QAOA: Performance, Mechanics, and Implementation on Near-Term Devices Zhou et al. (2018)

Paper review

Joost Bus

j.c.p.bus@student.tudelft.nl

Delft University of Technology, Netherlands

May 29, 2020

Overview

1. Questions addressed in this paper
2. Quantum and Classical
3. Patterns in the optimal parameters
4. Exploiting the patterns
   - INTERP
   - FOURIER

Current problems with QAOA

- Finding optimal parameters for QAOA is hard / not efficient in p
- Performance of QAOA beyond its lowest-depth variant ($p = 1$) is largely unknown.

What's done in this paper?

1. Propose heuristic strategies to effieciently optimise the variational parameters $\vec{\gamma}, \vec{\beta}$ by using found patterns from extensive benchmarking. These give quasi-optimal parameters, in the sense that they usually give the global optima. These parameters are found in $O(\text{poly}(p))$ time, as opposed to $2^{O(p)}$ when using (standard) random initalisation.
2. Benchmark the performance of QAOA and compare it with quantum annealing (Quantum vs. Quantum). Note, the main choke-point in QA is the presence of small spectral gaps.
3. Resource analysis on the experimental implementation of QAOA with near-term quantum devices.

What part is quantum and what part is classical?

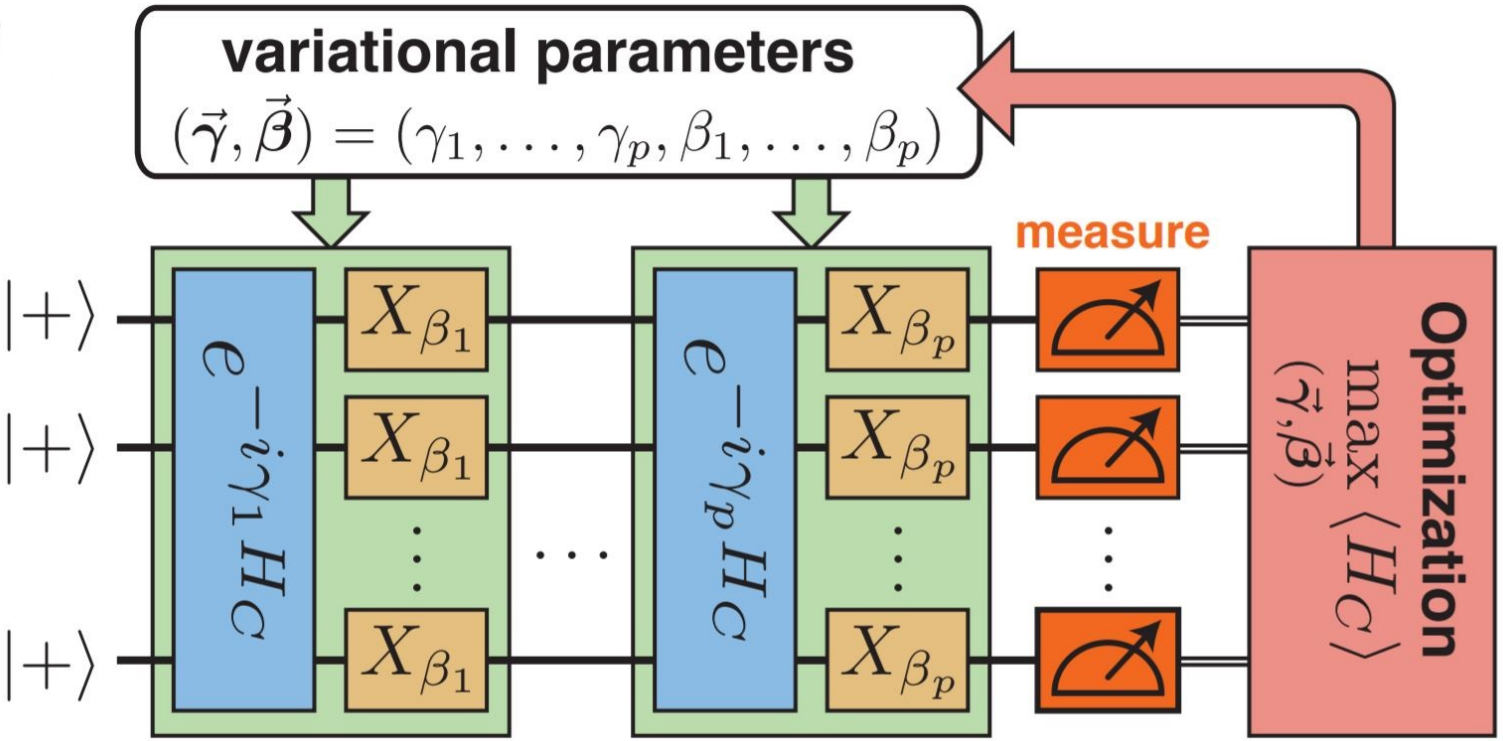


Figure: Zhou et al. (2019) page 2

What part is quantum and what part is classical?

We want to maximise the expectation value of the cost function $F_p \equiv \langle \vec{\gamma}, \vec{\beta} | H_C | \vec{\gamma}, \vec{\beta} \rangle$, where H_C is the cost function operator. To estimate this quantity, we make a call to the QPU and take the average over multiple shots. We use an optimisation scheme to determine the next parameters that are to be tried.

Several optimisation methods are mentioned by Zhou et al. these are

- BFGS
- Nelder-Mead (Visualisation)
- Bayesian Optimisation

SciPy has an optimisation package that comes in handy, you only need to input the objective function that needs to be optimised, together with an initial point (optionally with other parameters)

Patterns in the optimal parameters

Randomly initialised optimisation is not efficient in p. Zhou et al. instead looked for patterns in the optimal parameters found for u3R, w3R and complete graphs. For u3R graphs they found the following

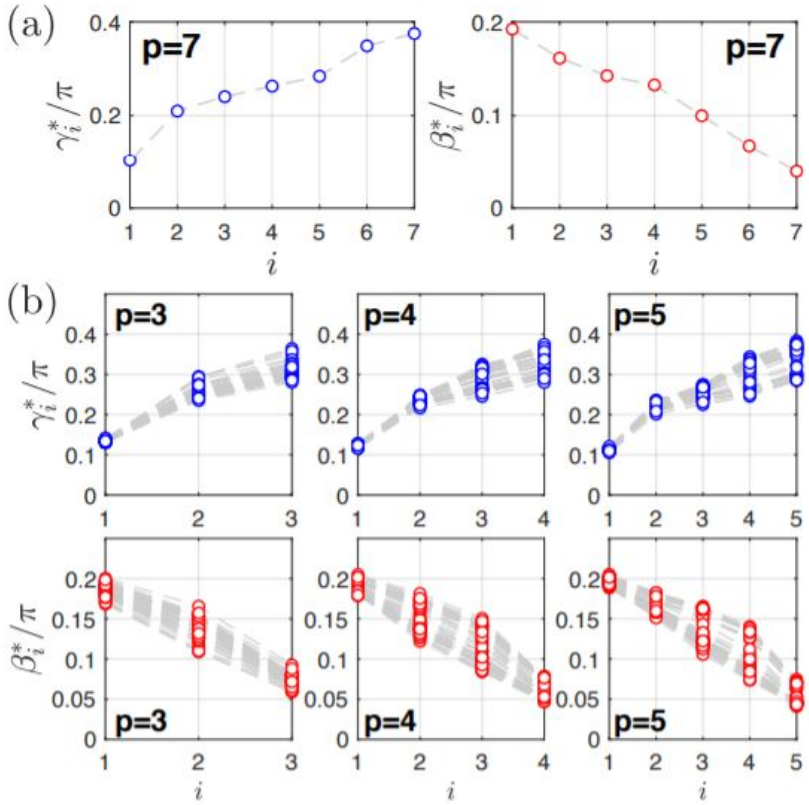
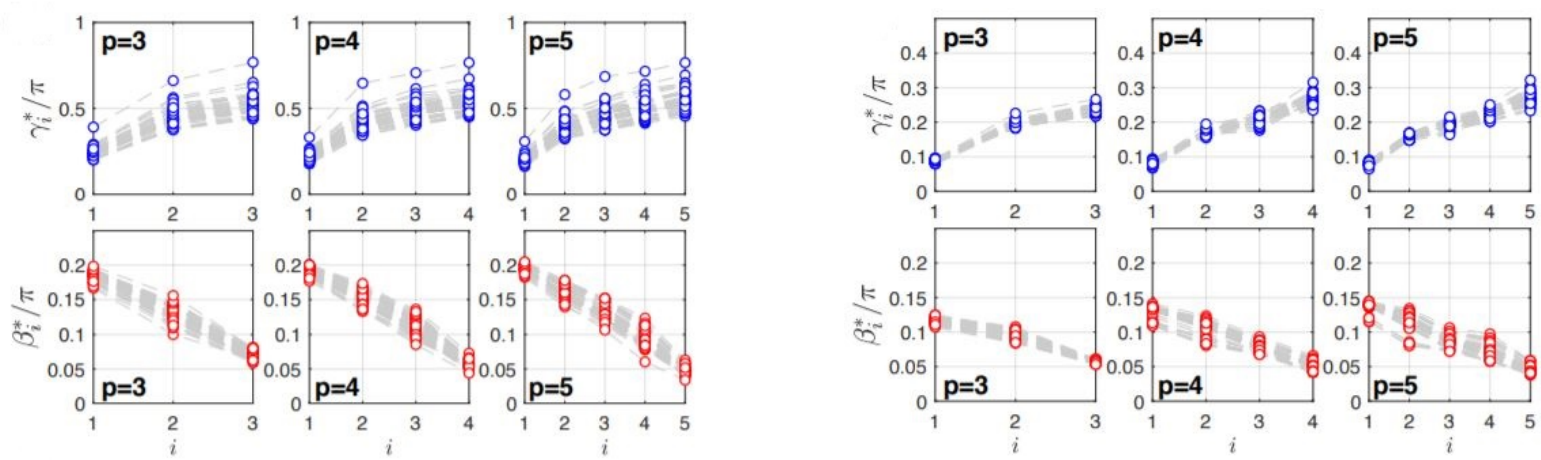


Figure: Optimal parameters for randomly generated u3R graphs
Zhou et al. (2019) page 4

Similar patterns were found for w3R and complete graphs.



(a) w3R graphs (b) Complete graphs

Figure: Zhou et al. (2019) page 16 (Appendix)

As can be seen, γ_i monotonically increases and β_i monotonically decreases. We can exploit this pattern in the optimisation!

Note, this pattern is also reminiscent of the Quantum Adiabatic algorithm, where the mixer hamiltonian H_B is gradually turned off (as $\beta \to 0$), and the cost hamiltonian H_C is gradually turned on.

Exploiting the patterns - proposed heuristic methods

Zhou et al. proposed two ways to exploit these patterns, these are poetically named

1. INTERP
2. FOURIER

Because $(\vec{\gamma}^*_{(p+1)}, \vec{\beta}^*_{(p+1)})$ closely resembles $(\vec{\gamma}^*_{(p)}, \vec{\beta}^*_{(p)})$, the idea is to find optimal parameters for low p, and from that choose a suitable initialisation for the optimisation for $p+1$

INTERP

INTERP uses linear interpolation to produce a good initial point $(\vec{\gamma}, \vec{\beta})$ for optimising the the QAOA parameters as one iteratively increases the level p

$$\left[\vec{\gamma}^{0}_{(p+1)}\right]_i = \frac{i-1}{p}\left[\vec{\gamma}^{L}_{(p)}\right]_{i-1} + \frac{p-i+1}{p}\left[\vec{\gamma}^{L}_{(p)}\right]_i \tag{1}$$

$$\left[\vec{\beta}^{0}_{(p+1)}\right]_i = \frac{i-1}{p}\left[\vec{\beta}^{L}_{(p)}\right]_{i-1} + \frac{p-i+1}{p}\left[\vec{\beta}^{L}_{(p)}\right]_i \tag{2}$$

for $i = 1, 2, \ldots, p+1$.

INTERP

Let's show an example. Suppose we have an (local optimal) set $\vec{\gamma}^{L}_{(3)} = (\gamma_1, \gamma_2, \gamma_3)$, then we can find a good initial point $\vec{\gamma}^{0}_{(4)} = (\gamma^0_1, \gamma^0_2, \gamma^0_3, \gamma^0_4)$ using the INTERP method.

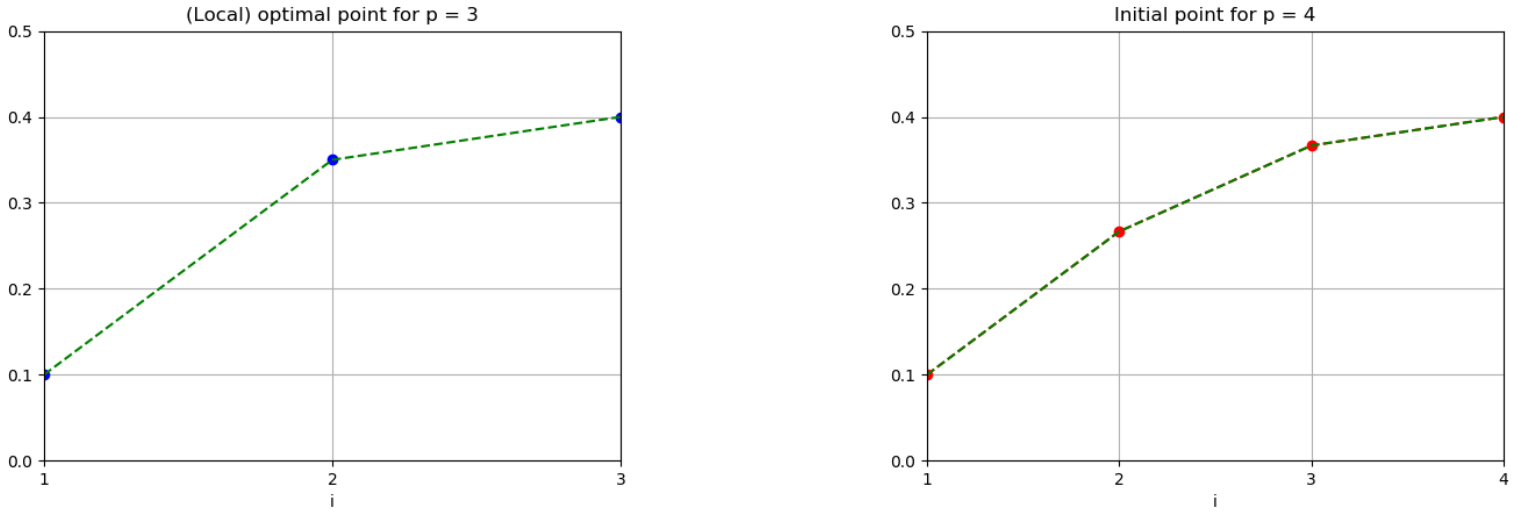


Figure: Illustration of INTERP method

We can use the same technique for $\vec{\beta}$

FOURIER

FOURIER is a somewhat more elaborate technique, but according to Zhou et al. it has a slight edge over INTERP. It's called FOURIER because it makes use of the discrete Fourier and sine transforms. In FOURIER we use a new representation of the p-level QAOA parameters using $(\vec{u}, \vec{v}) \in \mathbb{R}^{2q}$

$$\gamma_i = \sum_{k=1}^{q} u_k \sin\left(\left(k - \frac{1}{2}\right)\left(i - \frac{1}{2}\right)\frac{\pi}{p}\right) \tag{3}$$

$$\beta_i = \sum_{k=1}^{q} v_k \cos\left(\left(k - \frac{1}{2}\right)\left(i - \frac{1}{2}\right)\frac{\pi}{p}\right) \tag{4}$$

Here q labels the number of frequency components (and the maximum frequency component) we allow.

FOURIER

The strategy works by starting from $p = 1$ and then reusing the optimum at level p in $(\vec{u}, \vec{v})$-representation to generate a good initial point for level $p + 1$

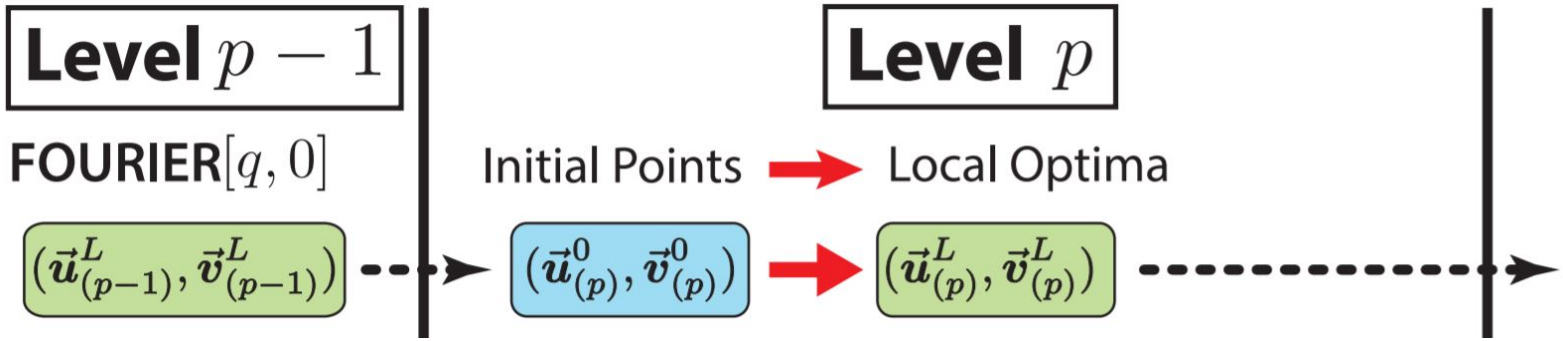


Figure: Zhou et al. (2019) page 17 (Appendix)

Natural choices for q are p, but a constant is also possible since the pattern in the optimal parameters seems smooth.

FOURIER[$q = p$,0] a.k.a. FOURIER[∞,0]

When $q = p$ we generate new initial points according to

$$\vec{u}^0_{(p+1)} = (\vec{u}^L_{(p)}, 0), \quad \vec{v}^0_{(p+1)} = (\vec{v}^L_{(p)}, 0) \tag{5}$$

All this means is that we generate a good initial point for level $p+1$ by adding a higher frequency component, initialized at zero amplitude.

FOURIER$[q, R]$

They proposed several variants of the FOURIER strategy: FOURIER$[q, R]$. This approach uses an extra R random bumps to get out of local optima.

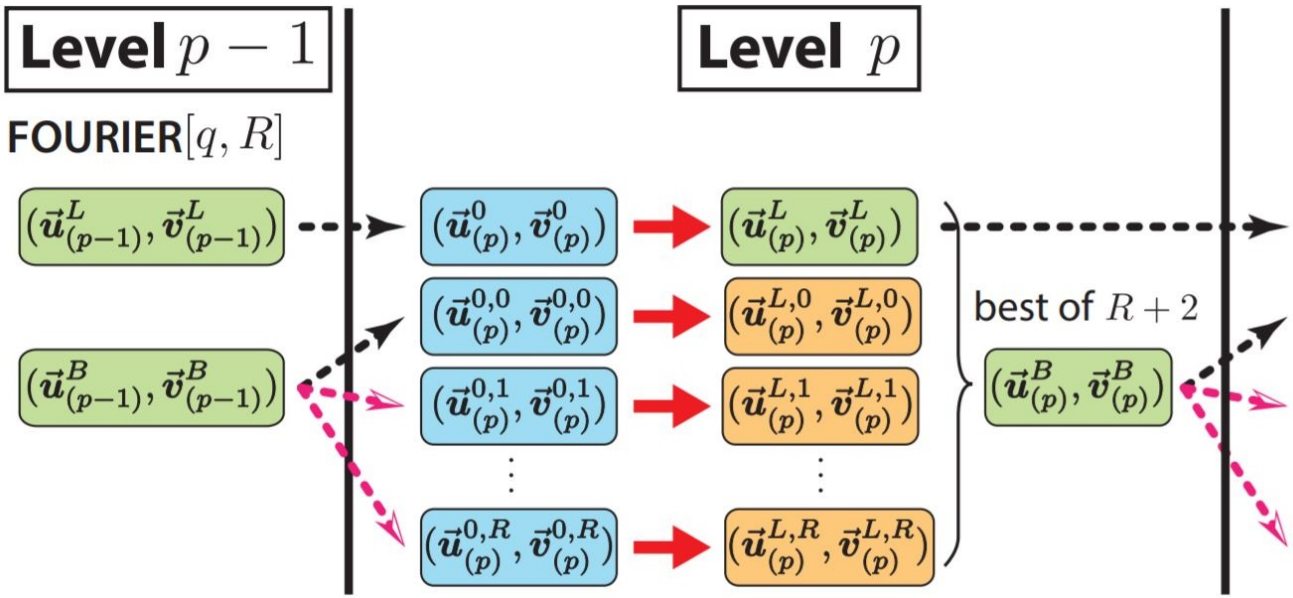


Figure: Zhou et al. (2019) page 17 (Appendix)

FOURIER$[\infty, R]$

In this variant we perturb the initial point to prevent getting stuck in a local minimum. We do this R times and keep the best set B for the next level $p+1$. For improved robustness of the method we also keep the unperturbed optimized parameters (basically the FOURIER$[\infty, 0]$ method)

$$\vec{u}^{0,r}_{(p+1)} = \begin{cases} (\vec{u}^{B}_{(p)}, 0) & r = 0 \\ (\vec{u}^{B}_{(p)} + \alpha \vec{u}^{P,r}_{(p)}, 0) & 1 \leq r \leq R \end{cases} \tag{6}$$

$$\vec{v}^{0,r}_{(p+1)} = \begin{cases} (\vec{v}^{B}_{(p)}, 0) & r = 0 \\ (\vec{v}^{B}_{(p)} + \alpha \vec{v}^{P,r}_{(p)}, 0) & 1 \leq r \leq R \end{cases} \tag{7}$$

where $\vec{u}^{P,r}_{(p)}$ and $\vec{v}^{P,r}_{(p)}$ are the perturbations; randomly drawn numbers from a normal distribution with mean 0 and variance $|\vec{u}^{B}_{(p)}|^2$, $|\vec{v}^{B}_{(p)}|^2$ respectively. The strength of the perturbation is α, according to the paper $\alpha = 0.6$ works well (by trial and error).

Performance of the heuristics

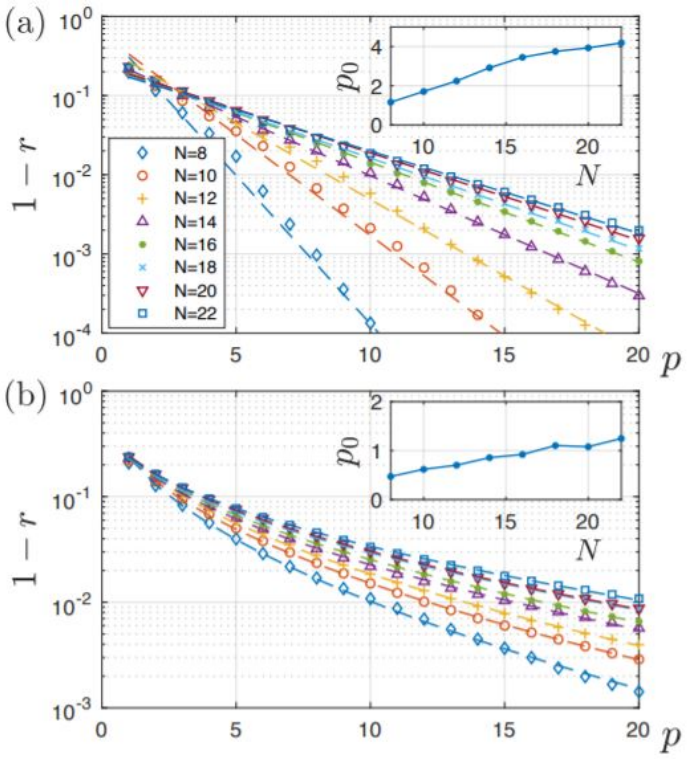


FIG. 4. Average performance of QAOA as measured by the fractional error $1-r$, plotted on log-linear scale. The results are obtained by applying our heuristic optimization strategies FOURIER to up to 100 random instances of (a) u3R graphs and (b) w3R graphs. Differently colored lines correspond to fitted lines to the average for different system size N, where the model function is $1-r \propto e^{-p/p_0}$ for unweighted graphs and $1-r \propto e^{-\sqrt{p/p_0}}$ for weighted graphs. Insets show the dependence of the fit parameters p_0 on the system size N.

Figure: Zhou et al. (2019)

Performance of the heuristics

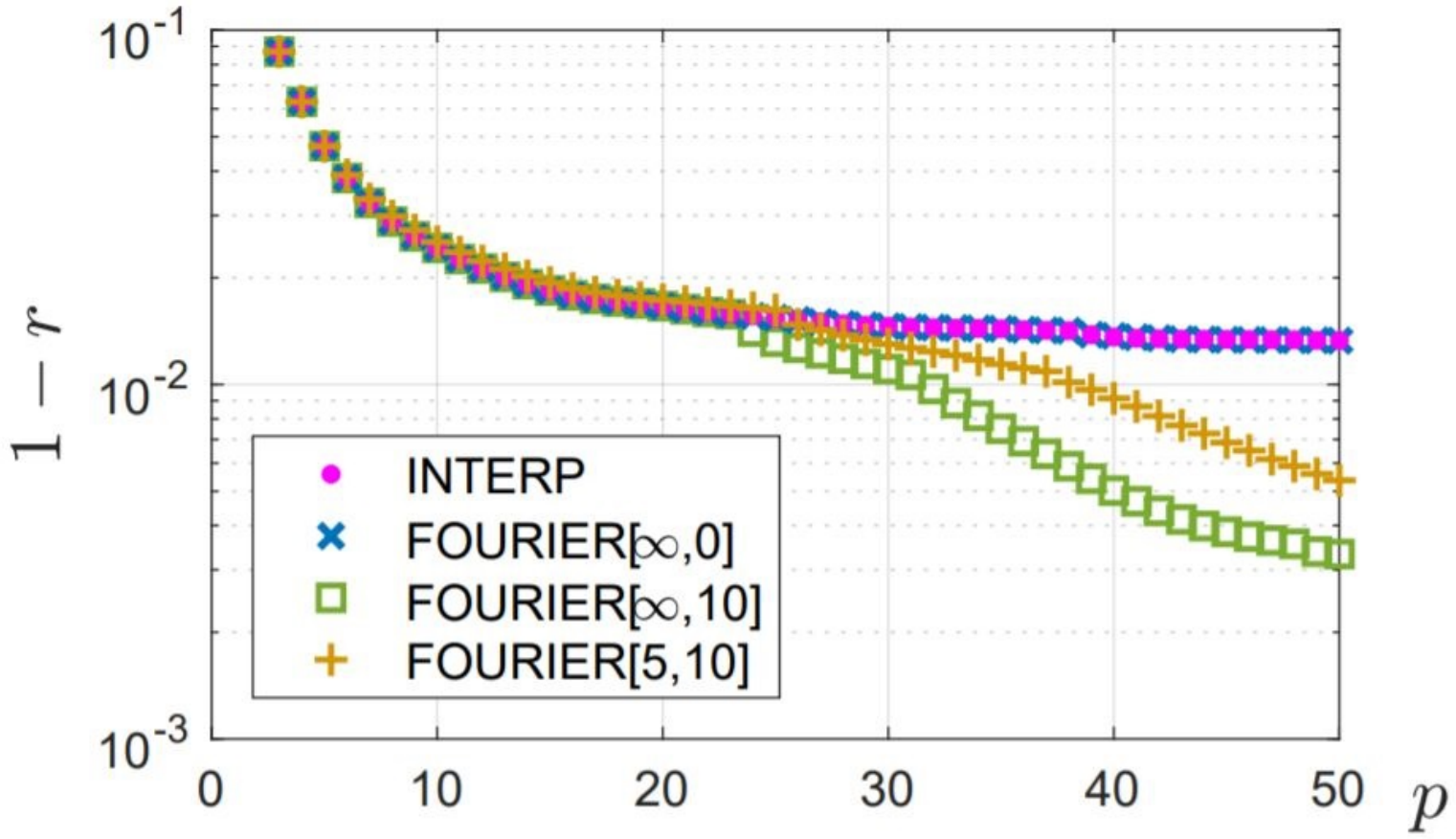


FIG. 11. The fractional error achieved by optimizing using our various heuristics on an example instance of 14-vertex w3R graph.

Figure: Zhou et al. (2019)