Quantum versus Thermal annealing (or D-wave versus Janus): seeking a fair comparison

Víctor Martín-Mayor

Dep. Física Teórica I, Universidad Complutense de Madrid
Janus Collaboration

In collaboration with Itay Hen (Information Sciences Institute, USC).

HPC 2014, Cetraro, July 8 2014.

Foreword

Monte Carlo is an extremely bad *method; it should be used only when all alternative methods are worse.*
(Alan Sokal, 1989).

Foreword

> *Monte Carlo is an extremely* bad *method; it should be used only when all alternative methods are worse.*
> (Alan Sokal, 1989).

Such desperate problems are common in Theoretical Physics.

Foreword

> *Monte Carlo is an extremely* bad *method; it should be used only when all alternative methods are worse.*
> (Alan Sokal, 1989).

Such desperate problems are common in Theoretical Physics.

Our applications are extremely computer *intensive* but *simple*: the Janus collaboration has dared to produce dedicated hardware.

Foreword

> *Monte Carlo is an extremely* bad *method; it should be used only when all alternative methods are worse.*
> (Alan Sokal, 1989).

Such desperate problems are common in Theoretical Physics.

Our applications are extremely computer *intensive* but *simple*: the Janus collaboration has dared to produce dedicated hardware.

Janus is a great success, but classical Monte Carlo is hitting an algorithmic wall.

Foreword

> *Monte Carlo is an extremely* bad *method; it should be used only when all alternative methods are worse.*
> (Alan Sokal, 1989).

Such desperate problems are common in Theoretical Physics.

Our applications are extremely computer *intensive* but *simple*: the Janus collaboration has dared to produce dedicated hardware.

Janus is a great success, but classical Monte Carlo is hitting an algorithmic wall.

Is quantum computing our breakthrough?

Plan of the presentation

1. A quick overview: spin-glasses.
2. Desperate problem, desperate solutions: the Janus computer.
3. The temperature chaos algorithmic wall.
4. D-wave, the chimera lattice and temperature chaos.

Spin-glasses (I)

Spin-glasses are disordered magnetic alloys.

They can be mapped (at zero temperature) to a Computer Science optimization problem:

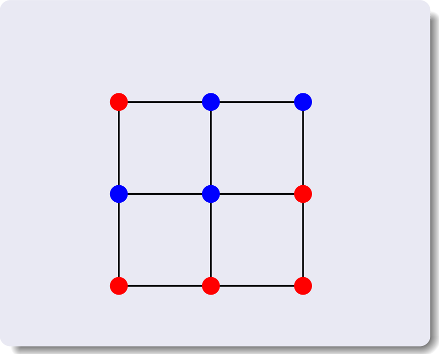
QUBO (Quadratic Unconstrained Binary Optimization)

$$E(\{b_i\}) = -\sum_{ij} Q_{i,j} b_i b_j - \sum_i h_i b_i \,.$$

Looks like minimizing a quadratic form, but this is not a Calculus exercise: $b_i = 0, 1$.

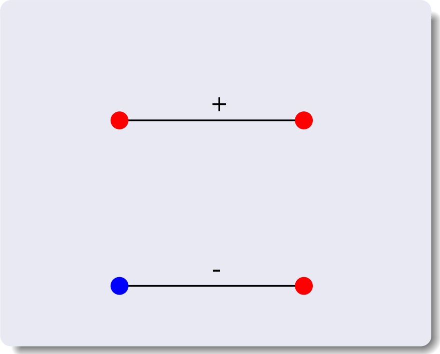
Spin glasses (II)

- We have a graph (V, E)
  Vertices: binary variables
  Edges: interactions.



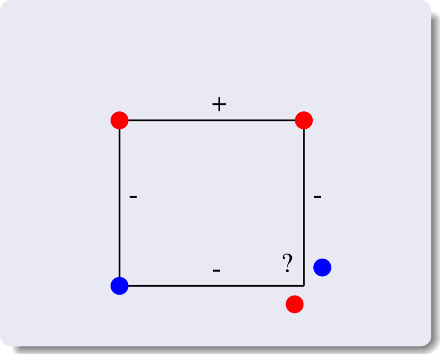
Spin glasses (II)

- We have a graph (V, E) Vertices: binary variables Edges: interactions.
- Interactions: ferromagnetic (+) or antiferromagnetic (-), 50% probability ($\longrightarrow$ instances).



Spin glasses (II)

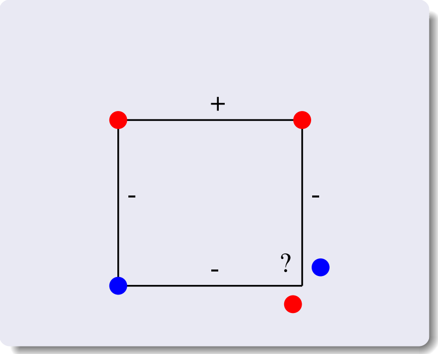
- We have a graph (V, E)
  Vertices: binary variables
  Edges: interactions.
- Interactions:
  ferromagnetic (+) or
  antiferromagnetic (-),
  50% probability ($\longrightarrow$ instances).
- Loops $\longrightarrow$ frustration.



Spin glasses (II)

- We have a graph (V, E)
  Vertices: binary variables
  Edges: interactions.
- Interactions:
  ferromagnetic (+) or
  antiferromagnetic (-),
  50% probability ($\longrightarrow$ instances).
- Loops $\longrightarrow$ frustration.

Minimum energy: NP-hard for non-planar graphs.



Spin glasses and computer science

Up to now, spin glasses perfectly useless materials but. . .

- An inspiration to understand NP-completeness (Zecchina, Mèzard, Parisi, etc.)
- A preferred bench-mark for quantum computing.
- A source of heuristic algorithms: Simulated Annealing (Kirkpatrick, Gelatt, Vecchi).

Spin glasses and computer science

Up to now, spin glasses perfectly useless materials but. . .

- An inspiration to understand NP-completeness (Zecchina, Mèzard, Parisi, etc.)
- A preferred bench-mark for quantum computing.
- A source of heuristic algorithms: Simulated Annealing (Kirkpatrick, Gelatt, Vecchi).

  Simulated Annealing is outdated for spin-glasses.
  Current method of choice: Parallel Tempering.

The Janus Collaboration

Team from 5 universities in Spain and Italy:

- Universidad Complutense de Madrid:
  M. Baity-Jesi, L.A. Fernandez, V. Martin-Mayor, A. Muñoz Sudupe
- Universidad de Extremadura:
  A. Gordillo-Guerrero, J.J. Ruiz-Lorenzo
- Università di Ferrara:
  M. Pivanti, S.F. Schifano, R. Tripiccione
- La Sapienza Università di Roma:
  A. Maiorano, E. Marinari, G. Parisi, F. Ricci-Tersenghi, D. Yllanes, B. Seoane
- Universidad de Zaragoza:
  R.A. Baños, A. Cruz, J.M. Gil-Narvión, M. Guidetti, D. Iñiguez, J. Monforte-Garcia, D. Navarro, S. Perez-Gaviro, A. Tarancon, P. Tellez.

Physicists and engineers dedicated to the design and exploitation of special-purpose computers, optimised for Monte Carlo simulations in condensed matter physics.

Desperate problems, desperate solutions: Janus (I)

Even with binary spins, simulation of spin-glasses is heavy in two respects:

1. Many ($\sim 10^3$) problem instances → embarrassingly parallel.

Desperate problems, desperate solutions: Janus (I)

Even with binary spins, simulation of spin-glasses is heavy in two respects:

1. Many ($\sim 10^3$) problem instances $\rightarrow$ embarrassingly parallel.
2. Single instance simulation **very long.**

Desperate problems, desperate solutions: Janus (I)

Even with binary spins, simulation of spin-glasses is heavy in two respects:

1. Many ($\sim 10^3$) problem instances → embarrassingly parallel.
2. Single instance simulation **very long.**
   For modest system sizes (i.e. $N = 32^3 = 32768$ spins):
   - Typical instance: 4.5 standard-CPU years (i.e. 1.4×10^{17} updates)

Desperate problems, desperate solutions: Janus (I)

Even with binary spins, simulation of spin-glasses is heavy in two respects:

1. Many ($\sim 10^3$) problem instances → embarrassingly parallel.
2. Single instance simulation **very long.**
   For modest system sizes (i.e. $N = 32^3 = 32768$ spins):
   - Typical instance: 4.5 standard-CPU years (i.e. 1.4×10^{17} updates)
   - Worst in 10^3 instances: 800 standard-CPU years (i.e. 2.7×10^{19} updates).

Desperate problems, desperate solutions: Janus (I)

Even with binary spins, simulation of spin-glasses is heavy in two respects:

1. Many ($\sim 10^3$) problem instances → embarrassingly parallel.
2. Single instance simulation **very long.**
   For modest system sizes (i.e. $N = 32^3 = 32768$ spins):
   - Typical instance: 4.5 standard-CPU years (i.e. 1.4×10^{17} updates)
   - Worst in 10^3 instances: 800 standard-CPU years (i.e. 2.7×10^{19} updates).

Fortunately, the spin update (the core algorithm) is very simple and (in principle) trivial to parallelize. But. . .

Desperate problems, desperate solutions: Janus (I)

Even with binary spins, simulation of spin-glasses is heavy in two respects:

1. Many ($\sim 10^3$) problem instances → embarrassingly parallel.
2. Single instance simulation **very long.**
   For modest system sizes (i.e. $N = 32^3 = 32768$ spins):
   - Typical instance: 4.5 standard-CPU years (i.e. 1.4×10^{17} updates)
   - Worst in 10^3 instances: 800 standard-CPU years (i.e. 2.7×10^{19} updates).

Fortunately, the spin update (the core algorithm) is very simple and (in principle) trivial to parallelize. But…

Modern architectures (GPU, Xeon, Xeon-ϕ) efficient only for larger N $\longrightarrow$ astronomical number of updates ($\sim \mathrm{e}^{cN}$, probably: *strong scaling*).

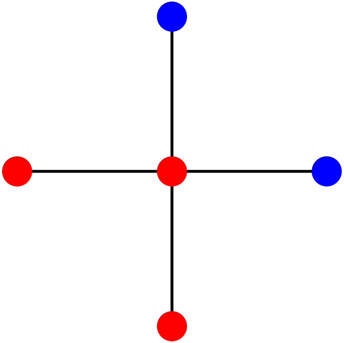
Desperate problems, desperate solutions: Janus (II)

The core algorithm

Metropolis:

An endless loop…

1. Pick a spin.



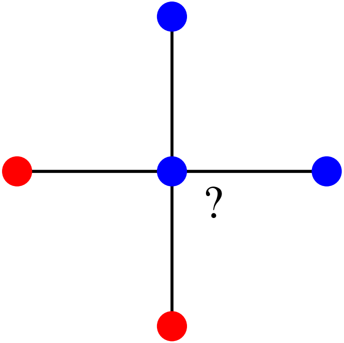
Desperate problems, desperate solutions: Janus (II)

The core algorithm

Metropolis:

An endless loop…

1. Pick a spin.
2. Flip it.
   ΔE: Energy change.
   $\Delta E < 0$?



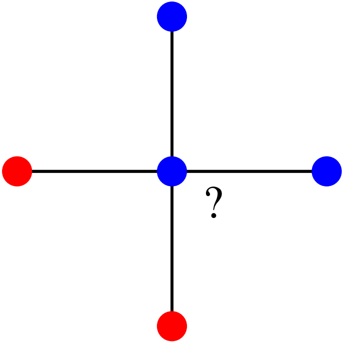
Desperate problems, desperate solutions: Janus (II)

The core algorithm

Metropolis:

An endless loop…

1. Pick a spin.
2. Flip it.
   ΔE: Energy change.
   $\Delta E < 0$?
   - Yes: done.



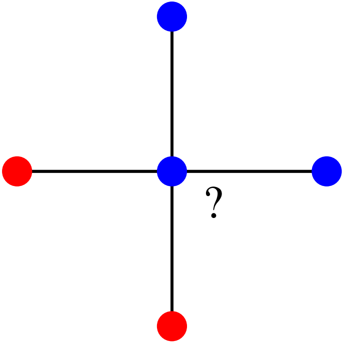
Desperate problems, desperate solutions: Janus (II)

The core algorithm

Metropolis:

An endless loop…

1. Pick a spin.
2. Flip it.
   ΔE: Energy change.
   $\Delta E < 0$?
   - Yes: done.
   - No: throw $0 < R < 1$ random.
     $R < \mathrm{e}^{-\Delta E/T}$?



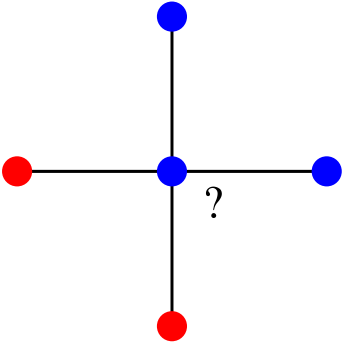
Desperate problems, desperate solutions: Janus (II)

The core algorithm

Metropolis:

An endless loop…

1. Pick a spin.
2. Flip it.
   ΔE: Energy change.
   $\Delta E < 0$?
   - Yes: done.
   - No: throw $0 < R < 1$ random.
     $R < \mathrm{e}^{-\Delta E/T}$?
     - Yes: done.



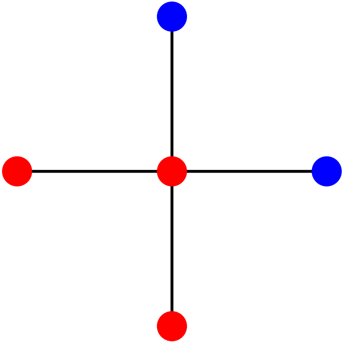
Desperate problems, desperate solutions: Janus (II)

The core algorithm

Metropolis:

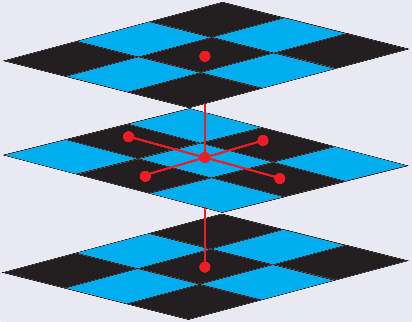
An endless loop…

1. Pick a spin.
2. Flip it.
   ΔE: Energy change.
   $\Delta E < 0$?
   - Yes: done.
   - No: throw $0 < R < 1$ random.
     $R < \mathrm{e}^{-\Delta E/T}$?
     - Yes: done.
     - No: flip back.



Desperate problems, desperate solutions: Janus (III)

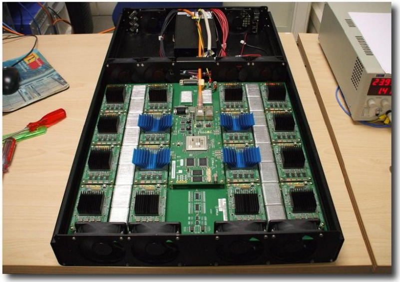
Parallelizable problem



- Parallelise within each instance
- We divide the lattice in a checkerboard scheme, all sites of the same colour can be updated simultaneously
- Memory bandwith: 13 bits to update one bit! Only solution: Memory “local to the processor”.

Desperate problems, desperate solutions: Janus (III)

Parallelizable problem



FPGA opportunity window:

- Large on-chip memory (several Mbits).
- Huge bandwidth on-chip “distributed “ memory ($\sim$ 10000 bits in and out per clock cycle).
- Large amount of logic → 1024 Spin-Update Engines.

Janus 1 (2008): ×1000 boost in spin-glasses simulations.

Green computer: ×0.001 energy consumption per update.

Janus 2: Summer 2014

Temperature chaos: the showstopper (I)

Increasing computing speed *x*1000, not such a big deal

- Pre-Janus era: up to $N = 16^3$ spins.
- Janus era: up to $N = 32^3$ spins.

Why?

Temperature chaos: the showstopper (I)

Increasing computing speed $x1000$, not such a big deal

- Pre-Janus era: up to $N = 16^3$ spins.
- Janus era: up to $N = 32^3$ spins.

Why?

We need to learn a bit about algorithms:

- Simulating at fixed temperature, simply not enough.

Temperature chaos: the showstopper (I)

Increasing computing speed *x*1000, not such a big deal

- Pre-Janus era: up to $N = 16^3$ spins.
- Janus era: up to $N = 32^3$ spins.

Why?

We need to learn a bit about algorithms:

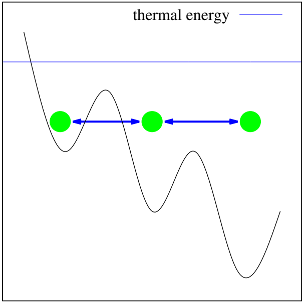
- Simulating at fixed temperature, simply not enough.
- Temperature needs to become dynamic.

Temperature chaos: the showstopper (II)

Simulated Annealing

Simplest protocol:

1. High T: easy exploration

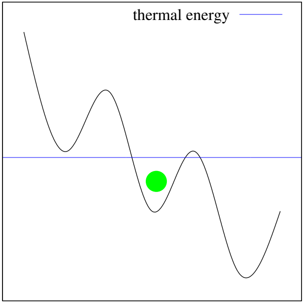


Temperature chaos: the showstopper (II)

Simulated Annealing

Simplest protocol:

1. High T: easy exploration
2. T-lowering protocol: Trapped at nearby local minimum.



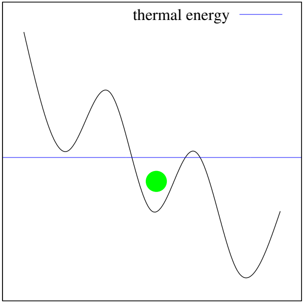
Temperature chaos: the showstopper (II)

Simulated Annealing

Simplest protocol:

1. High T: easy exploration
2. T-lowering protocol: Trapped at nearby local minimum.

Outdated algorithm.

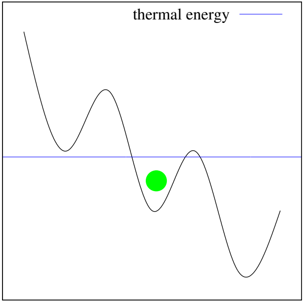


Temperature chaos: the showstopper (III)

Parallel Tempering

T raised or lowered:

1. Low *T*: local exploration

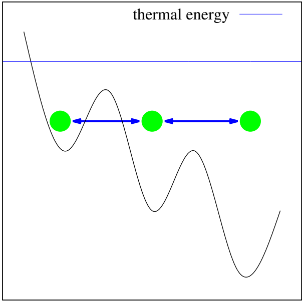


Temperature chaos: the showstopper (III)

Parallel Tempering

T raised or lowered:

1. Low *T*: local exploration
2. High *T*: global exploration

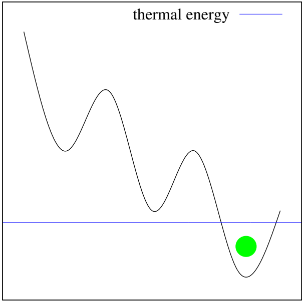


Temperature chaos: the showstopper (III)

Parallel Tempering

T raised or lowered:

1. Low *T*: local exploration
2. High *T*: global exploration
3. No trapping → better solution.

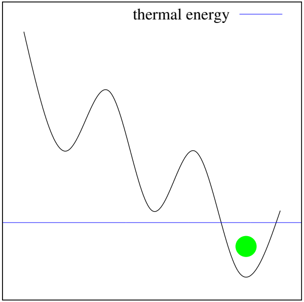


Temperature chaos: the showstopper (III)

Parallel Tempering

T raised or lowered:

1. Low T: local exploration
2. High T: global exploration
3. No trapping → better solution.



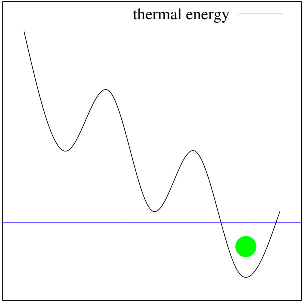
- N_T temperatures: simultaneous simulation of N_T clones (one at each temperature).

Temperature chaos: the showstopper (III)

Parallel Tempering

T raised or lowered:

1. Low T: local exploration
2. High T: global exploration
3. No trapping → better solution.



- N_T temperatures: simultaneous simulation of N_T clones (one at each temperature).
- Periodically, clones attempt to exchange their temperature. The rule preserves detailed balance.

Temperature chaos: the showstopper (IV)

It looks perfect! What can go wrong?

Temperature chaos: the showstopper (IV)

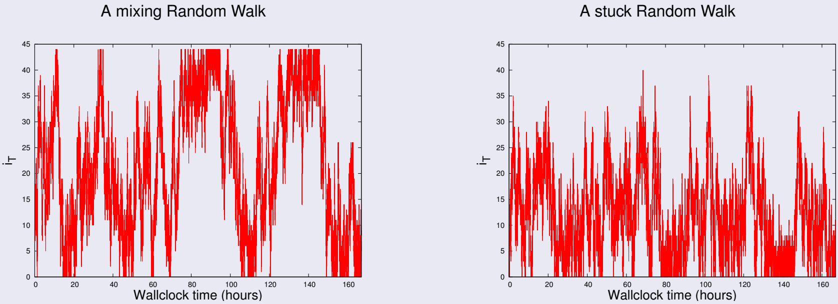
It looks perfect! What can go wrong?
Each clone performs a temperature Random Walk.

Temperature chaos: the showstopper (IV)

It looks perfect! What can go wrong?
Each clone performs a temperature Random Walk.

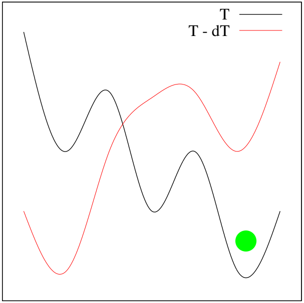
The simulation is *long enough* if all the clones visited all the temperatures several times. Mixing time: τ.

Random Walk in temperatures of a clone



Temperature chaos: the showstopper (V)

- Temperature chaos: general feature of spin-glasses.
- Relevant minima, completely different at nearby temperatures. T-random walk refuses to go across.



Temperature chaos: the showstopper (V)

- Temperature chaos: general feature of spin-glasses.
- Relevant minima, completely different at nearby temperatures. T-random walk refuses to go across.
- Temperature chaos is generic for large problem size N.

Temperature chaos: the showstopper (V)

- Temperature chaos: general feature of spin-glasses.
- Relevant minima, completely different at nearby temperatures. T-random walk refuses to go across.
- Temperature chaos is generic for large problem size N.
- In practice, specially for small N:

Temperature chaos: the showstopper (V)

- Temperature chaos: general feature of spin-glasses.
- Relevant minima, completely different at nearby temperatures. T-random walk refuses to go across.
- Temperature chaos is generic for large problem size N.
- In practice, specially for small N:
  1. The large majority of problem instances are *easy* (small τ).
  2. For some of them, though, τ inordinately large.
  3. The larger is N, the more frequently missbehaving instances appear $\longrightarrow$ difficult to assess algorithmic scaling with N.

Is quantum-computing our breakthrough? (I)

From an impressive insight (Richard P. Feynman, 1982)

NP-problems, specially simulation of quantum systems: best solved on quantum computers. . .

Is quantum-computing our breakthrough? (I)

. . . to (possibly) quantum-computing objects (2014).

D-wave Two

Is quantum-computing our breakthrough? (I)

. . . to (possibly) quantum-computing objects (2014).

A quantum annealer should:

1. Read accurately an instance.
2. Add a strong *transverse* magnetic field.

D-wave Two

Is quantum-computing our breakthrough? (I)

…to (possibly) quantum-computing objects (2014).

A quantum annealer should:

1. Read accurately an instance.
2. Add a strong *transverse* magnetic field.
3. At low enough T…
4. With low noise…
5. Slowly take field → 0.

D-wave Two

Is quantum-computing our breakthrough? (I)

…to (possibly) quantum-computing objects (2014).

A quantum annealer should:

1. Read accurately an instance.
2. Add a strong *transverse* magnetic field.
3. At low enough T…
4. With low noise…
5. Slowly take field → 0.

All requirements met? → global minimum.

D-wave Two

Is quantum-computing our breakthrough? (I)

. . . to (possibly) quantum-computing objects (2014).

A quantum annealer should:

1. Read accurately an instance.
2. Add a strong *transverse* magnetic field.
3. At low enough T. . .
4. With low noise. . .
5. Slowly take field → 0.

All requirements met? → global minimum.

D-wave Two

See talk by Bob Lucas in the next session!

Is quantum-computing our breakthrough? (II)

D-wave solves a toy problem:

- Small problems $N = 512$ (actually, $N = 503$ in USC).

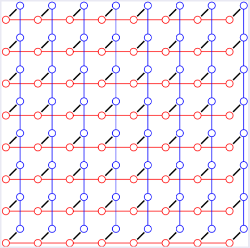
Is quantum-computing our breakthrough? (II)

D-wave solves a toy problem:

- Small problems $N = 512$ (actually, $N = 503$ in USC).
- Chimera graph: non-planar but 2D-like.

Two-dimensional penalties:

Chimera



Each blob: 8 q-bits

Is quantum-computing our breakthrough? (II)

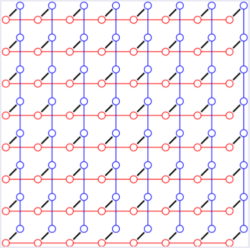
D-wave solves a toy problem:

- Small problems $N = 512$ (actually, $N = 503$ in USC).
- Chimera graph: non-planar but 2D-like.

Two-dimensional penalties:

- No SG phase for $T > 0$
  $T_c = 0 \longrightarrow$ easier problems.

Chimera



Each blob: 8 q-bits

Is quantum-computing our breakthrough? (II)

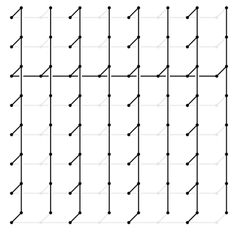
D-wave solves a toy problem:

- Small problems $N = 512$ (actually, $N = 503$ in USC).
- Chimera graph: non-planar but 2D-like.

Two-dimensional penalties:

- No SG phase for $T > 0$
  $T_c = 0 \longrightarrow$ easier problems.
- Small decycling set

Chimera



Main graph in Selby heuristics: 78% in a single tree (no loops!)

Is quantum-computing our breakthrough? (II)

D-wave solves a toy problem:

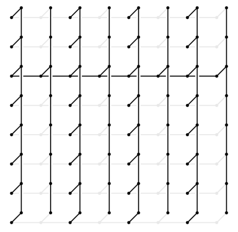
- Small problems $N = 512$ (actually, $N = 503$ in USC).
- Chimera graph: non-planar but 2D-like.

Two-dimensional penalties:

- No SG phase for $T > 0$
  $T_\mathrm{c} = 0 \longrightarrow$ easier problems.
- Small decycling set

$T = 0$ heuristics better than thermal methods (i.e. PT).

Chimera



Main graph in Selby heuristics: 78% in a single tree (no loops!)

Is quantum-computing our breakthrough? (II)

D-wave solves a toy problem:

- Small problems $N = 512$ (actually, $N = 503$ in USC).
- Chimera graph: non-planar but 2D-like.

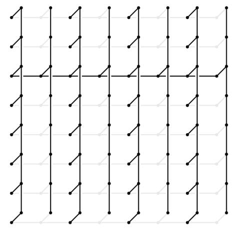
Two-dimensional penalties:

- No SG phase for $T > 0$
  $T_{\rm c} = 0 \longrightarrow$ easier problems.
- Small decycling set

  $T = 0$ heuristics better than thermal methods (i.e. PT).

  Are we learning something?

Chimera



Main graph in Selby heuristics: 78% in a single tree (no loops!)

Is quantum-computing our breakthrough? (III)

In three spatial dimensions only thermal annealing works.
The question: Is there chaos in chimera? Does D-wave overcome it?

Is quantum-computing our breakthrough? (III)

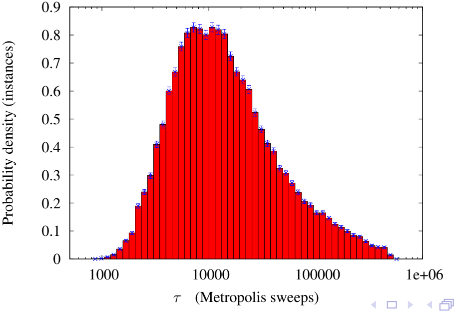
In three spatial dimensions only thermal annealing works.
The question: Is there chaos in chimera? Does D-wave overcome it?

Middleton et al.: chaos in square lattice, but $N = 2.6 \times 10^5$.
Chaos with only $N = 503$ q-bits?

Is quantum-computing our breakthrough? (III)

In three spatial dimensions only thermal annealing works.
The question: Is there chaos in chimera? Does D-wave overcome it?

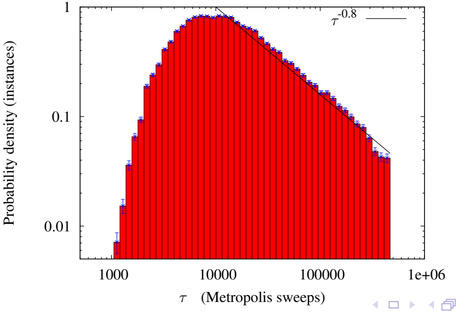
Middleton et al.: chaos in square lattice, but $N = 2.6 \times 10^5$.
Chaos with only $N = 503$ q-bits?
Not at first sight…



Is quantum-computing our breakthrough? (III)

In three spatial dimensions only thermal annealing works.
The question: Is there chaos in chimera? Does D-wave overcome it?

Middleton et al.: chaos in square lattice, but $N = 2.6 \times 10^5$.
Chaos with only $N = 503$ q-bits?
Not at first sight. . . But look at that fat tail!
2 in 10^4 instances: $\tau \gg 10^8$.



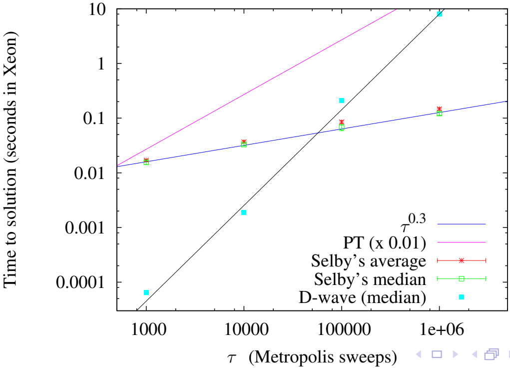
Conclusions

Meaningful algorithmic classification at fixed *N*: τ-scaling.

Conclusions

Meaningful algorithmic classification at fixed *N*: τ-scaling.

Parallel-Tempering: τ^{1}, Selby heuristics (2D!): $\tau^{b\approx 0.3}$, D-wave: $\tau^{a\approx 1.75}$.

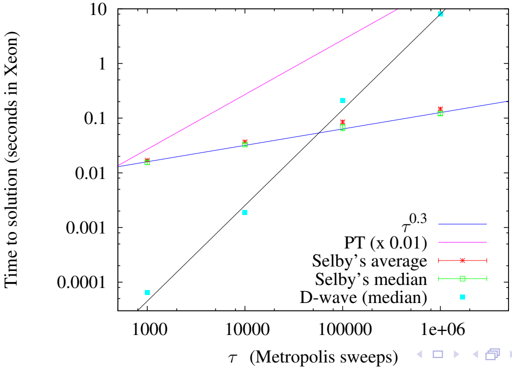


Conclusions

Meaningful algorithmic classification at fixed *N*: τ-scaling.

Parallel-Tempering: τ^1, Selby heuristics (2D!): $\tau^{b\approx 0.3}$, D-wave: $\tau^{a\approx 1.75}$.

The D-wave vs. Janus contest should be delayed until we achieve $a < 1$!



Many thanks to. . .

- The Janus collaboration, specially to:
  - Luis Antonio Fernández
  - Denis Navarro
  - Juan Jesús Ruiz-Lorenzo
- Itay Hen
- Bob Lucas, Lucio Grandinetti, and the meeting organizers
- . . . and to you (the audience), for your attention!