A generalized eigenvalue problem via a quantum annealer

Alex Gnech

ECT* (FBK)

Quantum Computing @ INFN 15/11/2022

Collaborators and support

- Filippo Fornetti (UNIPG and INFN Perugia)
- Tobias Frederico (ITA Brasil)
- Francesco Pederiva (UNITN and TIFPA)
- Matteo Rinaldi (UNIPG and INFN Perugia)
- Giovanni Salme' (INFN Roma)
- Sergio Scopetta (UNIPG and INFN Perugia)
- Michele Viviani (INFN Pisa)

Q@TN

INFN

D:WAVE

The Quantum Computing Company™

The Generalized Eigenvalue Problem

$$Av_i = \lambda_i B v_i$$

- A = Real symmetric or non symmetric matrix
- B = Real symmetric matrix
- v_i = Eigenvector
- λ_i = Eigenvalue (we look for the real one)

Solve the GEVP with an annealer

The Homogeneous Bethe-Salpeter Equation

- Describe bound state in a Relativistic Quantum Field Theory (in Minkowski space) framework (k/m >1/4)

- Studies of momentum distribution of the constituents inside bound states at JLab and EIC

- Strongly non perturbative regime of Quantum Field Theory

$$\phi(k;p_B,\beta) = G_0(k;p_B,\beta)\int d^4k'\mathcal{I}(k,k';p_B)\phi(k';p_B,\beta)$$

- Can be rewritten as a Non Symmetric Generalized Eigenvalue problem [1]

[1] W. De Paula et. al., PRD **94** 071901

Goals of the project

Outline of the talk

- The symmetric standard (and generalized) eigenvalue problem (Krakoff et al.)

- The non-symmetric standard eigenvalue problem

- Some practical examples

- The generalized non-symmetric eigenvalue problem (working on)

Avoid matrix operation: minimize computational cost as N increase

The Symmetric Case

$$Av_i = \lambda_i v_i$$

A = Real Symmetric Matrix

Finding the minimal eigenvalue with an annealer

- **Minimization problem** → Rayleigh-Ritz quotient

$$\mathcal{R} = \frac{\mathbf{v}^T A \mathbf{v}}{\mathbf{v}^T \mathbf{v}}$$

- **QUBO form (Ising problem)**

  Quadratic Unconstrained Binary Optimization

The Non Symmetric Case

$$Av_i = \lambda_i v_i$$

A = Real Non Symmetric Matrix (We are looking for real solutions)

- **Minimization problem** → Rayleigh-Ritz quotient

$$J(\mathbf{v}, \lambda) = \frac{|(A - \lambda I)\mathbf{v}|^2}{\mathbf{v}^T \mathbf{v}}$$ ★

- **QUBO form (Ising problem)**

  Quadratic Unconstrained Binary Optimization

★ S. Alliney et al. , Appl. Math. Modelling **16**, 148

Specific on the Non Symmetric Case

- We are looking for the real solutions (the largest for the BSE)

$$\lambda = \frac{1}{2}\frac{\mathbf{v}^T(A + A^T)\mathbf{v}}{|\mathbf{v}|^2} \pm i\sqrt{R(\mathbf{v})} \qquad R(\mathbf{v}) = \frac{\mathbf{v}^T A^T Q(\mathbf{v}) A\mathbf{v}}{|\mathbf{v}|^2} \qquad Q(\mathbf{v}) = \left(1 - \frac{\mathbf{v}\mathbf{v}^T}{|\mathbf{v}|^2}\right)$$

- R(**v**)>0. Real solution only for R(v)=0 → Minimization
  - All the real solution are equivalent for the annealer
  - It is not a quadratic problem! (Can be solved iteratively)

- Direct minimization of $J(\mathbf{v}, \lambda) = \mathbf{v}^T(A - \lambda I)^T(A - \lambda I)\mathbf{v}$
  - Need to select a starting λ
  - Purely quadratic

The problem in a QUBO form

- Rewriting the numbers in binary basis $$a = \mathbf{p}^T \cdot \mathbf{q}_b$$

$$\mathbf{p} = (-1, 2^{-1}, 2^{-2}, \ldots, 2^{-b+1})$$

$$\mathbf{q}_b = (q_0, q_1, q_2, \ldots, q_{b-1})$$

q_i = binary (0,1)

- The matrix that is used then is

$$\mathbf{v}^T A \mathbf{v} = \mathbf{X}^T Q \mathbf{X}$$

$$Q = P^T A P$$

$$P = \begin{pmatrix} \mathbf{p} & 0 & \ldots & 0 \\ 0 & \mathbf{p} & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \ldots & \mathbf{p} \end{pmatrix}$$

The dimension of matrix Q is (Nxb) x (Nxb)

The Algorithm

- (1) The algorithm is an approximation of the actual Rayleigh-Ritz quotient (or J)

$$\lambda_0 = \min_{|\mathbf{v}|=1} \mathbf{v}^T A \mathbf{v} \sim \min_{\mathbf{X} \in C_{n,b}} \mathbf{X}^T Q \mathbf{X} \qquad C_{n,b} \text{ discretized cube}$$

- (2) Step descendent phase $f(\mathbf{v}) = \mathbf{v}^T (A - \lambda I)\mathbf{v}$

$$f(\mathbf{v}) = f(\mathbf{v}_0) + \boxed{\nabla_{\mathbf{v}} f(\mathbf{v}) \cdot \delta + \delta^T \frac{H(f)}{2} \delta}$$

- An iterative procedure is used to find the minimal value of λ by increasing the precision when needed

- For the generalized case just substitute I with B

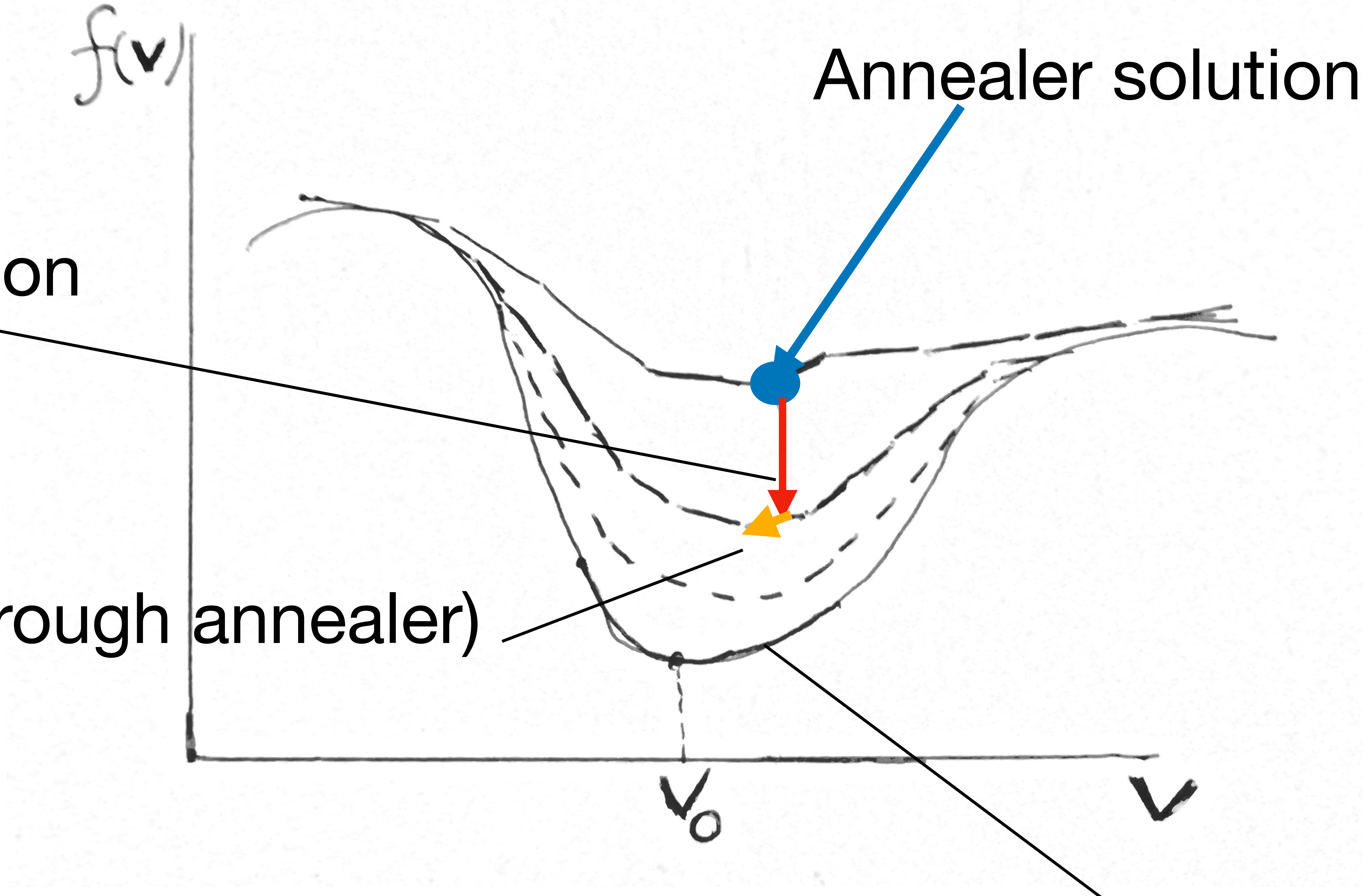
*B. Krakoff, et al., arXiv:2104.11311, and Plos one **17**, e0267954 (2022)

A visual realization

Annealer solution first step

Increasing the precision

$C_{n,b} \rightarrow t \cdot C_{n,b}$



Step descendent (through annealer)

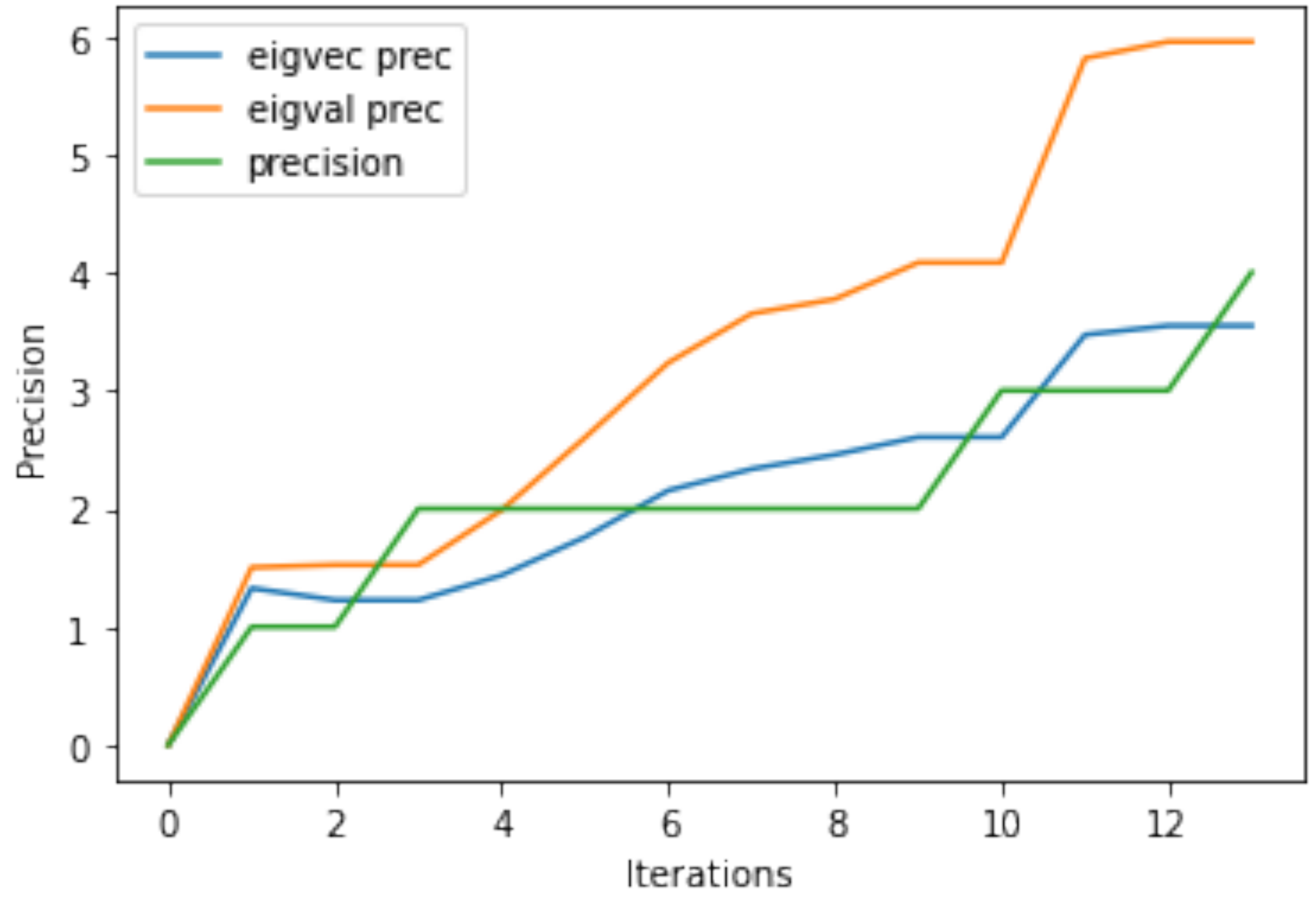
Actual function

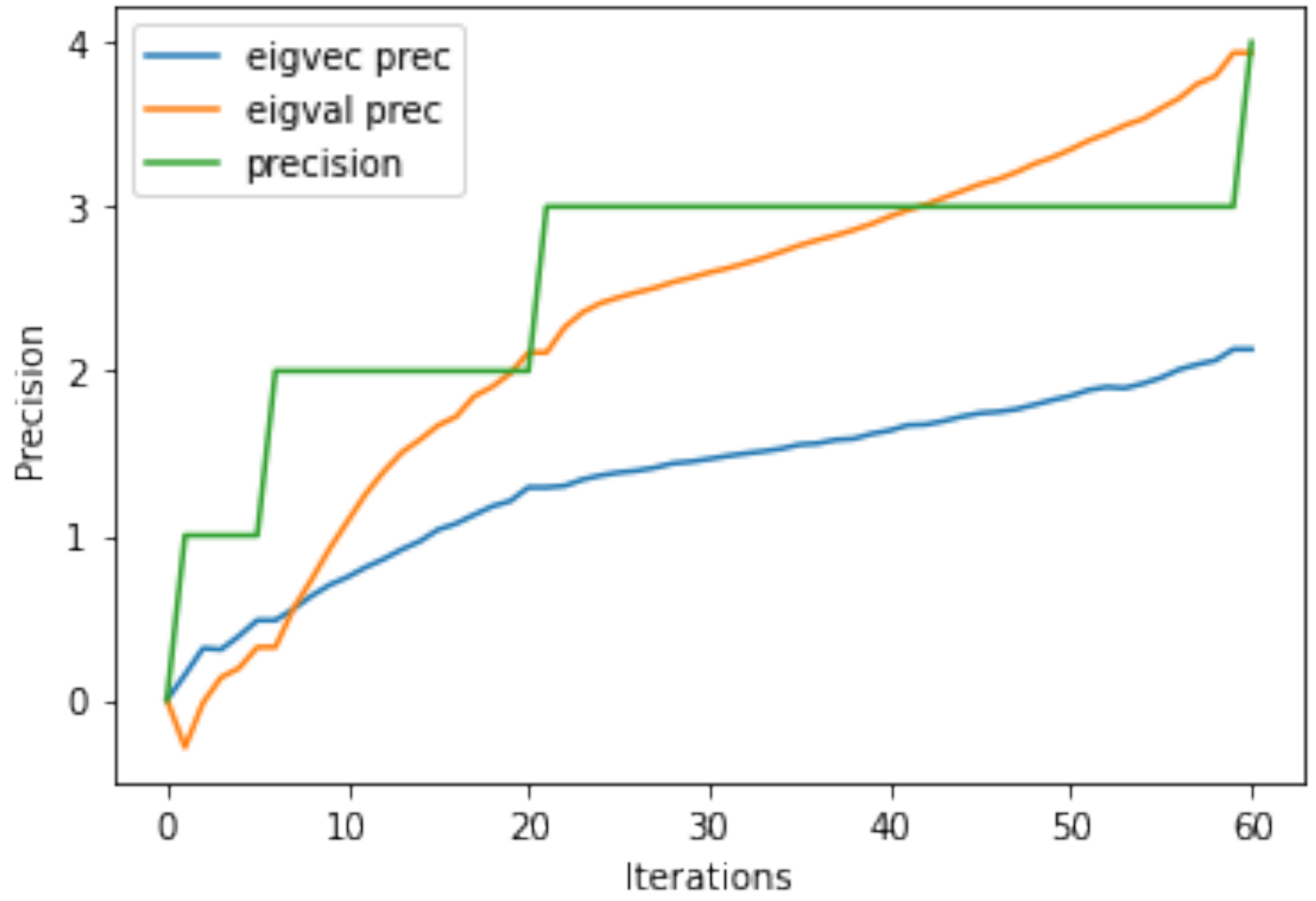
First step of the algorithm

- 4 X 4 matrix
- Starting $\lambda = tr(A)/N = -1.542$
- Actual result $\lambda = -5.379$
- 20 number of reads for annealing

Num of bits (b)	Eigenvalue
2	-5.288
4	-5.250
8	-5.322
16	-5.290
32	-4.901

What you get from the code (symmetric)





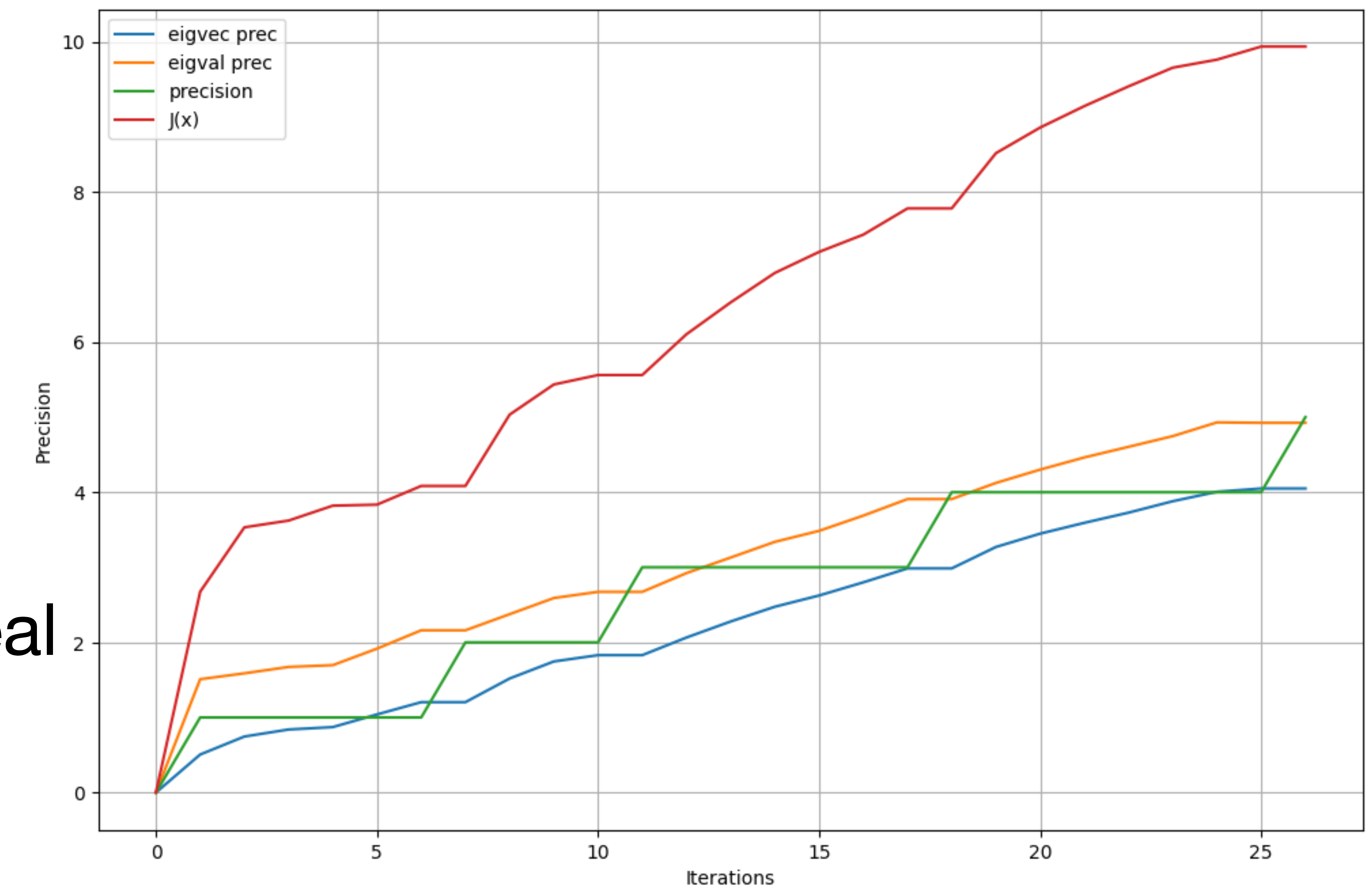
- Dimension 8x8
- Annealing time 0.250 s
- λ=-11.95623 (-11.95623)
- Precision 0.001

- Dimension 32x32
- Annealing time 1.08 s
- λ= -19.8999 (-19.9000)
- Precision 0.001

Different behavior due to gap among first and second eigenvalue

An example for the non-symmetric

- Matrix 12x12 (annealing time 0.49 s)
- λ=0.188213 (0.188224) p=0.0001
- We minimize J(**v**,λ)
- Limited by numerical precision on J ($p(J) \sim 10^{-16}$ $p(\lambda) \sim 10^{-8}$)
- In principle J can be used for all the real eigenvalue
- Hard when gaps among eigenvalues are small



The non symmetric case (issues)

- Direct minimization of $J(\mathbf{v},\lambda) = \mathbf{v}^T(A-\lambda I)^T(A-\lambda I)\mathbf{v}$
    - Need to select a starting point
    - Purely quadratic
- For the generalized case symmetric we just need

$$A - \lambda I \longrightarrow A - \lambda B$$

- For the non symmetric case

$$J(\mathbf{v},\lambda) = \frac{|(A-\lambda I)\mathbf{v}|^2}{\mathbf{v}^T\mathbf{v}} \longrightarrow J(\mathbf{v},\lambda) = \frac{|(A-\lambda B)\mathbf{v}|^2}{\mathbf{v}^T\mathbf{v}}$$

    - The algorithm is able to find the eigenvalue with a finite precision
    - The eigenvector is poorly reproduced

To do next (for non-symmetric matrix)

- Fix the generalized non symmetric algorithm
- Improving the selection of the starting eigenvalue
- Study of scaling with the dimension of the matrix
- Extension to not max or min eigenvalues

A realization of the algorithm for the symmetric case is available at
https://github.com/agnech00/CESM_QUBO