Quantum Gate Synthesis – SQUANDER
- Midterm report -

Balázs Menkó
Supervisor: Péter Rakyta

Eötvös Loránd University

March 26, 2025

Optimization process

In my last report I mentioned that I run a few optimization process with SQUANDER package and a few introduced gradient-free methods. In the following I would like to show the results. These result and source codes are also available on my GitHub page: github.com/menkobalazs/SMC-Lab-SQUANDER as a forked version of the original SQUANDER package

Simulations with 10 qubits

As a first step I started the optimization process with ten qubits and one hundred of quantum circuit layers. As I showed before there is an option to start the simulations with a zero parameter vector. The real part of the first eigenvalue of the Hamiltonian is the minimal value (or the ground state energy) of the system. It is visible with a dashed black line. In this simulation most of optimizer methods failed. In the upper left corner of Fig. 1 the trajectories does not start to go to minima, it means that the optimization process could not work due to the initial conditions. The slope of the functions shows the efficiency of the corresponding method. It means the ground state energy can be approachable under fewer cost function evaluation then in other cases. Here the *Powell* method approached the ground state energy.

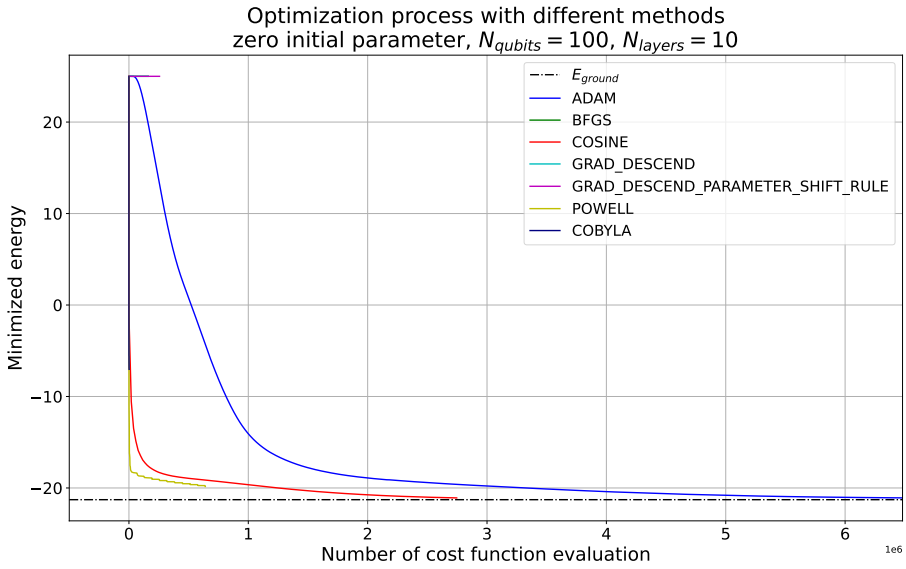

Figure 1: Minimization of cost function with zero initial parameters.

If we zoom in to the first 10% of the figure (see Fig. 2.) the *Cobyla* method is getting visible with its navy blue shade. The simulations started successfully,

but the inner iteration was not enough to approach the minima. The third gradient-free (*Nelder-Mead*) method is not visible on the figure. The reason is that the saving method do not write the data to file, except the first iteration. As a next step I will try to debug it.

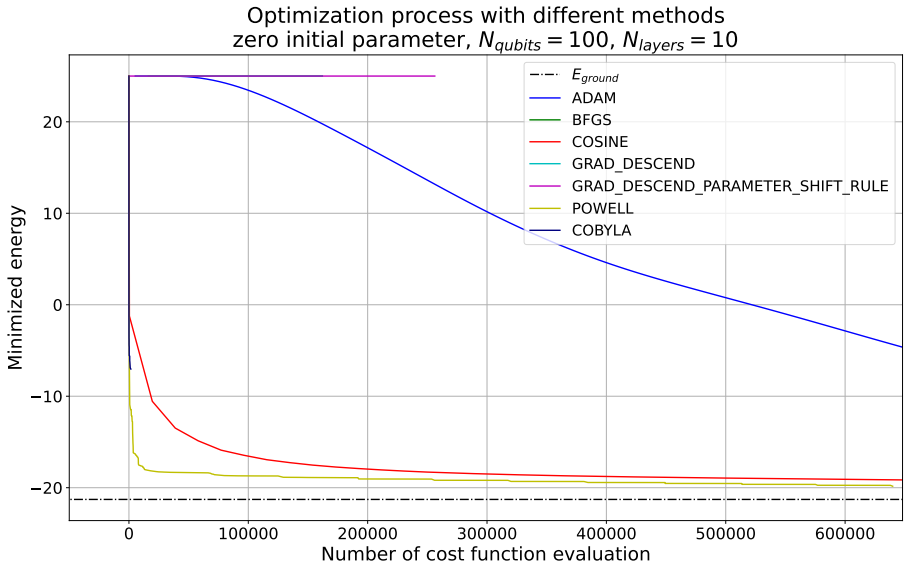

Figure 2: Minimization of cost function with zero initial parameters. Zoomed to the first 10% of Fig. 1.

The gradient-free method was simulated also with random initial parameter and it is visible on Fig. 3. Now the every method could start an optimization process, they did not stuck in the initial energy. In this case the *Powell* method also beats the *Cosine* method but if we zoom into the first 10% again it appears that it is not correct.

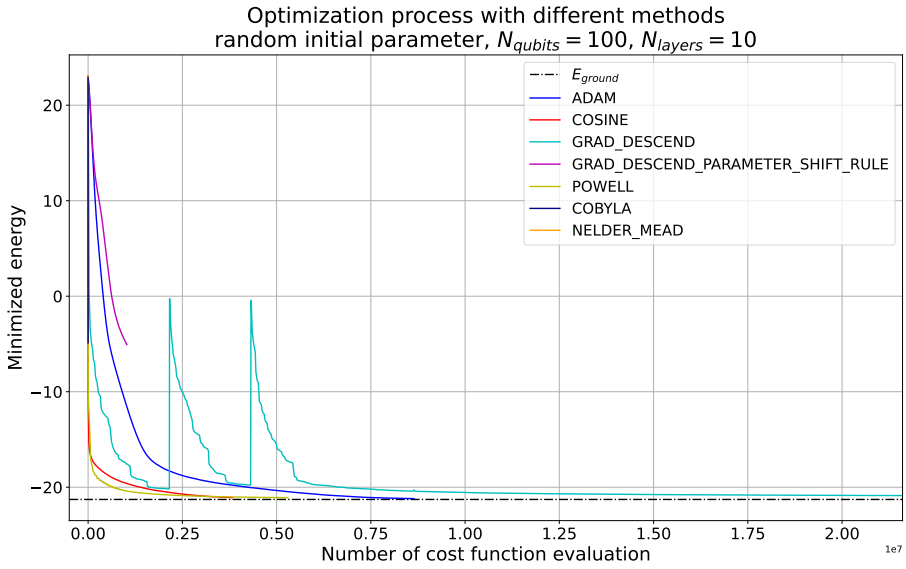

Figure 3: Minimization of cost function with random initial parameters.

On Fig. 4. it is visible that the *Powell* method got stuck for a while, but after it gets closer to the ground state energy.

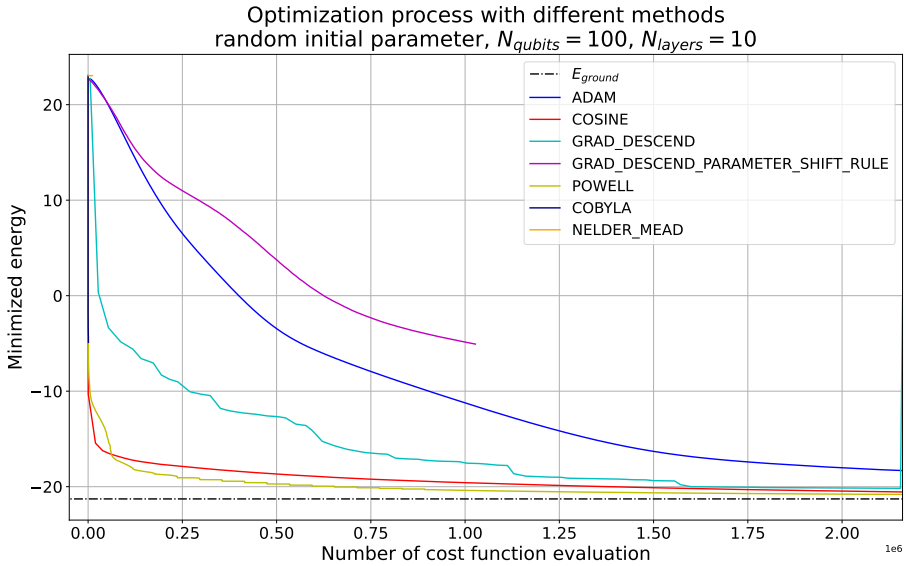

Figure 4: Minimization of cost function with random initial parameters.

Simulations with 14 qubits

For the simulations where I used more qubits, I did not use gradient-free methods due to the suggestion of my project leader. I used only the built-in methods in this case. Once again I started the simulation with zero parameter vector, and three out of five process stuck in the initial energy.

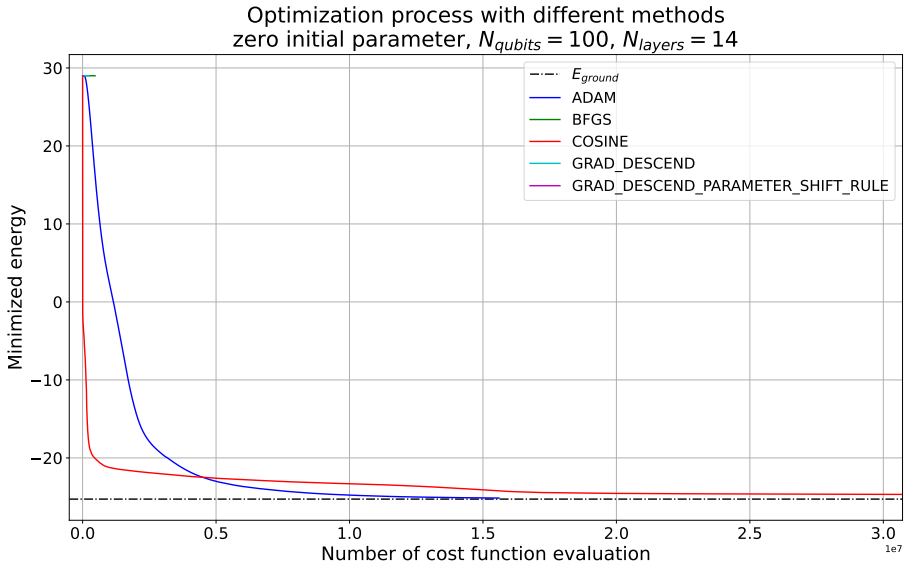

Figure 5: Minimization of cost function with zero initial parameters.

If we zoom into the first tenth of Fig. 5. We can see that the *Cosine* method created a similar trajectory as the *Powell* in case of Fig. 4.
The most interesting aspect of simulations using more qubits is when the initial vector is randomly chosen. Those method which did not worked earlier, works in this case, but still the *Cosine* method reaches the minima at the earliest. I ran the *BFGS* method multiple times, but in every case the beginning of the process did not saved. The *gradient descend* has the same peaks as before and it is still a searchable part of project for me. My project leader showed a few ways what can be a solution for this behavior, so I will try to find answers to this question to the next presentation event.
Finally the Fig. 7. will be more clear is we zoom into the beginning of the trajectories.

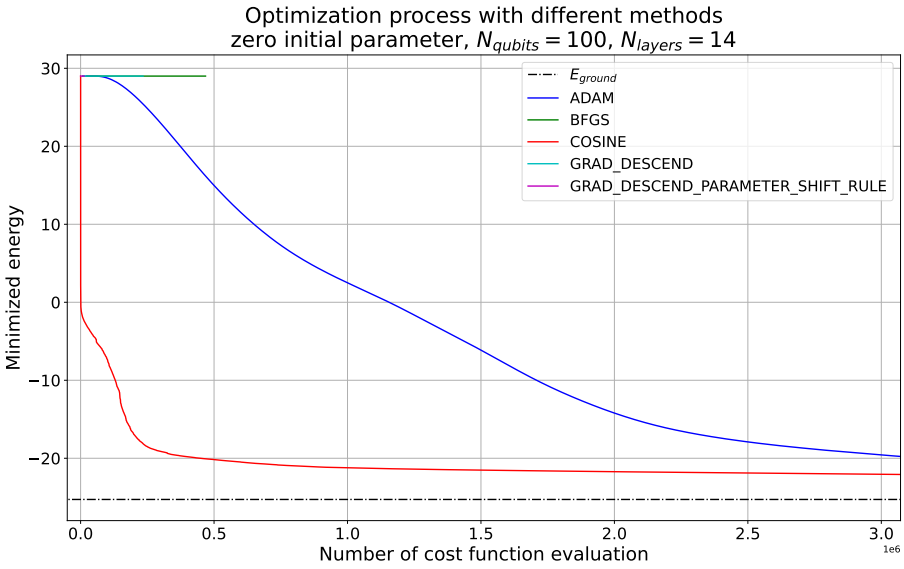

Figure 6: Minimization of cost function with zero initial parameters.

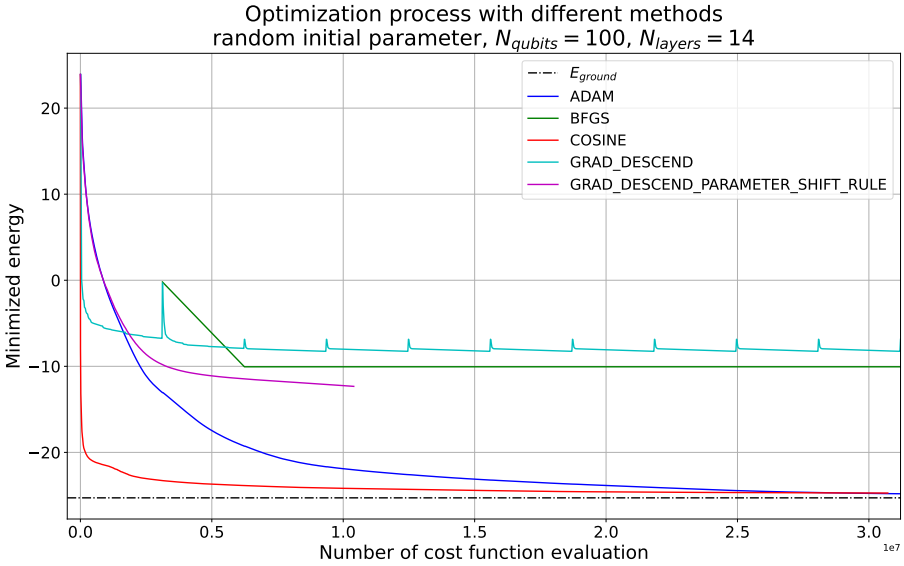

Figure 7: Minimization of cost function with random initial parameters.

Comparison

As a comparison to my results I would like to use the article cited in [1]. On Fig. 9. a few optimization process are visible. The name *This work* refers to

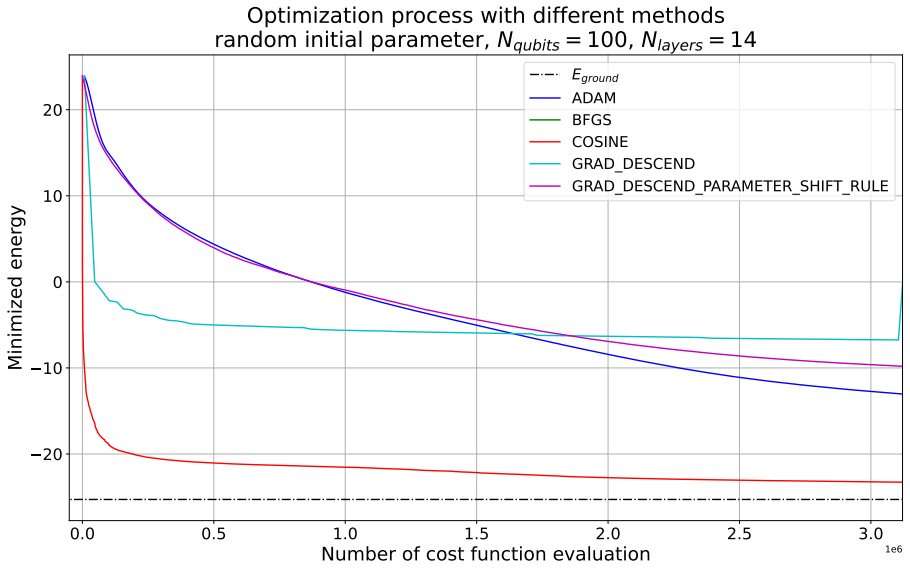

Figure 8: Minimization of cost function with random initial parameters.

Cosine method and the simulation was created with 10 qubits. In case of the Fig. 10. they used 16 qubits and 500 layers in the simulations.

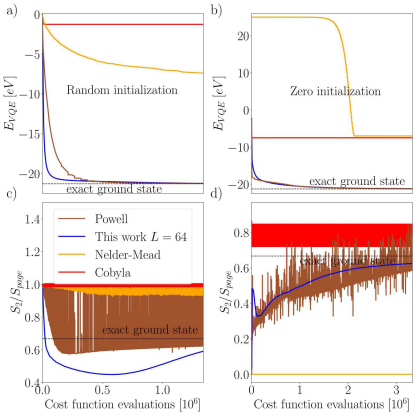

Figure 9: Fig. 8 from [1]

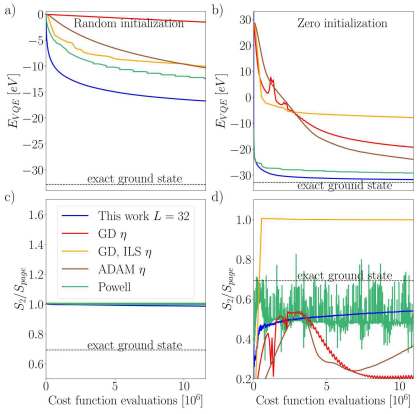

Figure 10: Fig. 9 from [1]

My consent to the development of SQUANDER package

While I tried to automatize the simulations in a bash script I realized that in the built-in C++ codes there is an issue with the name of the *gradient descend parameter shift rule* in lower case version. The word "parameter" was incorrectly written to "phase". Yesterday I sent a commit command with the modification. See the issue on Fig. 11.

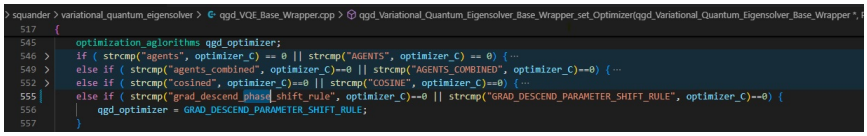

Figure 11: The issue in SQUANDER/qgd_VQE_Base_Wrapper.cpp file.

Further plans

As I wrote in the first section, mainly I plan to search the reason of the peaks in case of *Gradient Descend* method. Furthermore, I created log files for the simulations and I saved the runtime of the optimization process. In the future I will create bar charts to show the differences between the processing time.

References

[1] Jakab Nádori, Gregory Morse, Zita Majnay-Takács, Zoltán Zimborás, and Péter Rakyta. *Line search strategy for navigating through barren plateaus in quantum circuit training*, 2025. 2402.05227