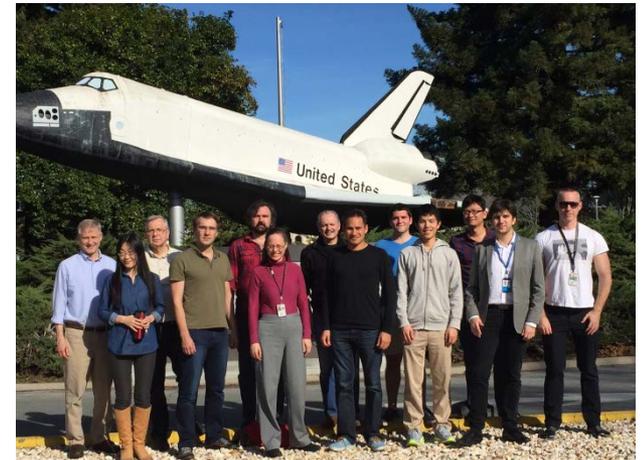


Quantum Annealing Programming Techniques for Discrete Optimization Problems

OUTLINE

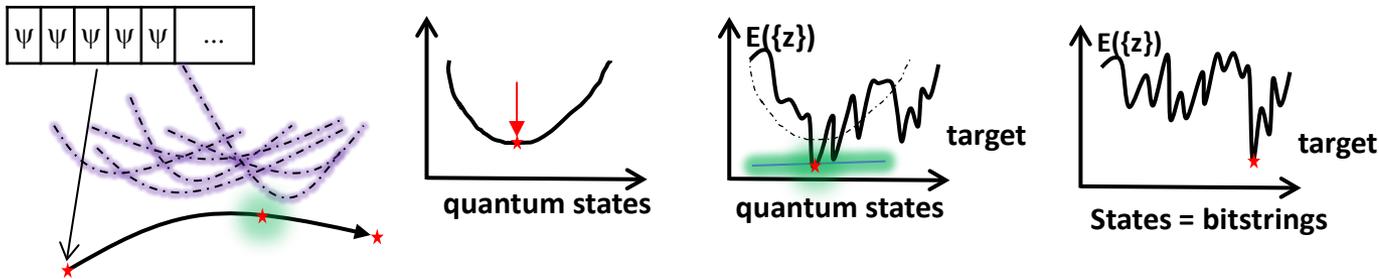
- Primer about quantum annealing and its challenges
- General programming/compiling/running applications
- The example of Job-Shop-Scheduling (JSP)
- Programming (mapping) techniques
- Compiling (embedding) techniques
- Running (annealing) techniques
- Annealing for problems of ASCR interests



<http://www.nas.nasa.gov/quantum>

Davide Venturelli, Alejandro Perdomo-Ortiz, Eleanor G. Rieffel, Bryan O’Gorman (NASA)
JSP In collaboration with: Dominic Marchand, Galo Rojo (1QBit)

PRIMER ABOUT QUANTUM ANNEALING

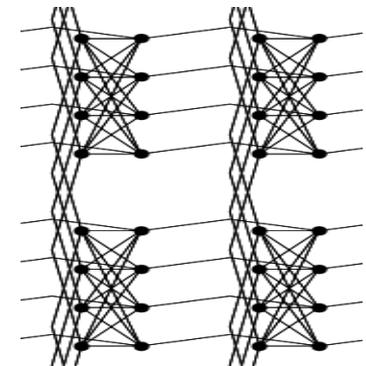


(Knysh et al. 2015)

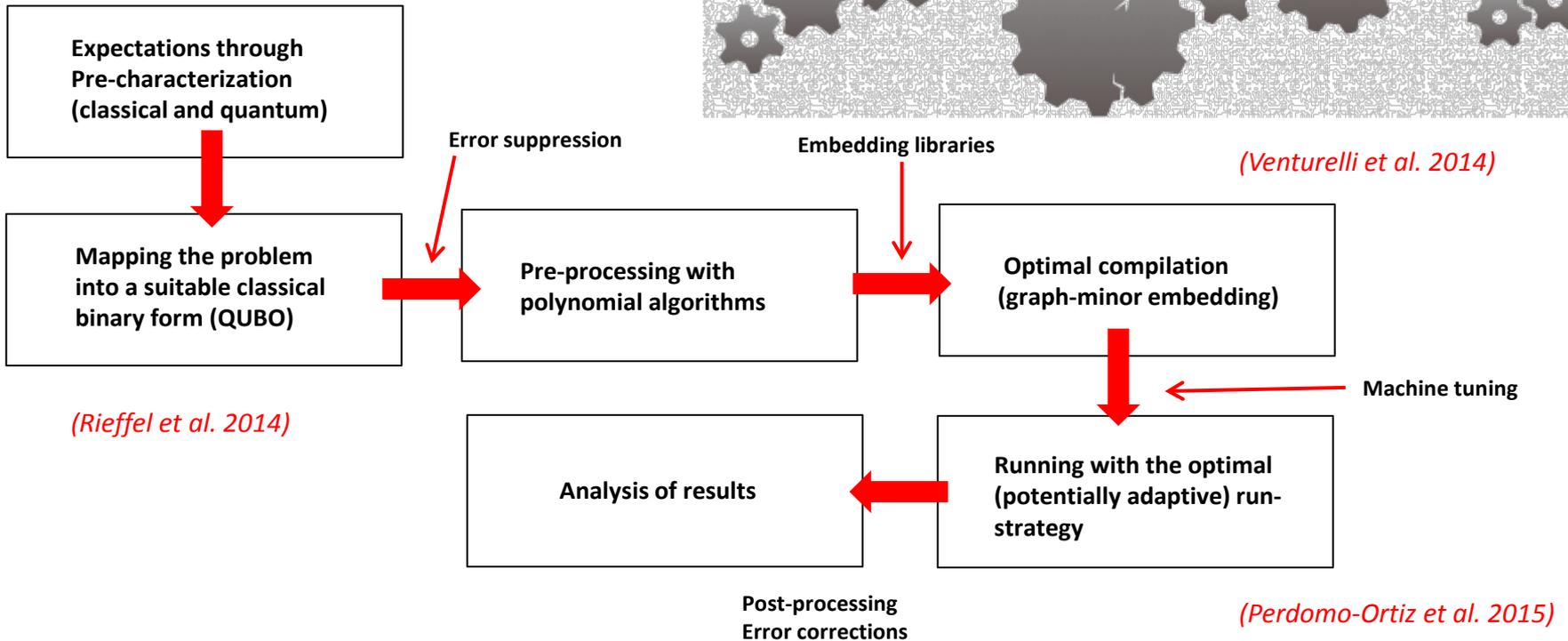
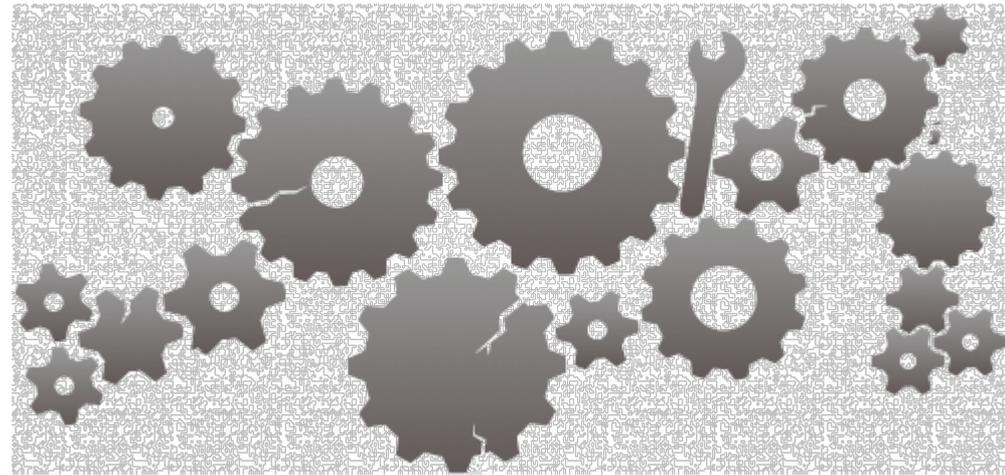
- Bottleneck is typically the **minimum gap**, but it is difficult to locate or compute, not necessarily at QPT.
- **No proof of exponential speedup** for easy to conceive Hamiltonians, mapping to universality unpractical.
- Research directly on this topic has only 15 years of history, with less than **300 theory papers published**.

PRIMER ABOUT QUANTUM ANNEALING DEVICES (e.g. D-Wave Two)

- They do not support arbitrary problem Hamiltonians: D-Wave has an Ising model on a “**chimera lattice**”
- They do not support arbitrary drivings: D-Wave has a stoquastic **transverse field**
- They do not operate as a closed system at $T=0$, D-Wave qubit decoherence is $\approx 5\text{ns}$ and **operating temperature $\approx 15\text{mK}$**
- Hamiltonian parameters are specified within a given precision, and they fluctuates over the annealing time, D-Wave has **$\approx 5\%$ of precision**
- They do not support arbitrary schedules, D-Wave has a **min $20\mu\text{s}$ annealing quench** interlaced with the problem Hamiltonian energy
- They cannot encode arbitrary energies: D-Wave machine **maximum energy is 3.2 Ghz**



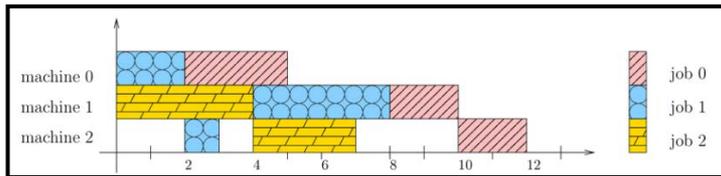
(Johnson et al. 2011)



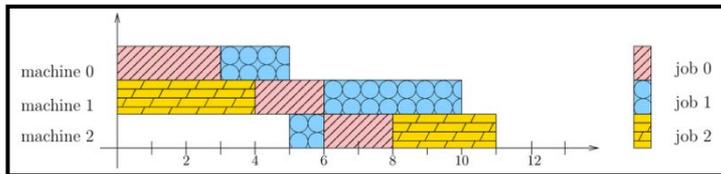
THE PROBLEM: A SIMPLE 3X3 EXAMPLE

	1 st operation	2 nd operation	3 rd operation
JOB 0	Machine 0 for 3t	Machine 1 for 2t	Machine 2 for 2t
JOB 1	Machine 0 for 2t	Machine 2 for 1t	Machine 1 for 4t
JOB 2	Machine 1 for 3t	Machine 2 for 3t	Machine 2 for 3t

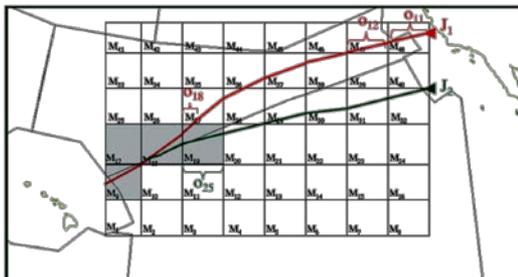
Feasible schedule with makespan 12



Optimal schedule with makespan 11

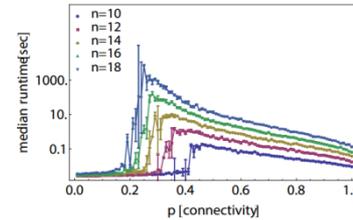


AERONAUTICS APPLICATIONS



(Banavar et al. 2007)

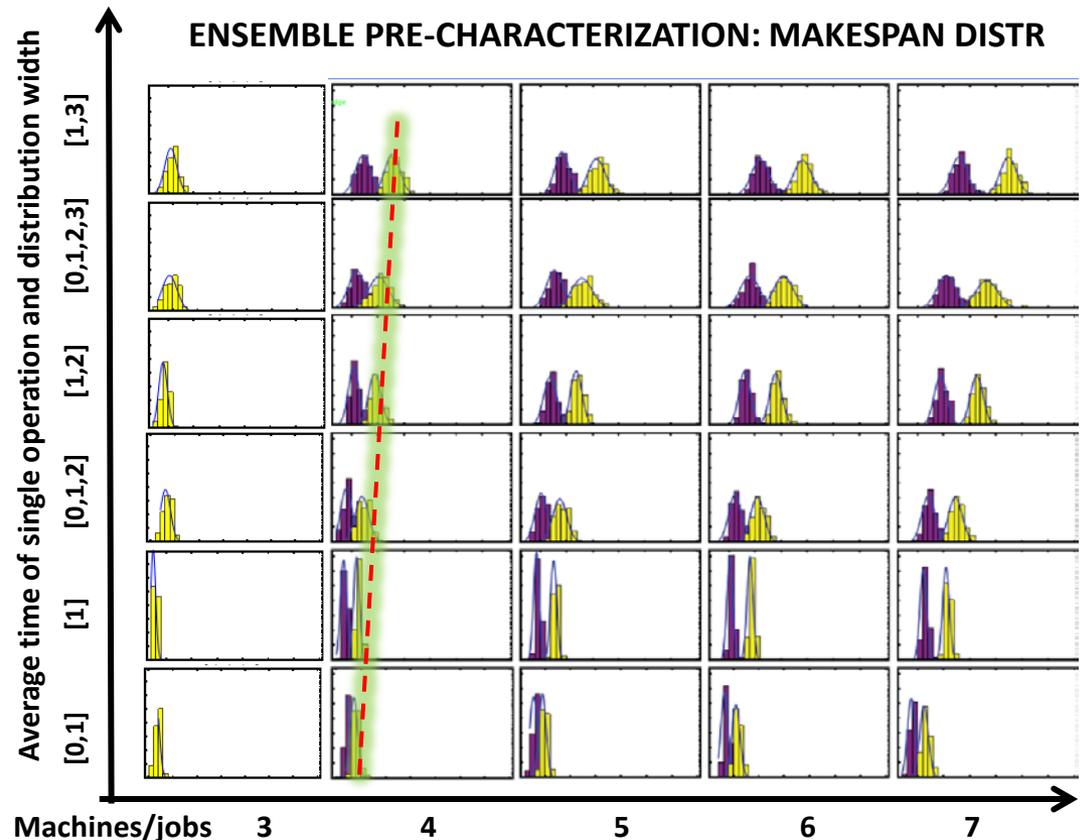
CLASSICAL INTRACTABILITY



(Rieffel et al. 2014)

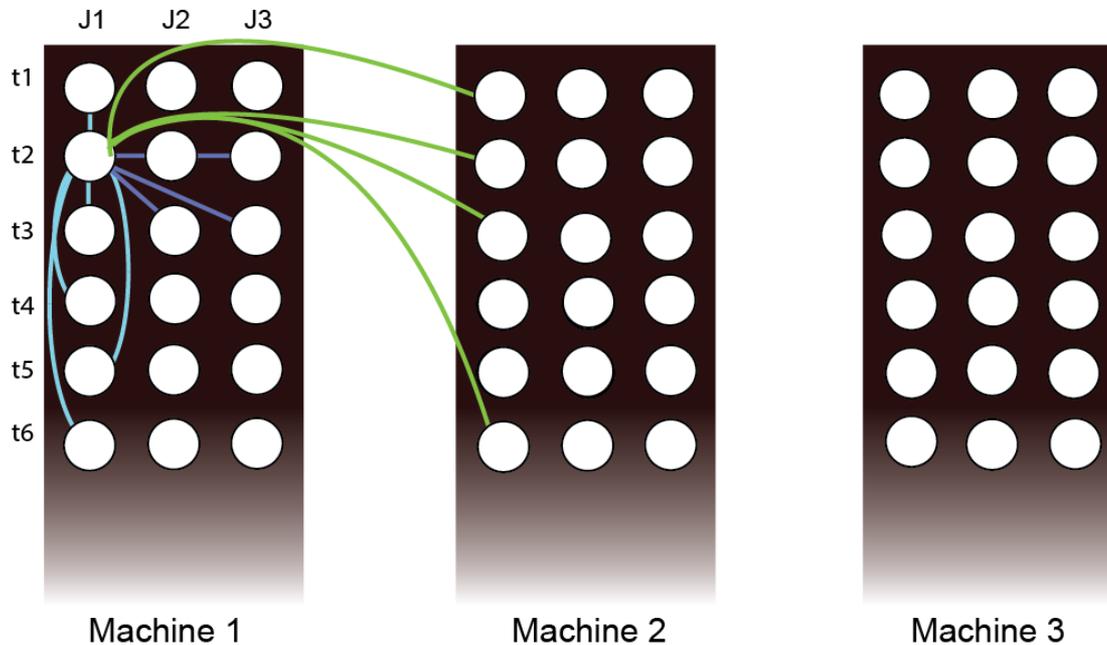
Size	Time	Best method
5x5 $\tau=[1,20]$	0.015 seconds	Scip
10x10 $\tau=[1,20]$	2.75 seconds	Gurobi
15x15 $\tau=[1,20]$	2430 seconds	Cplex (40%)

(Beck et al. 2014)

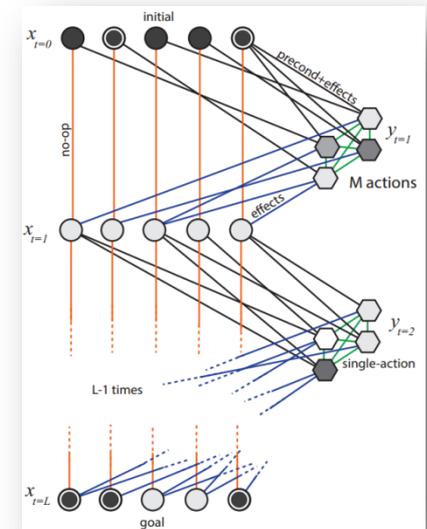


MAPPING INTO QUBO: THE TIME-SLICE APPROACH

$x_{nmt} = 1$ If the job n is executing on machine m at time t
 $x_{nmt} = 0$ otherwise



N M T bits required



(Rieffel et al. 2014)

$$\sum_{n,m} \left(\sum_t x_{mnt} - 1 \right)^2$$

$$\sum_{m,n} \left(\sum_{\bar{n} \neq n, \tau} x_{mnt} x_{m\bar{n}}(t+\tau) \right)$$

$$\sum_{(m,n,t), (\bar{m}, \bar{n}, \bar{t}) \in R_m} x_{mnt} x_{\bar{m}\bar{n}\bar{t}}$$

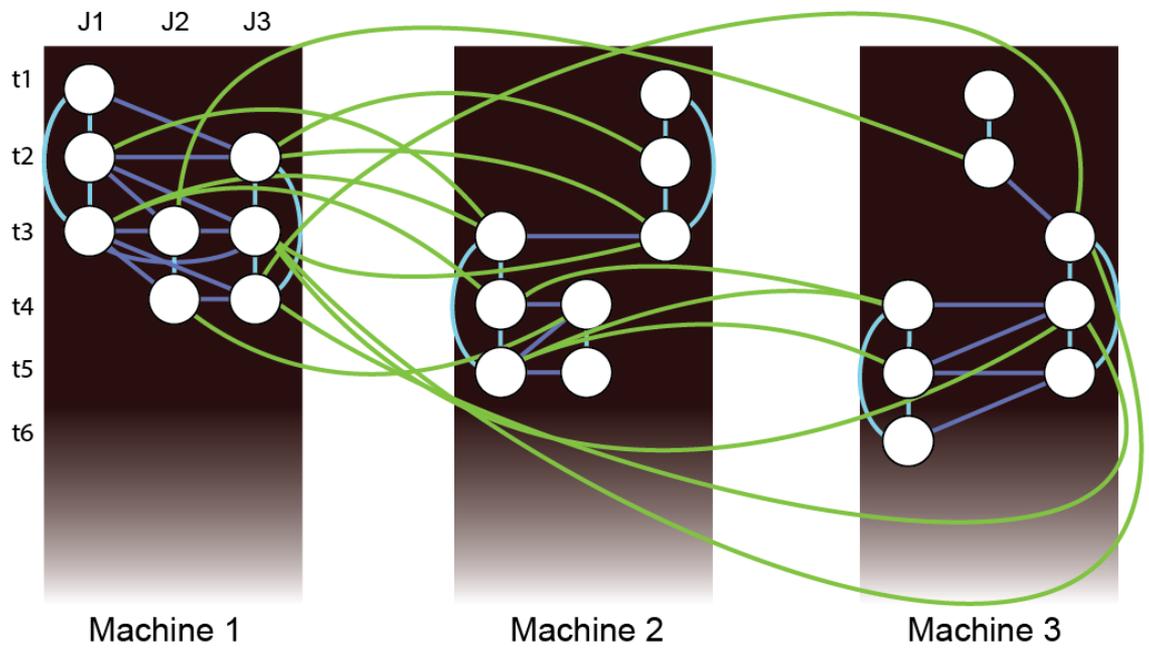
Note: scheduling problems naturally quadratic, this is not always the case.

(Venturelli et al. in prep.)

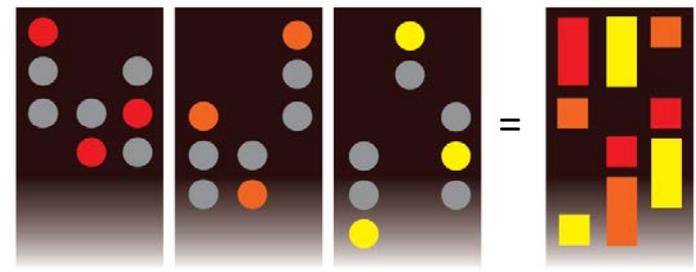
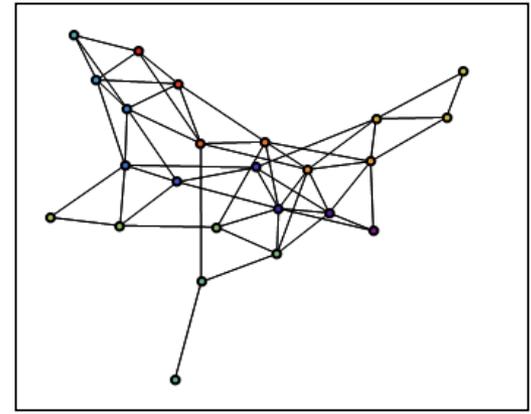
MAPPING INTO QUBO: POLYNOMIAL PRE-PROCESSING

$x_{nmt} = 1$ If the job n is executing on machine m at time t
 $x_{nmt} = 0$ otherwise

N M T bits required
- $NM(M \ll \tau + 1)$



Polynomial pre-processing:
 Trivial bounds on heads/tails of jobs.



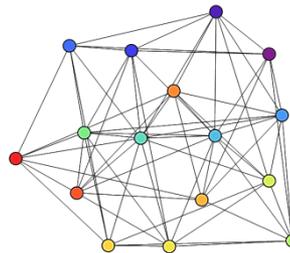
At the end the reading of the corresponding non-zero bits will immediately determine the schedule.

Note: ≈ 6000 logical qubits for intractable $N=15$ problems

COMPILING: GRAPH-MINOR EMBEDDING CHALLENGES

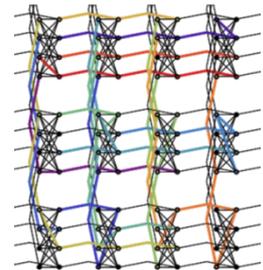
TOPOLOGICAL ASSIGNMENT OF CONNECTED COMPONENTS

Fully-connected substructures appears frequently in mappings and scale with the problem size.



$$\mathcal{E}(i) : \{1, \dots, n_L\} \rightarrow 2^{\{1, \dots, n_P\}}$$

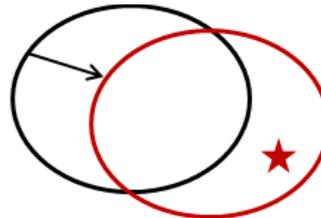
$$N_{HW}^* \simeq \frac{N_L^2}{k-2}$$



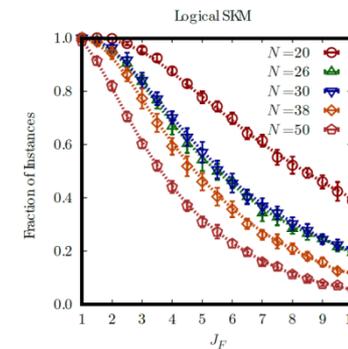
INCREASE IN PROBLEM MISPECIFICATION DUE TO CONTROL PRECISION ERRORS

Resolving multiple values of parameters (including embedding) will result into **loss of precision and problem misspecification**.

Ground state Ψ of Ideal Hamiltonian



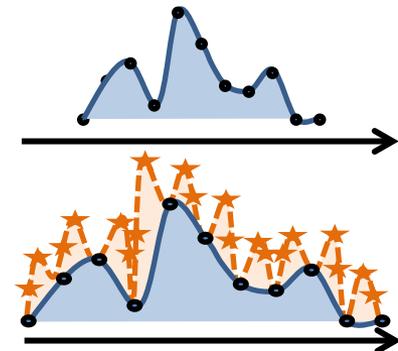
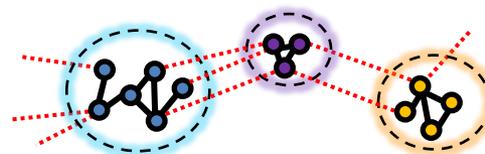
Ground state Ψ of ensemble of spoiled Hamiltonian



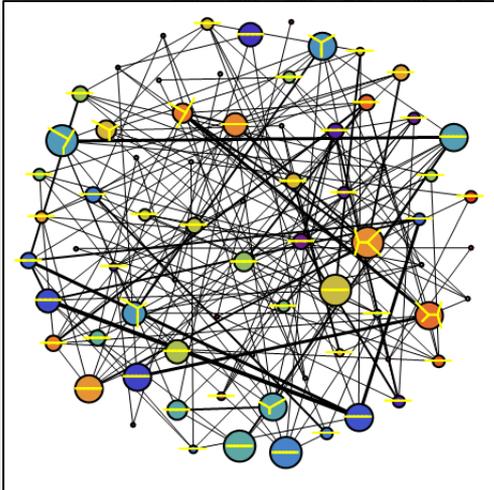
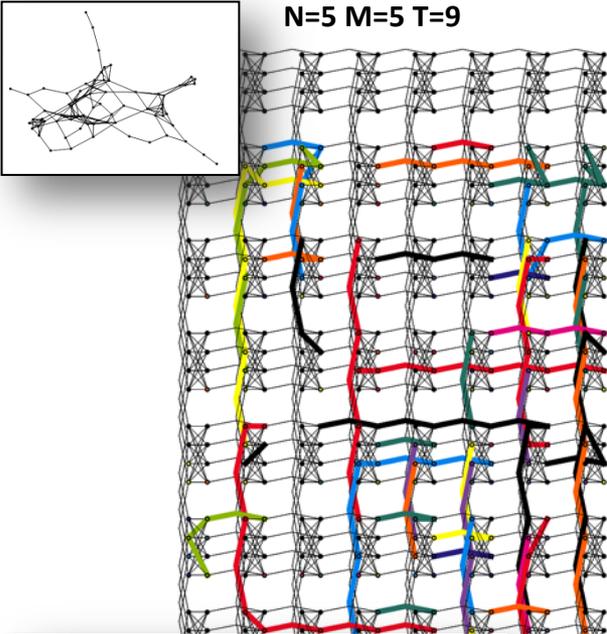
(Venturelli et al 2014)

(King et al 2015)

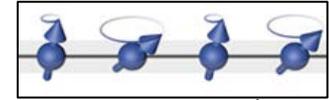
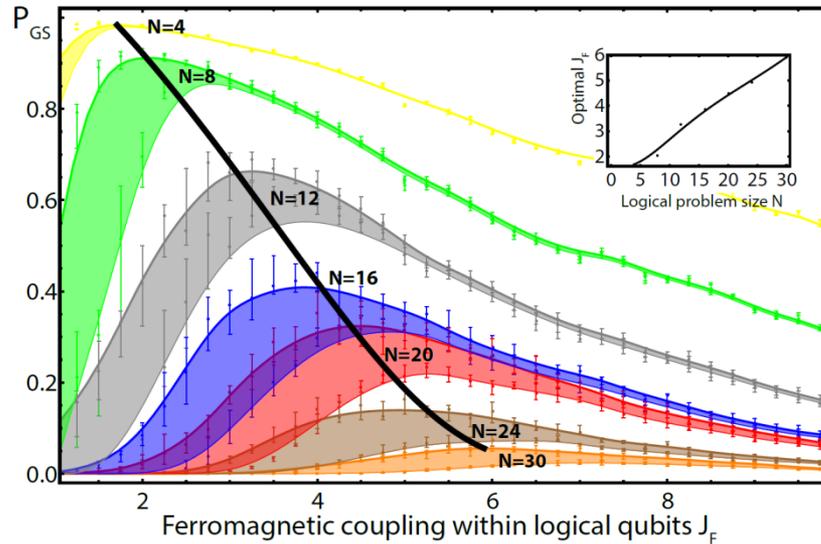
PHYSICAL CHANGE IN THE PROBLEM SPECTRUM RESULTS IN CHANGE IN PERFORMANCE



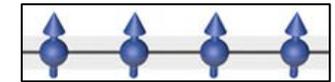
N=5 M=5 T=9



Early math considerations on parameter settings: the stronger the better



paramagnetic phase:
 $\Delta E \approx |B(t)-A(t)|$



ferromagnetic phase:
 $\Delta E \approx |A(t)/B(t)|^N$

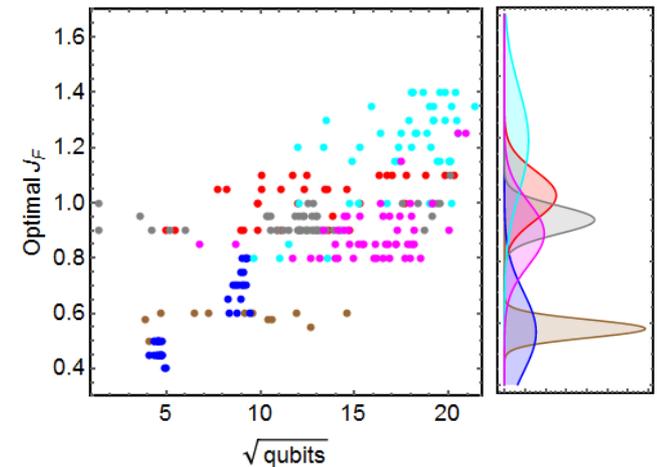
(Venturelli et al 2014)

Basic Method: search for the single optimal ferromagnetic embedding strength

CNF mapping		time-slice mapping							
J_{int}	J_{ext}	J_{int}	J_{ext}						
1.5	1.5	15873	108917	426 726	7737	35007	110 587	305 962	∞
1.4	1.5	13515	88025	537 766	6059	27 766	77 192	219 875	∞
1.3	1.5	12646	117 156	∞	4218	21 905	84 395	196 125	∞
1.2	1.5	16234	151 238	∞	1460	24 966	66 648	217 579	119 056
1.1	1.5	65 691	948 087	∞	5143	24 436	115 032	364 986	∞

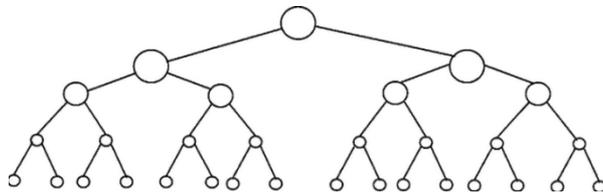
direct mapping										
J_{int}	J_{ext}									
1.5	1.5	57	108	273	655	2569	7907	12020	59 552	68 276
1.4	1.5	48	86	229	592	2136	4402	11 540	48 922	45 285
1.3	1.5	70	71	199	512	1698	4463	13 706	47 129	82 024
1.2	1.5	44	54	205	623	3503	13 153	20 613	78 794	162 471
1.1	1.5	62	68	210	818	4396	26 271	75 559	777 120	∞

(Rieffel et al. 2014)



RUN STRATEGIES: DECISION VS OPTIMIZATION, PERFORMANCE TUNING

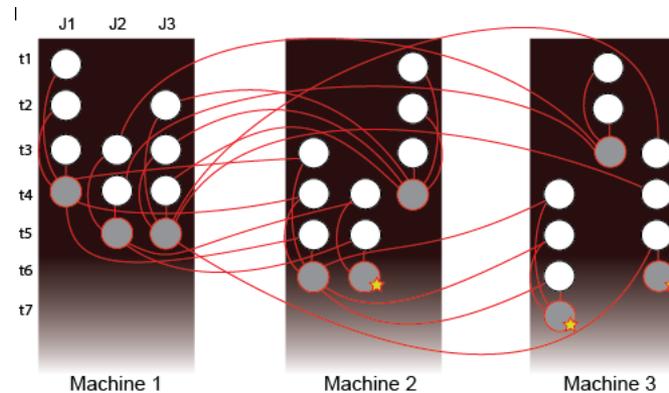
THE TIME-SLOT DISCRIMINATION METHOD FOR TIME-SLICED DECISION PROBLEMS (PRECISION LIMITED)



$\log_2(T)$ calls

Optimal solutions: $< \alpha$ energy

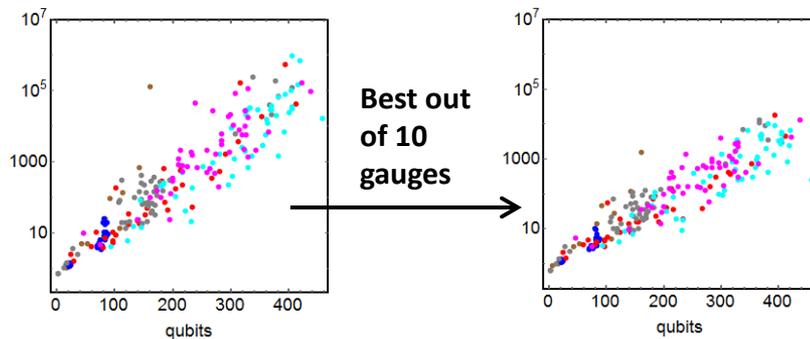
Suboptimal solutions: zero energy



Resolving k -time slots, given a maximum precision of ϵ for a problem of N jobs, with minimum logical penalty α :

$$\epsilon \approx \frac{\alpha}{N^k}$$

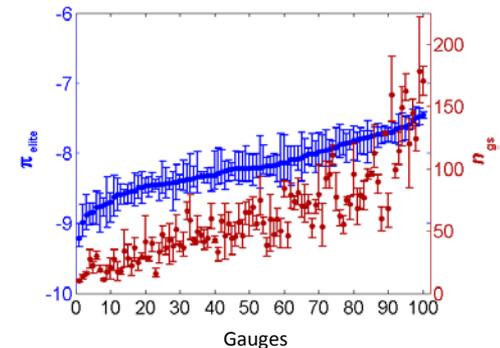
TUNING HARDWARE IMPERFECTIONS CAN HAVE A STRONG IMPACT.



Best out of 10 gauges

We can pre-select the best gauge by either Hamiltonian learning or with the use of “performance estimators”

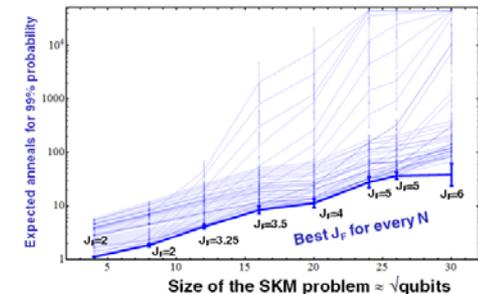
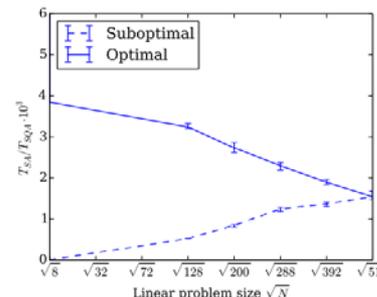
(Perdomo-Ortiz in prep.)

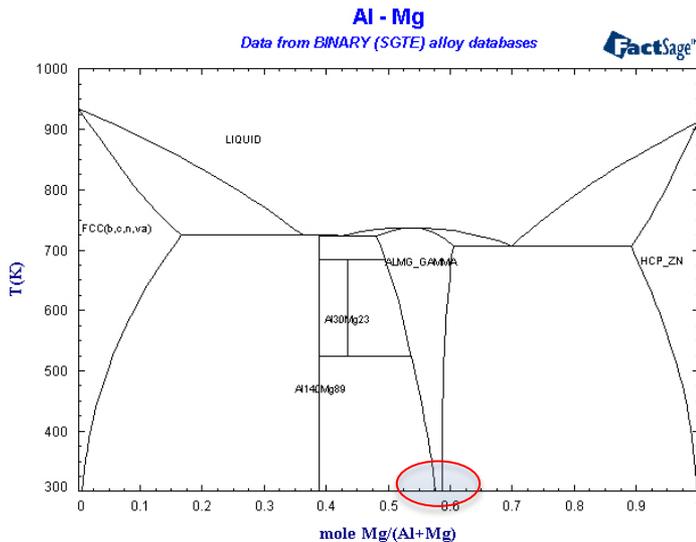


OPTIMIZING ALL RUNNING PARAMETERS (NOTABLY ANNEALING TIME): NECESSARY FOR BENCHMARKING AND FOR EVALUATING SPEEDUP.

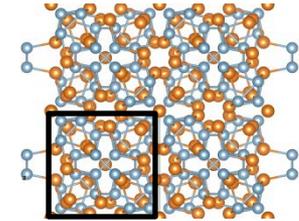
(Ronnow 2014)

(Venturelli et al 2014)

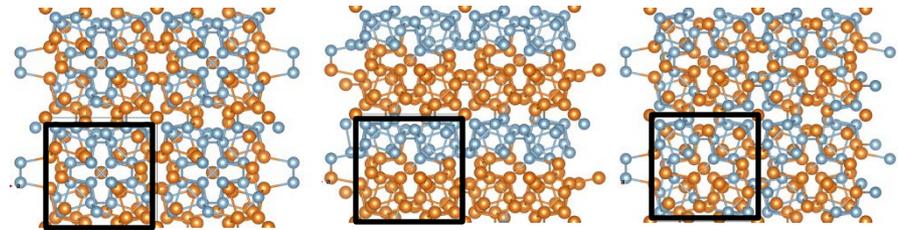




Toy example of binary alloy:
 $Mg_{17}Al_{12}$



2^{29} possible configurations!
Experimental annealing can get stuck on metastable



Objective: Finding new different stable structures

Cluster expansion approach: approximate the Hamiltonian using information from subset of the unit cells that are solved with Density Functional Theory. Minimize the resulting classical energy functional with Monte Carlo methods.

(Sanchez et al. 1984)

$$E(\sigma) = V_0 + \sum_{i,j} V_{ij} \sigma_i \sigma_j + \sum_{i,j,k} V_{ijk} \sigma_i \sigma_j \sigma_k + \dots$$

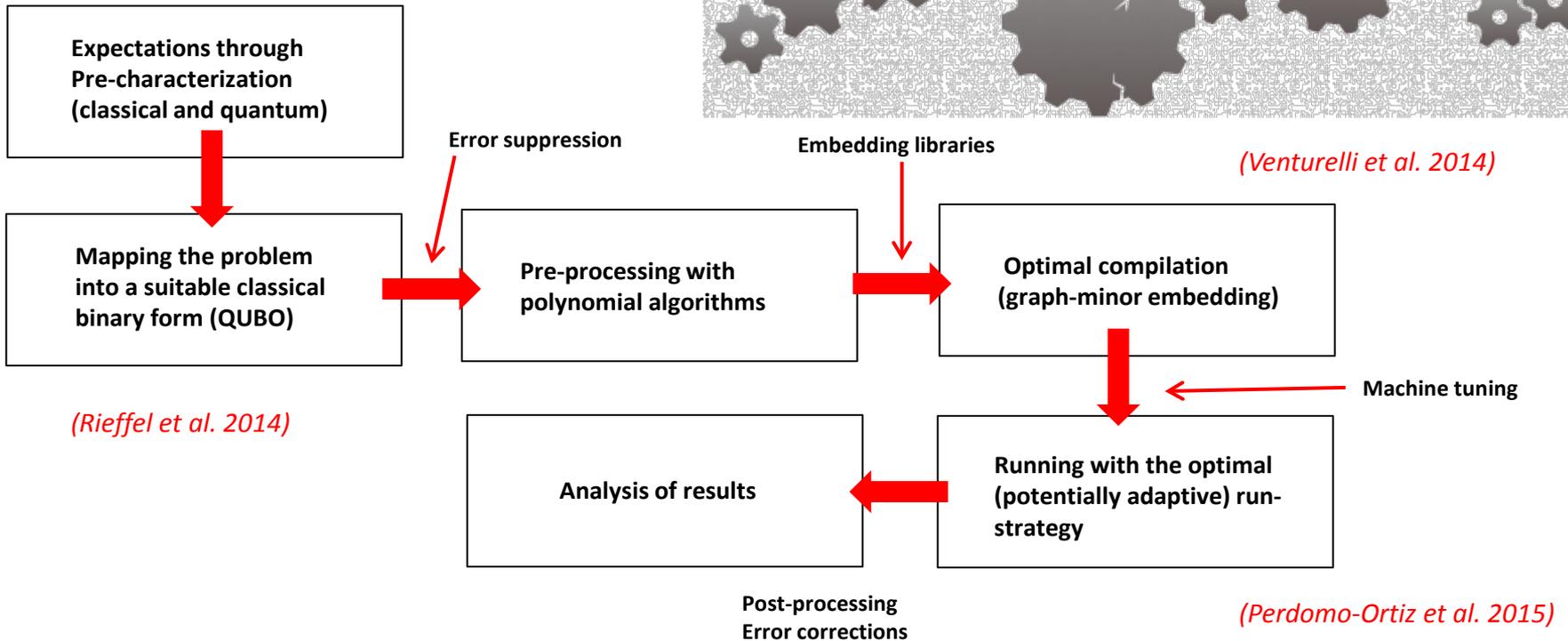
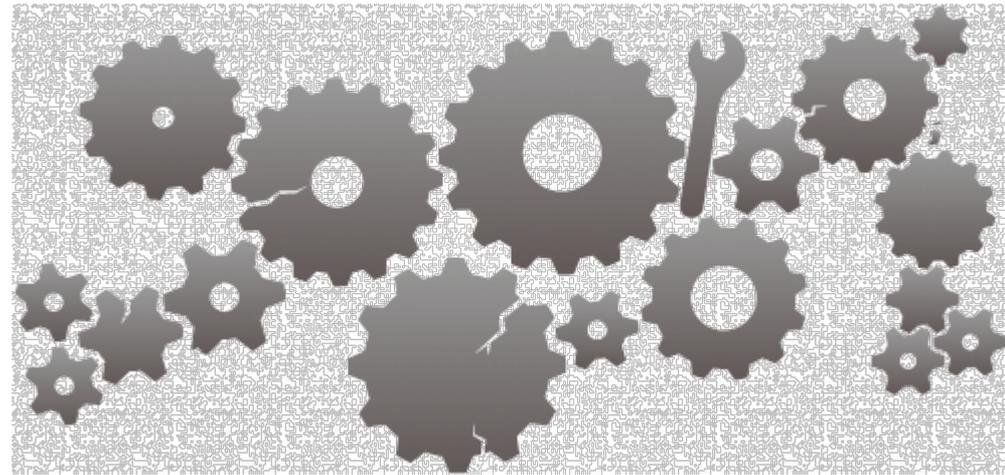
- Typical cluster expansion contains around $\approx 15-20$ V parameters and **3-local or maximum 4-local terms**.
- Limit of solvability for DFT ≈ 50 atoms. But long range interactions generate large non-local hamiltonians non easily minimizable by MonteCarlo. **Quantum Annealing can help?**

$$q_i q_j \rightarrow \tilde{q}_n$$

$$H_{\wedge}(q_i, q_j, \tilde{q}_n) = \delta(3\tilde{q}_n + q_i q_j - 2q_i \tilde{q}_n - 2q_j \tilde{q}_n).$$

In general, we add 1 more qubit every 3-body term

(Babbush et al. 2012)



SUMMARY

- ❑ We need to **identify problems** that are:

SMALL (Mapping+Embedding Bottleneck can allow programming in near-future machine)

HARD (Ideally on the verge of a phase transition of solvability)

RESILIENT TO MISSPECIFICATION (The Analog control errors are a precision bottleneck)

Where classical algorithms have hard time finding even an approximate solution

- ❑ To **design an annealing architecture** supporting programming of discrete optimization problems we need to take advantage of the programming strategies that are appearing in the problems of interest, that will be naturally fitting some **connectivity** of the hardware graph and some **precision requirements** of the problem.

- ❑ To **prepare a machine** to solve a class of problems we want to:

CALIBRATE it properly for the classes of instances

PRE-CHARACTERIZE the classes of problems to gain expectations useful for the embedding and running

Prepare **EMBEDDING LIBRARIES** that could fit the class with pre-determined parameters.