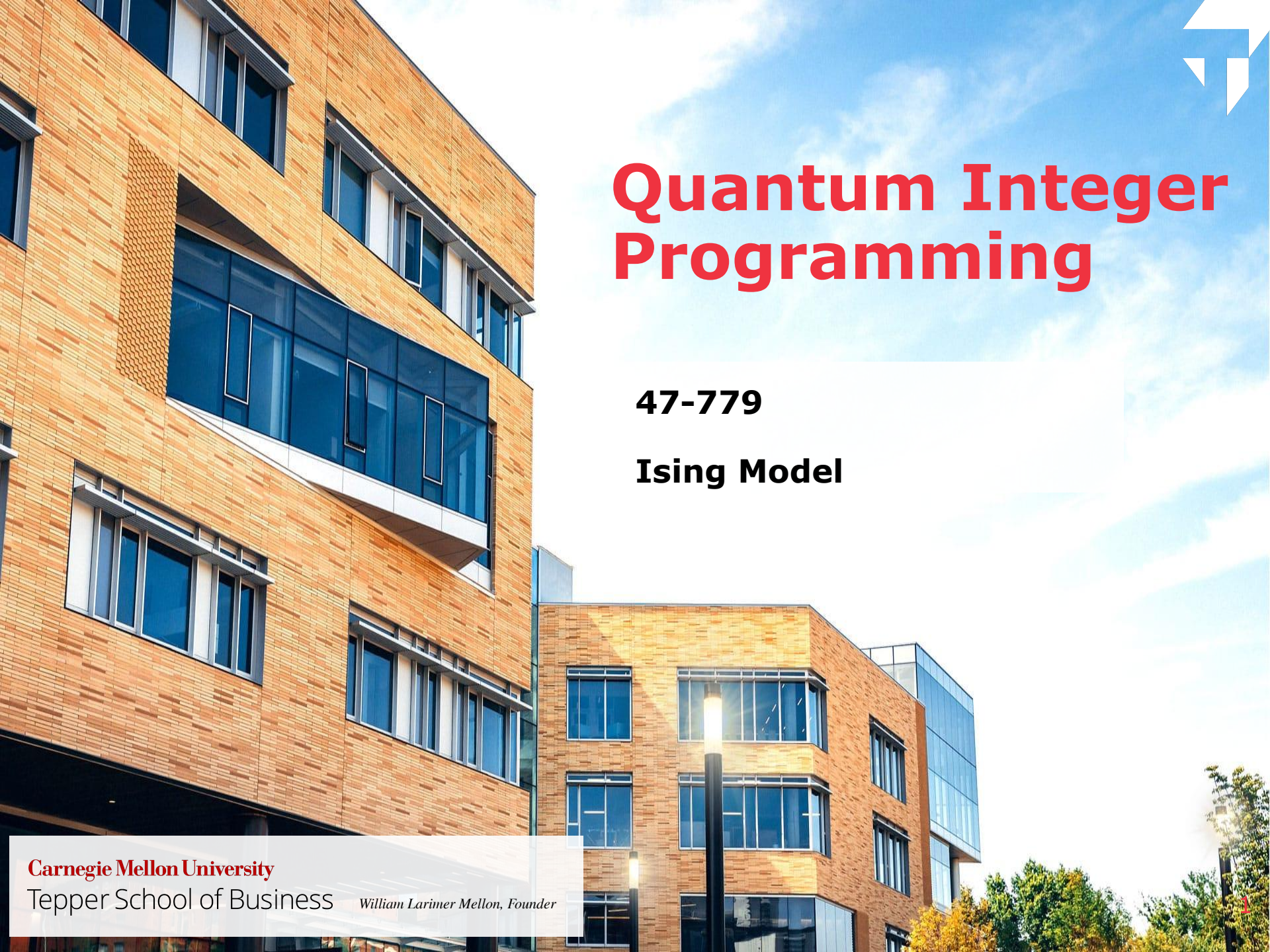


Quantum Integer Programming

47-779

Ising Model

Carnegie Mellon University
Tepper School of Business
William Larimer Mellon, Founder

Agenda

- Ising Model: Background, Physics
- Ising Model: Solutions
- Ising Model and Combinatorial Optimization
- Ising Model and Integer Programming

- Solving Ising Model: Metropolis-Hastings, MCMC Simulated Annealing
- Advanced Simulated Annealing

- Evaluating and Comparing Heuristics

Ising Model

Curiosity

1895, Pierre Curie (*Nobel Prize 1903*) finds that heating a magnet can cause it to lose its magnetic property, i.e., cause a "**phase transition**".

- But **Why?**

Model

1920 - Lenz introduced a model to explain this phase transition.

1925 - Lenz's student, Ising, solved a special 1-D case of the model

1940 - Onsager (*Nobel Prize 1968*) solves the 2-D case.

2000 - Istrail shows, via a Max-Cut formulation, that the much sought after 3-D case is NP-Complete

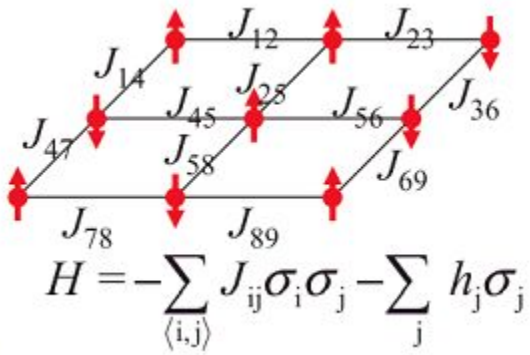
General lesson

1971 - Wilson (*Nobel Prize 1982*)**, Universality:** Systems with same number of dimensions and symmetries go through identical phase transitions.

Ising is the simplest model in theory space to captures properties of all sorts of interacting systems like magnets, water etc.

Ising Model

Mental model and applications



$$\sum_{s_1=0}^{1}\sum_{s_2=0}^{1}\cdots\sum_{s_n=0}^{1}$$

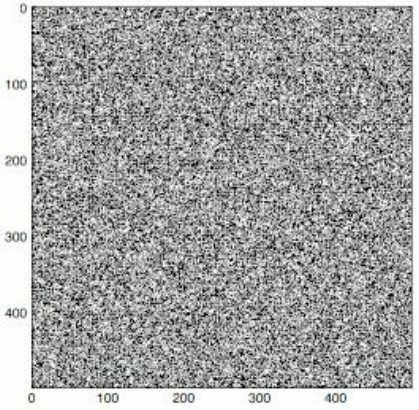
$$H = -J\sum_{i=1}^{N} s_i s_{i+1} - H\sum_{i=1}^{N} s_i$$

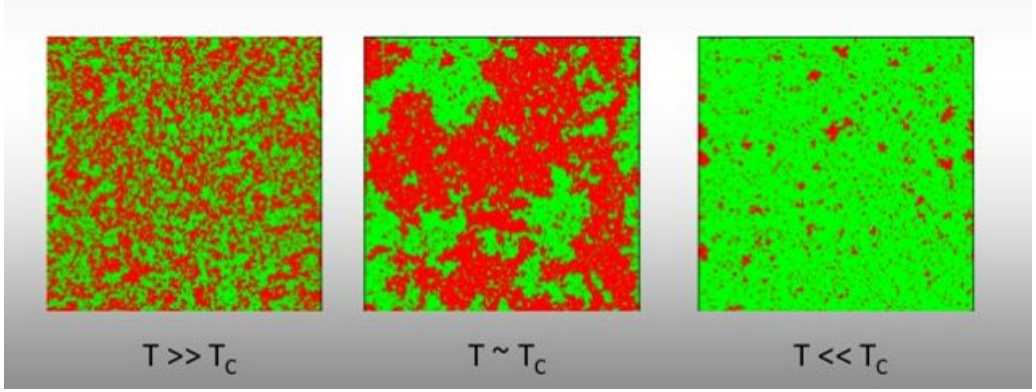
$$H = -\sum_{i=1}^{N}\left[Js_i s_{i+1} + H\frac{s_i + s_{i+1}}{2}\right]$$

$$Z = e^{\Psi} = \sum_j e^{-\beta H(\mathbf{x}_j, \mathbf{p}_j)}$$

$$z(x) = \begin{cases} -1 & \text{if } x = 0 \\ +1 & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$Z = \sum^{1}\sum^{1}\cdots\sum^{1}\exp\left(\beta\sum^{N}\left[Jz(s_i)z(s_{i+1}) + H\frac{z(s_i) + z(s_{i+1})}{2}\right]\right)$$





[1] https://en.wikipedia.org/wiki/Ising_model

Ising Model

Mathematical definition

$$H(\sigma) = -\sum_{(ij)\in E(G)} J_{ij}\sigma_i\sigma_j - \mu\sum_{i\in V(G)} h_i\sigma_i$$

- H Energy function or Hamiltonian
- $\sigma_i \in \{-1,+1\}^{V(G)}$ Spins for each site in the graph or lattice
- $G=(V,E)$ Graph or Lattice defining the interactions
- μ Magnetic moment
- $J_{ij}\begin{cases} >0, & \text{ferromagnetic interaction} \\ <0, & \text{antiferromagnetic interaction} \\ =0, & \text{no interaction} \end{cases}$

(Quadratic) Couplings

- $h_i\begin{cases} >0, & \text{site wanting to align with external field} \\ <0, & \text{site wanting to anti-align with external field} \\ =0, & \text{no external influence on site} \end{cases}$

Zeeman term, external longitudinal term, bias, ...

- Configuration (Boltzmann) probability: $P(\sigma,\beta) = e^{-\beta H(\sigma)}/Z(\beta)$
- Inverse temperature: $\beta = (k_B T)^{-1}$
- Partition function: $Z(\beta) = \sum_\sigma e^{-\beta H(\sigma)}$ (normalization in probability)

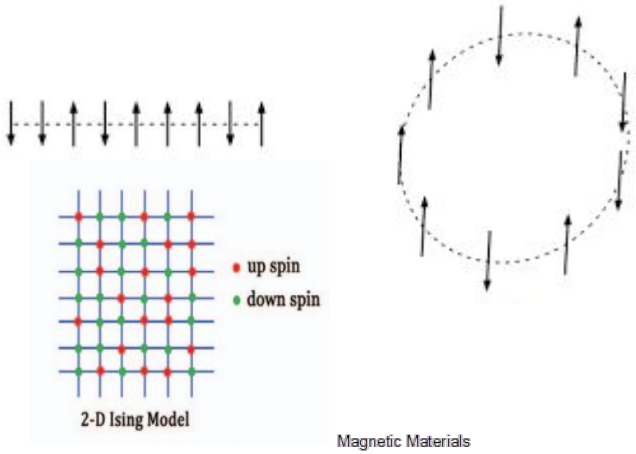
Ising Model - Solutions

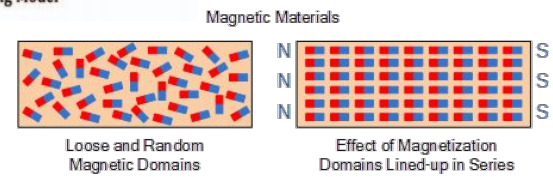
Solutions

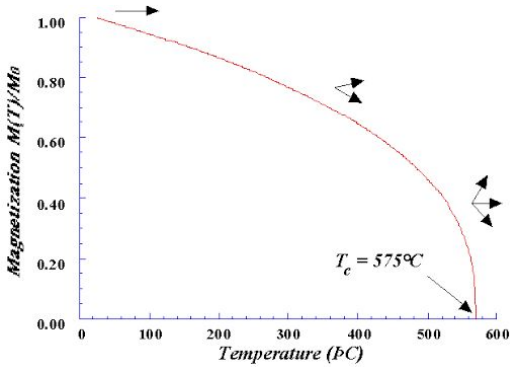
- 1d with circular or free boundary conditions, without external field.
  - Ising solution: No phase transition
- 1d with external field.
  - Phase transition at $J = h$
- 2d case
  - Onsager solution: Phase transition

$$M = \frac{1}{N}\sum_i \sigma_i = (1 - [\sinh 2\beta J_{vert} \sinh 2\beta J_{horz}]^{-2})^{1/8}$$

- 3d+ case
  - If graph is nonplanar, then the problem is NP-complete (proof via MAXCUT)
  - Mean-field approximation (assume continuity in interactions)







But what does it mean to solve this problem?

For some: compute meaningful statistical properties

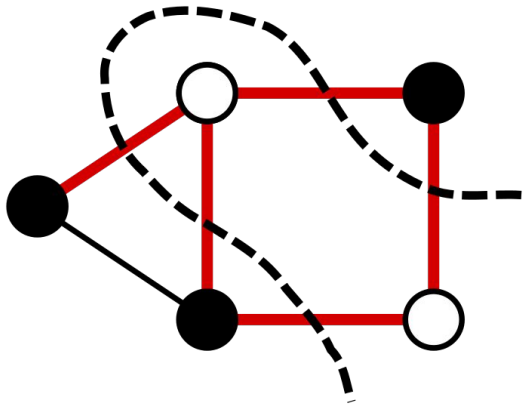
For others: What are the values of the spins?

[1] https://www.electronics-tutorials.ws/electromagnetism/magnetism.html
[2] http://www.irm.umn.edu/hg2m/hg2m_b/hg2m_b.html

Ising Model - Combinatorial Optimization

Starting from Ising Problem without external field

$$
\begin{aligned}
H(\sigma) &= -\sum_{(ij)\in E(G)} J_{ij}\sigma_i\sigma_j \\
&= -\sum_{(ij)\in E(V^+)} J_{ij} - \sum_{(ij)\in E(V^-)} J_{ij} + \sum_{(ij)\in \delta(V^+)} J_{ij} \\
&= -\sum_{(ij)\in E(G)} J_{ij} + 2\sum_{(ij)\in \delta(V^+)} J_{ij}
\end{aligned}
$$



where the set V^+ (V^-) are all the vertices with $\sigma = +1$ $(\sigma = -1)$ and their boundary (cut) is denoted by $\delta(V^+)$

Now consider that the graph has weighted edges W_{ij}

Then the size of the cut is $|\delta(V^+)| = \frac{1}{2}\sum_{(i,j)\in\delta(V^+)} W_{ij}$

Therefore we obtain $H(\sigma) = \sum_{(ij)\in E(G)} W_{ij} - 4|\delta(V^+)|$

When minimizing the Ising model, we are finding the maximum cut of the graph

$$\min_\sigma H(\sigma) = \sum_{(ij)\in E(G)} W_{ij} + 4\max_\sigma |\delta(V^+)|$$

[1] https://www.electronics-tutorials.ws/electromagnetism/magnetism.html
[2] http://www.irm.umn.edu/hg2m/hg2m_b/hg2m_b.html

Ising Model - Combinatorial Optimization

Starting from the minimization of the Ising Model

$$\min_{\sigma\in\{-1,+1\}^n} H(\sigma) = \min_{\sigma\in\{-1,+1\}^n} \sum_{(ij)\in E(G)} J_{ij}\sigma_i\sigma_j + \sum_{i\in V(G)} h_i\sigma_i$$

We can directly pose this problem as an Quadratic Unconstrained Binary Optimization (QUBO). The next lecture is going to be on this!

$$\min_{\sigma\in\{-1,+1\}^n} \sum_{(ij)\in E(G)} J_{ij}\sigma_i\sigma_j + \sum_{i\in V(G)} h_i\sigma_i =$$

$$\min_{\mathbf{x}\in\{0,1\}^n} \sum_{(ij)\in E(G)} x_i Q_{ij} x_j + \sum_{i\in V(G)} Q_{ii}x_i + c$$

with $Q_{ij} = 4J_{ij}, Q_{ii} = 2h_i - \sum_{j\in V(G)}(2J_{ij} + 2J_{ji}), c = \sum_{i<j} J_{ij} - \sum_{i\in V(G)} h_i$

Although this is already solvable using INLP programming tools, we can reformulate it as a ILP by adding a variable $x_{ij} = x_i x_j$ whose nonlinearity can be posed a linear inequalities.

Experimental results show this is the most efficient ILP formulation of the Ising problem

$$\min_{\mathbf{x}\in\{0,1\}^n} \sum_{(ij)\in E(G)} Q_{ij}x_{ij} + \sum_{i\in V(G)} Q_{ii}x_i + c$$

$$s.t.\, x_{ij} \geq x_i + x_j - 1, x_{ij} \leq x_i, x_{ij} \leq x_j \quad \forall (ij) \in E(G)$$

$$x_i \in \{0,1\} \quad \forall i \in V(G), x_{ij} \in \{0,1\} \quad \forall (ij) \in E(G)$$

[1] Billionnet, A., Elloumi, S.: Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. Mathematical Programming 109(1) (2007) 55–68

Ising problem as IP

Let’s go to the code

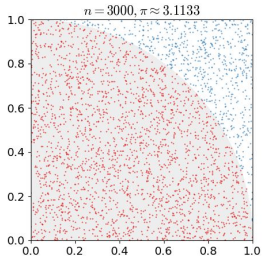
https://colab.research.google.com/github/bernalde/QuIP/blob/master/notebooks/Notebook%204%20-%20Ising%20Model.ipynb

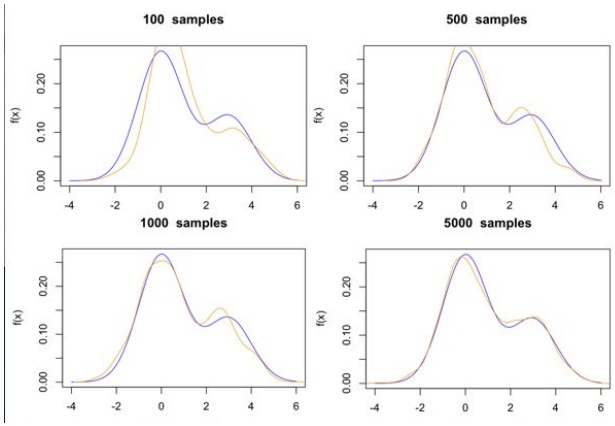
Metropolis-Hastings algorithm

Monte Carlo methods

Algorithms relying on random number generation.

1. Define domain of possible input.
2. Generate those inputs following a probability distribution.
3. Perform deterministic computation on the inputs.
4. Aggregate the results.





Markov-chain Monte Carlo

Generate a target distribution by sampling a Markov-chain with equilibrium distribution being the target.

Metropolis-Hastings

We want to approximate a distribution $P(x)$ using an initial function $f(x)$

Given initial function $f(x)$and a given point x_i in iteration i

1. Compute a new point to evaluate $\bar{x}$ from an arbitrary probability density $Q(\bar{x}|x_i)$
2. Calculate an acceptance ratio of that point based on $\alpha = f(\bar{x})/f(x_i)$
3. If $\alpha \leq \text{Rand}[0,1], x_{i+1} := \bar{x}$, else $x_{i+1} := x_i$

[1] https://en.wikipedia.org/wiki/Monte_Carlo_method
[2] https://en.wikipedia.org/wiki/Markov_chain_Monte_Carlo
[3] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and EdwardTeller. Equation of state calculations by fast computing machines.The Journal of Chemical Physics,21(6):1087–1092, 1953.

Ising Model - MCMC

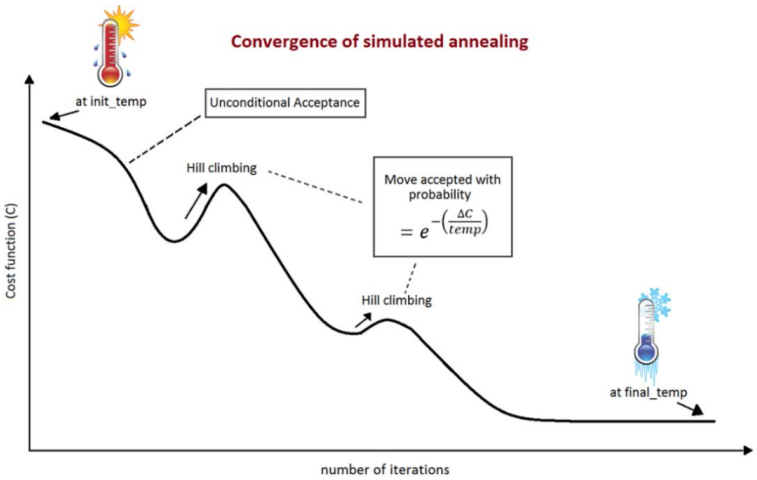
Ising model as Markov-Chain

The immediate probability $P(\boldsymbol{\sigma}^c, \beta) = e^{-\beta H(\boldsymbol{\sigma}^c)} / Z(\beta)$ of transitioning to a future state $\boldsymbol{\sigma}^f$ depends only in the current state $\boldsymbol{\sigma}^c = [\sigma_1^c, \cdots, \sigma_N^c]$

Given single flip dynamics, we can jump from any state to another.

Metropolis-Hastings Monte Carlo Algorithm for Ising Models

1) Start with a known configuration, $\boldsymbol{\sigma}^i = [\sigma_1^i, \cdots, \sigma_N^i]$ corresponding energy, $H(\boldsymbol{\sigma}^i)$ and temperature value $T = (k_B \beta)^{-1}$
2) Randomly change the configuration
   - Add small displacement $\boldsymbol{\sigma}^j = \boldsymbol{\sigma}^i + \delta$
3) Calculate new energy value $H(\boldsymbol{\sigma}^j)$
4) Compare to energy at previous position
   - If $H(\boldsymbol{\sigma}^j) < H(\boldsymbol{\sigma}^i)$ keep new position
   - If $H(\boldsymbol{\sigma}^j) > H(\boldsymbol{\sigma}^i)$ keep new position if Boltzmann factor for transition satisfies $\exp\left[-\frac{H(\boldsymbol{\sigma}^j) - H(\boldsymbol{\sigma}^i)}{k_B T}\right] \geq \text{Rand}[0,1]$
5) Repeat 2) - 4) K times



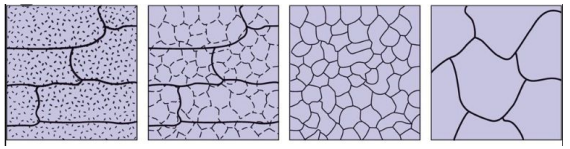
[1] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing.Science,220(4598):671–680, 1983.

William Larimer Mellon, Founder

Simulated Annealing

Concept coming from annealing in metallurgy

Slow cooling allows for perfect crystals (minimizing energy)

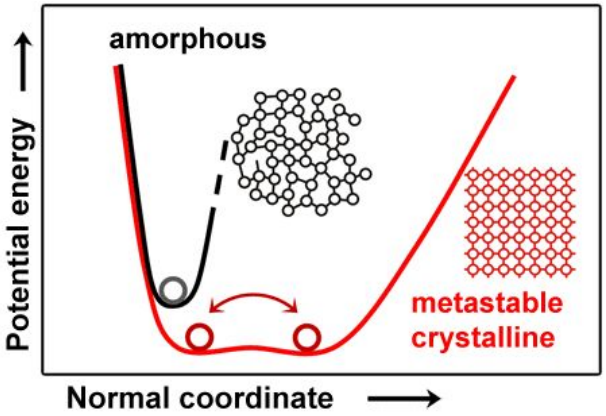


Simulated Annealing provides a temperature schedule for the Metropolis-Hastings method

1) Start at effective high temperature and gradually decrease the temperature by increments until

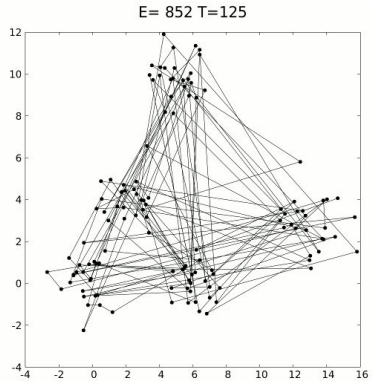
   is slightly above zero

2) At every temperature the Metropolis algorithm is run in a nested-loop



Interesting behavior:

- "Divide-and-conquer": Big features are solved early in the search and small features later while refining
- Ability to escape local-minima
- Guaranteed to reach lowest energy if temperature is lowered slowly enough



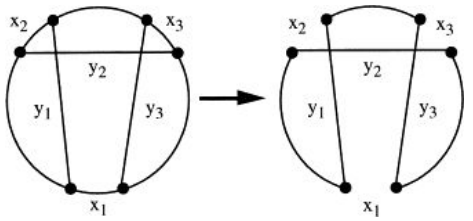
[1] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing.Science,220(4598):671–680, 1983.
[2] https://www.esrf.eu/news/general/phase-change-materials/index_html
[3] Alan Lang Chapter 8 Strain hardening and annealing.

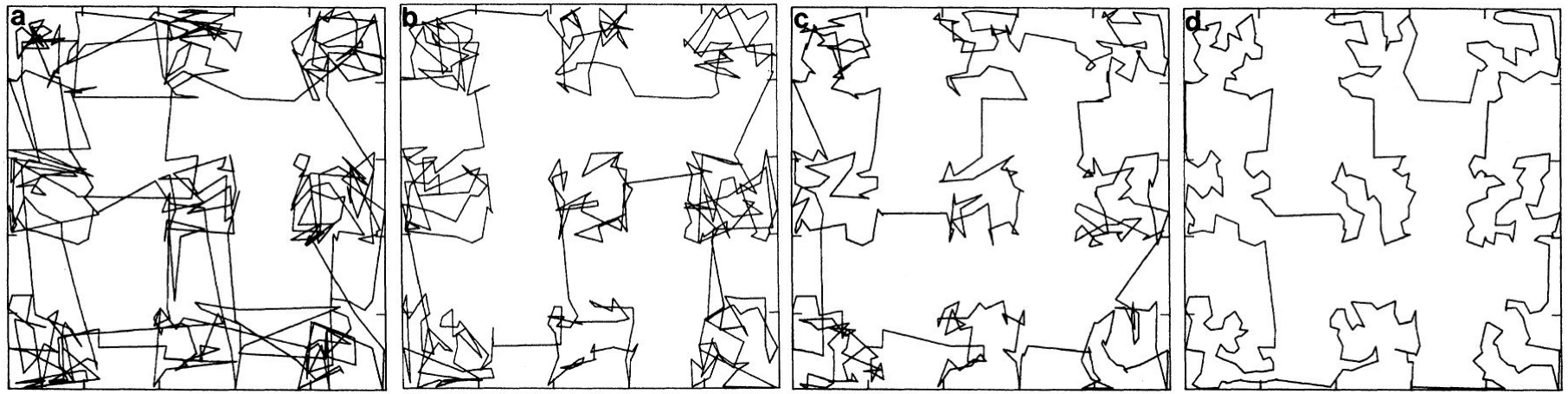
Simulated Annealing Results

For Traveling Salesman Problem (TSP)

Given a set of cities, an agent needs to visit them all once, reducing the total distance traveled.

- The most famous combinatorial optimization problem
- Back when simulated annealing was proposed was able to solve problems up to 6000 cities whereas other methods could only handle 30 cities
- The displacement δ is given by Lin and Kerrighan heuristic





$T = 1.2$ $T = 0.8$ $T = 0.4$ $T = 0.0$

[1] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing.Science,220(4598):671–680, 1983.
[2] Helsgaun, Keld. "An effective implementation of the Lin–Kernighan traveling salesman heuristic." European Journal of Operational Research 126.1 (2000): 106-130.

Simulated Annealing

Let’s go back to the code

https://colab.research.google.com/github/bernalde/QuIP/blob/master/notebooks/Notebook%204%20-%20Ising%20Model.ipynb

Advanced Simulated Annealing

As seen before, the displacement δ is key to performance.

In naive Simulated annealing the displacement can be a "single flip" $\delta : \sigma_i \mapsto -\sigma_i$

- For hard optimization problem this might require exponential time to converge.

What if the update happens between "clusters" of spins?

- This needs to be done carefully to guarantee energy conservation and ergodicity
  - In this context that one can reach any state from another given the Markov-chain
- Generate different replicas of the system at different temperatures and after certain Metropolis updates, the temperatures of two replicas r_1, r_2 are exchanged if

$$P(r_1 \leftrightarrow r_2) = \min\{1, \exp[(\beta_1 - \beta_2)(H(\sigma^1) - H(\sigma^2))]\}$$

  - Two temperatures are always exchanged if a replica at higher temperature has a lower energy than a replica with a lower temperature.
  - Otherwise, the exchange of the two temperatures is either accepted or rejected using the random number between 0 and 1

[1] S. Mandra, Z. Zhu, W. Wang, A. Perdomo-Ortiz, H. G. Katzgraber. Strengths and weaknesses of weak-strong cluster problems: A detailed overview of state-of-the-art classical heuristics versus quantum approaches. Physical Review A, 94(2), 022,337, 2016.

How to evaluate heuristics?

This is not a trivial question given that methods may have several parameters to tune, run on different hardware or there is no clear absolute metric.

Important metrics are time and solution quality.

Given an algorithm that runs several times, you would like to know how much should it take for you to get a solution with certain success probability.

Metric: Time to solution of expected runtime

$$TTS(m) = m\tau(m)\frac{\log(1-s)}{\log(1-p(m))}$$

- m number of times run, or sweeps in Simulated Annealing
- s success probability after m sweeps
- $p(m)$ probability of success to achieve (usually high $s = 0.99$)
- $\tau(m)$ time it takes to perform a single sweep

- $m\tau(m)$ Time the algorithm runs

It's going to be useful once benchmarking Quantum methods

[1] Venturelli, Davide, et al. "Quantum optimization of fully connected spin glasses." Physical Review X 5.3 (2015): 031040.

Time to solution

Let’s go back to the code

https://colab.research.google.com/github/bernalde/QuIP/blob/master/notebooks/Notebook%204%20-%20Ising%20Model.ipynb

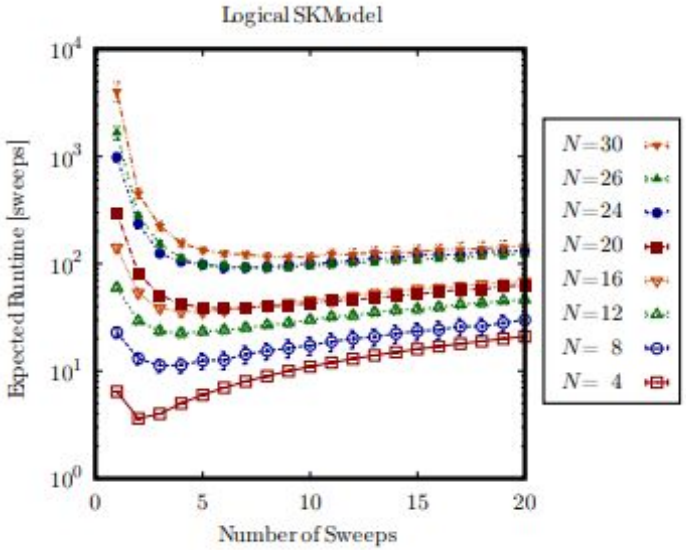
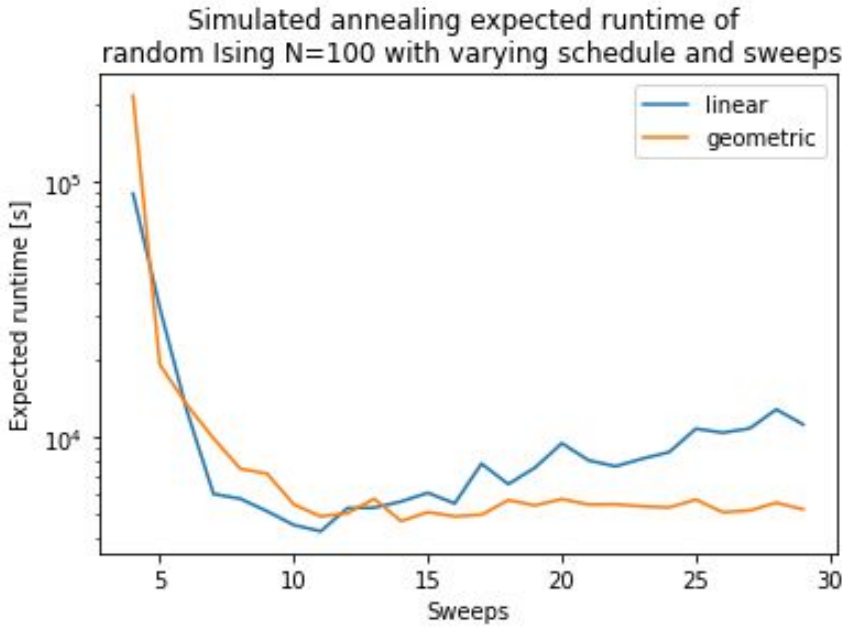


Figure 5: **Expected computational time T_{exp} as in Eq. (4), by varying the number of sweeps for the logical SK model, at different number of logical spins N.** The optimal number of sweeps (speed) is defined as the number of sweeps which minimizes T_{exp}.



[1] Venturelli, Davide, et al. "Quantum optimization of fully connected spin glasses." Physical Review X 5.3 (2015): 031040.

How to compare heuristics?

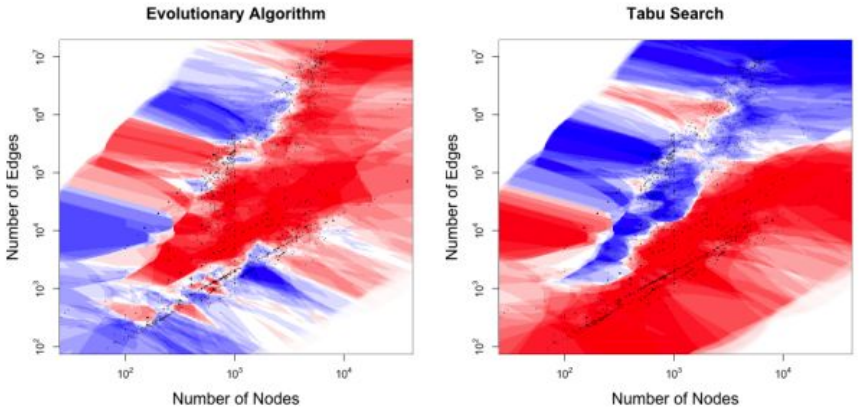
Several heuristics available for Max-Cut and QUBO

They compared 37 heuristics to solve these problems on the same computer, with similar implementation on an available library of problems

Problem	Name	Count	Description: Nodes	Description: Density	Reference
Max-Cut	G-set	71	800–20,000	0.0%–6.0%	Helmberg and Rendl (2000)
	Spin Glass	30	125–2,744	0.2%–4.8%	Burer et al. (2002)
	Torus	4	512–3,375	0.2%–1.2%	7^{th} DIMACS Implementation Challenge
QUBO	GKA	45	21–501	8.0%–99.0%	Glover et al. (1998)
	Beasley	60	51–2,501	9.9%–14.7%	Beasley (1998)
	P3-7	21	3,001–7,001	49.8%–99.5%	Palubeckis (2006)

Trained a decision tree to help users decide which heuristic to use in their problem.

"*We [Dunning, Gupta, Silberholz] evaluated each of the 37 heuristics over all 3,296 problem instances in the expanded instance library, consuming 2.0 CPU-years of processing power (20.1 CPU-days per heuristic), taking 12.4 days over the 60 machines (8.0 hours per heuristic), and costing $1,196 ($32.3 per heuristic)*"



Good Bad

[1] Dunning, Iain, Swati Gupta, and John Silberholz. "What works best when? A systematic evaluation of heuristics for Max-Cut and QUBO." INFORMS Journal on Computing 30.3 (2018): 608-624.
[2] https://github.com/MQLib/MQLib

How to compare heuristics?

Several heuristics available for Max-Cut and QUBO

Paper	Type	Short name	Description	Space Req.
Alkhamis et al. (1998)	Q	ALK98	Simulated annealing	$O(n+m)$
Beasley (1998)	Q	BEA98SA	Simulated annealing	$O(n+m)$
		BEA98TS	Tabu search	$O(n+m)$
Burer et al. (2002)	M	BUR02	Non-linear optimization with local search	$O(n+m)$
Duarte et al. (2005)	M	DUA05	Genetic algorithm with VNS as local search	$O(pn+m)$
Festa et al. (2002)	M	FES02G	GRASP with local search	$O(n+m)$
		FES02GP	GRASP with path-relinking	$O(\lambda n+m)$
		FES02V	VNS	$O(n+m)$
		FES02VP	VNS with path-relinking	$O(\lambda n+m)$
		FES02GV	GRASP with VNS local search	$O(n+m)$
		FES02GVP	GRASP & VNS with path-relinking	$O(\lambda n+m)$
Glover et al. (1998)	Q	GLO98	Tabu search	$O(tn+m)$
Glover et al. (2010)	Q	GLO10	Tabu search with long-term memory	$O(\lambda n+m)$
Hasan et al. (2000)	Q	HAS00GA	Genetic algorithm	$O((p+\tau)n+m)$
		HAS00TS	Tabu search	$O(n+m)$
Katayama et al. (2000)	Q	KAT00	Genetic algorithm with k-opt local search	$O(pn+m)$
Katayama and Narihisa (2001)	Q	KAT01	Simulated annealing	$O(n+m)$
Laguna et al. (2009)	M	LAG09CE	Cross-entropy method	$O(5.87\rho n^2+m)$
		LAG09HCE	Cross-entropy method with local search	$O(0.031n^2+m)$
Lodi et al. (1999)	Q	LOD99	Genetic algorithm	$O(pn+m)$
Lü et al. (2010)	Q	LU10	Genetic algorithm with tabu search	$O(pn+m)$
Merz and Freisleben (1999)	Q	MER99LS	Genetic algorithm, with crossover and local search	$O(\beta n+m)$
		MER99MU	Genetic algorithm, with mutation only	$O(\beta n+m)$
		MER99CR	Genetic algorithm, with crossover only	$O(\beta n+m)$
Merz and Freisleben (2002)	Q	MER02GR	GRASP without local search	$O(n+m)$
		MER02LS1	1-opt local search with random restarts	$O(n+m)$
		MER02LSK	k-opt local search with random restarts	$O(n+m)$
		MER02GRK	k-opt local search with GRASP	$O(n+m)$
Merz and Katayama (2004)	Q	MER04	Genetic algorithm, with k-opt local search	$O(\alpha n+m)$
Palubeckis (2004)	Q	PAL04T1	Tabu search	$O(n+m)$
		PAL04T2	Iterated tabu search	$O(n+m)$
		PAL04T3	Tabu search with GRASP	$O(n+m)$
		PAL04T4	Tabu search with long-term memory	$O((\lambda+\gamma)n+m)$
		PAL04T5	Iterated tabu search	$O(n+m)$
		PAL04MT	Tabu search	$O(n+m)$
Palubeckis (2006)	Q	PAL06	Iterated tabu search	$O(n+m)$
Pardalos et al. (2008)	Q	PAR08	Global equilibrium search	**

Heuristic	Solution (%) 5 n Best-of-5 Deviation	Avg. Rank
BUR02	**0.2**	10.7
FES02GVP	0.4	10.1
PAL04T3	0.5	**7.5**
FES02GP	0.6	14.2
FES02GV	0.6	11.2
PAL04T2	0.3	8.5
BEA98TS	2.1	16.6
LU10	1.2	13.1
FES02G	1.1	18.2
MER04	0.5	8.6
PAL04T1	1.9	15.2
MER99LS	0.3	10.0
MER02GRK	0.5	11.9
MER02LSK	0.7	13.3
PAL04T5	2.0	18.6
PAL06	1.4	17.3
ALK98	0.5	12.5
PAL04T4	2.6	21.7
GLO10	1.3	16.6
FES02V	0.9	13.8
KAT00	0.9	16.9
FES02VP	0.7	12.4
PAL04MT	2.8	24.1
MER02GR	2.8	24.3
GLO98	1.5	23.4
HAS00TS	1.1	20.3
HAS00GA	1.6	20.9
MER02LS1	2.6	25.1
PAR08	2.1	17.9
KAT01	2.1	26.3
DUA05	1.4	17.4
LOD99	4.9	30.2
LAG09HCE	1.6	20.4
BEA98SA	3.2	30.8
MER99CR	12.9	34.6
MER99MU	21.7	35.7
LAG09CE	29.2	36.9

[1] Dunning, Iain, Swati Gupta, and John Silberholz. "What works best when? A systematic evaluation of heuristics for Max-Cut and QUBO." INFORMS Journal on Computing 30.3 (2018): 608-624.
[2] https://github.com/MQLib/MQLib