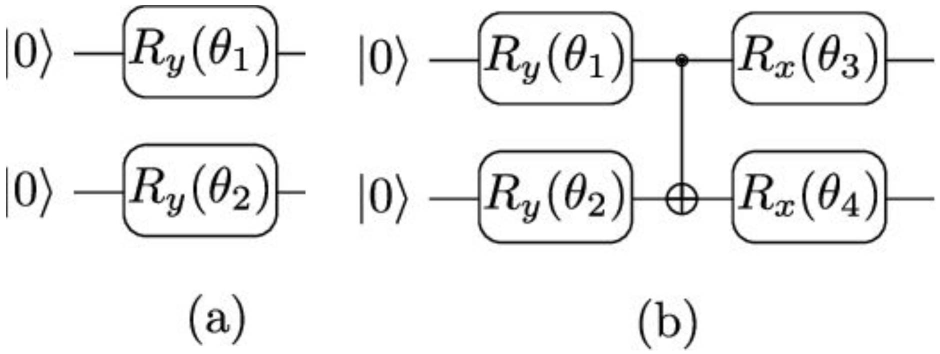
Example:

$$\hat{Y}_1 \hat{X}_2 \hat{Z}_3$$

Example:

$$\hat{H} = c_1 \hat{Z}_1 + c_2 \hat{X}_1 \hat{X}_2$$

1. Ansatz



Initialize:

$$|\psi_0\rangle = |00\rangle$$

$$\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$$

Prepare:

$$|\psi(\theta)\rangle = U(\theta)|\psi_0\rangle$$

Example:

$$|\psi(\theta)\rangle = R_y(\theta_1) \otimes R_y(\theta_2)|00\rangle$$

$$|\psi(\theta)\rangle = R_y(\theta_3) \otimes R_y(\theta_4)\, \mathrm{CNOT}_{1,2}\, (R_y(\theta_1) \otimes R_y(\theta_2)|00\rangle)$$

2. Measurement

$$E(\theta) = \langle \psi(\theta)|\hat{H}|\psi(\theta)\rangle$$

Example:

$$\langle \psi(\theta)|\hat{Z}_1|\psi(\theta)\rangle + \langle \psi(\theta)|\hat{X}_1 \hat{X}_2|\psi(\theta)\rangle$$

$$E(\theta) = c_1 \langle \hat{Z}_1 \rangle + c_2 \langle \hat{X}_1 \hat{X}_2 \rangle$$

$$E(\theta) = \sum_i c_i \langle \psi(\theta)|\hat{P}_i|\psi(\theta)\rangle$$

3. Optimize

Classical Optimization Method

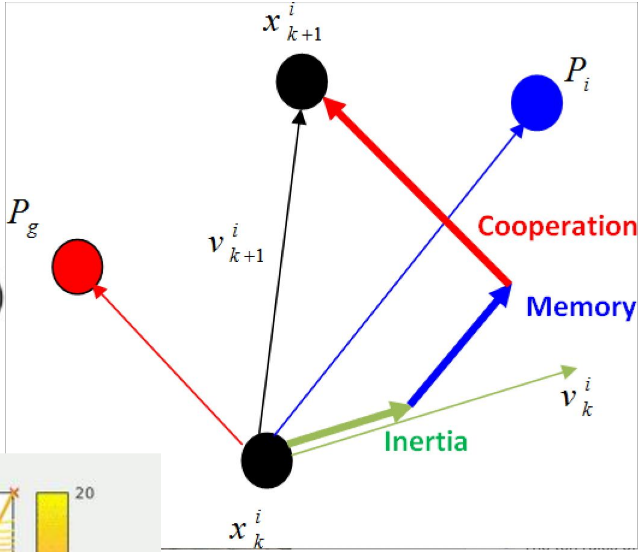
Gradient vs. Non-Gradient

Particle Swarm Optimization Review

Update Equations:

Velocity Update:

$$v_{i,d}(t+1) = \chi\left(wv_{i,d}(t) + c_1r_1(p_{i,d} - x_{i,d}(t)) + c_2r_2(g_d - x_{i,d}(t))\right)$$



Position Update:

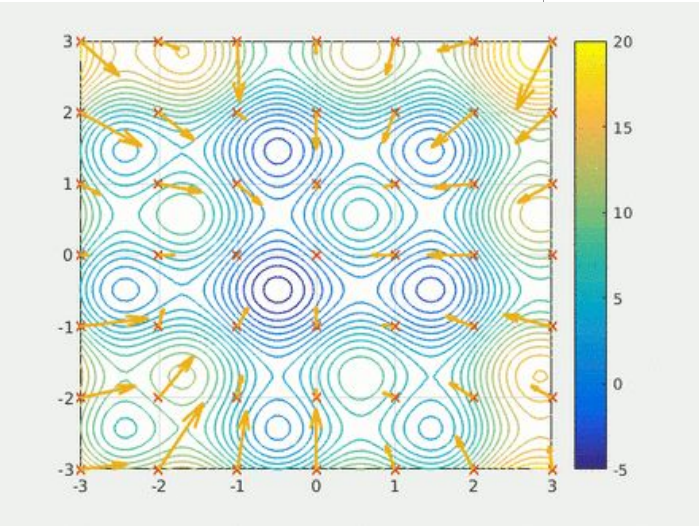
$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1)$$

Constrictor Factor:

$$\chi = \frac{2}{\left|2-\phi-\sqrt{\phi^2-4\phi}\right|}$$

where

$$\phi = c_1 + c_2$$



Harmonic Oscillator PSO

- **HOPSO** introduces harmonic oscillations to model the movement of particles
- Particle's position and velocity evolve like a **damped harmonic oscillator** (each particle's movement governed by oscillations around a central attractor point) – dimension independent oscillators
- Maintains energy conservation within the system → *eliminating explosions*
- *More fine-tunable*
- Each particle's position is influenced by both **amplitude** and **phase** (amplitude decays exponentially)
- After every iteration, the phase is recalculated using: $$\theta = \arccos\left(\frac{x(0) - a}{A_0}\right)$$
- The attractor is the equilibrium position for each particle, calculated as:

p_{ij} Personal best position

g_j Global best position

c_1, c_2 Cognitive & Social Coefficients (attraction weights)

$$a_{ij} = \frac{c_1 p_{ij} + c_2 g_j}{c_1 + c_2}$$

Algorithm 1: Harmonic Oscillator based Particle Swarm Optimization (HOPSO)

Input: Problem dimensions, Objective function, Algorithm parameters
Output: Candidate Optimal solution
Set constants c_1, c_2, λ, m for attraction weights and damping and minimal amplitude ration;
Initialize particles with random positions $x_{i,d}$ and velocities $v_{i,d}$;
Set initial personal best positions p_i by starting positions for each particle;
Choose initial global best position g;
Calculate position of attractors: $a_i = \frac{c_1 p_i + c_2 g}{c_1 + c_2}$;
Calculate initial amplitude:
$A_0 = \sqrt{(x(0) - a)^2 + \left(\frac{v(0) + \lambda(x(0) - a)}{\omega}\right)^2}$;
Calculate initial phase: $\theta = \arccos\left(\frac{x(0) - a}{A_0}\right)$;
while *iteration* $<$ *max_iterations* **do**
  foreach *particle* **do**
    foreach *dimension* **do**
      $A = \max(A_0 e^{-\lambda t}, \frac{|p_i - g|}{2} \cdot m)$;
      $x(t) = A_0 e^{-\lambda t} \cos(\omega t + \theta)$;
      $v(t) = -\omega(A_0 e^{-\lambda t} \sin(\omega t + \theta)) - \lambda x(t)$;
  foreach *particle* **do**
    Calculate Cost function from positions;
    if $Cost\_function(x_{i,d}(t)) < Cost\_function(p_i)$ **then**
      Update p_i, best value, time, attractors, amplitude, phase;
  if *personal best value* $<$ *global best energy* **then**
    Update global best value and g;
    foreach *particle and dimension* **do**
      Reset time, recalculate attractors, amplitude, phase;
  iteration ← iteration + 1;

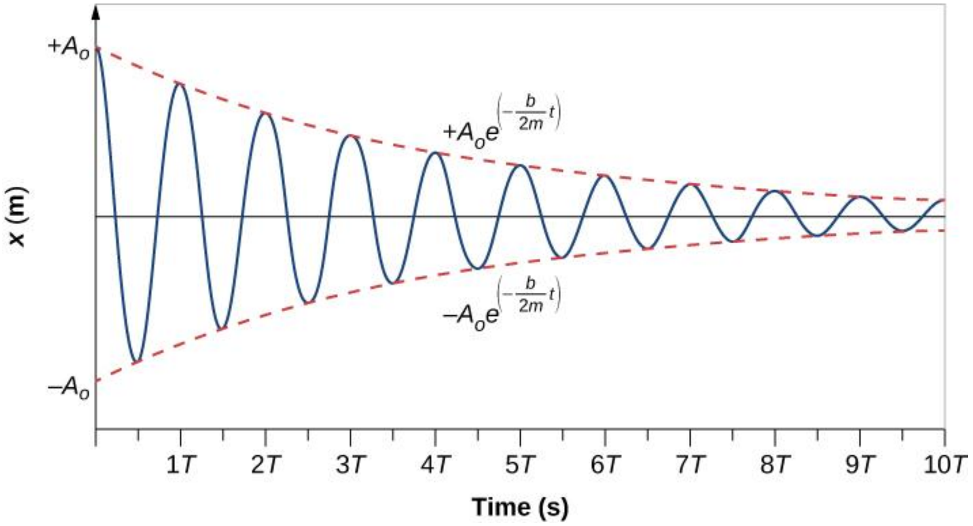
Introducing Initial Position, Velocity, and Parameters

Consider each particle's position in each dimension to be a solution to a damped-harmonic oscillator.

$$x(t) = A_0 e^{-\lambda t} \cos(\omega t + \theta)$$

Taking derivative of this yields velocity:

$$v(t) = -\omega(A_0 e^{-\lambda t} \sin(\omega t + \theta)) - \lambda(x(t))$$

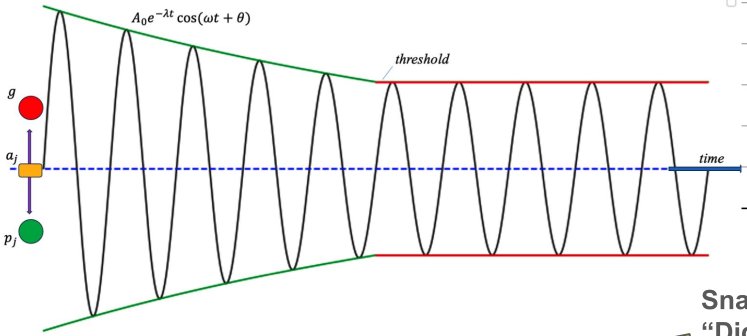


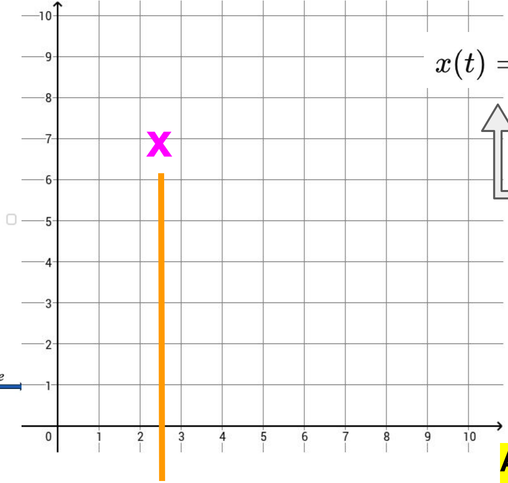
$$x(t), A_0, \lambda, \omega\ \theta$$

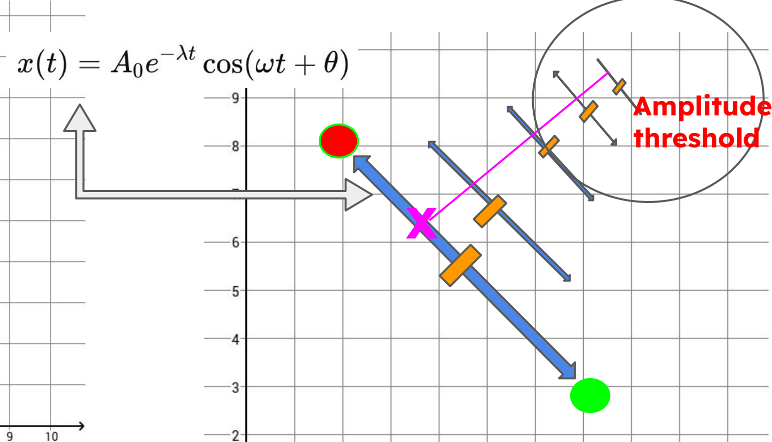
- **position of the particle at time *t***
- **initial amplitude**
- **damping factor**
- **angular frequency**
- **initial phase of the oscillation**

<u>***Visualization I***</u>

One-Dimension







After each iteration, we take a "snapshot"

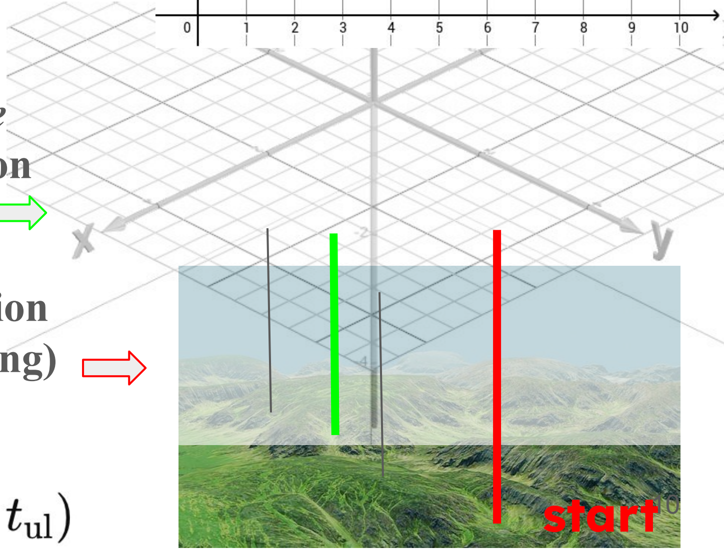
Snapshot: plug into cost function – "Did I find a better spot?"

Outcomes:

1. No, Better Position Found (No Improvement)
   a. no update is made, attractor stays the same
2. Yes, Personal Best found
   a. PB changes ⇒ recalculate attractor
3. Yes, Personal best + <u>Global best</u> found
   a. PB = GB ⇒ attractor = *g* position

Parameter Space (theta's = position coordinates)

E (Cost function output mapping)



Randomness in the snapshot:

$$t(i+1) = t(i) + \text{rand}(0, t_{\text{ul}})$$

continue

continued…

1. **No, Better Position Found (No Improvement)**
   a. **no update is made, attractor stays the same**

Continues to oscillate and damp as:

$$A(t) = A_0 e^{-\lambda t}$$

2. **Yes, Personal Best found**
   a. **PB changes ⇒ reset time**
   Current position, velocity
   Recalculate attractor,
   Recalculate amplitude, phase

3. **Yes, Personal best + <u>Global best</u> found**
   a. **PB = GB ⇒ attractor = *g* position**
   b. **Reset time**
   c. **Recalculate**
   attractor, amplitude, phase

For each dimension separately,

$$a_j = \frac{c_1 p_j + c_2 g}{c_1 + c_2}$$

The current position = initial position

The current velocity = initial velocity

$$A_0 = \sqrt{(x(0) - a_j)^2 + \left(\frac{v(0) + \lambda(x(0) - a_j)}{\omega}\right)^2}$$

$$\theta = \arccos\left(\frac{x(0) - a_j}{A_0}\right)$$

Recalculated amplitude

Previous amplitude

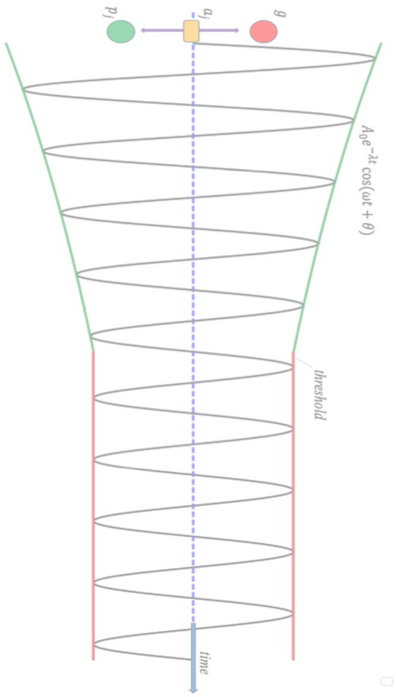
Amplitude Threshold

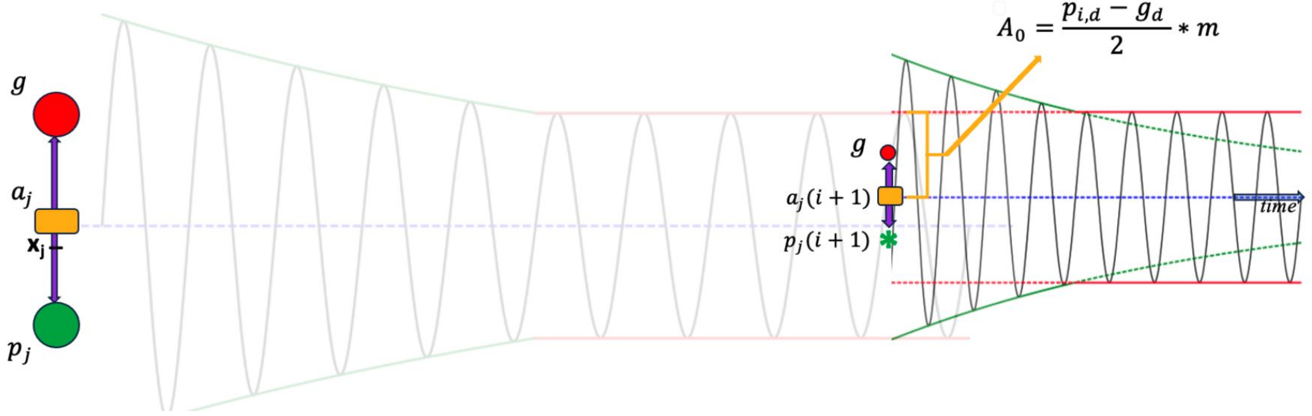
Condition: *Select greatest one*

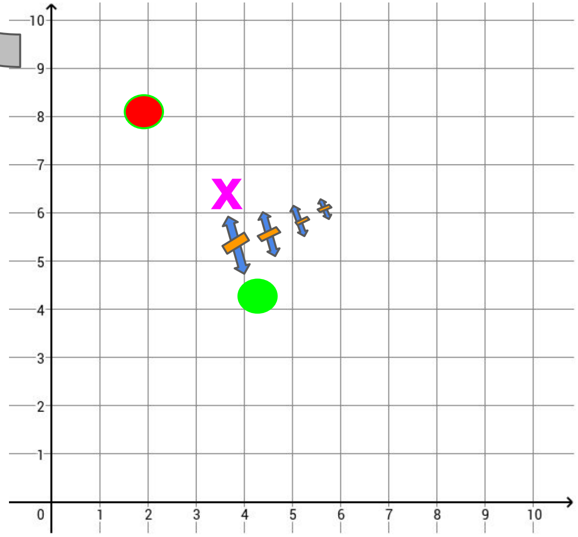
$$(A_0)_{i+1} = \max\left((A)_i, (A_0)_{i+1}, \frac{|p_{j,d} - g_d|}{2} \cdot m\right)$$

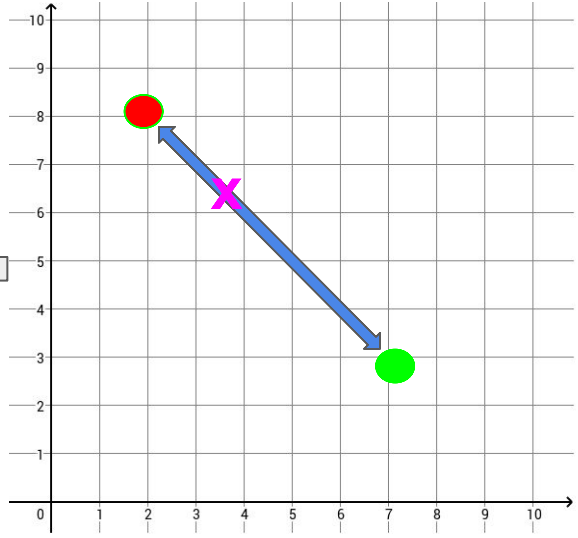
Visualization II

Reason for Amplitude Threshold









RESULTS I

https://www.sfu.ca/~ssurjano/optimization.html

Visualization of Commonly-used Test Functions

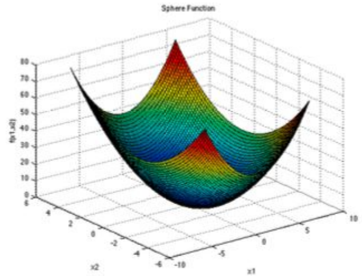


Figure 5.1: Sphere Function

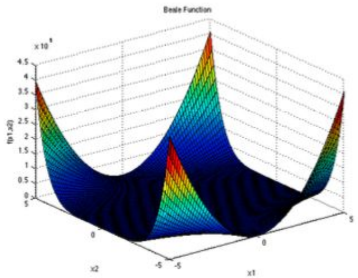


Figure 5.2: Beale Function

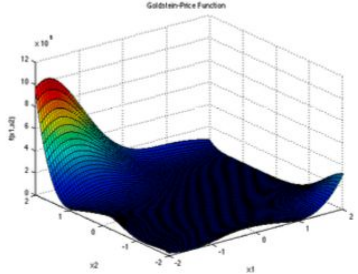


Figure 5.3: Goldstein-Price Function

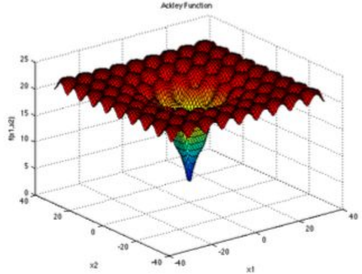


Figure 5.4: Ackley Function

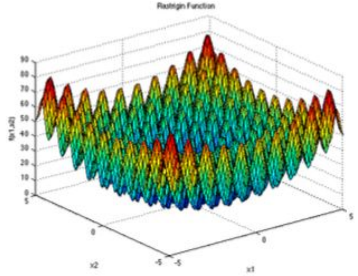


Figure 5.5: Rastrigin Function

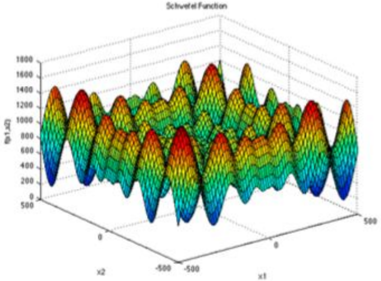


Figure 5.6: Schwefel Function

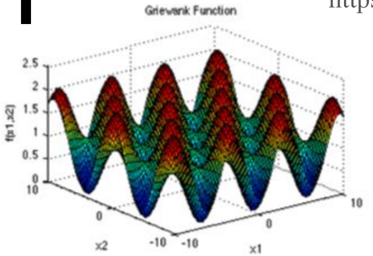


Figure 5.7: Griewank Function

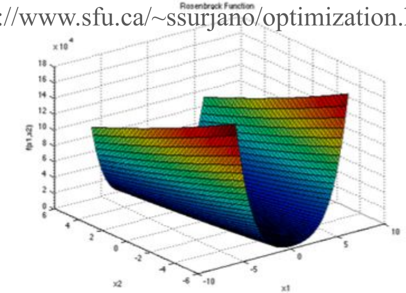


Figure 5.8: Rosenbrock Function

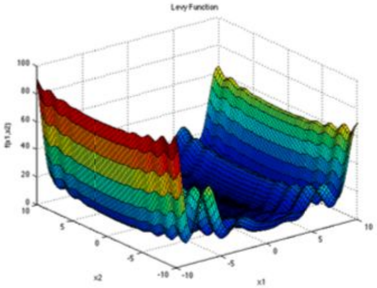


Figure 5.9: Levy Function

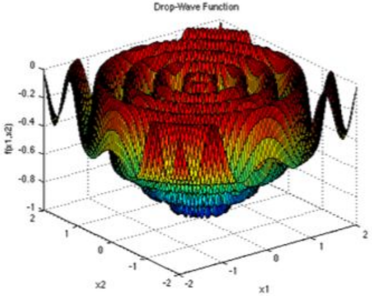


Figure 5.10: Drop-Wave Function

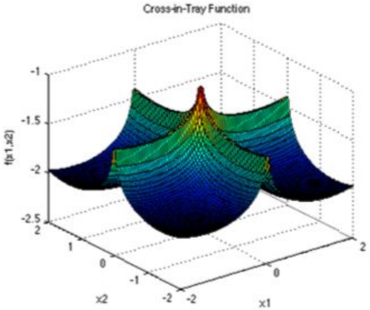


Figure 5.11: Cross-Tray Function

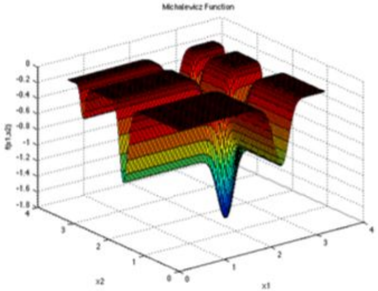
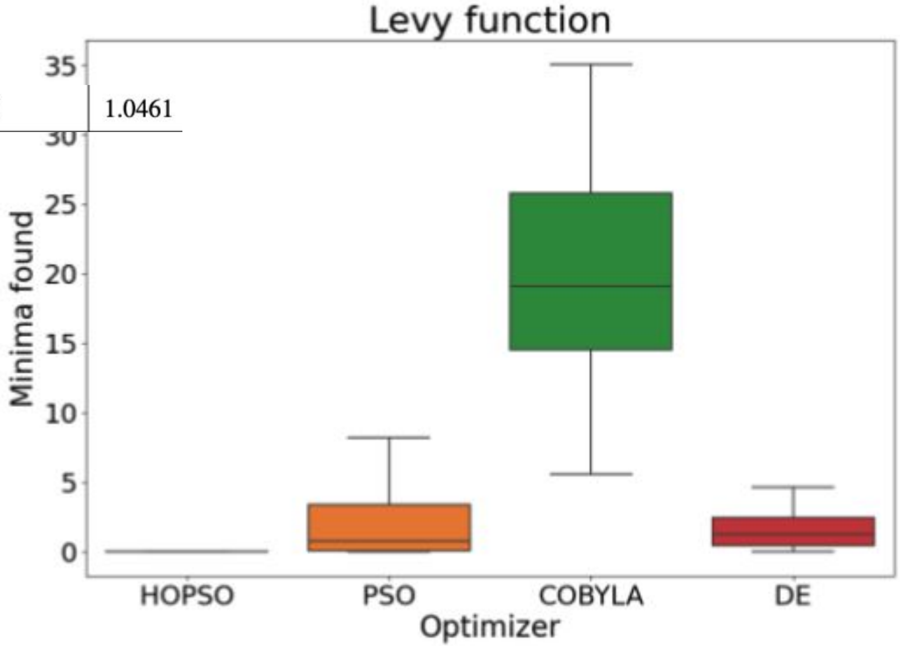


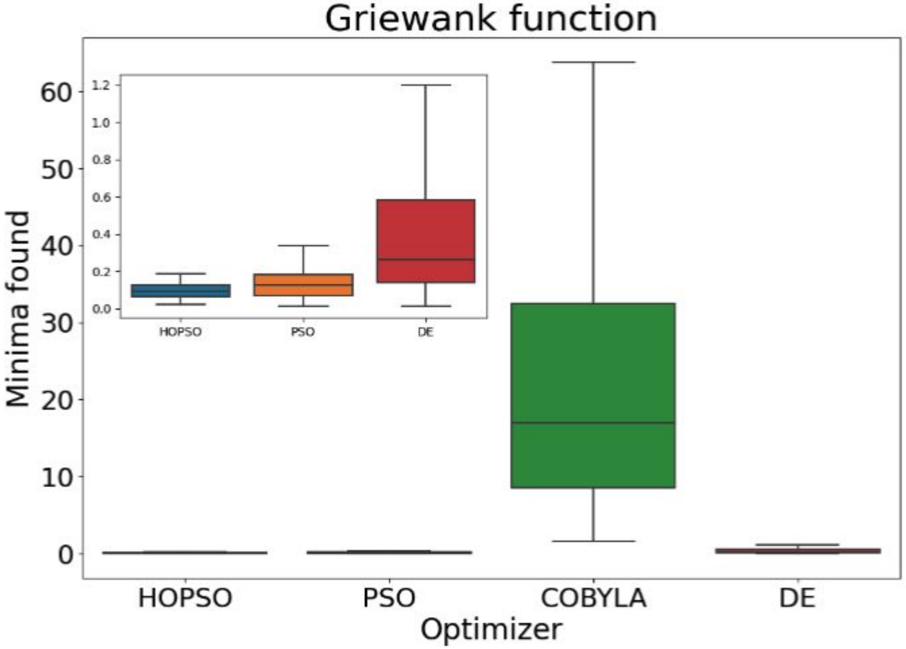
Figure 5.12: Michalewicz Function

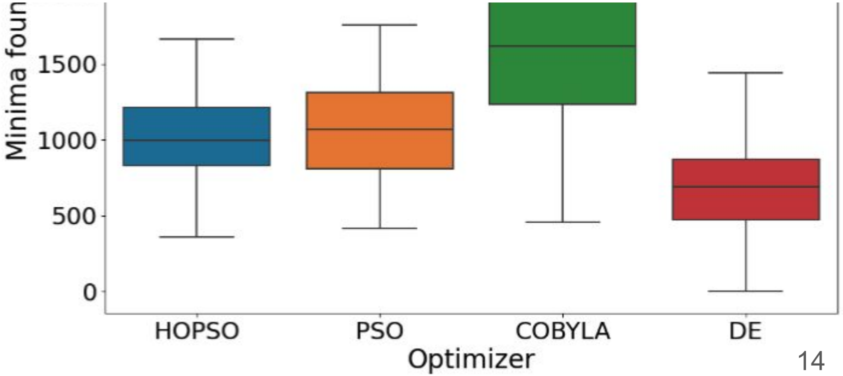
Results

mean

0.1033	0.1471	21.920	1.0461

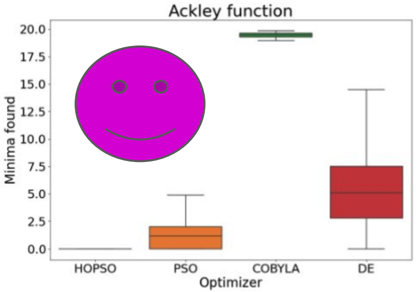




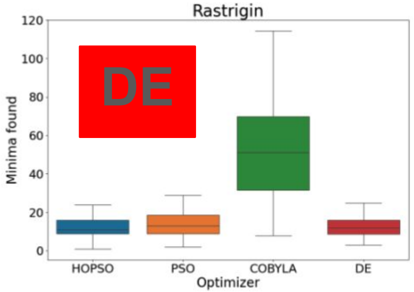


mean

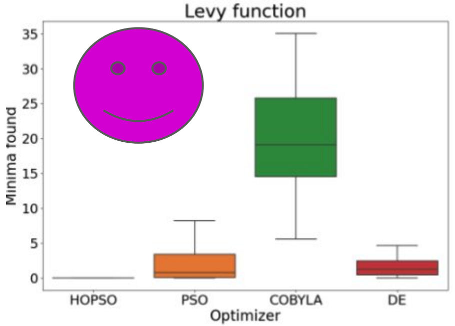
0.1033	0.1471	21.920	1.0461

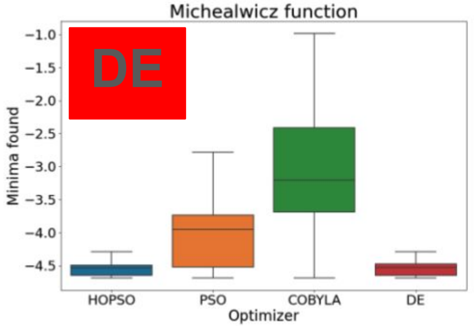


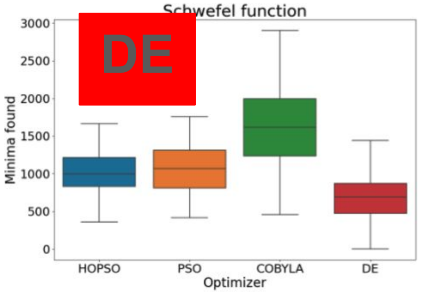
(a)

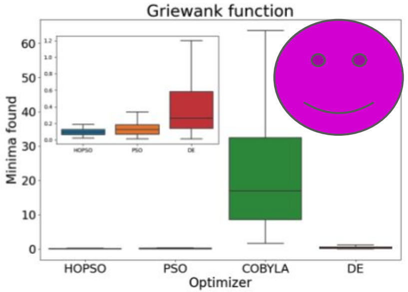


(b)

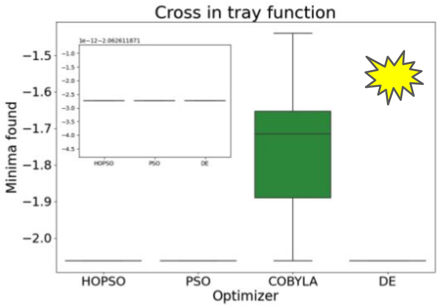




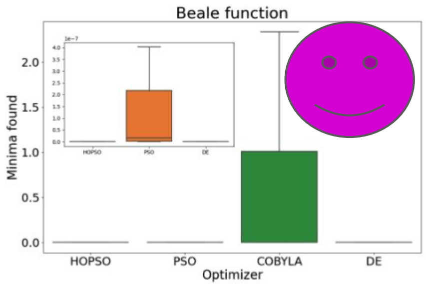




(d)

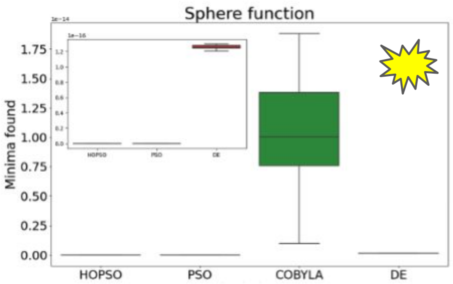


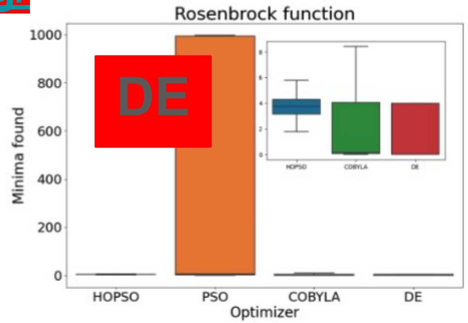
(a)

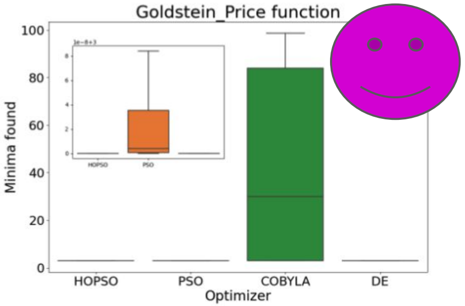


(b)

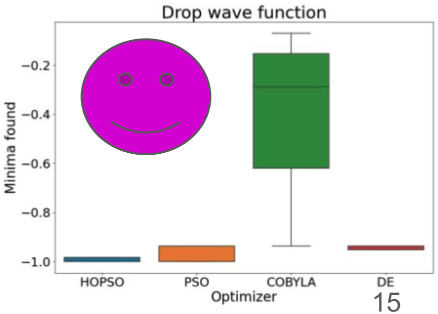
Arxiv: https://arxiv.org/







(c)



(d)

Citations

Tilly, J., Chen, H., Cao, S., Picozzi, D., Setia, K., Li, Y., Grant, E., Wossnig, L., Rungger, I., Booth, G. H., & Tennyson, J. (2022). The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, *986*, 1-128. https://doi.org/10.1016/j.physrep.2022.07.003

Waleed, Syed & Ullah, Inam & Khan, Wali Ullah & Khan, Ullah & Ateeq, · & Rehman, Ateeq & Rahman, Taj & Li, Shanbin. (2021). Resource allocation of 5G network by exploiting particle swarm optimization. Iran Journal of Computer Science. 4. 10.1007/s42044-021-00091-5.

https://www.fostshop.com/?ggcid=932830

https://en.wikipedia.org/wiki/Variational_quantum_eigensolver

Bonyadi, M. R.; Michalewicz, Z. (2017). "Particle swarm optimization for single objective continuous space problems: a review". *Evolutionary Computation*. **25** (1): 1–54. doi:10.1162/EVCO_r_00180. PMID 26953883. S2CID 8783143.

Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". *Proceedings of IEEE International Conference on Neural Networks*. Vol. IV. pp. 1942–1948. doi:10.1109/ICNN.1995.488968.

Shi, Y.; Eberhart, R.C. (1998). "A modified particle swarm optimizer". *Proceedings of IEEE International Conference on Evolutionary Computation*. pp. 69–73. doi:10.1109/ICEC.1998.699146.

Poli, R. (2008). "Analysis of the publications on the applications of particle swarm optimisation" (PDF). *Journal of Artificial Evolution and Applications*. **2008**: 1–10. doi:10.1155/2008/685175.

Clerc, M.; Kennedy, J. (2002). "The particle swarm - explosion, stability, and convergence in a multidimensional complex space". *IEEE Transactions on Evolutionary Computation*. **6** (1): 58–73. CiteSeerX 10.1.1.460.6608. doi:10.1109/4235.985692.