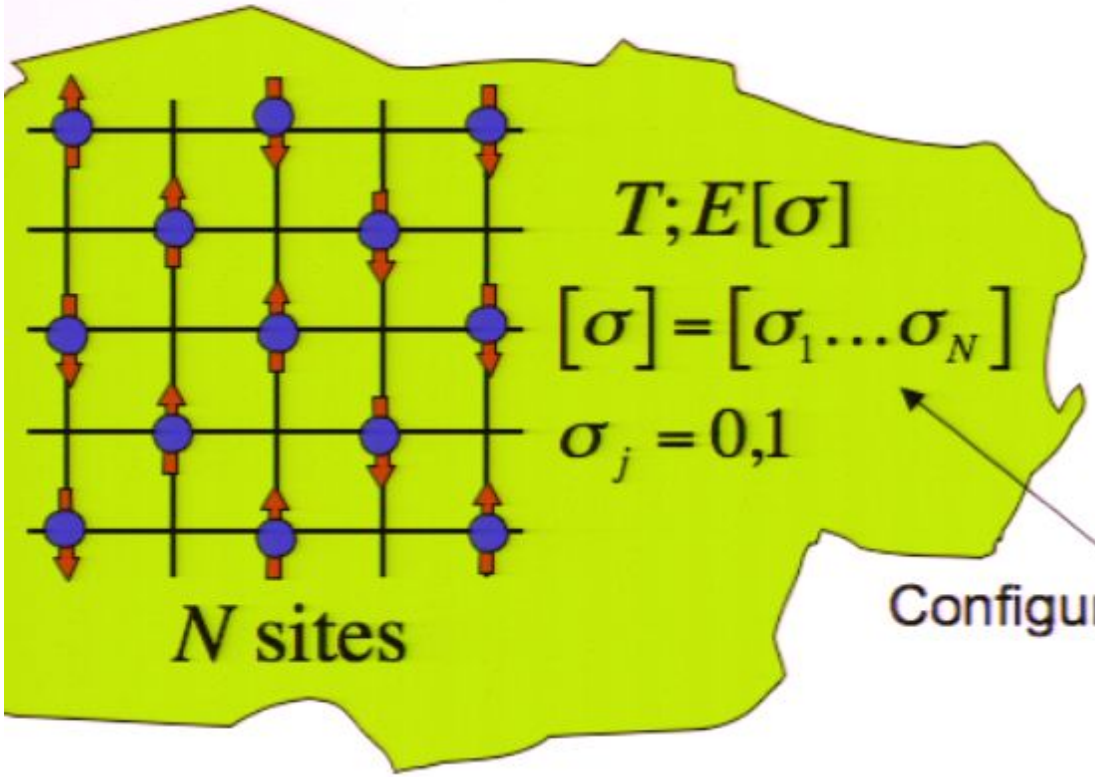
Title: Quantum Simulations of Classical Annealing Processes

Date: May 02, 2008 09:30 AM

URL: http://pirsa.org/08050033

Abstract: Quantum computers provide new resources to solve combinatorial optimization problems (COPs). Using techniques borrowed from quantum information theory, I will present a quantum algorithm that simulates classical annealing processes, where the (quantum) annealing rate greatly outperforms other classical methods like Markov chain Monte-Carlo based algorithms. Our quantum algorithm provides quadratic speedups to find both, the solution to particular instances of COPs, and the preparation of (quantum) Gibbs\' states.

Simulated Annealing and Markov Chain Monte Carlo (MCMC)



The solution to the COP is determined by the state that minimizes E (cost function)

$$E : [\sigma_1 \ldots \sigma_N] \rightarrow \mathbf{R}$$

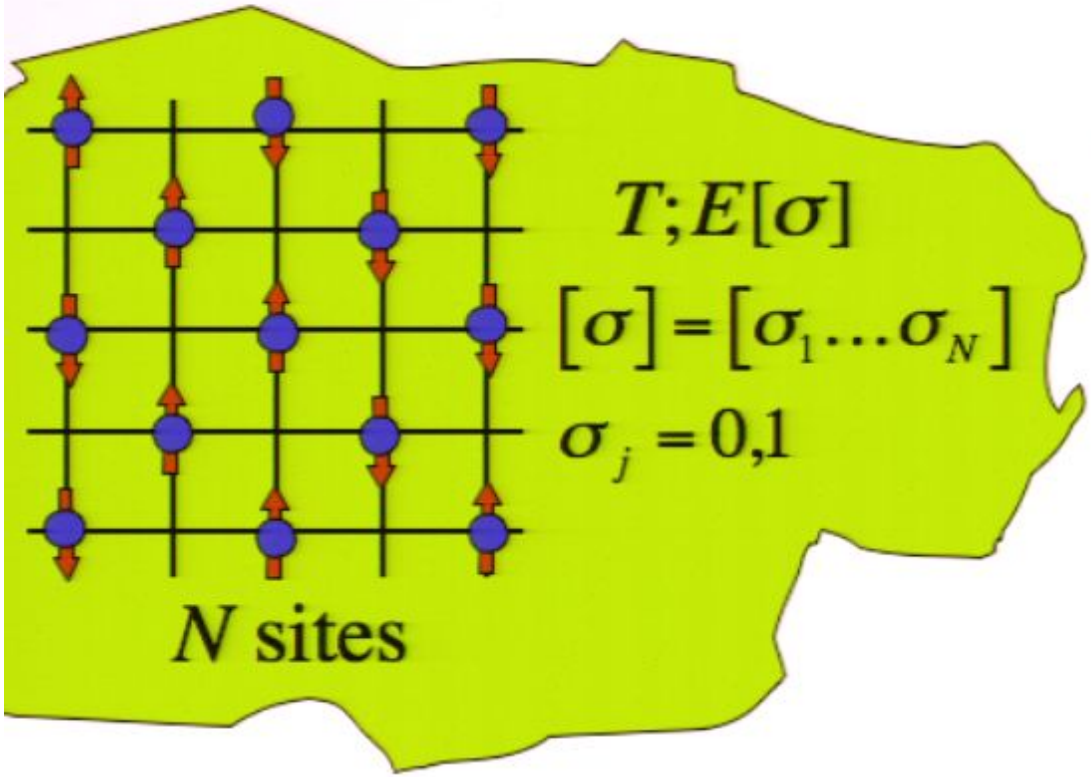
Configuration space: $d = 2^N$

Example: Finding the ground state (energy) of the Ising spin glass (NP hard)

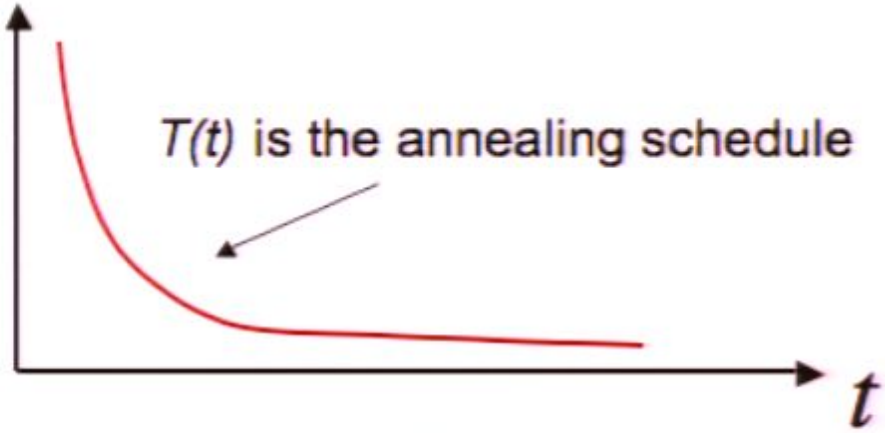
$$E[\sigma] = \sum_{<i,j>} J_{ij}\sigma_i\sigma_j + \sum_i B_i\sigma_i$$

The cost function is the energy of the configuration

Simulated Annealing and Markov Chain Monte Carlo (MCMC)



The idea is to find the ground state (solution) by decreasing T slowly



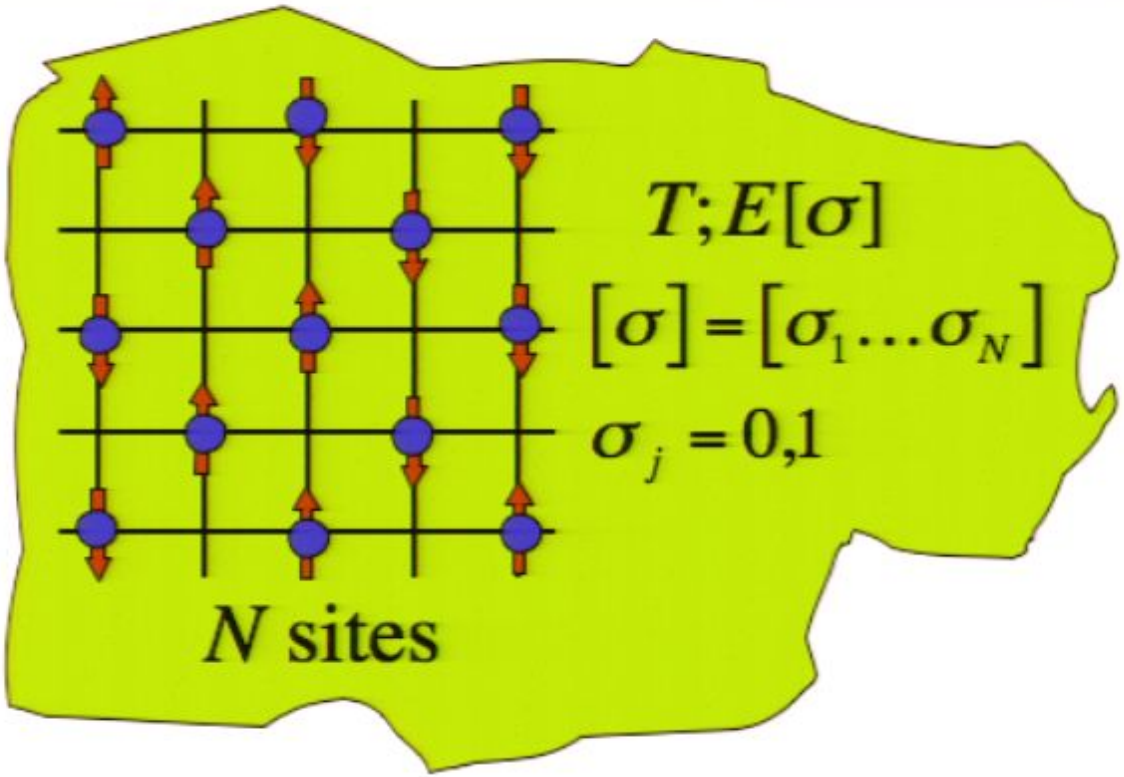
Ground state space

$$A_0 = \{[\sigma] / E[\sigma] \text{ is minimum}\}$$

$$\vec{g}_0 = \frac{1}{\# A_0} \sum_{\sigma \in A_0} [\sigma]$$

$$\partial_t T(t) << 1 \Rightarrow$$ The final state is in A_0 with high probability

Simulated Annealing and Markov Chain Monte Carlo (MCMC)



In equilibrium:

$$\beta = 1/T$$

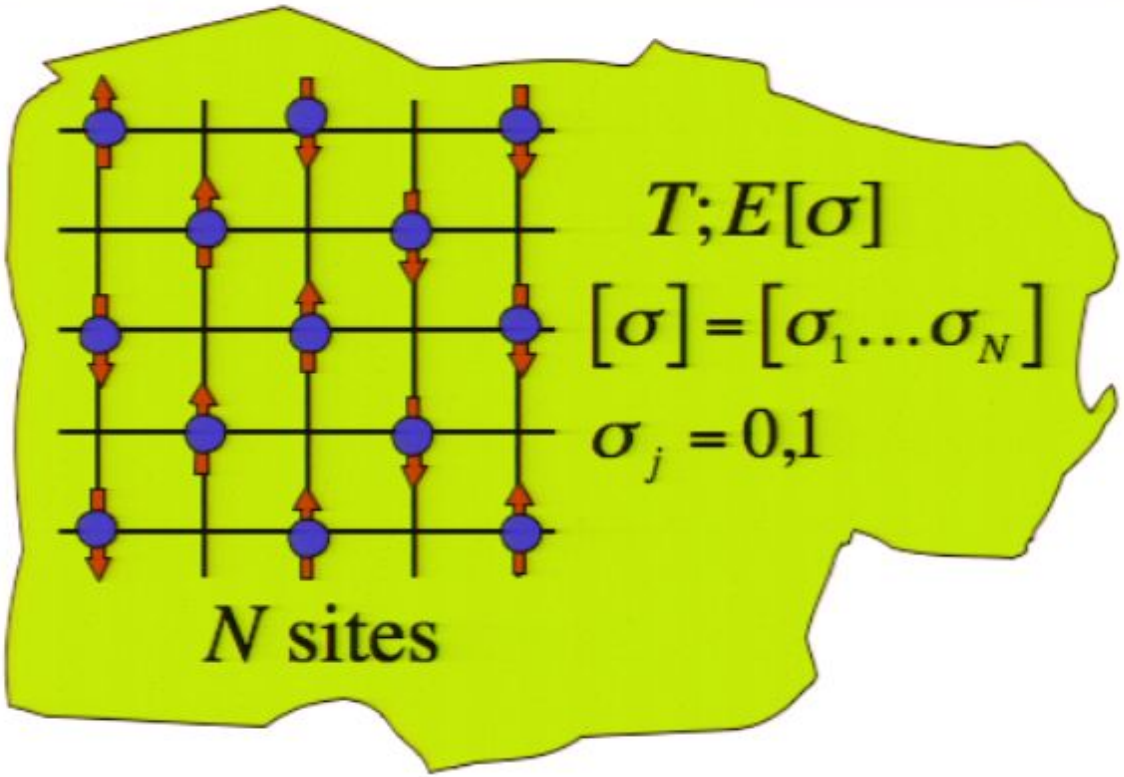
$$\vec{\pi}(\beta) = \frac{1}{Z}(e^{-\beta E[00...00]}, e^{-\beta E[00...01]}, \ldots, e^{-\beta E[11...11]})$$

Gibbs state

$$Z = \sum_{\sigma} e^{-\beta E[\sigma]}$$

Partition function

Simulated Annealing and Markov Chain Monte Carlo (MCMC)



In equilibrium:

$$\vec{\pi}(\beta) = \frac{1}{Z}(e^{-\beta E[00\ldots00]}, e^{-\beta E[00\ldots01]}, \ldots, e^{-\beta E[11\ldots11]})$$

If $\beta >> 1$ this is close to $\vec{g}_0$, the solution state

Simulated Annealing and Markov Chain Monte Carlo (MCMC)

1- Need to choose a transition rule

probability of transition:

$$p_{\sigma\sigma'} \rightarrow S(\beta) = \begin{bmatrix} p_{00\ldots0,0\ldots00} & p_{00\ldots0,00\ldots1} & \cdots & p_{00\ldots0,11\ldots1} \\ p_{00\ldots1,00\ldots0} & \vdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ p_{11\ldots1,00\ldots0} & \vdots & \cdots & p_{11\ldots1,11\ldots1} \end{bmatrix}$$

Transition matrix

$$\vec{\pi}(\beta)S(\beta) = \vec{\pi}(\beta),$$

$$\lim_{q\to\infty} \vec{u}[S(\beta)]^q = \vec{\pi}(\beta)$$

Example: Metropolis rule

$$p_{\sigma\sigma'} = \min\{1, e^{-\beta(E[\sigma']-E[\sigma])}\} \quad \text{for} \quad |\sigma'-\sigma| \le 1$$

$$p_{\sigma\sigma'} = 0 \quad \text{for} \quad |\sigma'-\sigma| > 1$$

Simulated Annealing and Markov Chain Monte Carlo (MCMC)

Some properties of the transition matrix:

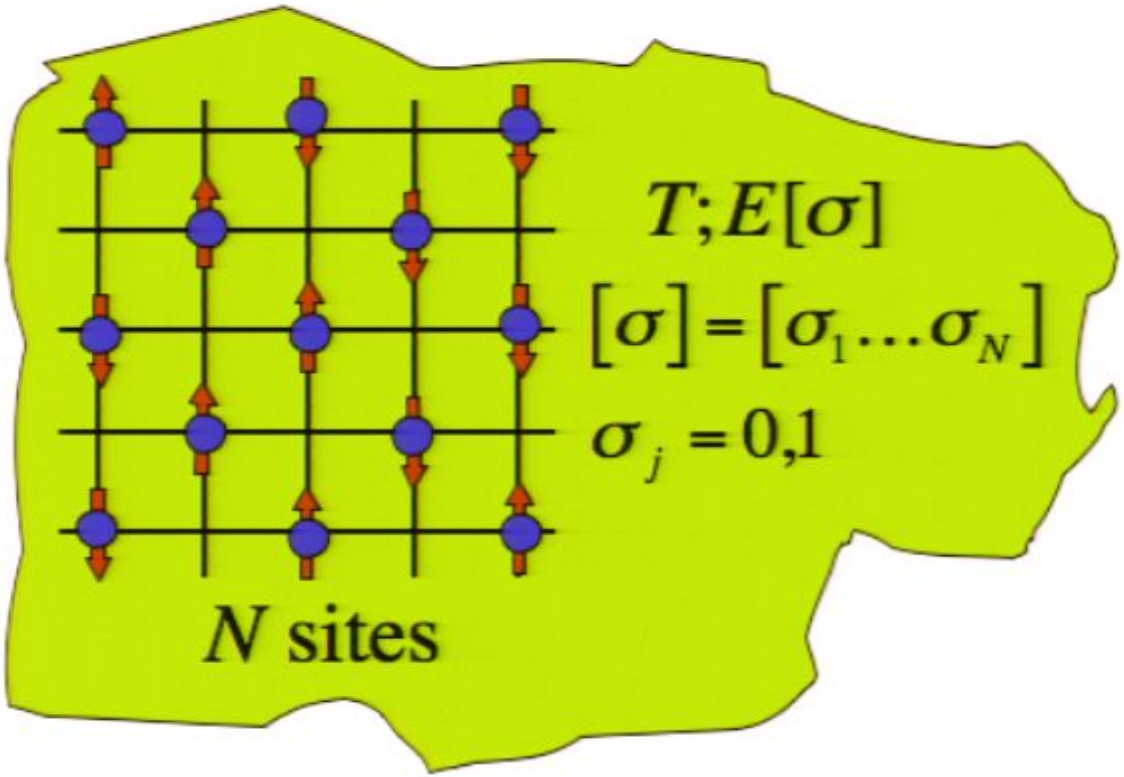
$$\sum_{[\sigma']} p_{\sigma\sigma'} = 1$$

Detailed balance : $p_{\sigma\sigma'}\pi_{\sigma} = p_{\sigma'\sigma}\pi_{\sigma'}$

Eigenvalues : $\lambda_1 = 1 > \lambda_2 \geq \ldots \geq \lambda_d \geq 0$

Mixing time : $\frac{1}{\Delta(\beta)}$; $\Delta(\beta) = 1 - \lambda_2(\beta)$ spectral gap

Simulated Annealing and Markov Chain Monte Carlo (MCMC)



In equilibrium:

$$\vec{\pi}(\beta) = \frac{1}{Z}(e^{-\beta E[00\ldots00]}, e^{-\beta E[00\ldots01]}, \ldots, e^{-\beta E[11\ldots11]})$$

If $\beta >> 1$ this is close to $\vec{g}_0$, the solution state

Simulated Annealing and Markov Chain Monte Carlo (MCMC)

1- Need to choose a transition rule

probability of transition:

$$p_{\sigma\sigma'} \rightarrow S(\beta) = \begin{bmatrix} p_{00\ldots0,0\ldots00} & p_{00\ldots0,00\ldots1} & \cdots & p_{00\ldots0,11\ldots1} \\ p_{00\ldots1,00\ldots0} & \vdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ p_{11\ldots1,00\ldots0} & \vdots & \cdots & p_{11\ldots1,11\ldots1} \end{bmatrix}$$

Transition matrix

$$\vec{\pi}(\beta)S(\beta) = \vec{\pi}(\beta),$$

$$\lim_{q\rightarrow\infty} \vec{u}[S(\beta)]^q = \vec{\pi}(\beta)$$

Example: Metropolis rule

$$p_{\sigma\sigma'} = \min\{1, e^{-\beta(E[\sigma']-E[\sigma])}\} \text{ for } |\sigma'-\sigma| \leq 1$$

$$p_{\sigma\sigma'} = 0 \text{ for } |\sigma'-\sigma| > 1$$

Simulated Annealing and Markov Chain Monte Carlo (MCMC)

Some properties of the transition matrix:

$$\sum_{[\sigma']} p_{\sigma\sigma'} = 1$$

Detailed balance : $p_{\sigma\sigma'}\pi_{\sigma} = p_{\sigma'\sigma}\pi_{\sigma'}$

Eigenvalues : $\lambda_1 = 1 > \lambda_2 \geq \ldots \geq \lambda_d \geq 0$

Mixing time : $\frac{1}{\Delta(\beta)}$; $\Delta(\beta) = 1 - \lambda_2(\beta)$ spectral gap

Simulated Annealing and Markov Chain Monte Carlo (MCMC)

- Need to choose an annealing schedule

$$\vec{u}_0 S(0) S(\delta\beta) \ldots S(P\delta\beta) = \vec{u}_f$$

Uniform distribution

Final distribution

$$\left.\begin{array}{l} \beta_f = P\delta\beta >> 1 \\ \delta\beta << 1 \end{array}\right\} \| \vec{u}_f - \vec{g}_0 \| << 1$$

The implementation complexity of the SA is given by P

Convergence rates for Simulated Annealing

$$\delta\beta = \min_\beta \Delta(\beta) = \Delta$$

$$\beta_f = \gamma^{-1} \log(d/\varepsilon^2)$$

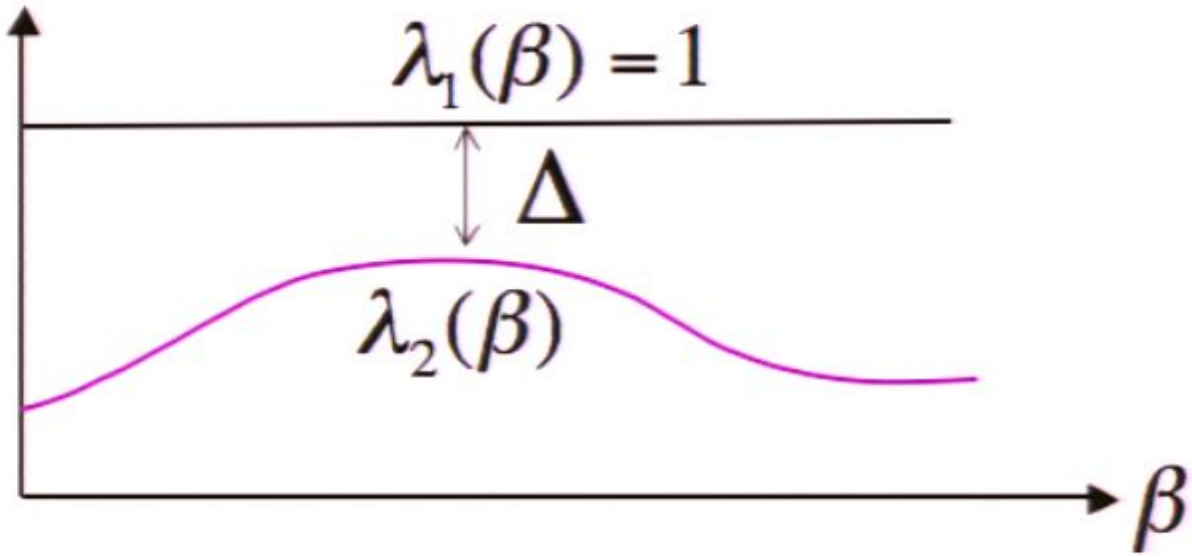
State space dimension

$$\Pr([\sigma] \notin A_0) \le \varepsilon \Rightarrow P = \frac{1}{\gamma} \frac{\log(d/\varepsilon^2)}{\Delta}$$

Spectral gap of E

Simulated Annealing and Markov Chain Monte Carlo (MCMC)

$$\Pr(\sigma \notin A_0) \le \varepsilon \Rightarrow P = \frac{1}{\gamma}\frac{\log(d/\varepsilon^2)}{\Delta}$$



For hard instances, Δ is exponentially small in N (log(state space dimension))

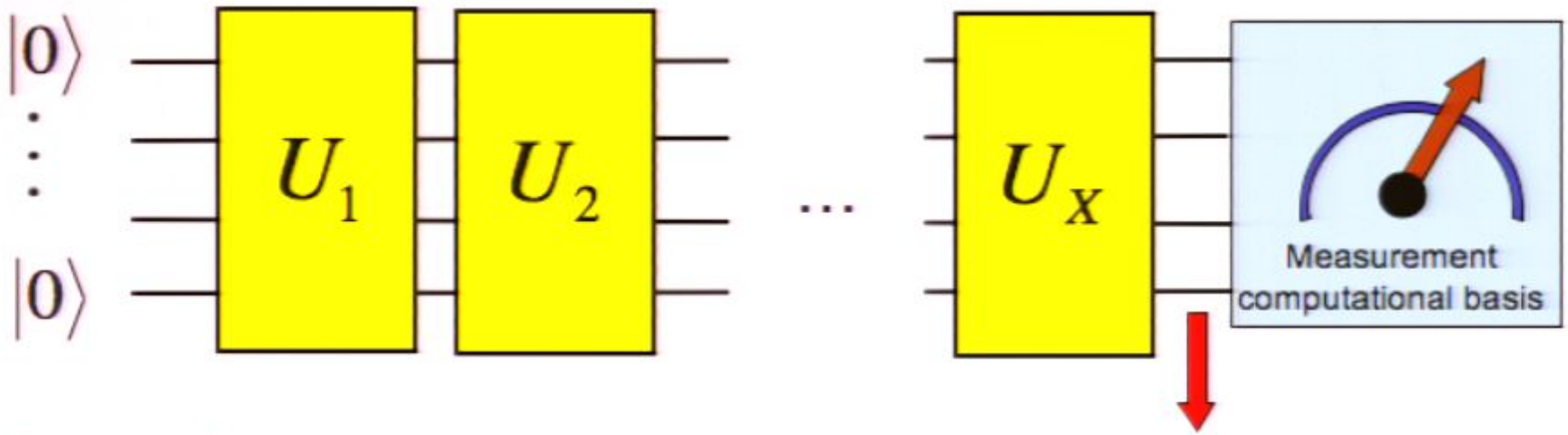
To solve combinatorial optimization problems, simulated annealing performs much more efficiently than exhaustive search!

The main implementation complexity is dominated by the inverse of the spectral gap of the stochastic matrix

Can quantum computers improve the dependence of P in Δ ?

Yes! They can provide a quadratic speedup

Quantum Algorithms for Computation of Properties of Classical Systems in Equilibrium



$$|\sigma\rangle = |\sigma_1 .. \sigma_N\rangle;$$

$$\sigma_j = 0,1$$

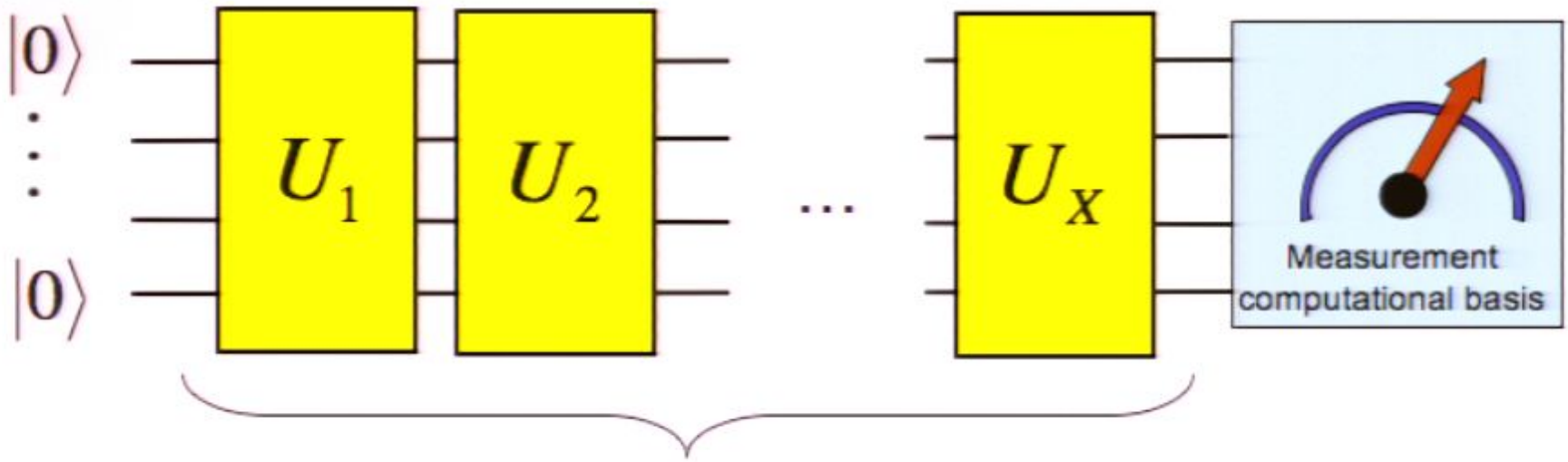
mplicit mapping) $[\sigma] \rightarrow |\sigma\rangle$

$$|\psi(\beta)\rangle = \sum_{\sigma} \sqrt{\pi_\sigma(\beta)}\,|\sigma\rangle$$

Projective measurements in the computational basis allow us to sample with the Boltzmann distribution:

$$\Pr(\sigma) = \pi_\sigma(\beta) = \frac{e^{-\beta E[\sigma]}}{Z}$$

Quantum Algorithms for Computation of Properties in Equilibrium



X:= Implementation complexity

$$\beta_f >> 1 \Rightarrow |\psi(\beta_f)\rangle \approx \frac{1}{\sqrt{\#A_0}} \sum_{\sigma \in A_0} |\sigma\rangle$$ equal superposition of ground states (solutions)

A possible strategy we will consider next is to prepare the desired state by "adiabatic evolution"

Quantum Algorithms for Computation of Properties in Equilibrium

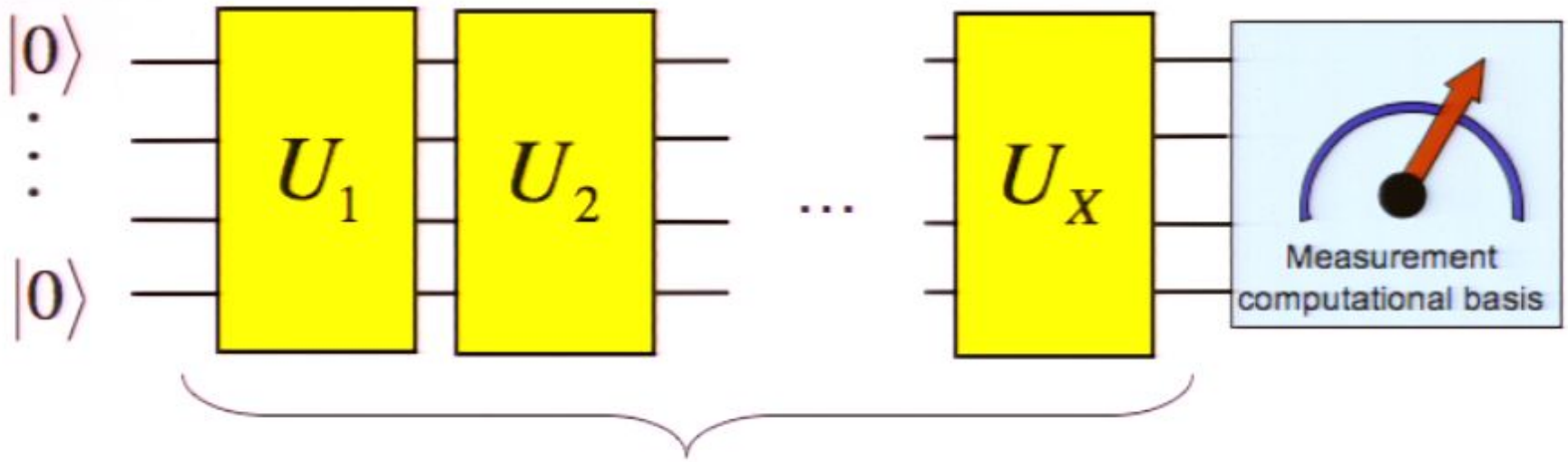
The quantum state $|\psi(\beta)\rangle = \sum_{\sigma} \sqrt{\pi_{\sigma}(\beta)}|\sigma\rangle$ can be shown to be the ground state of

$$e^{\beta\bar{E}/2} S(\beta) e^{-\beta\bar{E}/2} \equiv H(\beta) =$$

$$= \begin{bmatrix} p_{00\ldots0,00\ldots01} & \sqrt{p_{00\ldots0,00\ldots1} p_{00\ldots1,00\ldots0}} & \cdots & \sqrt{p_{00\ldots0,11\ldots1} p_{11\ldots1,00\ldots0}} \\ \sqrt{p_{00\ldots0,00\ldots1} p_{00\ldots1,00\ldots0}} & & & \vdots \\ \vdots & & \ddots & \vdots \\ \sqrt{p_{00\ldots0,11\ldots1} p_{11\ldots1,00\ldots0}} & \cdots & \cdots & p_{11\ldots1,11\ldots1} \end{bmatrix}$$

$$\bar{E} = \begin{bmatrix} E[1] & 0\ldots & 0 \\ \vdots & & \vdots \\ 0 & 0\ldots & E[d] \end{bmatrix}$$

Quantum Algorithms for Computation of Properties in Equilibrium

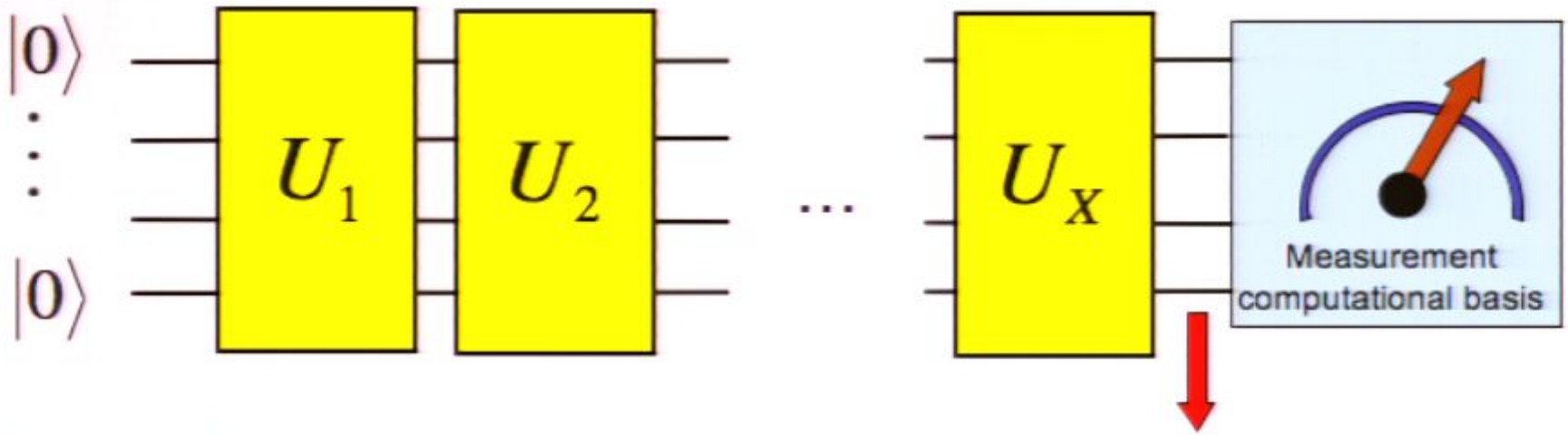


X:= Implementation complexity

$$\beta_f >> 1 \Rightarrow |\psi(\beta_f)\rangle \approx \frac{1}{\sqrt{\#A_0}} \sum_{\sigma \in A_0} |\sigma\rangle$$ equal superposition of ground states (solutions)

A possible strategy we will consider next is to prepare the desired state by "adiabatic evolution"

Quantum Algorithms for Computation of Properties of Classical Systems in Equilibrium



$$|\sigma\rangle = |\sigma_1 .. \sigma_N\rangle;$$

$$\sigma_j = 0,1$$

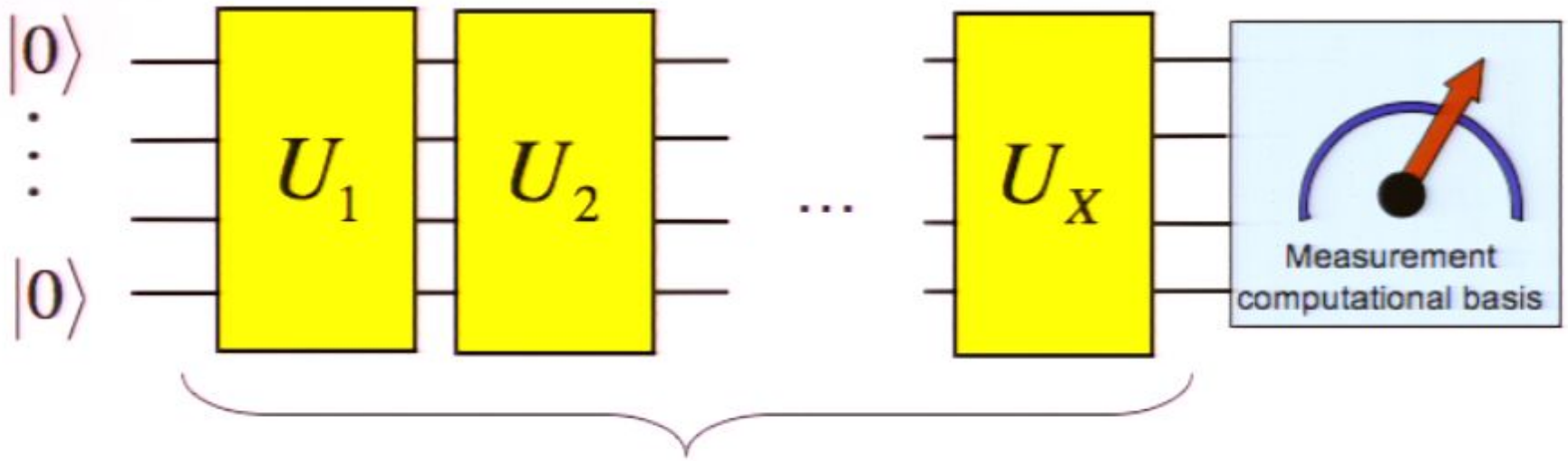
mplicit mapping) $[\sigma] \rightarrow |\sigma\rangle$

$$|\psi(\beta)\rangle = \sum_{\sigma} \sqrt{\pi_\sigma(\beta)}\,|\sigma\rangle$$

Projective measurements in the computational basis allow us to sample with the Boltzmann distribution:

$$\Pr(\sigma) = \pi_\sigma(\beta) = \frac{e^{-\beta E[\sigma]}}{Z}$$

Quantum Algorithms for Computation of Properties in Equilibrium



X:= Implementation complexity

$$\beta_f >> 1 \Rightarrow |\psi(\beta_f)\rangle \approx \frac{1}{\sqrt{\#A_0}} \sum_{\sigma \in A_0} |\sigma\rangle$$ equal superposition of ground states (solutions)

A possible strategy we will consider next is to prepare the desired state by "adiabatic evolution"

Quantum Algorithms for Computation of Properties in Equilibrium

The quantum state $|\psi(\beta)\rangle = \sum_{\sigma} \sqrt{\pi_{\sigma}(\beta)}|\sigma\rangle$ can be shown to be the ground state of

$$e^{\beta\bar{E}/2} S(\beta) e^{-\beta\bar{E}/2} \equiv H(\beta) =$$

$$= \begin{bmatrix} p_{00\ldots0,00\ldots01} & \sqrt{p_{00\ldots0,00\ldots1} p_{00\ldots1,00\ldots0}} & \cdots & \sqrt{p_{00\ldots0,11\ldots1} p_{11\ldots1,00\ldots0}} \\ \sqrt{p_{00\ldots0,00\ldots1} p_{00\ldots1,00\ldots0}} & & & \vdots \\ \vdots & & \ddots & \vdots \\ \sqrt{p_{00\ldots0,11\ldots1} p_{11\ldots1,00\ldots0}} & \cdots & \cdots & p_{11\ldots1,11\ldots1} \end{bmatrix}$$

$$\bar{E} = \begin{bmatrix} E[1] & 0\ldots & 0 \\ \vdots & & \vdots \\ 0 & 0\ldots & E[d] \end{bmatrix}$$

Quantum Algorithms for Computation of Properties in Equilibrium

The quantum state $|\psi(\beta)\rangle = \sum_{\sigma} \sqrt{\pi_\sigma(\beta)}|\sigma\rangle$ can be shown to be the ground state of $H(\beta)$

Same spectra and gap of S:

$$H(\beta)|\phi_j\rangle = \lambda_j|\phi_j\rangle; \quad \lambda_1 = 1 > \lambda_2 > \ldots > \lambda_d > 0$$

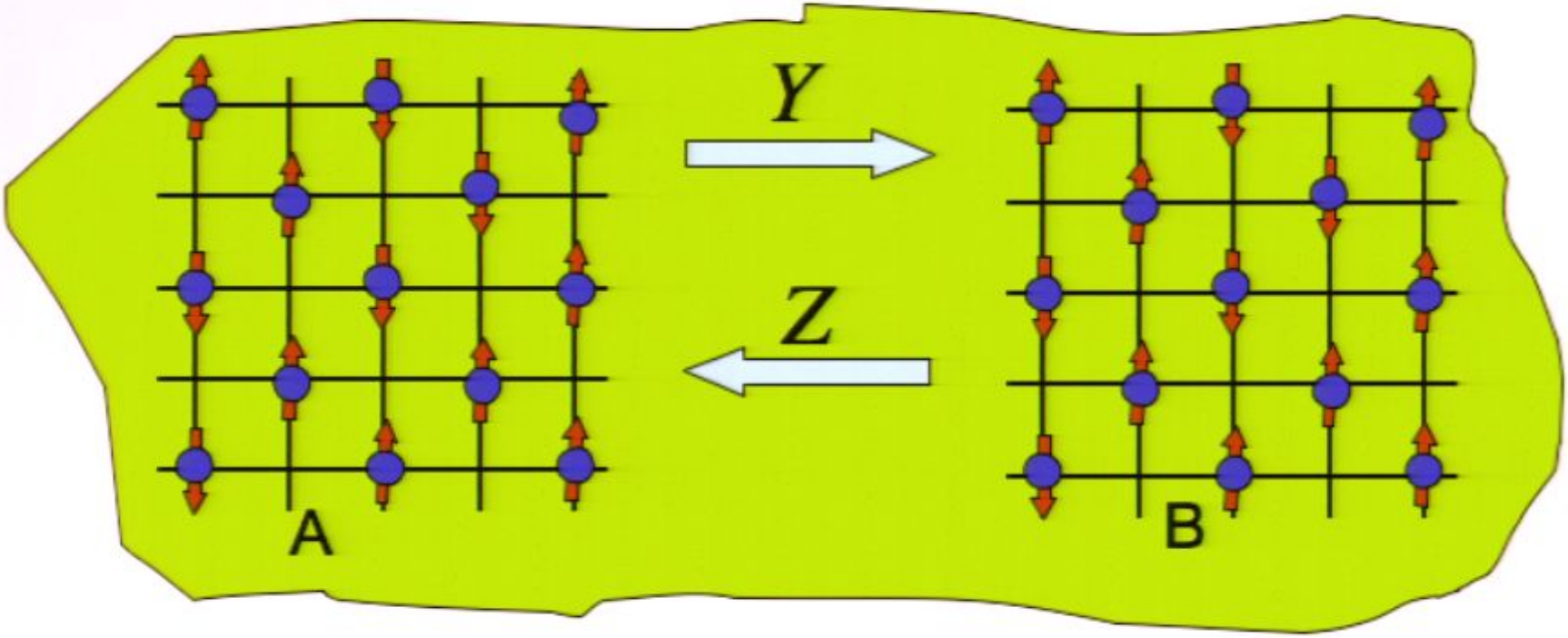
$$|\phi_1\rangle = \sum_{\sigma} \sqrt{\pi_\sigma(\beta)}|\sigma\rangle$$

An adiabatic quantum algorithm built to remain in the GS of $H(\beta(t))$ **will not** provide a quantum speed up

$$\mathrm{T} = O(1/\Delta^2)$$

Next : Build $\tilde{H}(\beta)$ with larger spectral gap!

Bipartite quantum walks: Quantization of Markov Processes



sometries

$$Y|\sigma\rangle = |\sigma\rangle_A \sum_{\sigma'} \sqrt{p_{\sigma\sigma'}}\,|\sigma'\rangle_B = U_Y|\sigma\rangle_A|0\rangle_B$$

$$Z|\sigma\rangle = \sum_{\sigma'} \sqrt{p_{\sigma\sigma'}}\,|\sigma'\rangle_A|\sigma\rangle_B = U_Z|0\rangle_A|\sigma\rangle_B$$

Each use of these is equivalent to one step in MCMC

M. Szegedy (2004) - F. Magniez et al. (2007) - R. Somma et al. (2007)

Bipartite quantum walks: Quantization of Markov Processes

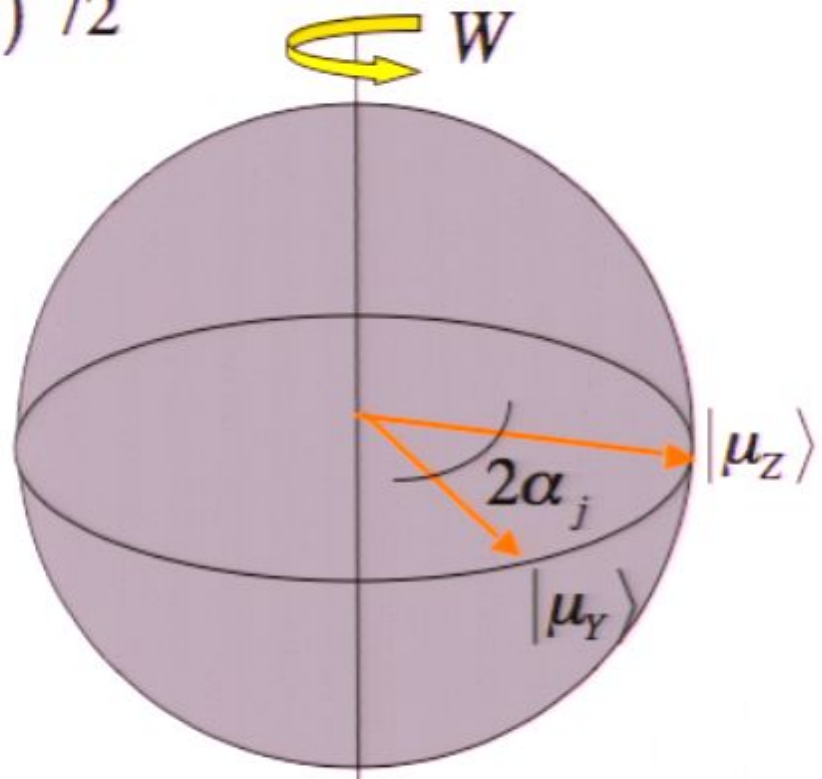
$$H(\beta) = Y^{+}Z$$

$$\lambda_j = \cos\alpha_j = \langle \phi_j | Y^{+}Z | \phi_j \rangle \equiv {}_A\langle \phi_j |_B \langle 0 | U_Y^{+} U_Z | 0 \rangle_A | \phi_j \rangle_B$$

$\langle \mu_Y |$ $\quad$ $| \mu_Z \rangle$

$$\lambda_1 = 1 \Rightarrow \alpha_1 = 0$$

$$\delta = 1 - \lambda_1 \approx (\alpha_2)^2 / 2$$



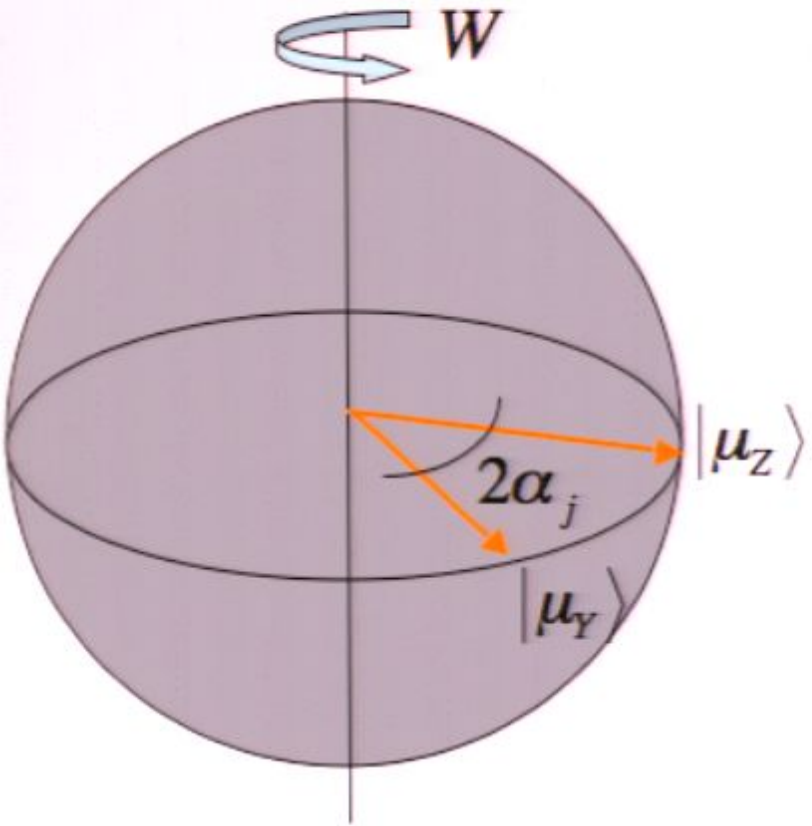
Quantum walk:

$$W = R_{\mu_Y} R_{\mu_Z}$$

$$R_{\mu_Y} = 2|\mu_Y\rangle\langle\mu_Y| - 1$$

$$R_{\mu_Z} = 2|\mu_Z\rangle\langle\mu_Z| - 1$$

Bipartite quantum walks: Quantization of Markov Processes



Quantum walk:

$$W = R_{\mu_Y} R_{\mu_Z} \equiv U_Y^+ R_{|0\rangle_A} U_Y U_Z^+ R_{|0\rangle_B} U_Z$$

The implementation complexity of *W* is equivalent to 4 steps of MCMC

$$R_{|0\rangle_A} = 2|0\rangle_A\langle 0| - 1$$

$$R_{|0\rangle_B} = 2|0\rangle_B\langle 0| - 1$$

$$W|\xi_j\rangle = e^{\pm i2\alpha_j}|\xi_j\rangle;$$

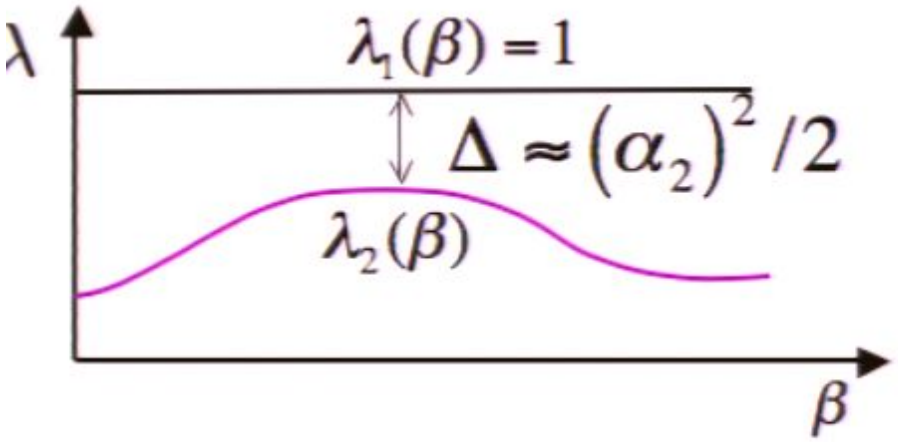
$$|\xi_1\rangle = |\phi_1\rangle_A|0\rangle_B = \sum_\sigma \sqrt{\pi_\sigma}|\sigma\rangle_A|0\rangle_B$$

Bipartite quantum walks: Quantization of Markov Processes

$$H(\beta)\left|\phi_j\right\rangle = \lambda_j \left|\phi_j\right\rangle;$$

$$\left|\phi_1\right\rangle = \sum_\sigma \sqrt{\pi_\sigma(\beta)}\left|\sigma\right\rangle;$$

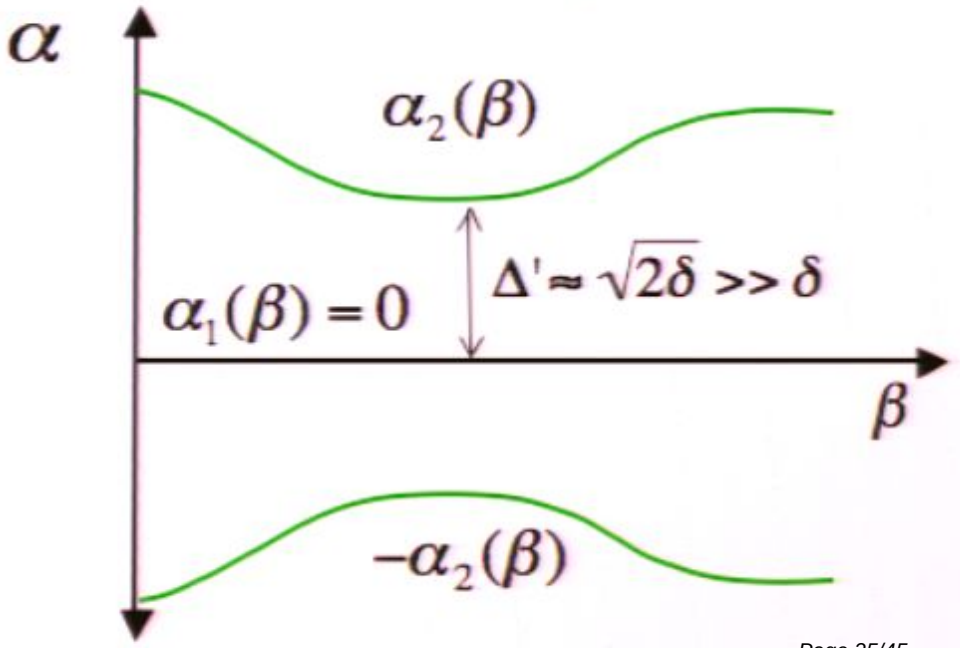
$$\lambda_1 = 1 > \lambda_2 > \ldots > \lambda_d > 0$$



$$W = R_{\mu_Y} R_{\mu_Z}$$

$$W\left|\zeta_j\right\rangle = e^{\pm i 2\alpha_j}\left|\zeta_j\right\rangle;$$

$$\left|\zeta_1\right\rangle = \left|\phi_1\right\rangle_A \left|0\right\rangle_B = \sum_\sigma \sqrt{\pi_\sigma}\left|\sigma\right\rangle_A \left|0\right\rangle_B$$



Quantum Simulated Annealing

The idea is to remain in the 0-eigenphase state $|\psi(\beta)\rangle_A |0\rangle_B$ of W at all β

To follow the desired state we make use of the Quantum Zeno effect

$$\left|\langle\psi(\beta)|\psi(\beta+\delta\beta)\rangle\right| \geq 1 - c(\delta\beta)^2$$

Algorithm:

Perform measurements in the (instantaneous) eigenbasis of

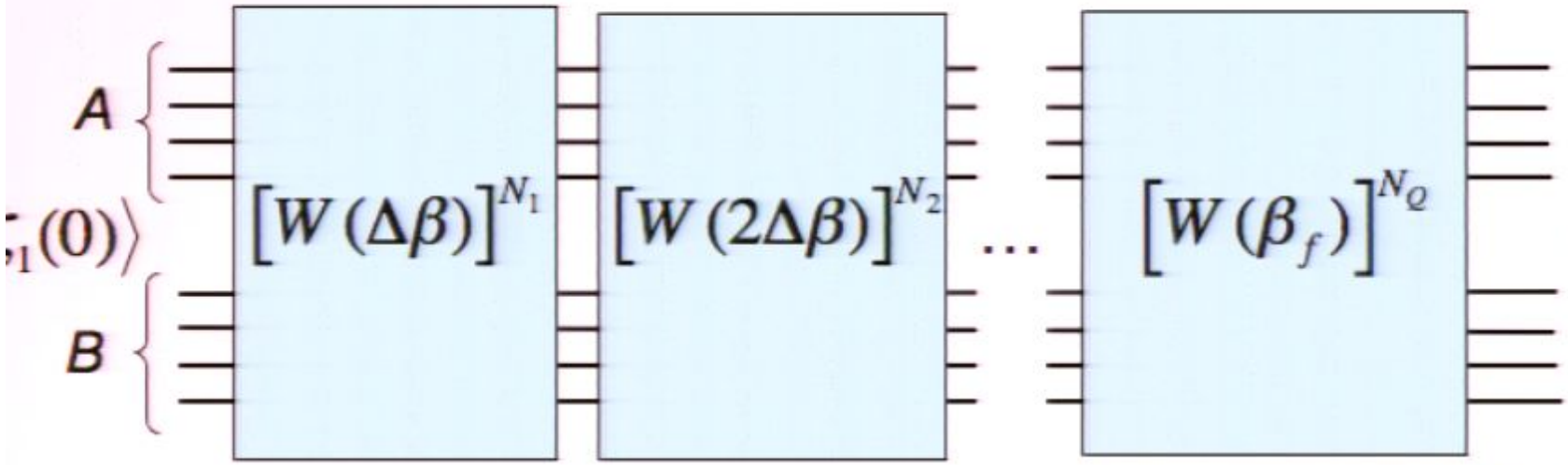
$$W(\beta=0), W(\beta=\delta\beta), \ldots, W(\beta_f = Q\delta\beta)$$

(Q: complexity of QSA)

Measurements in the instantaneous eigenbasis implies that after Q steps, the probability of remaining in the 0-eigenphase state is (ideally):

$$\Pr\left(|\psi(\beta_f)\rangle\right) \geq (1-\delta\beta^2)^Q \geq 1 - \underbrace{\beta_f \delta\beta}_{\varepsilon} \quad \Rightarrow \quad \delta\beta \approx \frac{\varepsilon}{\beta_f} << 1$$

Quantum Simulated Annealing: Implementation via randomization



$$= \sum_{\sigma} \sqrt{\pi_\sigma(\beta_f)} |\sigma\rangle_A |0\rangle$$

$$\frac{1}{\xi\alpha_2} \geq N_i \geq 0$$

Chosen with the uniform distribution
(This choice allows us to "simulate" the Quantum Zeno effect)

nverse of the spectral gap of W

Dephasing of the 0-eigenphase state with the others

Quantum Simulated Annealing

The idea is to remain in the 0-eigenphase state $|\psi(\beta)\rangle_A |0\rangle_B$ of W at all β

To follow the desired state we make use of the Quantum Zeno effect

$$\left|\langle\psi(\beta)|\psi(\beta+\delta\beta)\rangle\right| \geq 1 - c(\delta\beta)^2$$

Algorithm:

Perform measurements in the (instantaneous) eigenbasis of

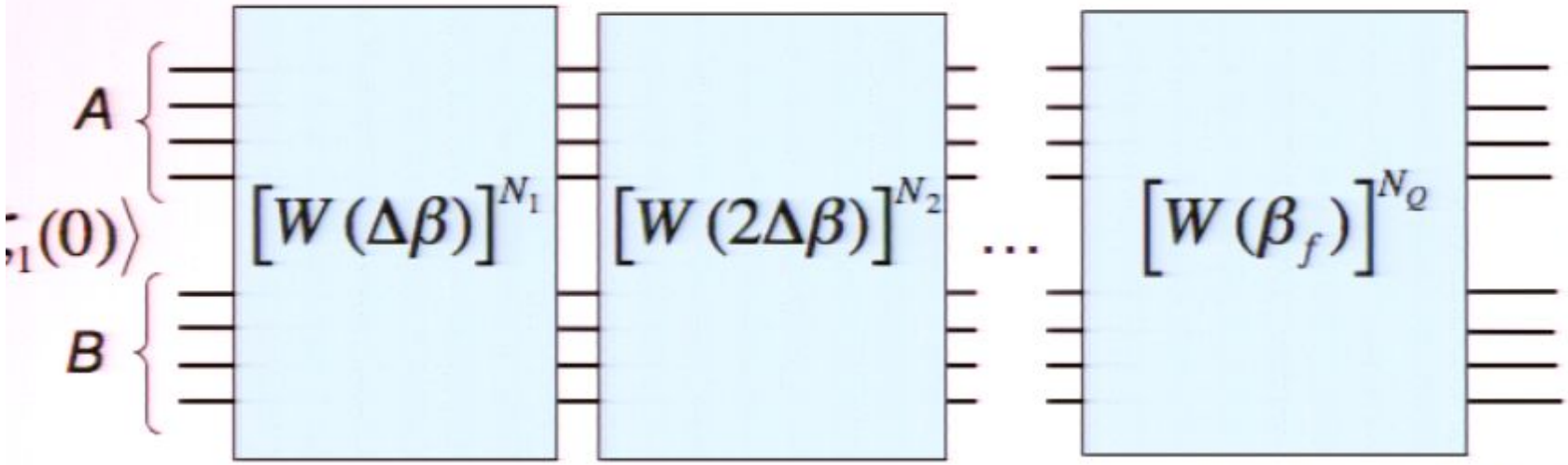
$$W(\beta=0), W(\beta=\delta\beta), \ldots, W(\beta_f = Q\delta\beta)$$

(Q: complexity of QSA)

Measurements in the instantaneous eigenbasis implies that after Q steps, the probability of remaining in the 0-eigenphase state is (ideally):

$$\Pr\left(|\psi(\beta_f)\rangle\right) \geq (1-\delta\beta^2)^Q \geq 1 - \underbrace{\beta_f \delta\beta}_{\varepsilon} \quad \Rightarrow \quad \delta\beta \approx \frac{\varepsilon}{\beta_f} << 1$$

Quantum Simulated Annealing: Implementation via randomization



$$= \sum_{\sigma} \sqrt{\pi_\sigma(\beta_f)} |\sigma\rangle_A |0\rangle$$

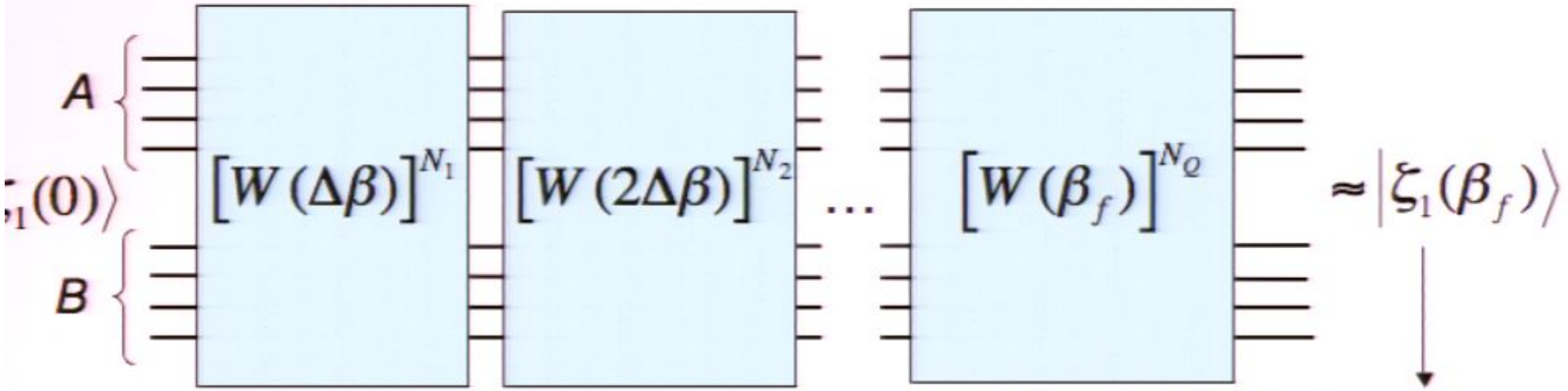
$$\frac{1}{\xi\alpha_2} \geq N_i \geq 0$$

Chosen with the uniform distribution
(This choice allows us to "simulate" the Quantum Zeno effect)

nverse of the spectral gap of W

Dephasing of the 0-eigenphase state with the others

Quantum Simulated Annealing



Convergence rates and resources:

of operations: $N_1 + N_2 + \ldots \approx \dfrac{Q}{\xi \underbrace{\alpha_2}_{\sqrt{\Delta}}}$

$$\Pr(\sigma \notin S_0) \le \varepsilon \Rightarrow Q = \frac{1}{\gamma^3} \frac{\log^3(d/\varepsilon^2)}{\varepsilon^2 \sqrt{\Delta}}$$

Quantum Simulated Annealing: Concluding Remarks

Classical simulated annealing:

$$\Pr(\sigma \notin S_0) \le \varepsilon \Rightarrow P = \frac{1}{\gamma}\frac{\log(d/\varepsilon^2)}{\Delta}$$

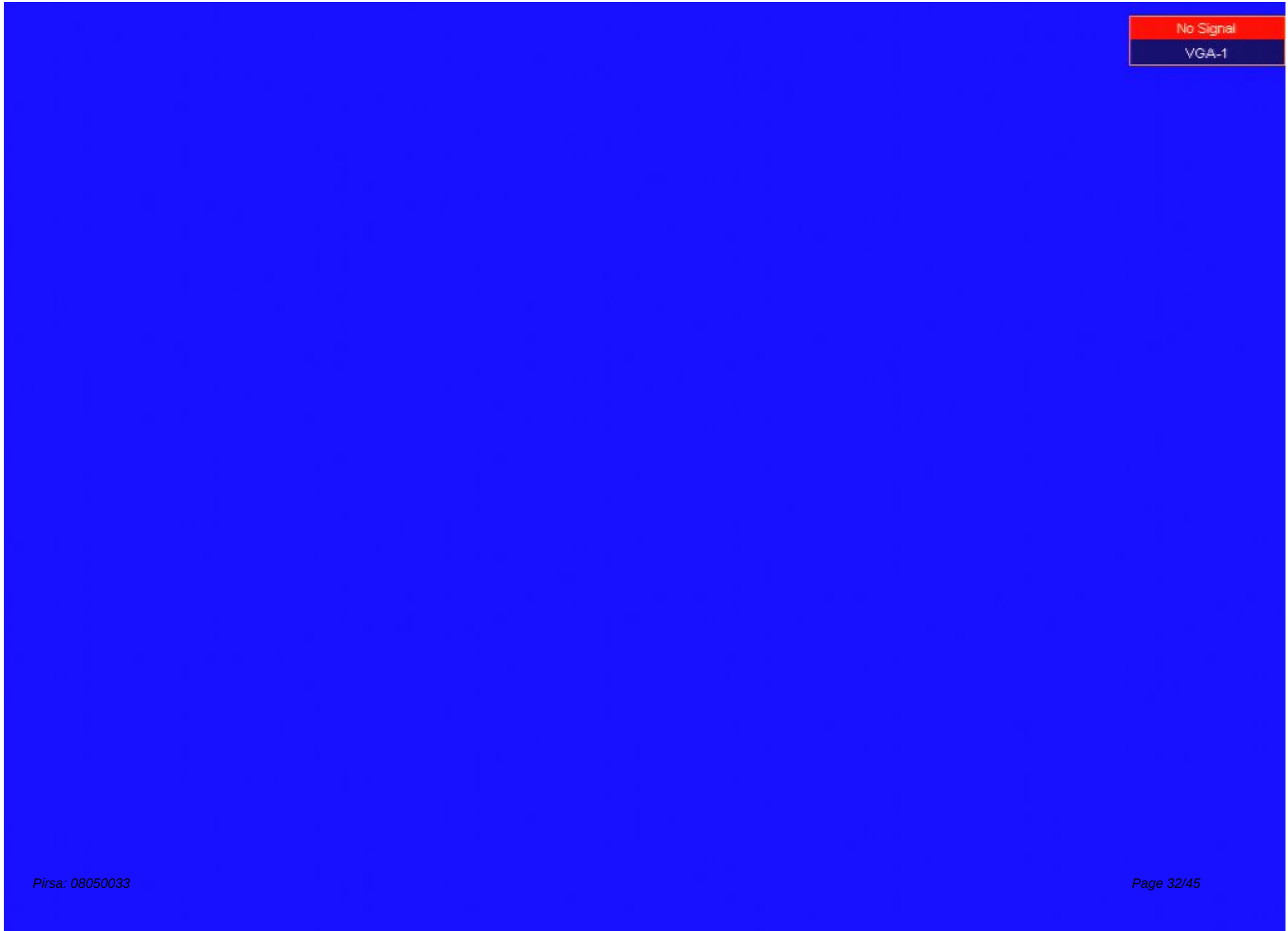
Classical simulated annealing:

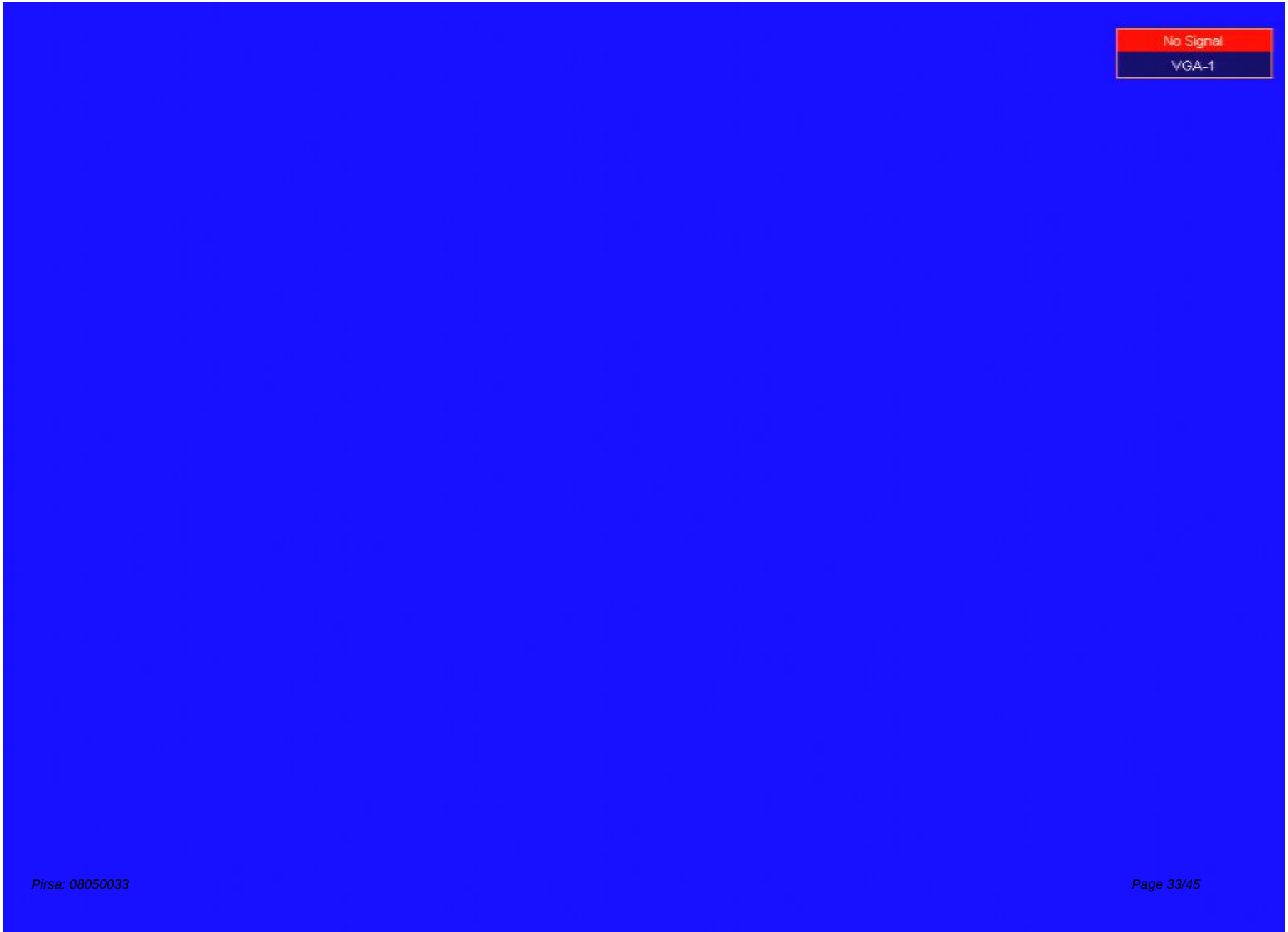
$$\Pr(\sigma \notin S_0) \le \varepsilon \Rightarrow Q = \frac{1}{\gamma^3}\frac{\log^3(d/\varepsilon^2)}{\varepsilon^2\sqrt{\Delta}}$$

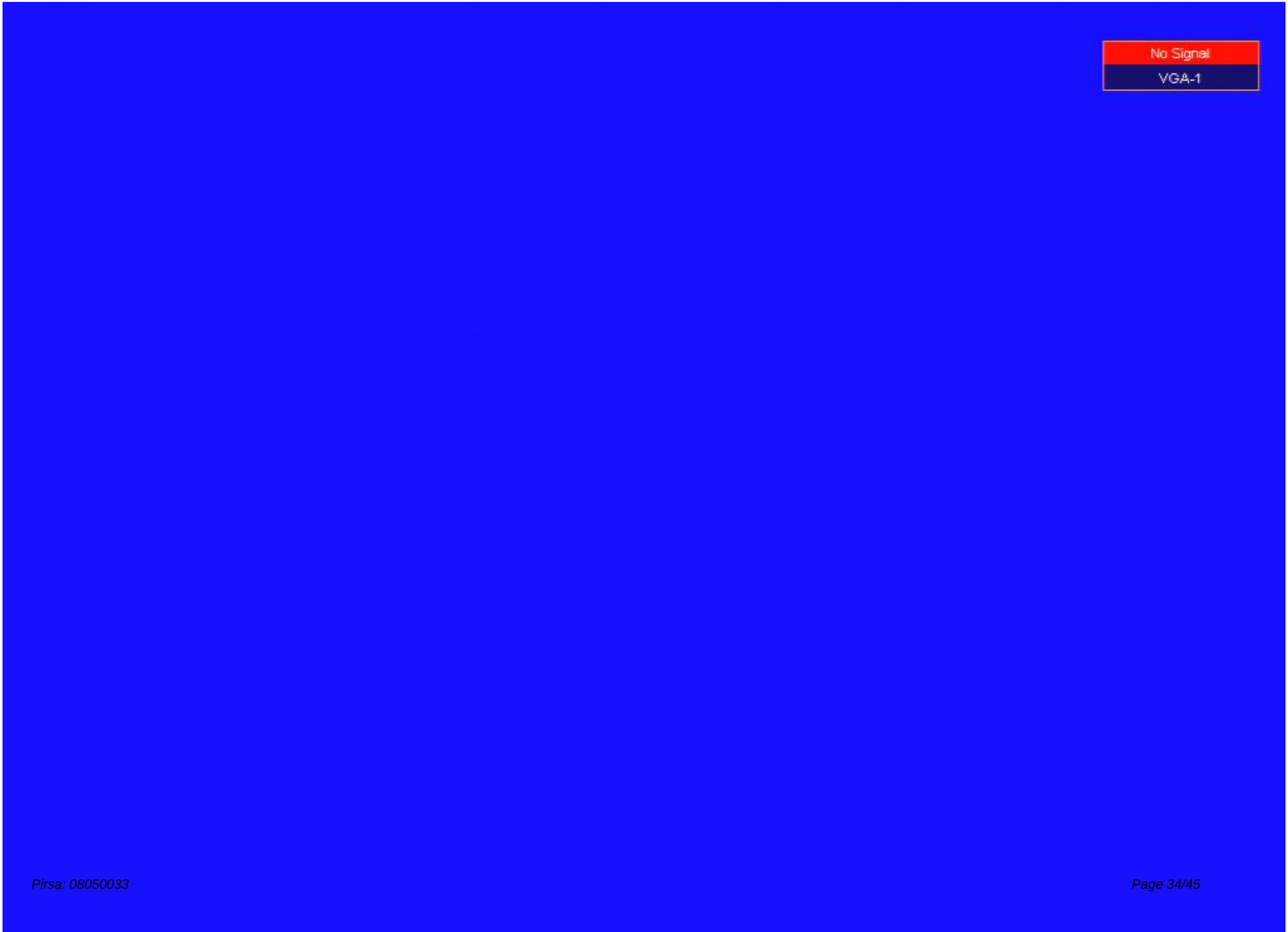
-For NP-complete, one can argue that γ<1/poly(N)

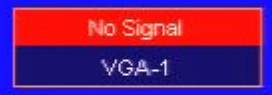
-The dependence in ε can be made logarithmic by repetition of the QSA

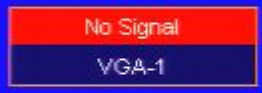
⇒ The QSA greatly outperforms SA for hard instances

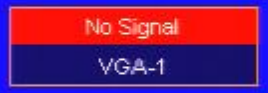

No Signal
VGA-1

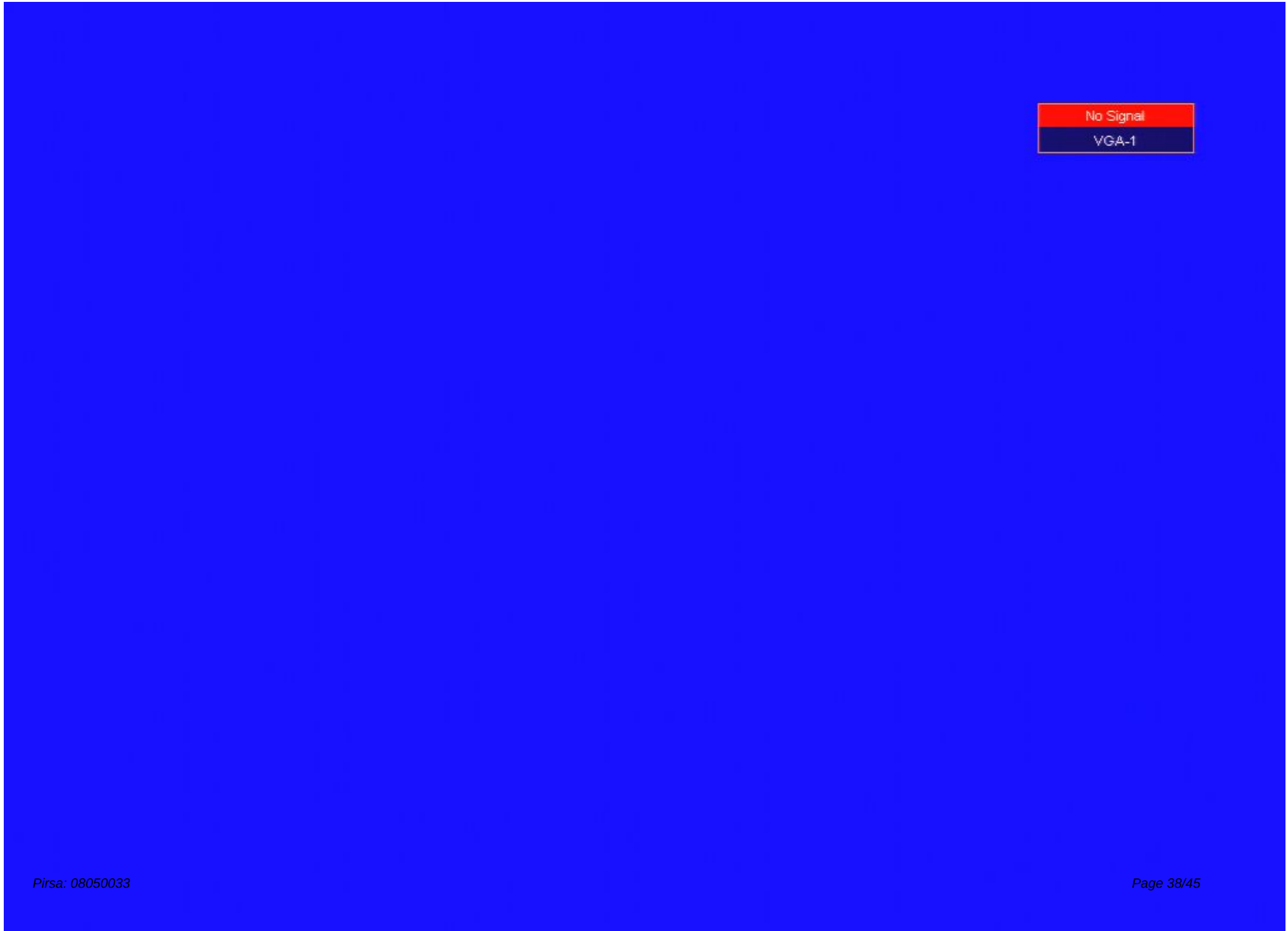

No Signal
VGA-1

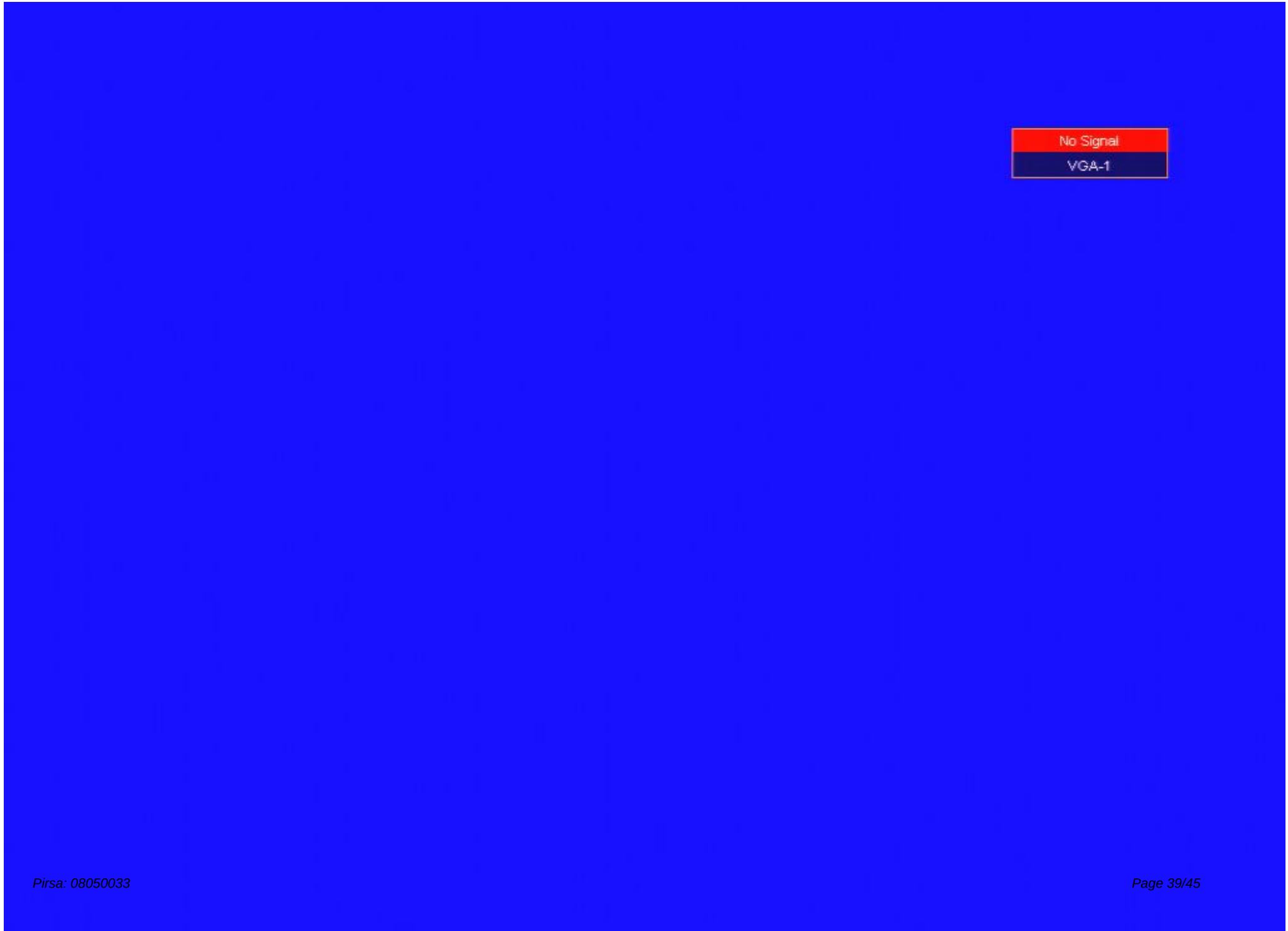

No Signal
VGA-1

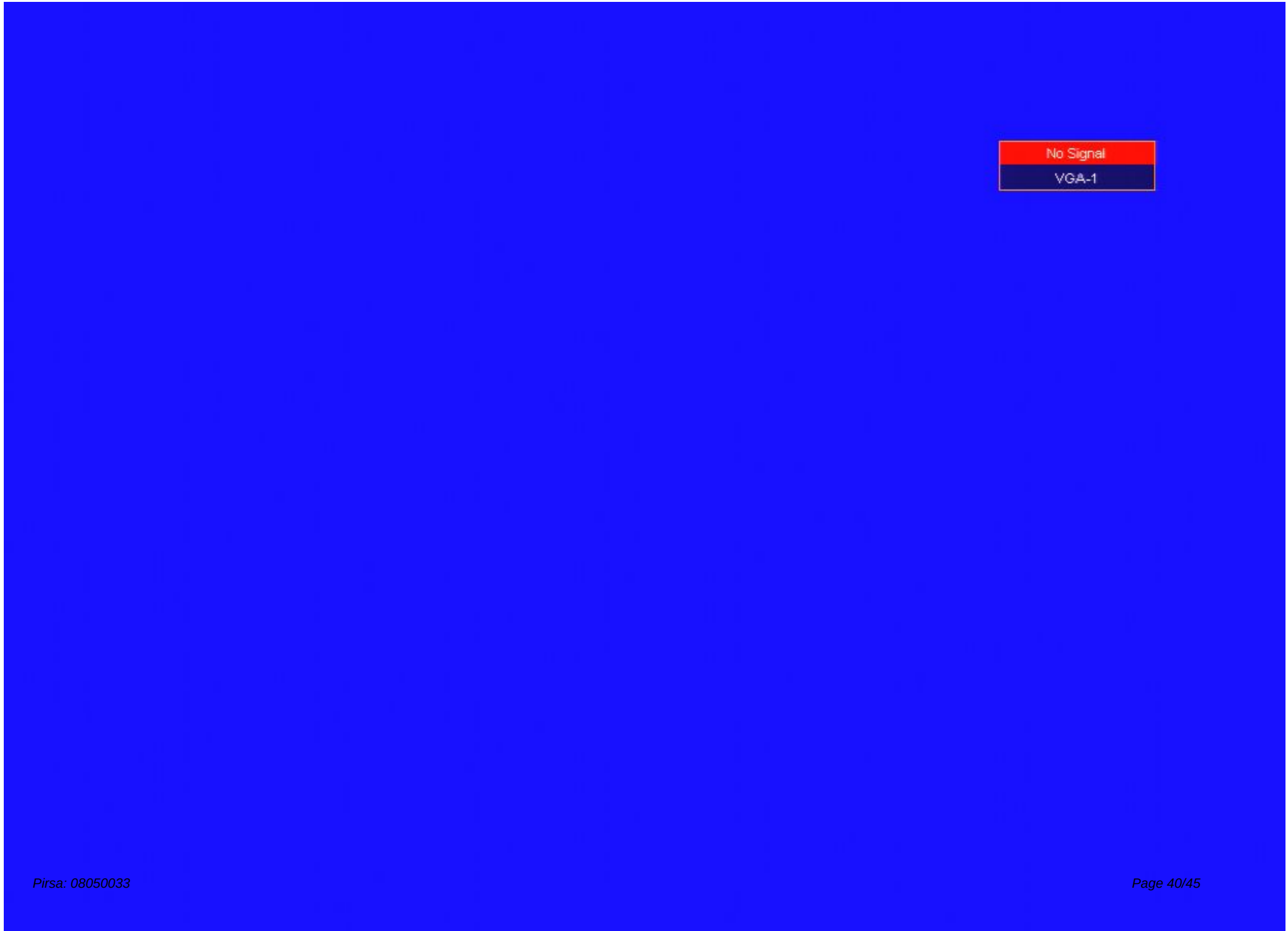

No Signal
VGA-1

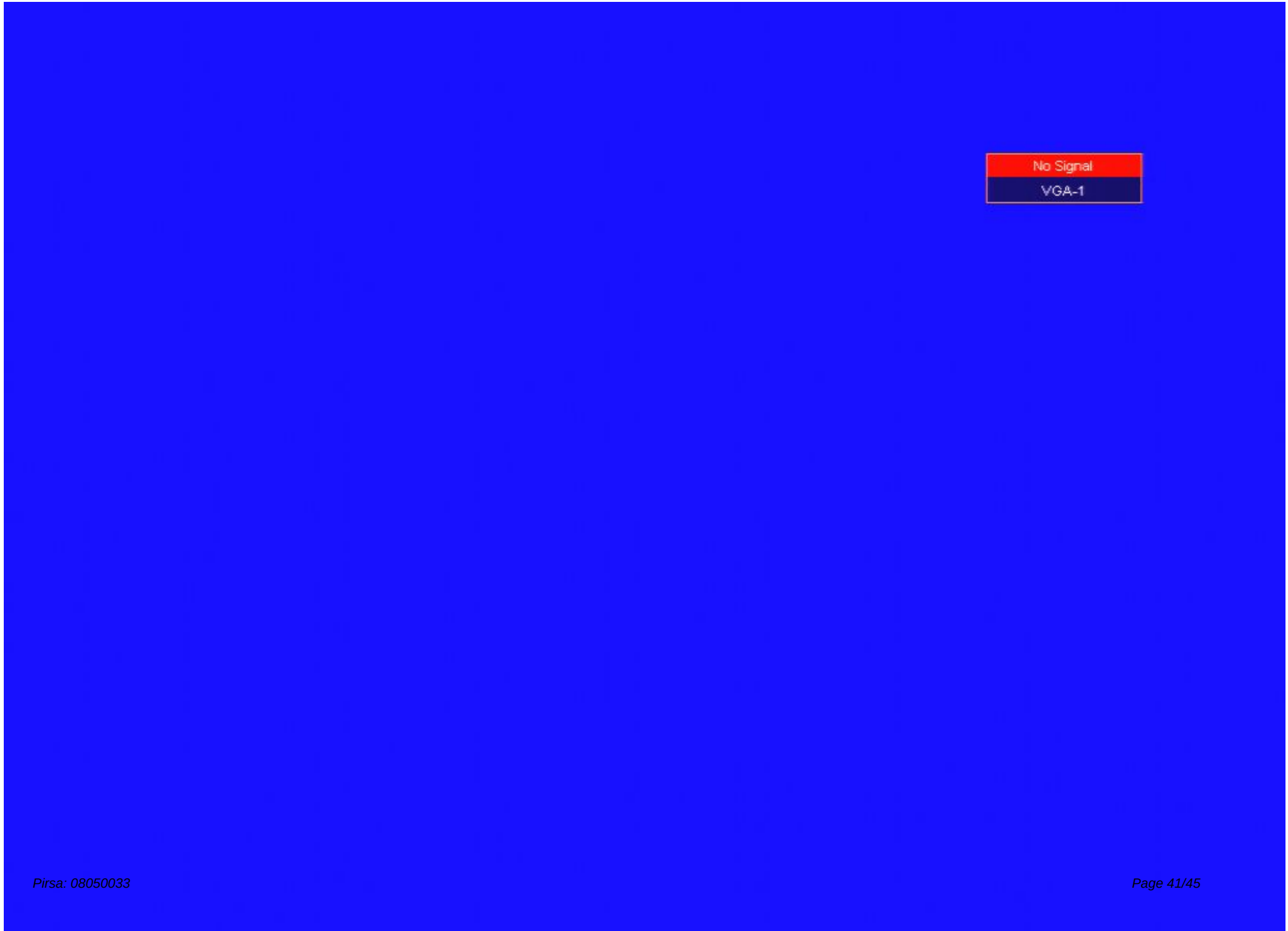

No Signal
VGA-1

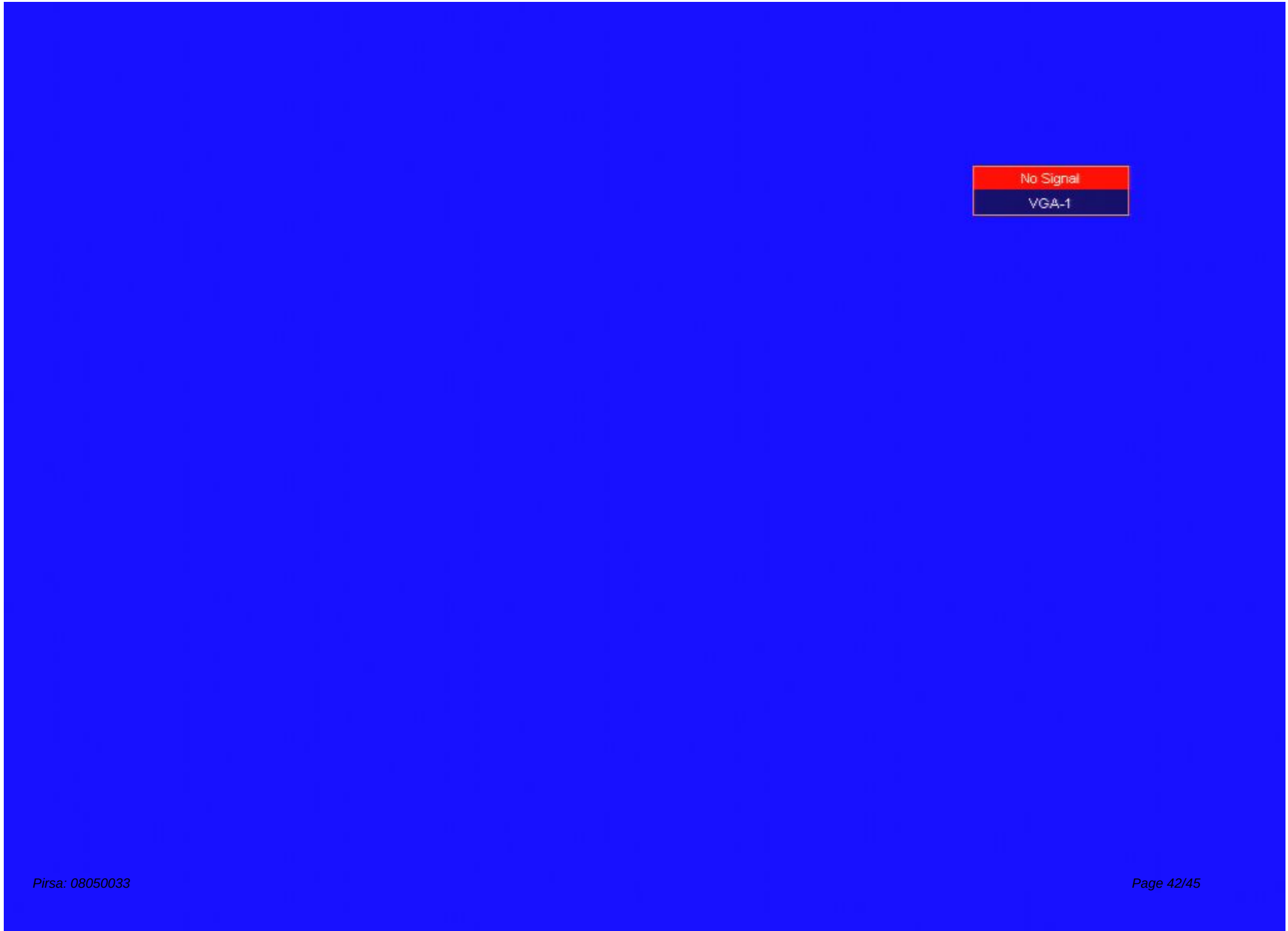

No Signal
VGA-1

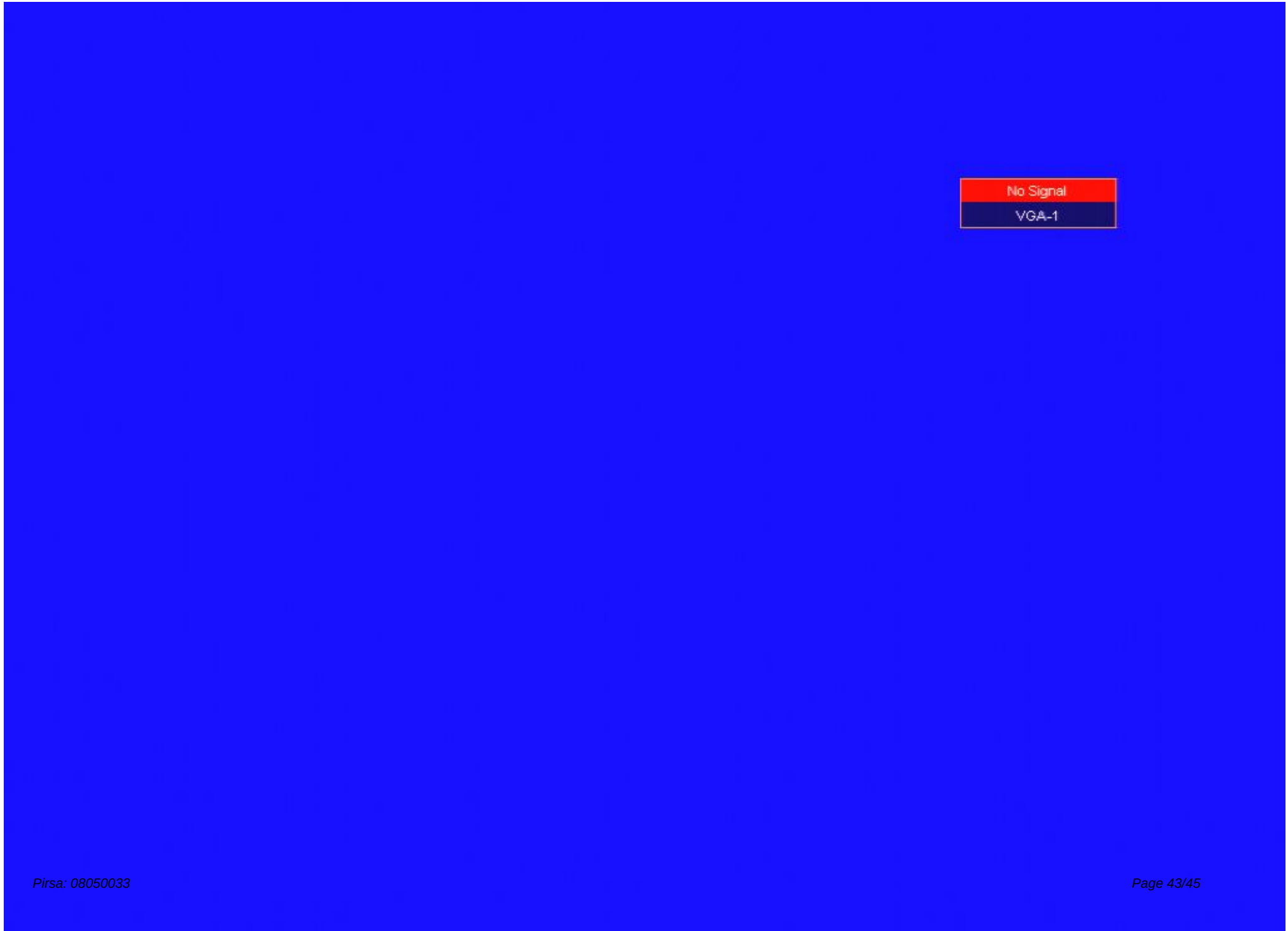

No Signal
VGA-1


No Signal
VGA-1

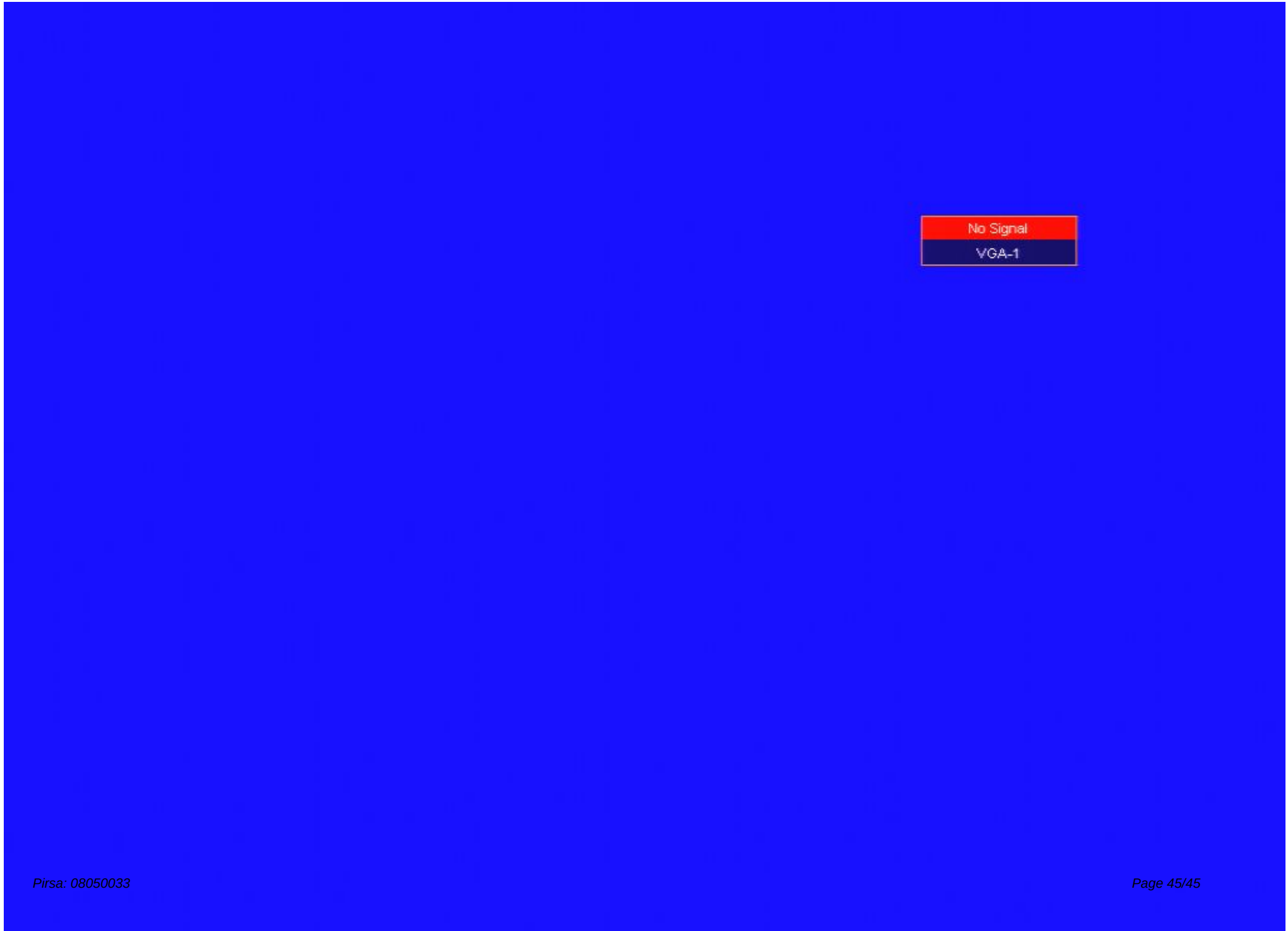

No Signal
VGA-1


No Signal
VGA-1


No Signal
VGA-1


No Signal
VGA-1


No Signal
VGA-1


No Signal
VGA-1