Title: Adiabatic optimization without a priori knowledge of the spectral gap

Date: Sep 19, 2018 04:00 PM

URL: http://pirsa.org/18090052

Abstract: <p>Performing a quantum adiabatic optimization (AO) algorithm with the time-dependent Hamiltonian H(t) requires one to have some idea of the spectral gap Î³(t) of H(t) at all times t. With only a promise on the size of the minimal gap Î³_{min}, a typical statement of the adiabatic theorem promises a runtime of either O(Î³_{min}⁻²) or worse. In this talk, I provide an explicit algorithm that, with access to an oracle that returns the spectral gap Î³(t) to within some multiplicative constant, reliably performs QAO in time OÌƒ(Î³_{min}⁻¹) with at most O(log(Î³_{min}⁻¹)) oracle queries. I then construct such an oracle using only computational basis measurements for the toy problem of a complete graph driving Hamiltonian on V vertices and arbitrary cost function. I explain why one cannot simply perform adiabatic Grover search and prove that one can still perform QAO in time OÌƒ(V^{2/3}) without any a priori knowledge of Î³(t). This work was done in collaboration with Brad Lackey, Aike Liu, and Kianna Wan.</p>

A bashful adiabatic algorithm

Adiabatic optimization without heuristics

Michael Jarret

Perimeter Institute

Sept 19, 2018

Joint work with:

Brad Lackey (UMD)

Kianna Wan (PI,Waterloo → Stanford)

Aike Liu (UIUC, Summer at PI)

Initial proposal: arXiv:1804.06857
Current results: Forthcoming

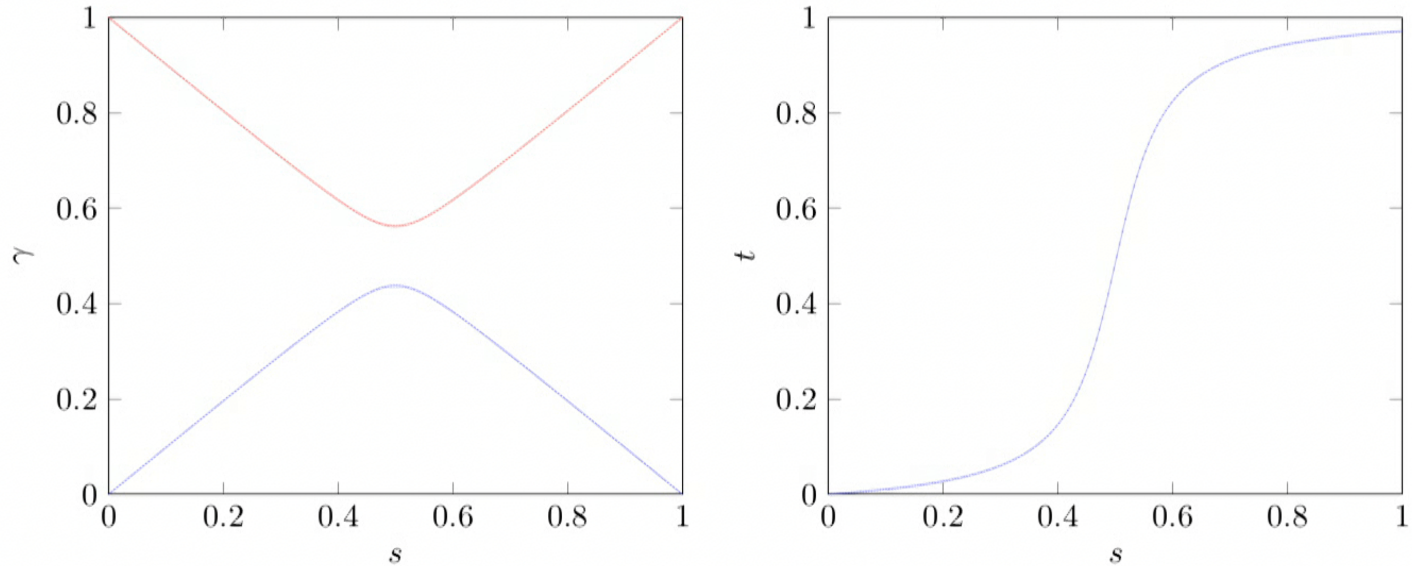
Will share manuscript *iff* you ask nicely

Adiabatic Optimization

- Goal is to use time-dependent Hamiltonian $H(s)$ where $H(0)$ is known and $H(1)$ is an optimization problem
- Ground state of $H(1)$ solves the optimization problem
- Usually see something like $H(s) = (1-s)H(0) + sH(1)$
- Use a second time parameter $t \in [0, T]$ such that ds/dt is sufficiently small, ie. make T big enough
- Adiabatic theorem gives scaling like $T \sim O(\gamma_{\min}^{-2})$ or $T \sim O(\gamma_{\min}^{-3})$
- Typically guess T, hope for the best
- Some strategies exist for doing better, but limited and little rigor (or require more machinery)

Adiabatic Grover

$$H(s) = (1-s)\left(I - \tfrac{1}{V}\textstyle\sum_{u,v}|u\rangle\langle v|\right) + sW, \qquad W = \mathrm{diag}(0,1,1,\ldots,1)$$



$O(\sqrt{V}) = O(1/\gamma_{\min})$ [Roland and Cerf, 2001]

Beyond adiabatic Grover

- What happens if $W \neq \text{diag}(0, 1, \ldots, 1)$?
  - SAT: Energy of $\langle b|W|b\rangle$ may not be the same as SAT/UNSAT

Beyond adiabatic Grover

- What happens if $W \neq \text{diag}(0, 1, \ldots, 1)$?
  - SAT: Energy of $\langle b|W|b\rangle$ may not be the same as SAT/UNSAT
  - Implemented Hamiltonian could have some (known) noise in W
  - Preprocessing could be expensive
- Standard approach is still to just guess a runtime and go
- But then the (rigorous) time is $O(1/\gamma_{\text{min}}^2)$ or $O(1/\gamma_{\text{min}}^3)$

Beyond adiabatic Grover

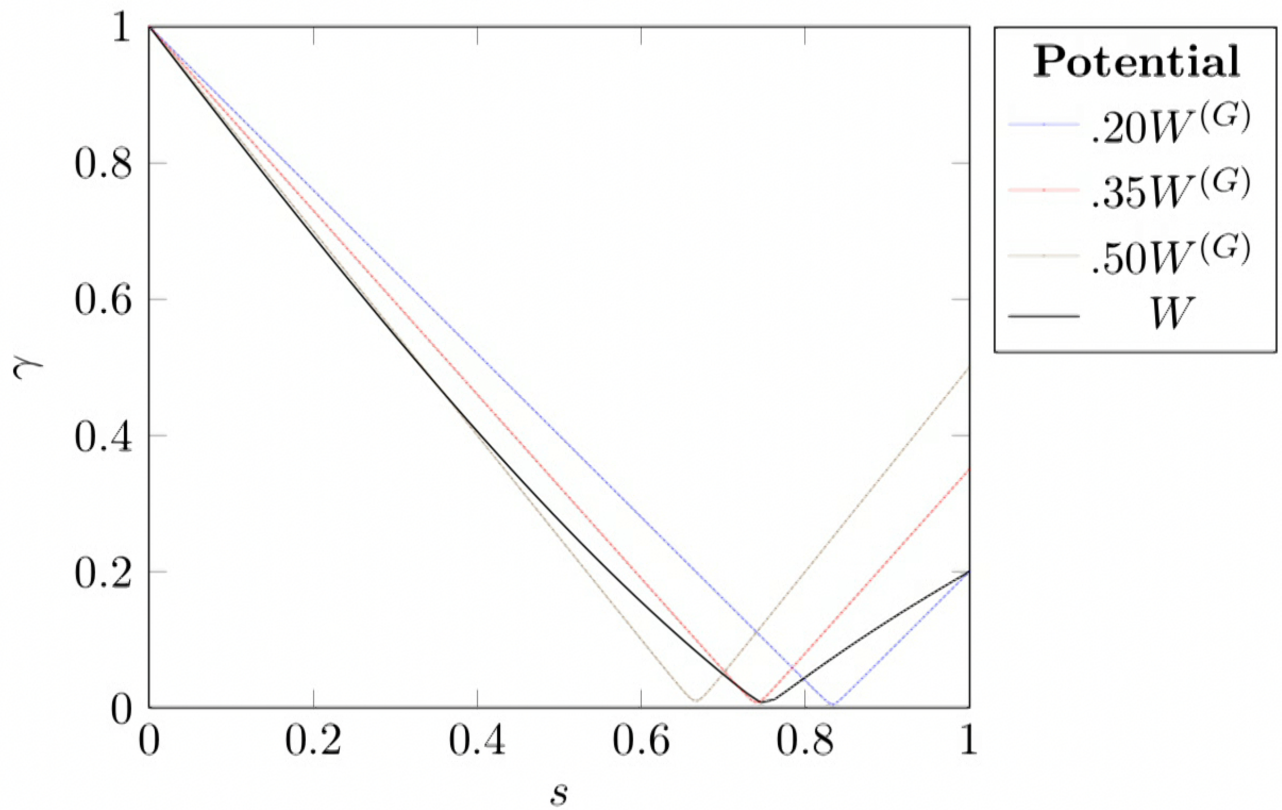
- What happens if $W \neq \mathrm{diag}(0, 1, \ldots, 1)$?
  - SAT: Energy of $\langle b|W|b\rangle$ may not be the same as SAT/UNSAT
  - Implemented Hamiltonian could have some (known) noise in W
  - Preprocessing could be expensive
- Standard approach is still to just guess a runtime and go
- But then the (rigorous) time is $O(1/\gamma_{\min}^2)$ or $O(1/\gamma_{\min}^3)$
- Some suggest approximation methods or other heuristics for better scaling

Beyond adiabatic Grover

- What happens if $W \neq \mathrm{diag}(0, 1, \ldots, 1)$?
  - SAT: Energy of $\langle b|W|b\rangle$ may not be the same as SAT/UNSAT
  - Implemented Hamiltonian could have some (known) noise in W
  - Preprocessing could be expensive
- Standard approach is still to just guess a runtime and go
- But then the (rigorous) time is $O(1/\gamma_{\min}^2)$ or $O(1/\gamma_{\min}^3)$
- Some suggest approximation methods or other heuristics for better scaling
- ...not good enough for certainty

Beyond adiabatic Grover

$$H = (1-s)L + sW$$



Beyond Adiabatic Grover

One somewhat promising, commonly suggested strategy is to try inserting some intermediate Hamiltonian [Farhi, Crosson, ... '14]:

$$H(s) = (1 - s)H(0) + s(1 - s)H_? + sH(1)$$

but we're still guessing at $H_?$

Want a strategy that isn't heuristic, is reliable, and can presumably be performed.

Need a different algorithm

Setup:

1. I have a fixed driving Hamiltonian $H(0)$

Need a different algorithm

Setup:

1. I have a fixed driving Hamiltonian $H(0)$
2. You give me a cost function $H(1) = W$ as an oracle
3. I produce the best possible schedule $H(s)$ and its result

Need a different algorithm

Setup:

1. I have a fixed driving Hamiltonian $H(0)$
2. You give me a cost function $H(1) = W$ as an oracle
3. I produce the best possible schedule $H(s)$ and its result

Two questions seem important:

1. Suppose we have access to an oracle capable of estimating the gap, how many queries do we need to find a good schedule?
2. Assuming that works out well, can we build the thing?

Need a different algorithm

We'll need to collect a few tools

- Adaptations of the adiabatic theorem
- Weyl's inequalities

Need a different algorithm

We'll need to collect a few tools

- Adaptations of the adiabatic theorem
- Weyl's inequalities
- Some spectral graph theory

Adiabatic Theorem [JRS 06]

$$\|\widetilde{P}(s) - P(s)\| \leq \frac{1}{T}\left[\frac{\|\dot{H}(0)\|}{\gamma(0)^2} + \frac{\|\dot{H}(s)\|}{\gamma(s)^2} + \int_0^s ds' \left(7\frac{\|\dot{H}(s')\|^2}{\gamma(s')^3} + \frac{\|\ddot{H}(s')\|}{\gamma(s')^2}\right)\right]$$

Adiabatic Theorem [JRS 06]

$$\|\widetilde{P}(s) - P(s)\| \leq \frac{1}{T}\left[\frac{\|\dot{H}(0)\|}{\gamma(0)^2} + \frac{\|\dot{H}(s)\|}{\gamma(s)^2} + \int_0^s ds' \left(7\frac{\|\dot{H}(s')\|^2}{\gamma(s')^3} + \frac{\|\ddot{H}(s')\|}{\gamma(s')^2}\right)\right]$$

What if $\|\dot{H}(s)\| \leq c\gamma(s)$ and $\|\ddot{H}(s)\| = 0$?

Adiabatic Theorem [JRS 06]

$$\|\widetilde{P}(s) - P(s)\| \leq \frac{1}{T}\left[\frac{\|\dot{H}(0)\|}{\gamma(0)^2} + \frac{\|\dot{H}(s)\|}{\gamma(s)^2} + \int_0^s ds' \left(7\frac{\|\dot{H}(s')\|^2}{\gamma(s')^3} + \frac{\|\ddot{H}(s')\|}{\gamma(s')^2}\right)\right]$$

What if $\|\dot{H}(s)\| \leq c\gamma(s)$ and $\|\ddot{H}(s)\| = 0$?

$$\|\widetilde{P}(s) - P(s)\| \leq \frac{2c + 7c^2}{T\gamma_{\min}}$$

Weyl's inequality

Weyl's inequality is actually a bit tighter than

$$|\lambda_i(H+\Delta) - \lambda_i(H)| \leq \|\Delta\|$$

which we convert to

$$|\gamma(H+\Delta) - \gamma(H)| \leq 2\|\Delta\|.$$

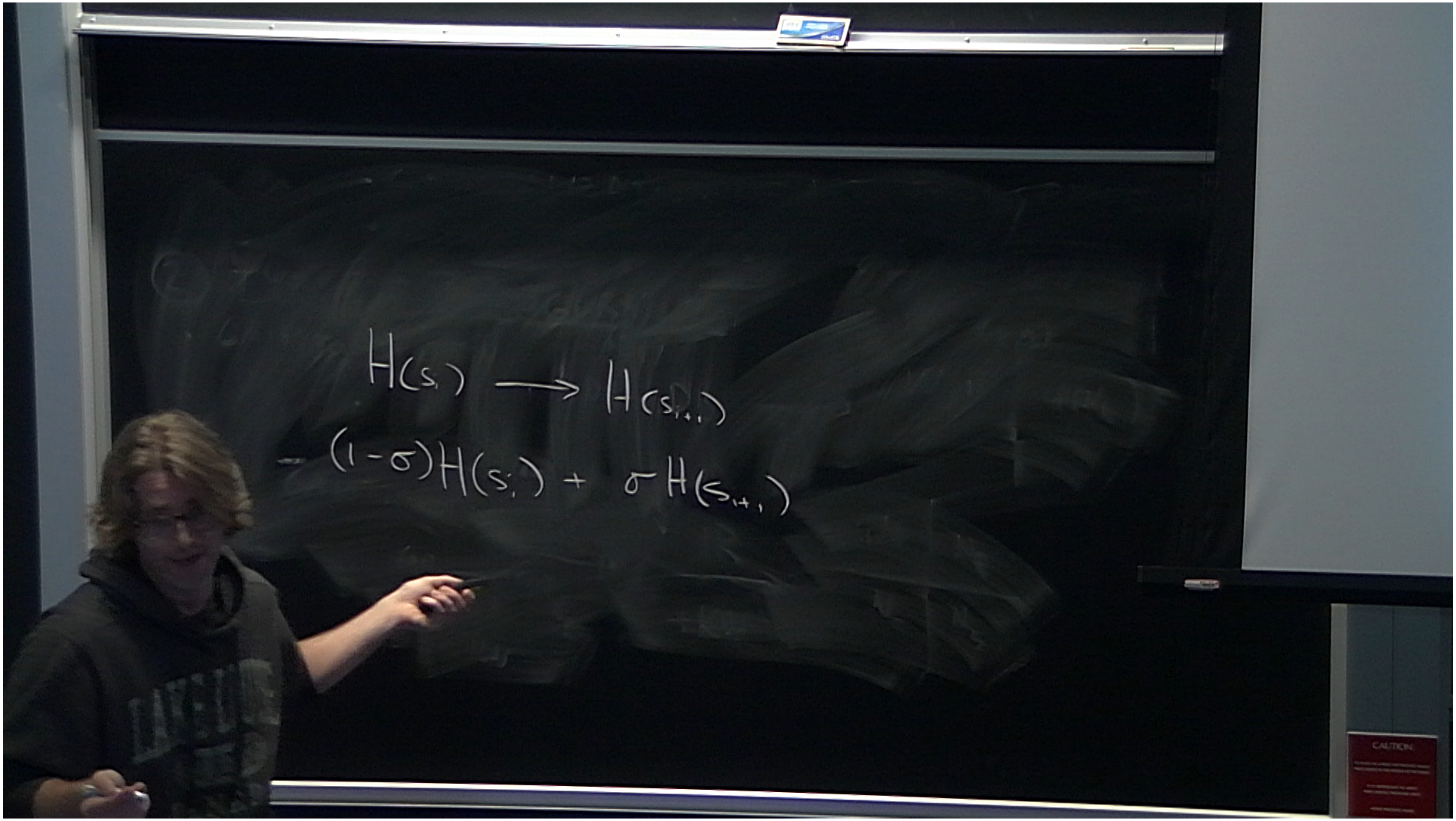
For $H(s)$ linear,

$$\left|\gamma(H+\delta s\dot{H}) - \gamma(H)\right| \leq 2\delta s\|\dot{H}\| \leq 4\delta s\|H\|.$$

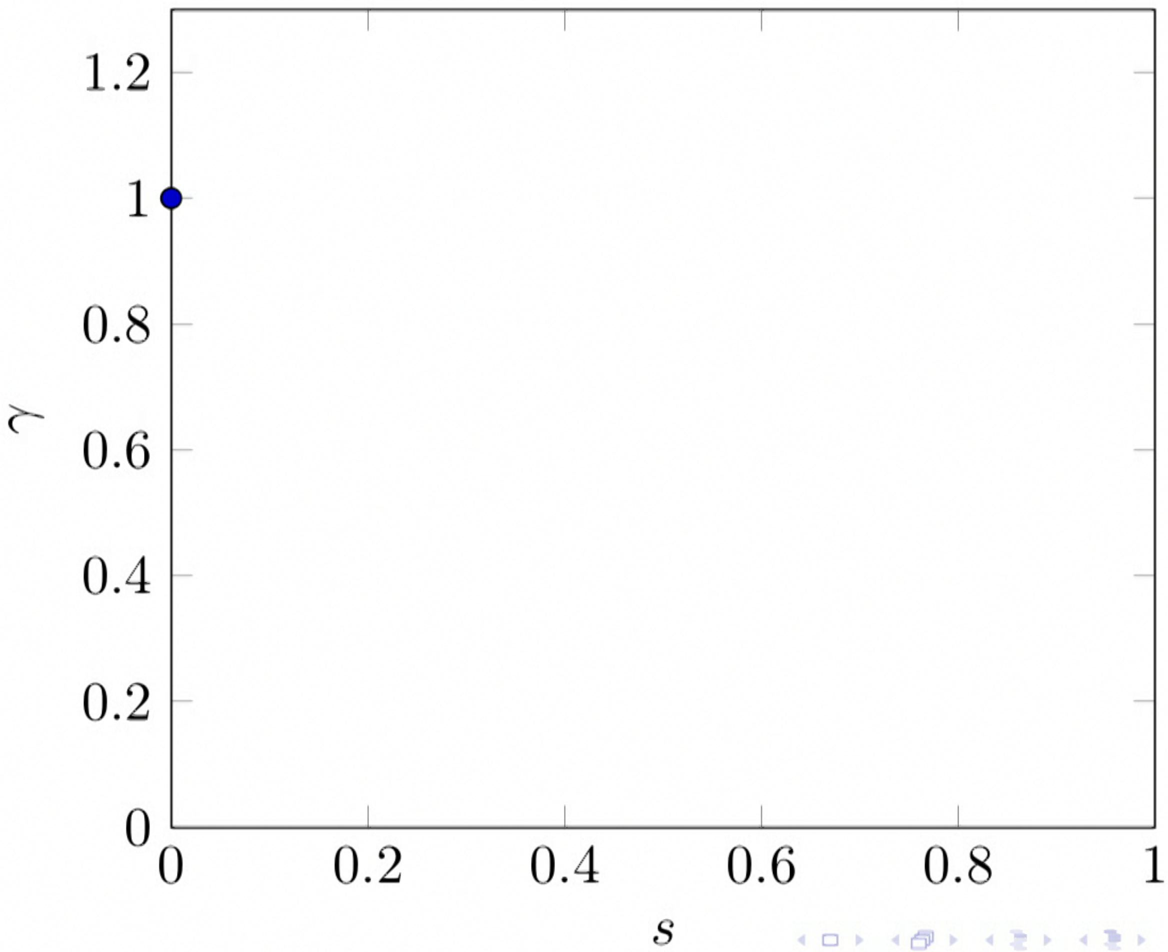
BAA

Algorithm 1 Bashful Adiabatic Algorithm

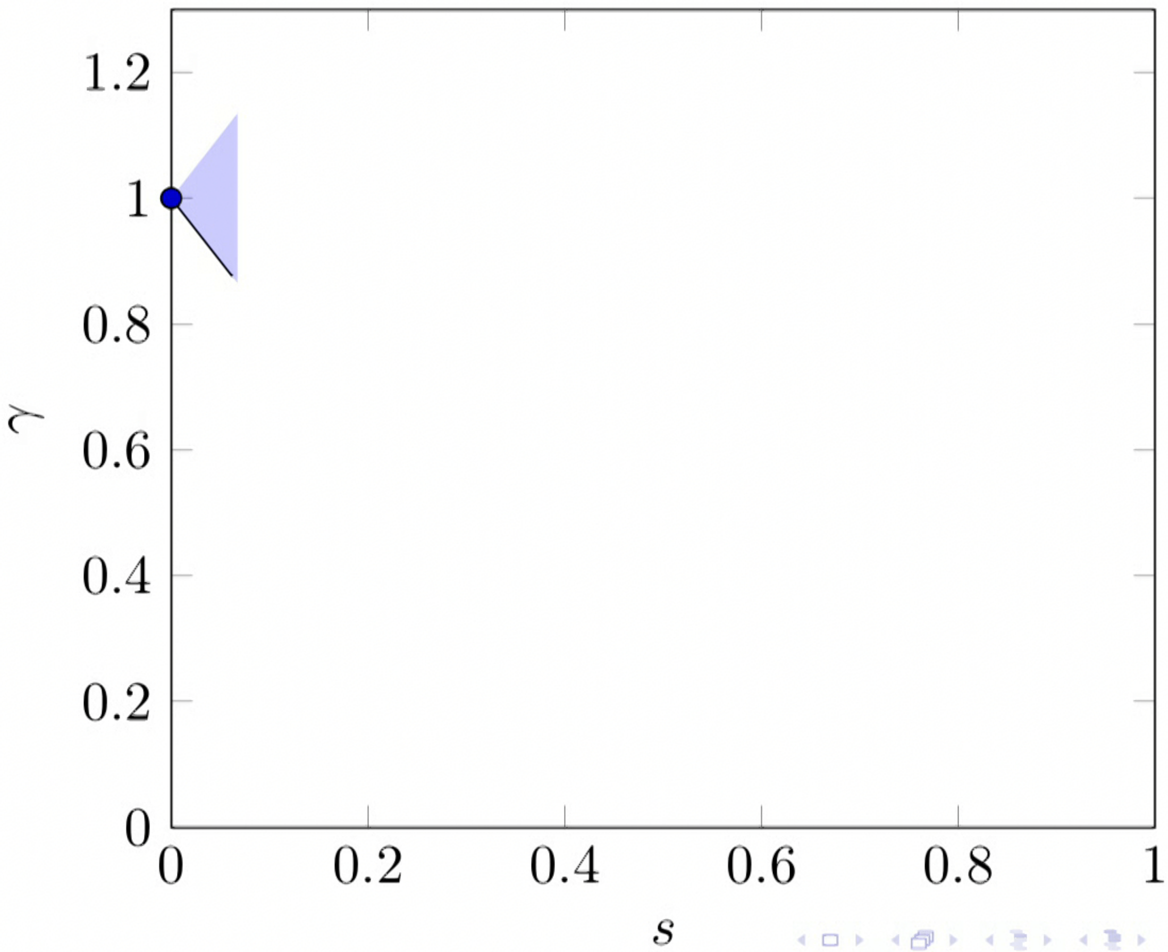
$s \leftarrow 0$
$\gamma \leftarrow \gamma(0)$
$\vec{\gamma} \leftarrow [(0, \gamma)]$
while $s < 1$ **do**
    $\delta s \leftarrow c_0 \gamma / 4$
    $\gamma \leftarrow \textsc{GetGap}(s, \delta s, \gamma, c_0)$
    $s \leftarrow s + \delta s$
    Append (s, γ) to $\vec{\gamma}$
return Adiabatically prepared state by $\vec{\gamma}$


$H(s_i) \longrightarrow H(s_{i+1})$
$(1-\sigma)H(s_i) + \sigma H(s_{i+1})$

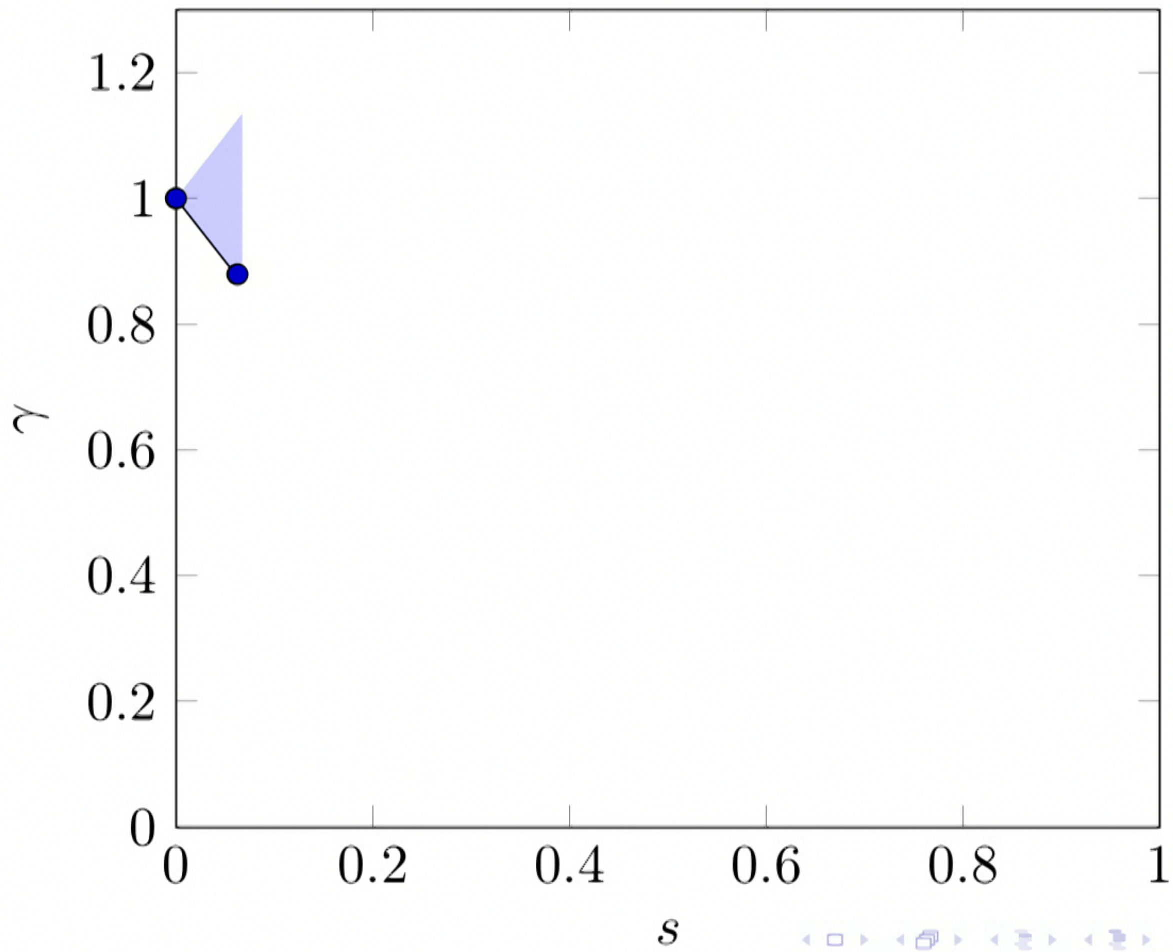
BAA behavior



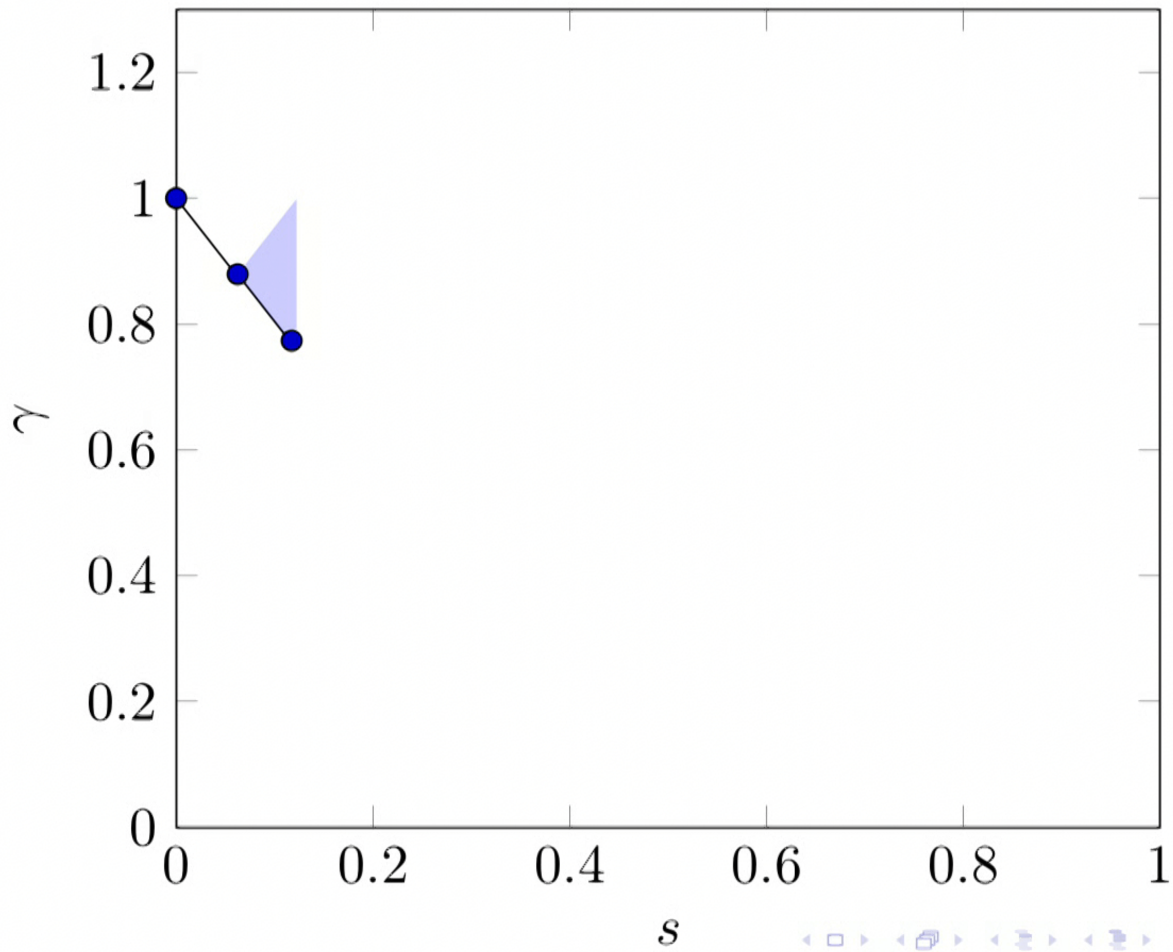
BAA behavior



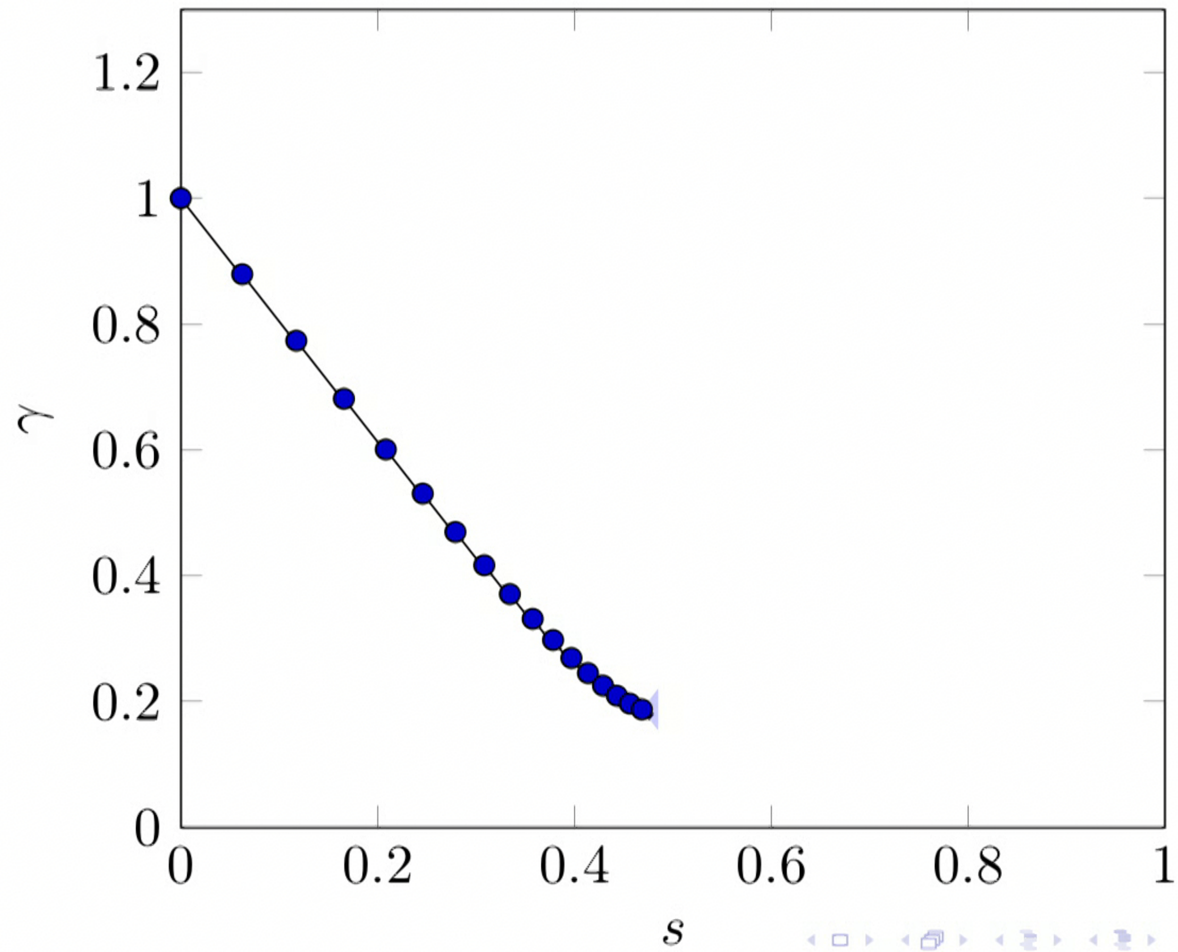
BAA behavior



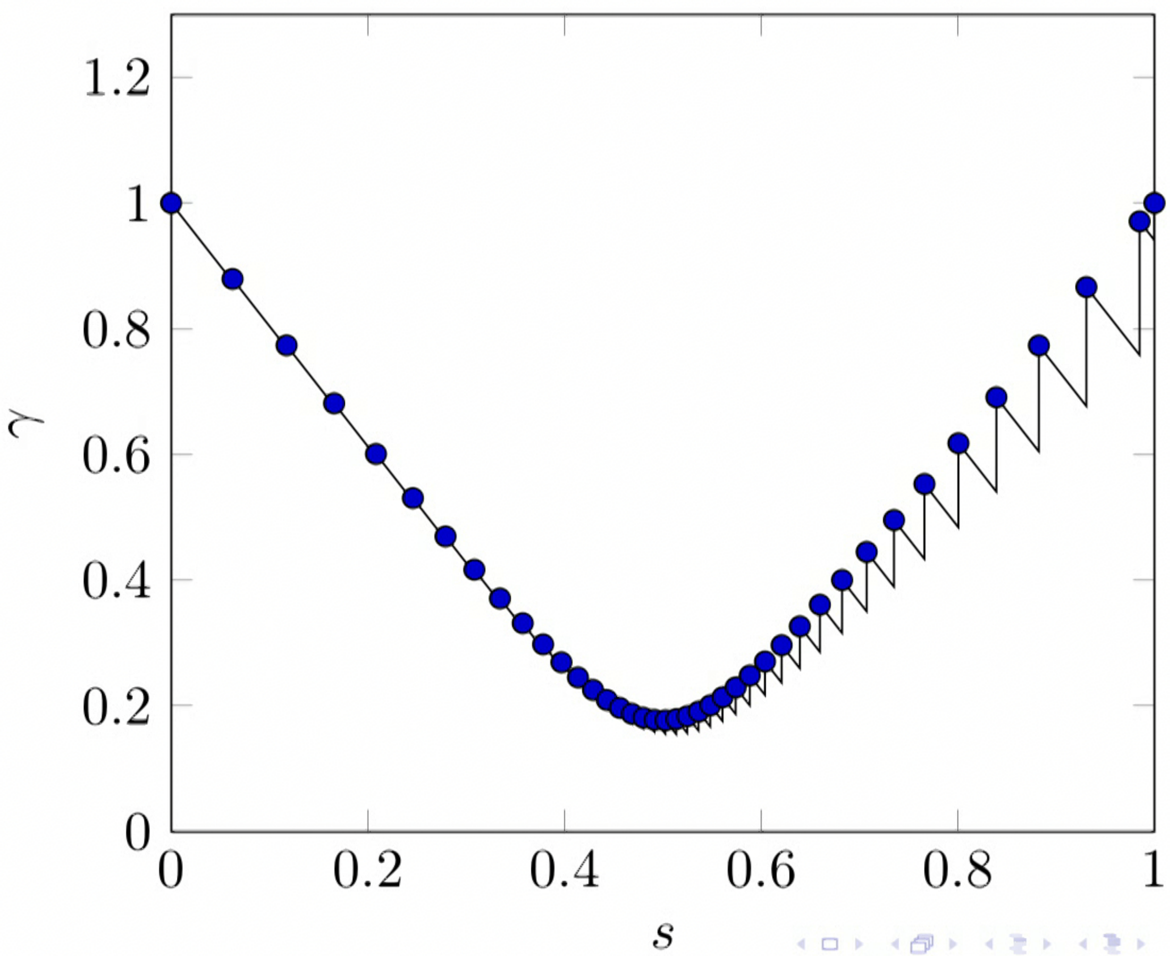
BAA behavior



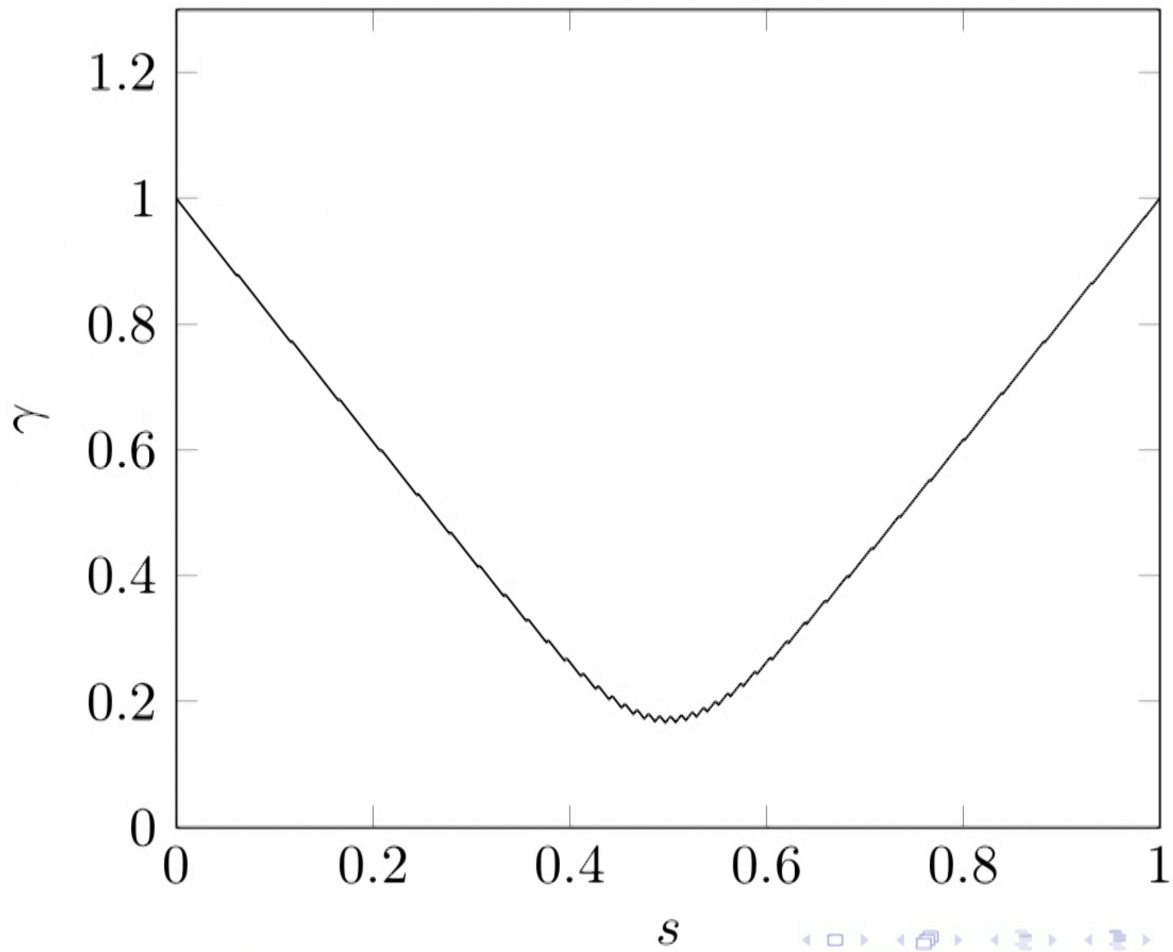
BAA behavior



BAA behavior



BAA behavior



Query Complexity

But, how many calls do we need?

Query Complexity

If GetGap obeys

$$\text{GetGap}(s) \geq \alpha|s - s_{\min}| + \beta\gamma_{\min}$$

then BAA requires at most

$$2\left\lceil \frac{\log\left(1+\frac{\alpha}{\beta\gamma_{\min}}\right)}{\log\left(1+\frac{c\alpha}{4}\right)} \right\rceil = O\left(\log\left(\gamma_{\min}^{-1}\right)\right)$$

queries to GetGap(s) to return a schedule such that

$$|\text{GetGap}(s) - \gamma(s)| \leq c\gamma(s).$$

Runtime

To see this, note that while $s \leq s_{\text{min}}$, BAA uses the recurrence

$$\begin{aligned} s_{i+1} &= s_i + \frac{\text{GETGAP}(s)}{4} \\ &\geq \frac{\alpha}{4}|s_i - s_{\text{min}}| + s_i + \beta\gamma_{\text{min}} \\ &= s_i\left(1 - \frac{\alpha}{4}\right) + \left(\beta\gamma_{\text{min}} + s_{\text{min}}\frac{\alpha}{4}\right) \end{aligned}$$

so the lower bound obeys

$$s_i \geq \left(\frac{4\beta\gamma_{\text{min}}}{\alpha} + s_{\text{min}}\right)\left(1 - (1 - \alpha/4)^i\right)$$

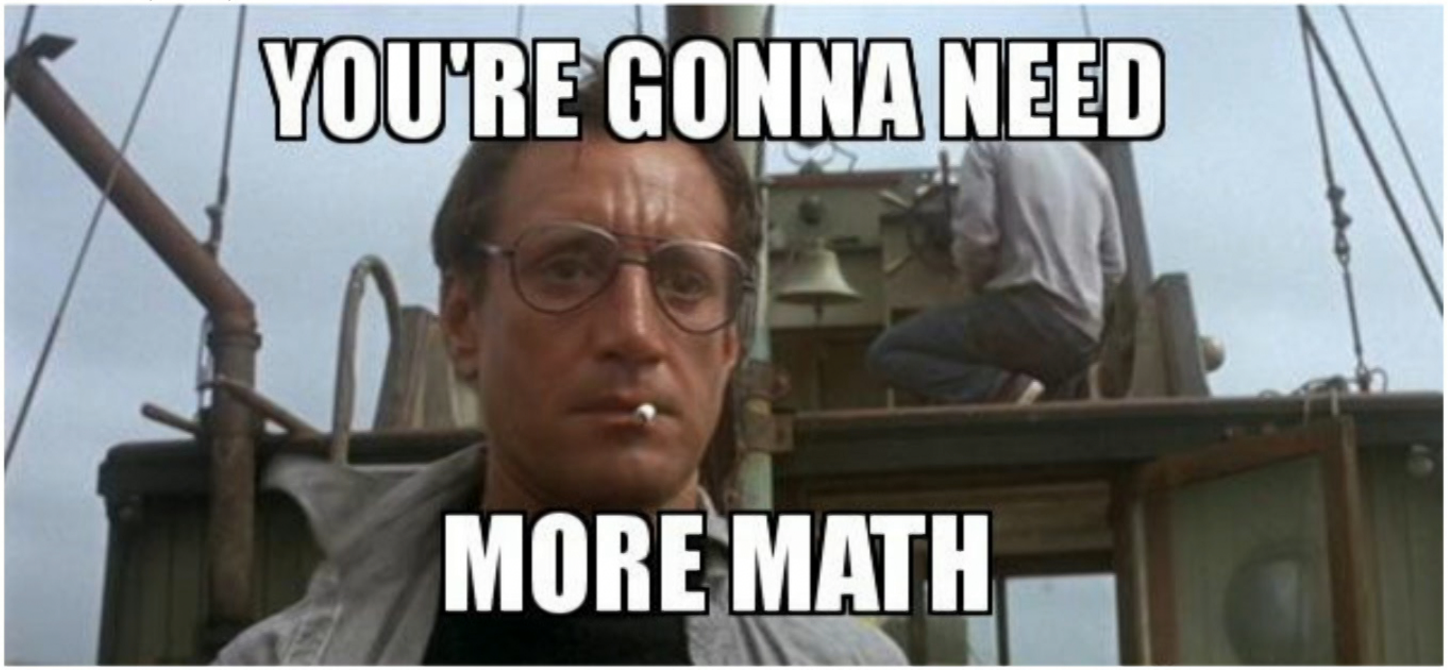
$\implies i \sim O(\log(1/\gamma_{\text{min}}))$ queries for $s_i \geq s_{\text{min}}$, similar for $s \in [s_{\text{min}}, 1]$

Okay... now what?

- So, uh, can we do this?

Okay... now what?

- So, uh, can we do this?



Cheeger inequalities

For a Hamiltonian $H(s) = L + W$ with ground state ϕ, where

$$L = \sum_u d_u |u\rangle\langle u| - \sum_{(u,v)\in E} |u\rangle\langle v|$$

corresponding to the weighted graph $G = (V, E)$ and W is diagonal, then

$$2h \geq \gamma(s) \geq \sqrt{h^2 + d^2} - d$$

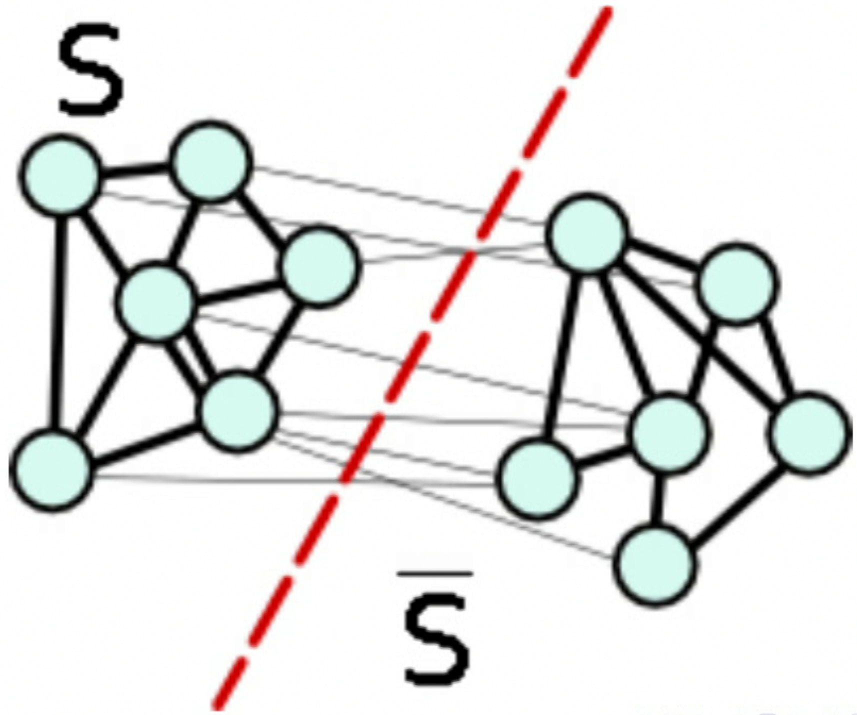
where $d = \max_u d_u$ and

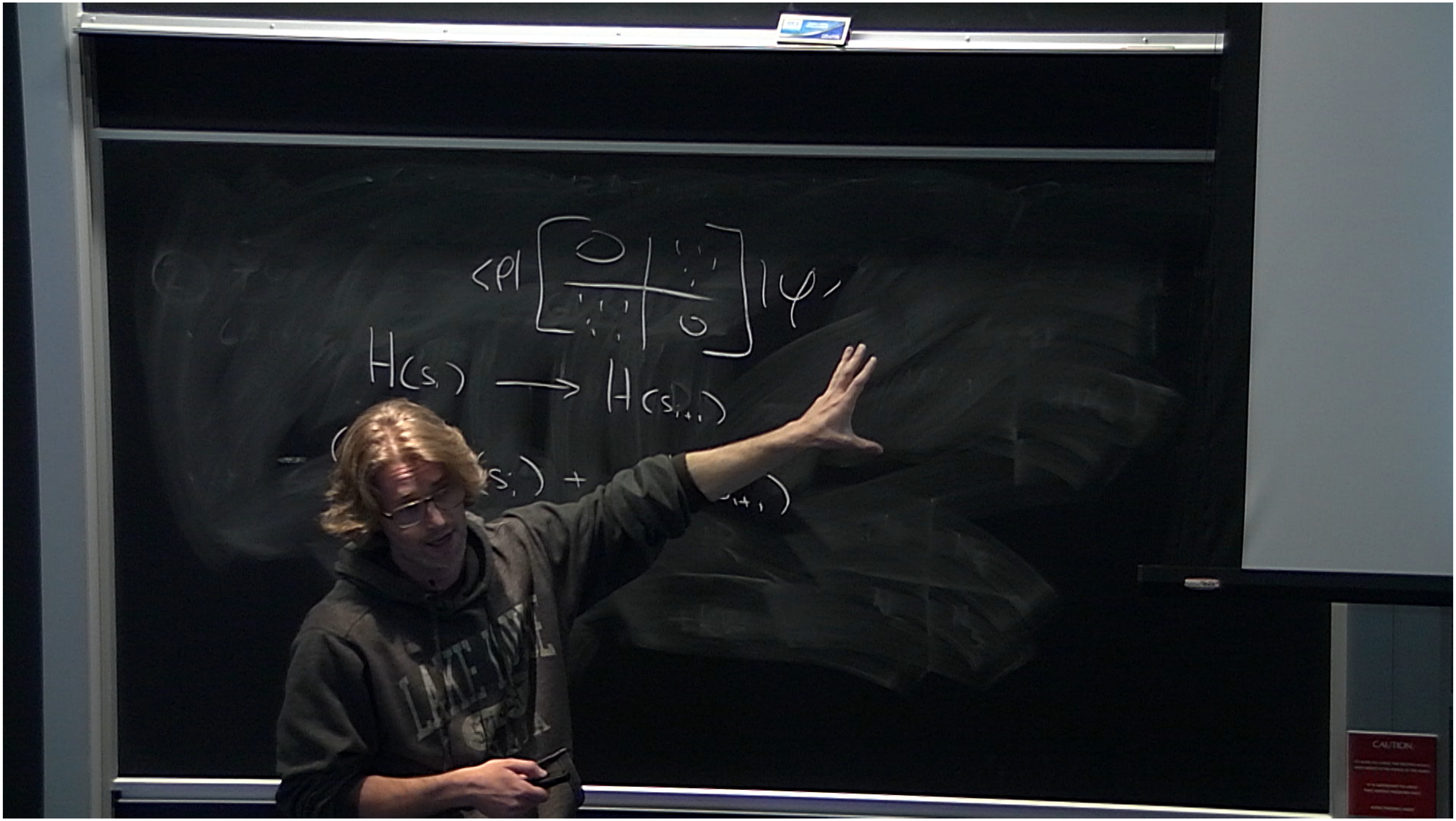
$$h = \min_{S\subset V} \max_{S'\in\{S,\overline{S}\}} \frac{\sum_{\substack{(u,v)\in E\\ u\in S, v\notin S}} \phi(u)\phi(v)}{\sum_{u\in S'} \phi^2(u)}.$$

(Similar inequalities hold for more general Hermitian L, W. [me, 2018])

Cheeger inequalities

$$h = \min_{S \subset V} \max_{S' \in \{S, \overline{S}\}} \frac{\sum_{\substack{\{u,v\} \in E \\ u \in S, v \notin S}} \phi(u)\phi(v)}{\sum_{u \in S'} \phi^2(u)}.$$





H(s_i) ⟶ H(s_{i+1})

Our problem

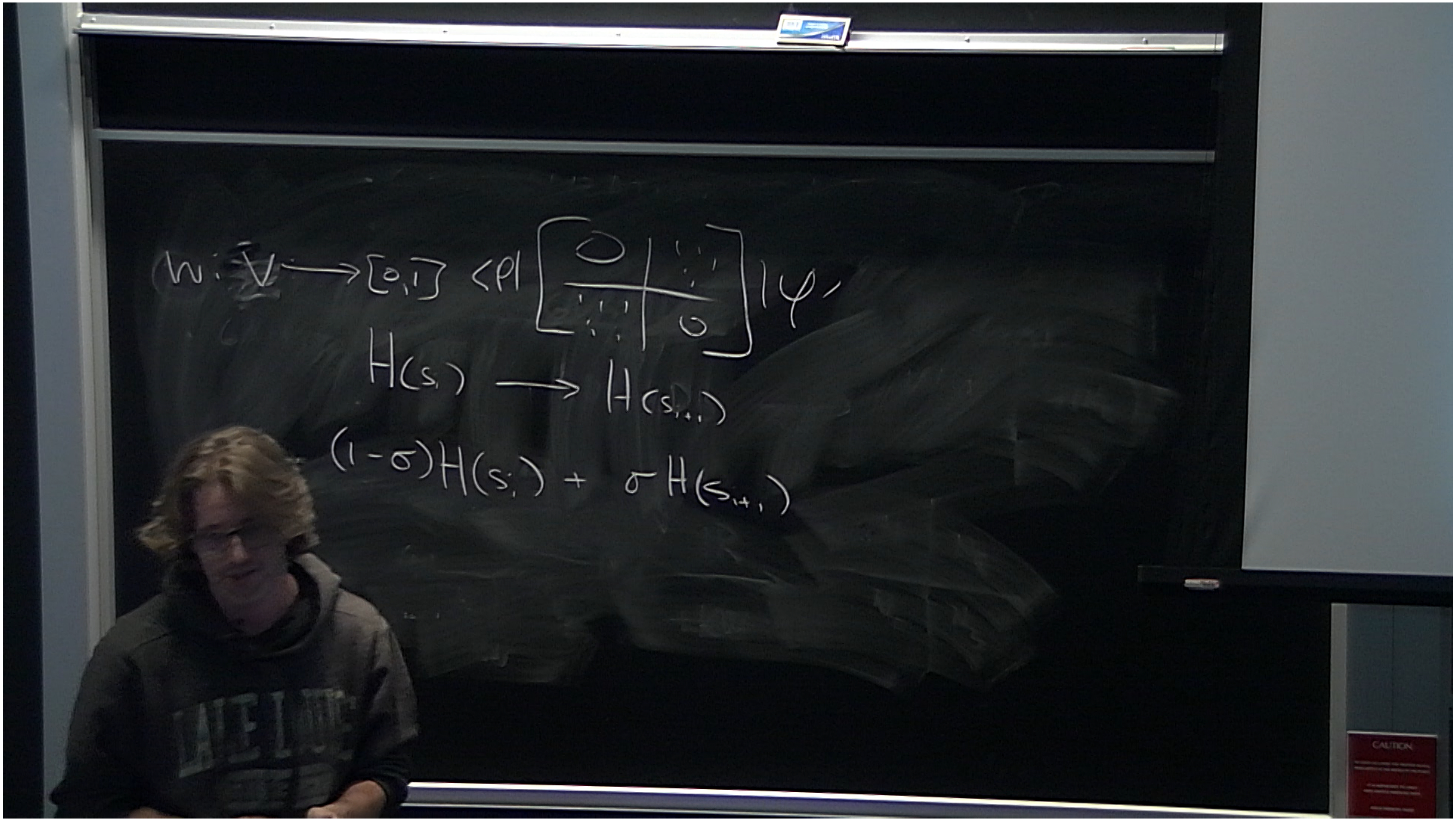
We only want to impose the following restrictions

1. $W(m) = 0$
2. $\lambda_{V-1}(W)/\lambda_1(W) \leq \kappa$
3. $\gamma(W)$ is finite.

We'll consider the toy problem of the complete graph Laplacian or

$$L = I - \frac{1}{d}\sum_{v,v'}|v\rangle\langle v'|$$

$d = V - 1.$


w: V → [0,1]
H(s_i) → H(s_i+1)
(1-σ)H(s_i) + σH(s_i+1)

Cheeger inequalities for our problem

All information is contained in ϕ_0! But... the inequality

$$2h \geq \gamma(s) \geq \sqrt{h^2 + d^2} - d$$

can be quadratically weak when $h << d$. (In our case $d = V - 1$ is optimally bad.) So, gotta tighten it up.

Our problem

When $H = L + W$ for the Laplacian $L = V - \sum_{v,v'} |v\rangle\langle v'|$, and

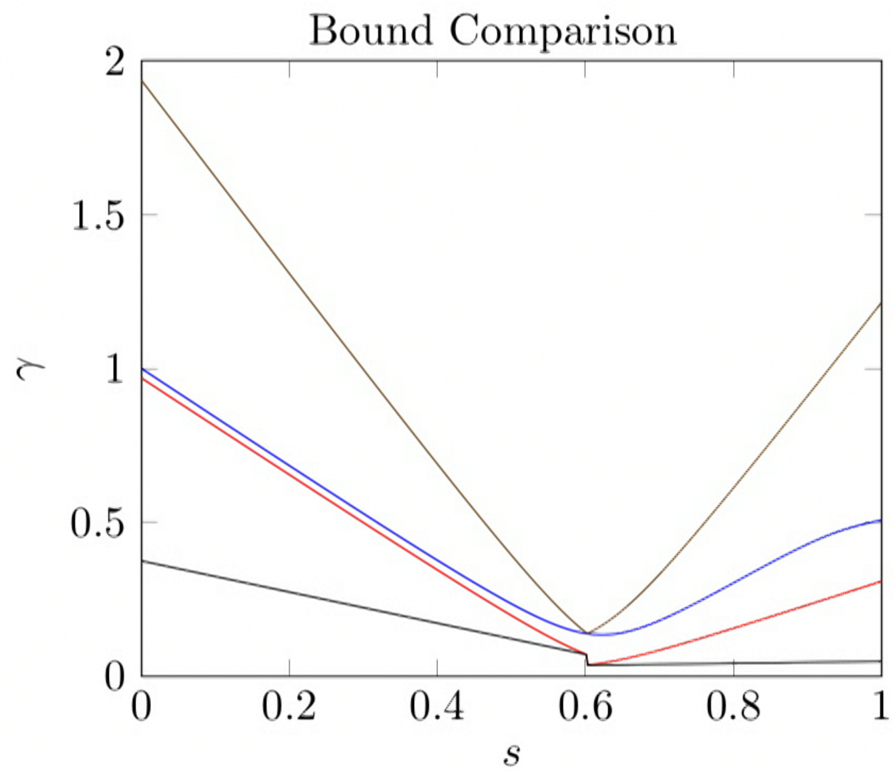
$$h_m = \min_m \max_{S' \in \{\{m\}, V \setminus \{m\}\}} \frac{\sum_{u \neq m} \phi(m)\phi(u)}{\sum_{u \in S'} \phi^2(m)}$$
$$= \min_m \frac{\phi(m)\left(\|\phi\|_1 - \phi(m)\right)}{\min(\phi^2(m), 1 - \phi^2(m))}$$
$$= \min_m \left(\frac{\|\phi\|_1}{\phi(m)} - 1\right) \max\left(1, \frac{\phi^2(m)}{1 - \phi^2(m)}\right) \geq \frac{\|\phi\|_1}{\phi(m)} - 1.$$

Then,

$$2h_m \geq \gamma(s) \geq \frac{h_m}{\kappa}.$$

So, everything here is within a constant factor.

BAA behavior



Our Problem

Can we learn it?
Eigenvectors satisfy

$$(V + W(u) - \lambda)\phi(u) = \sum_{v} \phi(v)$$

so

$$(V - \lambda_0)\phi(m) = \|\phi\|_1$$

or

$$V - \lambda_0 = \frac{\|\phi\|_1}{\phi(m)} = h_m + 1$$

whenever $\phi_m \leq 1/\sqrt{2}$.

Our Problem

Also,

$$
\begin{aligned}
(V + W(u) - \lambda_0)\phi(u) &= \|\phi\|_1 \\
\phi(u) &= \frac{\|\phi\|_1}{V + W(u) - \lambda_0} \\
\|\phi\|_1 &= \sum_u \frac{\|\phi\|_1}{V + W(u) - \lambda_0} \\
1 &= \sum_u \frac{1}{V + W(u) - \lambda_0} \\
&= \sum_u \frac{1}{W(u) + h_m + 1}.
\end{aligned}
$$

so finding the root of this expression will give us a bound on h_m.

GetGap

With probability $1-p$, GetGap requires at most $O\left(V^{2/3}\log(1/p)\right)$ queries to W to create a function FindRoot$(\cdot)$ such that

$$|\text{FindRoot}(s,\delta s,h,c)-h_m|\leq ch_m$$

where $h_m \geq V^{2/3}$ and

$$\sum_u \frac{1}{W_u+h_m+1}-1=0$$

for any constants κ, c. (Hoeffding)

Our oracle

Algorithm 2 Complete graph oracle

Require: A probability of failure p, the number of vertices V, the cost function W

Global $S_{\min} \leftarrow [1, 1]$

function $\textsc{GetGap}(s, \delta s, \gamma, c)$

  if $s \geq S_{\min}[0]$ **then return** $\textsc{FinishSchedule}(s, \delta s, \gamma, c)$

  $h \leftarrow \frac{1}{1-c}\left(\frac{\gamma}{1-s} + 1\right) - 1$

  if $s = 0$ **then** $h \leftarrow V - 1$

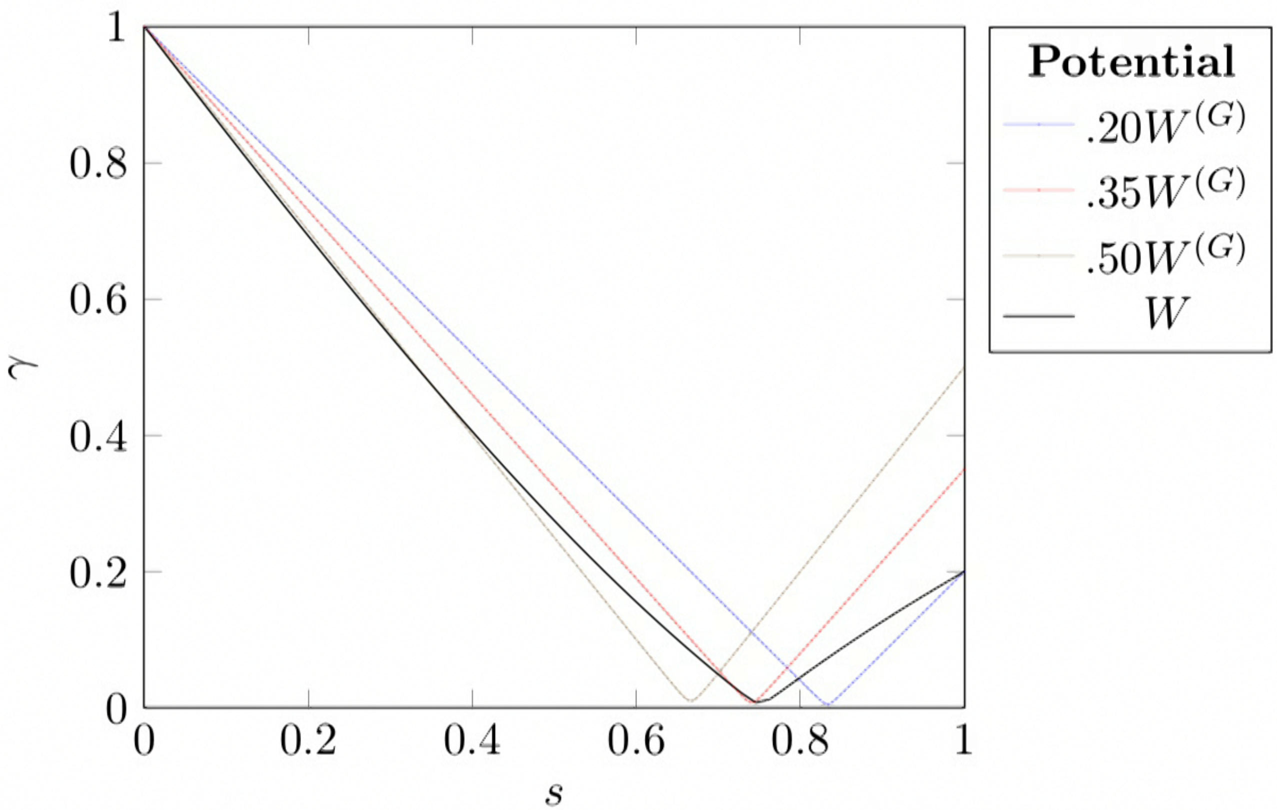
  $h \leftarrow \textsc{FindRoot}(s, \delta s, h, c)$

  if $h = 0$ **then return** $\textsc{FinishSchedule}(s, \delta s, \gamma, c)$

  return $(1 - s - \delta s)\left((1 - c)(h + 1) - 1\right)$

GetGap

$$H = (1 - s)L + sW$$



FinishSchedule

$\kappa = \lambda_{V-1}(W)/\lambda_1(W)$

- FinishSchedule just follows the lower linear envelope discussed previously. This can introduce an additional $O((\kappa-1)^{2/3}V^{1/6})$ queries since it doesn't obey the recurrence.

FinishSchedule

$\kappa = \lambda_{V-1}(W)/\lambda_1(W)$

- FinishSchedule just follows the lower linear envelope discussed previously. This can introduce an additional $O((\kappa-1)^{2/3}V^{1/6})$ queries since it doesn't obey the recurrence.
- Total queries to GetGap by BAA is then at most $\widetilde{O}(\log(1/\gamma_{\min}) + (\kappa-1)^{2/3}V^{1/6})$

FinishSchedule

$\kappa = \lambda_{V-1}(W)/\lambda_1(W)$

- FinishSchedule just follows the lower linear envelope discussed previously. This can introduce an additional $O((\kappa-1)^{2/3}V^{1/6})$ queries since it doesn't obey the recurrence.
- Total queries to GetGap by BAA is then at most $\widetilde{O}(\log(1/\gamma_{\min}) + (\kappa-1)^{2/3}V^{1/6})$
- Total samples to build oracle is at most $O(1+(\kappa-1)^{2/3}V^{2/3}\log(1/p))$
- Total runtime of adiabatic process following this bound is then $\widetilde{O}\left(\frac{1}{\epsilon\gamma_{\min}}\log(1/\gamma_{\min} + (\kappa-1)^{2/3}\frac{V^{1/6}}{\gamma_{\min}\epsilon}\right) = O(\frac{\sqrt{V}}{\epsilon} + \frac{(\kappa-1)^{2/3}}{\epsilon}V^{2/3})$

FinishSchedule

$\kappa = \lambda_{V-1}(W)/\lambda_1(W)$

- FinishSchedule just follows the lower linear envelope discussed previously. This can introduce an additional $O((\kappa-1)^{2/3}V^{1/6})$ queries since it doesn't obey the recurrence.
- Total queries to GetGap by BAA is then at most $\widetilde{O}(\log(1/\gamma_{\min}) + (\kappa-1)^{2/3}V^{1/6})$
- Total samples to build oracle is at most $O(1+(\kappa-1)^{2/3}V^{2/3}\log(1/p))$
- Total runtime of adiabatic process following this bound is then $\widetilde{O}\left(\frac{1}{\epsilon\gamma_{\min}}\log(1/\gamma_{\min} + (\kappa-1)^{2/3}\frac{V^{1/6}}{\gamma_{\min}\epsilon}\right) = O(\frac{\sqrt{V}}{\epsilon} + \frac{(\kappa-1)^{2/3}}{\epsilon}V^{2/3})$
- Total runtime of algorithm BAA is therefore $O\left(\left(\sqrt{V}+(\kappa-1)^{2/3}V^{2/3}\right)\log(\sqrt{V})/\epsilon\right)$ and produces a state $\||\psi\rangle - |\phi\rangle\| \leq \epsilon$ with probability $\Omega\left(1-1/\sqrt{V}\right)$.

Optimization

Algorithm 3 Optimize

function OPTIMIZE(W)
    $\delta \leftarrow \frac{3}{2}\log_V(\frac{3}{2})$
    $N \leftarrow \frac{\log(V^{2/3}\log(\sqrt{V}))}{1+\log(\epsilon^{-1})}$
    for $i \in [\![\frac{1}{4\delta}]\!]$ **do**
        $\kappa \leftarrow 1 + V^{i\delta-\frac{1}{4}}$. ▷ Guess $\kappa(W) \leq \kappa$
        $\Psi \leftarrow [\text{BAA}(\text{GETGAP}_\kappa)]_{i=0}^{N}$ ▷ Collect the results of BAA
        $\Psi \leftarrow$ MEASURE(Ψ)
        if $W_b = 0$ for any $b \in \Psi$ **then return** $|b\rangle$
    $\kappa \leftarrow 1/\gamma(1)$
    return BAA(GETGAP$_\kappa$)

Expected runtime matches runtime up to log factors.
Probability of failure $p \leq \frac{1}{V^{2/3}}$

TLFA (Too long fell asleep)

Lessons we've learned:

- Not at all likely that heuristic schedules can give reliable results

TLFA (Too long fell asleep)

Lessons we've learned:

- Not at all likely that heuristic schedules can give reliable results
- BAA can obtain a reliable quantum advantage on general problems
- QAO is a special purpose algorithm...
- Design hardware to match the algorithm, not try to find algorithm that matches hardware.

Outlook

- Oracle can be improved, but probably no asymptotic advantage

Outlook

- Oracle can be improved, but probably no asymptotic advantage
- Generalization to multiple marked states should be "easy"

Outlook

- Oracle can be improved, but probably no asymptotic advantage
- Generalization to multiple marked states should be "easy"
- Generalize to other graphs

Outlook

- Oracle can be improved, but probably no asymptotic advantage
- Generalization to multiple marked states should be "easy"
- Generalize to other graphs
- Remove the restriction $\min_m W_m \approx 0$

Outlook

- Oracle can be improved, but probably no asymptotic advantage
- Generalization to multiple marked states should be "easy"
- Generalize to other graphs
- Remove the restriction $\min_m W_m \approx 0$
- Looks like this is doable with $O(\log(V))$ fixed X-basis measurements
- Other ideas, use this for faster MC simulations?

Outlook

- Oracle can be improved, but probably no asymptotic advantage
- Generalization to multiple marked states should be "easy"
- Generalize to other graphs
- Remove the restriction $\min_m W_m \approx 0$
- Looks like this is doable with $O(\log(V))$ fixed X-basis measurements
- Other ideas, use this for faster MC simulations?
- Classical SAT solving?