Advanced Parallel School 2022
Quantum Computing – Day 3
Advanced QA applications

Mengoni Riccardo, PhD

16 Feb 2022

Quantum Computing @ CINECA

CINECA: Italian HPC center

CINECA Quantum Computing Lab:

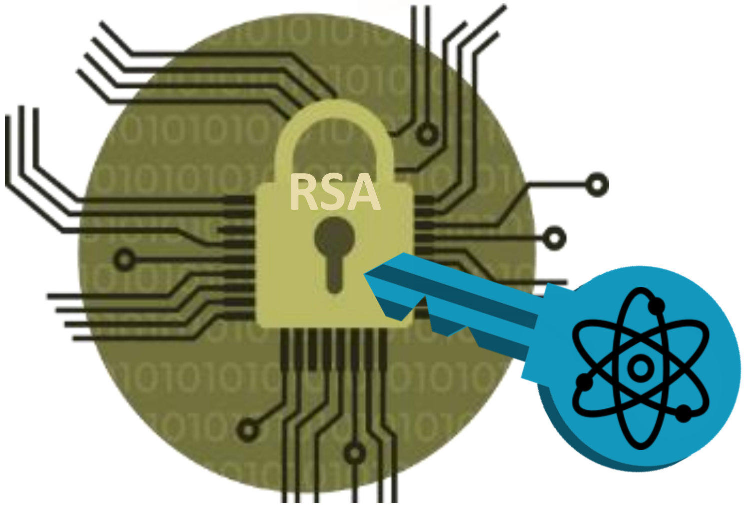
- Collaborate with Universities, Industries and QC startups

- Internship programs, Courses and Conference (HPCQC)

https://www.quantumcomputinglab.cineca.it

r.mengoni@cineca.it

QUANTUM COMPUTING LAB



Riccardo Mengoni, Daniele Ottaviani, Paolino Iorio

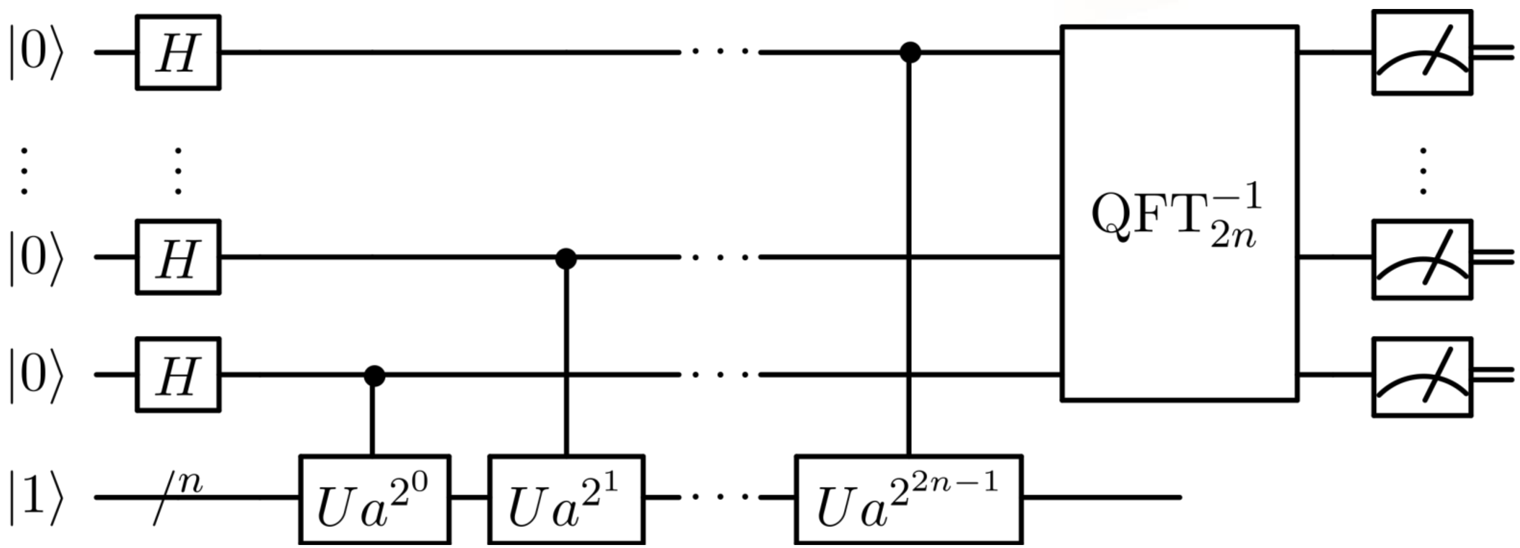
Breaking RSA security with Quanutm Annealing

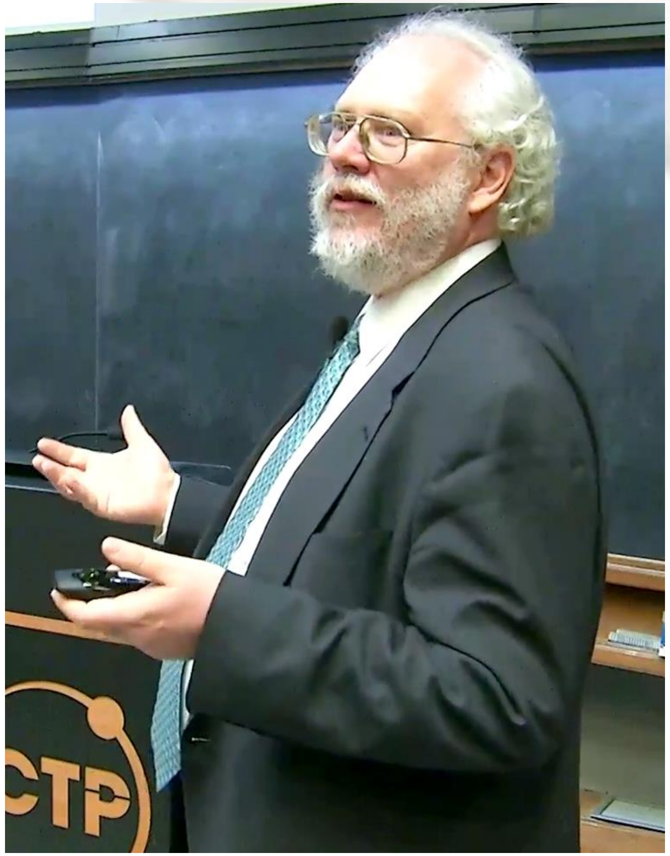
Overview:

Today we use several encryption schemes

Cryptographic Algorithm	Type
AES-256	Symmetric
SHA-256, SHA-3	Hash functions
Diffie-Hellman	Asymmetric (DLP)
RSA	Asymmetric (Factorization)
ECDSA, ECDH	Asymmetric (Elliptic Curve)
DSA	Asymmetric (DLP)

Overview: Shor's algorithm (1994)





Exponential Speedup

Overview: Grover search

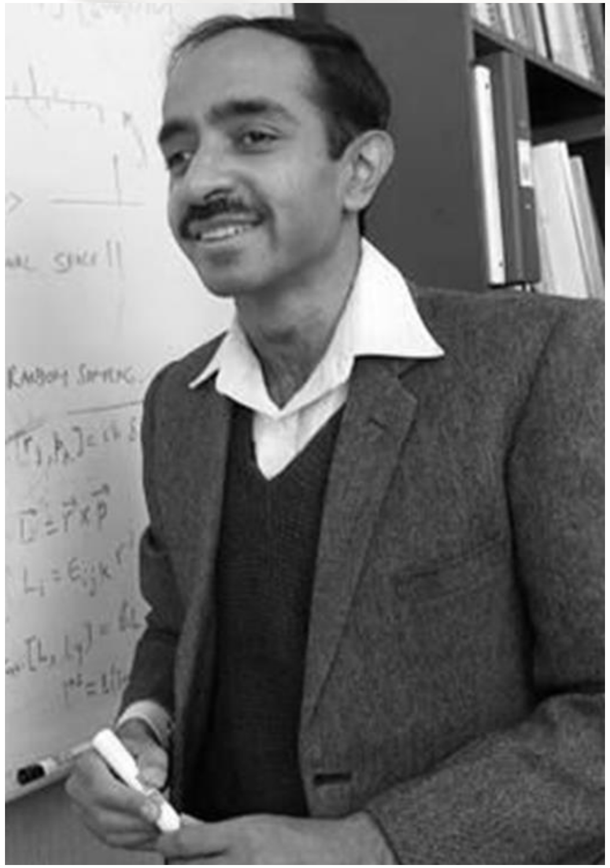
Run-time brute-force algorithm:

$$d^N$$

Run-time Grover search:

$$\sqrt{d^N}$$

Quadratic Speedup



Overview

Impact of Shor and Grover algorithms of cybersecurity

Cryptographic Algorithm	Type	Affected by	Threat Level
AES-256	Symmetric	Grover's Algorithm	Low
SHA-256, SHA-3	Hash functions	Grover's Algorithm	Low
Diffie-Hellman	Asymmetric (DLP)	Shor's Algorithm	Very High
RSA	Asymmetric (Factorization)	Shor's Algorithm	Very High
ECDSA, ECDH	Asymmetric (Elliptic Curve)	Shor's Algorithm	Very High
DSA	Asymmetric (DLP)	Shor's Algorithm	Very High

Overview

Impact of Shor and Grover algorithms of cybersecurity

Cryptographic Algorithm	Type	Affected by	Threat Level
AES-256	Symmetric	Grover's Algorithm	Low
SHA-256, SHA-3	Hash functions	Grover's Algorithm	Low
Diffie-Hellman	Asymmetric (DLP)	Shor's Algorithm	Very High
RSA	Asymmetric (Factorization)	Shor's Algorithm	Very High
ECDSA, ECDH	Asymmetric (Elliptic Curve)	Shor's Algorithm	Very High
DSA	Asymmetric (DLP)	Shor's Algorithm	Very High

Overview

Quantum Computing → In theory could hack Cryptosystems

WHY?

Shor's quantum factorization algorithm is exponentially faster than the best classical algorithm → But there is currently no <u>scalable</u> general purpose quantum computer that implements Shor's algorithm

Circuit	N Qubits	N Operations	Time
Simplicity	10240	$\simeq$ 8 billions	$\simeq$ 23 mins
Speed	$\simeq$ 5 millions	$\simeq$ 15000	$\simeq$ 2.5 ms
Qubits	4096	$\simeq$ 256 billions	$\simeq$ 12 hours
Tradeoff I	$\simeq$ 100000	$\simeq$ 1 billion	$\simeq$ 3 mins
Tradeoff II	$\simeq$ 10000	$\simeq$ 6 millions	$\simeq$ 1 sec

Objective

It is possible to formulate the factorization problem as a QUBO problem solvable via Quantum Annealing

Objective

Factorization of prime numbers ($\boldsymbol{N} = \boldsymbol{p} \times \boldsymbol{q}$) with D-Wave quantum annealer

Problem formulation

Given $\boldsymbol{N}$ **integer umber**

The problem is to find prime numbers $\boldsymbol{p}$ and $\boldsymbol{q}$ such that

$$\boldsymbol{N} = \boldsymbol{p} \times \boldsymbol{q}$$

Eg. For N=143, its prime factors are p=11 and q=13

QUBO formulation

$$\boxed{\boldsymbol{N = p \times q}}$$

Expressing the above numbers N, p and q in binary

$$N = \sum_{i=0}^{L_n - 1} 2^i n_i, \qquad p = \sum_{j=0}^{L_p - 1} 2^j p_j \qquad \text{and} \qquad q = \sum_{k=0}^{L_q - 1} 2^k q_k.$$

Eg. N =143   in binary (1, 0, 0, 0, 1, 1, 1, 1)

QUBO formulation

1) Direct Method

Objective function: $\boxed{O(p, q) = (N - p \cdot q)^2}$

Substituting binary form

$$O(p, q) = \left[\left(\sum_{i=0}^{L_n - 1} 2^i n_i \right) - \left(\sum_{j=0}^{L_p - 1} 2^j p_j \right) \cdot \left(\sum_{k=0}^{L_q - 1} 2^k q_k \right) \right]^2$$

QUBO formulation

1) Multiplication Table Method

Columns	7	6	5	4	3	2	1	0
$\mathbf{p} \cdot \mathbf{q}$					1	p_2	p_1	1
				q_1	$p_2 q_1$	$p_1 q_1$	q_1	
			q_2	$p_2 q_2$	$p_1 q_2$	q_2		
		1	p_2	p_1	1			
Carries	c_{67} c_{57}	c_{56} c_{46}	c_{45} c_{35}	c_{34} c_{24}	c_{23}	c_{12}		
N	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0

Multiplication table associated to N=p*q

QUBO formulation

Columns	7	6	5	4	3	2	1	0
$\mathbf{p} \cdot \mathbf{q}$					1	p_2	p_1	1
				q_1	p_2q_1	p_1q_1	q_1	
			q_2	p_2q_2	p_1q_2	q_2		
		1	p_2	p_1	1			
Carries	c_{67} c_{57}	c_{56} c_{46}	c_{45} c_{35}	c_{34} c_{24}	c_{23}	c_{12}		
N	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0

QUBO formulation

Columns	7	6	5	4	3	2	1	0
$\mathbf{p} \cdot \mathbf{q}$					1	p_2	p_1	1
				q_1	$p_2 q_1$	$p_1 q_1$	q_1	
			q_2	$p_2 q_2$	$p_1 q_2$	q_2		
		1	p_2	p_1	1			
Carries	c_{67} c_{57}	c_{56} c_{46}	c_{45} c_{35}	c_{34} c_{24}	c_{23}	c_{12}		
N	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0

QUBO formulation

Columns	7	6	5	4	3	2	1	0
$\mathbf{p} \cdot \mathbf{q}$					1	p_2	p_1	1
				q_1	p_2q_1	p_1q_1	q_1	
			q_2	p_2q_2	p_1q_2	q_2		
		1	p_2	p_1	1			
Carries	c_{67} c_{57}	c_{56} c_{46}	c_{45} c_{35}	c_{34} c_{24}	c_{23}	c_{12}		
N	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0

$$\begin{cases} \boxed{p_1 + q_1 - 2c_{12} = n_1} \\ p_2 + p_1q_1 + q_2 + c_{12} - (2c_{23} + 4c_{24}) = n_2 \\ 1 + p_2q_1 + p_1q_2 + 1 + c_{23} - (2c_{34} + 4c_{35}) = n_3 \\ q_1 + p_2q_2 + p_1 + c_{24} + c_{34} - (2c_{45} + 4c_{46}) = n_4 \\ p_2 + q_2 + c_{45} + c_{35} - (2c_{56} + 4c_{57}) = n_5 \\ 1 + c_{56} + c_{46} - 2c_{67} = n_6 \\ c_{57} + c_{67} = n_7 \end{cases}$$

QUBO formulation

Columns	7	6	5	4	3	2	1	0
p · q					1	p_2	p_1	1
				q_1	p_2q_1	p_1q_1	q_1	
			q_2	p_2q_2	p_1q_2	q_2		
		1	p_2	p_1	1			
Carries	c_{67}	c_{56}	c_{45}	c_{34}	c_{23}	c_{12}		
	c_{57}	c_{46}	c_{35}	c_{24}				
N	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0

$$\begin{cases} \boxed{p_1 + q_1 - 2c_{12} = n_1} \\ p_2 + p_1q_1 + q_2 + c_{12} - (2c_{23} + 4c_{24}) = n_2 \\ 1 + p_2q_1 + p_1q_2 + 1 + c_{23} - (2c_{34} + 4c_{35}) = n_3 \\ q_1 + p_2q_2 + p_1 + c_{24} + c_{34} - (2c_{45} + 4c_{46}) = n_4 \\ p_2 + q_2 + c_{45} + c_{35} - (2c_{56} + 4c_{57}) = n_5 \\ 1 + c_{56} + c_{46} - 2c_{67} = n_6 \\ c_{57} + c_{67} = n_7 \end{cases}$$

$$O(p,q) = \boxed{(p_1 + q_1 - n_1 - 2c_{12})^2} +$$
$$+ (p_2 + p_1q_1 + q_2 + c_{12} - n_2 - 2c_{23} - 4c_{24})^2 + \ \dots +$$
$$+ (1 + c_{56} + c_{46} - n_6 - 2c_{67}) + (c_{57} + c_{67} - n_7)^2$$

QUBO formulation

Columns	7	6	5	4	3	2	1	0
$\mathbf{p}\cdot\mathbf{q}$					1	p_2	p_1	1
				q_1	p_2q_1	p_1q_1	q_1	
			q_2	p_2q_2	p_1q_2	q_2		
		1	p_2	p_1	1			
Carries	c_{67} c_{57}	c_{56} c_{46}	c_{45} c_{35}	c_{34} c_{24}	c_{23}	c_{12}		
N	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0

$$\begin{cases} p_1 + q_1 - 2c_{12} = n_1 \\ \boxed{p_2 + p_1q_1 + q_2 + c_{12} - (2c_{23} + 4c_{24}) = n_2} \\ 1 + p_2q_1 + p_1q_2 + 1 + c_{23} - (2c_{34} + 4c_{35}) = n_3 \\ q_1 + p_2q_2 + p_1 + c_{24} + c_{34} - (2c_{45} + 4c_{46}) = n_4 \\ p_2 + q_2 + c_{45} + c_{35} - (2c_{56} + 4c_{57}) = n_5 \\ 1 + c_{56} + c_{46} - 2c_{67} = n_6 \\ c_{57} + c_{67} = n_7 \end{cases}$$

$$O(p,q) = (p_1 + q_1 - n_1 - 2c_{12})^2 + \\ + \boxed{(p_2 + p_1q_1 + q_2 + c_{12} - n_2 - 2c_{23} - 4c_{24})^2} + \ldots + \\ + (1 + c_{56} + c_{46} - n_6 - 2c_{67}) + (c_{57} + c_{67} - n_7)^2$$

QUBO formulation

Columns	7	6	5	4	3	2	1	0
$\mathbf{p} \cdot \mathbf{q}$					1	p_2	p_1	1
				q_1	$p_2 q_1$	$p_1 q_1$	q_1	
			q_2	$p_2 q_2$	$p_1 q_2$	q_2		
		1	p_2	p_1	1			
Carries	c_{67}	c_{56}	c_{45}	c_{34}	c_{23}	c_{12}		
	c_{57}	c_{46}	c_{35}	c_{24}				
N	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0

$$\begin{cases} p_1 + q_1 - 2c_{12} = n_1 \\ p_2 + p_1 q_1 + q_2 + c_{12} - (2c_{23} + 4c_{24}) = n_2 \\ 1 + p_2 q_1 + p_1 q_2 + 1 + c_{23} - (2c_{34} + 4c_{35}) = n_3 \\ q_1 + p_2 q_2 + p_1 + c_{24} + c_{34} - (2c_{45} + 4c_{46}) = n_4 \\ p_2 + q_2 + c_{45} + c_{35} - (2c_{56} + 4c_{57}) = n_5 \\ 1 + c_{56} + c_{46} - 2c_{67} = n_6 \\ \boxed{c_{57} + c_{67} = n_7} \end{cases}$$

$$O(p,q) = (p_1 + q_1 - n_1 - 2c_{12})^2 +$$
$$+ (p_2 + p_1 q_1 + q_2 + c_{12} - n_2 - 2c_{23} - 4c_{24})^2 + \ \ldots +$$
$$+ (1 + c_{56} + c_{46} - n_6 - 2c_{67}) + \boxed{(c_{57} + c_{67} - n_7)^2}$$

QUBO formulation

1) Block-Multiplication Table Method

Blocks	**III**			**II**		**I**		
$\mathbf{p} \cdot \mathbf{q}$					1	p_2	p_1	1
				q_1	$p_2 q_1$	$p_1 q_1$	q_1	
			q_2	$p_2 q_2$	$p_1 q_2$	q_2		
		1	p_2	p_1	1			
Carries		c_4	c_3	c_2	c_1			
N	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0

Multiplication table associated to N=p*q is divided in blocks in order to reduce the number of carries, hence variables involved

QUBO formulation

Blocks	III			II		I		
$\mathbf{p} \cdot \mathbf{q}$					1	p_2	p_1	1
				q_1	p_2q_1	p_1q_1	q_1	
			q_2	p_2q_2	p_1q_2	q_2		
		1	p_2	p_1	1			
Carries		c_4	c_3	c_2	c_1			
N	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0

$$
\begin{cases}
(p_1 + q_1) + 2(p_2 + p_1q_1 + q_2) - (8c_2 + 4c_1) = n_1 + 2n_2 \\
(1 + p_2q_1 + p_1q_2 + 1 + c_1) + 2(q_1 + p_2q_2 + p_1 + c_2) - \\
-(8c_4 + 4c_3) = n_3 + 2n_4 \\
(q_2 + p_2 + c_3) + 2(1 + c_4) = n_5 + 2n_6 + 4n_7
\end{cases}
$$

$$
\begin{aligned}
O(p,q) = {} & [(p_1 + q_1) + 2(p_2 + p_1q_1 + q_2) - (8c_2 + 4c_1) - \\
& -(n_1 + 2n_2)]^2 + [(1 + p_2q_1 + p_1q_2 + 1 + c_1) + \\
& +2(q_1 + p_2q_2 + p_1 + c_2) - (8c_4 + 4c_3) - (n_3 + 2n_4)]^2 + \\
& + [(q_2 + p_2 + c_3) + 2(1 + c_4) - (n_5 + 2n_6 + 4n_7)]^2
\end{aligned}
$$

QUBO formulation

Blocks	III			II		I		
$\mathbf{p \cdot q}$					1	p_2	p_1	1
				q_1	p_2q_1	p_1q_1	q_1	
			q_2	p_2q_2	p_1q_2	q_2		
		1	p_2	p_1	1			
Carries		c_4	c_3	c_2	c_1			
N	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0

$$\begin{cases} (p_1+q_1)+2(p_2+p_1q_1+q_2)-(8c_2+4c_1)=n_1+2n_2 \\ \boxed{\begin{aligned}&(1+p_2q_1+p_1q_2+1+c_1)+2(q_1+p_2q_2+p_1+c_2)- \\ &-(8c_4+4c_3)=n_3+2n_4\end{aligned}} \\ (q_2+p_2+c_3)+2(1+c_4)=n_5+2n_6+4n_7 \end{cases}$$

$$\begin{aligned} O(p,q) = {} & [(p_1+q_1)+2(p_2+p_1q_1+q_2)-(8c_2+4c_1)- \\ & -(n_1+2n_2)]^2 + \boxed{[(1+p_2q_1+p_1q_2+1+c_1)+} \\ & \boxed{+2(q_1+p_2q_2+p_1+c_2)-(8c_4+4c_3)-(n_3+2n_4)]^2}+ \\ & +[(q_2+p_2+c_3)+2(1+c_4)-(n_5+2n_6+4n_7)]^2 \end{aligned}$$

QUBO formulation

Blocks	III			II		I		
$\mathbf{p} \cdot \mathbf{q}$					1	p_2	p_1	1
				q_1	p_2q_1	p_1q_1	q_1	
			q_2	p_2q_2	p_1q_2	q_2		
		1	p_2	p_1	1			
Carries		c_4	c_3	c_2	c_1			
N	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0

$$
\begin{cases}
(p_1+q_1)+2(p_2+p_1q_1+q_2)-(8c_2+4c_1)=n_1+2n_2 \\
(1+p_2q_1+p_1q_2+1+c_1)+2(q_1+p_2q_2+p_1+c_2)- \\
-(8c_4+4c_3)=n_3+2n_4 \\
(q_2+p_2+c_3)+2(1+c_4)=n_5+2n_6+4n_7
\end{cases}
$$

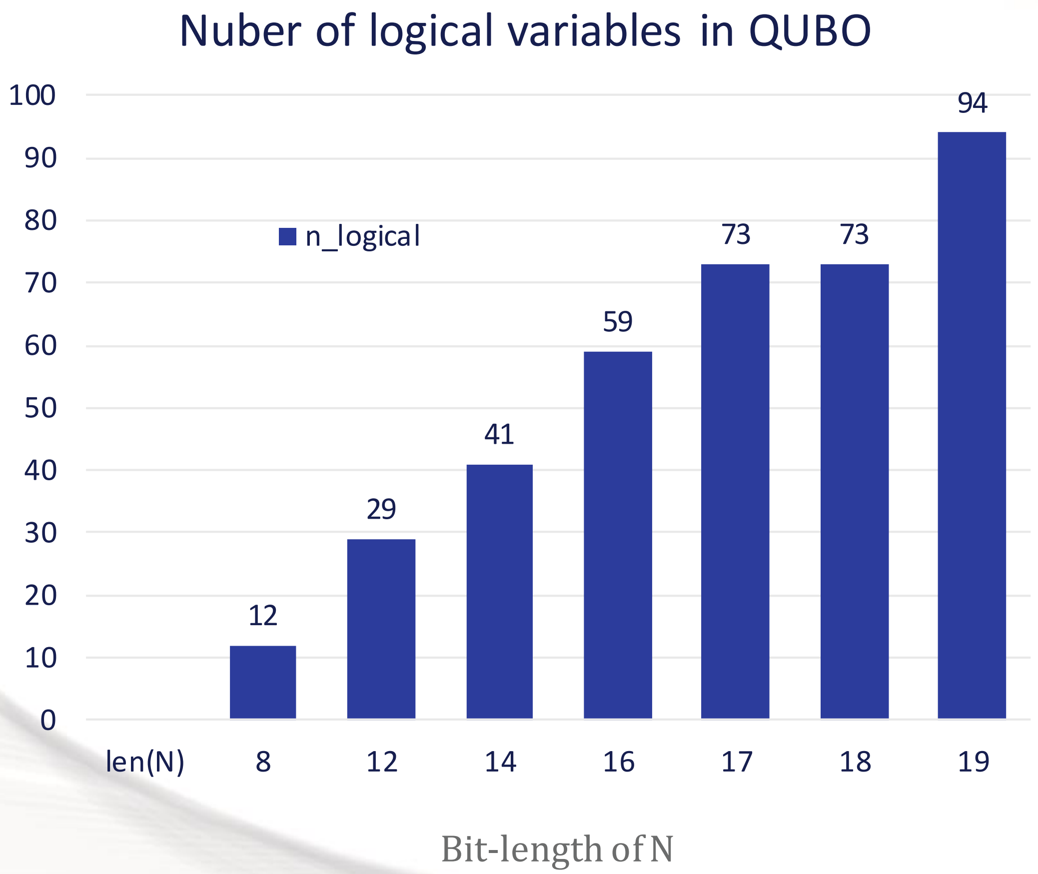
$$
\begin{aligned}
O(p,q) = {} & [(p_1+q_1)+2(p_2+p_1q_1+q_2)-(8c_2+4c_1)- \\
& -(n_1+2n_2)]^2+[(1+p_2q_1+p_1q_2+1+c_1)+ \\
& +2(q_1+p_2q_2+p_1+c_2)-(8c_4+4c_3)-(n_3+2n_4)]^2+ \\
& +[(q_2+p_2+c_3)+2(1+c_4)-(n_5+2n_6+4n_7)]^2
\end{aligned}
$$

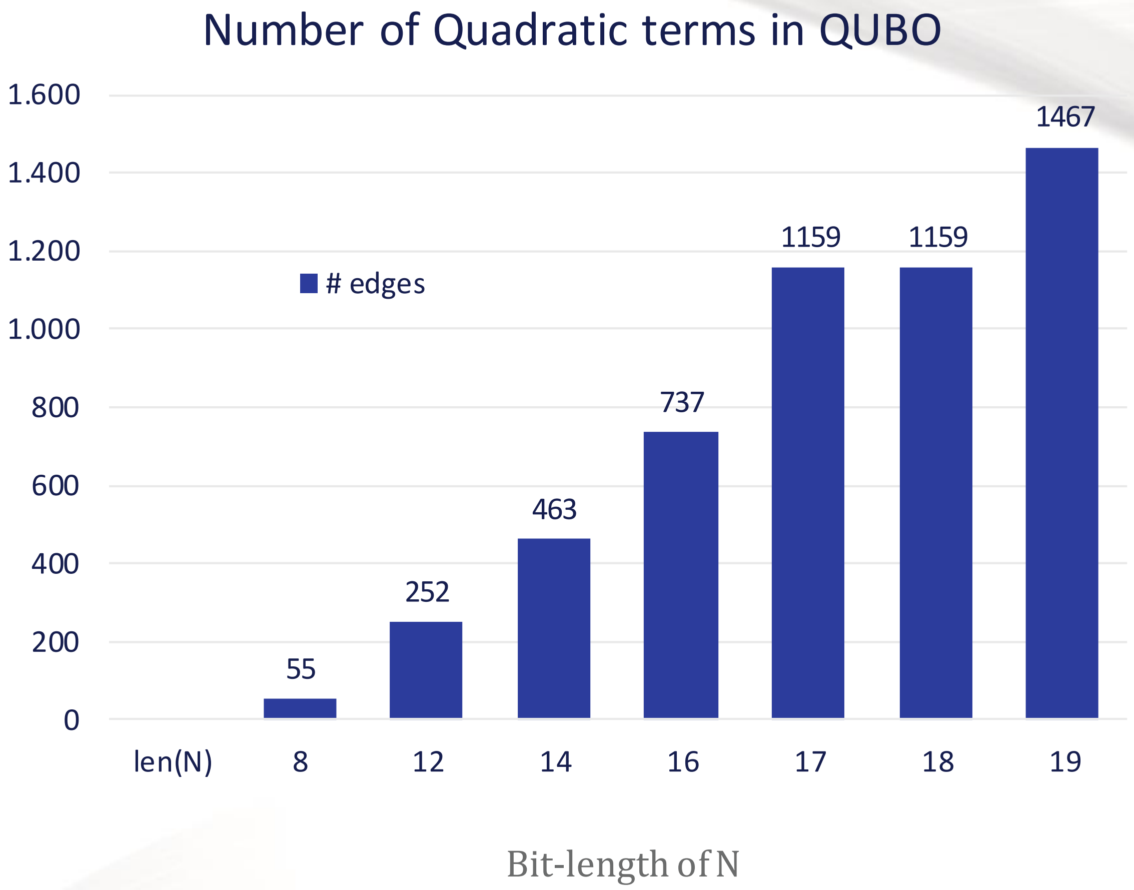
Only three terms in the Objective function

Analysis of QUBO resources

N	p	q	Length in bit of N: $len(N)$	Number of variables in QUBO: $n_{logical}$	Number of quadratic terms in QUBO
143	13	11	8	12	55
3127	59	53	**12**	29	252
8881	107	83	**14**	41	463
59989	251	239	16	59	737
103459	307	337	**17**	73	1159
231037	499	363	**18**	73	1159
376289	659	571	19	94	1467

Analysis of QUBO resources





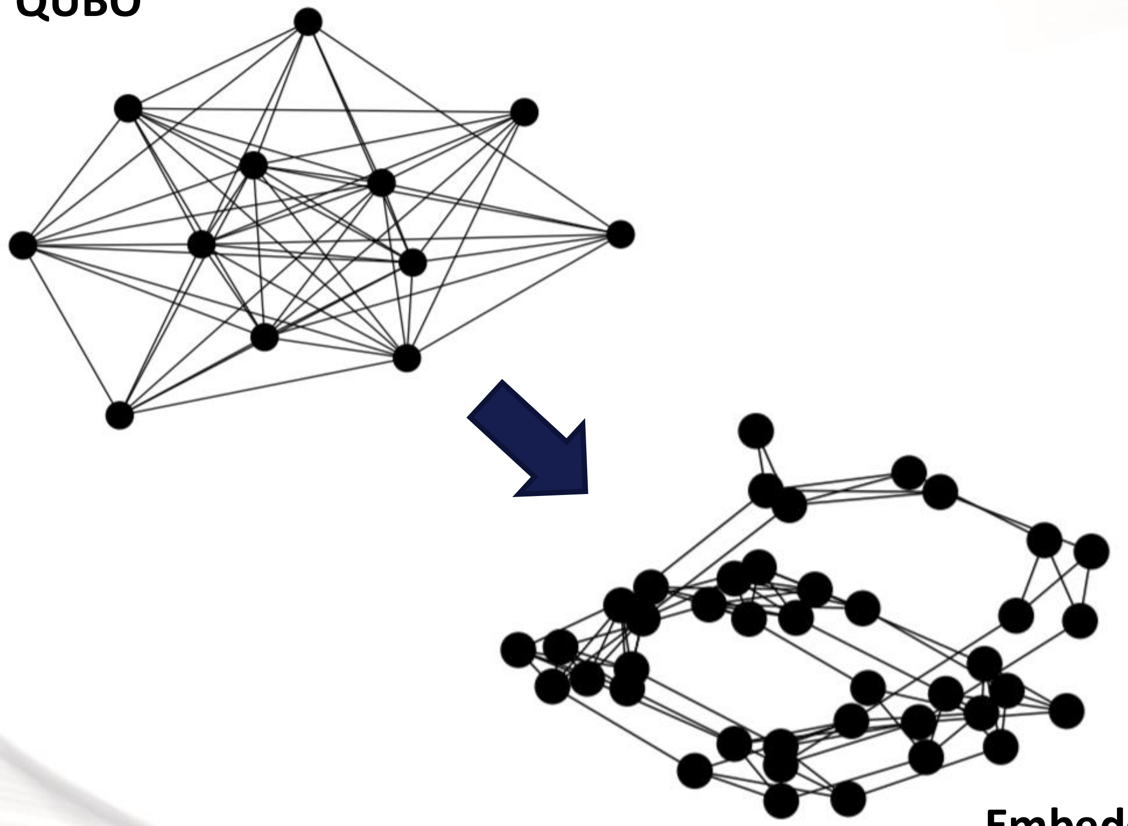
Embedding QUBO problem into the D-Wave topology

Embedding:

- The **"logical" graph (QUBO)** is mapped into the **physical structure of the D-Wave** (side image) using a **heuristic algorithm**
- This **heuristic algorithm** searches for the best minor embedding and returns the **"embedded" graph**
- In the embedded graph, **single logical variables are mapped into chains of physical variables**. This leads to an **increase in the size of the embedded problem**.

Embedding QUBO problem into the D-Wave topology

QUBO



Embedded

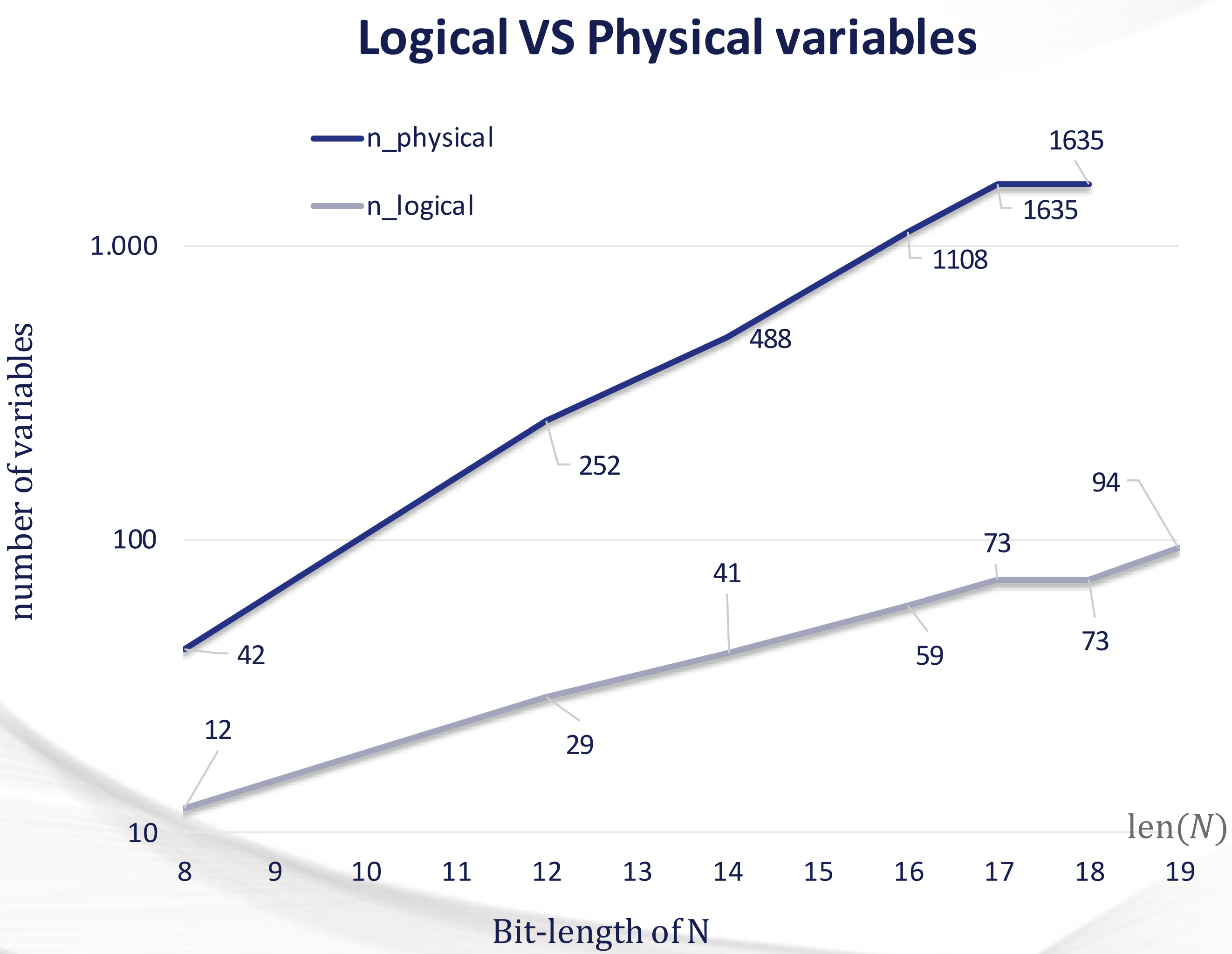
Embedding:

- The **"logical" graph (QUBO)** is mapped into the **physical structure of the D-Wave** (side image) using a **heuristic algorithm**

- This **heuristic algorithm** searches for the best minor embedding and returns the **"embedded" graph**

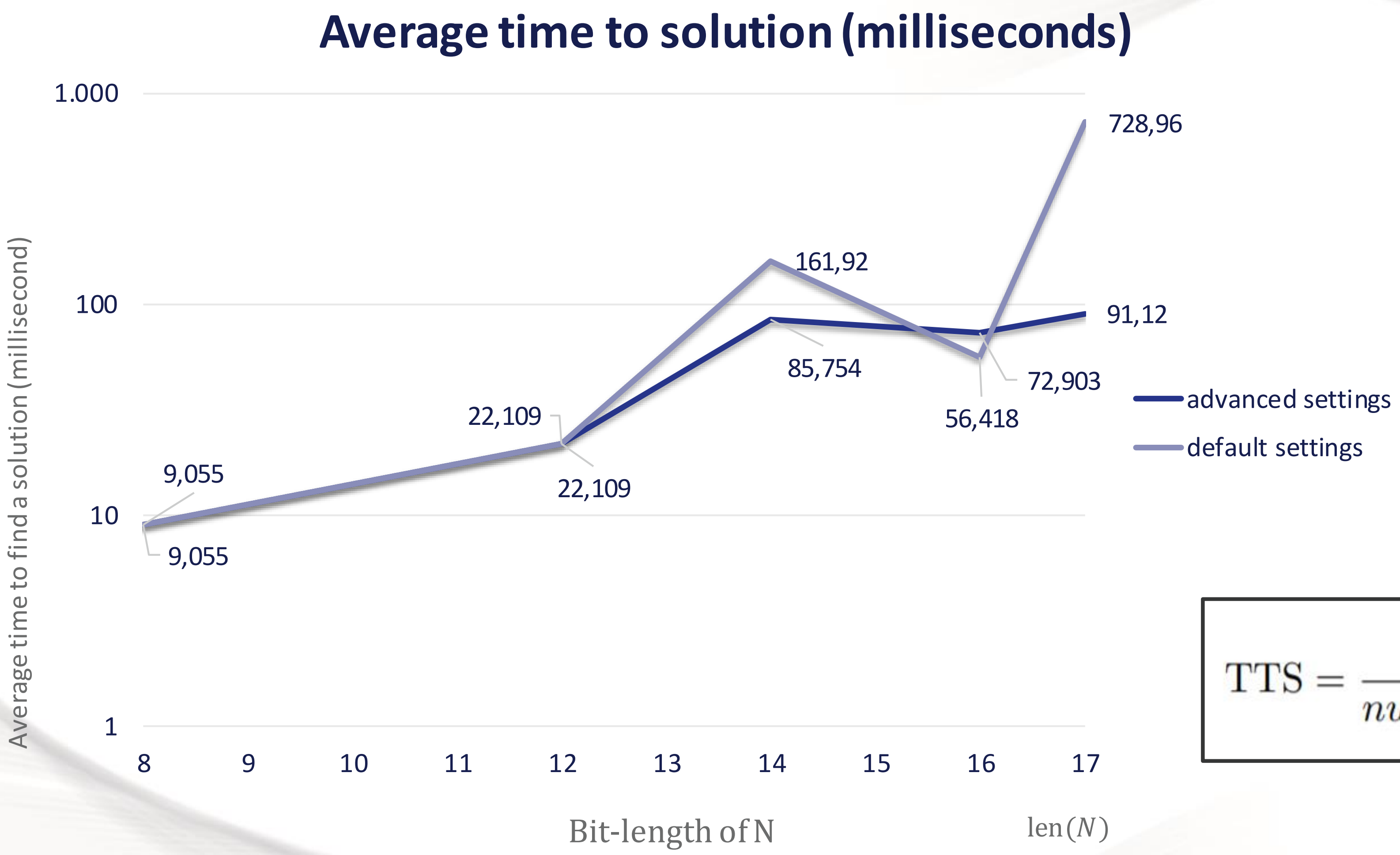
- In the embedded graph, **single logical variables are mapped into chains of physical variables**. This leads to an **increase in the size of the embedded problem**.

Embedding QUBO problem into the D-Wave topology



- **n_physical**: number of physical variables (physical qubits) found with minor embedding algorithm
- **n_logical:** number of logical variables of the QUBO problem
- **Increase in the gap** between physical qubits and logical variables as the number to be factored **increases N**

D-Wave Runs: Time To Solution (TTS)



Advanced settings:

- Extended J range
- Flux drift compensation
- Annealing offsets

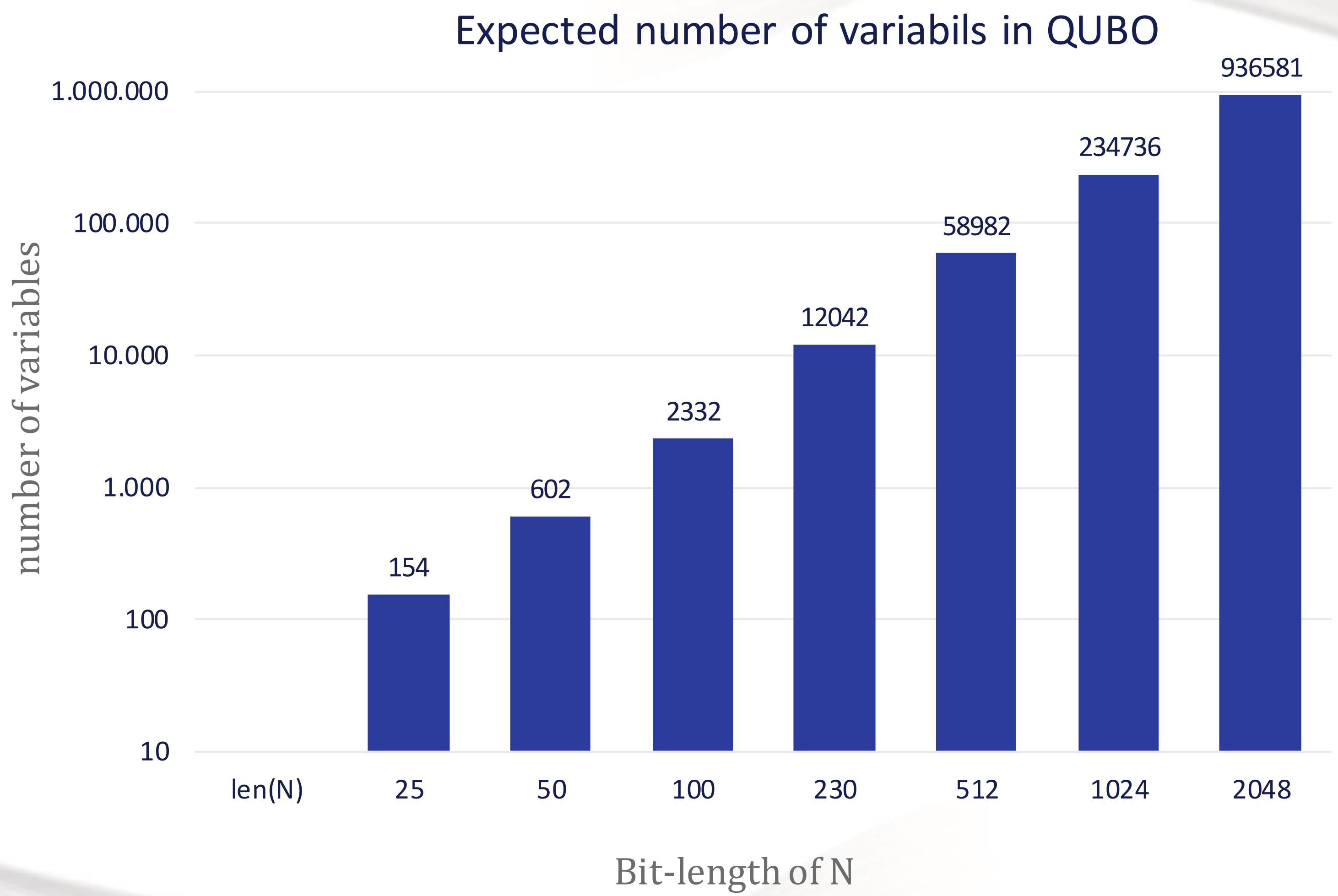
$$\text{TTS} = \frac{total\ QPU\ access\ time}{number\ of\ times\ N\ is\ factored\ correctly}$$

D-Wave Runs: Time To Solution (TTS)

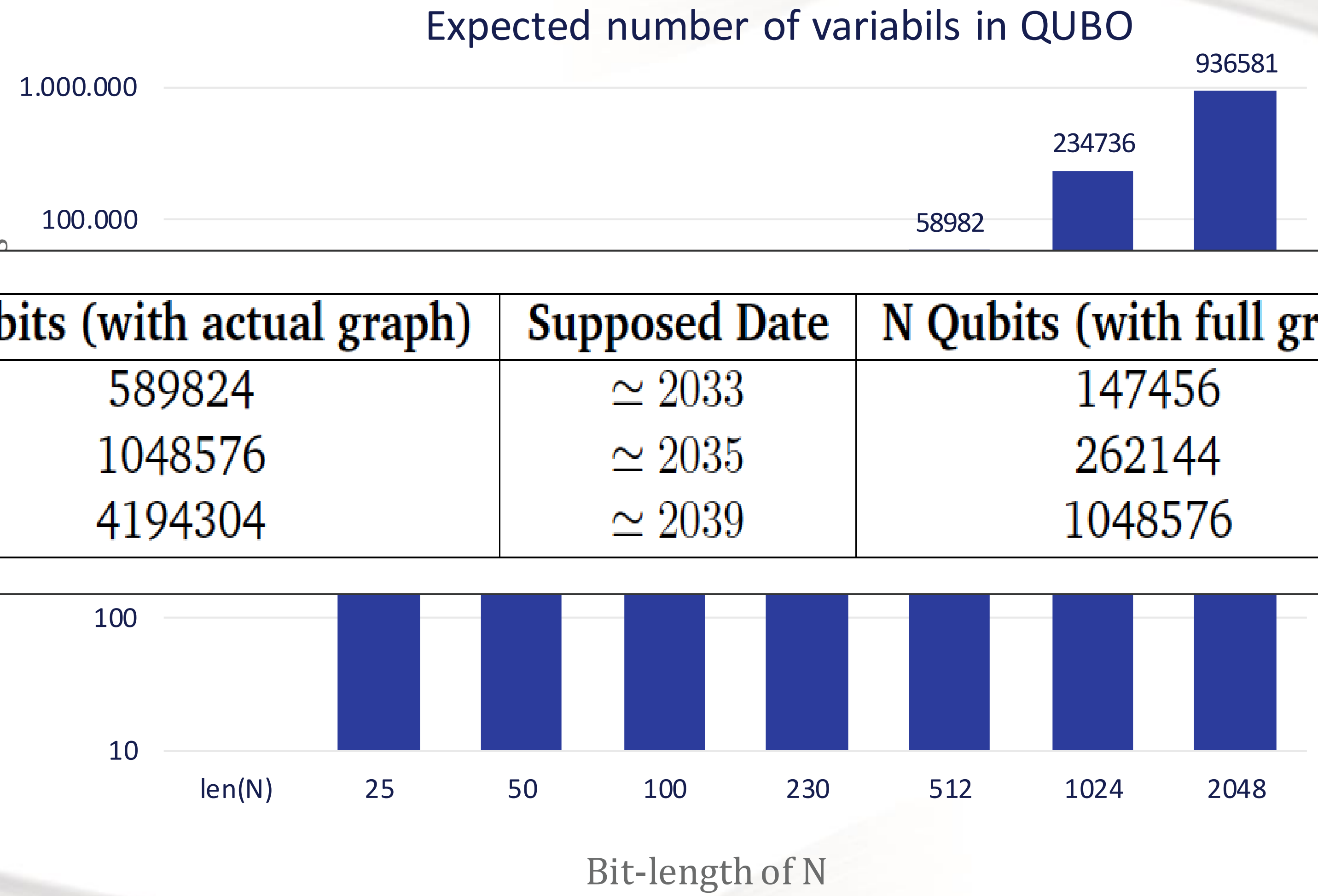
N	Bit-length of N: $len(N)$	TTS: Default settings (milliseconds)	TTS: Advanced settings (milliseconds)
143	8	9,055	9,055
3127	12	22,109	22,109
8881	14	161,92	85,754
59989	16	56,418	72,903
103459	17	728,96	91,12
231037	18	not found	not found
376289	19	not found	not found

Stimiamo che il più potente supercomputer al mondo, Summit (150mila TFlop/s), riesca a fattorizzare un RSA-80 in 90 millisecondi

Expected QUBO variables increasing problem size



Expected QUBO variables increasing problem size



RSA type	N Qubits (with actual graph)	Supposed Date	N Qubits (with full graph)	Supposed Date
RSA-768	589824	$\simeq 2033$	147456	$\simeq 2029$
RSA-1024	1048576	$\simeq 2035$	262144	$\simeq 2031$
RSA-2048	4194304	$\simeq 2039$	1048576	$\simeq 2035$

Conclusions

Factorization of prime numbers with D-Wave quantum annealer

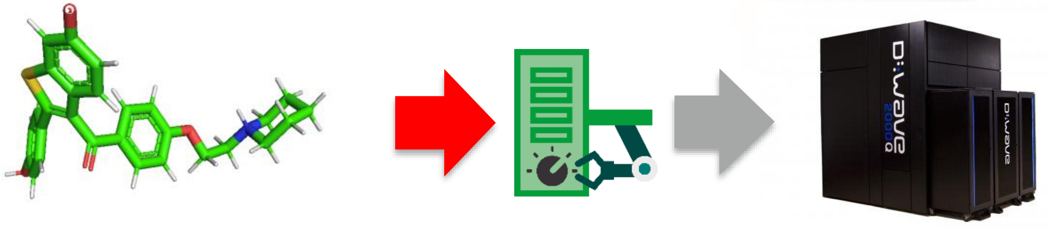
- **Formulation and analysis of the associated QUBO problem:**
  - Block matrix importance
  - - Linear scaling of logic variables

- **Embedding of QUBO into D-Wave hardware topology:**

- Polinomial Scaling polinomiale of physical variables
- Max embedding found for *len(N)=17*

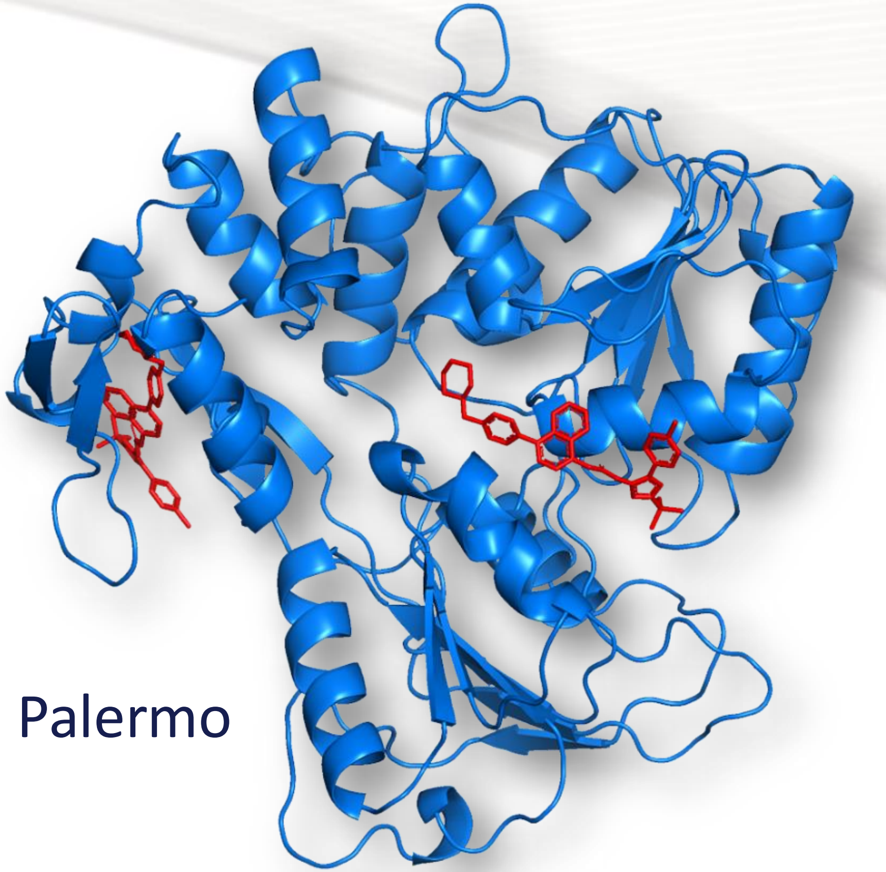
- **Average Time To Solution (TTS) found using D-Wave:**

- Max problem dimension → factorization *N=103459* i.e. *len(N)=17*
- *Average Time To Solution (TTS)* below 100 millisecondis (with advanced settings)



POLITECNICO MILANO 1863
CINECA

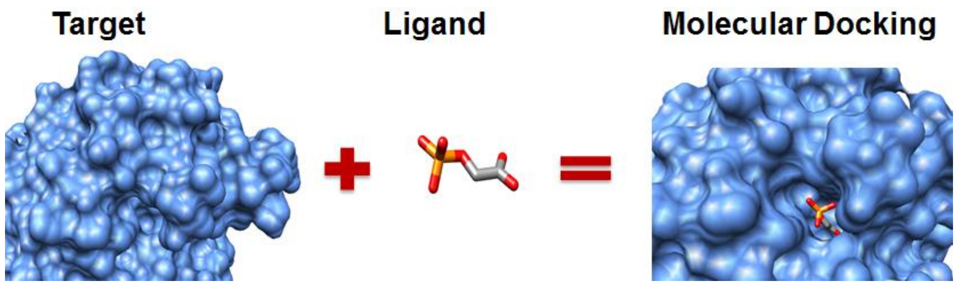
Riccardo Mengoni, Kevin Mato, Daniele Ottaviani, Gianluca Palermo

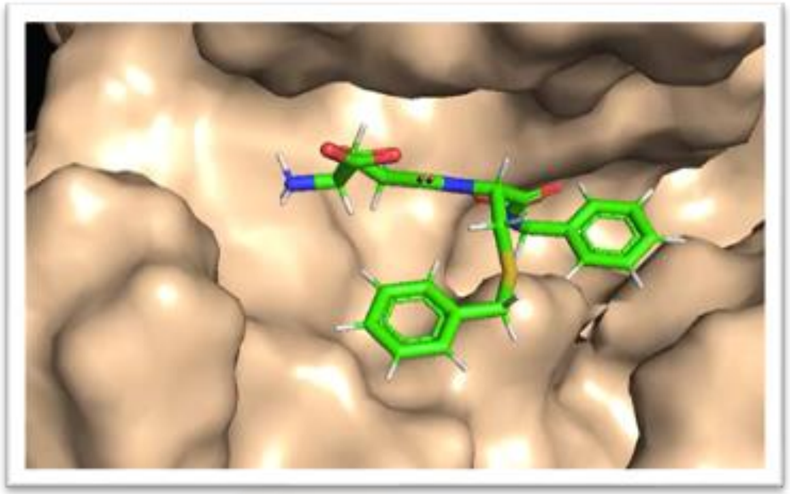


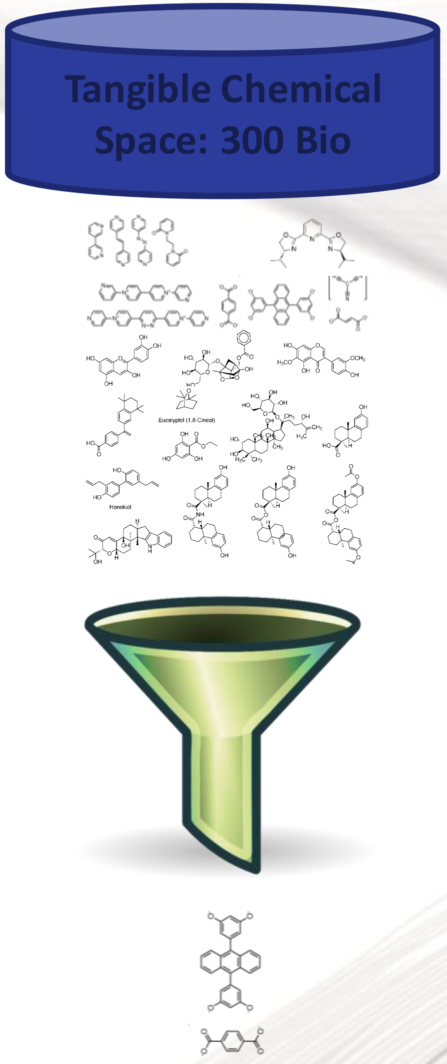
Molecule Unfolding with Quantum Annealing

Molecular Docking for Virtual Screening

- **Molecular docking** is a method to calculate the **preferred position and shape** of one **molecule** to a second when bound to each other
  - Shape Complementarity
  - Scoring function to evaluate the binding affinity







3 Phases Process

Ligand Expansion

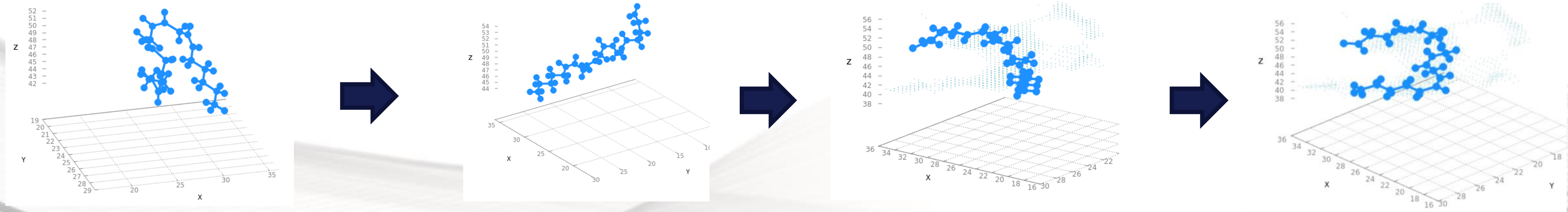
- MOL2 ligand elaboration
- Identification of the rotatable bonds
- Internal distances maximization
- Removes tool related bias (e.g. smile-to-3D)

Initial Placement

- Ligand main fragments decomposition
- Ligand initial poses Identification
- Placement of the ligand into the pocket with rigid roto-translations

Shape Refinement

- Use of the rotatable bonds to modify the ligand shape and to match the protein pocket
- Docking Score Maximization

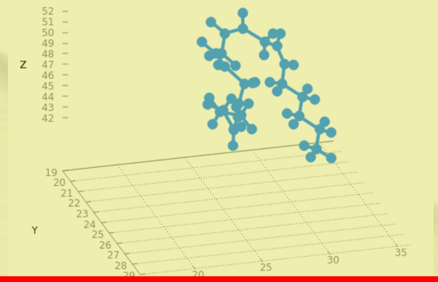


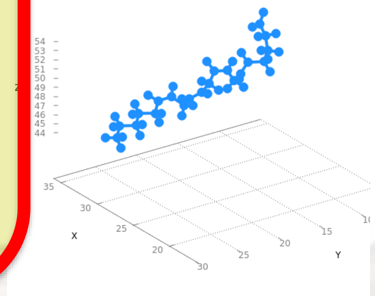
3 Phases Process

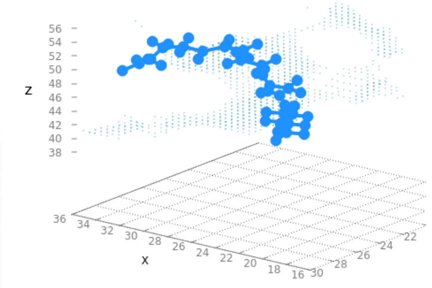
Ligand Expansion
- MOL2 ligand elaboration
- Identification of the rotatable bonds
- Internal distances maximization
- Removes tool related bias (e.g. smile-to-3D)

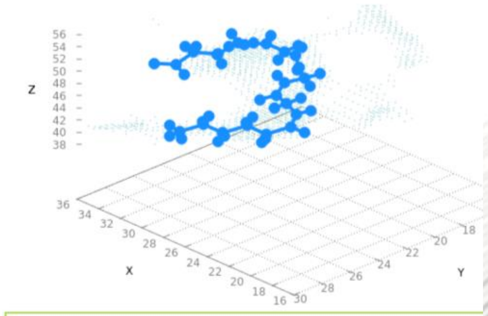
Initial Placement
- Ligand main fragments decomposition
- Ligand initial poses Identification
- Placement of the ligand into the pocket with rigid roto-translations

Shape Refinement
- Use of the rotatable bonds to modify the ligand shape and to match the protein pocket
- Docking Score Maximization



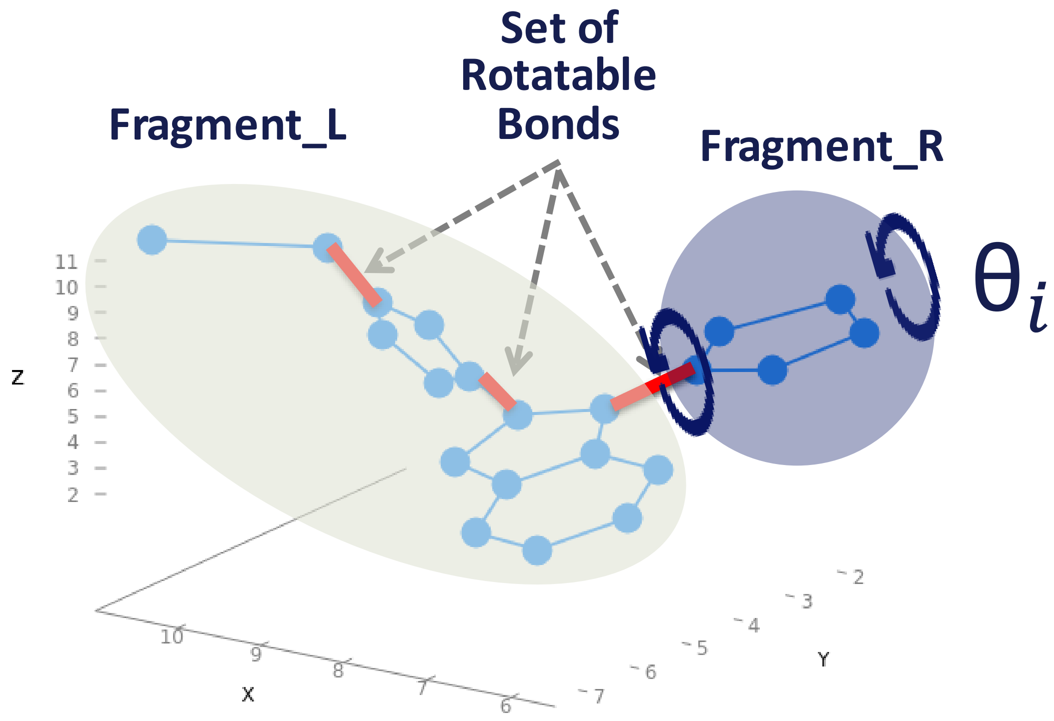






Problem Definition

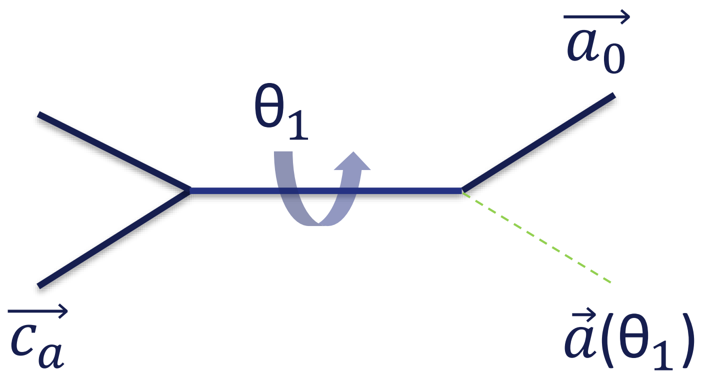
Objective: **find** the unfolded **torsion configuration** that maximizes the molecular volume, or equivalently, that **maximizes the distances between fragments.**

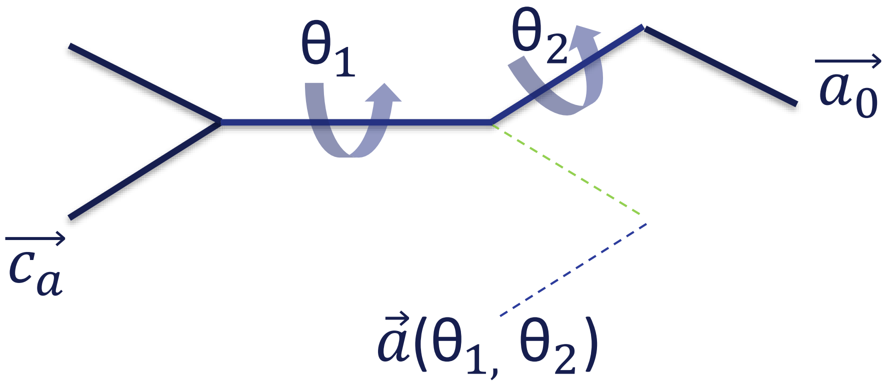


$$\theta = [\theta_1, \ldots, \theta_M]$$

Problem Definition

- To each **torsion** is associated is a **rotation matrix R**.

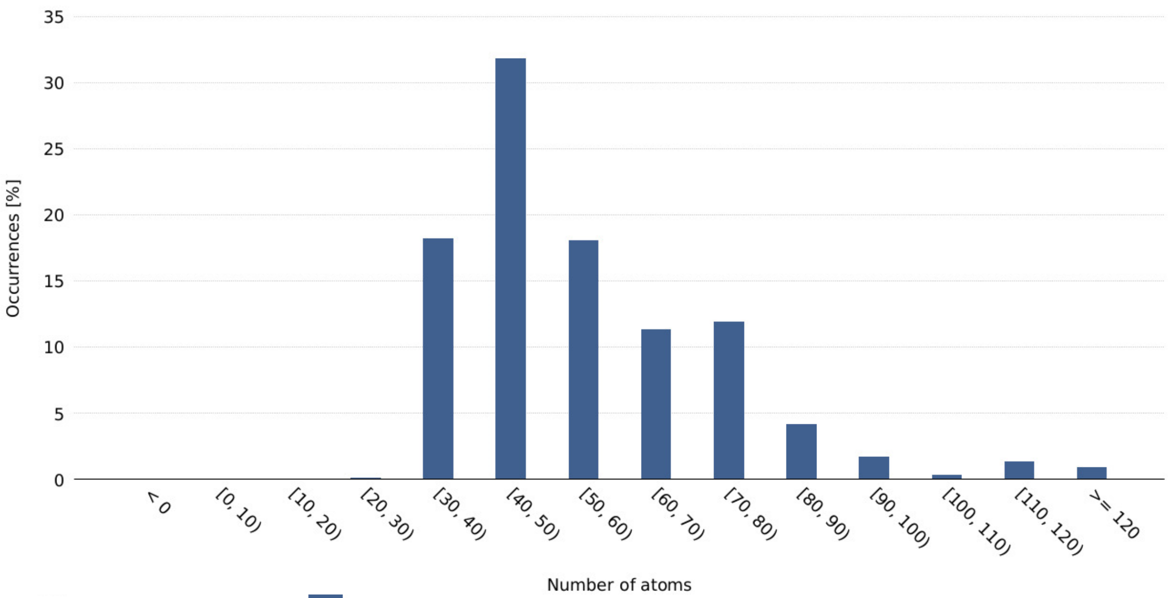


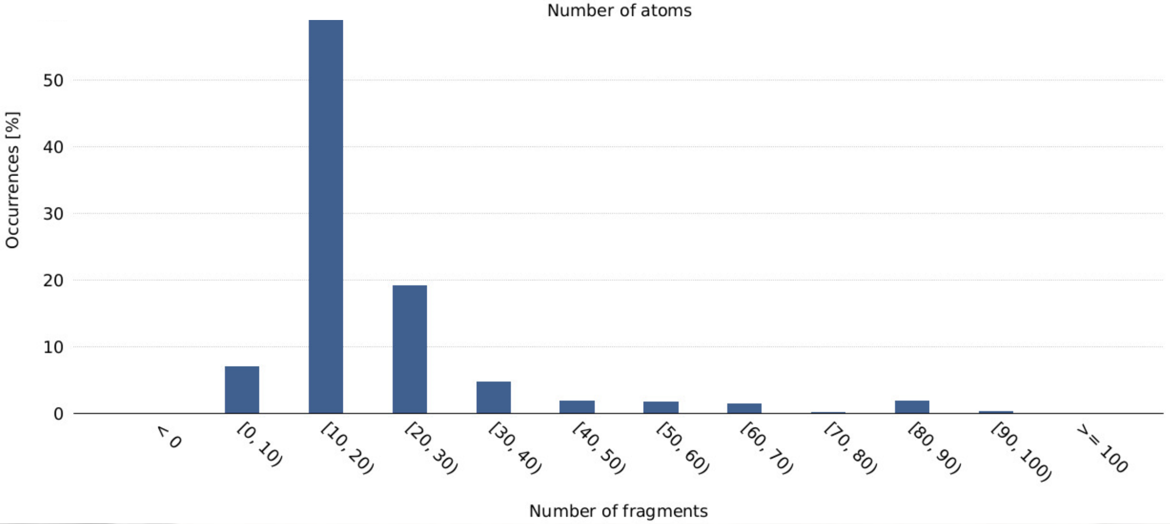


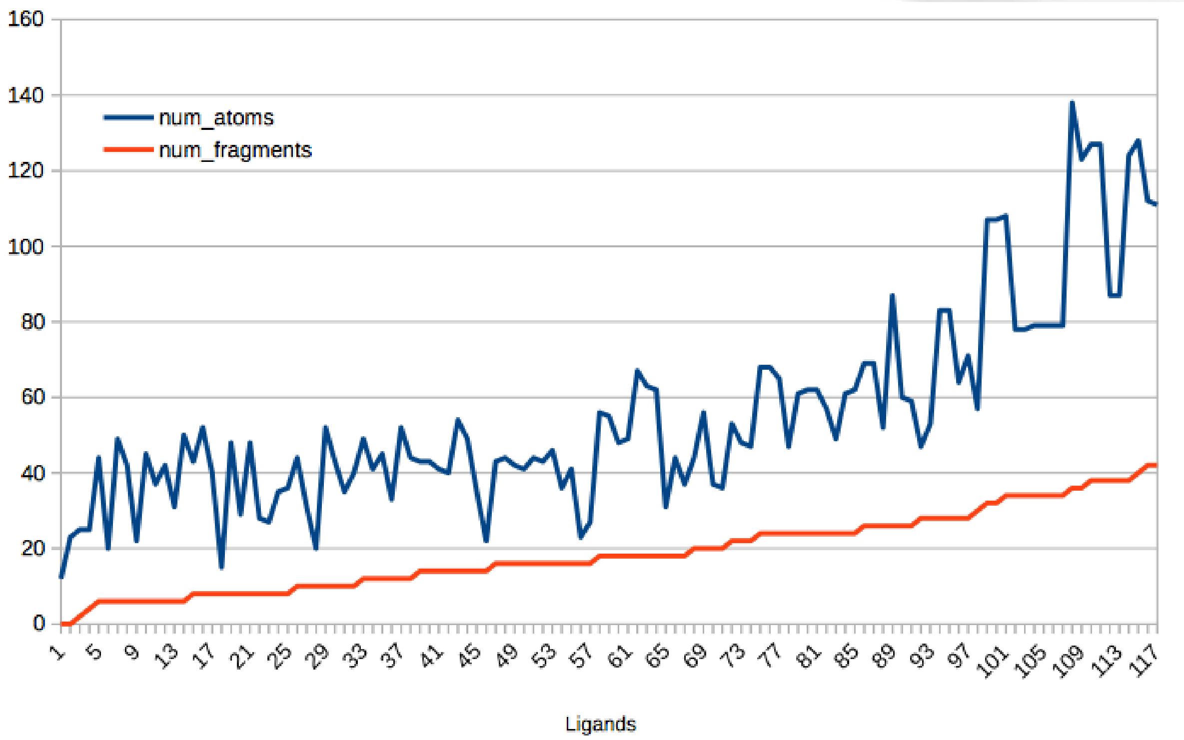
$$\vec{a}(\theta_1) = R(\theta_1)\overrightarrow{a_0}$$

$$\vec{a}(\theta_{1,}\ \theta_2)=R(\theta_2)R(\theta_1)\overrightarrow{a_0}$$

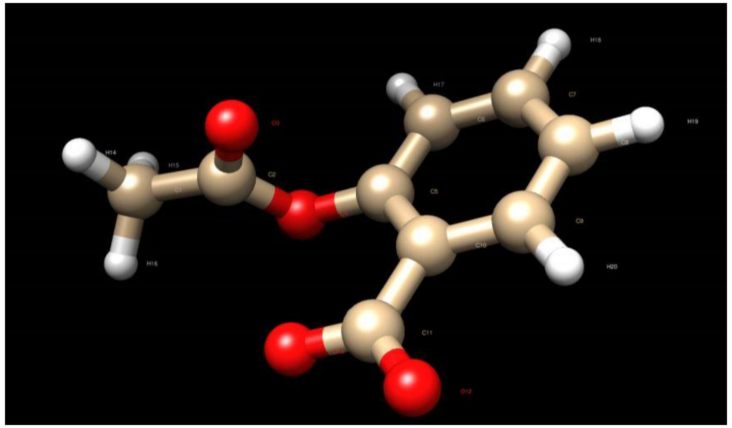
Overview of the problem size (ComplexDB)





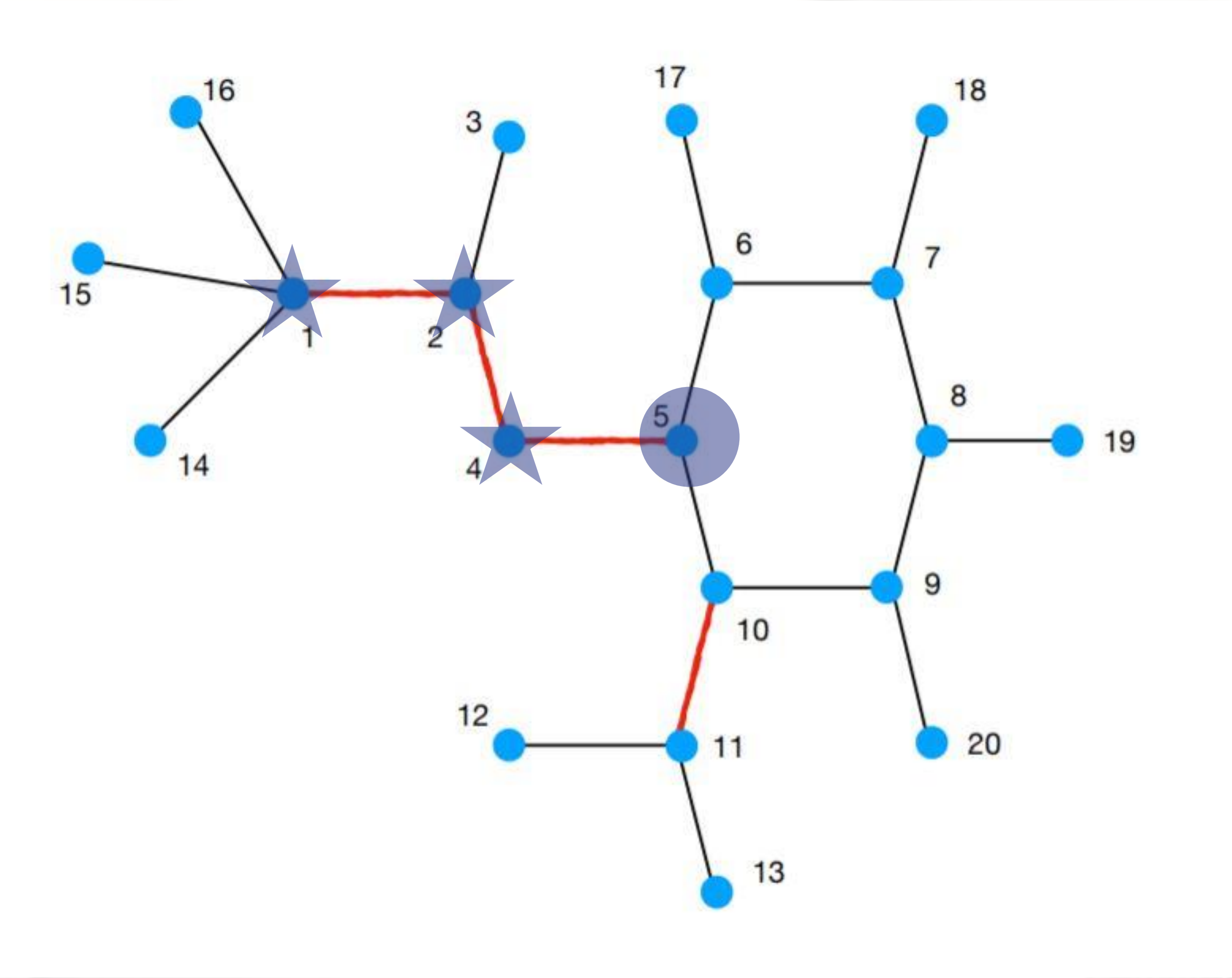


Molecule Unfolding: Rotatable Betweenness Centrality



Betweenness centrality:

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$



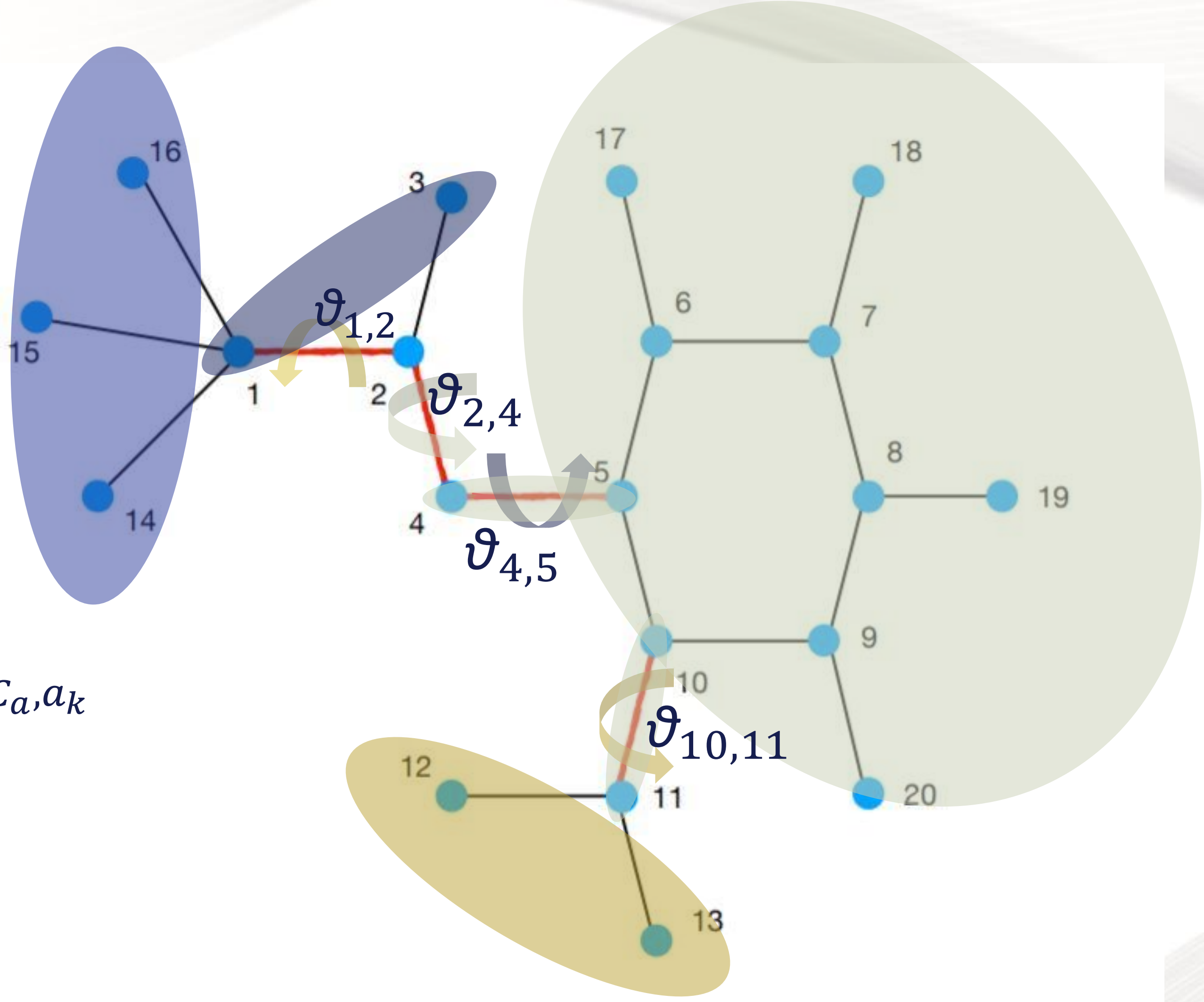
Molecule Unfolding: Rotatables Influence Set

Rotatables Influence set:

$$I_S = E_{C_a,a_k} \cap E_R$$

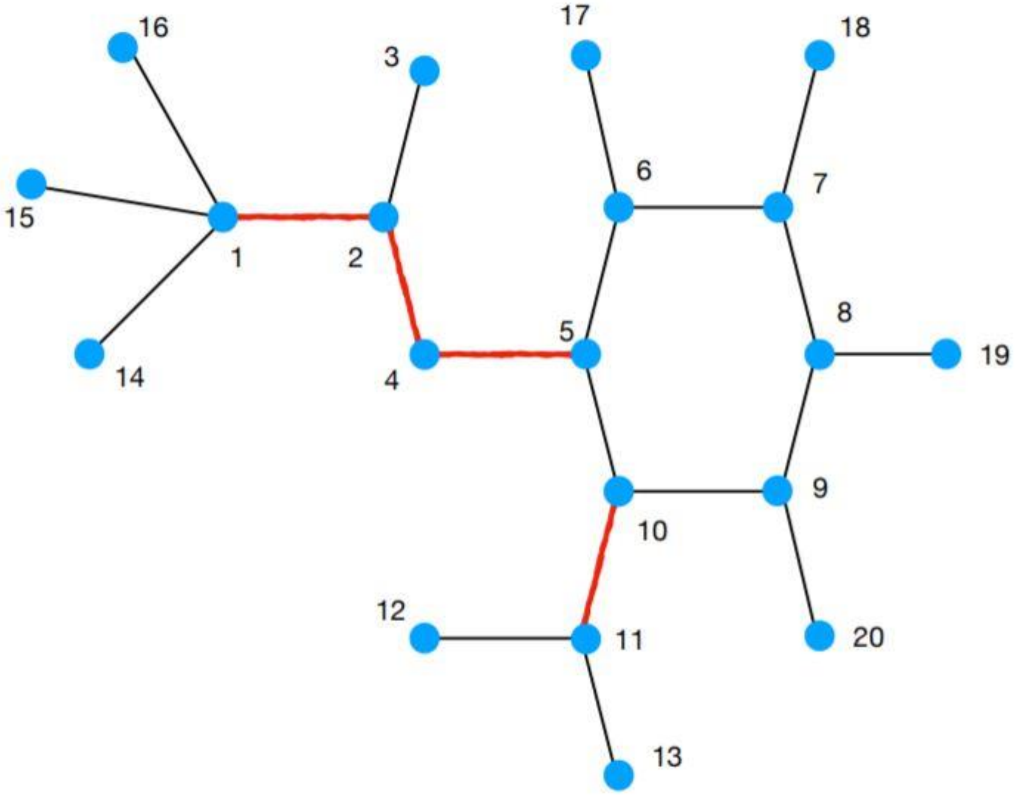
$E_R = Rotable\ bonds;$

$E_{C_a,a_k} = Bonds\ on\ the\ shortest\ path\ \ \sigma_{C_a,a_k}$



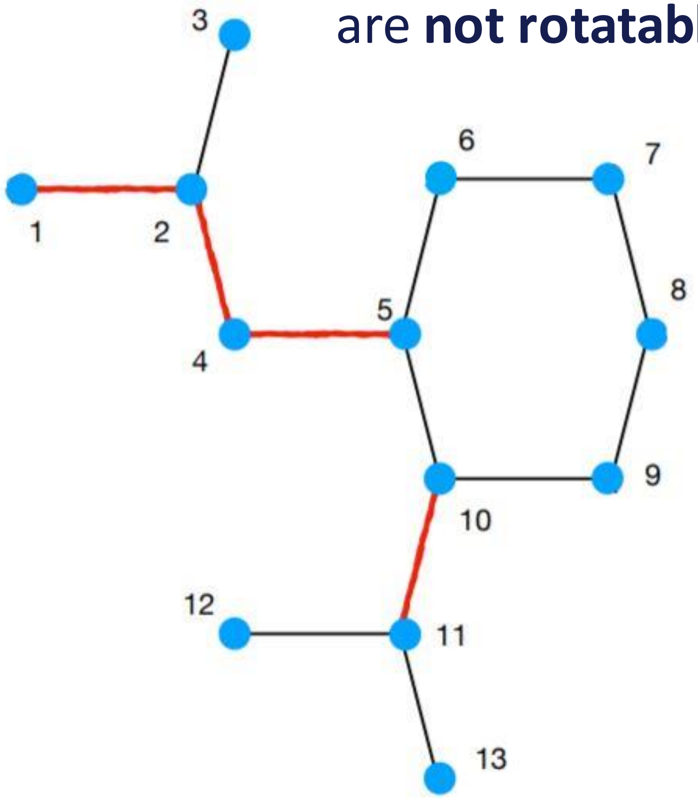
Molecule Unfolding

Original 2D molecule:

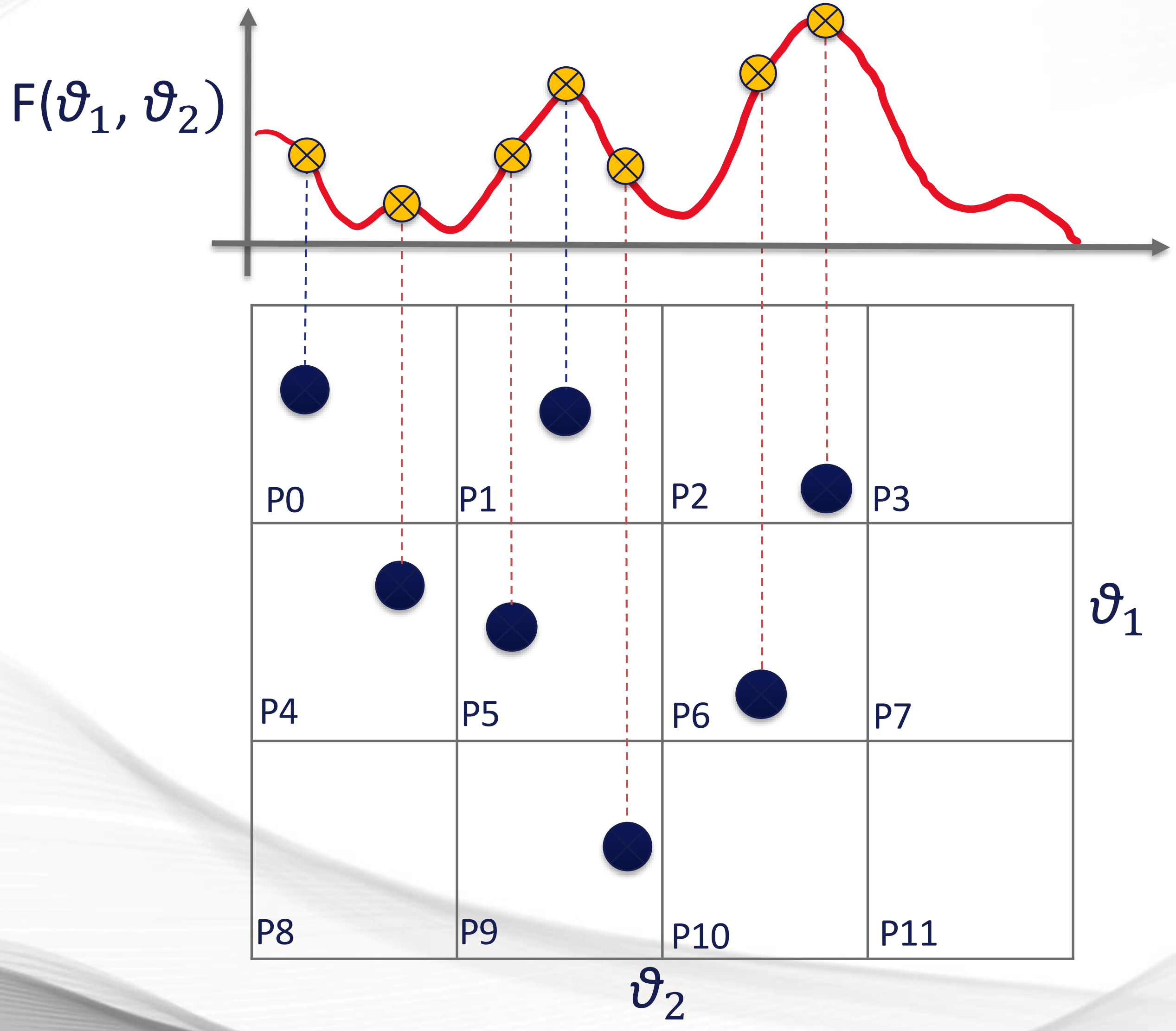


Without Hydrogen:

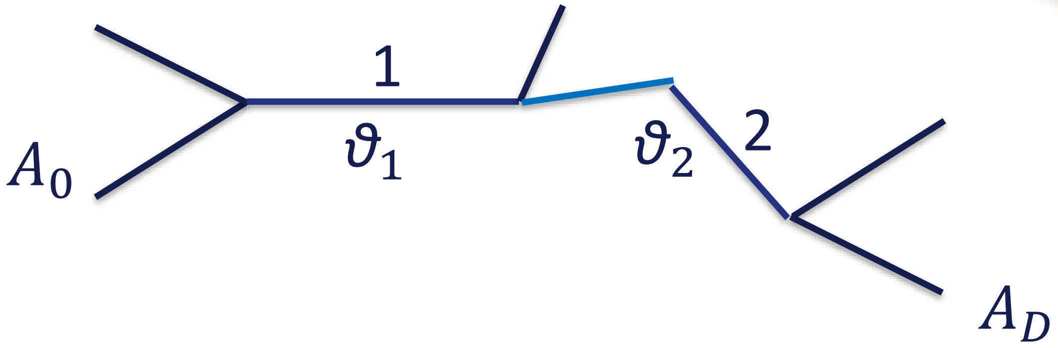
Double and **amide** bonds are **not rotatable**



Molecule Conformation Exploration: Random Search



Random Search

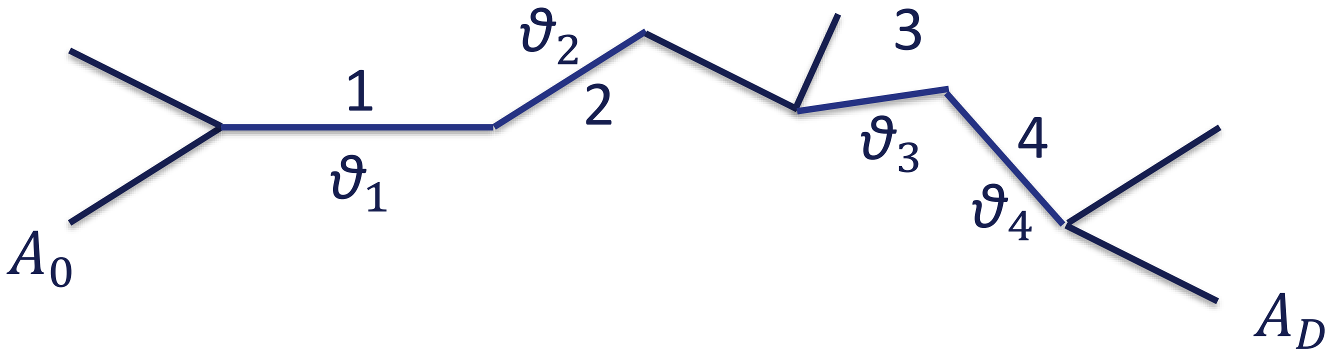


Molecule Conformation Exploration: Greedy

M= Measure total sum of internal distances

ϑ# = physical rotation of torsion# for all possible angles

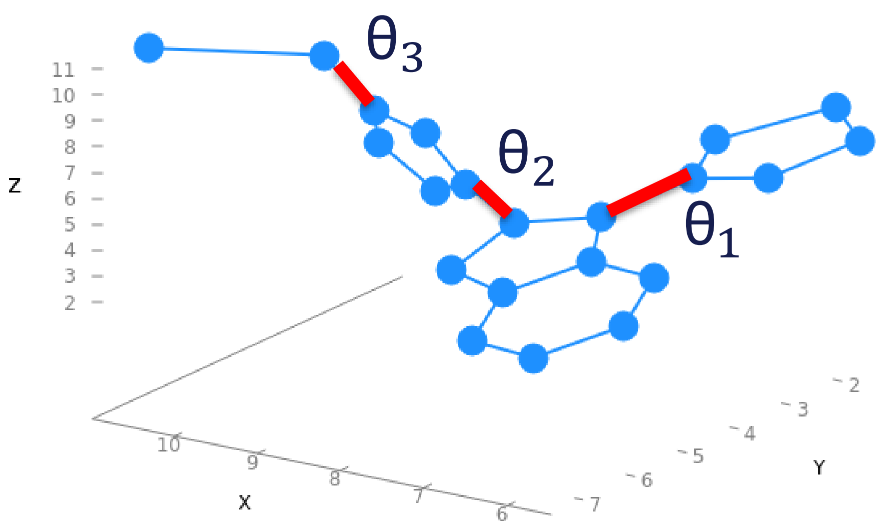
GeoDock-inspired



Greedy:

- **GeoDock** = ϑ1 - M - ϑ 2 - M - ϑ 3 - M - ϑ 4 - M
- **Batch** = i) ϑ 1, ϑ 2 - M - ϑ 3, ϑ 4 - M ii) ϑ 1, ϑ 2 , ϑ 3 - M - ϑ 4 - M iii) ϑ 1, ϑ 2 , ϑ 3 , ϑ 4 - M

Combinatorial Optimization Problem Definition



is convenient to identify a **conformation of a molecule** with M torsions by a **torsion vector**

$$[\theta_1, \ldots, \theta_M]$$

Where each torsion θ_N can assume values in [0, 2π).

Objective: **find** the unfolded **torsion configuration** that maximizes the molecular volume, or equivalently, that **maximizes the distances between fragments**

Combinatorial Optimization Problem Definition

Objective: find the **unfolded torsion configuration**

$$[\theta_1^{unfold}, \ldots, \theta_M^{unfold}]$$

that **maximizes the sum of distances** $D_{ab}(\theta)$ **between fragments** a and b

$$\boxed{D(\theta) = \sum_{a,b} D_{ab}(\theta)^2}$$

where $D_{ab}(\theta)^2 = ||\vec{a}_0 - R(\theta)\vec{b}_0||^2$

Consider a **discretization of the torsion angle** θ_i **into** d **possible values**

$$\theta_i = [\theta_i^1, \theta_i^2, \theta_i^3, ..., \theta_i^d]$$

And introduce a **binary variable** $\boldsymbol{x_{ik}}$ with $1 \leq k \leq d$, such that

$$x_{ik} = \begin{cases} 1 & \text{if } \theta_i = \theta_i^k; \\ 0 & \text{otherwise.} \end{cases}$$

with the constraint

$$\sum_{k=1}^{d} x_{ik} = 1$$

Constructing the Binary Optimization problem

This also induces a **discretization of the sine and cosine** for each torsion

$$\sin(\theta_i) = \sum_{k=1}^{d} \sin(\theta_i^k)\ x_{ik} \qquad \cos(\theta_i) = \sum_{k=1}^{d} \cos(\theta_i^k)\ x_{ik}$$

With such encoding, the **rotation matrix** $\boldsymbol{R(\theta_i)}$ associated **the torsion angle** θ_i becomes **a function of** all the **binary variables** $\boldsymbol{x_{ik}}$ needed to represent the angle θ_i

$$R(\theta_i) = R(x_{i1}, x_{i2}, \ldots, x_{id})$$

Constructing the High-order Unconstrained Binary Optimization (HUBO) problem

The **general form of the HUBO** optimization function is

$$O(x_{ik}) = A\sum_i \left(\sum_{k=1}^{d} x_{ik} - 1\right)^2 - \sum_{a,b} D_{ab}(\theta)^2$$

where the pairwise **distances are expressed using the binary variables**

$$D_{ab}(\theta)^2 = ||\vec{a}_0 - R(\theta)\vec{b}_0||^2$$

In general, **if $\boldsymbol{D_{ab}}$(θ) depends on *m* torsions**, D_{ab}(θ) contains terms up to the *m*-th order, hence the **highest order in the HUBO is *2m***

HUBO Problem Structure

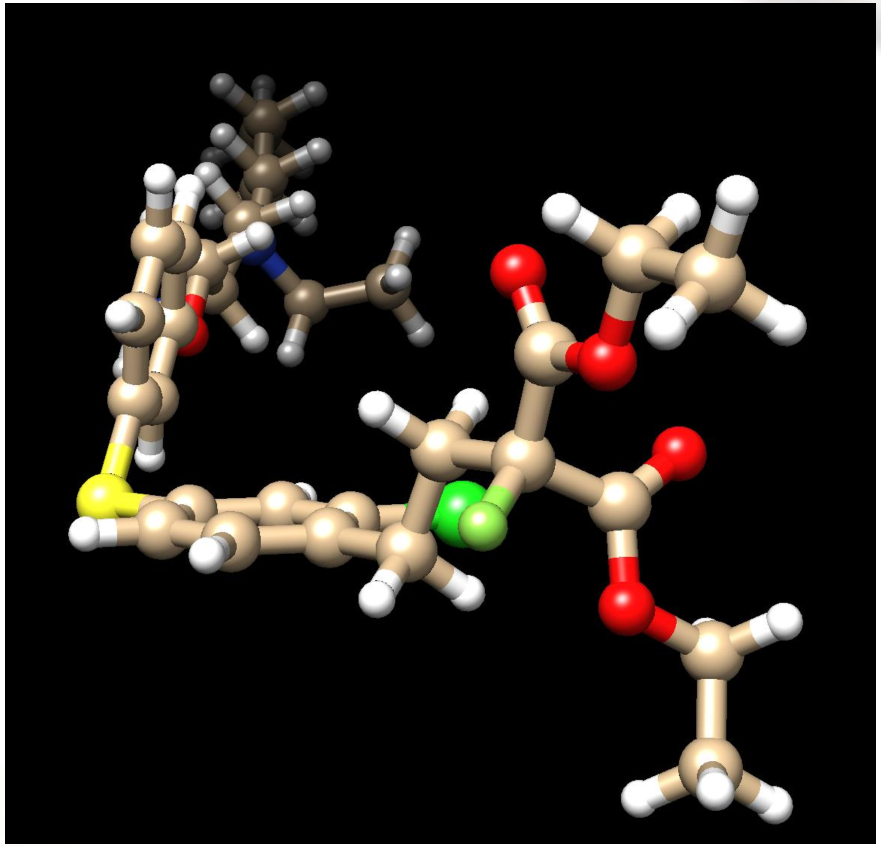
In order to obtain a **precision of $\Delta\theta_i$**, the **number of variables** needed **for each torsion** is

$$d = \frac{2\pi}{\Delta\theta_i} = \frac{2\pi}{\theta_i^{k+1} - \theta_i^k}$$

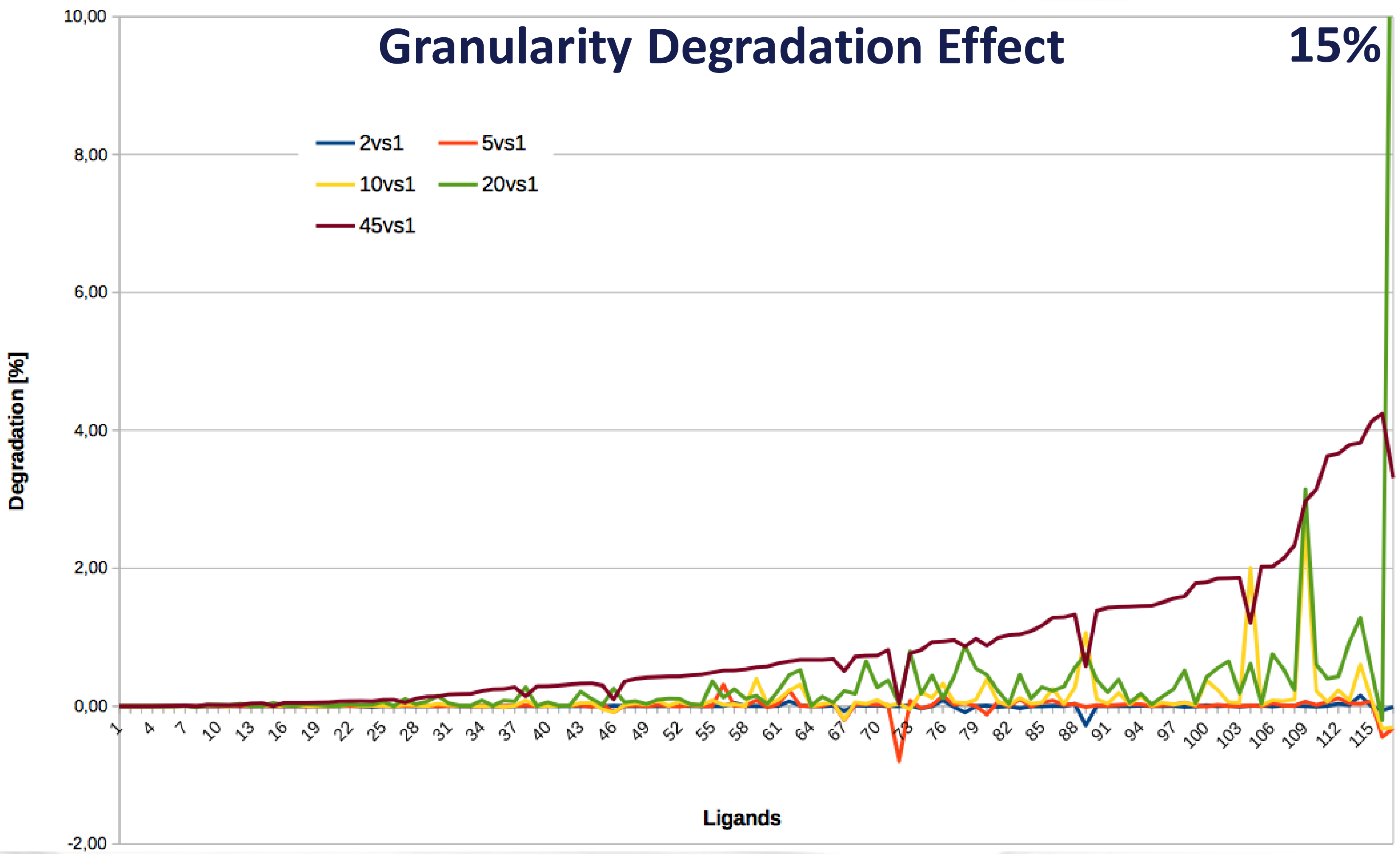
Given a molecule with ***M*** **torsions**, the total **number of binary variables x_{ik}** in the HUBO

$$n = d \times M = \frac{2\pi}{\Delta\theta_i} \times M$$

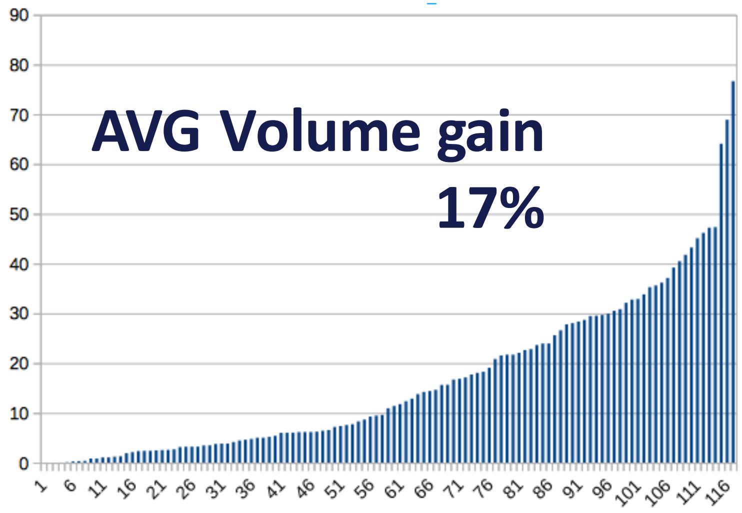
Molecules: 20 to 50 atoms - 10 torsions



Angle subsampling effect on the unfolding degradation



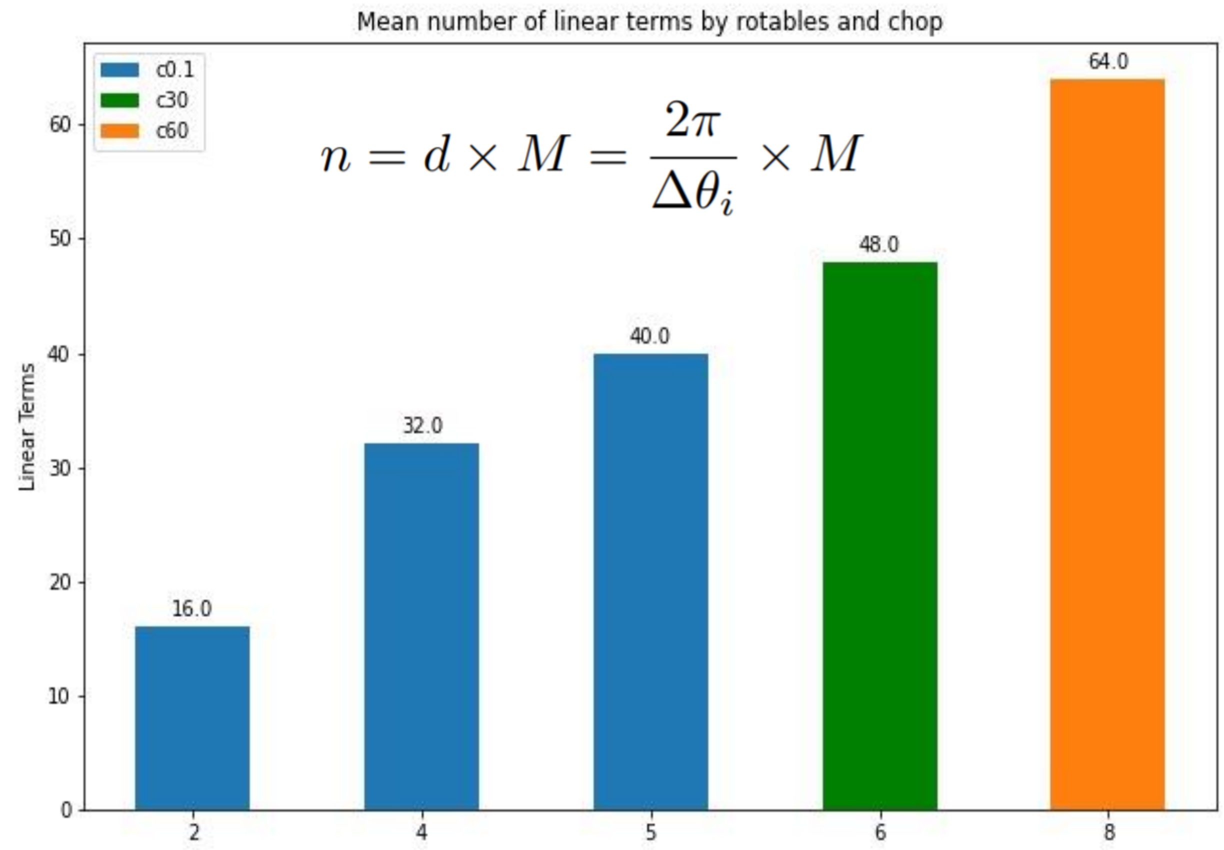
Unfolded vs Initial



AVG Volume gain
17%

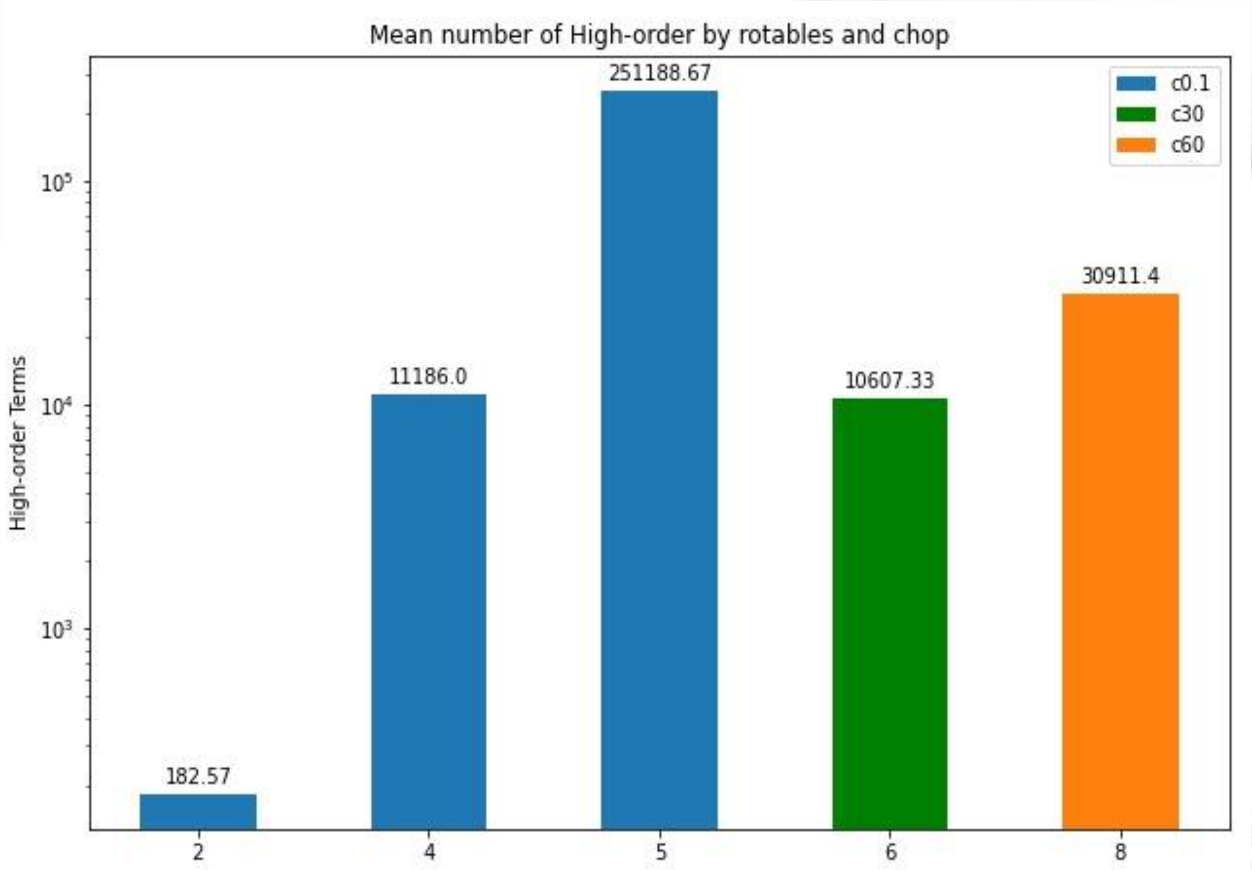
HUBO Problem Structure at $\Delta\theta_i = \pi/4$

HUBO linear terms



Number of torsions

HUBO high order terms (Log-scale)



Number of torsions

HUBO Problem Approximation

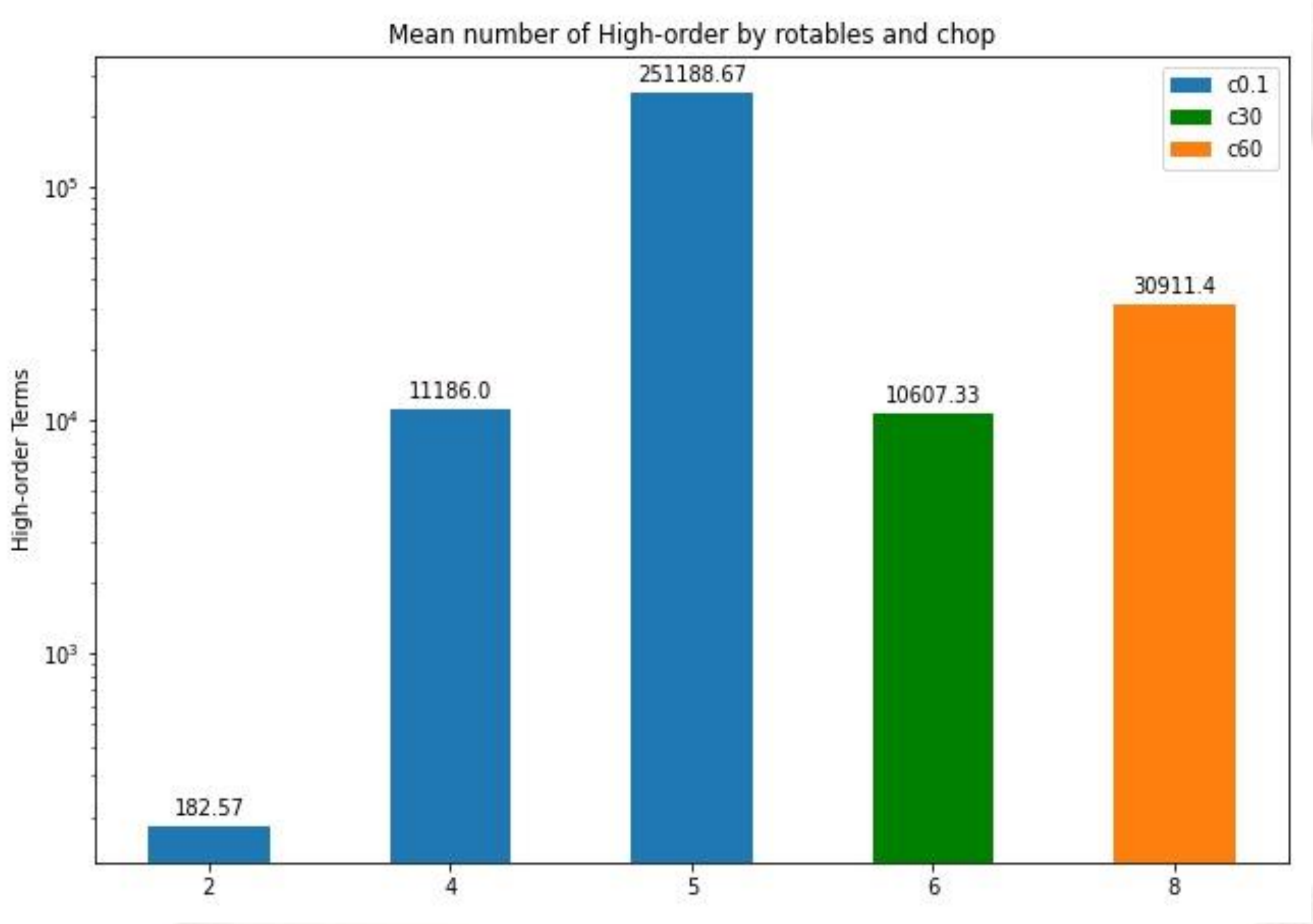
Delete HUBO terms below a certain **threshold.** Applied in **two phases:**

1. **Speed up** the **construction** of the **HUBOs;**

2. **Speed up** the transformation of **HUBOs into QUBOs** (done via dimod.make_quadratic);

Approximated HUBO problems **solvable** with ***DW2000Q*** and ***Advantage***

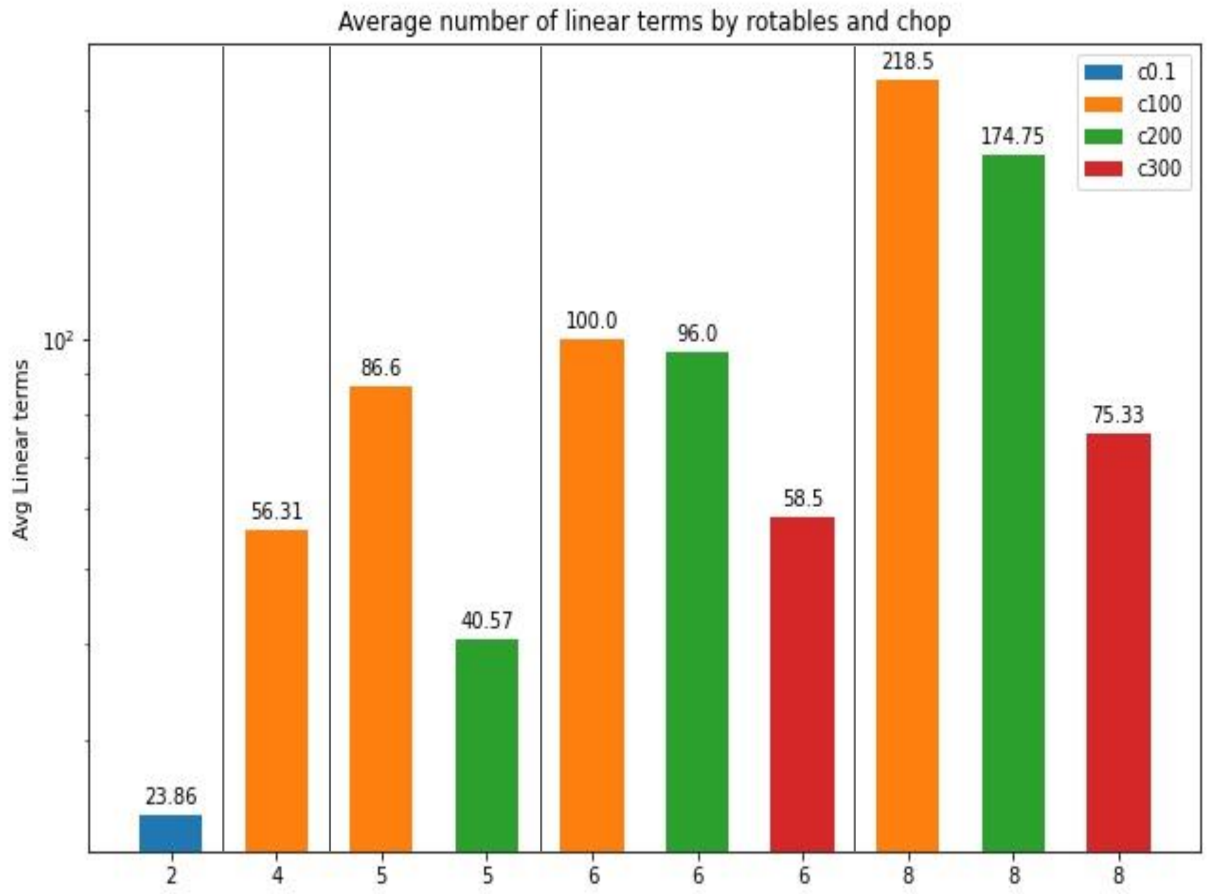
HUBO high order terms (Log-scale)



Number of torsions

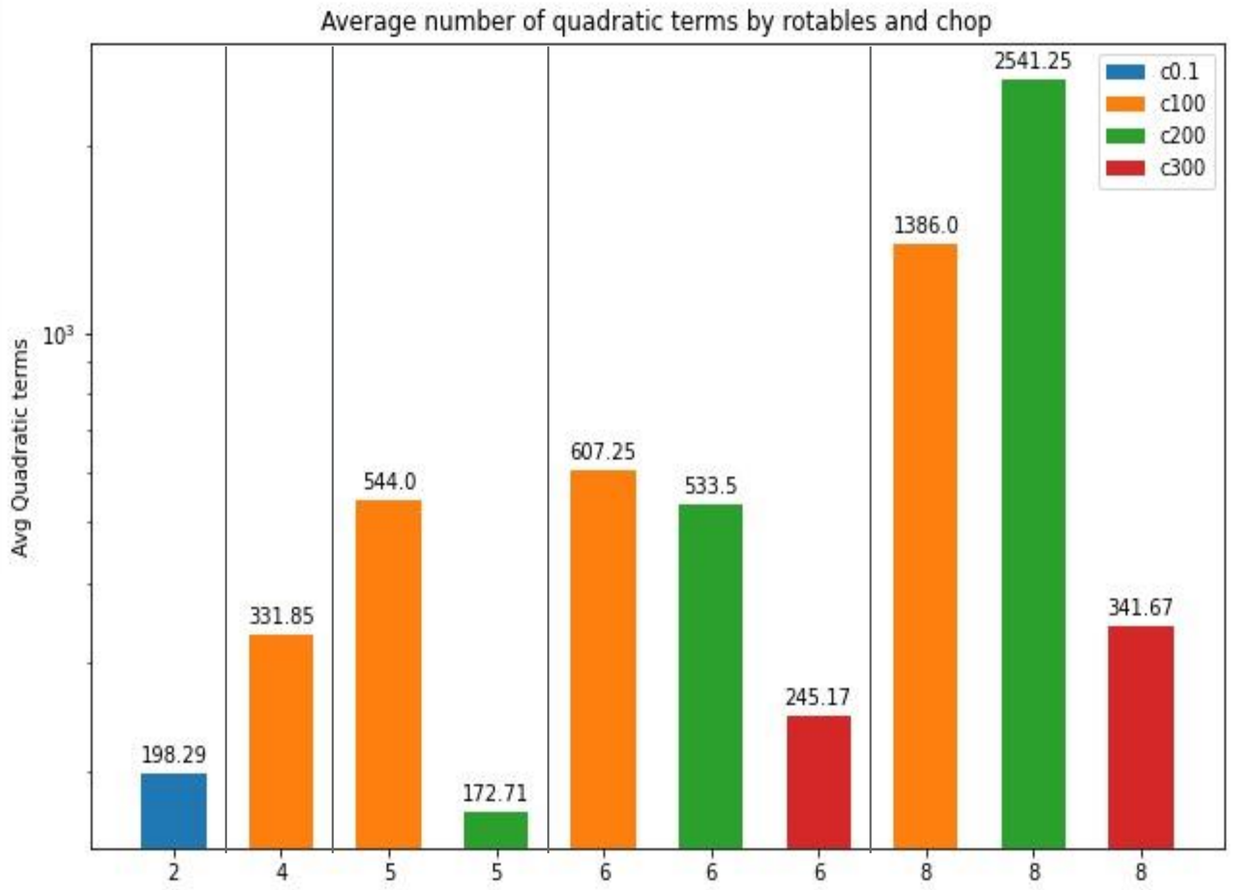
Form HUBOs to QUBOs

QUBO linear terms



Number of torsions

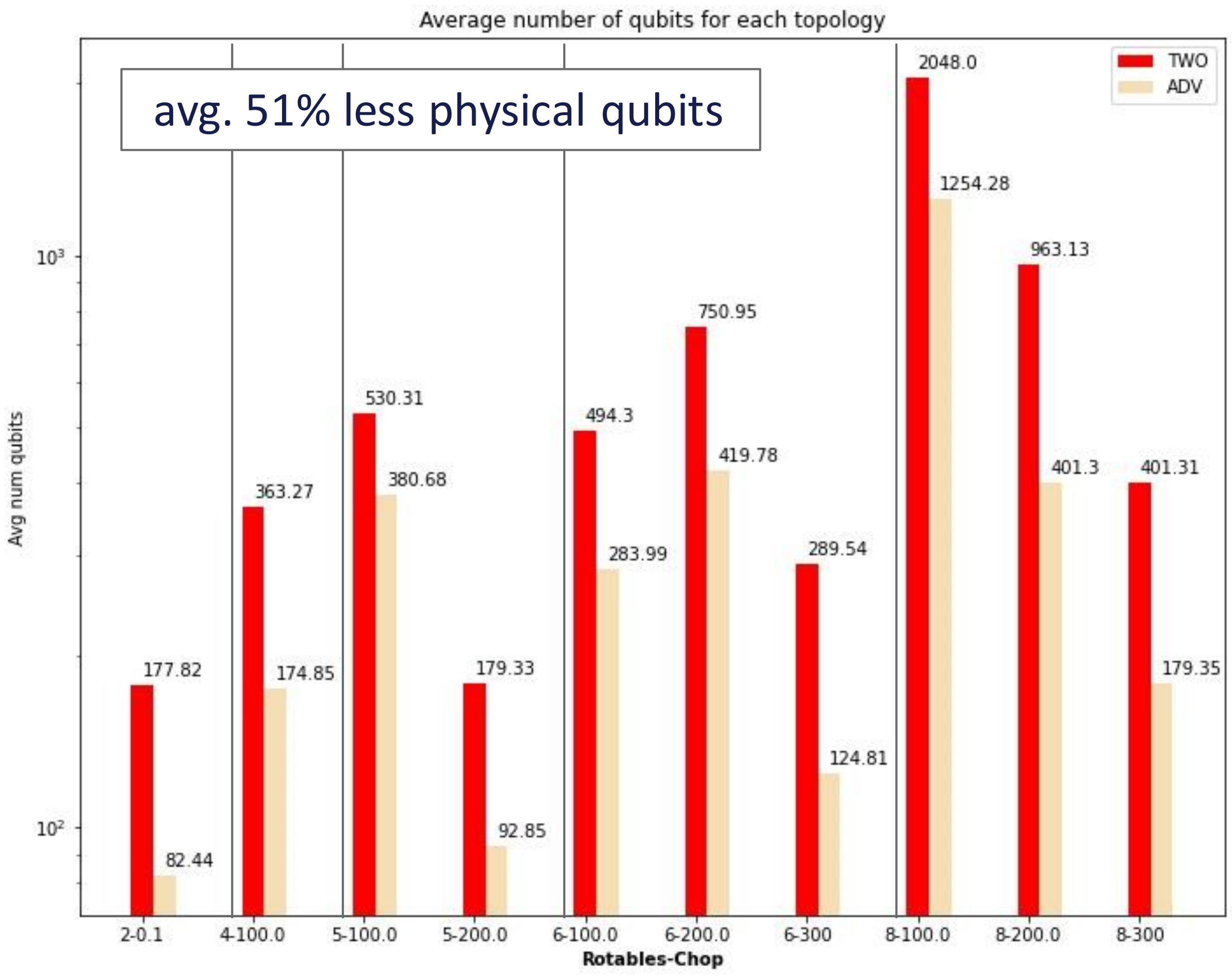
QUBO quadratic terms



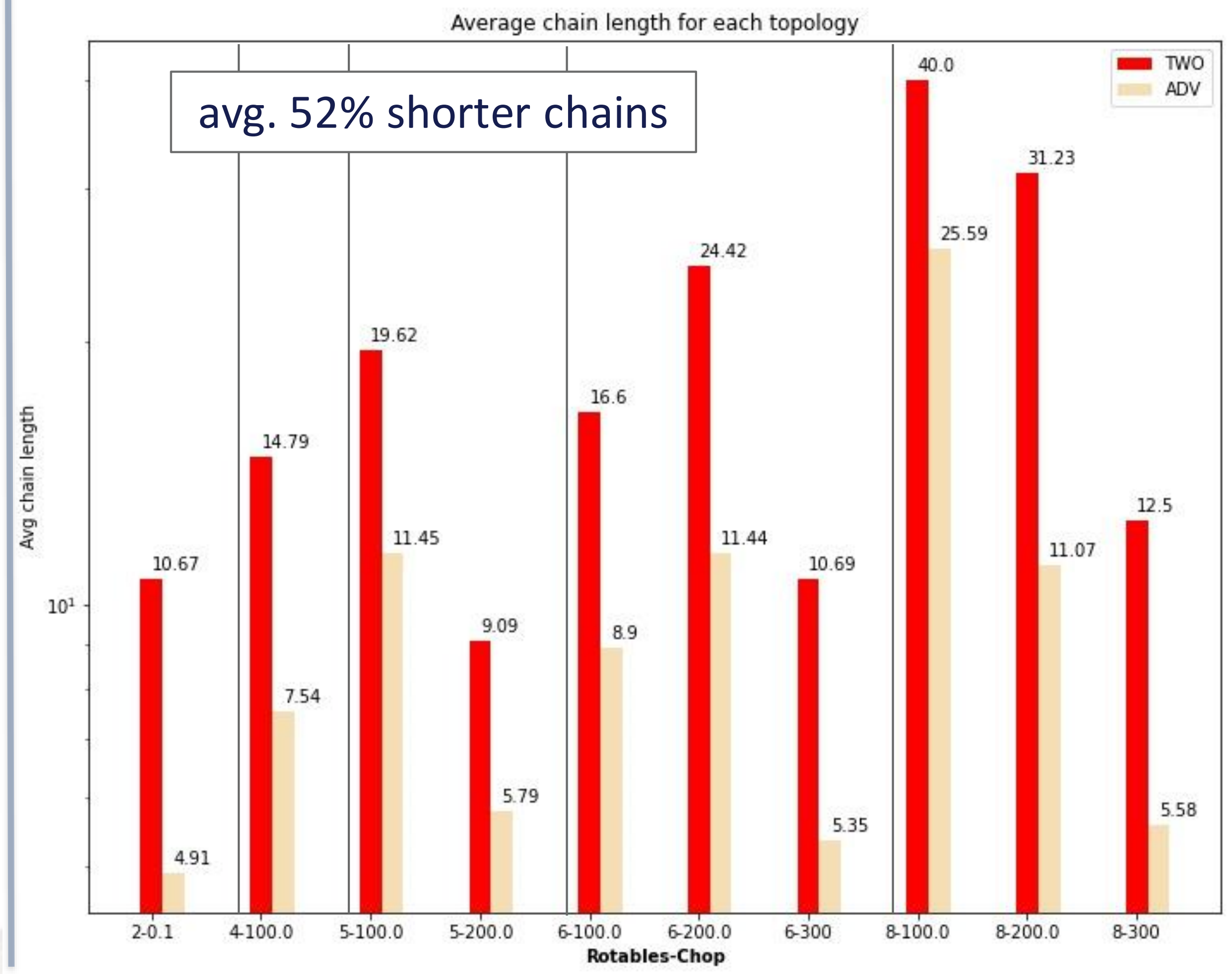
Number of torsions

Embeddings DW2000Q & Advantage

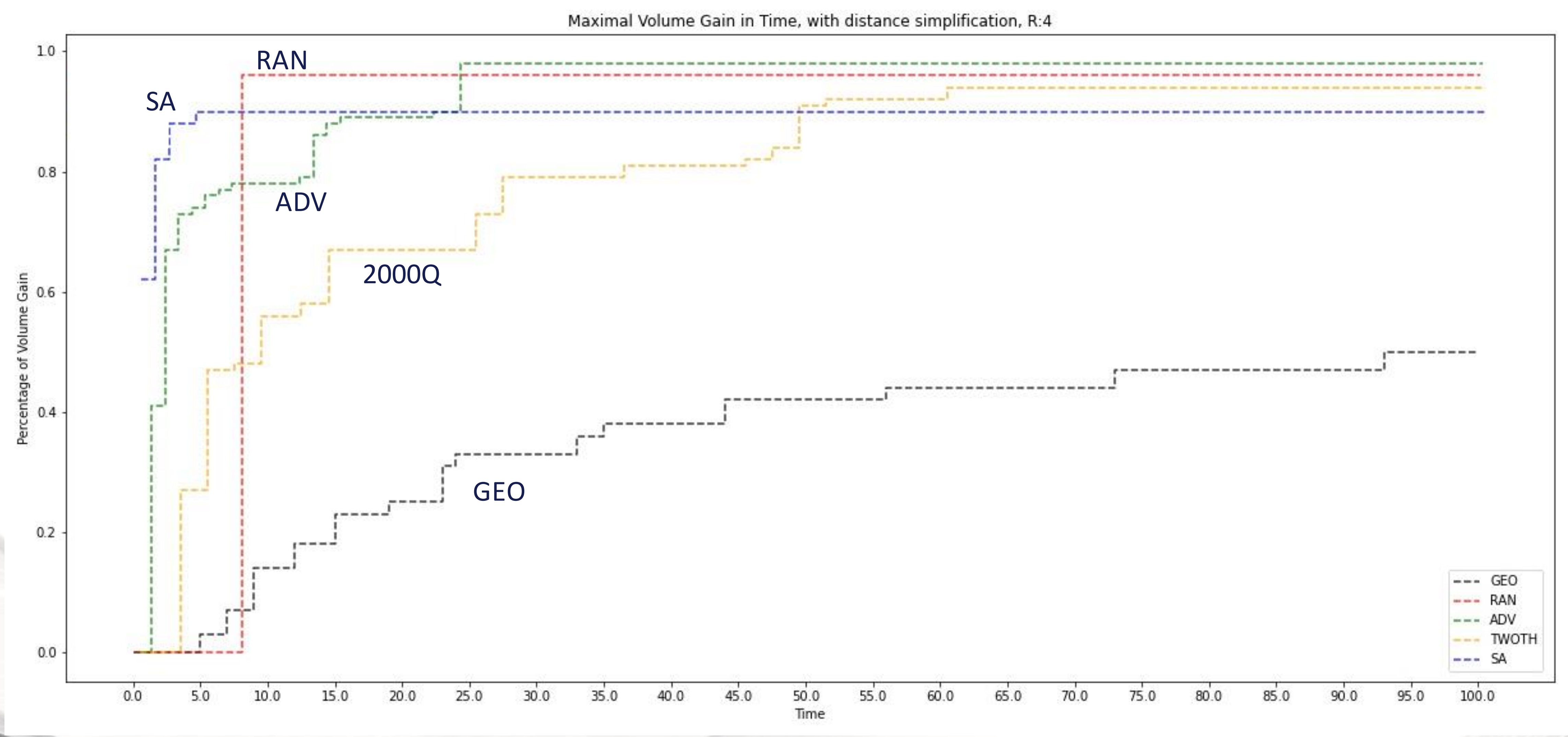
Number of physical qubits



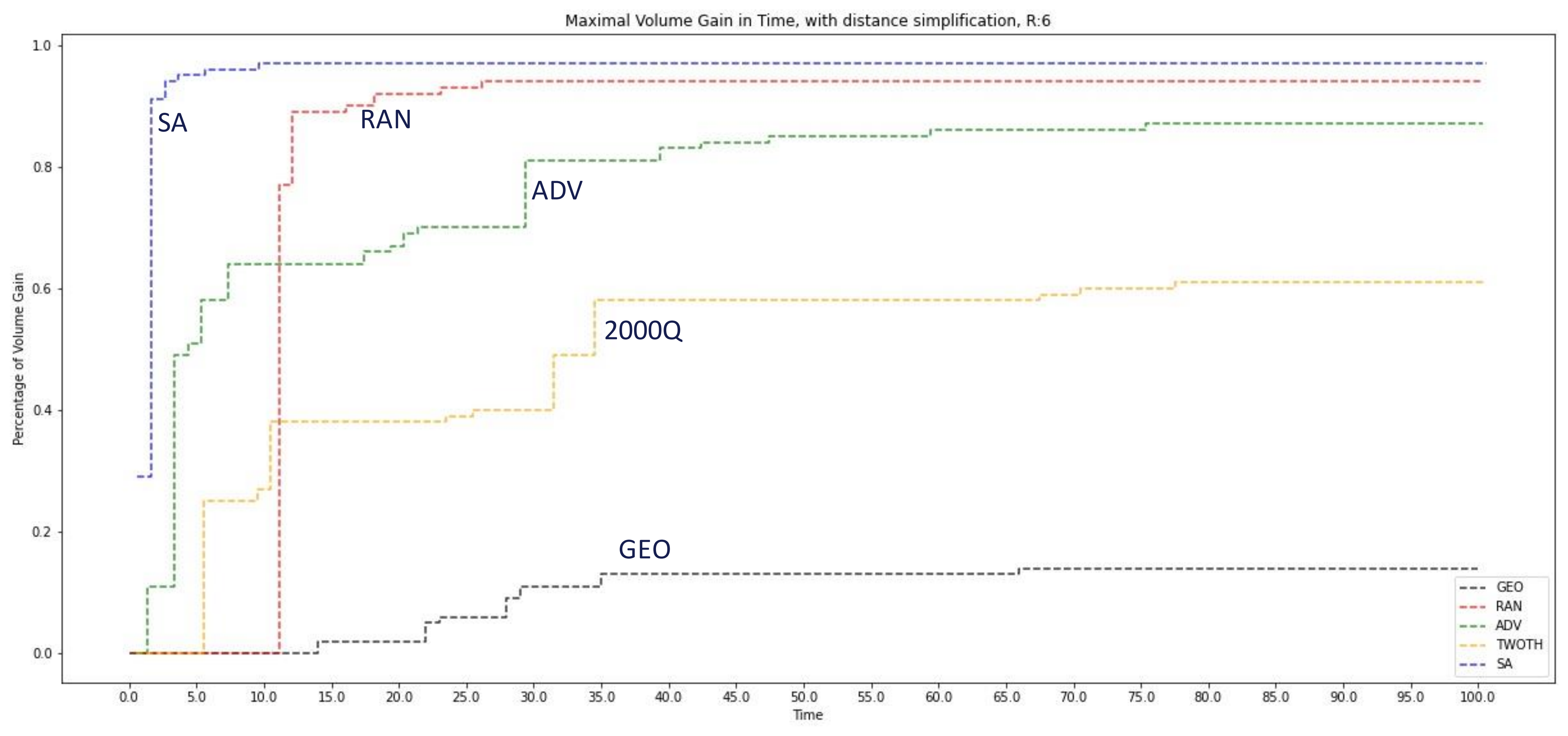
AVG chain length



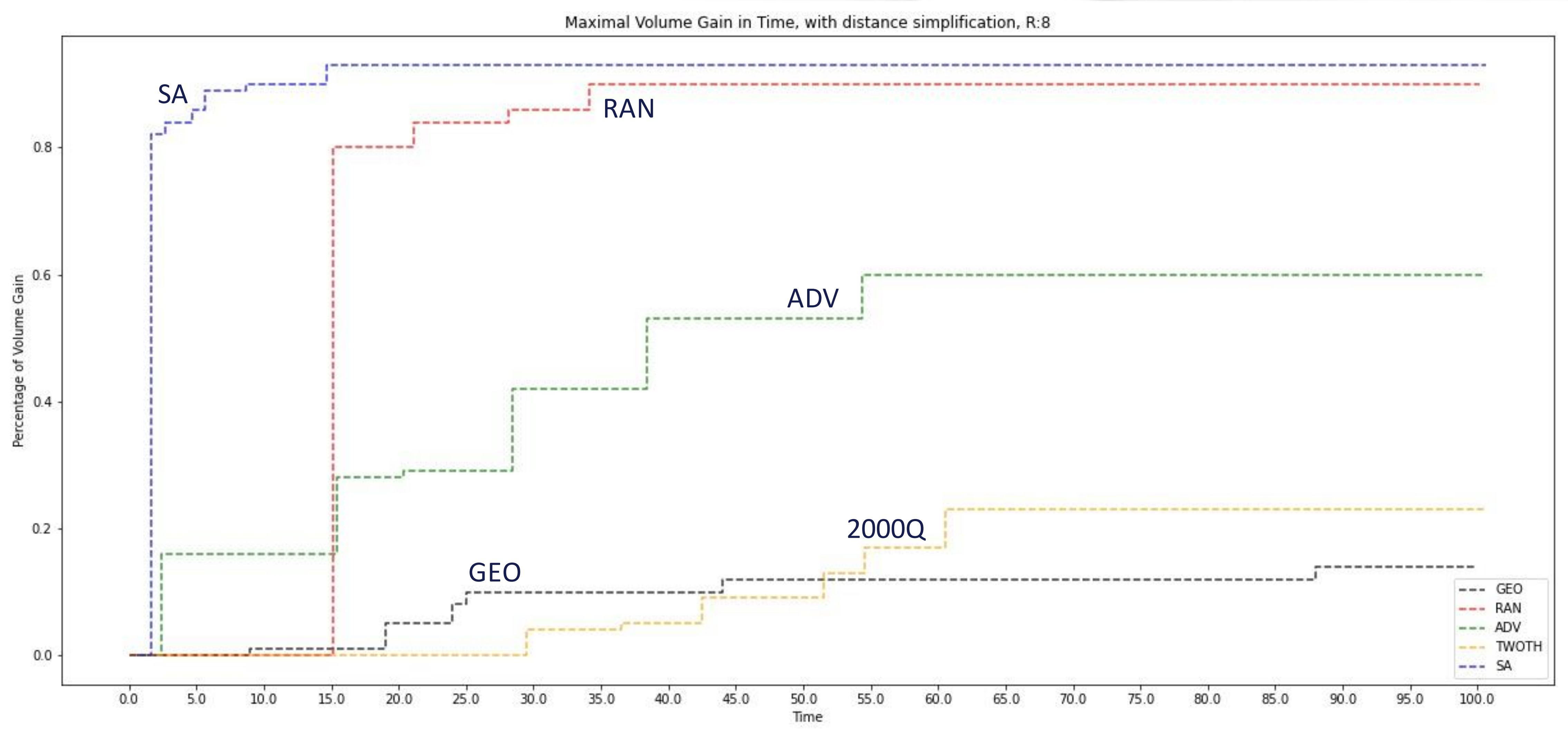
Results, 4 Torsions : Volume Gain in Time (seconds)



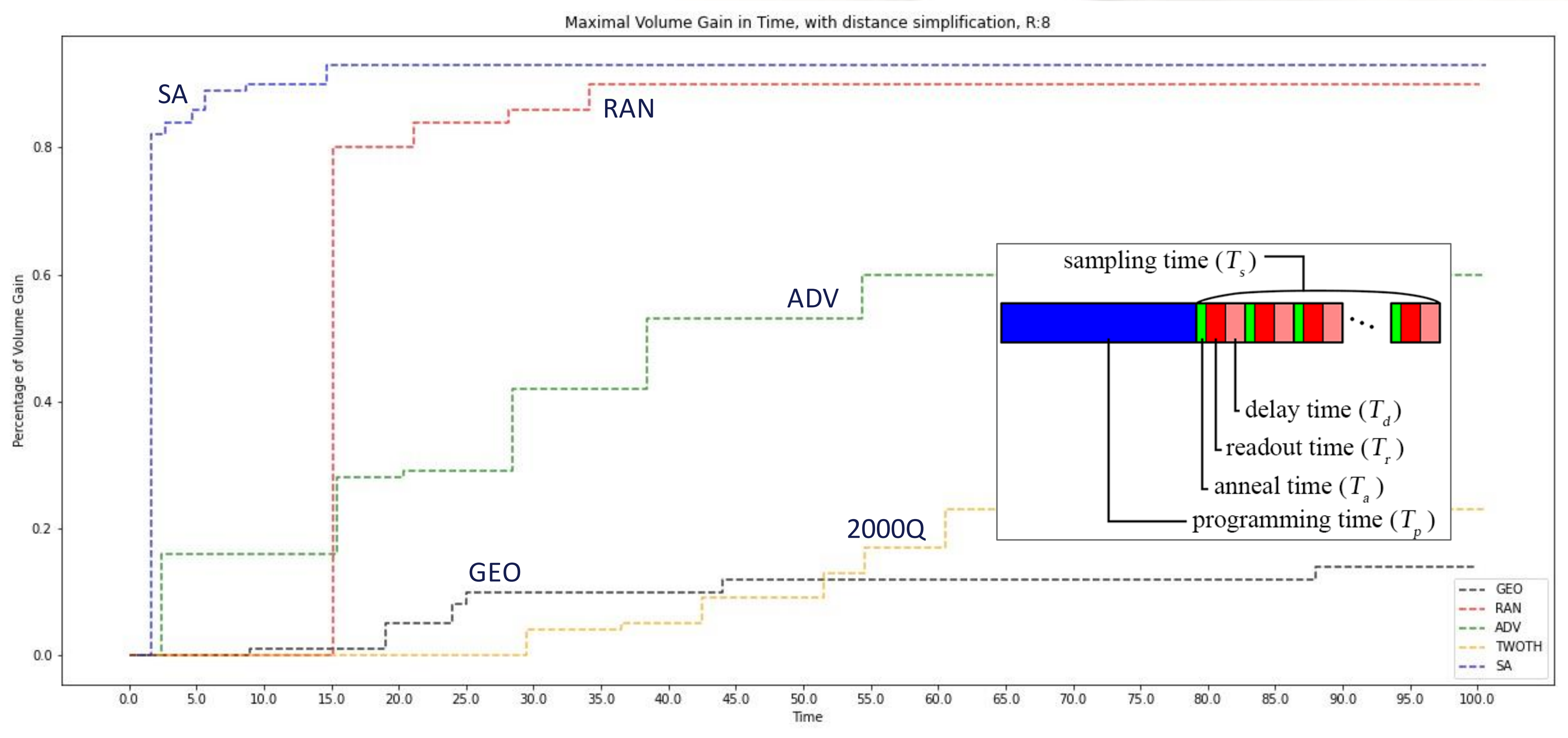
Results, 6 Torsions : Volume Gain in Time (seconds)



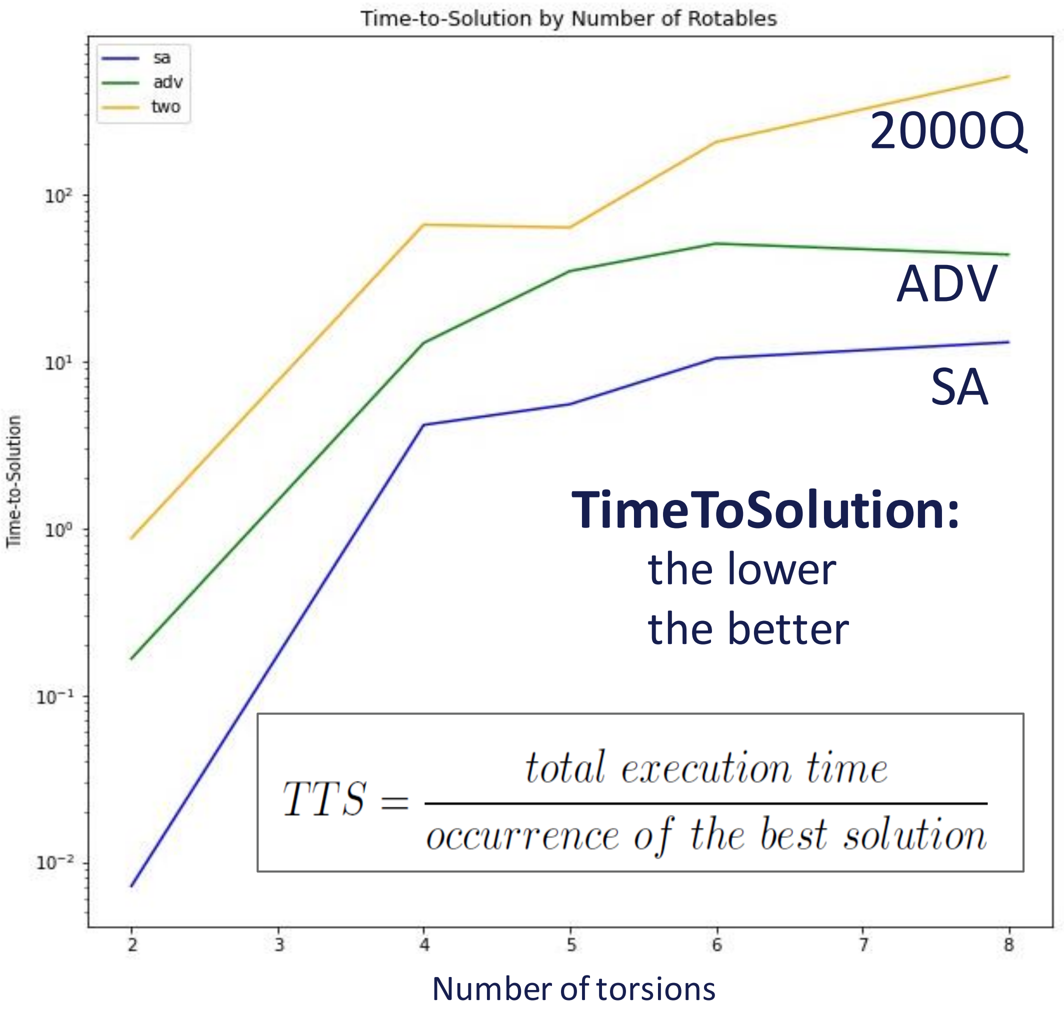
Results, 8 Torsions : Volume Gain in Time (seconds)

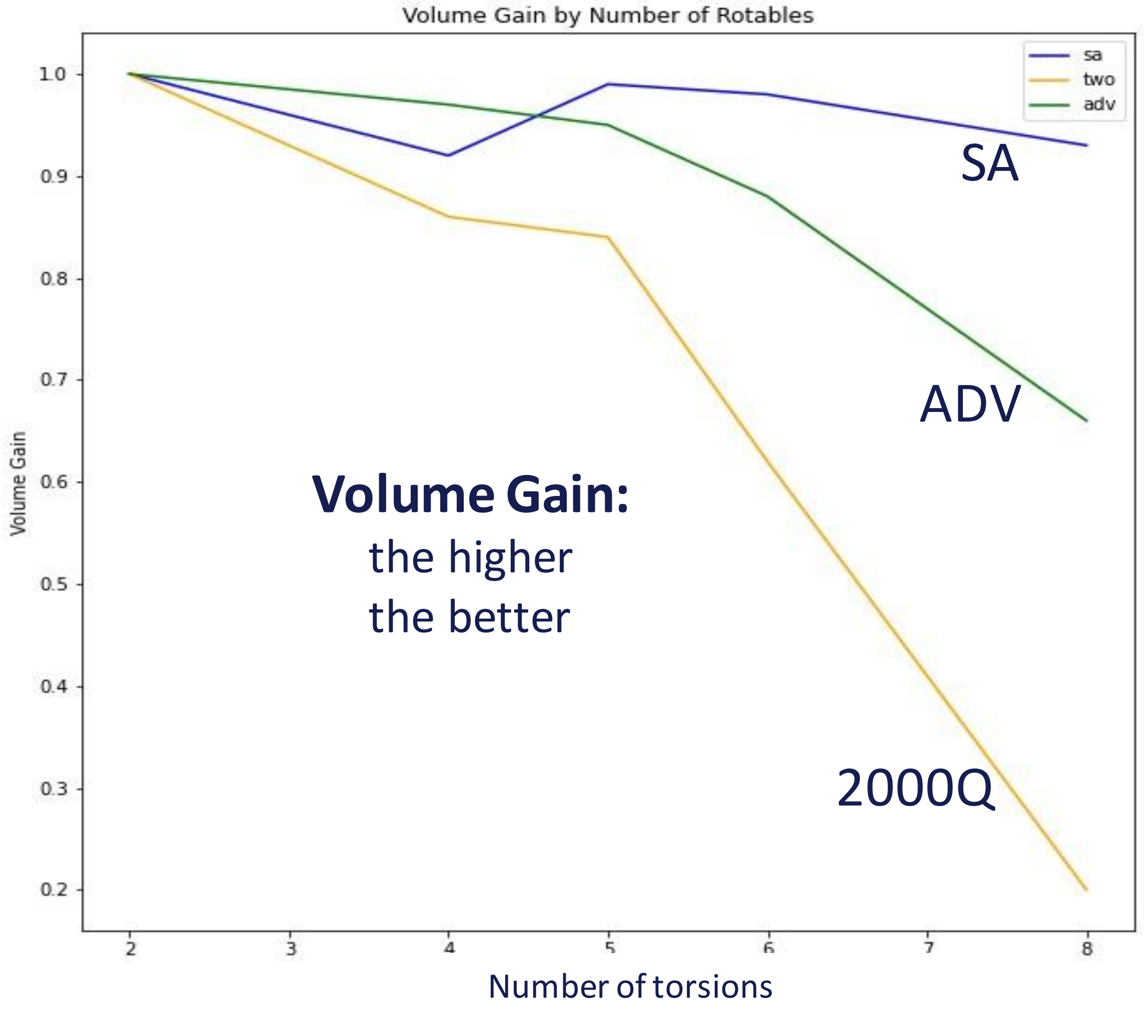


Results, 8 Torsions : Volume Gain in Time (seconds)

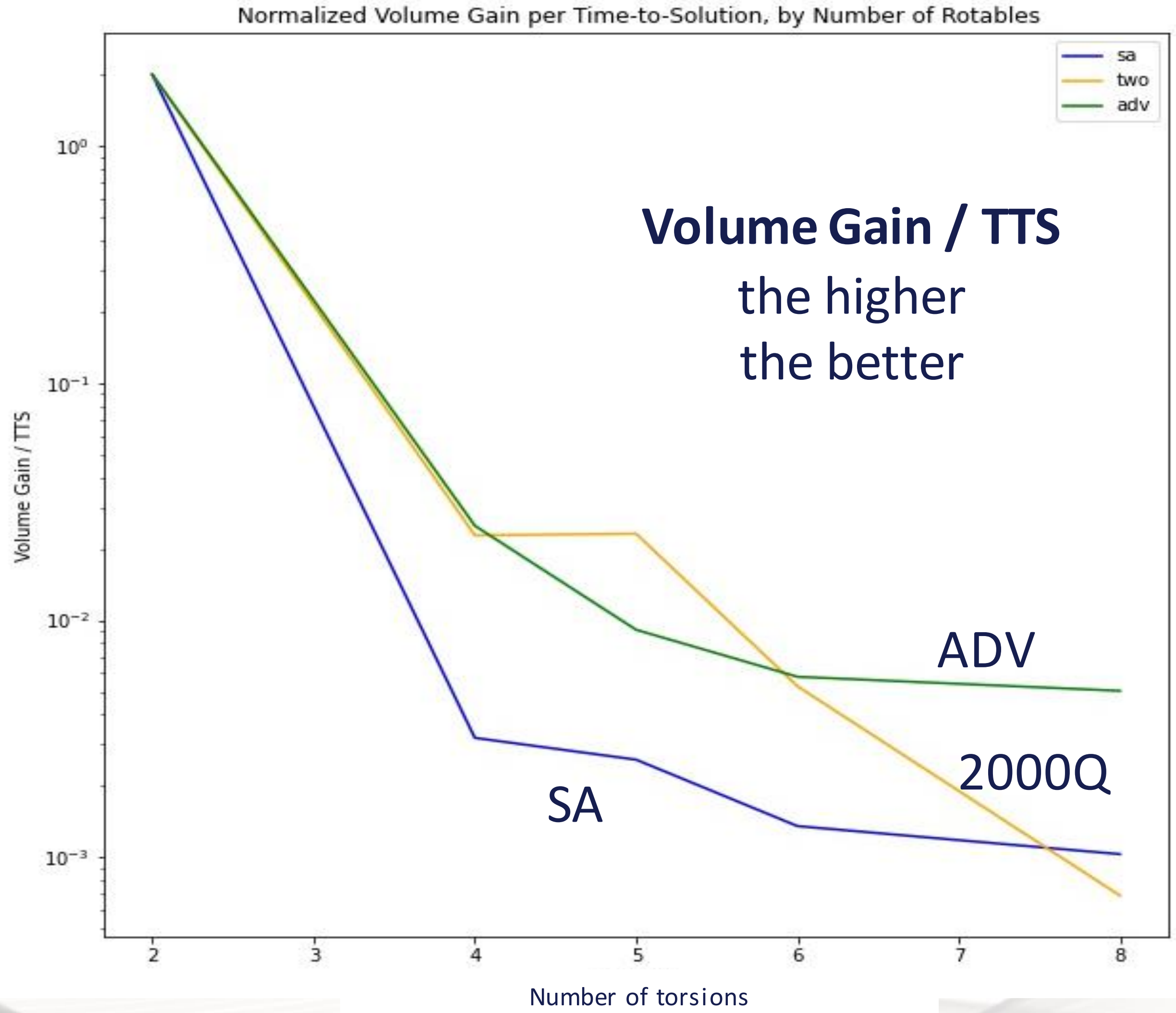


Results: Time To Solution (TTS) & Volume Gain





Results: Normalized Volume Gain per TTS



Normalized Volume Gain per TTS:

- Takes into account **both quality** of solution and **TTS**
- Measures **how fast** the method **fails to provide good solutions**
- ***Advantage*** has **lower avg. slope** with respect to **SA** and **DW2000Q**

sa	two	adv
1.609948	6.0189816	1.36206524

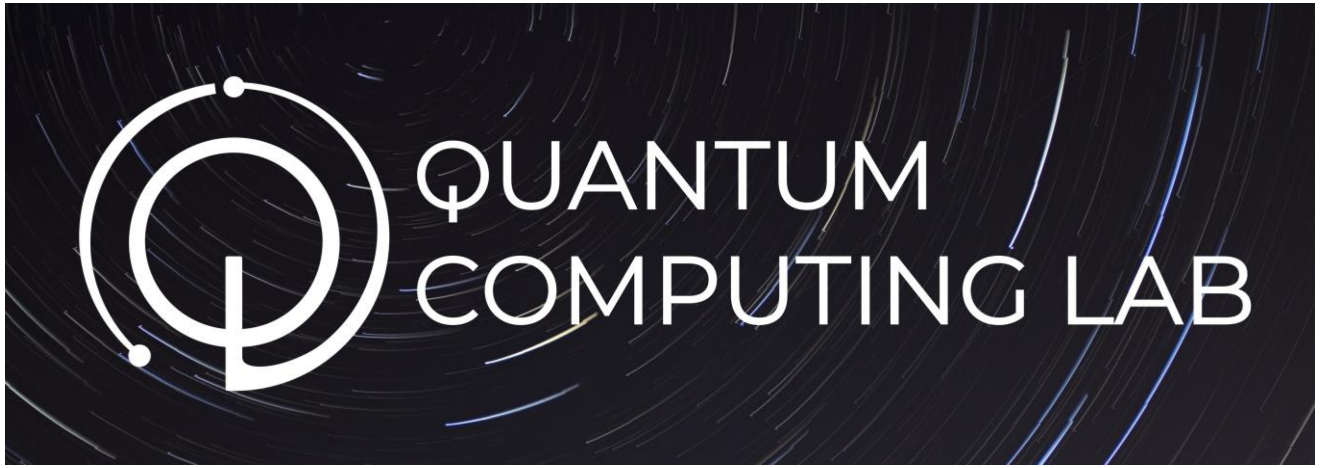
Conclusions

- We tackled the problem of **Molecular Unfolding,** an important step in molecular docking.
- New **HUBO formulation** that can be solved on **D-WAVE annealers** has been developed.
- We have observed that by **increasing** the **approximation threshold with** the **problem size**, it is **possible to embed** formulations that couldn't be otherwise.
- **Embedding** our problems on **Advantage**, compared to the **DW2000Q**, cost **51% less** in terms of **physical qubits** and with **chains 52% shorter**.
- In terms of **absolute time (seconds), SA** is the **fastest method** to provide **close to optimal solutions**.
- **Advantage** significantly **outperforms DW2000Q** in terms of **TTS** and **VolumeGain** by increasing torsions. **Advantage** also show **a better NormalizedVolumeGain/TTS scaling w.r.t. SA**

Quantum Computing @ CINECA

CINECA Quantum Computing Lab:

- Collaborate with Universities, Industries and QC startups
- Internship programs, Courses and Conference (HPCQC)



https://www.quantumcomputinglab.cineca.it

r.mengoni@cineca.it

d.ottaviani@cineca.it