University of St Andrews | FOUNDED 1413

Combinatorics, Chemistry, and Crystals

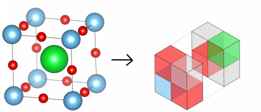
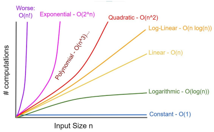
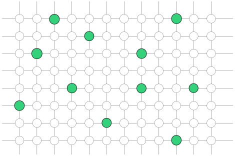
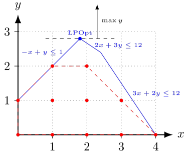
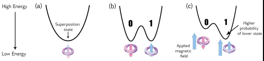
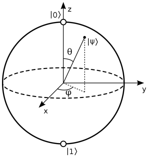
Slides at duncanadamson.github.io/talks/slides/LRC2024.pdf

Duncan Adamson

November 12, 2024

Computer Science and Chemistry

Current Interdisciplinary Links with Computer Science and New Perspectives

<table>
<tr>
<td>Crystals as Discrete Structures
</td>
<td>Computational hardness of CSP

</td>
<td>The k-Centre problem for crystals
</td>
</tr>
<tr>
<td>Integer & quadratic programming

</td>
<td>Quantum Computing

</td>
<td>
</td>
</tr>
</table>

Combinatorics for Chemists

- The mathematics of “counting”.

Combinatorics for Chemists

- The mathematics of “counting”.
- The mathematics of *discrete structures*.

Combinatorics for Chemists

- The mathematics of “counting”.
- The mathematics of *discrete structures*.
- The mathematics of optimisation.

Combinatorics for Chemists

- The mathematics of “counting”.
- The mathematics of *discrete structures*.
- The mathematics of optimisation.
- The mathematics of finding good structures.

Chemistry for Combinatorialists

- The science of substances.

Chemistry for Combinatorialists

- The science of substances.
- The science of:

Chemistry for Combinatorialists

- The science of substances.
- The science of:
  - acids,

Chemistry for Combinatorialists

- The science of substances.
- The science of:
  - acids,
  - conductors,

Chemistry for Combinatorialists

- The science of substances.
- The science of:
  - acids,
  - conductors,
  - batteries,

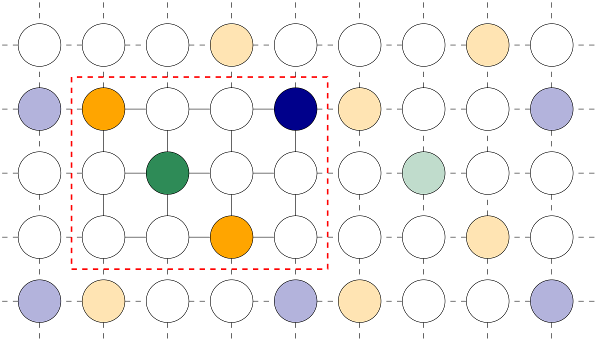
Chemistry for Combinatorialists

- The science of substances.
- The science of:
    - acids,
    - conductors,
    - batteries,
    - reactions,

Chemistry for Combinatorialists

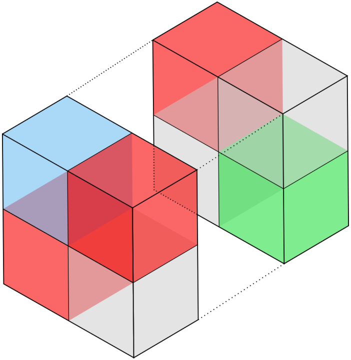
- The science of substances.
- The science of:
  - acids,
  - conductors,
  - batteries,
  - reactions,
  - *structures*.

Stuctures

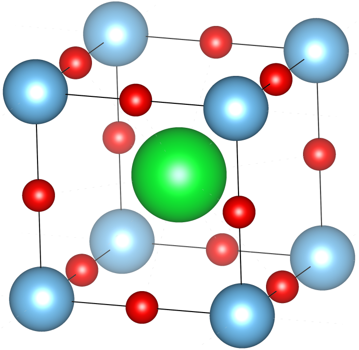


Structures

The natural intersection between chemistry and combinatorics is *structures*.



Combinatorial representation.



Chemical Representation.

Structures

- Computers are optimised for dealing with *discrete* structures.

```
>>> 0.1 + 0.2
0.30000000000000004
```

- This contrasts with the more natural representation, where chemical structures are defined in terms of the angles and length of bonds.
- We will focus on *inorganic crystal structures*, however much of these problems can be extended to other areas of computational chemistry.

Problem: How can we represent a crystal structure in discrete space?

Problem: How can we find good crystal structures with this representation?

Representing Crystals

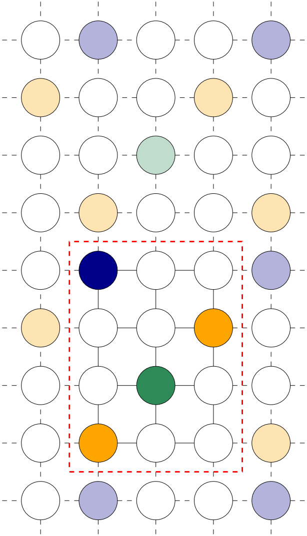
We represent crystal structures as a periodic motif defined by *translationally invariant three-dimensional array* over a *finite alphabet of symbols*.

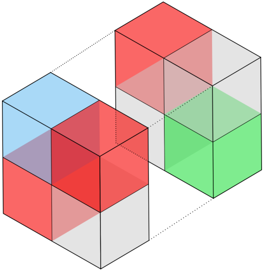
Periodic Motif: Crystals are, functionally, an infinite repeating structure. The periodic motif, also known as the *unit cell*, is the basic unit of this structure.

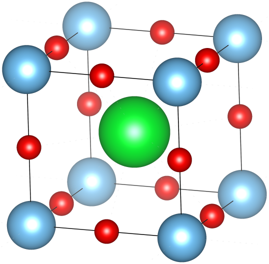
Three Dimensional Array: We represent the unit cell by a 3D grid, with each cell either empty or containing some ion.

Transitionally Invariant: We treat two arrays, A and B, as representing the same crystal if one can create A by shifting every ion in B be some vector $\overline{v}$.

Representing Crystals







Good Crystals

- A crystal is only going to exist in reality if the ions are going to hold together.
- As a metric to represent stability, we use *pairwise interaction*.

Pairwise Potential: A function computing the attraction (or repulsion) between a pair of ions at a given position in a unit cell.

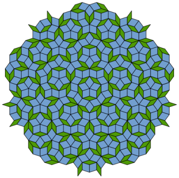
Crystal Structure Prediction Problem (CSP).
Goal: Minimise the sum of pairwise potential between every pair of ions in the unit cell.

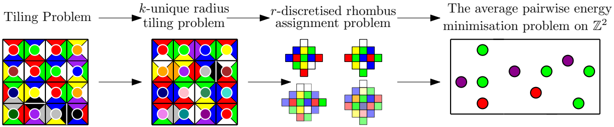
Hard Problems

Can we solve CSP?

Crystal Structure Prediction

In general, CSP is **undecidable**.

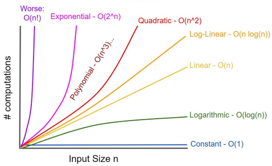




This means that no algorithm exists to solve this problem in general.

Fixed Unit cell CSP

If the size of the unit cell is fixed, then there is a finite number of possible structures. However, this remains **NP-Complete**, meaning that it is unlikely that this problem can be guaranteed to be solved optimally and quickly.



Solving CSP

Solving Hard Problems

- In general, we know we can not solve this problem quickly.

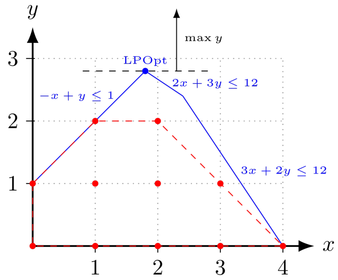
Solving Hard Problems

- In general, we know we can not solve this problem quickly.
- Fortunately, computer science has some tools to solve hard problems exactly quickly **most of the time**.

Solving Hard Problems

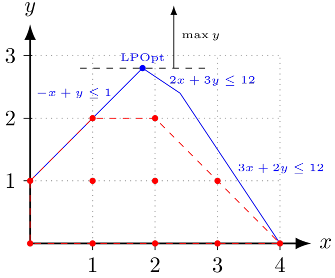
- In general, we know we can not solve this problem quickly.
- Fortunately, computer science has some tools to solve hard problems exactly quickly **most of the time**.
- We have used two such tools:
  - Integer Programming,
  - Quantum Computing.
- Both of these require us to represent crystal as *discrete structures*.

Integer Programming



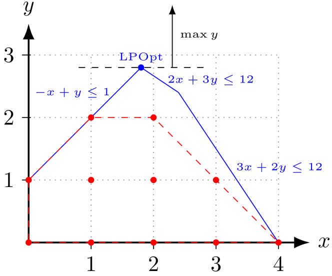
- **Integer Programming** provide a model for solving NP-hard problems **exactly**.
- In this problem, we represent CSP as an optimisation problem, using integer (binary) variables to represent the structure.

Integer Programming



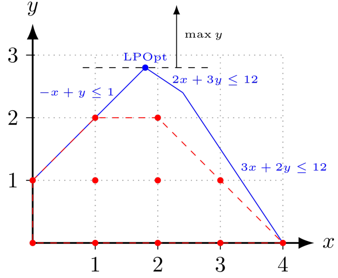
- We use a set of equations to constrain the space so that all **candidate solutions** satisfy some basic conditions regarding the composition and common sense (ions can't overlap).

Integer Programming



- Finally, we optimise over the total pairwise energy.
- This gives a relatively fast way of exactly solving even hard problems using highly studied tools.

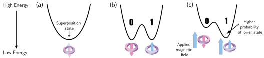
Integer Programming

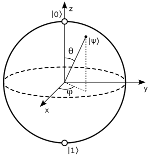


- Finally, we optimise over the total pairwise energy.
- This gives a relatively fast way of exactly solving even hard problems using highly studied tools.
- A successful proof of concept of integer programming for crystal structure prediction was published in Nature.

Quantum computing

- Quantum computing is the frontier in solving hard problems.
- We have restructured our integer programming model to work on the D-Wave quantum machine.
- Our goal is to make **quantum ready** algorithms.
    - This means algorithms that work with what we have now, and will get better with better computers.





What next?

Quantum Ready Algorithms

- **Quantum Computing:** Even if it is always 10 years away, we want to hit the ground running when large scale quantum computing becomes a reality.
  - **Goal:** Develop **hybrid algorithms**, algorithms that can work on classical computers, while offloading a certain amount of work to quantum computers.
  - The amount offloaded will scale with the power of the quantum machines.
  - In the short term, we will be able to use the algorithms immediately.
  - In the long run, this will allow us to have a pure quantum algorithm.

Classical Gaurentees

- There are two other approaches in TCS that can give soloutions to hard problems.

Classical Gaurentees

- There are two other approaches in TCS that can give soloutions to hard problems.
- **Approximation Algorithms with Guarantees:**
  - **Goal:** Develop **approximation algorithms**, where we don't know if we can get the optimal answer, but we know that we can get an answer that is "good enough" in a short amount of time.
  - "Good enough" means we have some known upper bound on how far away from the optimal we are.

Classical Gaurentees

- There are two other approaches in TCS that can give soloutions to hard problems.
- **Approximation Algorithms with Guarantees:**
  - **Goal:** Develop **approximation algorithms**, where we don't know if we can get the optimal answer, but we know that we can get an answer that is "good enough" in a short amount of time.
  - "Good enough" means we have some known upper bound on how far away from the optimal we are.
- **Parameterised Algorithms:**
  - CSP is hard in theory, but only for very specific instances.
  - This means we can't guarantee that we can solve the problem in general, but we might be able to solve it for limited settings.
  - A parameterised algorithm is an algorithm where we know we can solve the problem, if we limit the instance in some way (e.g., only using specific ion species).

Summary

- CSP is very hard to optimise in general, but TCS has tools to solve hard problems optimally.
- Three open fronts for future research:
  - Quantum Computing,
  - Approximation Algorithms,
  - Parameterised Algorithms.