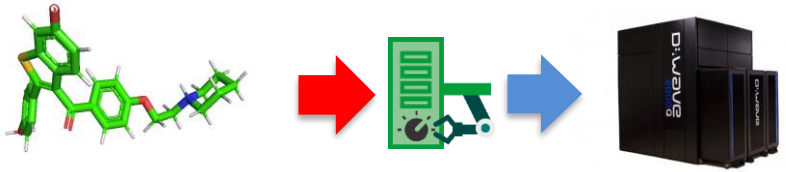
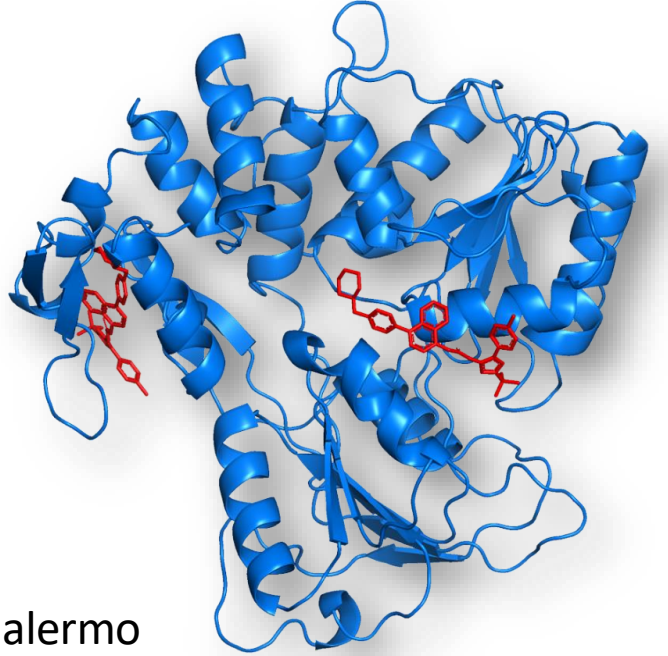




Riccardo Mengoni, Kevin Mato, Daniele Ottaviani, Gianluca Palermo

Molecule Unfolding with Quantum Annealing

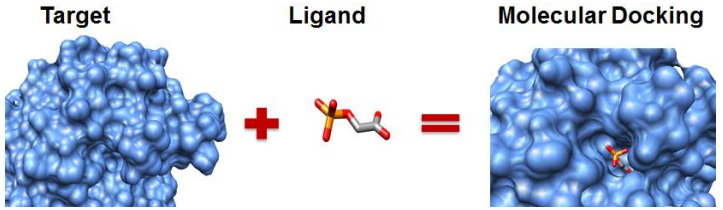
Intro to Quantum Computing

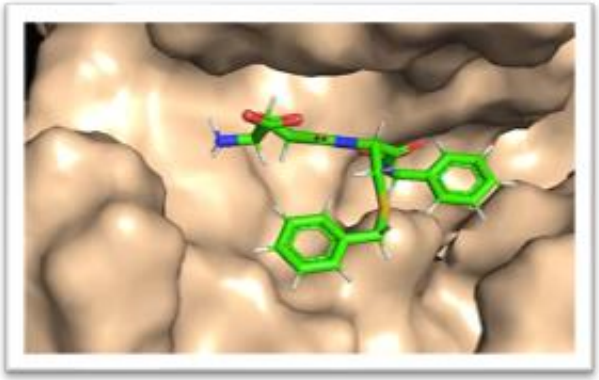
25 June 2021

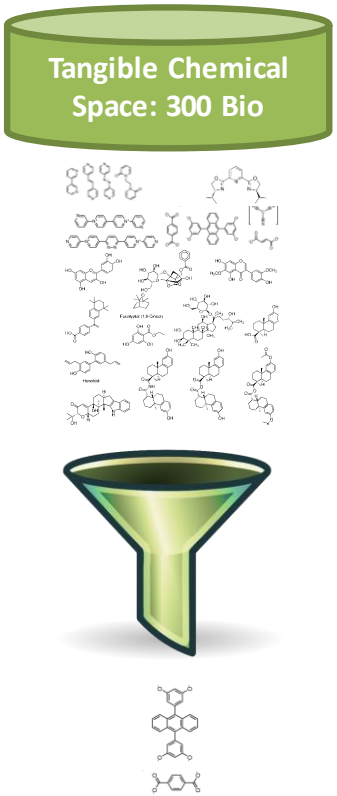
Molecular Docking for Virtual Screening

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

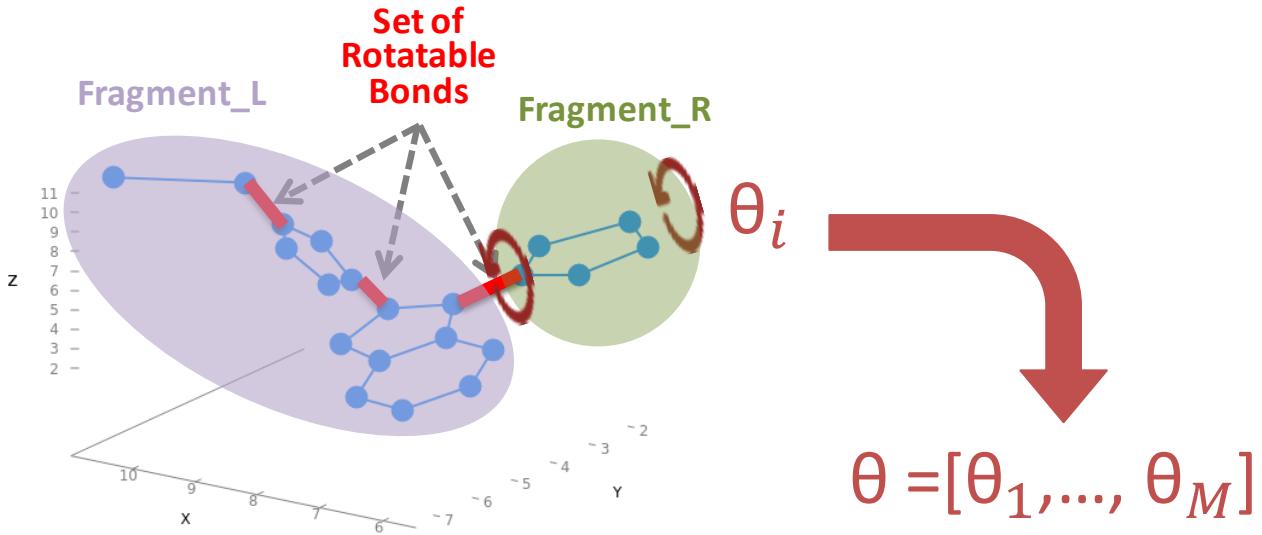
- Ligand main fragments decomposition
- Ligand initial poses Identification
- Placement of the ligand into the pocket with rigid roto-translations

Shape Refinement

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

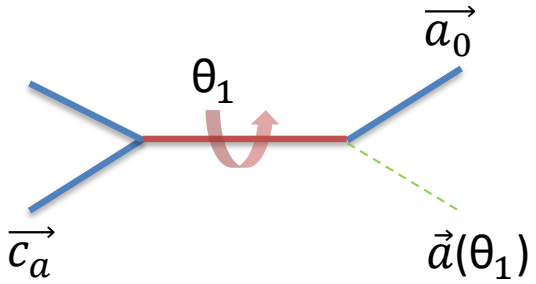
Problem Definition

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

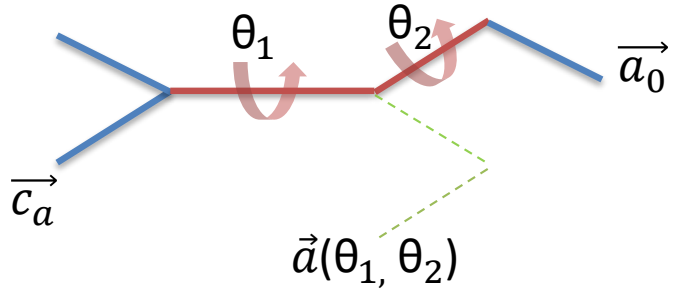


Problem Definition

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



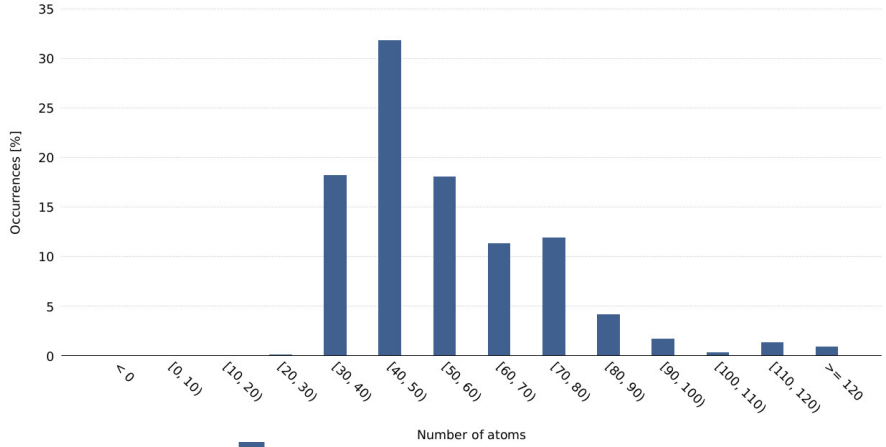
$$\vec{a}(\theta_1) = R(\theta_1)\overrightarrow{a_0}$$

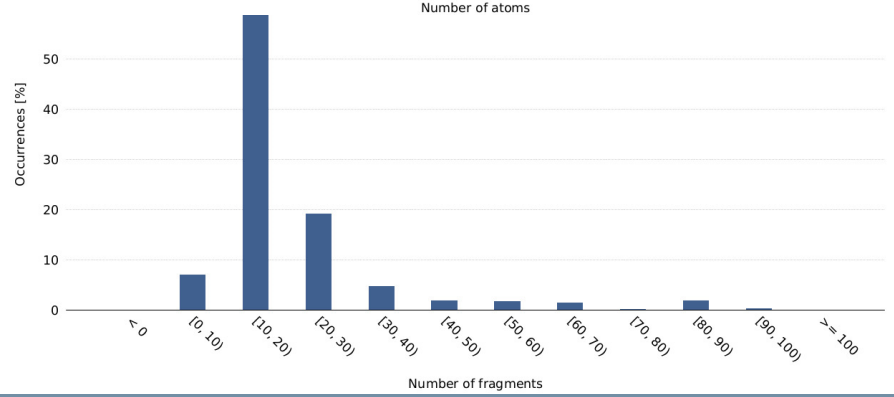


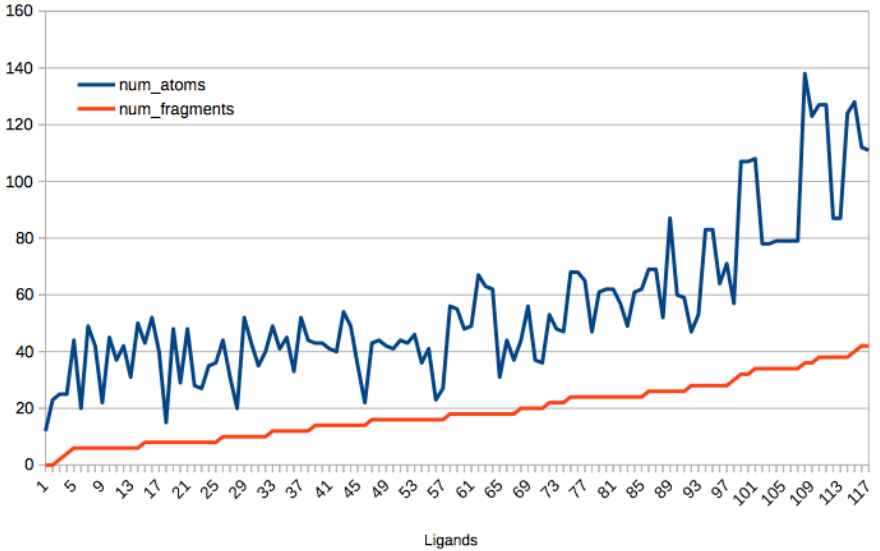
$$\vec{a}(\theta_1, \theta_2) = R(\theta_2)R(\theta_1)\overrightarrow{a_0}$$

CINECA POLITECNICO MILANO 1863

Overview of the problem size (ComplexDB)

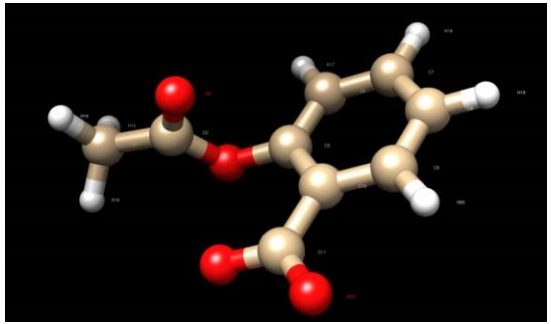






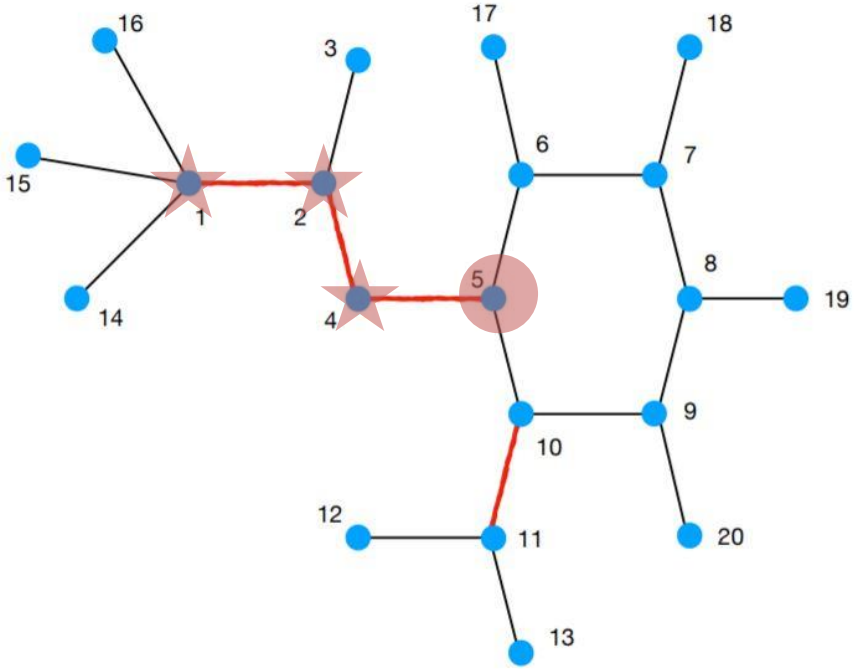
CINECA POLITECNICO MILANO 1863

Molecule Unfolding: Rotatable Betweenness Centrality



Betweenness centrality:

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



CINECA POLITECNICO MILANO 1863

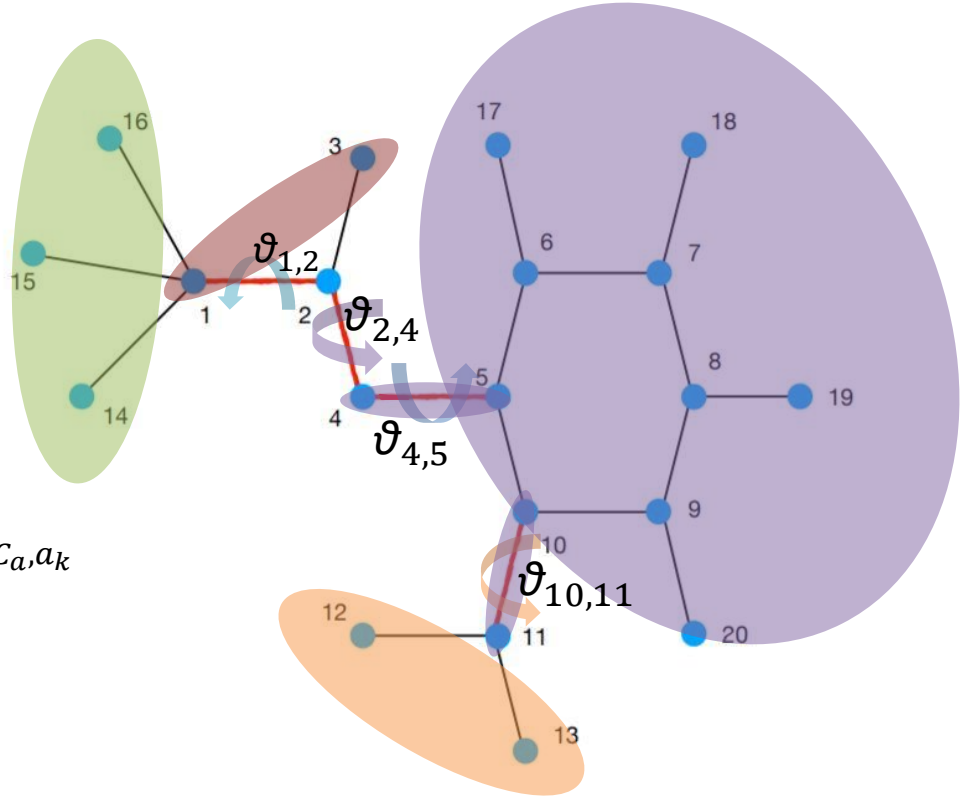
Molecule Unfolding: Rotatables Influence Set

Rotatables Influence set:

$$\mathrm{I_S} = \mathrm{E}_{C_a,a_k} \cap E_R$$

$E_R = Rotable\ bonds;$

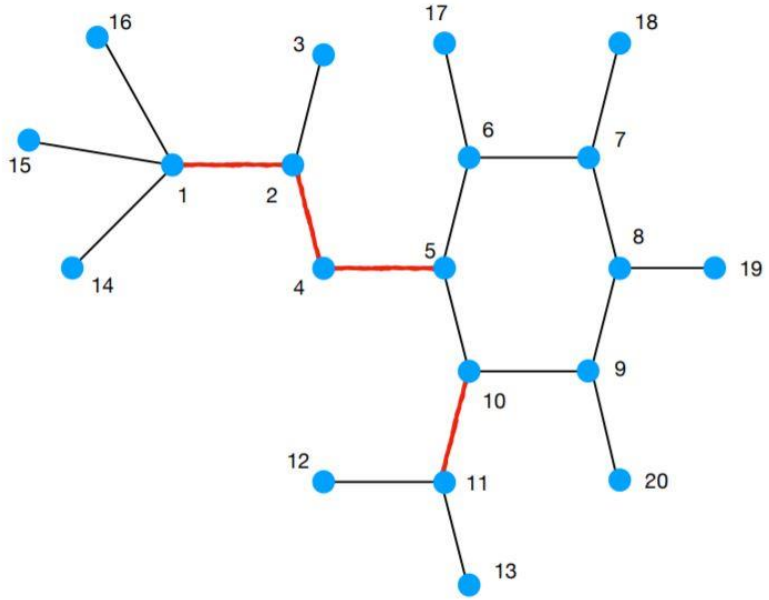
$E_{C_a,a_k} = Bonds\ on\ the\ shortest\ path\ \sigma_{C_a,a_k}$



CINECA POLITECNICO MILANO 1863

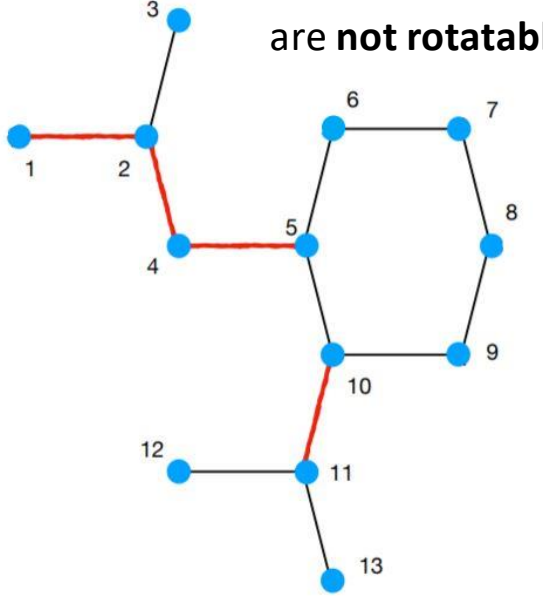
Molecule Unfolding

Original 2D molecule:



Without Hydrogen:

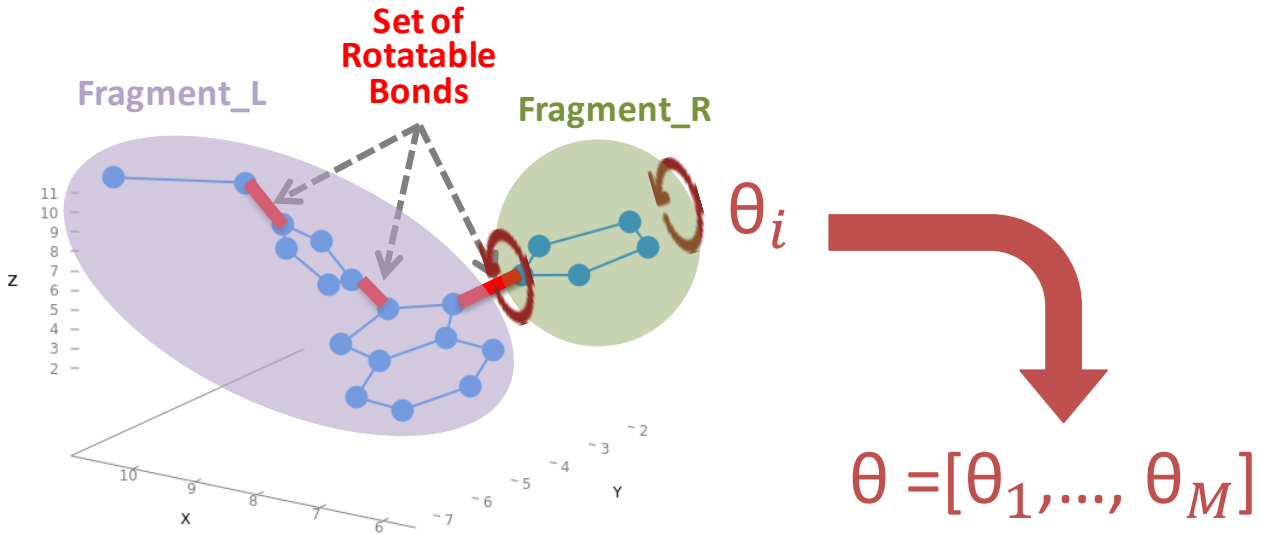
Double and **amide** bonds are **not rotatable**



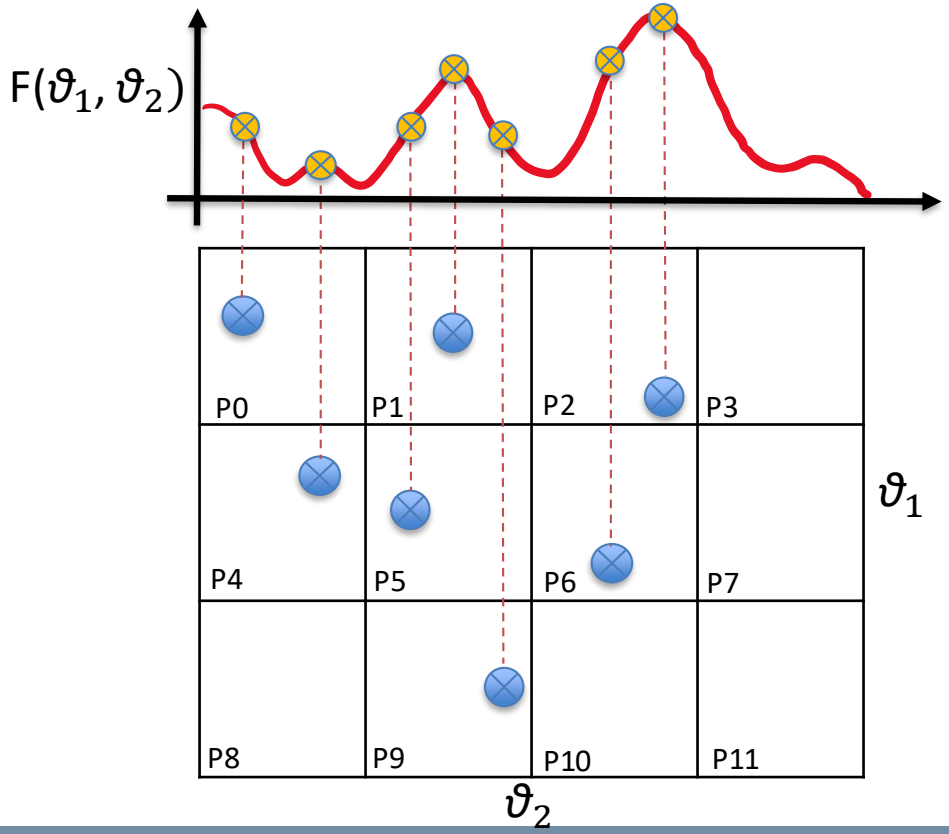
CINECA POLITECNICO MILANO 1863

Problem Definition

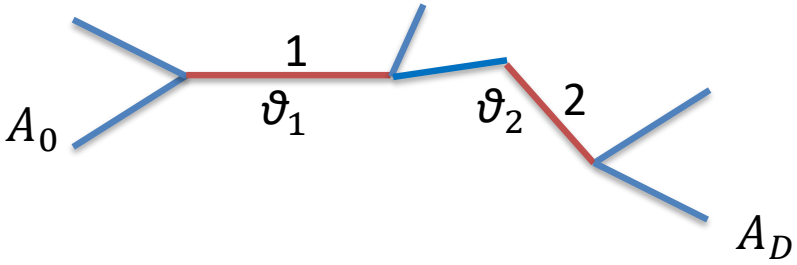
Objective: **find** the unfolded **torsion configuration** that maximizes the molecular volume, or equivalently, that **maximizes the distances between fragments.**



Molecule Conformation Exploration: Random Search



Random Search

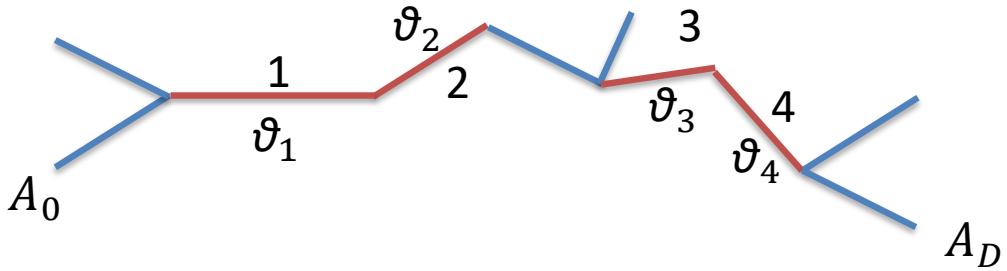


Molecule Conformation Exploration: Greedy

M= Measure total sum of internal distances

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

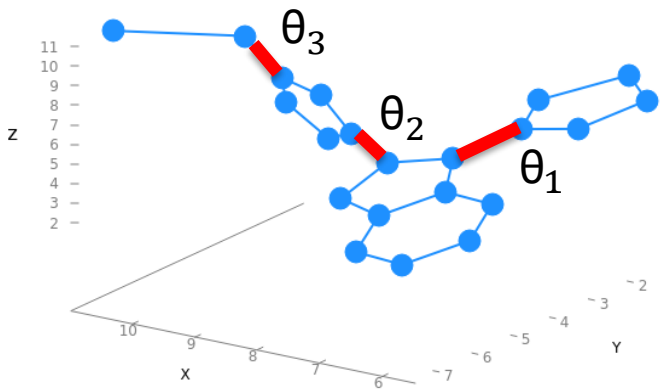
GeoDock-inspired



Greedy:

- **GeoDock** = ϑ1 - M - ϑ2 - M - ϑ3 - M - ϑ4 - M
- **Batch** = i) ϑ1, ϑ2 - M - ϑ3, ϑ4 - M ii) ϑ1, ϑ2 , ϑ3 - M - ϑ4 - M iii) ϑ1, ϑ2 , ϑ3 , ϑ4 - M

Combinatorial Optimization Problem Definition



It 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**

CINECA POLITECNICO MILANO 1863

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*

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

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

Constructing the Binary Optimization problem

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

HUBO Problem Structure

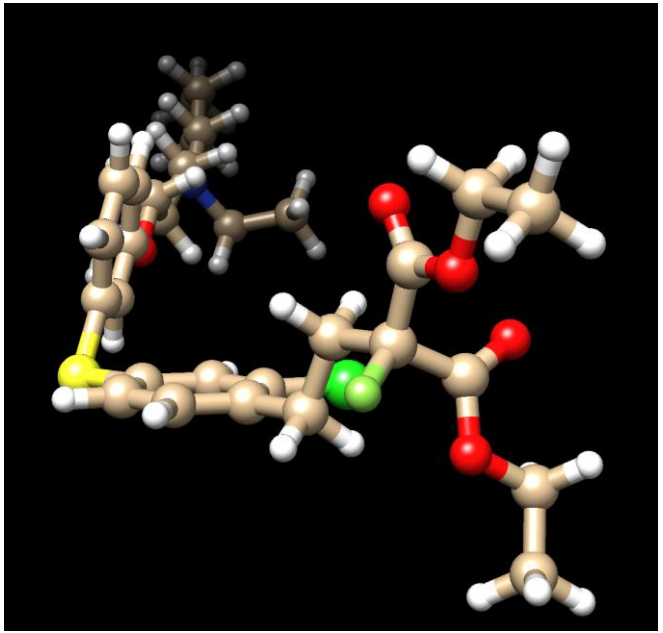
In order to obtain a **precision of $\Delta\theta_i$**, the **number of variables** needed **for each torsion** is

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

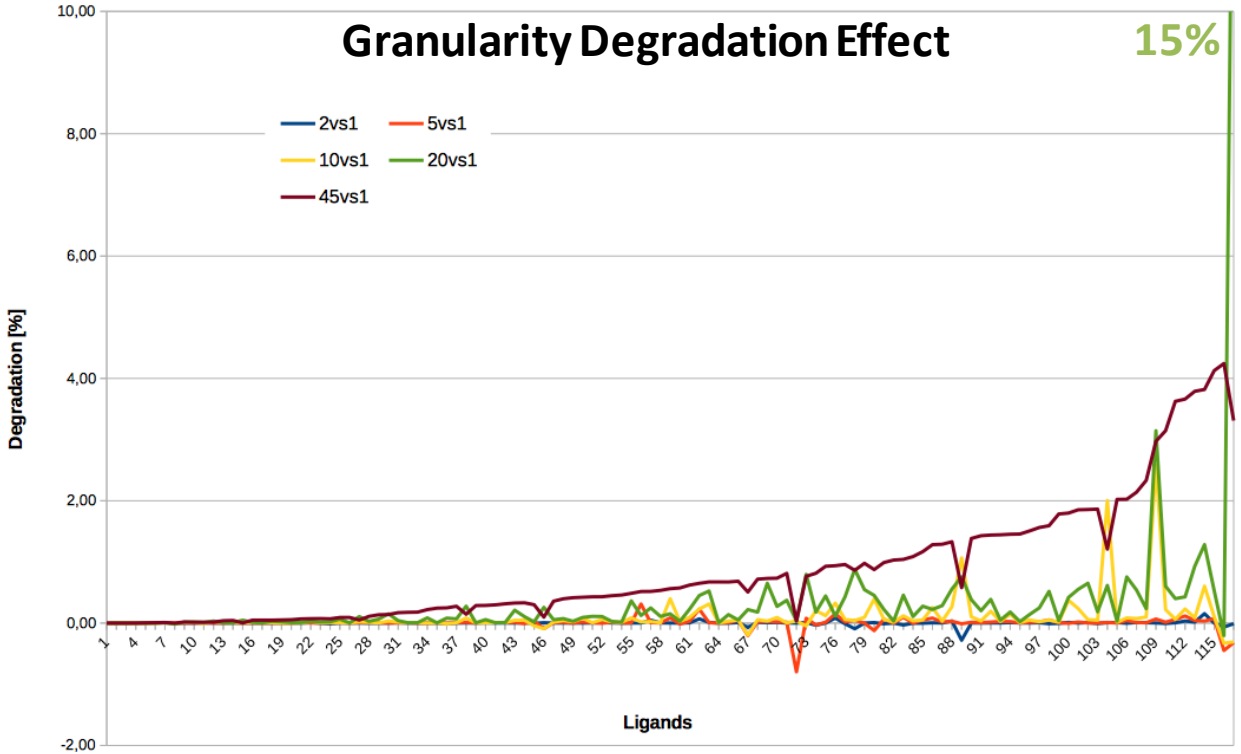
Given a molecule with ***M* torsions**, the total **number of binary variables x_{ik}** in the HUBO

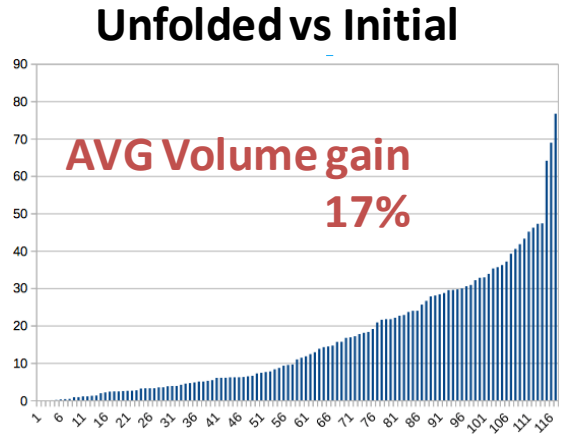
$$n = d \times M = \frac{2\pi}{\Delta\theta_i} \times M$$

Molecules: 20 to 50 atoms - 10 torsions



Angle subsampling effect on the unfolding degradation

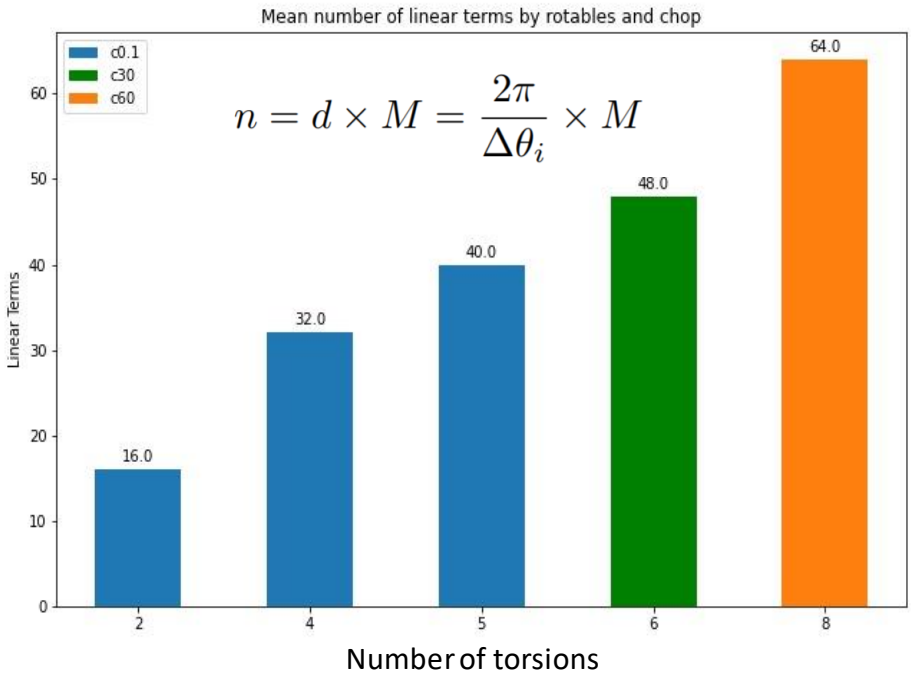




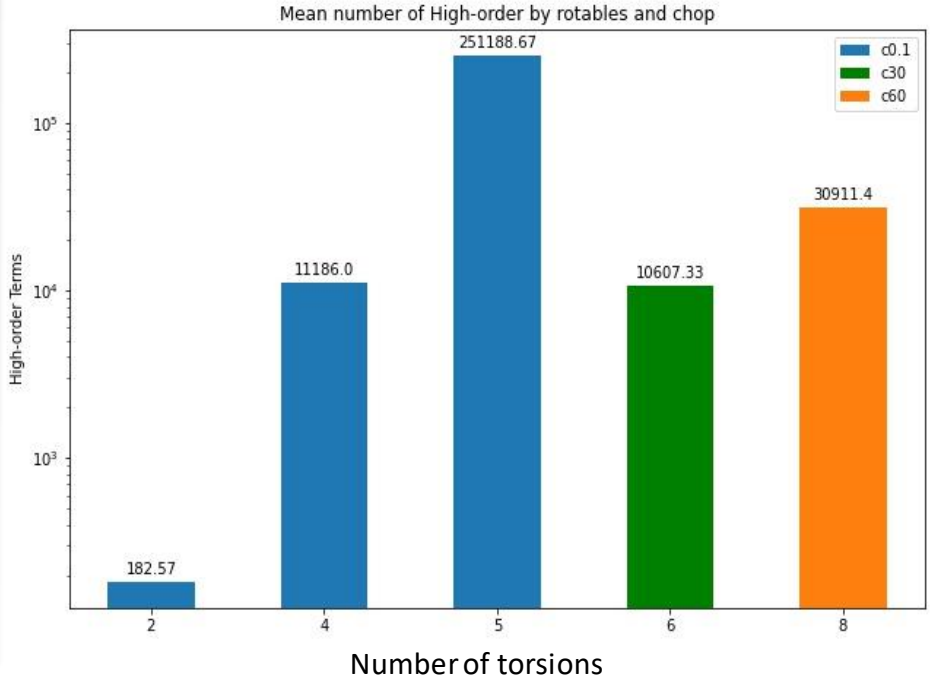
CINECA POLITECNICO MILANO 1863

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

HUBO linear terms



HUBO high order terms (Log-scale)



CINECA POLITECNICO MILANO 1863

HUBO Problem Approximation

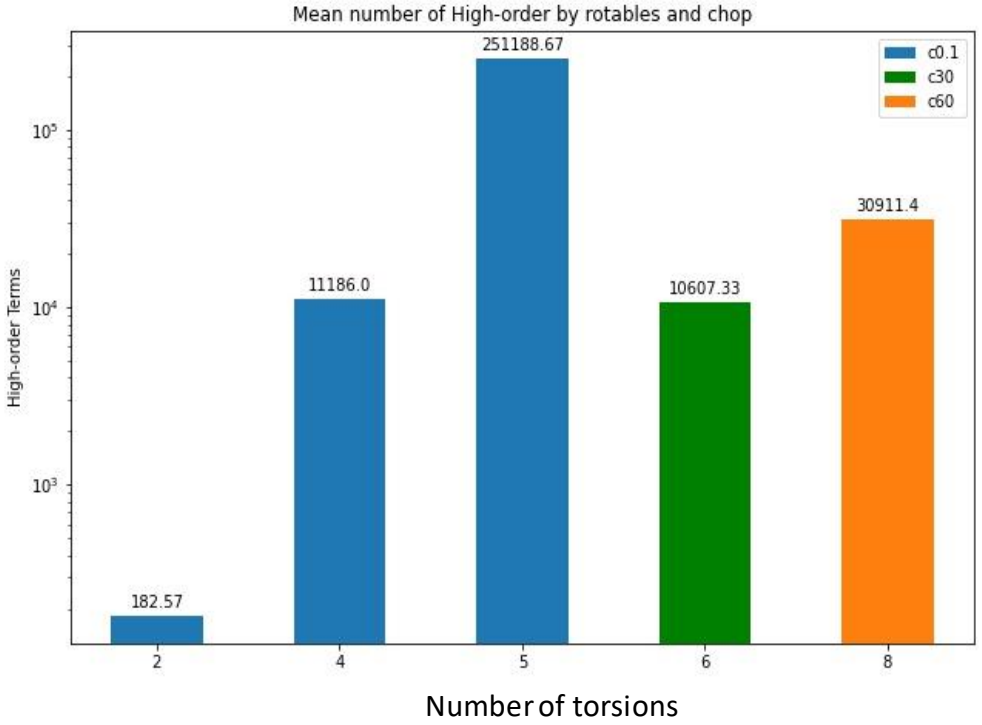
Delete HUBO terms below a certain **threshold.** Applied in **two phases:**

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***

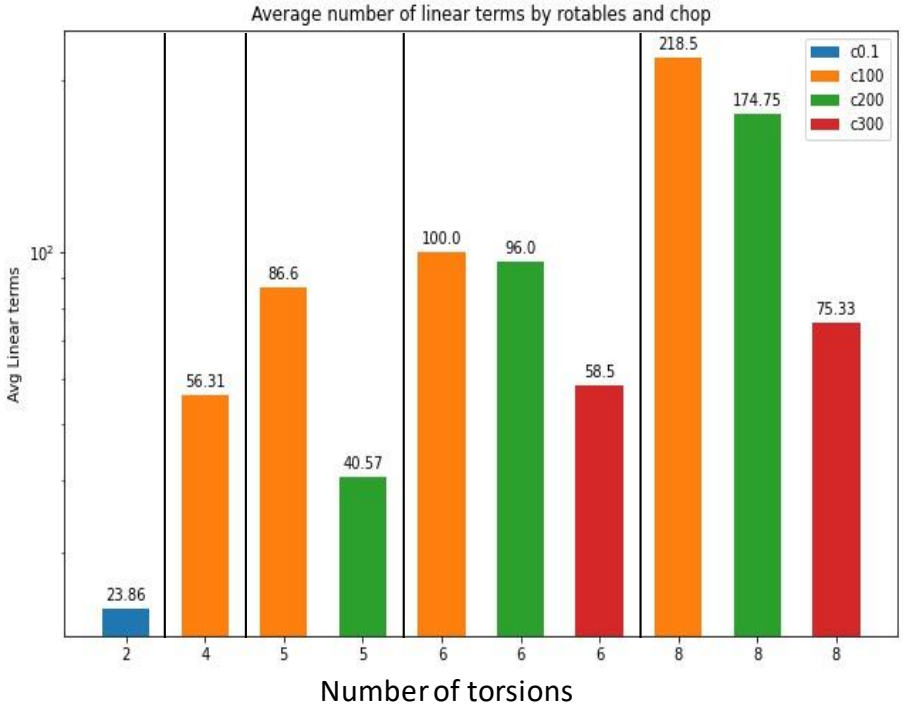
HUBO high order terms (Log-scale)



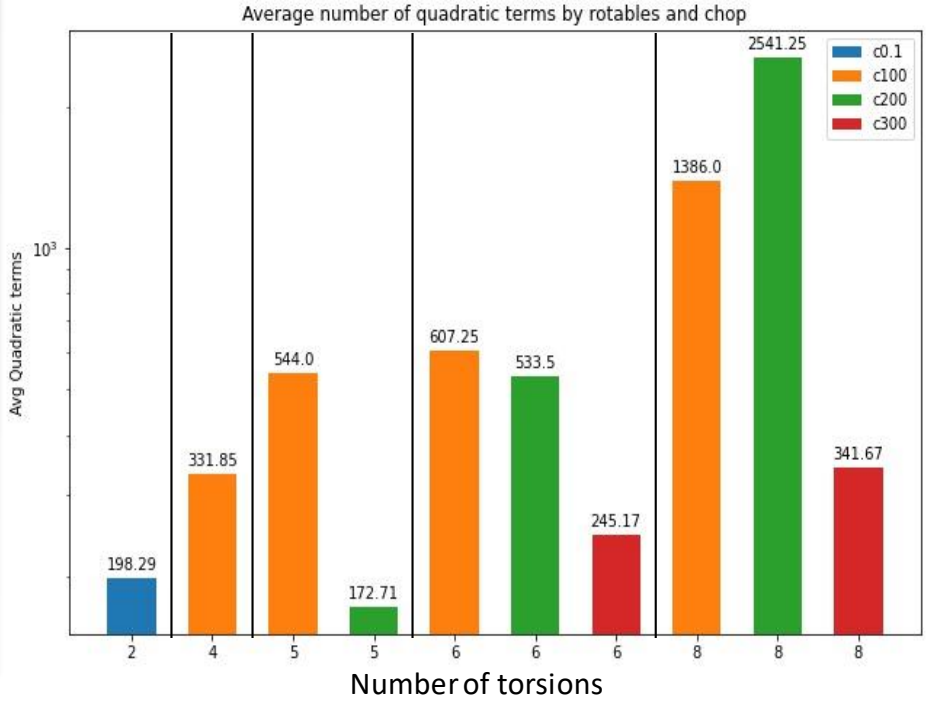
CINECA POLITECNICO MILANO 1863

Form HUBOs to QUBOs

QUBO linear terms



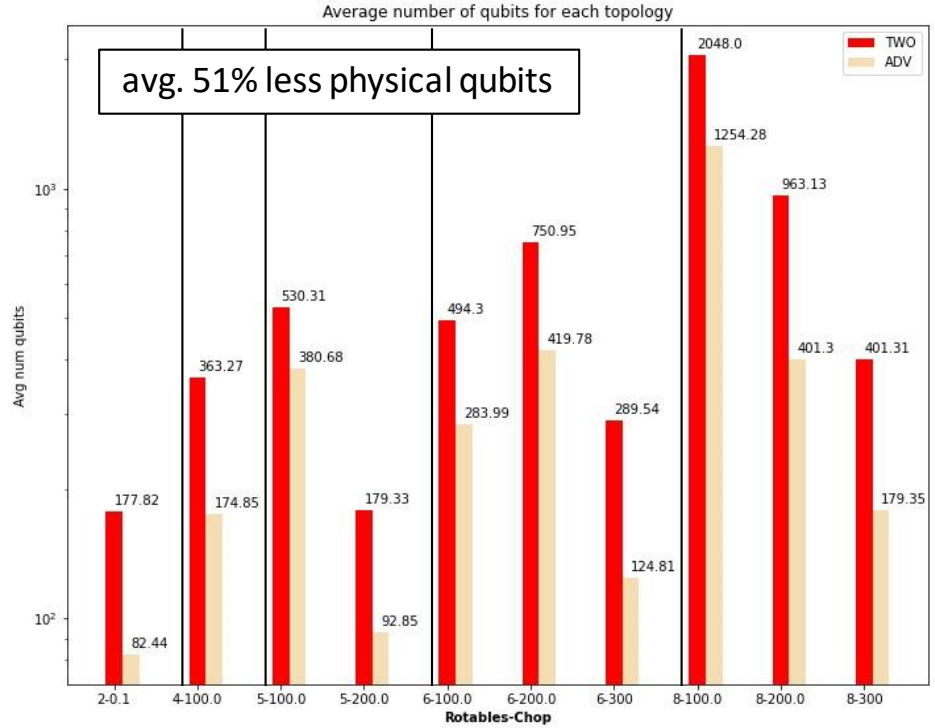
QUBO quadratic terms



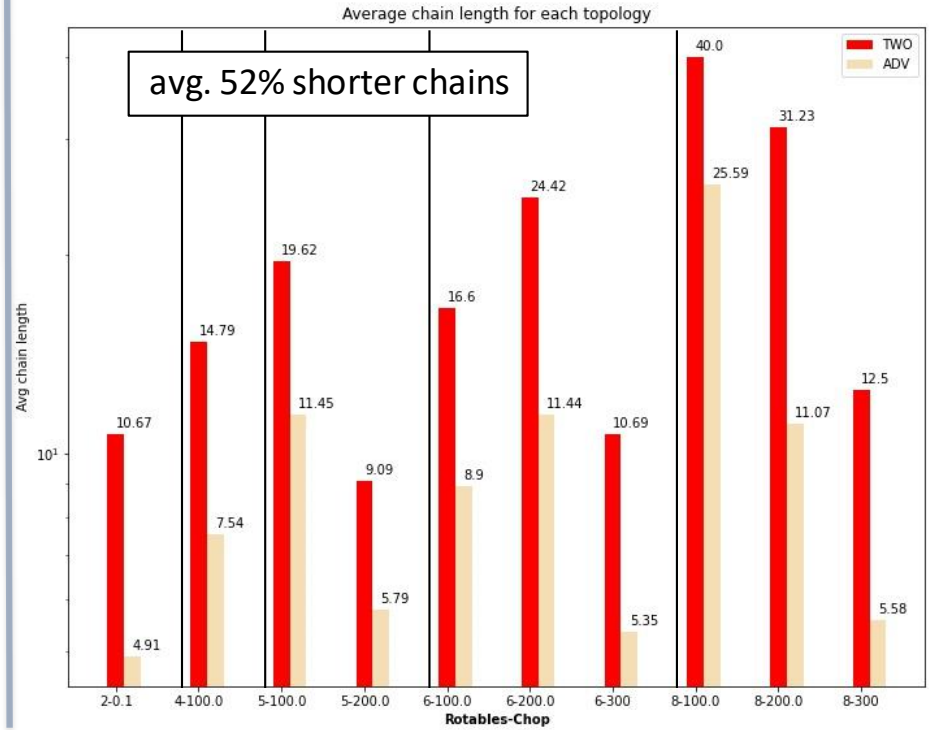
CINECA POLITECNICO MILANO 1863

Embeddings DW2000Q & Advantage

Number of physical qubits

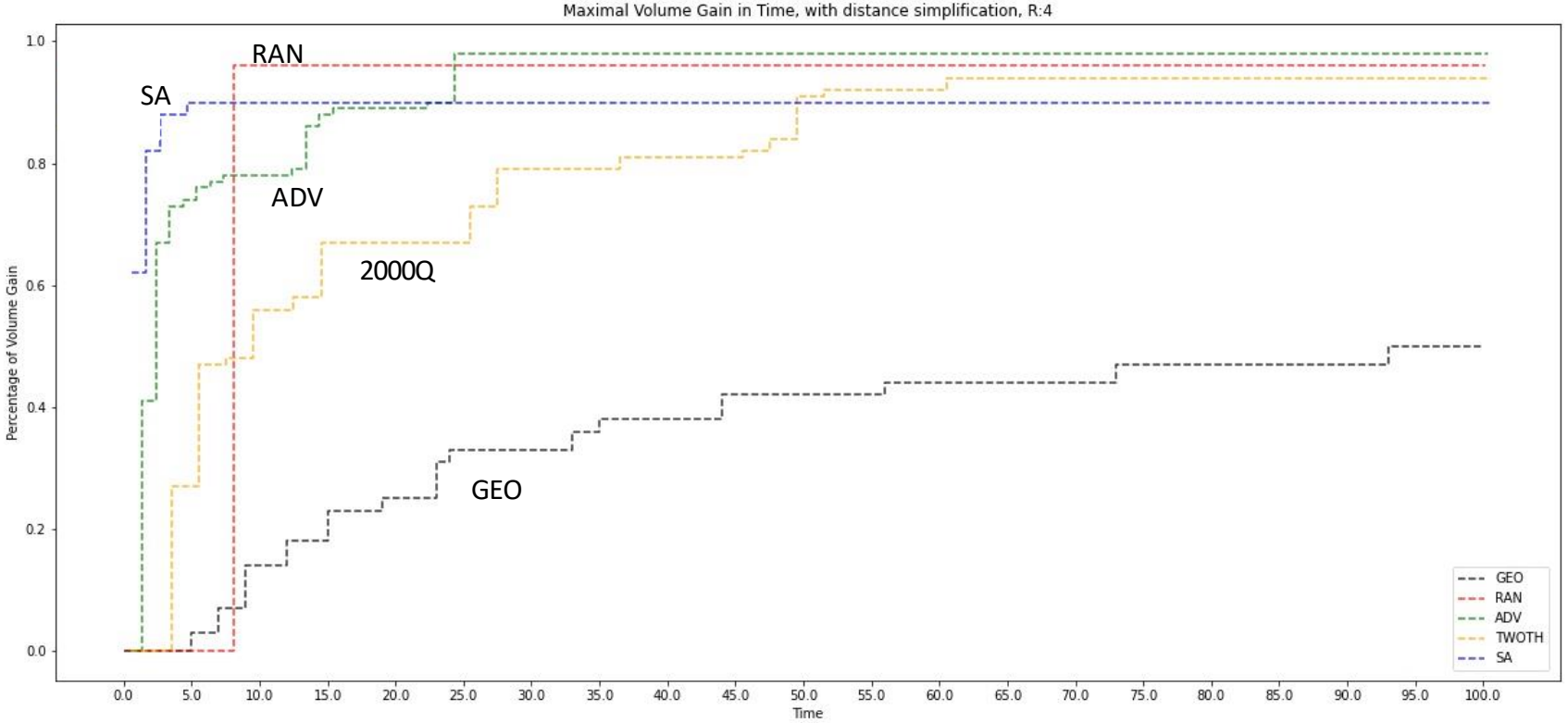


AVG chain length

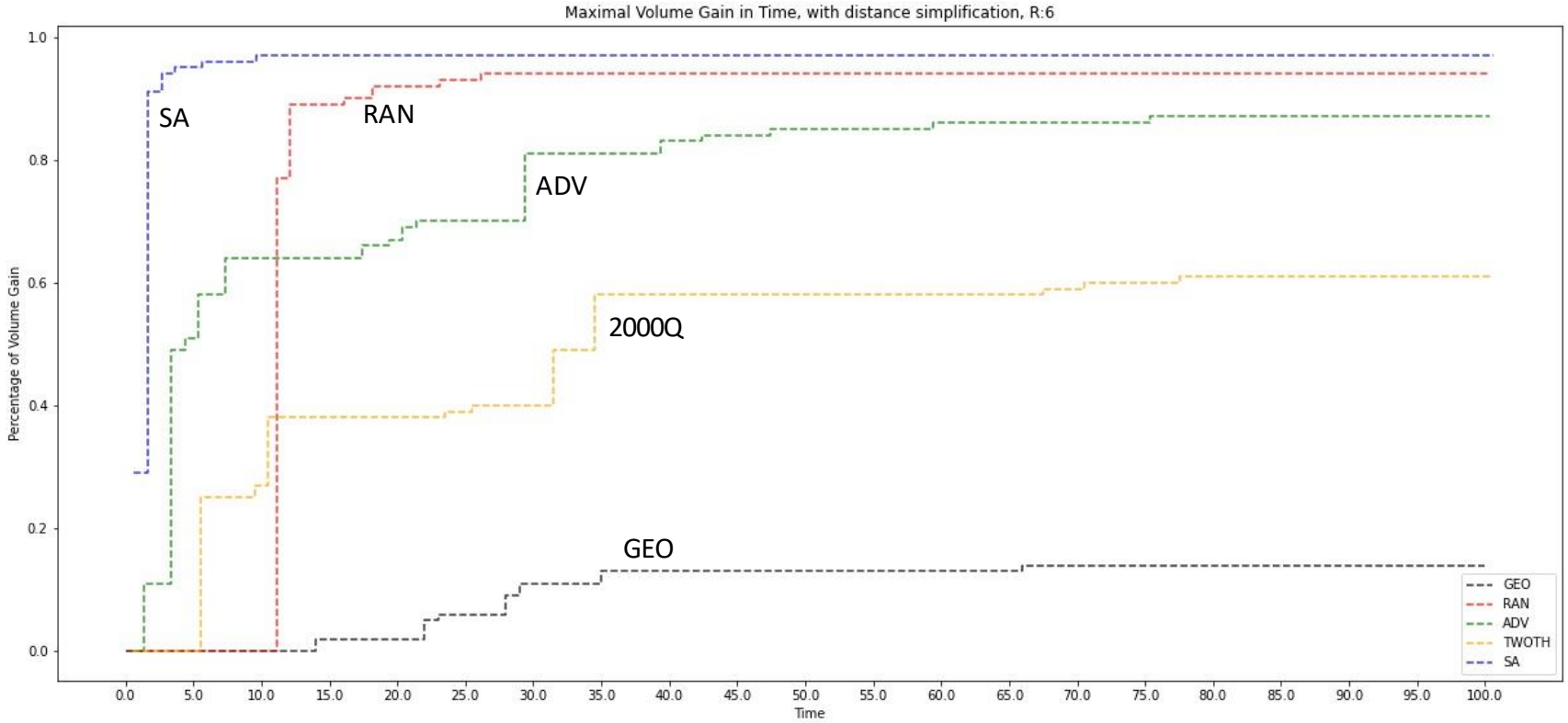


CINECA POLITECNICO MILANO 1863

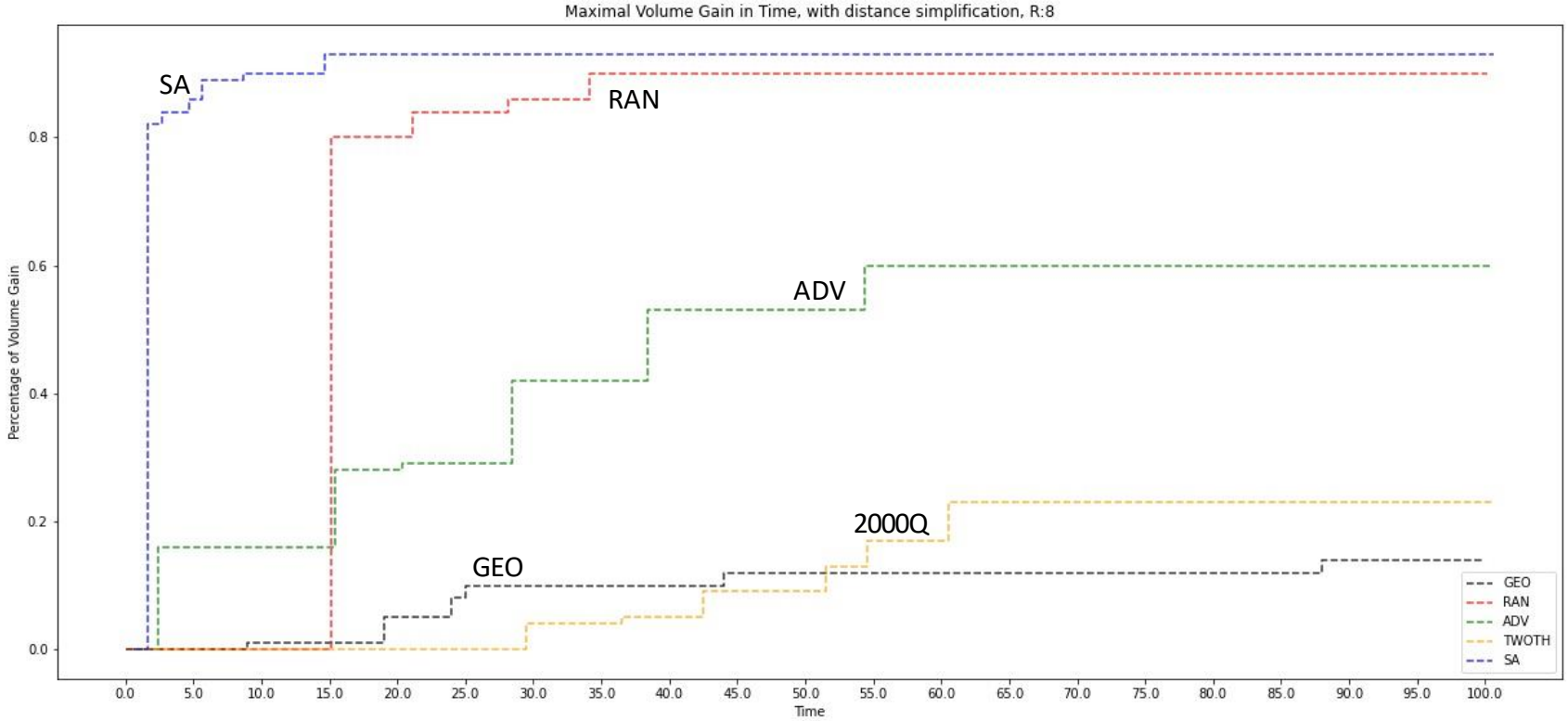
Results, 4 Torsions : Volume Gain in Time (seconds)



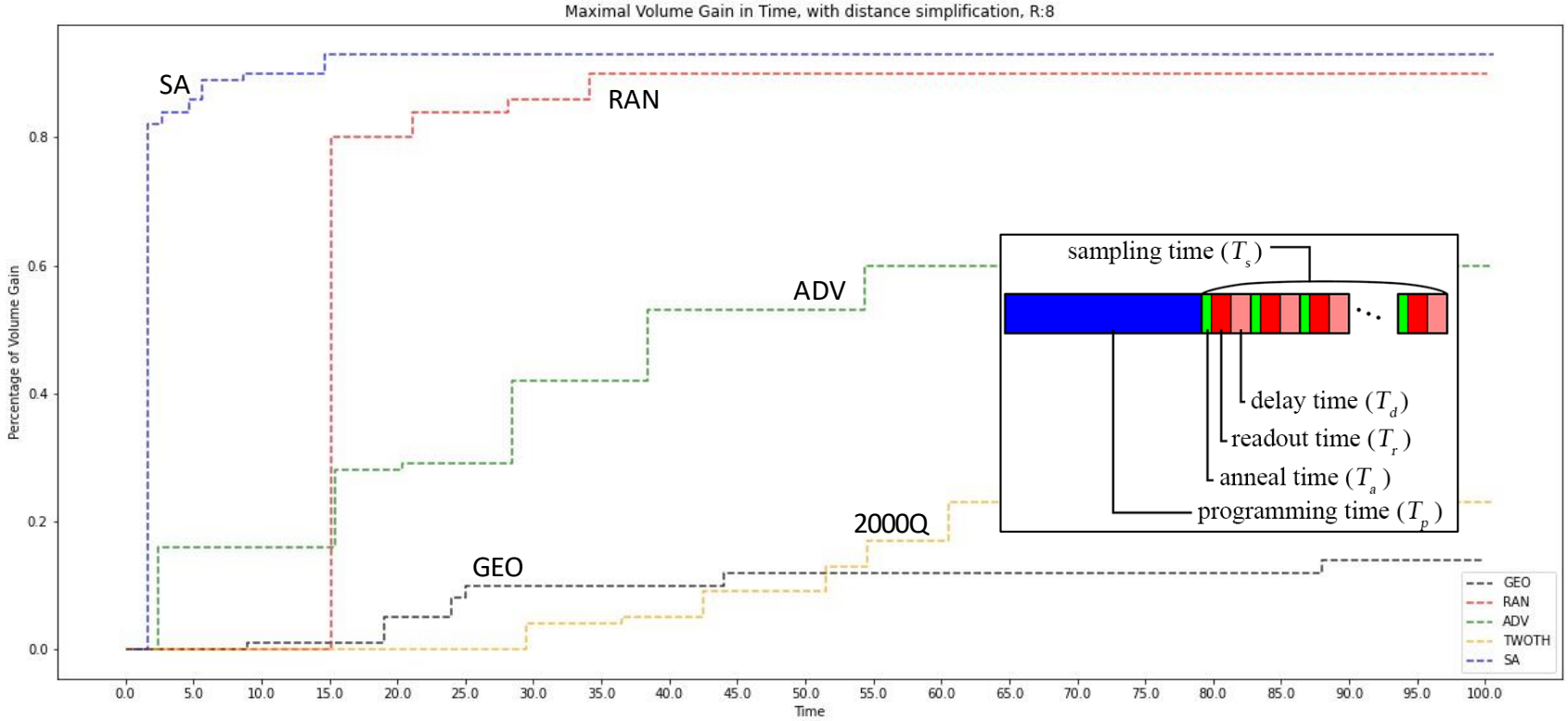
Results, 6 Torsions : Volume Gain in Time (seconds)



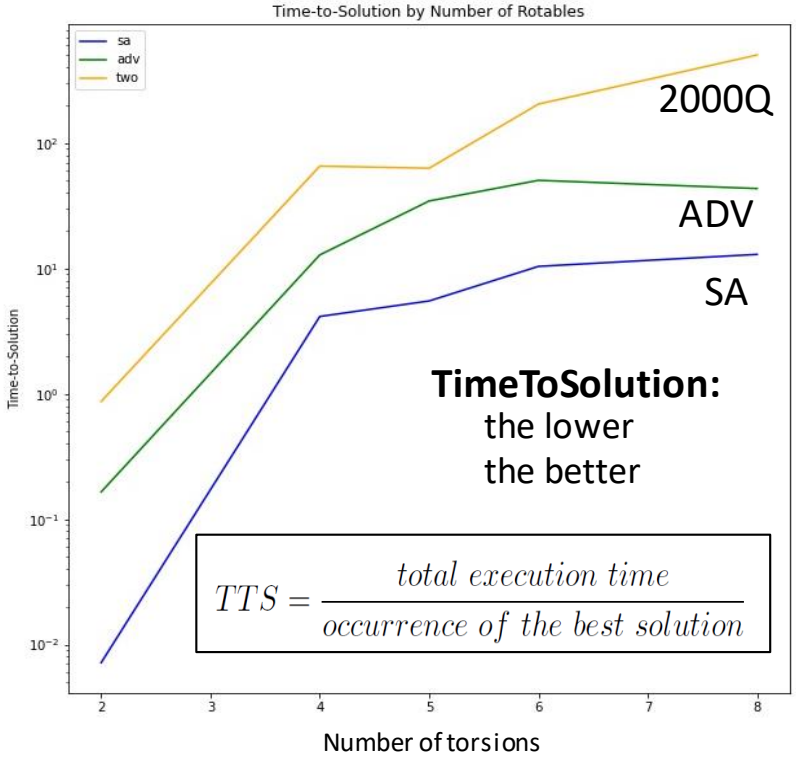
Results, 8 Torsions : Volume Gain in Time (seconds)



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

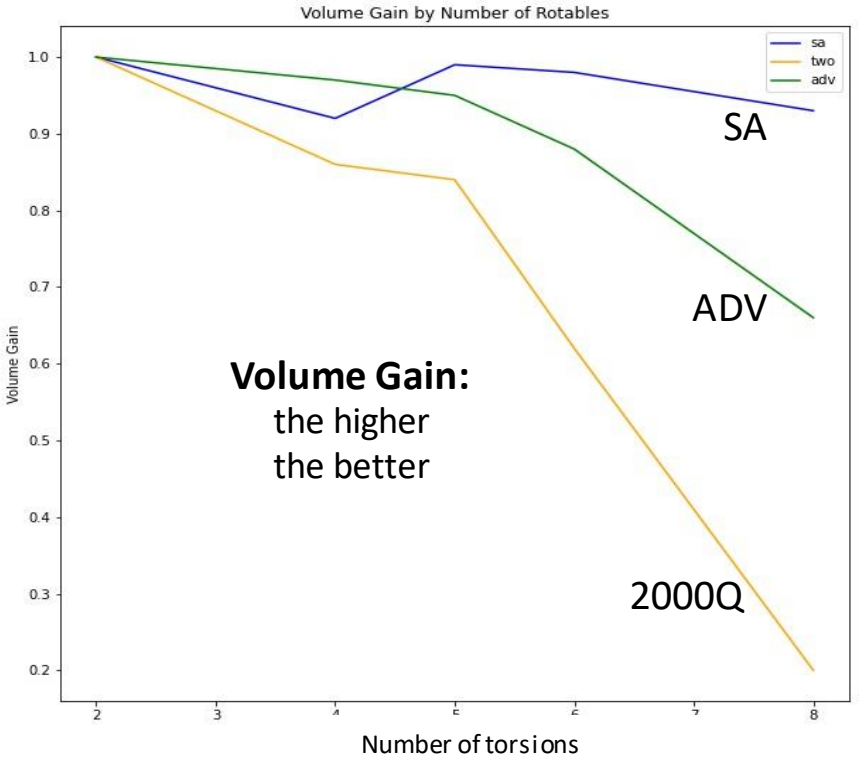


Results: Time To Solution (TTS) & Volume Gain



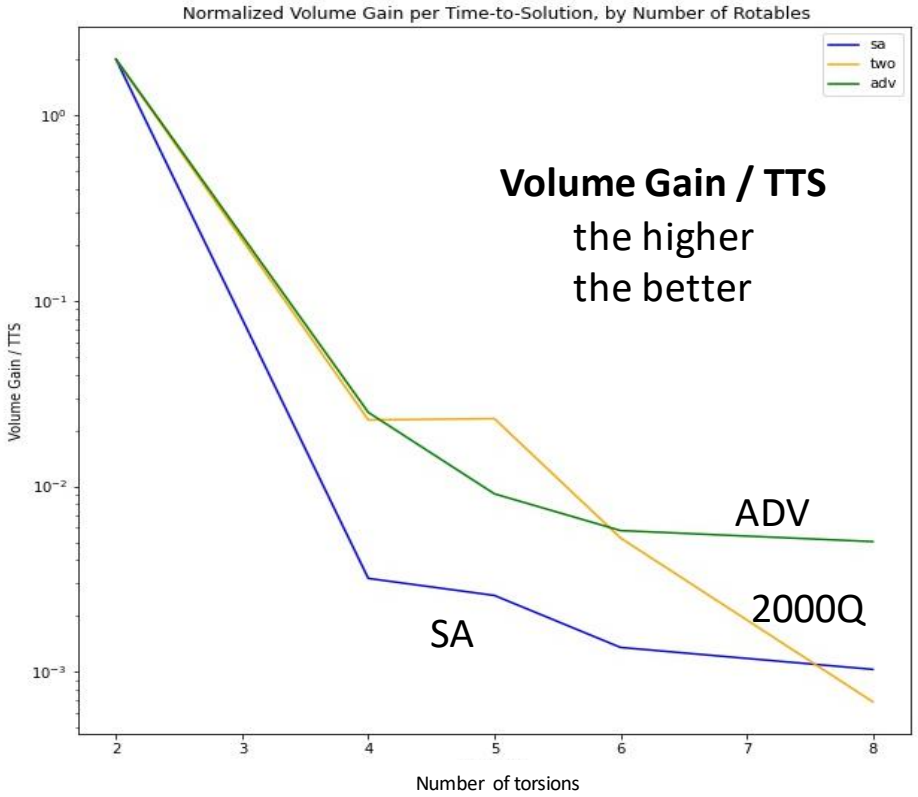
TimeToSolution:
the lower
the better

$$TTS = \frac{total\ execution\ time}{occurrence\ of\ the\ best\ solution}$$



Volume Gain:
the higher
the better

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**