# Bond-Energy-Guided Program Synthesis for Chemical Reaction Network Discovery

**Thomas Merlijn Bood**[1]
**Supervisor(s): Dr. Sebastijan Dumančić**[1]**, Reuben Gardos Reid**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Thomas Merlijn Bood
Final project course: CSE3000 Research Project
Thesis committee: Dr. Sebastijan Dumančić, Reuben Gardos Reid, Jana Weber

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

**Abstract**

Chemical Reaction Network (CRN) discovery is a time-consuming task that can be automated using program synthesis. However, the search space for realistic CRNs is large, making exhaustive search intractable. This paper investigates whether incorporating bond-breaking energy as a search heuristic can improve the efficiency of grammar-based CRN discovery. This paper proposes two heuristics: a maximum-bond-order heuristic and a more sophisticated delta-energy heuristic, which prioritise reactions with lower net energy change. Both are benchmarked against naive breadth-first search across an entire synthesis pipeline and the three individual stages, for seven Chemical Reaction Networks: water formation, methane combustion, photosynthesis, ethylene glycol formation, methyl acetate hydrolysis, an esterification reaction, and fermentation of glucose. Results show that both the Delta-Energy and Max-Bond heuristics consistently reduce the number of candidates explored and total runtime compared to BFS, with Delta-Energy generally outperforming the Max-Bond heuristic. However, neither heuristic guarantees improvement in all cases. These findings suggest that energy-guided search is a promising direction for scalable CRN discovery, with more accurate bond-energy estimation being a natural avenue for future work.

# 1 Introduction

When chemists need to predict the behaviour of chemicals, optimise reaction conditions for industrial processes, and design more efficient synthetic routes for pharmaceutical compounds, they make use of Chemical Reaction Networks (CRNs). Chemical Reaction Networks are mathematical models that describe how chemical species interact according to a set of reactions. However, manually determining these models for large and complex systems can be a time-consuming and difficult task. Therefore, advancements are being made to automate the discovery of CRNs [5].

From a computer science perspective these Chemical Reaction Networks (CRNs) can be considered stochastic programs, where reactions are probabilistic events governed by reaction rates and molecular concentrations [2]. The execution of such a stochastic program gives a concentration profile that shows how the concentration of each chemical species changes over time. For CRN discovery we look at the inverse of the problem: given a set of observed concentration profiles from experiments, determine the possible underlying chemical reaction network that generated them.

Program synthesis, the process of deriving programs that satisfy certain behaviour, offers a principled approach to automating CRN discovery as the resulting concentration profile defines the required behaviour. In previous works the use of program synthesis for CRN discovery has already been explored with success [2, 6].

These works still face challenges for larger networks, due to the quickly scaling nature of the search space: as the number of species and reactions grows, the number of potential networks explodes, making exhaustive search intractable. Without any guidance, the search has to treat each potential candidate equally, without any regard to the plausibility or likelihood of the candidate. Integrating domain knowledge in the synthesis process addresses this problem by restricting or guiding the search to more plausible solutions. However, there remains a vast amount of domain knowledge, such as bond-breaking energy, whose principled integration within a synthesis framework is both under-explored and practically valuable. Heuristics to estimate bond energetics can offer a way to guide the search within program synthesis to more energetically favourable solutions.

This paper aims to investigate whether a bond-energy heuristic can improve the efficiency of CRN discovery via program synthesis. This is done by answering the following question:

> Can bond-energy-guided heuristic search in program synthesis of chemical reaction networks reduce the number of evaluated candidates and total runtime before the target network is found compared to uninformed search?

Showing a reduction in the number of candidates explored or runtime required to find the target network, would indicate that the synthesiser is more likely to find the target even when search cannot be done exhaustively. This is important because for real-world problems the search space for CRNs will often be too large for exhaustive search.

The rest of this paper is structured as follows, chapter 2 provides the necessary background on CRNs and bond energy, followed by an overview of program synthesis and its previous applications to CRN discovery. Next, chapter 3 will outline the Max-Bond and Delta-Energy heuristics and their implementation. Chapter 4 presents the environment and benchmarks used, followed by an analysis of the baseline breadth-first search against the heuristics guided search across the full pipeline and individual stages. The implications and interpretation of the research in a broader context are discussed in chapter 5. The limitations, assumptions, and scientific quality are addressed in chapter 6. Chapter 7 concludes that energy-guided search can improve CRN synthesis and formulates recommendations for future research.

# 2 Background and Related Work

The following section defines Chemical Reaction Networks (CRNs) and bond energy, outlines the mechanics of program synthesis, and reviews its previous applications to CRN discovery.

**Chemical Reaction Networks as Stochastic Programs** Chemists use Chemical Reaction Networks (CRNs) as a way to describe how a set of chemical species interact and transform by defining reactions and respective rates. A CRN consists of a set of chemical species (the molecules), and a set of reactions that describe how one set of species turns into another set of species. For example, consider the following simple CRN:

$$2\,\mathsf{H_2} + \mathsf{O_2} \longrightarrow 2\,\mathsf{H_2O} \tag{1}$$

This CRN consists of a single reaction, and has three species ($\mathsf{H_2}, \mathsf{O_2}, \mathsf{H_2O}$), the species on the left side of the arrow (in the example $\mathsf{H_2}, \mathsf{O_2}$) are called the reactants and the species on the right side ($\mathsf{H_2O}$) are called the products.

From a computer science perspective these CRNs can be considered stochastic programs, where reactions are probabilistic events governed by reaction rates and molecular concentrations [2]. The likelihood of a reaction occurring at any given moment (also known as the propensity of a reaction) is then determined by a function of the current molecular counts and a rate constant. The output of executing such a stochastic program is a concentration profile: a time series showing how the abundance of each chemical species changes over time. Taking the water synthesis example from before and assuming a reaction rate of 0.2. This roughly means that whenever an $\mathsf{O_2}$ and two $\mathsf{H_2}$ collide there is a 20% probability of a reaction occurring and reactants turning into products. The corresponding concentration profile for this is example is shown in Figure 1.

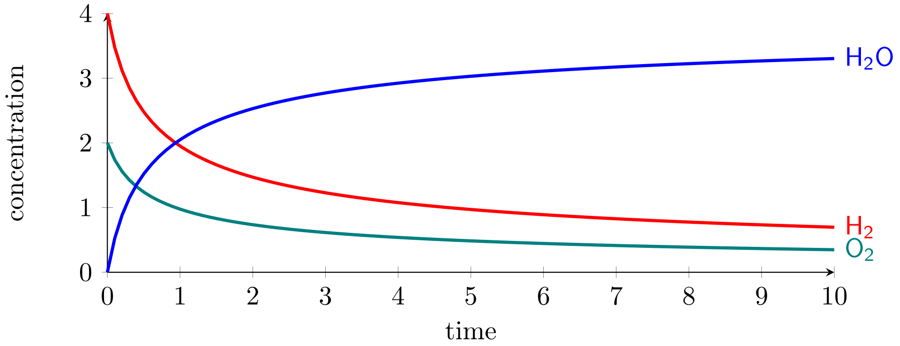

Figure 1: Concentration profile for $2\,H_2 + O_2 \longrightarrow 2\,H_2O$ with reaction rate of 0.2. The $H_2$ and $O_2$ concentrations decrease as the $H_2O$ concentration rises.

This research focusses on the problem of CRN discovery, which is the inverse problem: given observed concentration profiles from experiments, can we automatically synthesize the possible underlying chemical reaction network that generated them?

**Bond Energy** Whenever a chemical reaction occurs, molecules react and certain bonds are broken and new bonds are formed. The energy required to break a bond between two atoms is called the Bond dissociation enthalpy (BDE). The energy of molecular bonds can have a large impact on the reactivity of molecules [4]. Determining exact BDE values is difficult as they are influenced by numerous factors, including the involved atoms, the arrangement and even the environment [4]. Therefore, chemists often work with standard bond-energy values, which are average BDE values for a specific bond type.

**CRN Discovery via Program Synthesis** Program synthesis is the process of searching over candidate programs to derive programs that satisfy certain behaviour [3]. In order to do this we define 4 different components:

- **A grammar** that defines the search space to iterate over.
- **Constraints** that introduce rules based on domain knowledge to limit the search space by eliminating infeasible results.
- **A search strategy** that defines the order in which the search tree is traversed.
- **An example set** to define the expect behaviour of the solution.

With these components combined, the search strategy allows for iteration over all programs defined by the grammar and constraints. Every candidate program is then compared to the example set, and if the behaviour corresponds to the examples we have found a potential solution.

One of the main search strategies is top-down search. Top-down search builds candidates incrementally by expanding the grammar rules step by step. During the search partially defined candidates still contain holes: sub-expressions that have not yet been expanded into concrete values. For example when synthesising a molecule there may be a hole for an atom, meaning you know there will be an atom, but all concrete options still need to be explored. Each iteration of the search process expands a hole in the next candidate of the search queue by replacing it with its possible completions. This either results in a fully defined program (one with no holes left) to be returned or a new partial representation to add into the search queue and continue to the next iteration. There are alternative search strategies, such as bottom-up search. However, this paper is limited to only using top-down based approaches.

CRN discovery can be seen as recovering a program (the CRN) that specifies some expected behaviour (the concentration profile), this makes it a logical and useful application for program synthesis. Program synthesis techniques also make it possible to reduce the space of reaction networks by encoding domain-specific constraints such as mass conservation. Through simulation of the candidate networks it is possible to verify whether they reproduce the observed dynamics.

### Prior work

Researchers have already shown that syntax guided synthesis can be used to automate CRN discovery [2]. In their paper the researchers encode the entire synthesis problem as a Satisfiability Modulo Theories problem, that can then be solved using a SMT solver. The authors encode the problem via a proposed sketching language for CRNs: a language to describe partially defined CRNs by allowing components such as species or reaction rates to be defined as unknown variables. The goal of the synthesis can then be seen as filling in these unknown values such that it results in a feasible CRN. Feasibility is defined by a set of desired conditions such as having all defined species be part of a reaction in the network. All candidates are then assigned a score to determine how 'good' a CRN is. The researchers then look for the CRN with minimal cost amongst all feasible candidates to find an optimal solution.

In his paper, Wijers proposes a grammar-based system for CRN synthesis that this research builds on directly [6]. The system structures CRN discovery according to a staged pipeline, with each stage synthesising candidates to produce the inputs for the next stage. In the first stage candidate molecules are generated based on a simplified version of the openSMILES[1] grammar: A widely used notion to represent molecular structures as strings. The second stage uses the synthesised molecules, together with already known molecules, as inputs to synthesise balanced reactions. The final stage assembles networks by selecting a subset from the synthesised reactions. Search across all three stages is done using a top-down breadth-first search iterator. In this prototype Wijers also implements some basic constraints including atom valence, or a consumption constraint ensuring molecules whose concentrations drop are a reactant for some selected reaction. The candidate networks are evaluated by a function based on the desired traits, such as comparing simulation results to the concentration profiles of the target network. This approach has been shown to perform well for small cases. But due to the fast scaling nature of the search space for larger networks, this system has not yet been proven to scale.

## 3 Search Heuristics for Low Energy Bond Breaking

Synthesising a CRN requires searching through a large space of candidate molecules, reactions, and networks. Searching without guidance means all candidates are treated as equally promising. However, in practice not all reactions are as equally likely. Bond energies have a large effect on the reactivity of molecules [4]. Incorporating this domain knowledge as a heuristic allows the search to prioritise more energetically favourable candidates.

---

[1] OpenSMILES specification http://opensmiles.org/

This section presents two heuristics that estimate reaction energy and describes how they are integrated in the existing system. Both heuristics use bond order as a proxy for energy. The first heuristic only looks at the maximum bond-order of the broken bonds. The second heuristic computes a delta-energy estimate as the difference between the sum of the bond energies in all reactants and all products.

## 3.1 Maximum Bond-Order Heuristic

The first heuristic that will be explained is that of a simple, maximum bond-order implementation. This heuristic is based on the principle that how likely a reaction occurs is often strongly influenced by the hardest bond to break. Since the bond order of a bond generally correlates with the required energy to break that bond, this heuristic will look at all bonds that get broken and estimates the likelihood based on the max-order of all the broken bonds. This means that reactions that only break single bonds will be prioritised before breaking double bonds. This can be written as follows:

$$E_1 = \max_{b \in B} f(b)$$

Here $E_1$ is the estimated reaction energy, $B$ the set of bonds in the reactants, and $f$ is a function that returns the bond order of a specific bond.

In order to maintain admissibility, holes (non-terminal grammar rules in a node) have a value of zero. This way they represent a lower bound and will never overestimate the highest bond-order. Furthermore, the system falls back to breadth-first search whenever a node does not have any bonds defined. This is done to allow the search strategy to differentiate nodes initially until any bonds are defined.

## 3.2 Delta-Energy Heuristic

The next heuristic to be considered looks at the $\Delta E$ of a reaction. This heuristic is based on the concept that exothermic reactions are more energetically favourable, so reactions with lower or negative $\Delta E$ are explored first. In this heuristic $\Delta E$ is defined as the total bond energy of the reactants minus the total bond energy of all products. In order to determine the energy of a certain molecule every bond gets assigned an estimated bond energy. The sum of all bond energies will then be assigned to the molecule as the total energy of the molecule. In an ideal case the estimated bond energy would correspond with the bond dissociation enthalpy, however this value is difficult to compute as it is influenced by numerous factors [4]. Therefore we use estimations based on an average dissociation enthalpy for each bond order. Since the current system only supports H, O, and C atoms we limit the averages to only bonds that include these atoms. Taking the average KJ/mol for all standard bond dissociation enthalpy values involving the atoms Hydrogen, Oxygen, or Carbon for the respective bond order, we get: 361 for single bonds, 617 for double bonds, 838 for triple bonds [1, Table 11.3b]. The reactions are then prioritised such that lower $\Delta E$ values are explored first. This can be written as follows:

$$E_2 = \sum_{b \in B_R} f(b) - \sum_{b \in B_P} f(b)$$

Here $E_2$ is the estimated $\Delta E$ reaction energy, $B_R$ the set of bonds in the reactants, $B_P$ the set of bonds in the products, and $f$ is a function that returns the estimated energy of a specific bond.

In order to differentiate between reactions with a similar energy estimate, we make the simplifying assumption that larger reactions consisting of more molecules are less likely to occur. So in the heuristic we prioritise partial candidates with fewer holes. This is done by returning a negative value for holes in the reactants and a positive value for holes in the products, encouraging holes to be filled more quickly. By keeping this value small compared to the estimated energy values, search will primarily be guided by delta-energy and secondary by the number of holes.

### 3.3 Herb.jl Heuristic Integration

This research builds on the prototype system developed by Wijers [6]. This prototype is built using the Herb.jl framework[2]. In the Herb.jl framework top-down iterators are defined using a priority queue. The order of the priority queue is dependent on a priority function. By updating the priority function to correspond with the heuristics, the order of the priority queue is changed and thus leads to a different order in which the search space is explored. The iterator's priority function is updated to return the heuristic energy value of the corresponding (partial) representation of a molecule, reaction, or network, potentially containing holes to indicate some not yet explored representations. This also means that it is possible there are no bonds defined, in this case the maximum bond order returned would be 0. This case is handled by falling back to BFS: the returned value is a constant amount more than the parent node's priority value.

## 4 Experimental Setup and Results

This section first details the environment and benchmarks used in the experiments, followed by the results to answer four research questions. The main research question of this paper is whether energy-guided search can reduce the number of explored candidates and total runtime compared to uninformed search. To answer this, we evaluate each stage of the pipeline individually and end-to-end based on the following questions:

**Q1:** Do energy-guided heuristics reduce the number of candidates explored and total runtime across the full synthesis pipeline?

**Q2:** Do energy-guided heuristics reduce the number of candidates explored and total runtime of the Atoms → Molecules stage?

**Q3:** Do energy-guided heuristics reduce the number of candidates explored and total runtime of the Molecules → Reactions stage?

**Q4:** Do energy-guided heuristics reduce the number of candidates explored and total runtime of the Reactions → Networks stage?

### 4.1 General Setup

In order to determine the effectiveness of each heuristic we compare three different search strategies: the baseline breadth-first search (BFS) against search guided by the *Max-Bond* or *Delta-Energy* heuristic. The evaluation covers all individual pipeline stages, as well as a

---

[2] Herb.jl: https://herb-ai.github.io/

full integrated pipeline, across seven benchmark problems:

- **Water** formation, with $O_2$ unknown: $2H_2 + O_2 \longrightarrow 2H_2O$
- **Methane** combustion, with $CH_4$ unknown: $CH_4 + 2O_2 \longrightarrow 2H_2O + CO_2$
- **Photosynthesis**, with $CO_2$ unknown: $6CO_2 + 6H_2O \longrightarrow C_6H_{12}O_6 + 6O_2$
- **Hydrolysis** of methyl acetate, with $H_2, CH_3OH$ unknown:
  $CH_3COOCH_3 + H_2O \longrightarrow CH_3COOH + CH_3OH$
- **Ethylene** glycol, with $C_2H_4O$ unknown: $H_2O + C_2H_4O \longrightarrow C_2H_6O_2$
- **Esterification**, with $H_2O, CH_2O_2, CH_3OH$ unknown: which is a network consisting of two reactions
  - $CH_2O_2 + C_2H_6O \longrightarrow C_3H_6O_2 + H_2O$
  - $C_3H_6O_2 + CH_3OH \longrightarrow C_2H_4O_2 + C_2H_6O$
- **Fermentation** of glucose, without any unknown: which is a network consisting of four reactions [3]
  - $C_6H_{12}O_6 + 2NAD \longrightarrow 2C_3H_4O_3 + 2NADH, H^+$
  - $C_3H_4O_3 + NADH, H^+ \longrightarrow CH_3CHOHCOOH + NAD$
  - $C_3H_4O_3 \longrightarrow CH_3CHO + CO_2$
  - $NADH, H^+ + CH_3CHO \longrightarrow CH_3CH_2OH + NAD$

For each benchmark we track the number of candidates synthesised and total runtime before the target network is found. For the initial molecules synthesis stage we distinguish between the total number of candidates explored and the number of unique candidates. The molecule grammar allows for symmetries and thus can synthesise structurally equivalent molecules multiple times. The total number gives a better overview of the amount of work done by a strategy, but the unique count better represents the output of the stage, as only the set of unique molecules is passed to the next stage. The reaction and network grammars have good symmetry-breaking constraints, leading to the total and unique counts being equal in these stages. For the full pipeline results in Table 1, only the number of unique synthesised molecules combined with the already known molecules is reported, as these are the molecules passed to the next stage. The corresponding total counts are still available in Table 2.

All runs have a 10-minutes time limit before the run is considered to have timed out. All reported times are the median of five runs whenever no time out occurred. All experiments are run on an AMD Ryzen 7 5800H. For the results of the full pipeline and all individual stages the molecule synthesiser uses a maximum search depth of 10, the reactions synthesiser a maximum depth of 20 and the network synthesiser a maximum depth of 6.

## 4.2 Q1: Full Pipeline

Both heuristics show a reduction in explored candidates and total runtime across most benchmarks, with Delta-Energy achieving the biggest improvements.

The full pipeline runs all three stages with the same strategy end-to-end: first all unknown molecules are synthesised from atoms, reactions are assembled from those molecules, and full networks are built using those reactions. Table 1 shows the total number of unique candidates resulting from each stage of the pipeline.

---

[3] NAD and $NADH, H^+$ are simplified in the experiments to still simulate the hydrogen transfer, as the current system does not support the full representation of these molecules

Table 1: Unique number of candidates resulting from each stage in the full pipeline (molecules (union of synthesised and known) | reactions | networks).

| Problem | Strategy | Molecules | Reactions | Networks |
|---|---|---:|---:|---:|
| Water | BFS | 3 | 1 | 1 |
| | Max-Bond | 3 | 1 | 1 |
| | Delta-Energy | 3 | 2 | 1 |
| Methane | BFS | 23 | 5713 | 7 |
| | Max-Bond | 16 | 525 | 2 |
| | Delta-Energy | 8 | 42 | 2 |
| Photosynthesis | BFS | 5 | 15 | 3 |
| | Max-Bond | 5 | 15 | 3 |
| | Delta-Energy | 5 | 13 | 3 |
| Hydrolysis | BFS | 66 | 47106 | 8 |
| | Max-Bond | 50 | 14095 | 8 |
| | Delta-Energy | 22 | 431 | 6 |
| Ethylene | BFS | 292 | 6301 | 3 |
| | Max-Bond | 289 | 6342 | 4 |
| | Delta-Energy | 244 | 6842 | 3 |
| Esterification | BFS | 67 | 44954 | 2001 |
| | Max-Bond | 51 | 13098 | 1235 |
| | Delta-Energy | 23 | 420 | 138 |
| Fermentation | BFS | 8 | 69 | 2912 |
| | Max-Bond | 8 | 60 | 1592 |
| | Delta-Energy | 8 | 70 | 3063 |

Looking at the total runtimes in Figure 2, a few observations stand out. Delta-Energy dominates most problems. For Methane, Delta-Energy completes in 0.0226 s versus 2.24 s for BFS (99× faster). For Hydrolysis, Delta-Energy takes 0.11 s versus 21.2 s for BFS (192× faster). For Esterification, Delta-Energy takes 0.16 s versus 119.03 for BFS (744× faster). In all of these cases, Delta-Energy explores far fewer candidates at every stage. The smaller molecule pools synthesised by Delta-Energy (8 vs. 23 for Methane; 22 vs. 66 for Hydrolysis; 23 vs. 67 for Esterification) directly restrict the reaction search space, amplifying the heuristic's effect in later stages.

The performance of Max-Bond generally sits between Delta-Energy and BFS, showing some more moderate improvements over BFS. The exception to this is the fermentation problem, here Max-Bond performs the best out of all three, being nearly twice as fast with 1.78 s compared to BFS which is second fastest at 3.32 s. This is likely due to the fact that fewer reactions are synthesised and it thus has a smaller search space when synthesising the networks.

The fermentation problem also shows that Delta-Energy is not guaranteed to be the fastest and is actually the slowest of the three strategies. However, it still performs similar to BFS, synthesising only one extra reaction (70 vs. 69) and having a slightly slower runtime of 3.42 s instead of BFS 3.32 s.

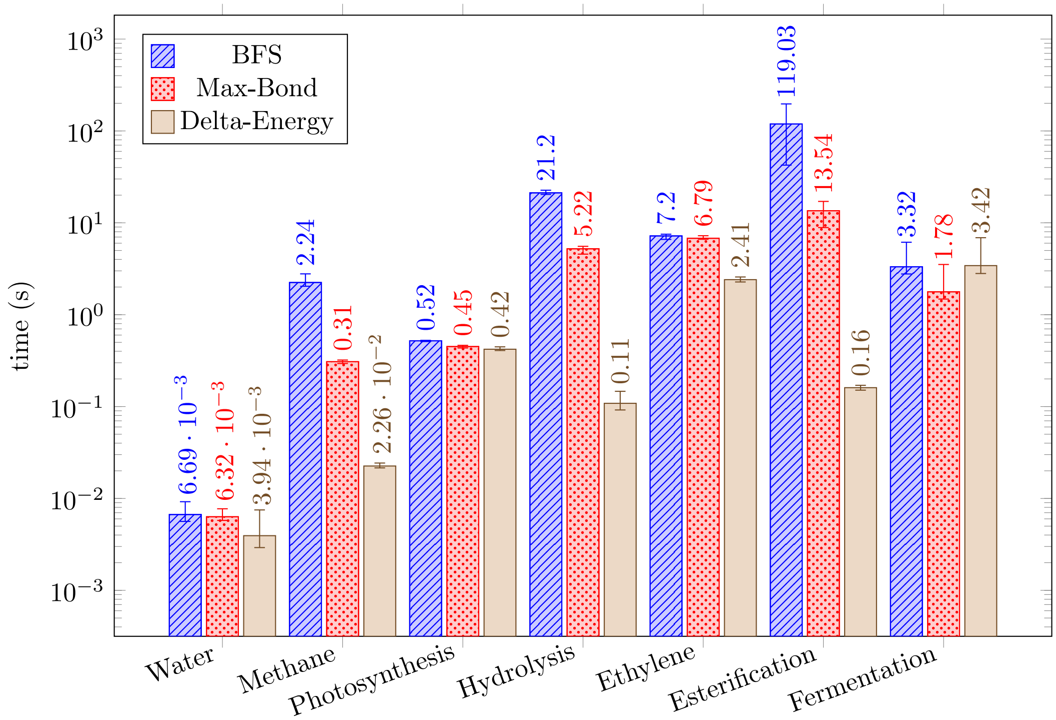

Figure 2: Median runtimes for the full Atoms → Molecules → Reactions → Networks pipeline. Note the logarithmic $y$-axis. Error bars show 95% confidence intervals.

### 4.3 Q2: Atoms to Molecules

Delta-Energy shows a consistent reduction in both total and unique candidates explored as well as reduced runtimes over BFS, while Max-Bond achieves more moderate improvements.

This stage of the pipeline synthesises the unknown molecules from the set of atoms in the known molecules. Table 2 reports the number of candidates explored and the unique number of candidates before all target molecules are found. The number of unique candidates can be different from the total number of explored candidates as symmetries in the grammar allow for the same structure to be explored multiple times. The fermentation problem has been excluded from the results in this step as there are no unknown molecules.

Delta-Energy consistently explores the fewest number of unique candidates. For the methane problem Delta-Energy reduces the total candidates explored from 72 (for BFS) to 12, a six times reduction and unique candidates go from 23 to 8 a nearly three times reduction. The Photosynthesis problem is the only case where Delta-Energy synthesises more total molecules than BFS, 5 instead of 4. However, the number of unique molecules still remains the same across all search strategies. Hydrolysis shows similar reduction as the methane problem, a 6.72× total reduction (222 to 33) and 3.2× unique (64 to 20). Max-Bond provides a smaller improvement over BFS, reducing the total candidates for the Methane problem to 36 and for Hydrolysis to 190.

A notable observation is that the Hydrolysis and Esterification problems produce identical results. This is because for both problems the synthesiser finds $CH_3OH$ last and thus terminates after synthesising the same set of molecules.

Table 2: Candidate molecules explored in the Atoms → Molecules stage. “Total” includes all revisited candidates; “Unique” counts distinct structures. The Fermentation problem is omitted, as there are no unknown molecules.

| | BFS | | Max-Bond | | Delta-Energy | |
|---|---|---|---|---|---|---|
| Problem | Total | Unique | Total | Unique | Total | Unique |
| Water | 2 | 2 | 2 | 2 | 2 | 2 |
| Methane | 72 | 23 | 36 | 16 | 12 | 8 |
| Photosynthesis | 4 | 4 | 4 | 4 | 5 | 4 |
| Hydrolysis | 222 | 64 | 190 | 48 | 33 | 20 |
| Ethylene | 1402 | 291 | 1383 | 288 | 541 | 243 |
| Esterification | 222 | 64 | 190 | 48 | 33 | 20 |

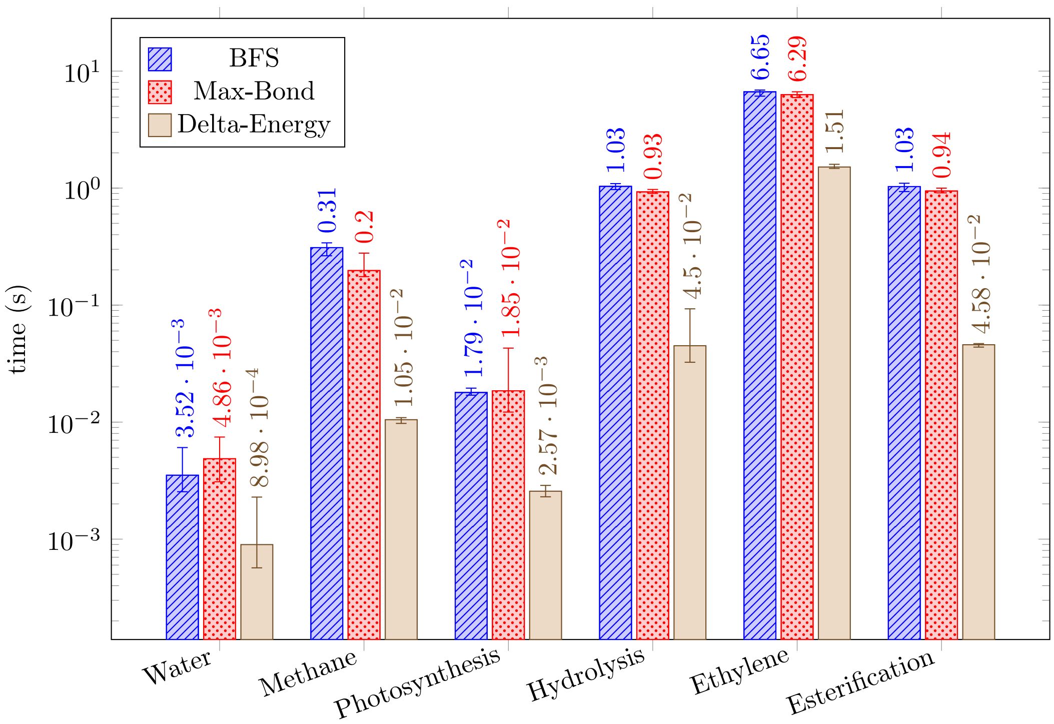

Figure 3: Median runtimes for the Atoms → Molecules stage. The Fermentation problem is omitted, as there are no unknown molecules. Note the logarithmic scale of the $y$-axis. Error bars show 95% confidence intervals.

The reductions in candidate count also seem to lead to shorter runtimes, as shown in Figure 3. Delta-Energy again consistently outperforms the other strategies, achieving speed-ups of 22.9× over BFS for the Hydrolysis problem (0.045 s vs. 1.03 s). Even for the photosynthesis problem, where all iterators explore similar numbers and Delta-Energy actually synthesises an extra total molecule, Delta-Energy still achieves a 6.96× time reduction (2.57 ms vs. 17.9 ms). This suggests that the speed-ups achieved by Delta-Energy are not only due to a reduced number of candidates to explore, but also that the ordering reduces per-candidate overhead.

## 4.4 Q3: Molecules to Reactions

In this stage the heuristics reduce candidates and runtime for most problems, but there are some cases in which BFS performs better, most notably in the Ethylene problem.

This stage of the pipeline synthesises the reactions based on a set of molecules. To prevent any variation in the preceding stage from impacting the performance for this stage, all strategies receive the same input: the smallest number of unique molecules synthesised by any strategy in the Atoms → Molecules stage combined with the already known molecules (or, for Fermentation, just the full known set). Table 3 reports the number of reactions synthesised before all target reactions are found.

Table 3: Reaction candidates explored in the Molecules → Reactions stage.

| Problem | number of initial molecules | BFS | Max-Bond | Delta-Energy |
|---|---|---|---|---|
| Water | 3 | 1 | 1 | 2 |
| Methane | 8 | 33 | 36 | 36 |
| Photosynthesis | 5 | 15 | 15 | 13 |
| Hydrolysis | 22 | 86 | 81 | 81 |
| Ethylene | 244 | 2603 | 4919 | 4919 |
| Esterification | 23 | 150 | 112 | 112 |
| Fermentation | 8 | 69 | 60 | 70 |

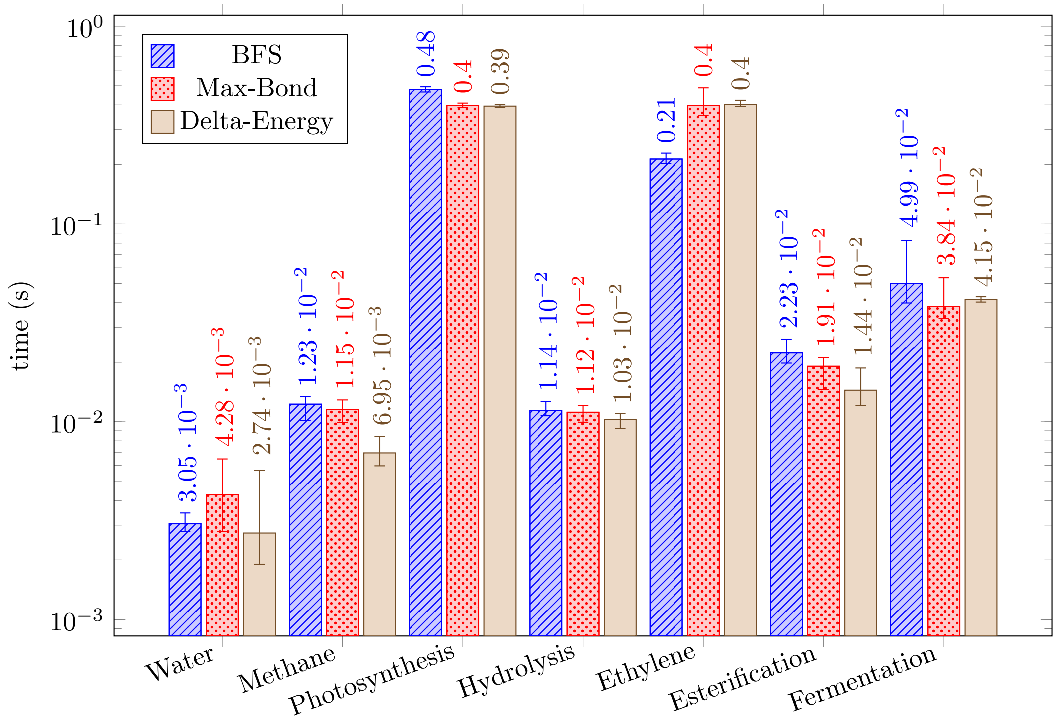

Figure 4: Median runtimes for the Molecules → Reactions stage. Note the logarithmic $y$-axis. Error bars show 95% confidence intervals.

The heuristics seem to match or reduce the number of candidates for the Photosynthesis, Hydrolysis and Esterification problems compared to BFS. However, for Methane and Ethylene both heuristics synthesise more reactions before the target is found. Although the difference is quite small for Methane (33 for BFS, and 36 for the heuristic guided search), in the Ethylene problem both heuristics synthesise nearly twice as many candidates (4919 vs. 2603). As both heuristics have equal performance, this indicates the issue is likely caused by a shared trait in the heuristics. As all molecules in this problem only contain single bonds the heuristics are unable to distinguish the target reaction from other reactions containing only single bonds, losing the ability for the heuristics to prioritise the target over these other candidates. As the unguided BFS search is not affected by how likely reactions are it can find the target sooner. This shows unguided search can outperform energy-guided search, whenever the heuristics are unable to properly differentiate candidates.

Looking at the runtimes in Figure 4, the photosynthesis problem takes the longest, despite having only 5 molecules to choose from, and synthesising only 13-15 unique candidates. This suggests that the total number of molecules in a target reaction has a large impact on the runtime for this stage, not just the number of unique molecules. Another interesting observation is that Delta-Energy still outperforms BFS in terms of runtime on all problems except for Ethylene, even when candidate counts are the same or worse, indicating a reduced per-candidate overhead.

## 4.5 Q4: Reactions to Networks

For the final stage, performance is nearly equal across all three strategies, with every problem having the same number of candidates explored by each strategy.

The final stage of the pipeline assembles networks by taking a subset of reactions from a provided set of reactions. Trying to minimise effects from previous stages, all strategies receive the same fixed set of reactions (the smallest set produced by any strategy in the preceding stage). The total number of networks synthesised is shown in Table 4.

Figure 5 summarises the runtime performance of the different benchmarks. These results seem to indicate a nearly equal performance across all of the benchmarks. The three strategies always synthesise the exact same number of candidates. There are still slight variations in runtime between them (e.g. for hydrolysis, BFS takes 7.82 ms, Max-Bond 9.07 ms, and Delta-Energy 7.34 ms), yet these variations are not as extreme as in some of the other stages. This similarity in performance is likely explained by the simplicity of the problems, with five out of the seven target networks consisting of only a single reaction. Additionally, there is a constraint limiting the candidate networks to only those with all known molecules present. This constraint reduces the number of candidates to discover, meaning the ordering has less of an effect.

Table 4: Unique network candidates explored in the Reactions → Networks stage.

| Problem | number of initial reactions | BFS | Max-Bond | Delta-Energy |
|---|---|---|---|---|
| Water | 3 | 1 | 1 | 1 |
| Methane | 33 | 2 | 2 | 2 |
| Photosynthesis | 13 | 3 | 3 | 3 |
| Hydrolysis | 81 | 4 | 4 | 4 |
| Ethylene | 2603 | 3 | 3 | 3 |
| Esterification | 112 | 235 | 235 | 235 |
| Fermentation | 60 | 1592 | 1592 | 1592 |

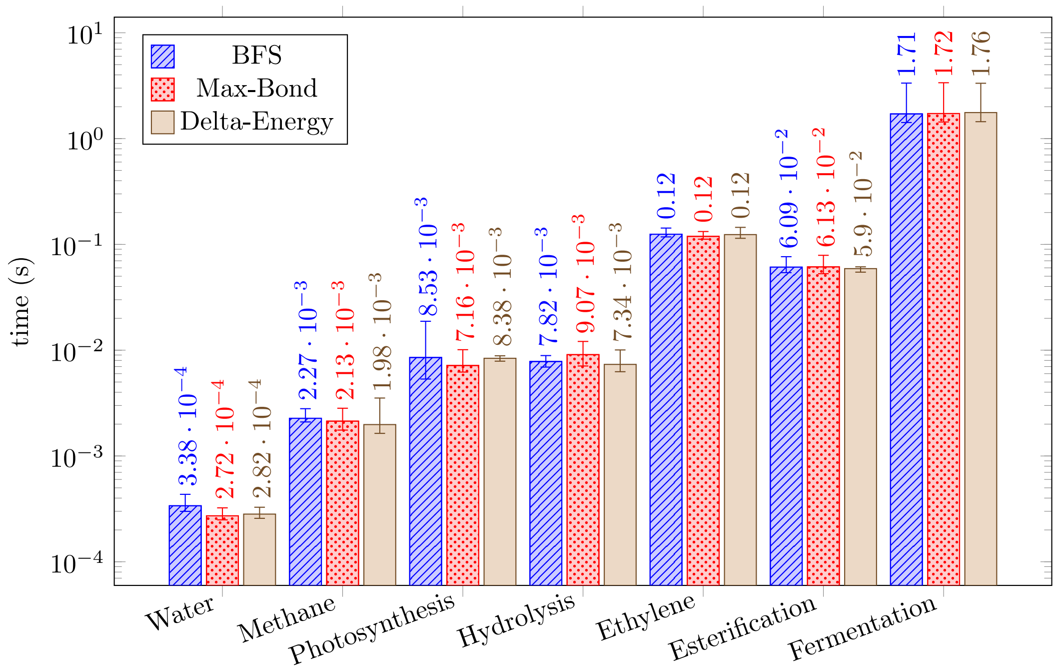

Figure 5: Median runtimes for the Reactions → Networks stage. Note the logarithmic $y$-axis. Error bars show 95% confidence intervals.

## 5 Discussion

The results indicate that for most problems both heuristics can reduce the number of candidates explored and total runtime in CRN synthesis. This is achieved while the heuristics maintain correctness, as they only reorder and do not prune the search space: given sufficient time all three search strategies will find the same solutions.

Delta-Energy outperforms Max-Bond in most cases, which makes sense given that Delta-Energy is able to consider all bonds instead of only the maximum bond-order, allowing for a better-informed and more specific prioritisation. However, Max-Bond outperforms Delta-Energy on the full network fermentation problem, being nearly twice as fast. This is possibly because some competing candidates have a more favourable estimated energy than the target, without having a lower maximum bond-order. As a result, Delta-Energy prioritises these candidates ahead of the target, while Max-Bond does not.

Then there are also cases in which the baseline BFS outperforms both heuristics, most notably for ethylene reaction synthesis, where both heuristics explore nearly twice as many candidates. Since both heuristics show the same increase, this suggests that the heuristics are unable to properly prioritise the reactions in this case. An explanation for this is that, since all molecules in this problem only contain single bonds, neither heuristic can distinguish between the target reaction and any other reaction involving only single bonds. This is caused by simplifying BDE estimation to be based only on bond orders. The ability to provide a meaningful relative ordering is then lost as all reactions that have equal bond orders are also scored equal. This would suggest that more accurate bond-energy estimation could address this issue.

The staged pipeline structure means that improvements in earlier stages have a bigger impact, as earlier improvements compound. Synthesising fewer molecules before finding all unknown molecules, results in a smaller search space of reactions. A smaller search space leads to fewer synthesised reactions and a smaller search space of networks. This compounding effect explains why the largest improvements are in the full pipeline results, with smaller differences in the isolated stages.

Finally, most of the results in reaction and network synthesis stages were similar across all strategies. This could indicate that the strategies perform similarly for these stages. However, it is also plausible that by fixing the input to the smallest set from the preceding stage, the resulting search space in the evaluated stage is not large enough to notice any meaningful differences between the three strategies.

## 6 Responsible Research

**Scientific soundness** All experiments have been run on the same machine (AMD Ryzen 7 5800H), with fixed depth limits for each stage. In order to minimise the effects of outliers the median runtime of five runs has been reported, together with 95% confidence intervals based on a t-distribution and log-transformation. In addition, the code has been made available to best allow a reader to reproduce the setup and results.[4]

**Objectivity** In this research we look to study the effects of implementing the concepts of bond-breaking energy in the program synthesis of CRNs. However, determining exact values for Bond dissociation enthalpy (BDE) is complex due to the values being influenced by numerous factors [4]. So in order to still be able to achieve results and use energy as a search heuristic, we simplify the model.

This means abstracting away from reality. So it is important to reflect on what decisions were made when deciding what concepts to abstract. Firstly, we mainly determine the energy of a reaction by looking at the type of bond. There are other factors such as the atoms involved that have a high impact on the BDE value, this does allow us to hugely simplify while keeping relative ordering (lower energy reactions will generally get lower energy estimates).

This simplification can also lead to mistakes. For example, the methane combustion reaction is exothermic, yet the heuristic estimates it as a net-zero change. This happens as the Delta-Energy heuristic uses averaged values per bond order. Therefore it treats the required energy to break the two oxygen bonds as equal to the energy released by forming carbon dioxide. Similarly, it treats the energy required to break methane's bonds as equal to the energy released when forming water, causing the heuristic to assign the reaction a net-zero value. Despite this, Delta-Energy was still the fastest method for the methane problem. Looking only at the reaction synthesis, BFS returns fewer candidates than Delta-Energy, yet BFS is still slower. This is because Delta-Energy seems to have a reduced per-candidate overhead, so the additional candidates are still synthesised in a faster time. Looking only at bond orders also seems to cause extra candidates synthesised for the ethylene problem, where both heuristics explore nearly twice as many candidates as BFS. Since all molecules in this problem contain only single bonds, neither heuristic can distinguish between the target reaction and any other reaction involving solely single bonds, as reactions with equal bond orders are scored equally.

---

[4] The used code is available at: `https://doi.org/10.5281/zenodo.20759393`

The simplifications and assumptions made in the heuristic are not the only limitations in this research. First of all, the seven problems that have been used as benchmarks are real chemical networks, yet they are still relatively small and simplified. These problems are used to help present the potential of using energy-guided search. However, they fail to represent the full complexity of larger real-world CRNs. Finally, the results of this research are based on a single specific system, this means that any variation (e.g. different grammar or depth limits) might lead to different behaviour of the heuristics.

**Ethical Responsibility** Knowledge is power, and with power comes responsibility. This means that while this research can help in the better and quicker discovery of CRNs in for example medicine, it could also be used to do harm. The tool developed here is general-purpose in nature: the same mechanism that could identify synthesis routes for pharmaceuticals could be directed towards hazardous or toxic substances. Nonetheless, there is a limited amount of risk as the current system only produces a potential reaction network and not any physical laboratory instructions. This means significant domain expertise is still required to act on its output. However, as these sorts of tools grow the risks might increase and researchers should therefore remain careful and continue to consider the potential applications and effects of these systems.

**LLM usage** Claude (Anthropic) was used to assist writing this thesis in two main ways. First to assist with grammar and spelling corrections, suggesting restructuring sections for clarity, and tightening argumentation in sections initially drafted by the author. The generated changes were then critically reviewed before getting incorporated. Second, and most extensively, to provide structured critical feedback on drafts of sections or the full work, identifying issues on content, structure, and language, which were then reviewed and if needed addressed by manually rewriting the relevant sections.

# 7 Conclusions and Future Work

This research has found that bond-energy based heuristics can improve CRN synthesis search. By guiding the search process, these heuristics can reduce the number of candidates synthesised and speed up discovery of the target network. Both the Max-Bond and Delta-Energy heuristics showed reduced runtime and candidate count across most of the tested problems and stages compared to BFS. Delta-Energy achieves the largest improvements (up to 744× on the full pipeline esterification problem) and Max-Bond achieves more moderate but still significant improvements. Results seem to indicate that the biggest improvements are made in the molecule synthesis step. The improvements in the early stages also compound, as having synthesised fewer candidates leads to a smaller search space for the next stage. However, neither heuristic is guaranteed to be an improvement over BFS. This can be best seen in the molecules to reactions stage for the ethylene problem, where both heuristics synthesised nearly double the number of reactions, resulting in also having longer runtimes. This is expected to be caused by the heuristics not being able to properly differentiate between reactions with the same bond-orders, but different actual BDE values.

While these results indicate that energy-guided search can lead to performance improvements in synthesisers for CRN discovery, there are still many areas for further research. Firstly, as the current Delta-Energy heuristic already demonstrates a performance improvement, but mainly has problems when not being able to differentiate reactions, it would be interesting to investigate how more sophisticated heuristics would perform. For example, having different energy estimates based on the atoms present in the bond pair (e.g. differentiate between C-H and O-H) can help to better estimate the actual BDE values and might therefore lead to a more accurate heuristic and greater improvements. Secondly, since the results show that no heuristic fully dominates at all stages, potential improvements might be found by studying the effects of combining different heuristics for each stage, either through stage-specific designed heuristics or through some adaptive heuristic selection. Furthermore, the difference in the number of total and unique candidates in the molecule synthesis stage indicates another area for improvement. Looking into the effects of changing the grammar and constraints to allow for fewer symmetries might also lead to better performance. Finally, testing the heuristics on larger CRNs could help to better understand the scalability of the observed behaviours.

# References

[1] Peter Atkins and Julio de Paula. *Physical Chemistry.* W.H. Freeman & Company, 8 edition, March 2006.

[2] Luca Cardelli, Milan Češka, Martin Fränzle, Marta Kwiatkowska, Luca Laurenti, Nicola Paoletti, and Max Whitby. Syntax-guided optimal synthesis for chemical reaction networks. In Rupak Majumdar and Viktor Kunčak, editors, *Computer Aided Verification*, pages 375–395, Cham, 2017. Springer International Publishing.

[3] Sumit Gulwani, Oleksandr Polozov, and Rishabh Singh. Program synthesis. *Foundations and Trends in Programming Languages*, 4(1-2):1–119, 07 2017.

[4] J. P. Leal. Back to basics: The case of ethane bond enthalpy terms. *Chemical Thermodynamics and Thermal Analysis*, 13:100125, March 2024.

[5] Jan P. Unsleber and Markus Reiher. The Exploration of Chemical Reaction Networks. *Annual Review of Physical Chemistry*, 71:121–142, April 2020.

[6] Richard Wijers. Automated discovery of chemical reaction networks using program synthesis. Master's thesis, TU Delft, 2025.