Applications of Annealing to Production Planning Optimization

Zac Brennan, Joseph David, Kemal Aziez Rachmansyah, Rikuto Shigemi

Industrial Mentor: Dr. Ryoji Miyazaki
Academic Mentor: Dr. Yuki Irie

Presentation Outline

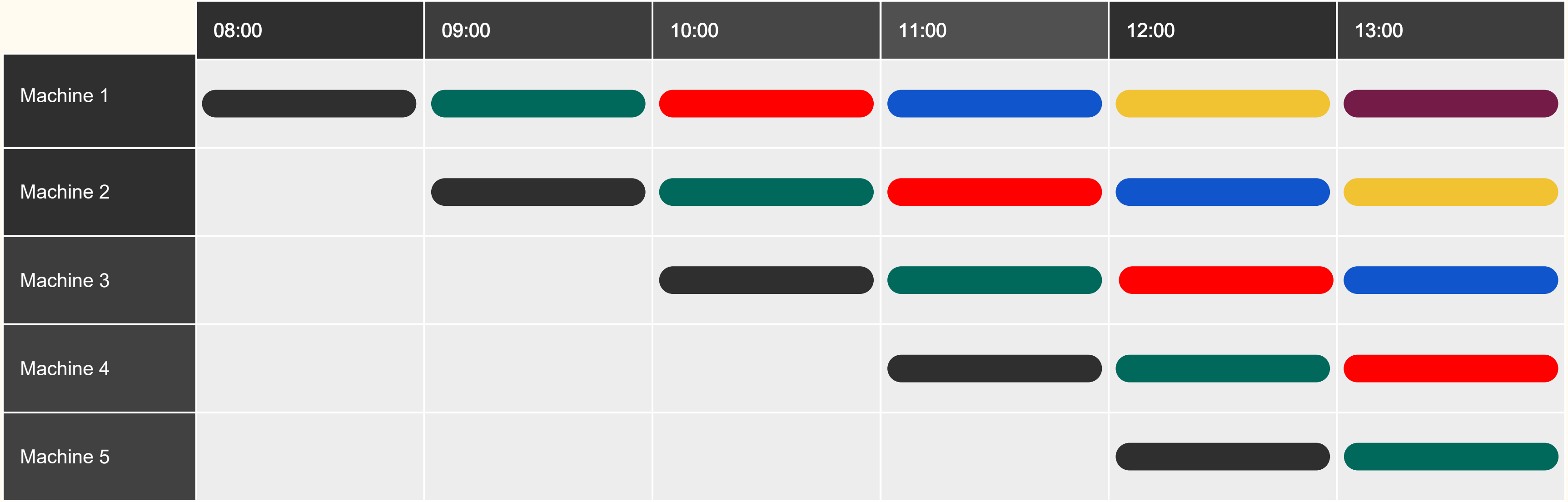
1. Introduction
    a. The scheduling problem
    b. Our goals
2. Simulated Annealing & Job Shop Scheduling Formulation
3. Variations of Simulated Annealing
    a. Vector Annealing, External Constraints, Variable Pruning
4. Results
5. Observations on Quantum Annealing

Part 1

Introduction

Increasing complexity from manufacturing

- Mass manufacturing on assembly lines is low complexity scheduling
- Jobs have identical production requirements

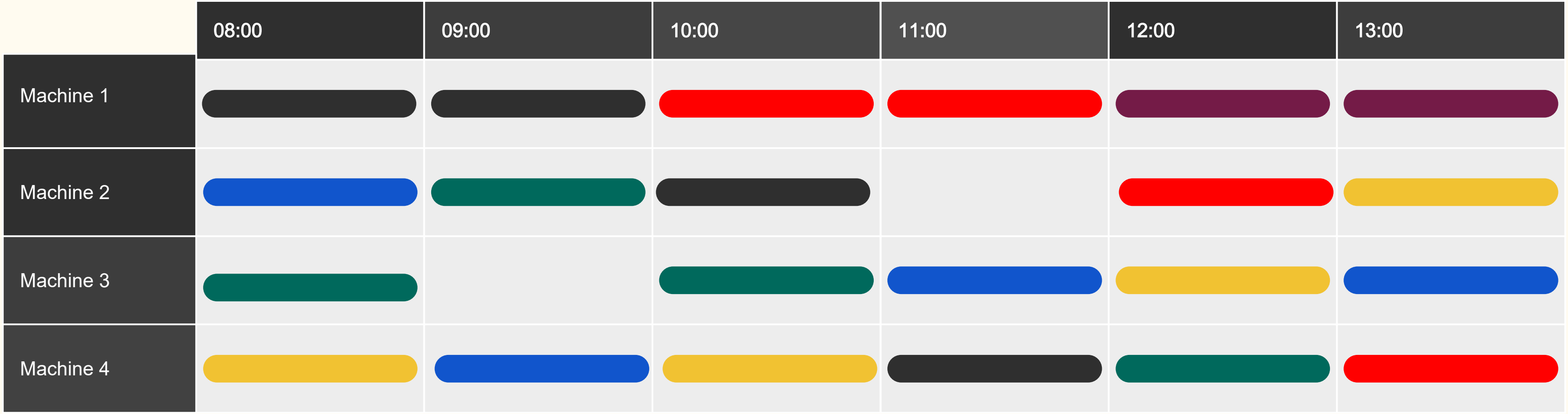


	08:00	09:00	10:00	11:00	12:00	13:00
Machine 1	⬛	🟩	🟥	🟦	🟨	🟪
Machine 2		⬛	🟩	🟥	🟦	🟨
Machine 3			⬛	🟩	🟥	🟦
Machine 4				⬛	🟩	🟥
Machine 5					⬛	🟩

Note: all these jobs are to create the same product

Increasing complexity from manufacturing

- High-mix low-volume production increases variety of manufacturing requirements
- Advancements in general-purpose manufacturing such as industrial 3D Printing, laser cutting, and CNC leads to more multi-purpose manufacturing machines

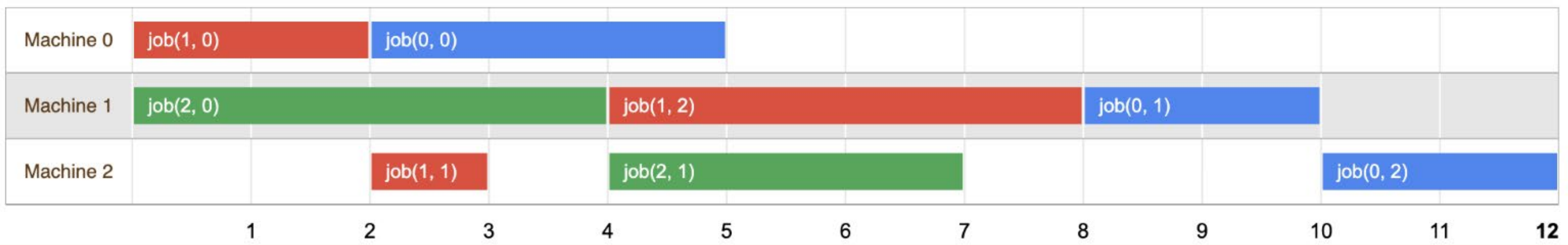


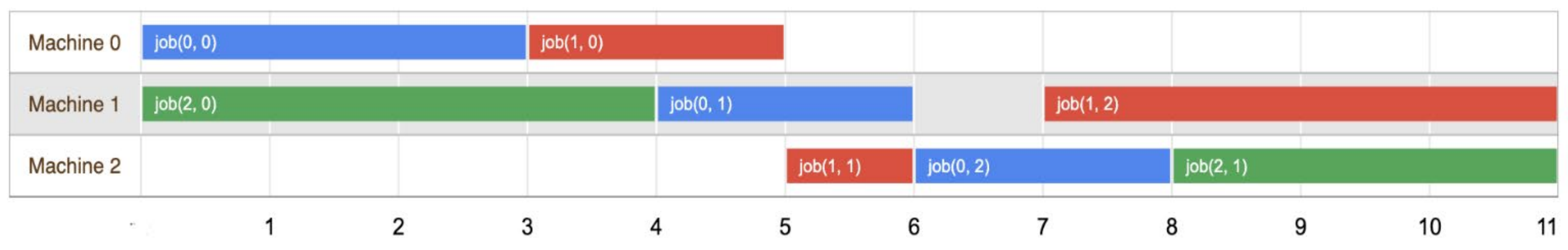
	08:00	09:00	10:00	11:00	12:00	13:00
Machine 1	⬛	⬛	🟥	🟥	🟪	🟪
Machine 2	🟦	🟩	⬛		🟥	🟨
Machine 3	🟩		🟩	🟦	🟨	🟦
Machine 4	🟨	🟦	🟨	⬛	🟩	🟥

- Consolidation of machines and non-linear manufacturing dramatically increase complexity

Production Planning - The Job Shop Problem

- Scheduling problem where multiple jobs are processed on multiple machines with certain operational precedence rules





Job 0: Operations 0, 1, and 2

Job 1: Operations 0, 1, and 2

Job 2: Operations 0 and 1

- Minimize *makespan*, the total schedule length

https://developers.google.com/optimization/scheduling/job_shop under CCA 4.0 License

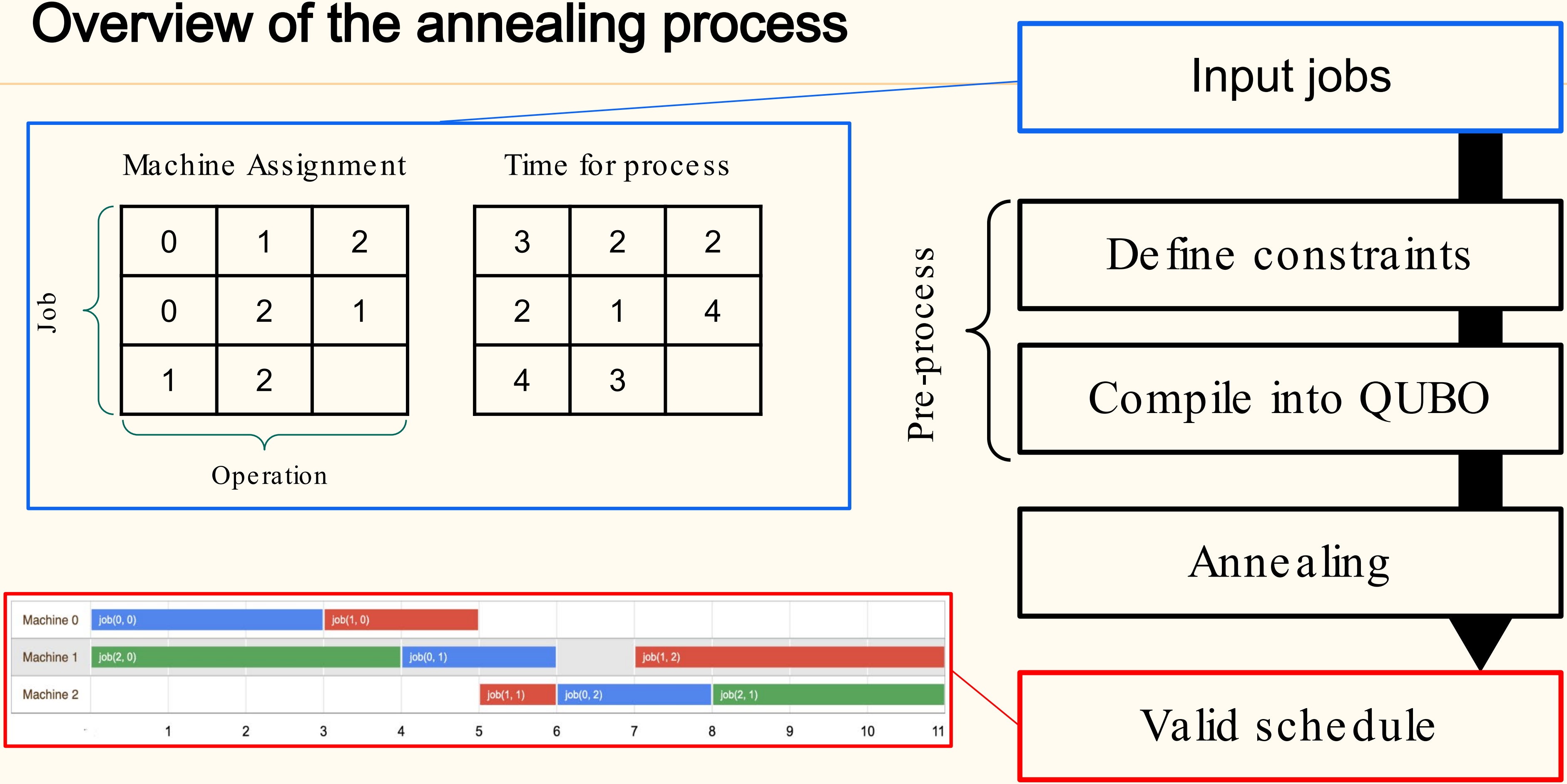
Our Project - Background

- *Simulated Annealing* is a promising method of solving these types of combinatorial optimization problems
    - Particularly for a “good enough” solution, where near-optimal solutions are acceptable

Computers

- NEC has developed their own annealing methods for their supercomputer SX-Aurora TSUBASA
- D-Wave Quantum Annealing has recently been made commercially available

Overview of the annealing process



Input jobs

Machine Assignment

0	1	2
0	2	1
1	2	

Time for process

3	2	2
2	1	4
4	3	

Our Project - Goals

- Understand the size and scope of problems currently feasible on SX-Aurora and D-Wave Systems, and whether it can handle suitably complex problems

- Compare the performance of SX-Aurora TSUBASA to D-Wave and classical simulated annealing in creating production schedules

- Consider modifications of standard annealing algorithms to improve efficiency on standard combinatorial problems such as Job Shop Scheduling

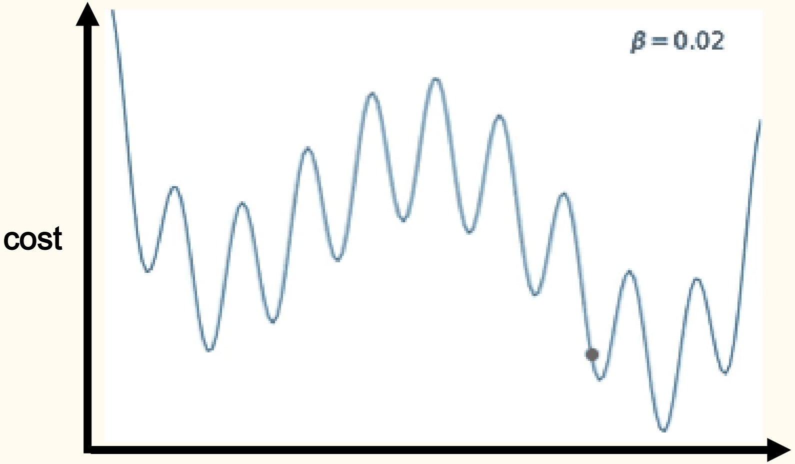
Part 2

Simulated Annealing & JSSP Formulation

Simulated Annealing (SA)

S. Kirkpatrick, C. D. Gelatt Jr., M. P. Vecchi (1983)

- Optimization Method, inspired by Annealing process in Metallurgy
    - Heat the material ➔ Recrystallize ➔ Slowly cool it down
- Have a chance to go up based on cost (energy) difference and temperature.
- SA take samples from the whole space, so it could escape local minimums.



- If $E_1 \geq E_2$,

$$P(E_1 \rightarrow E_2) = 1$$

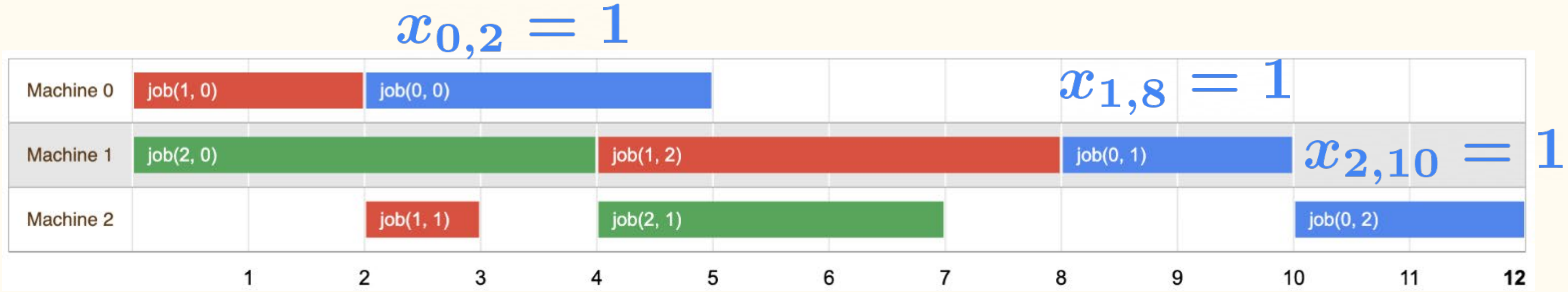
- If $E_1 < E_2$,

$$P(E_1 \rightarrow E_2) = \exp\left(-\frac{E_2 - E_1}{Temperature}\right)$$

https://www.smapip.is.tohoku.ac.jp/

Job Shop Scheduling Problem Formulation

- Define set of binary variables $(x_{i,t})_{i=0,t=0}^{N,T}$, where N is the **total # of operations** and T is **timespan**.



$x_{0,2} = 1$

$x_{1,8} = 1$

$x_{2,10} = 1$

$$x_{i,t} = \begin{cases} 1, & \text{if operation } i \text{ begins at time } t \\ 0, & \text{otherwise} \end{cases}$$

https://developers.google.com/optimization/scheduling/job_shop under CCA 4.0 License

Problem Formulation: Constraints

- Include as cost rather than restrict search space.
- Example: One-hot constraint
  - As a mathematical expression in $(x_{i,t})$

$$\sum_{i=0}^{N}\left(1-\sum_{t=0}^{T}x_{i,t}\right)^{2}$$

More Constraints : Order, Overlap, Deadline

Constraints

$$H = C_{obj}H_{obj} + \underline{C_1H_1 + \dots + C_NH_N}$$

Test Cases: Number of Variables, Constraints

Problem	6x6	7x7	10x5	10x10	6x6	7x7	10x5	10x10
# of Var	2340	3675	3750	11500	3060	6125	5000	14500

Baseline

Complex

	One Hot	Overlap	Order	Trans.	Deadline
Baseline	X	X	X		
Complex	X	X	X	X	X

Time To Solution - How to measure performance

- Balance runtime with solution quality
  - TTS is expected amount of time to return a near optimal result
  - t = the length of a single annealing iteration
  - R = the number of annealing iterations to get a good solution with probability $p_d = .99$
  - $p_s(t)$ = Probability of a valid schedule with single run of length t

$$(1 - p_s(t))^R = 1 - p_d \Rightarrow R = \frac{\ln(1 - p_d)}{\ln(1 - p_s(t))}$$

$$TTS(t) = t \cdot R$$

Part 3

Variations of Simulated Annealing

Vector Annealing

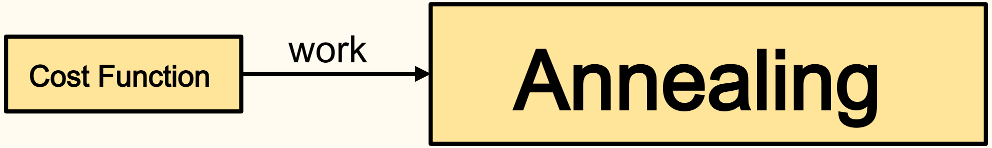
- Simulated Annealing done on Vector Machines such as SX-Aurora Tsubasa made by NEC.

- SX-Aurora Tsubasa can process one-dimensional array of data (vectors) effectively and efficiently.
  - Parallel operations

- Vector Annealing inspired our classical simulated annealing modifications, such as…

https://www.nec.com/en/global/solutions/hpc/sx/vector_engine.html

SA Modification



1. <u>External Constraint</u>

   [M. Kumagai, K. Komatsu, F. Takano, T. Araki, M. Sato, H. Kobaya
   (2021)]

   ○ Move a constraint outside the cost function.

   ○ After every sweep, “bad configurations” are swapped

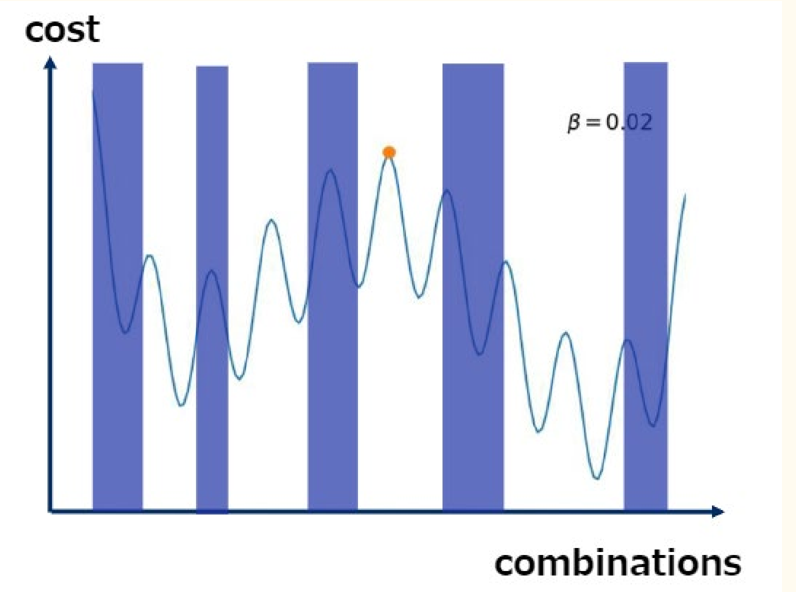
   ○ Limit the search space according to the constraints.

2. <u>Variable Pruning</u>

   [D. Venturelli, D.J.J. Marchand, G. Rojo (2015)]

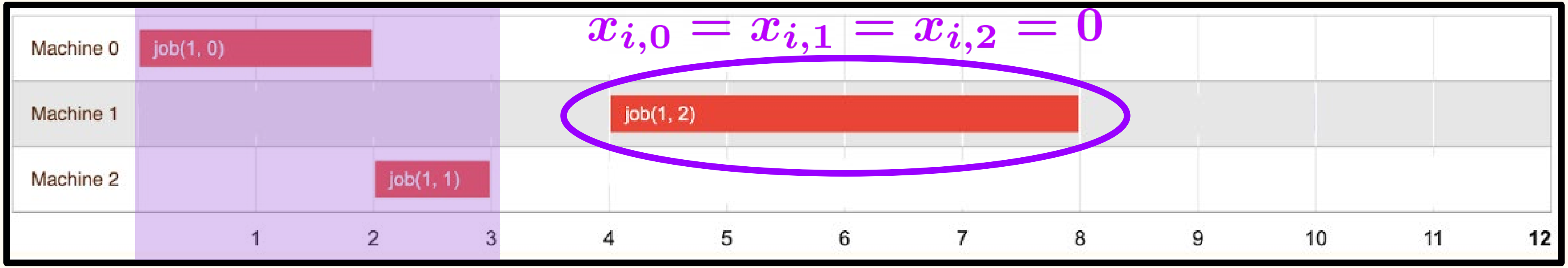
   ○ Some variables are guaranteed to be zero.

   ○ We delete those variables from the cost function.

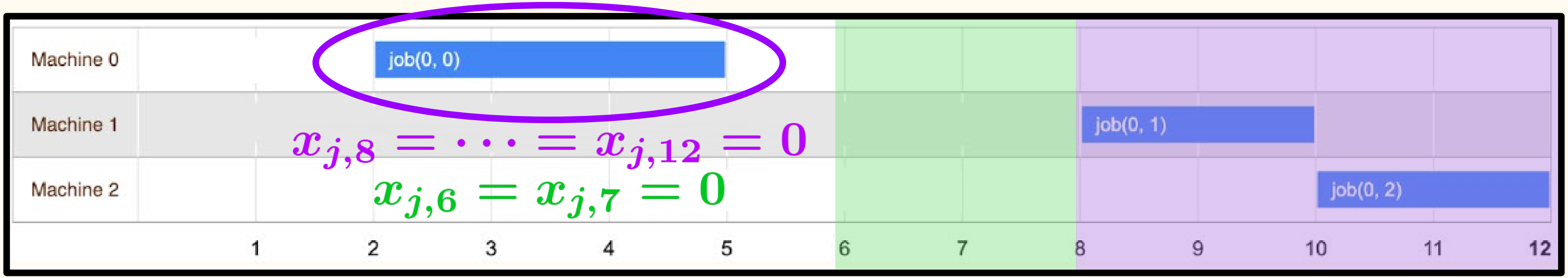


The cost function is simpler ➔ Less coefficients to balance!
The search space is smaller ➔ Faster and more accurate result!

Variable Pruning



$$x_{i,0} = x_{i,1} = x_{i,2} = 0$$



$$x_{j,8} = \cdots = x_{j,12} = 0$$

$$x_{j,6} = x_{j,7} = 0$$

Number of Pruned Variables: $N_{ops} \sum_{i=0}^{N_{jobs} \times N_{ops}} p_i$

Our Tools

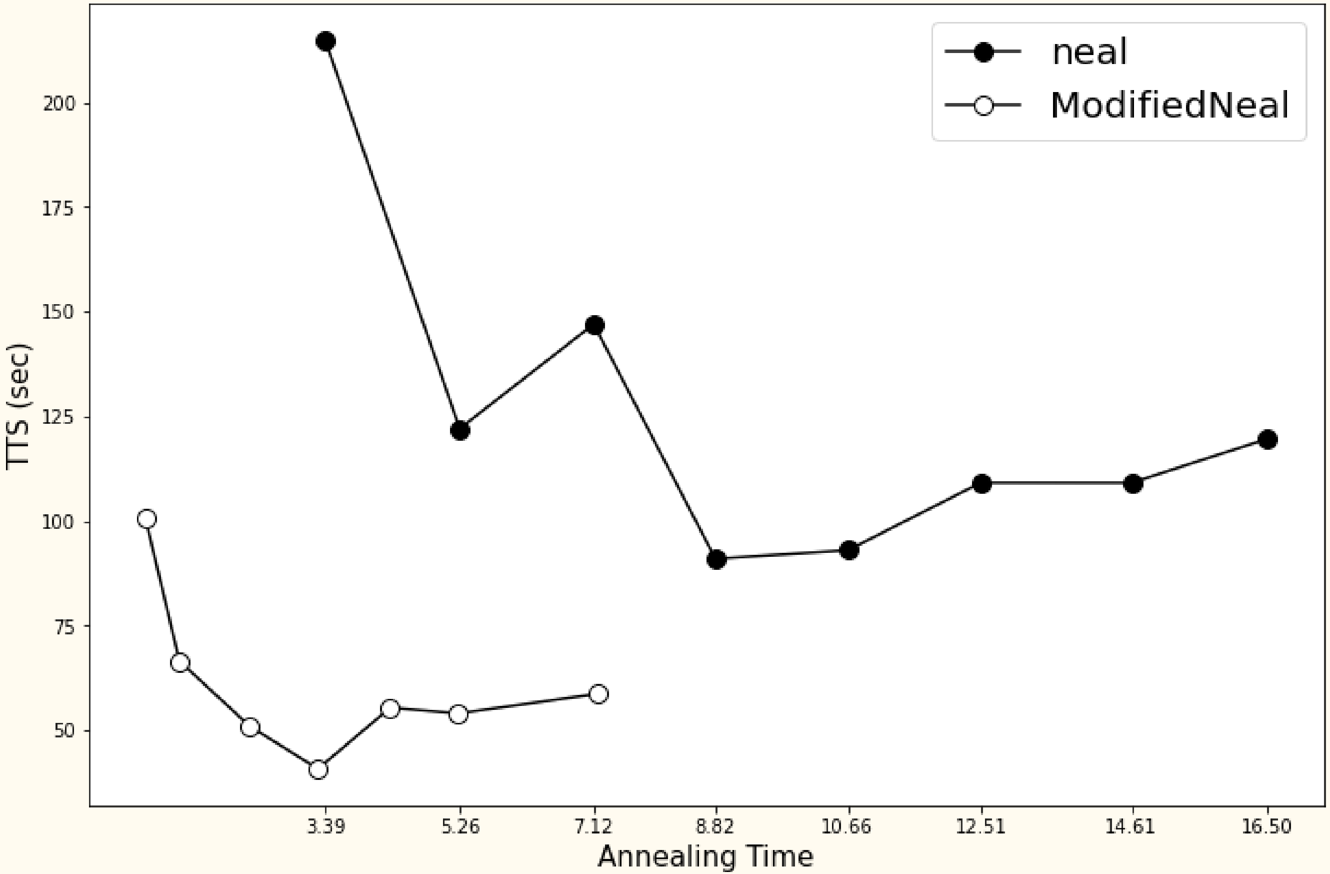
	Simulated Annealing (neal)	**Simulated Annealing with External Constraints (Modified neal / nealmod)**	**Vector Annealing (VA)**	**Quantum Annealing (QA)**
Machine	Personal Computer, Vector Machine	Personal Computer, Vector Machine	Vector Machine	Quantum Annealer
Python Package	dwave-neal	Modified dwave-neal	VectorAnnealing (exclusive to SX-Aurora)	dwave-system

We can use Variable Pruning in all of these tools

Part 4

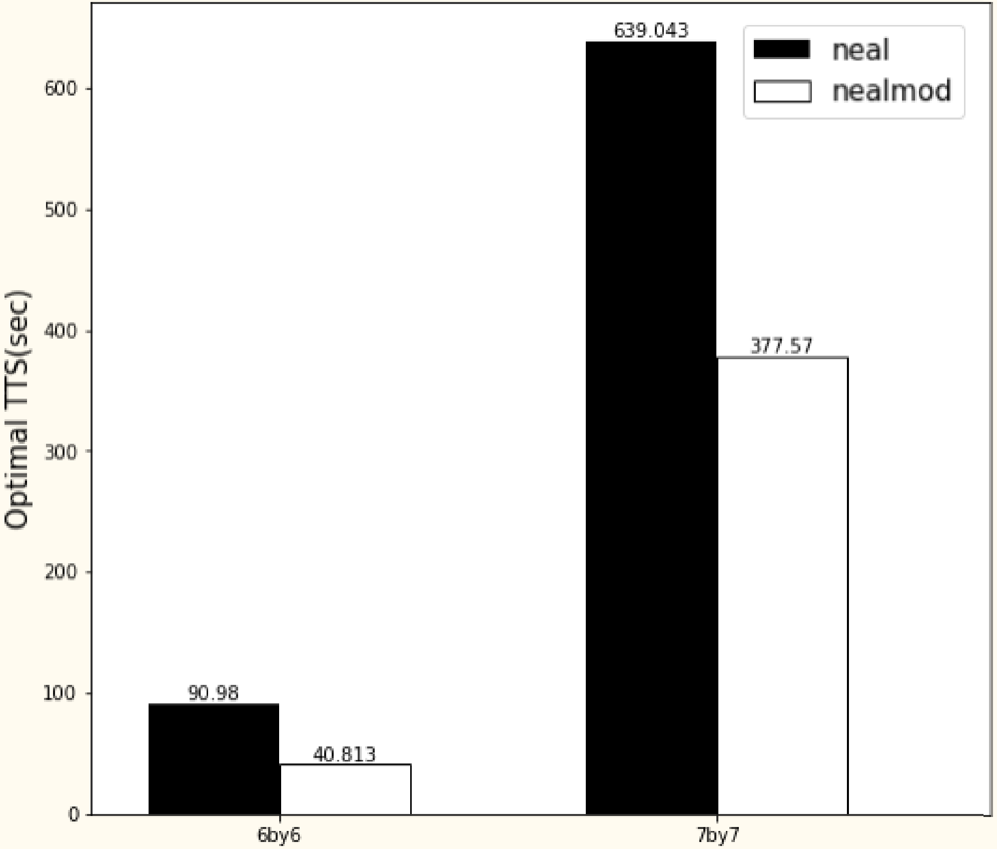
Simulated Annealing Results

External Constraints: neal vs nealmod



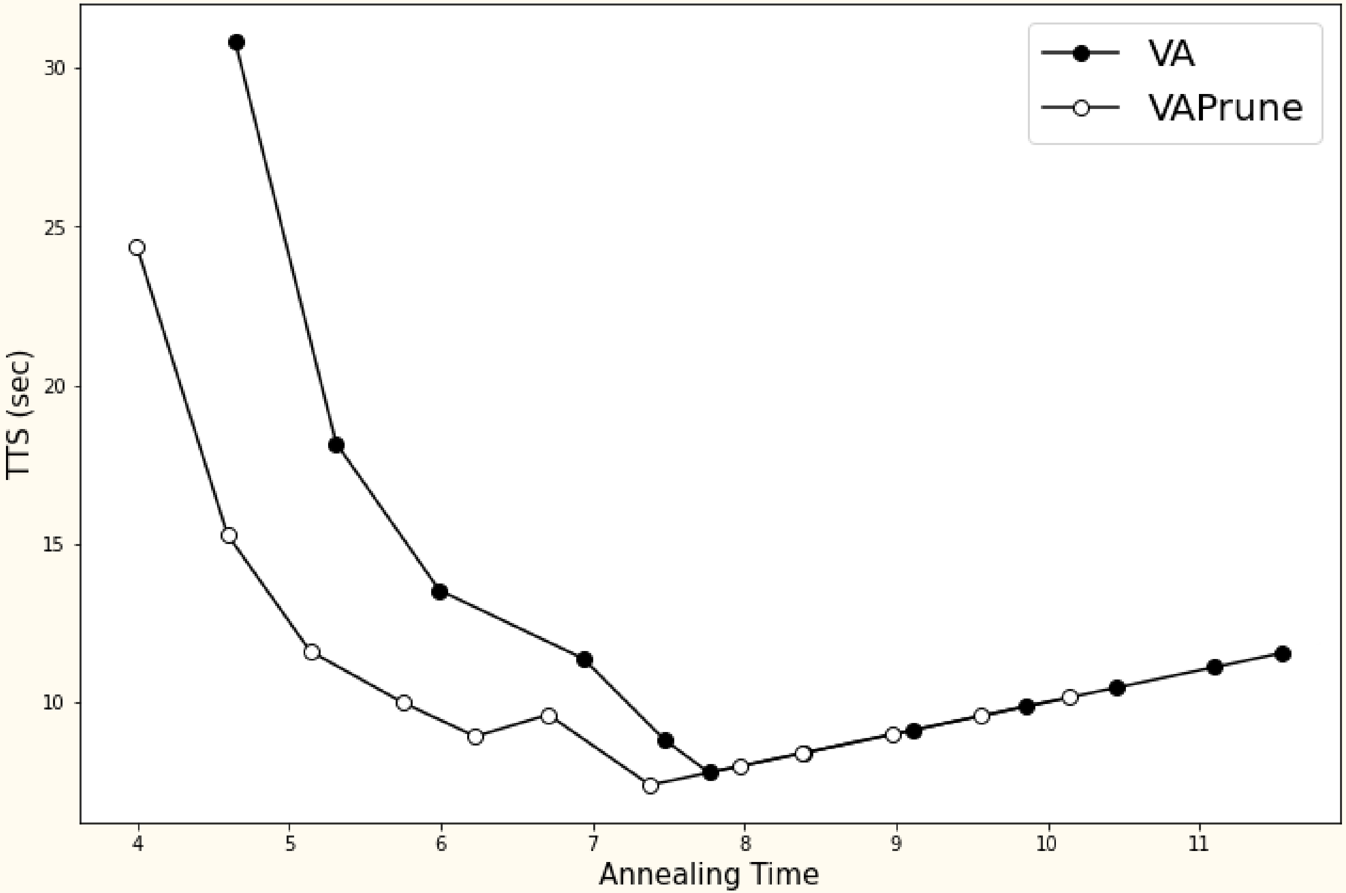
6x6
Standard
Formulation

External Constraints: neal vs nealmod



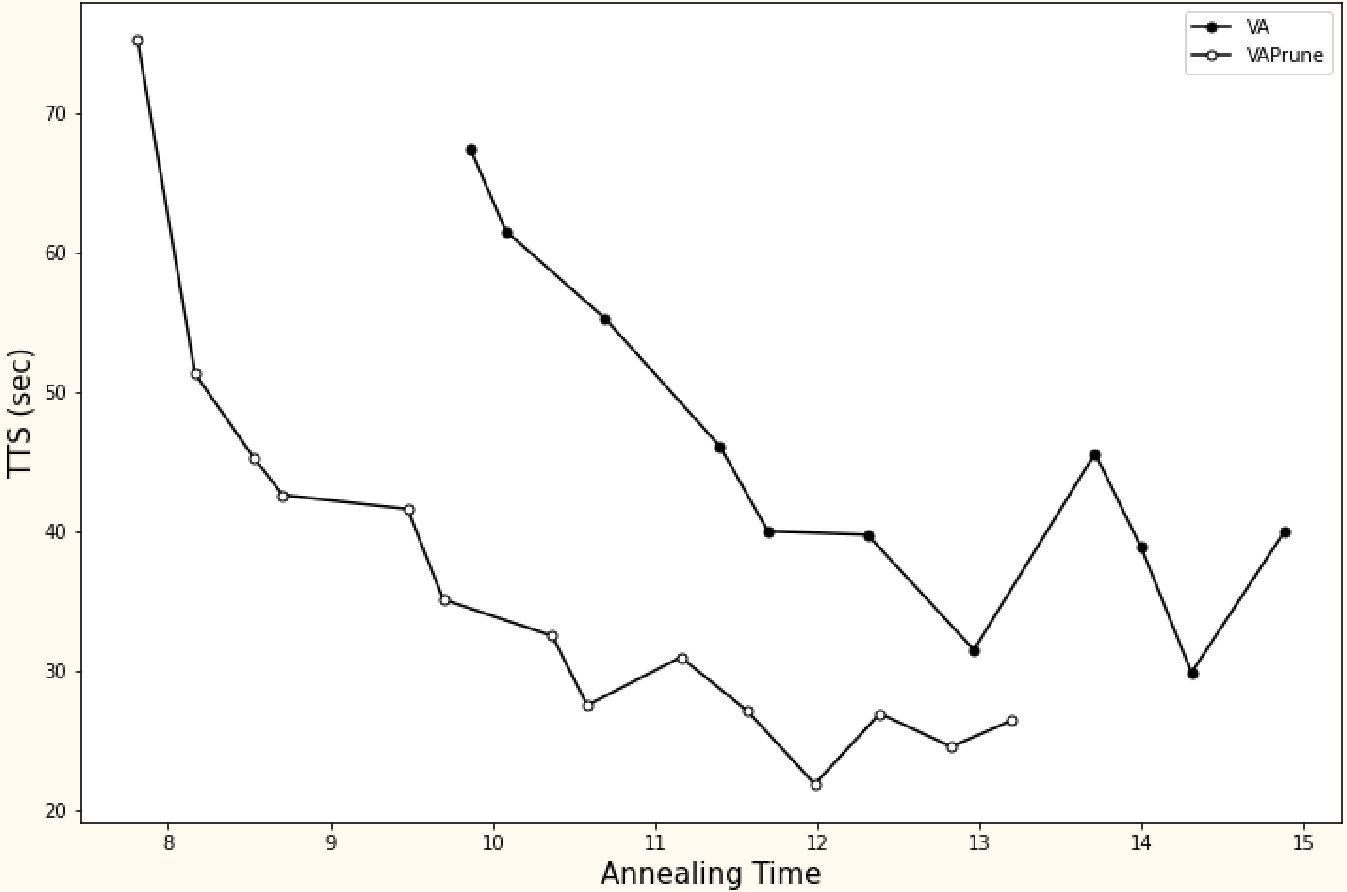
Standard Formulation

Time to Solution Example: 10x10 Vector Annealing



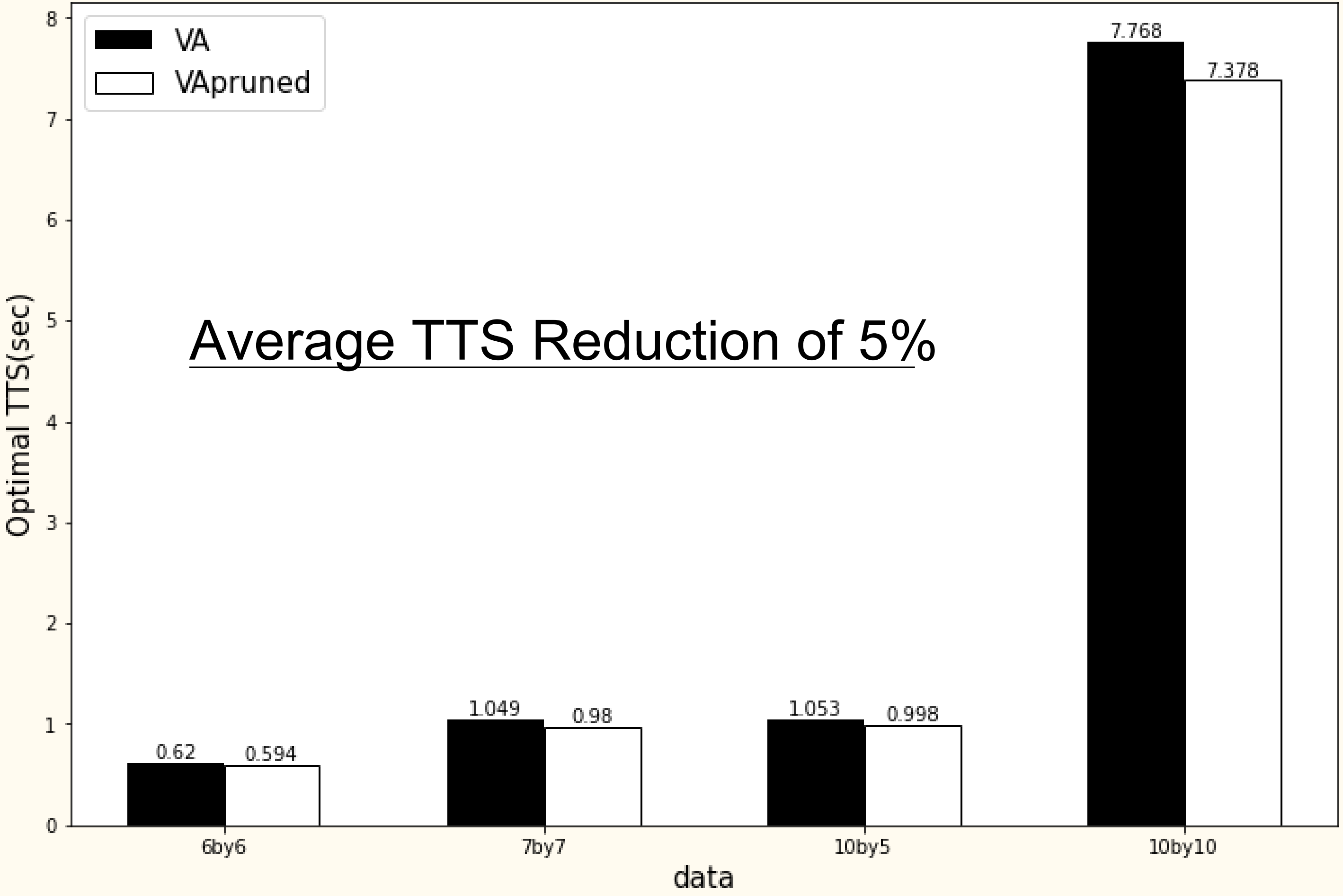
Standard Formulation

Time to Solution Example: 10x10 Vector Annealing



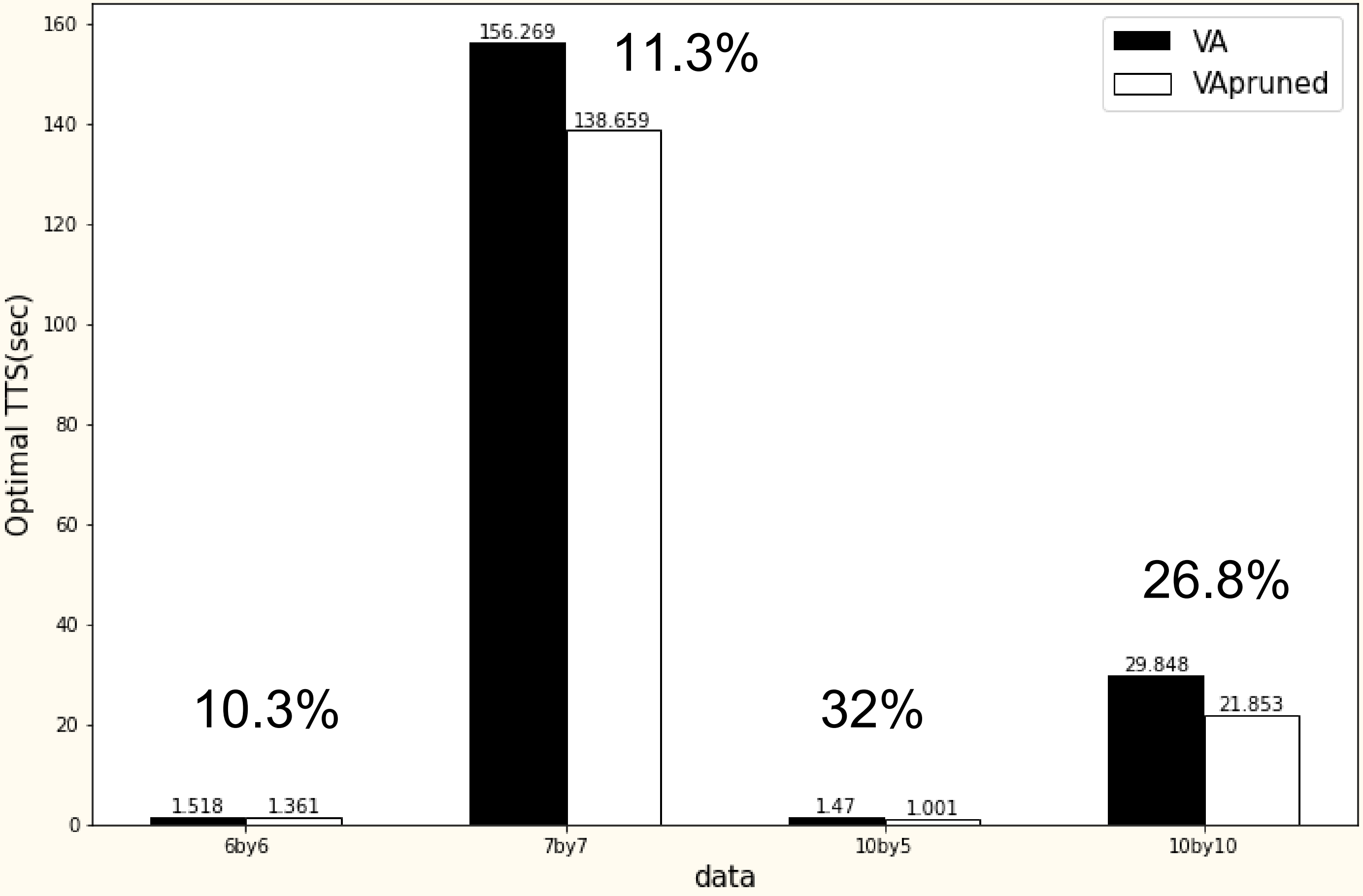
Complex Formulation

Standard Formulation



Average TTS Reduction of 5%

Complex Formulation



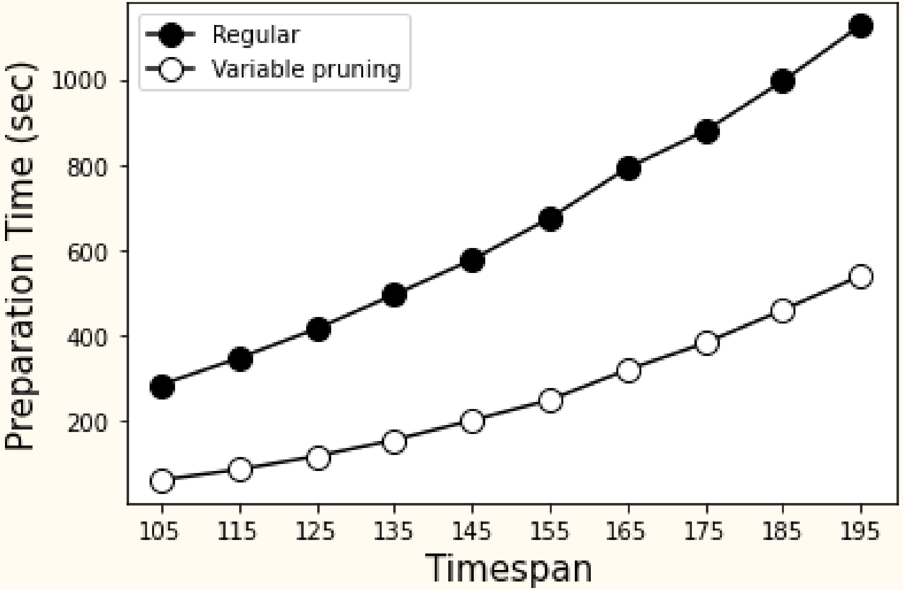
10.3%

11.3%

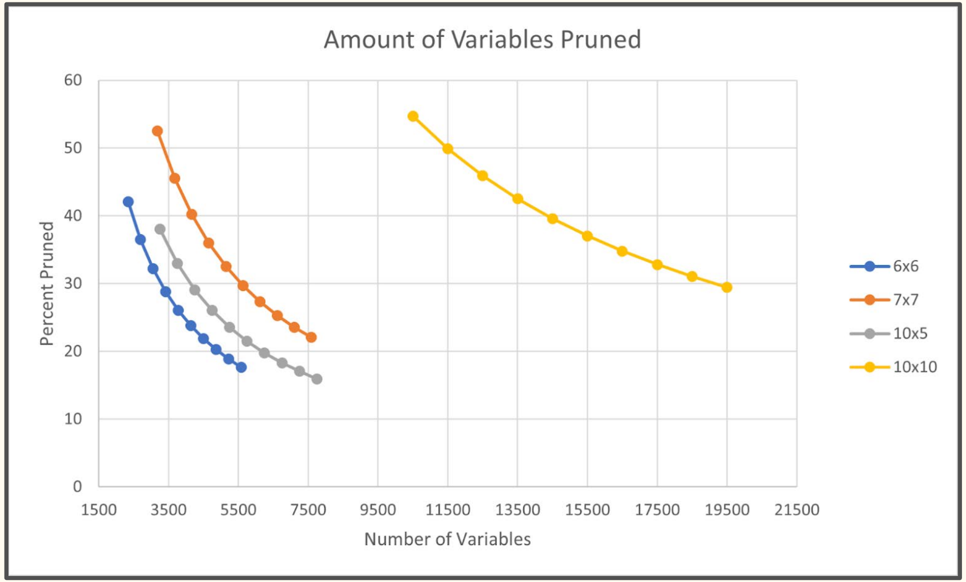
32%

26.8%

Preparation Time and Results



10x10



Simulated Annealing Summary

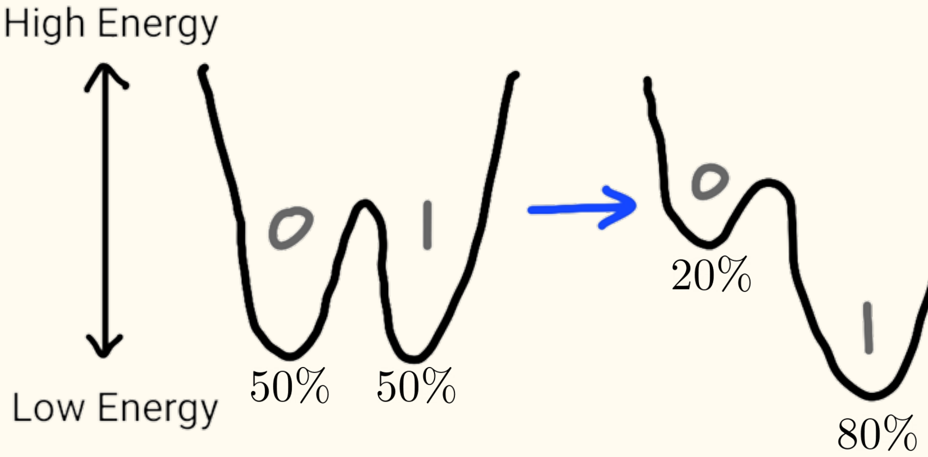
- Vector Annealing outperforms classical Simulated Annealing
  - Larger and more complex problems
- Variable Pruning has significant benefits
  - Multi-function machine environments
- Generalization of External Constraint methods
- Vector Annealing can solve a JSSP with up to 15,000 variables in a reasonable time (under one minute).

Part 5

Observations on Quantum Annealing

Quantum Annealing: How It Works

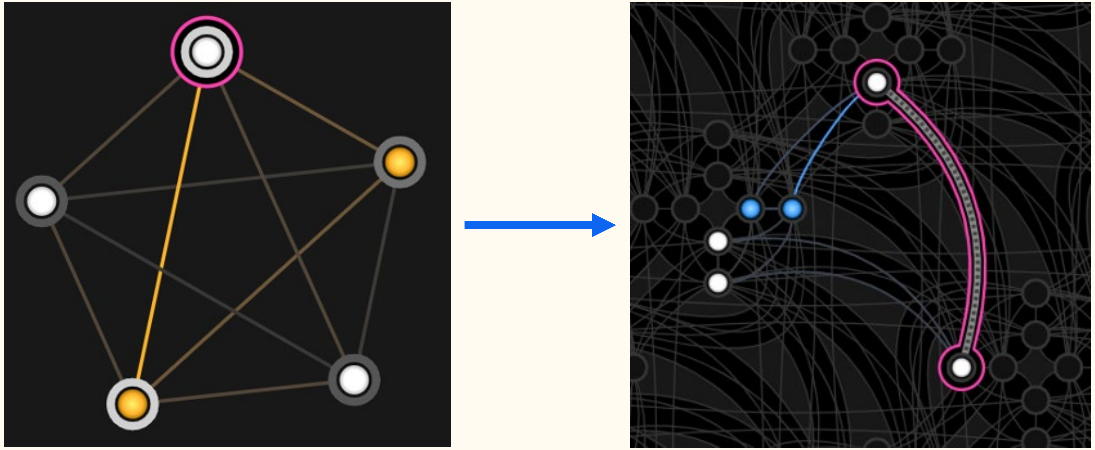
- Variable are represented by **qubits** .
- Qubits have a 50%-50% chance to end up having value 0 or 1 at the end of annealing.
- We can change the probability according to the cost function.



$$H = \cdots - 8x_i + \ldots$$

Graph Embedding

- If we have a term $x_i x_j$, we must connect those two qubits.
- Two qubits are not always connected.

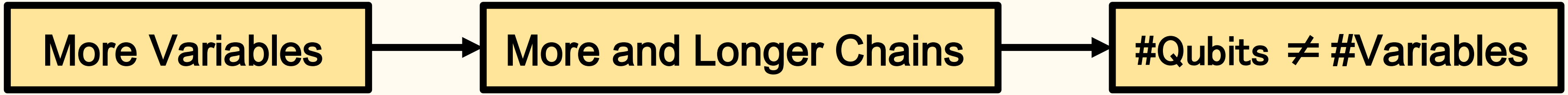


Problem graph

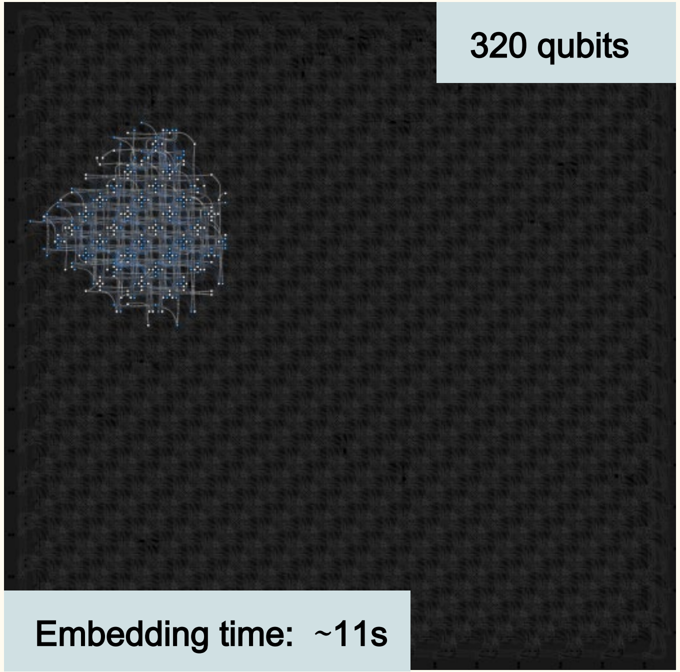
Embedded graph

- Sometimes we need a **chain** of qubits to represent one variable.

QA Capabilities



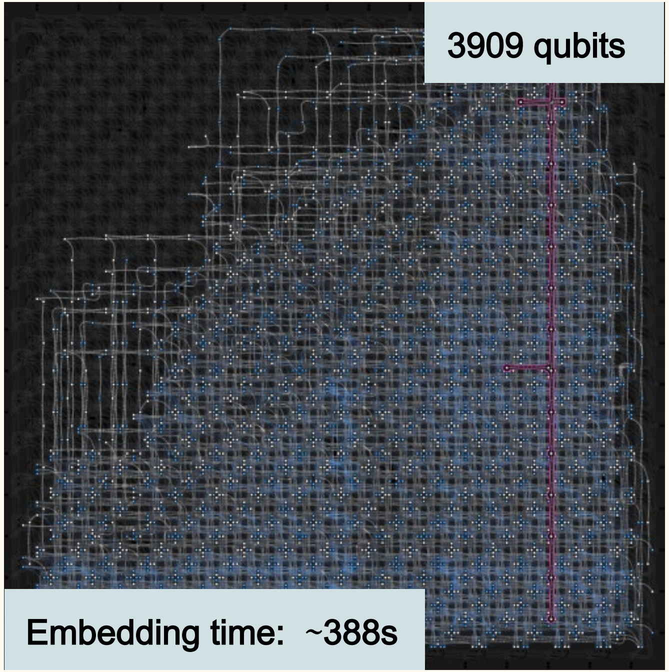
More Variables → More and Longer Chains → #Qubits $\neq$ #Variables



320 qubits

Embedding time: ~11s

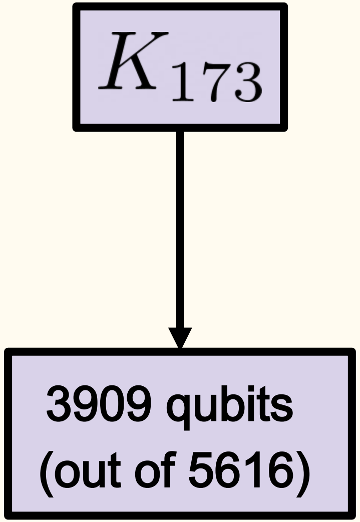
$N = 50$



3909 qubits

Embedding time: ~388s

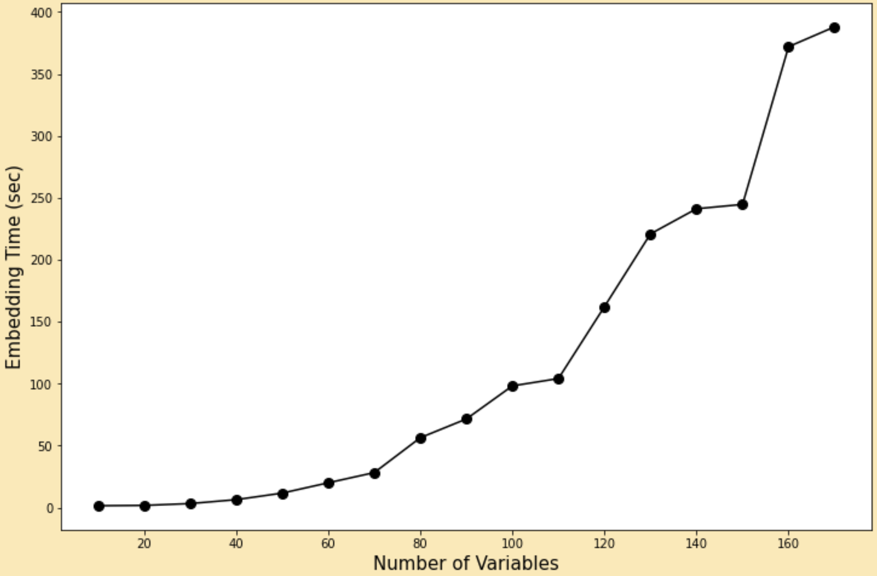
$N = 173$



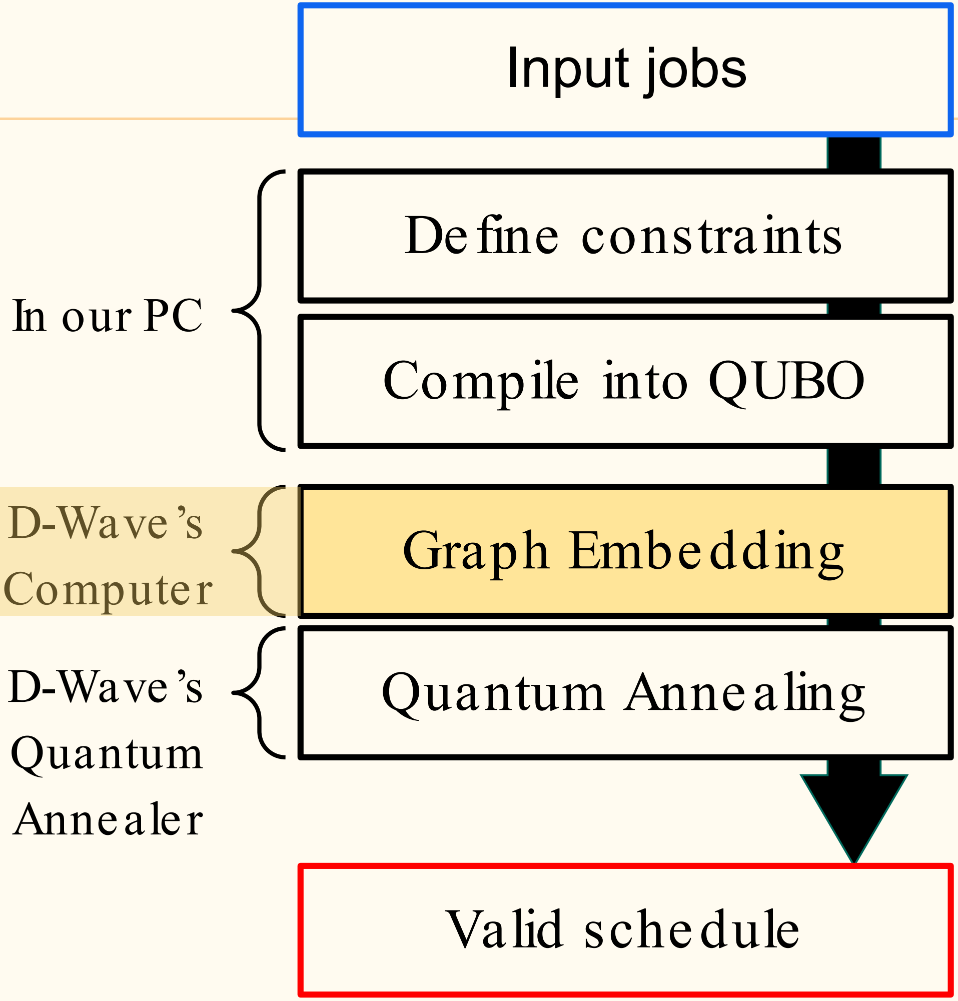
K_{173}

3909 qubits (out of 5616)

Graph Embedding Time



(All problems are complete graphs)



Input jobs

In our PC

Define constraints

Compile into QUBO

D-Wave's Computer

Graph Embedding

D-Wave's Quantum Annealer

Quantum Annealing

Valid schedule

Problem Dependency

- QA is better on solving certain problems.
- For example, Number Partitioning Problem.

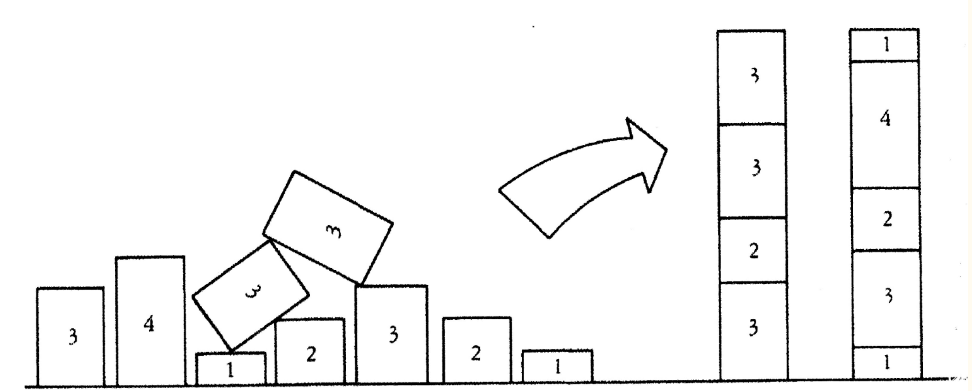


Figure 30.1 An example of the partition problem

$$H = \left(\sum_{i=1}^{N} S_i (2x_i - 1) \right)^2$$

S_i : i^{th} positive integer

With same number of variables,
QA solves Number Partitioning Problem reliably,
but QA rarely gives an optimal solution for Job Shop Scheduling Problem.

Optimal Solution Percentage

- Problem matrix and graph complexity doesn't seem to correlate with QA's effectiveness.
- We suspect that Number Partitioning Problem have many optimal solutions.

Experiment with $N = 16$, total number of solutions: 65536, for 10000 runs

	Number Partitioning Problem	**Job Shop Scheduling Problem**	**Custom Problem Matrix**
Number of Optimal Solutions	3356	12	12376
Optimal Solution Percentage	5.1208%	0.018%	18.884%
Success Rate to Find Optimal Solution	**8.58%**	**0.03%**	**62.95%**

Small Problem Comparison

Comparison for 2x2 Job Shop Scheduling Problem, with 16 variables

	neal	**Modified neal**	**VA**	**VA with Variable Pruning**	**QA**
Average Annealing Time	0.0011189s	0.0010309s	0.281s	0.179s	**0.002s**
Success Rate to Find Optimal Solution	97%	99%	100%	100%	**0.03%**
TTS	0.001469s	0.0010309s	0.281s	0.179s	**30.696s**

QA Summary

Although Quantum Annealing is very fast,

- Quantum Annealing can only solve small problems.
- Quantum Annealing can give optimal solution to problem with many optimal solutions.

Compared to other annealing method,
Quantum Annealing is not suitable for solving Job Shop Scheduling Problem

To improve that, we think Quantum Annealer needs:

- More connections between qubits. (rather than more qubits)
- Faster graph embedding algorithm.
- Better annealing schedule to reach optimal solutions reliably.

Final Conclusions

- Vector Annealing can solve a JSSP with up to 15,000 variables in a reasonable time (under one minute).
- Variable Pruning has significant benefits with multi-purpose manufacturing machines
- External Constraints is applicable to classical SA
- Due to scale and inaccuracy, Quantum Annealing is not currently suitable for solving Job Shop Scheduling problems.

Thank You!

ご清聴ありがとうございました。