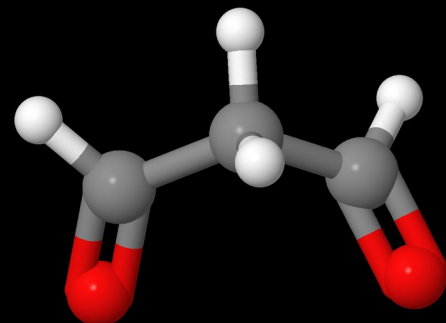
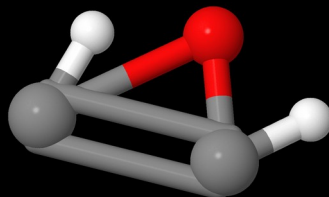
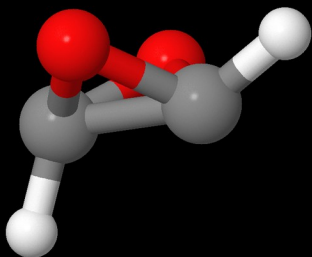


Génération automatique de molécules chimiques



Plan

Plan

- Introduction et mise en contexte

Plan

- Introduction et mise en contexte
- Première modélisation avec liaisons simples

Plan

- Introduction et mise en contexte
- Première modélisation avec liaisons simples
- Gestion des liaisons doubles et triples

Plan

- Introduction et mise en contexte
- Première modélisation avec liaisons simples
- Gestion des liaisons doubles et triples
- Plongement dans l'espace 3D : gestion des coordonnées

Plan

- Introduction et mise en contexte
- Première modélisation avec liaisons simples
- Gestion des liaisons doubles et triples
- Plongement dans l'espace 3D : gestion des coordonnées
- Détails d'implémentations

Plan

- Introduction et mise en contexte
- Première modélisation avec liaisons simples
- Gestion des liaisons doubles et triples
- Plongement dans l'espace 3D : gestion des coordonnées
- Détails d'implémentations
- Conclusion

Cadre du projet

- Encadrant : Nicolas Prcovic enseignant-chercheur
- Équipe COALA du LIS



Cadre du projet

- Encadrant : Nicolas Prcovic enseignant-chercheur
- Équipe COALA du LIS
- Sujet interdisciplinaire : chimie & informatique

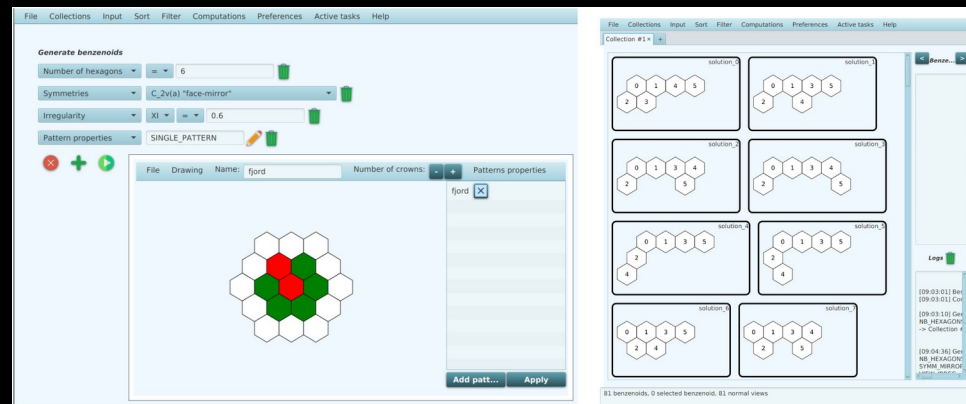


Mise en contexte

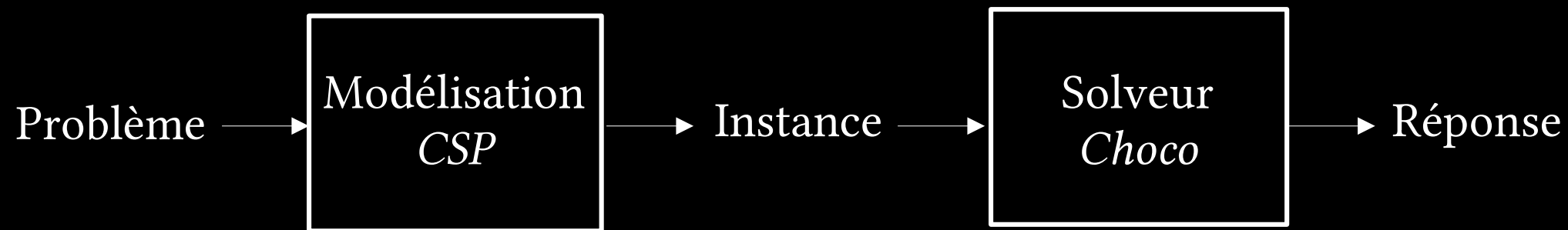
- Adrien Varet, ancien doctorant
- **Sujet de thèse** : utilisation du formalisme CSP pour résoudre des problématiques liées aux benzénoïdes

Mise en contexte

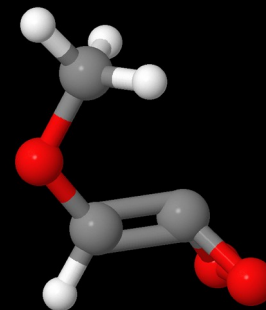
- Adrien Varet, ancien doctorant
- **Sujet de thèse** : utilisation du formalisme CSP pour résoudre des problématiques liées aux benzénoïdes
→ Création de l'application **BenzAI**



Utilisation de la PPC



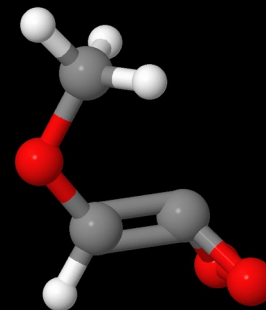
Utilisation de la PPC



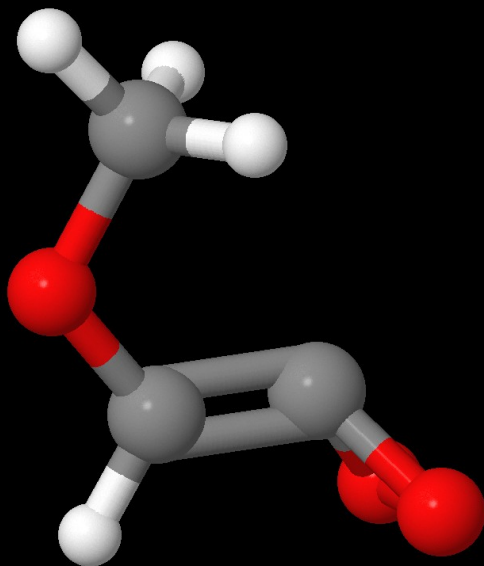
Utilisation de la PPC



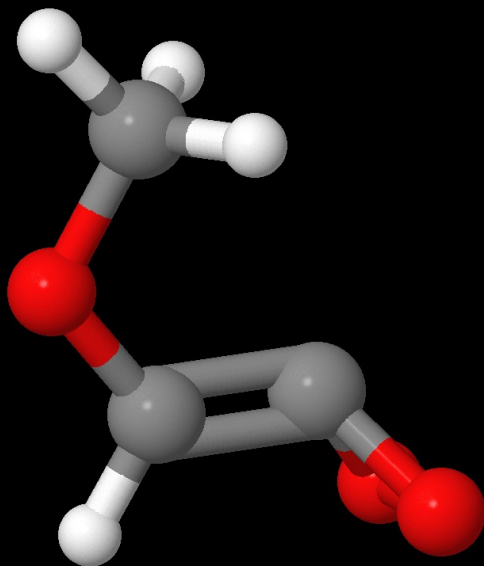
$\mathcal{P} = (X, D, C)$



Qu'est-ce qu'une structure moléculaire ?

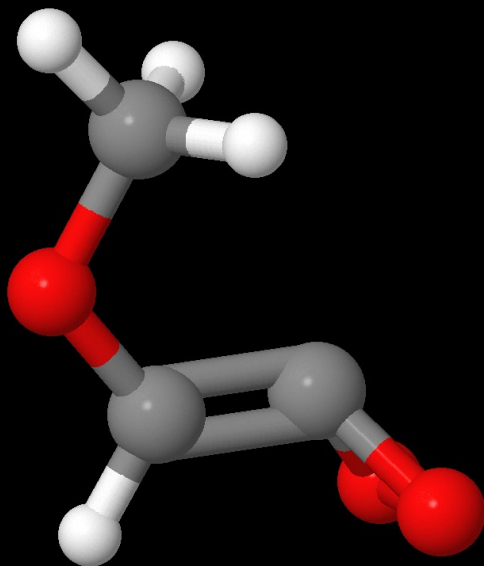


Qu'est-ce qu'une structure moléculaire ?



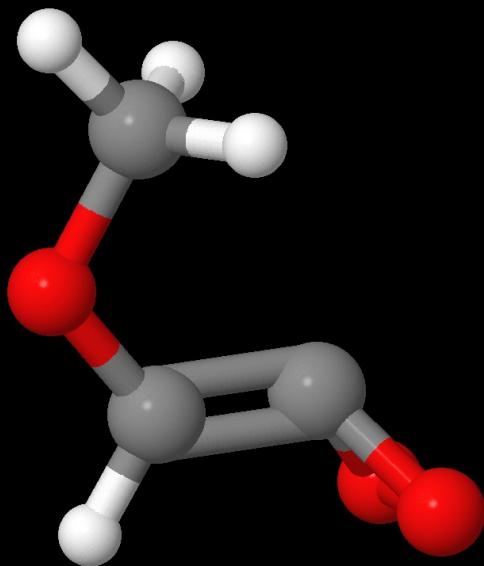
- Graphe étiqueté :

Qu'est-ce qu'une structure moléculaire ?



- **Graphe étiqueté :**
→ Sommet = atome

Qu'est-ce qu'une structure moléculaire ?

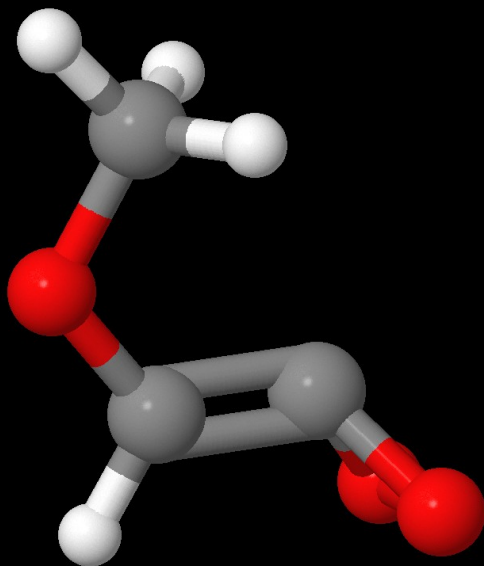


- **Graphe étiqueté :**

→ Sommet = atome

→ Étiquette = type d'atome

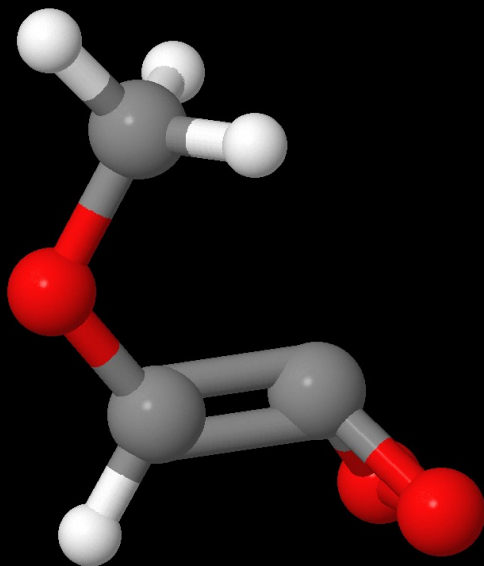
Qu'est-ce qu'une structure moléculaire ?



- Graphe étiqueté :

- Sommet = atome
- Étiquette = type d'atome
- Arête = liaison entre deux atomes
 - *Plusieurs types de liaisons*

Qu'est-ce qu'une structure moléculaire ?



- Graphe étiqueté :

→ Sommet = atome

→ Étiquette = type d'atome

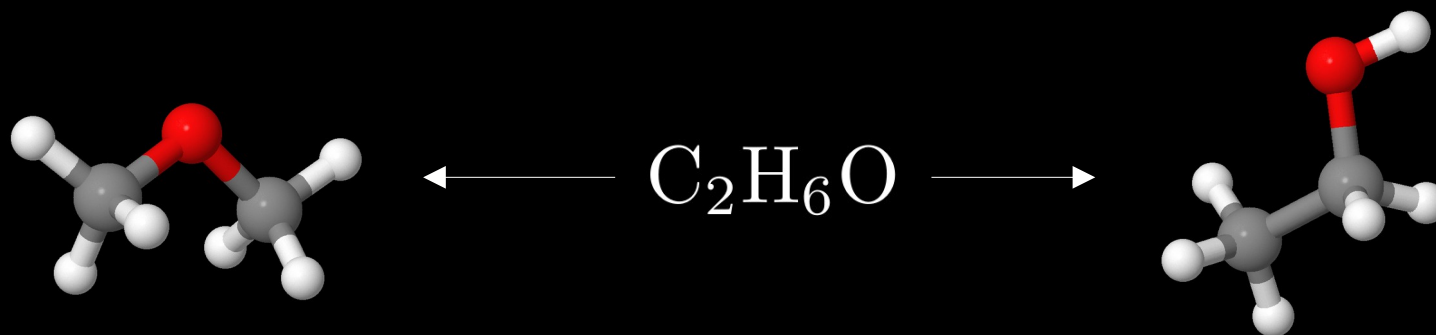
→ Arête = liaison entre deux atomes

→ *Plusieurs types de liaisons*

- Connexité

Une formule chimique : plusieurs structures moléculaires

Une formule
↓
Liste d'isomère



Objectif : trouver l'ensemble des isomères

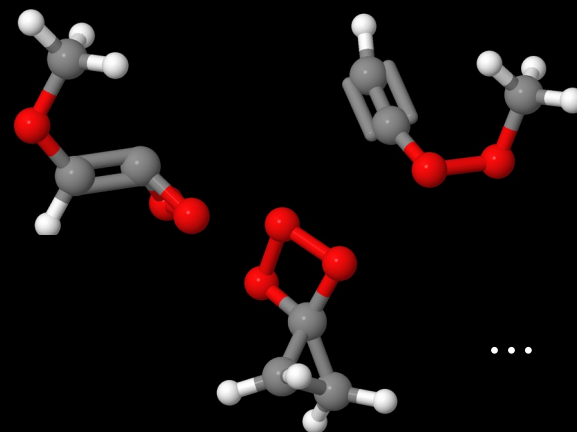
Formule chimique

$$\mathcal{P} = (X, D, C)$$

→ Listes des isomères



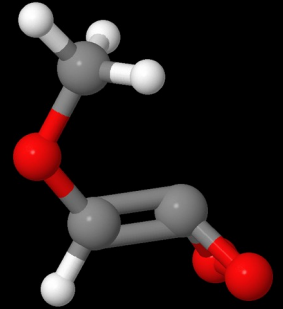
???



Modélisation



$$\mathcal{P} = (X, D, C)$$

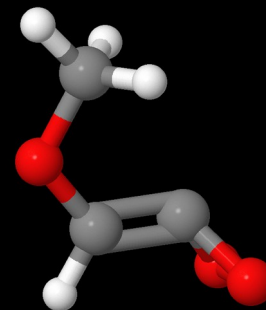


- $X = \{G_{mol} = (V, E)\}, |V| = n$

Modélisation



$$\mathcal{P} = (X, D, C)$$

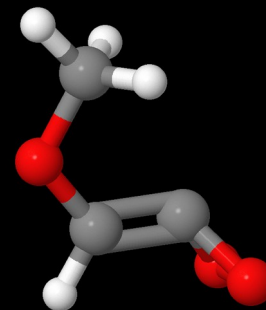


- $X = \{G_{mol} = (V, E)\}, |V| = n$
- $D = d_{G_{mol}}, \text{ où } d_{G_{mol}} = [(V, \emptyset) : K_n]$

Modélisation



$$\mathcal{P} = (X, D, C)$$

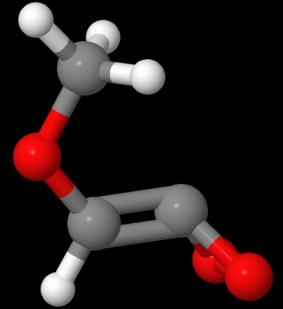


- $X = \{G_{mol} = (V, E)\}, |V| = n$
- $D = d_{G_{mol}}, \text{ où } d_{G_{mol}} = [(V, \emptyset) : K_n]$
- C l'ensemble des contraintes suivantes :

Modélisation



$$\mathcal{P} = (X, D, C)$$

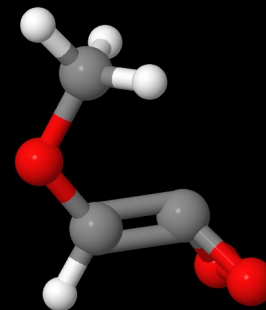


- $X = \{G_{mol} = (V, E)\}, |V| = n$
- $D = d_{G_{mol}}, \text{ où } d_{G_{mol}} = [(V, \emptyset) : K_n]$
- C l'ensemble des contraintes suivantes :
 - **La contrainte induite de valence :**
 $\forall i \in V, \text{ degree}(i) = \text{valence}(i)$

Modélisation

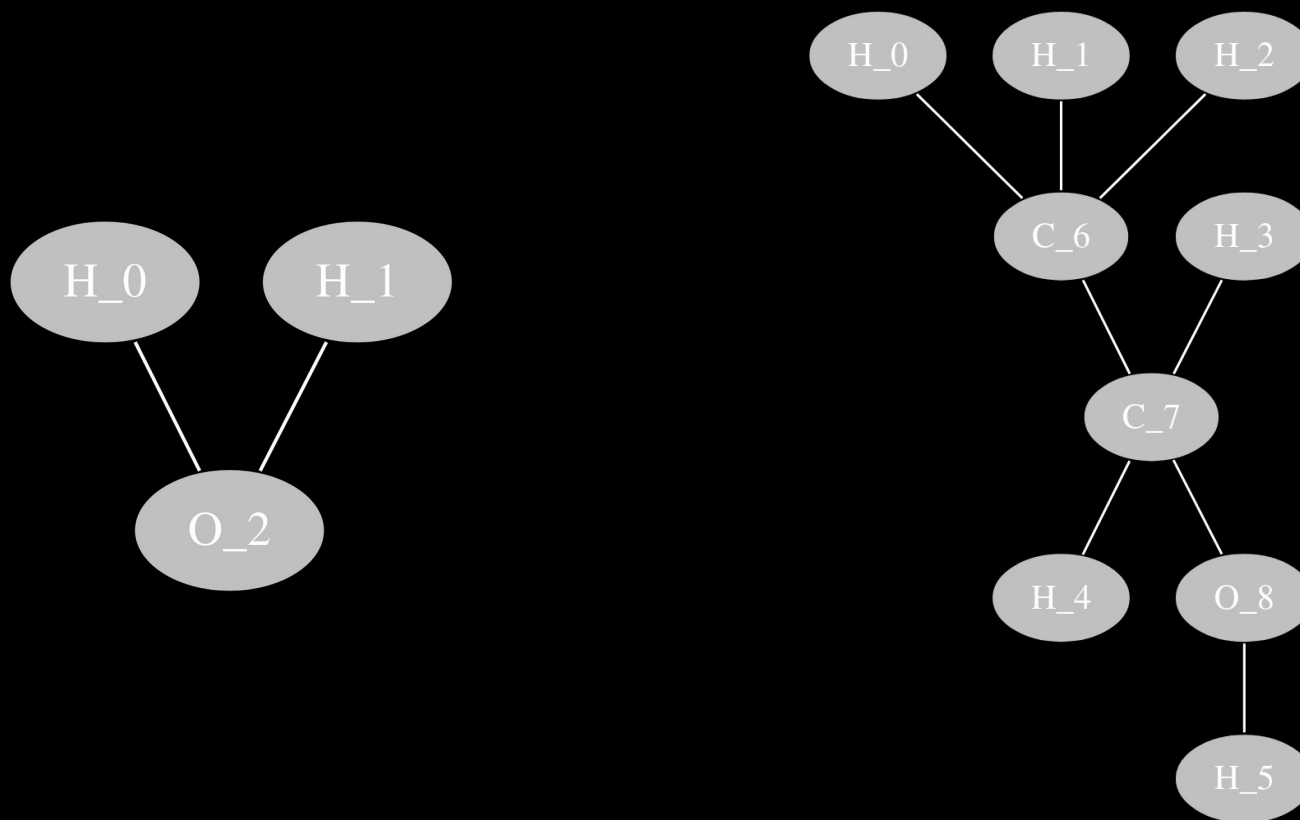


$$\mathcal{P} = (X, D, C)$$

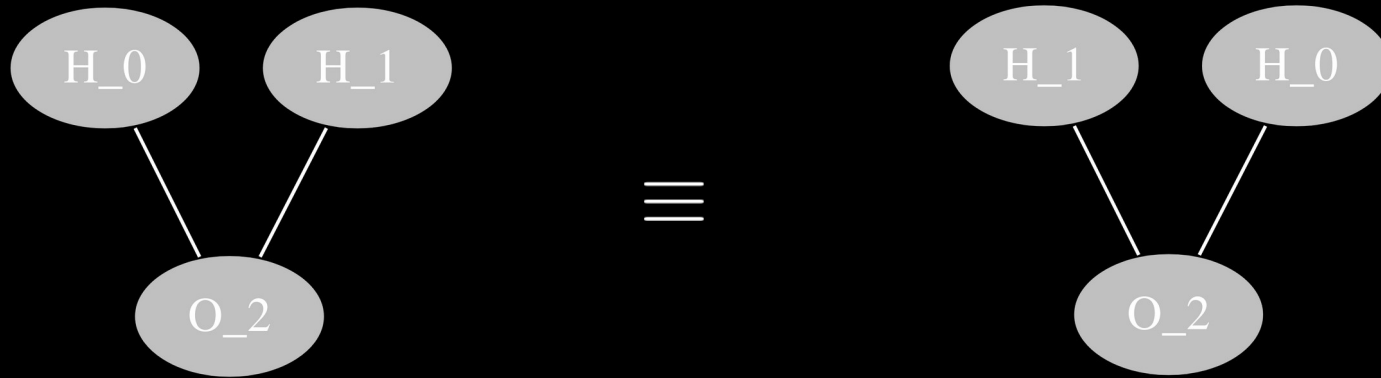


- $X = \{G_{mol} = (V, E)\}, |V| = n$
- $D = d_{G_{mol}}, \text{ où } d_{G_{mol}} = [(V, \emptyset) : K_n]$
- C l'ensemble des contraintes suivantes :
 - **La contrainte induite de valence :**
 $\forall i \in V, \text{degree}(i) = \text{valence}(i)$
 - **La contrainte de connexité.**

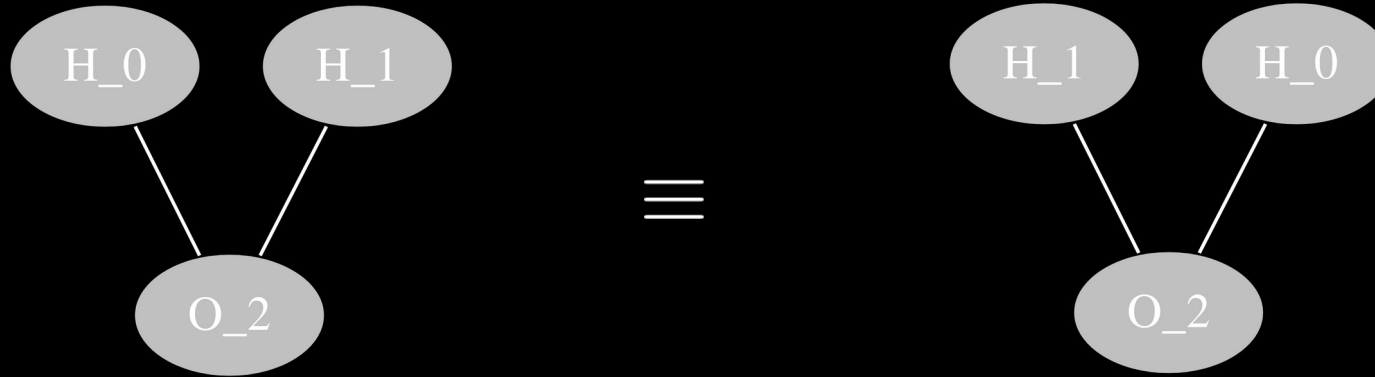
Résultats de la modélisation initiale



Problème d'isomorphisme

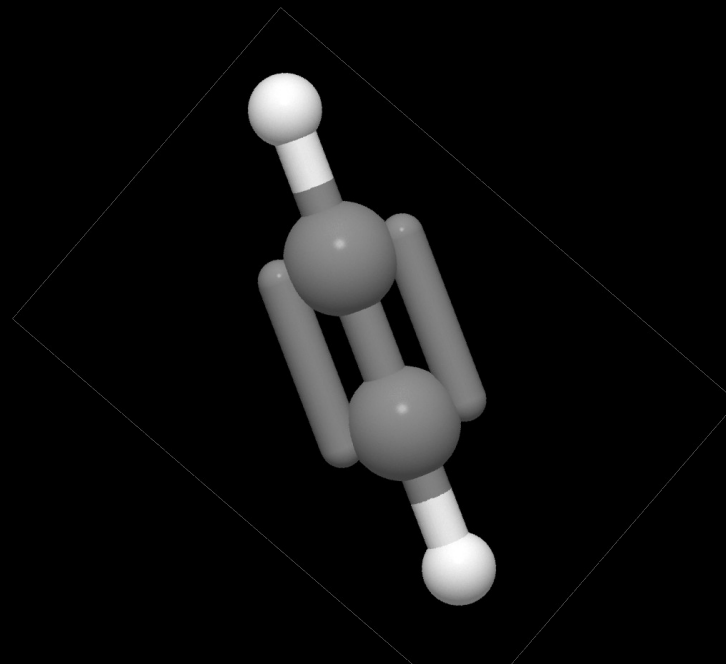
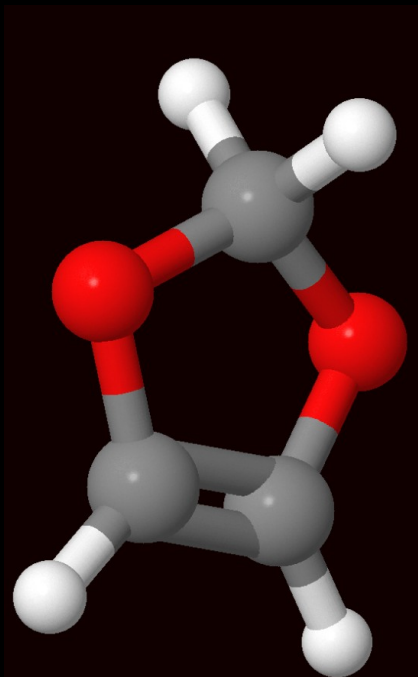


Problème d'isomorphisme



- Test d'isomorphisme : algorithme VF2

Première amélioration : doubles et triples liaisons



Première amélioration : doubles et triples liaisons

- Pour les variables : $liaisons \in \mathcal{M}_{n \times n}(\mathbb{N})$, $\forall i \neq j$:

Première amélioration : doubles et triples liaisons

- Pour les variables : $liaisons \in \mathcal{M}_{n \times n}(\mathbb{N})$, $\forall i \neq j$:

$$liaisons(i, j) = \begin{cases} 0 & \text{si } i \text{ et } j \text{ ne sont pas en liaisons} \\ 1 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{simple} \\ 2 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{double} \\ 3 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{triple} \end{cases}$$

Première amélioration : doubles et triples liaisons

- Pour les variables : $liaisons \in \mathcal{M}_{n \times n}(\mathbb{N})$, $\forall i \neq j$:

$$liaisons(i, j) = \begin{cases} 0 & \text{si } i \text{ et } j \text{ ne sont pas en liaisons} \\ 1 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{simple} \\ 2 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{double} \\ 3 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{triple} \end{cases}$$

- Pour les contraintes :

Première amélioration : doubles et triples liaisons

- Pour les variables : $liaisons \in \mathcal{M}_{n \times n}(\mathbb{N})$, $\forall i \neq j$:

$$liaisons(i, j) = \begin{cases} 0 & \text{si } i \text{ et } j \text{ ne sont pas en liaisons} \\ 1 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{simple} \\ 2 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{double} \\ 3 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{triple} \end{cases}$$

- Pour les contraintes :
 - Contraintes sur le nombre de liaisons de chaque atome : $\forall i \in V$,

$$\sum_{j=0}^{n-1} liaisons(i, j) = \text{valence}(i)$$

Première amélioration : doubles et triples liaisons

- Pour les variables : $liaisons \in \mathcal{M}_{n \times n}(\mathbb{N})$, $\forall i \neq j$:

$$liaisons(i, j) = \begin{cases} 0 & \text{si } i \text{ et } j \text{ ne sont pas en liaisons} \\ 1 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{simple} \\ 2 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{double} \\ 3 & \text{si } i \text{ et } j \text{ sont reliés par une liaison } \mathbf{triple} \end{cases}$$

- Pour les contraintes :

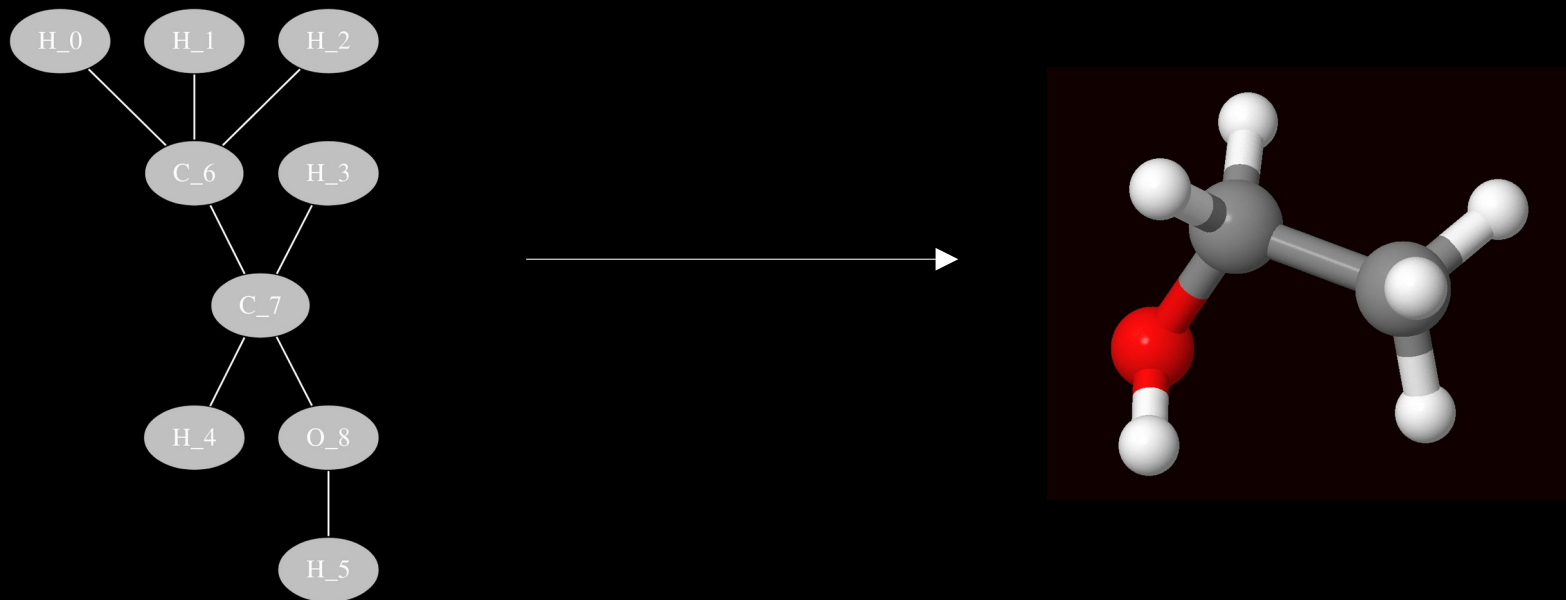
- Contraintes sur le nombre de liaisons de chaque atome : $\forall i \in V$,

$$\sum_{j=0}^{n-1} liaisons(i, j) = valence(i)$$

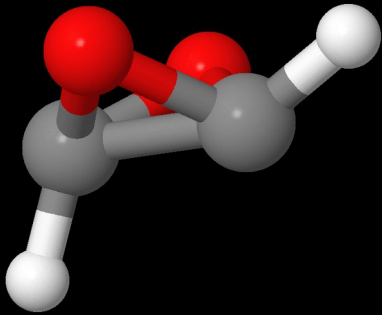
- Contraintes sur les arêtes du graphe en fonctions des liaisons : $\forall i, j \in V, i \neq j$,

$$liaisons(i, j) > 0 \Rightarrow (i, j) \in E$$

Deuxième modélisation : plongement dans l'espace

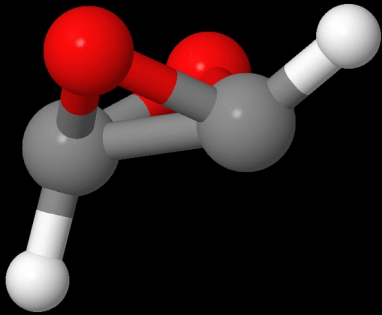


Deuxième amélioration : plongement dans l'espace



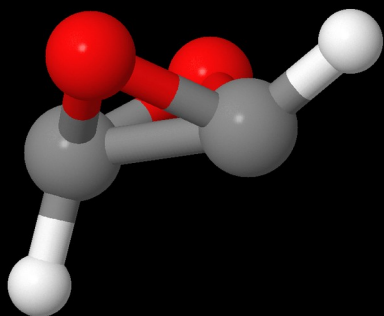
- Distance de liaisons : $[d_{\min} ; d_{\max}]$
→ Dépend du type d'atomes et du type de liaison

Deuxième amélioration : plongement dans l'espace



- Distance de liaisons : $[d_{\min} ; d_{\max}]$
→ Dépend du type d'atomes et du type de liaison
- Distance de non liaisons : $> d_{\max}$

Deuxième amélioration : plongement dans l'espace



- Distance de liaisons : $[d_{\min} ; d_{\max}]$
→ Dépend du type d'atomes et du type de liaison
- Distance de non liaisons : $> d_{\max}$

« Existe-t'il un placement dans l'espace respectant les contraintes de distances de liaisons/non liaisons ? »

Deuxième modélisation : plongement dans l'espace

- $X = \{x_i | i \in V\} \cup \{y_i | i \in V\} \cup \{z_i | i \in V\}$ où :
 - x_i est la coordonnée 1D du sommet i
 - y_i est la coordonnée 2D dimension du sommet i
 - z_i est la coordonnée 3D dimension du sommet i

Deuxième modélisation : plongement dans l'espace

- $X = \{x_i | i \in V\} \cup \{y_i | i \in V\} \cup \{z_i | i \in V\}$ où :
 - x_i est la coordonnée 1D du sommet i
 - y_i est la coordonnée 2D dimension du sommet i
 - z_i est la coordonnée 3D dimension du sommet i
- $D = \{d_{x_i} | i \in V\} \cup \{d_{y_i} | i \in V\} \cup \{d_{z_i} | i \in V\}$ où :
 - $d_{x_i} = [-300; 300], \forall i$
 - $d_{y_i} = [-300; 300], \forall i$
 - $d_{z_i} = [-300; 300], \forall i$

Deuxième modélisation : plongement dans l'espace

- $X = \{x_i | i \in V\} \cup \{y_i | i \in V\} \cup \{z_i | i \in V\}$ où :
 - x_i est la coordonnée 1D du sommet i
 - y_i est la coordonnée 2D dimension du sommet i
 - z_i est la coordonnée 3D dimension du sommet i
- $D = \{d_{x_i} | i \in V\} \cup \{d_{y_i} | i \in V\} \cup \{d_{z_i} | i \in V\}$ où :
 - $d_{x_i} = [-300; 300], \forall i$
 - $d_{y_i} = [-300; 300], \forall i$
 - $d_{z_i} = [-300; 300], \forall i$
- C
 - **Les contraintes sur les distances minimum entre chaque atomes liés :**
Si $(i, j) \in E$, alors $distance(i, j) \geq dist_min(type(i))(type(j))$

Deuxième modélisation : plongement dans l'espace

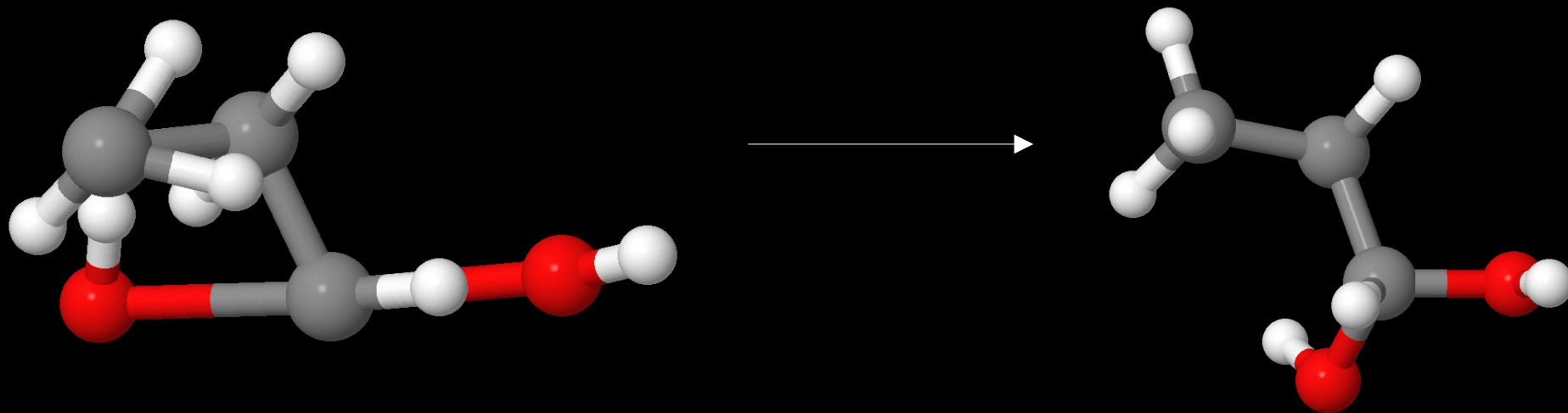
- $X = \{x_i | i \in V\} \cup \{y_i | i \in V\} \cup \{z_i | i \in V\}$ où :
 - x_i est la coordonnée 1D du sommet i
 - y_i est la coordonnée 2D dimension du sommet i
 - z_i est la coordonnée 3D dimension du sommet i
- $D = \{d_{x_i} | i \in V\} \cup \{d_{y_i} | i \in V\} \cup \{d_{z_i} | i \in V\}$ où :
 - $d_{x_i} = [-300; 300], \forall i$
 - $d_{y_i} = [-300; 300], \forall i$
 - $d_{z_i} = [-300; 300], \forall i$
- C
 - **Les contraintes sur les distances minimum entre chaque atomes liés :**
Si $(i, j) \in E$, alors $distance(i, j) \geq dist_min(type(i))(type(j))$
 - **Les contraintes sur les distances maximum entre chaque atomes liés :**
Si $(i, j) \in E$, alors $distance(i, j) \leq dist_max(type(i))(type(j))$

Deuxième modélisation : plongement dans l'espace

- $X = \{x_i | i \in V\} \cup \{y_i | i \in V\} \cup \{z_i | i \in V\}$ où :
 - x_i est la coordonnée 1D du sommet i
 - y_i est la coordonnée 2D dimension du sommet i
 - z_i est la coordonnée 3D dimension du sommet i
- $D = \{d_{x_i} | i \in V\} \cup \{d_{y_i} | i \in V\} \cup \{d_{z_i} | i \in V\}$ où :
 - $d_{x_i} = [-300; 300], \forall i$
 - $d_{y_i} = [-300; 300], \forall i$
 - $d_{z_i} = [-300; 300], \forall i$
- C
 - Les contraintes sur les distances minimum entre chaque atomes liés :
Si $(i, j) \in E$, alors $distance(i, j) \geq dist_min(type(i))(type(j))$
 - Les contraintes sur les distances maximum entre chaque atomes liés :
Si $(i, j) \in E$, alors $distance(i, j) \leq dist_max(type(i))(type(j))$
 - Les contraintes sur les distances minimum entre les atomes non-liés :
Si $(i, j) \notin E$, alors $distance(i, j) > dist_max(type(i))(type(j))$

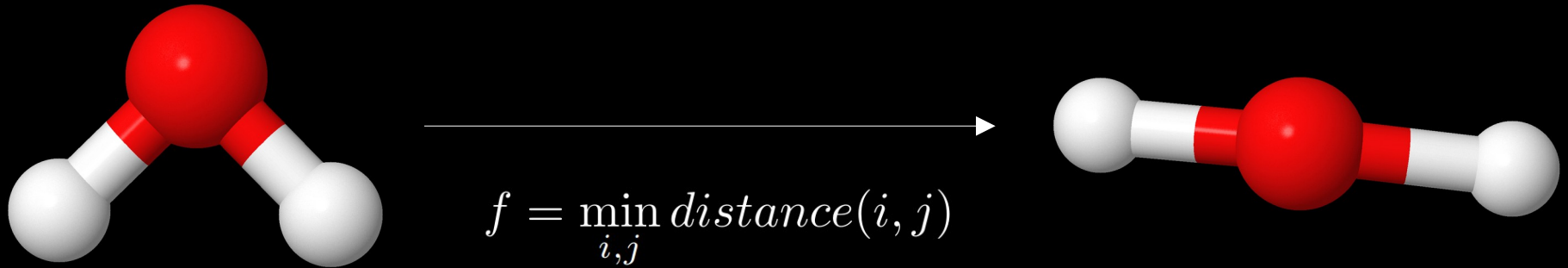
Deuxième modélisation : avec optimisation

- Stabilité moléculaire : limiter répulsions des électrons
→ modélisation pas assez poussée



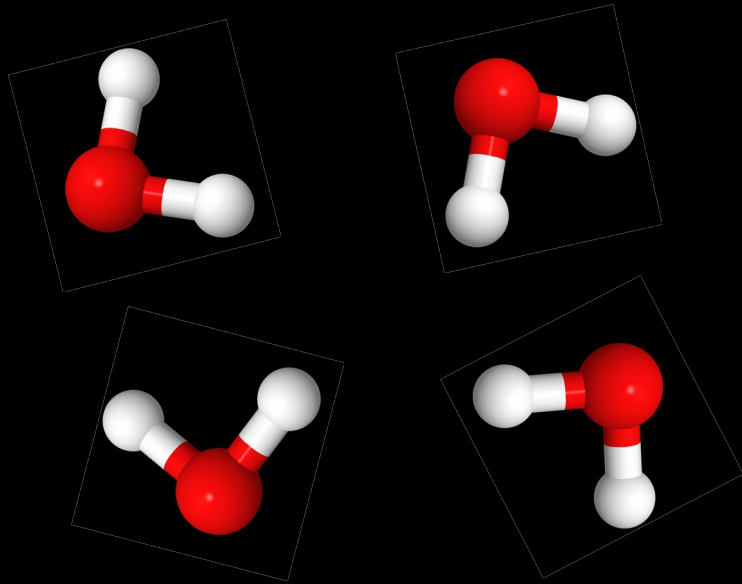
Deuxième modélisation : avec optimisation

- Optimisation pour maximiser les distances



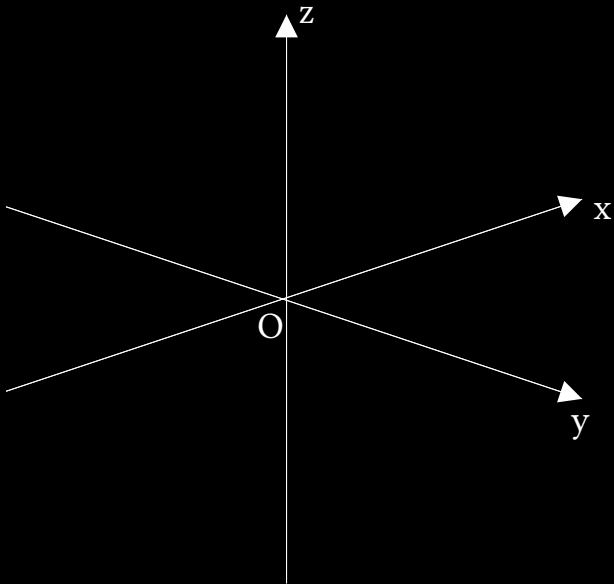
Deuxième modélisation : avec optimisation

- Problème de symétrie dans l'espace



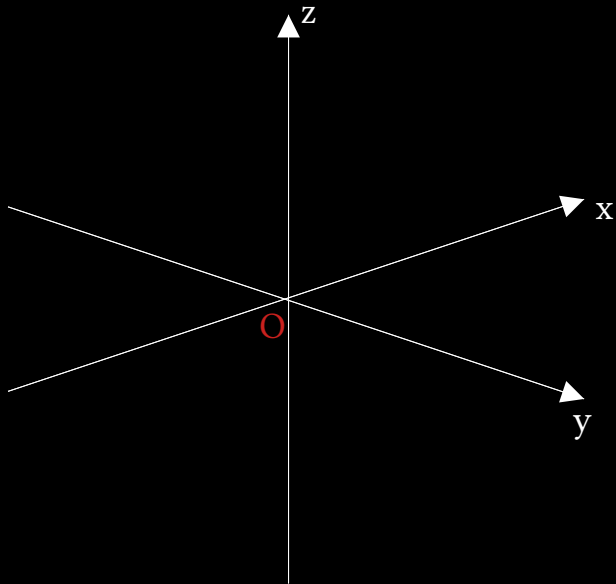
Deuxième modélisation : avec optimisation

- Problème de symétrie dans l'espace



Deuxième modélisation : avec optimisation

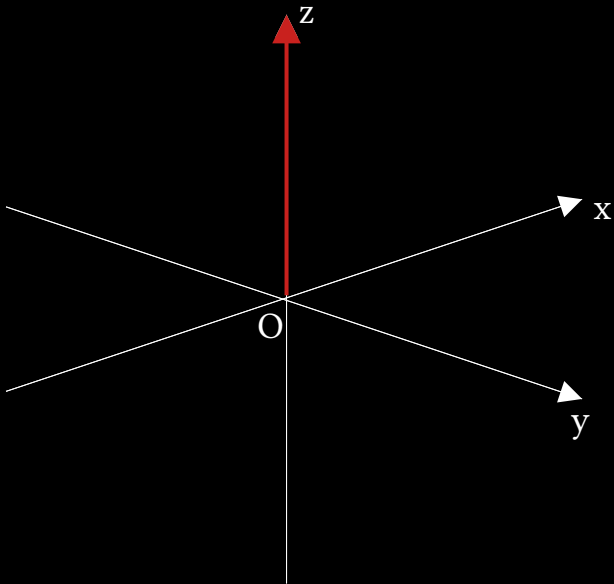
- Problème de symétrie dans l'espace



- $x(0) = 0, y(0) = 0, z(0) = 0$

Deuxième modélisation : avec optimisation

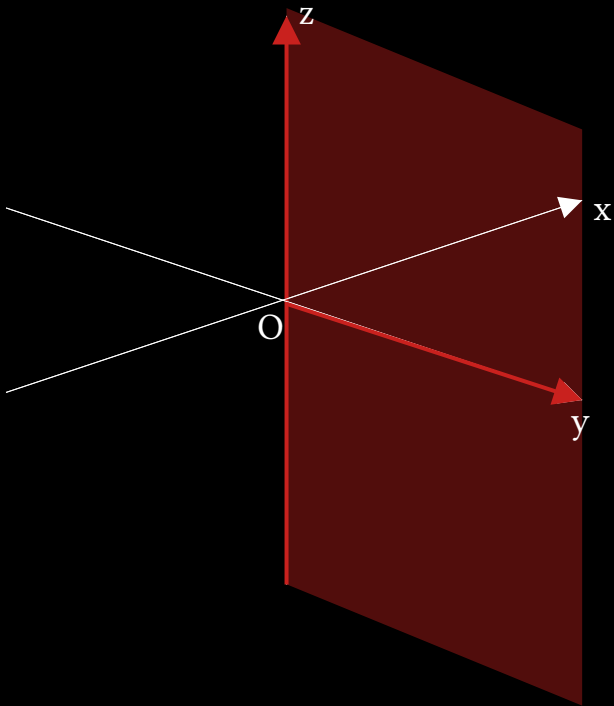
- Problème de symétrie dans l'espace



- $x(0) = 0, y(0) = 0, z(0) = 0$
- $x(1) = 0, y(1) = 0$ et $z(1) > 0$

Deuxième modélisation : avec optimisation

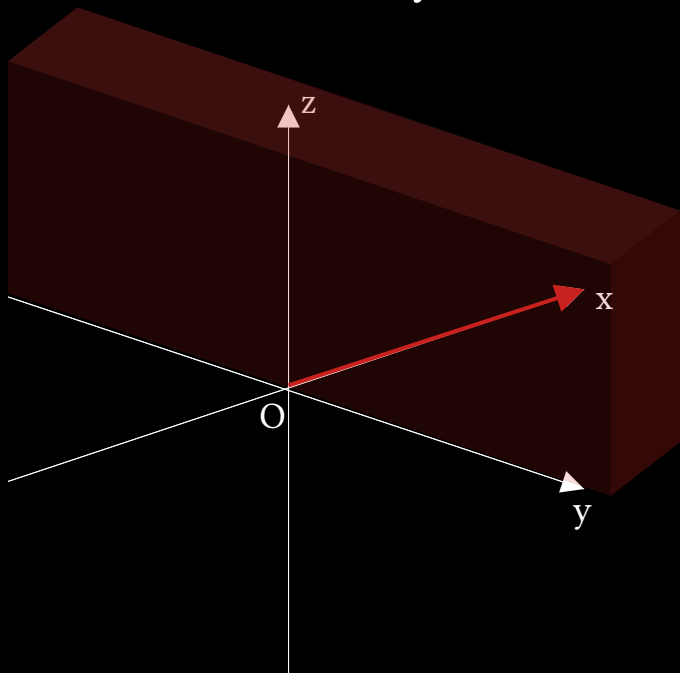
- Problème de symétrie dans l'espace



- $x(0) = 0, y(0) = 0, z(0) = 0$
- $x(1) = 0, y(1) = 0$ et $z(1) > 0$
- $x(2) = 0$ et $y(2) > 0$

Deuxième modélisation : avec optimisation

- Problème de symétrie dans l'espace



- $x(0) = 0, y(0) = 0, z(0) = 0$
- $x(1) = 0, y(1) = 0$ et $z(1) > 0$
- $x(2) = 0$ et $y(2) > 0$
- $x(3) > 0$

Détails d'implémentation

Côté code : les différentes classes

- Applications : *MainViz* et *MainSpace*
- Modelisation : *Modelisation*, *GraphModelisation*...
- Gérer les molécules : *Atom*, *AtomIndexer*, *Molecule_Utils*, *BondDistance*...
- Sortie et expériences : *CML_generator*, *InstanceMaker*, *ExperienceMaker*

Côté code : les différentes fonctionnalités

- Trouver tous les isomères d'une molécule
- Générer des fichiers CML
- Imposer des type de liaison
- Générer des graphes 2D

Format d'entrée

- **Exécution simple :**

→ Entrée sous forme de json

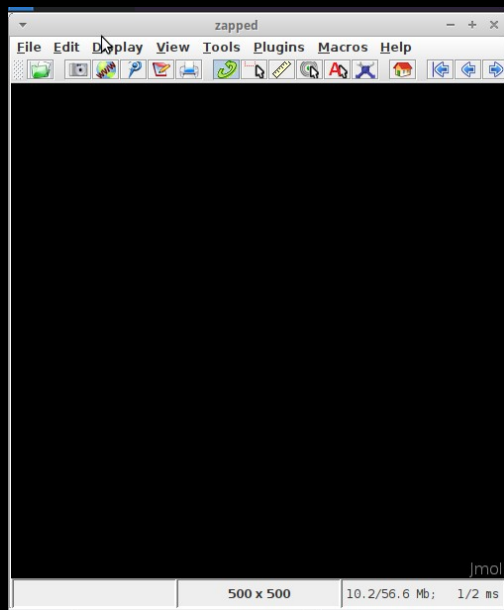
```
{  
  "types": [type1, ...],  
  "quantities": [nb_atome_type1, ...],  
  "structure": [[atome1, type_liaison, atome2], ...]  
}
```

- **Exécution multiple (experiences) :**

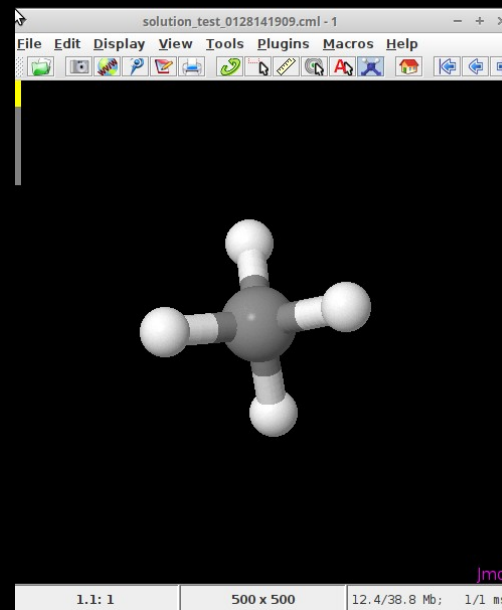
→ Entrée sous forme de dictionnaire

{“C:3,H:8,O:2”, “H:2,O:1”,...}

Pour la visualisation 3D : logiciel Jmol

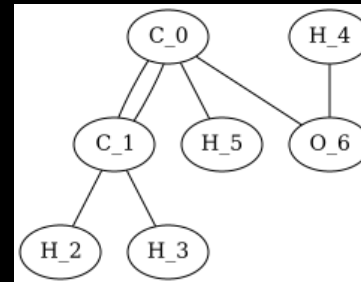
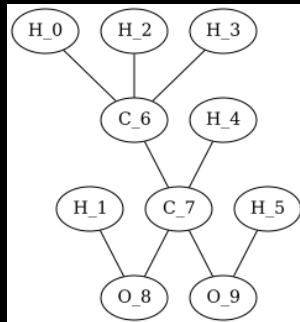


Fichier CML



Pour la visualisation des graphes : MainViz

- Librairie *graphviz*
- Expérimentation
- Vision plus globale : pas besoin de résolution de coordonnées



Protocole d'expérimentations

Package *experiences* :

- *ExperienceMaker.java*
- Choix entre *MainViz* ou *MainSpace*
- **Sortie** : nombre de structure de graphe, de solution (coordonées), temps d'exécution..

→ Démonstration

Côté implémentation : plus de détails

- Github :

https://github.com/Paulpey13/chemical_molecule_generation/

→ Javadoc & README

Résultats d'expériences : sans optimisation

Molécule	Nb isomère(s)	Nb graphe(s)	Temps total	Temps moyen par coordonnées
H ₂ O	1	1	406 ms	8 ms
CH ₄ O	2	4	521 ms	18 ms
C ₂ H ₄	2	6	516 ms	14 ms
C ₂ O ₂	4	5	636 ms	7 ms
CH ₄ O ₂	3	20	652 ms	23 ms
C ₂ H ₆ O	3	140	878 ms	71 ms
CH ₄ O ₃	4	120	1037 ms	80 ms
C ₂ O ₂ H ₂	10	44	1161 ms	10 ms
C ₃ O ₂	8	42	1335 ms	57 ms
C ₂ H ₂ O ₄	11	274	1712 ms	49 ms
C ₃ OH ₂	10	69	1961 ms	92 ms
H ₂ O ₆	2	720	2187 ms	781 ms
C ₄ O ₂	29	636	11881 ms	331 ms
C ₂ O ₂ H ₆	6	1120	14319 ms	1120 ms
C ₃ O ₂ H ₂	35	543	46567 ms	1252 ms
H ₂ O ₈	2	40320	185232 ms	90482 ms
C ₄ O ₂ H ₂	164	10488	522071 ms	3086 ms

Résultats d'expériences : avec optimisation

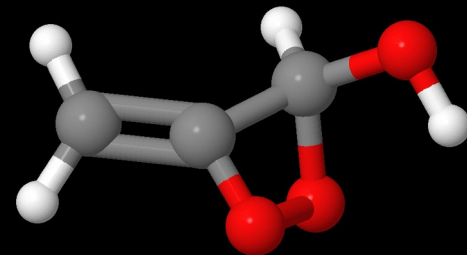
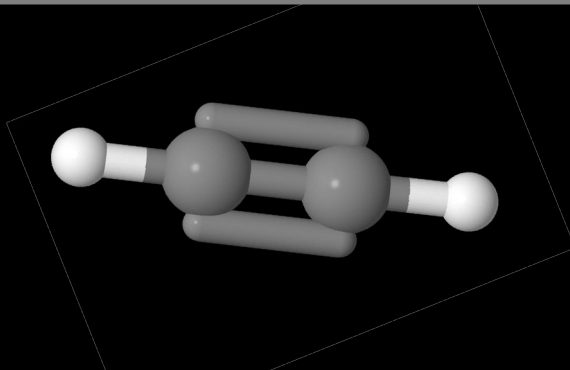
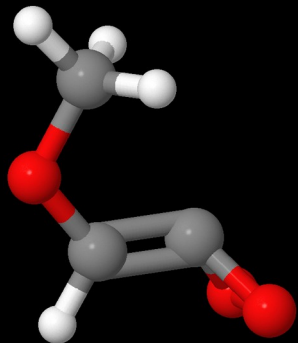
Molécule	Nb isomère(s)	Nb graphe(s)	Temps total	Temps moyen par coordonnées
H ₂ O	1	1	445 ms	31 ms
CO ₂	1	1	707 ms	283 ms
CH ₂ O	1	1	6988 ms	6569 ms
H ₂ O ₂	2	2	8787 ms	4152 ms
C ₂ H ₂ O	2	2	11744 ms	5631 ms

Conclusion

- Difficultés :
 - Documentation peu fournie
 - Isomorphisme de graphe
 - Allier informatique et chimie
- Travaux futurs :
 - Diminuer la combinatoire en ajoutant des contraintes
 - Interface graphique

Conclusion : ce qu'on a appris

- Programmation par contrainte
- Solveur choco
- Respecter des objectifs d'équipe
- Connaissances en chimie



Merci de votre attention.

