Enabling Automatic Generation of Accurate Kinetic Models for Complicated Chemical Systems

by

Kehang Han

M. S. Chemical Engineering Practice
Massachusetts Institute of Technology, 2014

B. S. Chemical Engineering and Industrial Bioengineering
Tsinghua University, 2012

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN CHEMICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Signature redacted

Author ....

Department of Chemical Engineering
May 23, 2018

Signature redacted

Certified by...............................

William H. Green
Hoyt C. Hottel Professor of Chemical Engineering
Thesis Supervisor

Signature redacted

Accepted by..................................

Patrick S. Doyle
Robert T. Haslam Professor of Chemical Engineering
Chairman, Committee for Graduate Students

Enabling Automatic Generation of Accurate Kinetic Models for Complicated Chemical Systems

by

Kehang Han

Submitted to the Department of Chemical Engineering
on May 23, 2018 in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Chemical Engineering

Abstract

The past decades have seen much progress in predictive kinetic modeling. Reaction mechanisms have shown increased predictive capability, providing key insights into chemical transformations under conditions of interest. Coupled and integrated in multiscale-multiphysics models, reaction mechanisms help elucidate physical phenomena that are driven by chemical kinetics and are recognized as a necessary tool for chemical selection, reactor design and process optimization. These past kinetic modeling achievements have opened new opportunities for novel scientific applications in chemical kinetics community and encouraged kinetic modelers to study even more complex chemical systems.

As one can expect, the system complexity significantly increases modeling cost in both reaction mechanism construction and simulation. Over the years we have seen formulation of various lumping strategies. Despite simplicity, the lumping strategy introduces an intrinsic error where the lumps contain molecules with very different reactivities. Frequently, oversimplified models using the kinetic parameters fitted from a very limited set of pilot experiments, resulting in poor accuracy in extrapolation.

This thesis focuses on automated detailed kinetic modeling strategy using Reaction Mechanism Generator (RMG). RMG-generated models more faithfully represent the chemistry so they have superior extrapolation potential. But as system complexity increases, several computational limitations prevent RMG from converging. This thesis has made several contributions: reducing memory usage, boosting algorithm scalability, improving thermochemistry estimation accuracy, which eventually expand RMG's modeling capability toward large complex systems. These contributions are available to the kinetics community through the RMG software package. To demonstrate the improved modeling capability of RMG, the thesis also includes a large chemical application: heavy oil thermal decomposition under geological conditions via a C18 model compound, phenyldodecane.

As an extension of RMG, the thesis also explores a promising alternative to detailed kinetic modeling when dealing with extremely large chemical systems: fragment-based kinetic modeling, which generates a reaction network in fragment space rather than molecule space. The thesis shows via a case study that the new method creates a much smaller reaction network but with similar prediction accuracy on feedstock conversion and products' molecular weight distribution compared to its counterpart model generated by RMG.

Thesis Supervisor: William H. Green
Title: Hoyt C. Hottel Professor of Chemical Engineering

Dedication

To my mom and dad,
whose love makes me march forward fearlessly.

Acknowledgments

My thesis work can never reach this far without the help and support from the lovely people here. I feel extremely lucky to have them in my journey of pursuing PhD at MIT.

First, thank you to Professor William Green. It's been a real pleasure to be your student. You ability of listening to everyone, understanding individual situations, and creating an atmosphere where all of us enjoy work and interaction has greatly inspired me, and I wish I can also have that "magic power" in my future career. Along the journey, your countless feedback and encouragements have turned out extremely helpful to shape my perspective towards research and life.

Thank you to my thesis committee, Professor Paul Barton and Professor Richard Braatz. Your expertise on numerical analysis, parallelization, and software in general has always been invaluable resource.

I also thank my Practice School Station Director Dr. Robert Hanlon. He is a great educator who has very high standards but also is ready to provide practical advices for his students to improve. I'm grateful for his guidance at all levels at Texas station and his suggestions for my future growth.

Next, I'd like to thank Dr. Nick Vandewiele for his mentorship. It is a pleasant experience to work closely with Nick on some of the most challenging projects during my PhD. The numberless collaborative coding sessions exposed me with many good practices from designing a software feature, laying out roadmap to eventually implementing it with necessary tests. I'll never forget the time we together brought RMG-server back to life in E40 basement, the trip to NYC for Intel Parallelization Workshop and the discussions we had about big dreams for RMG future and our careers.

I am also very grateful to another mentor Dr. Connie Gao, the previous lead developer of RMG. She is a brilliant and extremely productive and helpful developer who maintained earlier versions of RMG-Py, designed and created RMG-website and group website. She inspired me to take more responsibility in the group, which eventually led me to serve as the next lead developer of RMG and the administrator of RMG-server.

I thank RMG developer team: Professor Richard West, Nathan Yee, Max Liu, Mark Goldman, Alon Dana, Mark Payne, Matt Johnson, Ryan Gillis, Zachary Buras, Colin Grambow and Hiroumi Tani for their dedication to improving the software, ressolving user-reported issues, updating documentation and developer wiki and providing user trainings.

I would also like to thank my collaborators in ENI projects: Jianghuai Cai, Peng Zhang, Mark Payne, Dr. Sarah Khanniche and Lily Wang and in thermochemistry projects: Dr. Adeel Jamal, Colin Grambow, Dr. Zachary Buras, Connor Coley and Dr. Yi-Pei Li. I especially thank Dr. Adeel Jamal for tirelessly calculating an important set of polycyclics to validating my heuristic method, Lawrence Lai and Dr. Soumya Gudiyella for guiding a pure computationalist through high-pressure experiments, and Connor Coley for sharing his initial work on physical property prediction using convolutional neural network to get my new thermochemistry estimator started.

I also thank my squash partners in the group: Mark Goldman and Peng Zhang, with whom I had so much fun as well as exercise needed for a software developer. In particular, I enjoyed the "exercise-time" discussion with them, especially on societal effects of our research that have been overlooked by many.

A big thank-you to all the Green group members for making the group such a wonderful place to work. A special thank-you to Gwen Wilcox for being always supportive and resourceful. Your tireless efforts make sure the whole Green group is up and running smoothly at any moment.

Thank you to all my friends for accompanying me in these years, especially Huijun, Junjie, Botao, Tailin, Zhilong, Liza, Jae Jung, Xiaoxue. Whether it's cycling around Boston, watching movies, jogging along the Charles River, or doing fun projects, great time is always guaranteed.

Finally, I thank my family: mom, dad, sister and BB. Your love is always there, even at this moment of writing I can totally feel it.

Contents

List of Figures

List of Tables

1

INTRODUCTION

Since the birth of chemical engineering, modeling has been a central tool to guide us from what we know to tackle the unkown. In early times when few tools were available, researchers set up simple models to explain known chemical observations and anticipate new ones. Failed predictions in return served as valuable "raw materials" for refining understanding. Gradually, chemical engineering has gone from an experience-summarizing field, toward a predictive science. Nowadays the advancement in computer power has enabled us to leverage the capability of complex models to predict molecular properties, optimize chemical units, automate manufacturing etc.

Throughout the history of chemical engineering, the prediction and modeling of reactive systems has always been one of the most challenging and rewarding fields. With the successes in method development of molecular dynamics, quantum mechanics, numerical simulations, the past decades have seen many advances in predictive kinetic modeling. Reaction mechanisms have shown increased predictive capability, providing key insights on chemical transformations under conditions of interest. Coupled and integrated in multiscale-multiphysics models, reaction mechanisms help elucidate physical phenomena that are driven by chemical kinetics and are recognized as a necessary tool for design and optimization of chemical systems/processes [1–3], selection of future fuels [4–6], and to assist in environmental policymaking [7].

These past kinetic modeling achievements have opened new opportunities for novel scientific applications in the chemical kinetics community and encouraged kinetic modelers to study even more complex chemical systems. For instance, heavy feedstocks originally represented by a single-component surrogate of low-carbon number hydrocarbon (e.g., the use of heptane to represent diesel fuel) can choose a more realistic multicomponent formulation with larger hydrocarbons.

On the other hand, the changing global energy landscape also fuels the need to provide solutions for chemical systems of increasing complexity. For instance, over the recent years the produced crude oils have become heavier with rich heteroatomic content (e.g., sulfur-containing species), while the transportation sector maintains high demand for light and clean products such as gasoline, jet fuel and diesel. The conflicting trends draw great research attention to study heavy petroleum fraction conversion and complex heteroatomic behaviors.

As one can expect, the system complexity significantly increases modeling cost in both reaction mechanism construction and simulation. Over the years we have seen formulation of various lumping strategies. Generally there are three choices for modeling detail level balanced by system complexity (e.g., carbons in feedstock), as shown in Figure 1.1.

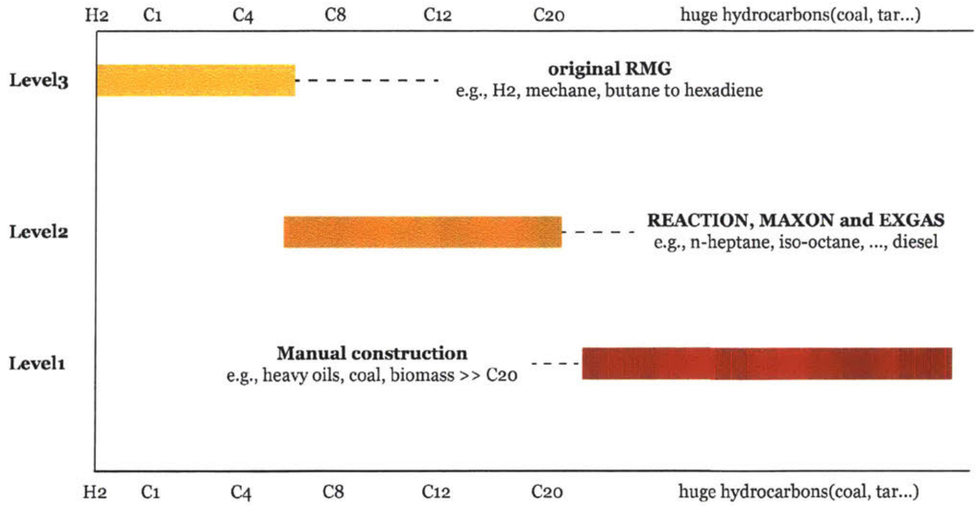

H2
C1
C4
C8
C12
C20
huge hydrocarbons(coal, tar...)
Level3
original RMG
e.g., H2, mechane, butane to hexadiene
Level2
REACTION, MAXON and EXGAS
e.g., n-heptane, iso-octane, ..., diesel
Level1
Manual construction
e.g., heavy oils, coal, biomass >> C20
H2
C1
C4
C8
C12
C20
huge hydrocarbons(coal, tar...)

Figure 1.1: Illustration of trade off between modeling detail and chemical system complexity.

Level 1 is usually chosen for the most complex systems (e.g., coal, biomass, vacuum residue, etc.) where people have little knowledge on either composition or structure of the feedstock. See an example of 3-lump model for the catalytic cracking process in Figure 1.2 [8].

Level 2 is a more detailed strategy where species representation in mechanisms is at molecule level but many isomers are lumped in one species. This effectively reduces model complexity since the number of isomers increases exponentially with carbon number in a hydrocarbon: C_5H_{12} has 3 isomers, $C_{10}H_{22}$ has 75, and $C_{20}H_{42}$ has 366,319. For instance, Ranzi *et al.* proposed a scheme where radical isomers such as alkyl, peroxy, hydroperoxyalkyl are lumped into properly selected species. A resulting mechanism generated for n-heptane combustion process has no more than 145 species [9], which is later used in modelling C7-C20 chemistry. Literature has shown that at level 2, modelers were able to construct mechanisms for up to C20 systems (e.g., large component in diesel) [10]. However, the accuracy of this approximation depends strongly on application characteristics, lumping schemes, and model parameters estimation (e.g., thermochemistry properties for lumped species and kinetics for lumped reactions). For instance, with systems where people care about the certain isomer in product distribution that are sensitive to isomerization or intramolecular reactions, e.g. low temperature ignition and cyclization, the prediction accuracy is largely affected.

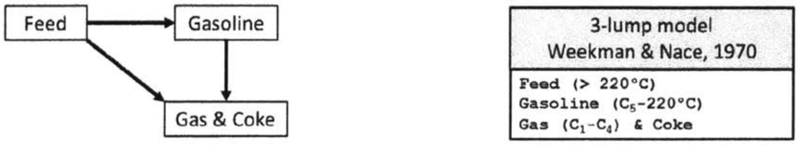


Figure 1.2: A simple lumped model for catalytic cracking process by Weekman and Nace. From Oliveira *et al.*

Level 3 is the most detailed modeling approach where the mechanisms distinguish all the isomers. In the past, chemical processes were often modelled at level 1 and level 2 due to lack of computing power and good understanding of underlying chemistry. Today, with the advances in computational chemistry (e.g., *ab initio* calculations) and emergence of fast numeric solvers, kinetic models can now be constructed at this level and applied to relatively simple systems. The detailed structural information of every isomer allows direct connection of the reaction mechanisms to individual "model compound" studies, *ab initio* calculations, and LFER (Linear Free Energy Relationship, e.g. Benson Group Additivity Method) estimates. On the other hand, level 3 incorporates relevant elementary pathways into the mechanisms so that the resulting models reflect most fundamental chemistry and have highest predictive potential.

1.1 Automatic reaction mechanism generation

In systems with relatively unselective chemistry, such as pyrolysis, combustion, partial oxidation and many polymerizations, tens to even hundreds of thousands of species are present. The increased detail requires keeping track of all of these species and reactions with explicit representation of molecules. That makes manual model construction tedious, error-prone and often biased. Thus, over past decades various automatic kinetic model generation packages have been developed [11–15]. They errorlessly distinguish unique species and reactions, consistently construct kinetic models using data that is continually updated, e.g., reaction templates and estimation parameters, thus are increasingly adopted in many applications. Among them, the Reaction Mechanism Generator (abbreviation: RMG) is an actively maintained open source generator developed by Green Group at MIT and West Group at Northeastern University. It is designed to automate kinetic mechanism generation at level 3, and has been used throughout this thesis. More details of the software can be found elsewhere [15, 16].

1.2 Thesis overview

Before this thesis, RMG was suceesfully applied to a collection of relatively small chemical systems (usually $<$ C8): dimethyl ether, propane, butanol, neopentane, hexane, hexadiene etc [4, 17–20]. As system complexity increases, several computational limitations prevent RMG from converging to mechanism completion. As will be uncovered in later chapters, many of them are deeply related to RMG algorithm design.

This thesis has made several contributions to identifying root issues, providing corresponding solutions, and eventually expanding RMG's modeling capability toward large complex systems. To demonstrate the improved modeling capability, the thesis also includes a large chemical application: heavy oil thermal decomposition under geological conditions via a C18 model compound, phenyldodecane.

Chapter 2 discusses one aspect of convergence difficulty: high memory usage. It's directly associated with side byproduct species and reactions RMG has to store, whose number grows exponentially with chemical mechanism size. A memory-efficient algorithm is presented for coping with the combinatorial complexity. The algorithm carefully identifies unimportant species during model generation and prunes them as well as their reactions. The new algorithm reduces memory usage by about a factor of 4 for a wide range of applications without sacrificing accuracy; with fixed computer memory it enables convergence of reaction mechanisms about twice as large as previously possible. The increased capability opens the possibility of discovering unexplored reaction networks and modeling more complicated reacting systems.

Chapter 3 focuses on a second aspect of convergence difficulty: low execution efficiency. There are causes from three sources: programming language, algorithm scalability, and application size. Algorithm scalability was improved in this thesis. We examined a wide range of applications and identified bottlenecks in efficiency performance. For instance, large systems usually suffer most from combinatorial reaction generation, while applications with QMTP (Quantum Mechanics Thermodynamic Property) enabled were bottlenecked by the slow quantum calculations. The chapter presents two methods to speedup reaction generation: reaction filtering and parallelization, as well as one method to speedup chemical data computation: concurrent computation of thermochemistry. These methods jointly bring significant speedup (e.g., 50 times in early conversion of phenyldodecane).

There is a hidden cause to convergence difficulty: inaccurate parameter estimation. For instance, the thermochemistry error of a species can mislead RMG to explore unnecessary pathways, indirectly wasting computer power and memory. In general, cyclic and polycyclic species suffer poor thermochemistry estimation using existing methods. Chapter 4 provides a fast heuristic method that extends the group additivity method with two additional algorithms: similarity match and bicyclic decomposition. It significantly reduces $H_f(298\text{ K})$ estimation error from over 60 kcal/mol (RMG's original group additivity method) to around 5 kcal/mol, $C_p(298\text{ K})$ error from 9 cal/mol/K to 1 cal/mol/K, and $S(298\text{ K})$ error from 70 cal/mol/K to 7 cal/mol/K. This method also works well for heteroatomic polycyclics.

As RMG models increasingly complex chemical systems, the heuristic estimator encounters molecules which breaks its hidden assumptions: large fused polycyclics. New insights are needed to adapt the thermochemistry estimator to new molecule domains. Instead of proposing new heuristics, Chapter 5 discusses the possibility of creating a self-adapting estimator. It presents a new machine learning appraoch using molecular convolutional neural networks (MCNN) which helps gain higher accuracy than heuristic method without asking for human insights. We also designed the uncertainty estimation scheme for the MCNN estimator, which eventually leads to the construction of a pipeline that makes MCNN estimator self-evolve over time.

Chapter 6 presents a large chemical application that RMG originally wasn't able to model smoothly: thermal decomposition of oil at geological conditions using a C18 heavy oil analog, phenyldodecane (PDD). New version of RMG was used to automatically construct a full decomposition mechanism of PDD pyrolysis. The RMG-generated model successfully achieved good agreements with various of experimental datasets on both reactant conversion and major product distributions.

Chapter 7 explores an alternative modeling approach to current RMG algorithm. It is mainly tailored for chemical systems larger than the ones RMG is currently able to model. Instead of focusing on modeling at molecule level, it describes chemical reactions at fragment level (a fragment is a part of a molecule containing one or more functional groups). Via a case study with the same application presented in Chapter 6, we show that the new method creates a much smaller reaction model but with similar prediction accuracy on feedstock conversion and products' molecular weight distribution compared to its counterpart model generated by RMG.

Finally, Chapter 8 discusses several recommendations for future work in predictive chemical kinetics.

1.3 References

[1] C. K. Westbrook, W. J. Pitz, O. Herbinet, H. J. Curran, and E. J. Silke. "A comprehensive detailed chemical kinetic reaction mechanism for combustion of n-alkane hydrocarbons from n-octane to n-hexadecane." *Combustion and Flame* 156 (1), Jan. 2009, pp. 181–199. ISSN: 0010-2180. DOI: 10.1016/j.combustflame.2008.07.014. URL: http://www.sciencedirect.com/science/article/pii/S0010218008002125.

[2] F. Battin-Leclerc, J. M. Simmie, and E. Blurock, eds. *Cleaner Combustion: Developing Detailed Chemical Kinetic Models.* en. Green Energy and Technology. London: Springer-Verlag, 2013. ISBN: 978-1-4471-5306-1. URL: //www.springer.com/us/book/9781447153061.

[3] Sabbe Maarten K., Van Geem Kevin M., Reyniers Marie-Françoise, and Marin Guy B. "First principle-based simulation of ethane steam cracking." *AIChE Journal* 57 (2), Jan. 2011, pp. 482–496. ISSN: 0001-1541. DOI: 10.1002/aic.12269. URL: https://onlinelibrary-wiley-com.libproxy.mit.edu/doi/full/10.1002/aic.12269.

[4] M. R. Harper, K. M. Van Geem, S. P. Pyl, G. B. Marin, and W. H. Green. "Comprehensive reaction mechanism for n-butanol pyrolysis and combustion." *Combustion and Flame* 158 (1), Jan. 2011, pp. 16–41. ISSN: 0010-2180. DOI: 10.1016/j.combustflame.2010.06.002. URL: http://www.sciencedirect.com/science/article/pii/S0010218010001586.

[5] N. M. Vandewiele, G. R. Magoon, K. M. Van Geem, M.-F. Reyniers, W. H. Green, and G. B. Marin. "Kinetic Modeling of Jet Propellant-10 Pyrolysis." *Energy & Fuels* 29 (1), Jan. 2015, pp. 413–427. ISSN: 0887-0624. DOI: 10.1021/ef502274r. URL: https://doi.org/10.1021/ef502274r.

[6] C. W. Gao, A. G. Vandeputte, N. W. Yee, W. H. Green, R. E. Bonomi, G. R. Magoon, H.-W. Wong, O. O. Oluwole, D. K. Lewis, N. M. Vandewiele, and K. M. Van Geem. "JP-10 combustion studied with shock tube experiments and modeled with automatic reaction mechanism generation." *Combustion and Flame* 162 (8), Aug. 2015, pp. 3115–

3129. ISSN: 0010-2180. DOI: 10.1016/j.combustflame.2015.02.010. URL: http://www.sciencedirect.com/science/article/pii/S0010218015000528.

[7] S. H. Schneider. "Integrated assessment modeling of global climate change: Transparent rational tool for policy making or opaque screen hiding value-laden assumptions?" en. *Environmental Modeling & Assessment* 2 (4), Dec. 1997, pp. 229–249. ISSN: 1420-2026, 1573-2967. DOI: 10.1023/A:1019090117643. URL: https://link.springer.com/article/10.1023/A:1019090117643.

[8] L. P. d. Oliveira, D. Hudebine, D. Guillaume, and J. J. Verstraete. "A Review of Kinetic Modeling Methodologies for Complex Processes." en. *Oil & Gas Science and Technology – Revue d'IFP Energies nouvelles* 71 (3), May 2016, p. 45. ISSN: 1294-4475, 1953-8189. DOI: 10.2516/ogst/2016011. URL: https://ogst.ifpenergiesnouvelles.fr/articles/ogst/abs/2016/03/ogst150117/ogst150117.html.

[9] E. Ranzi, M. Dente, A. Goldaniga, G. Bozzano, and T. Faravelli. "Lumping procedures in detailed kinetic modeling of gasification, pyrolysis, partial oxidation and combustion of hydrocarbon mixtures." *Progress in Energy and Combustion Science* 27 (1), Jan. 2001, pp. 99–139. ISSN: 0360-1285. DOI: 10.1016/S0360-1285(00)00013-7. URL: http://www.sciencedirect.com/science/article/pii/S0360128500000137.

[10] F. Battin-Leclerc. "Detailed chemical kinetic models for the low-temperature combustion of hydrocarbons with application to gasoline and diesel fuel surrogates." *Progress in Energy and Combustion Science* 34 (4), Aug. 2008, pp. 440–498. ISSN: 0360-1285. DOI: 10.1016/j.pecs.2007.10.002. URL: http://www.sciencedirect.com/science/article/pii/S0360128507000627.

[11] E. Ranzi, T. Faravelli, P. Gaffuri, and A. Sogaro. "Low-temperature combustion: Automatic generation of primary oxidation reactions and lumping procedures." *Combustion and Flame* 102 (1–2), July 1995, pp. 179–192. ISSN: 0010-2180. DOI: 10.1016/0010-2180(94)00253-O.

[12] F. Battin-Leclerc. "Development of kinetic models for the formation and degradation of unsaturated hydrocarbons at high temperature." en. *Physical Chemistry Chemical Physics* 4 (11), May 2002, pp. 2072–2078. ISSN: 1463-9084. DOI: 10.1039/B110563A.

[13] L. J. Broadbelt, S. M. Stark, and M. T. Klein. "Computer Generated Pyrolysis Modeling: On-the-Fly Generation of Species, Reactions, and Rates." *Industrial & Engineering Chemistry Research* 33 (4), Apr. 1994, pp. 790–799. ISSN: 0888-5885. DOI: 10.1021/ie00028a003.

[14] J. Song. "Building robust chemical reaction mechanisms : next generation of automatic model construction software." eng. Thesis (Ph. D.)–Massachusetts Institute of Technology, Dept. of Chemical Engineering, 2004. Thesis. Massachusetts Institute of Technology, 2004.

[15] C. W. Gao, J. W. Allen, W. H. Green, and R. H. West. "Reaction Mechanism Generator: Automatic construction of chemical kinetic mechanisms." *Computer Physics Communications* 203, June 2016, pp. 212–225. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2016.02.013.

[16] C. W. Gao. "Automatic reaction mechanism generation: High Fidelity Predictive Modeling of Combustion Processes." eng. Thesis. Massachusetts Institute of Technology, 2016. URL: http://dspace.mit.edu/handle/1721.1/104205.

[17] E. E. Dames, A. S. Rosen, B. W. Weber, C. W. Gao, C.-J. Sung, and W. H. Green. "A detailed combined experimental and theoretical study on dimethyl ether/propane blended oxidation." *Combustion and Flame* 168, June 2016, pp. 310–330. ISSN: 0010-2180. DOI: 10.1016/j.combustflame.2016.02.021. URL: http://www.sciencedirect.com/science/article/pii/S0010218016000778.

[18] S. V. Petway, H. Ismail, W. H. Green, E. G. Estupiñán, L. E. Jusinski, and C. A. Taatjes. "Measurements and Automated Mechanism Generation Modeling of OH Production in Photolytically Initiated Oxidation of the Neopentyl Radical." *The Journal of Physical Chemistry A* 111 (19), May 2007, pp. 3891–3900. ISSN: 1089-5639. DOI: 10.1021/jp0668549. URL: https://doi.org/10.1021/jp0668549.

[19] Van Geem Kevin M., Reyniers Marie-Francoise, Marin Guy B., Song Jing, Green William H., and Matheu David M. "Automatic reaction network generation using RMG for steam cracking of n-hexane." *AIChE Journal* 52 (2), Oct. 2005, pp. 718–730. ISSN: 0001-1541. DOI: 10.1002/aic.10655. URL: https://onlinelibrary-wiley-com.libproxy.mit.edu/doi/abs/10.1002/aic.10655.

[20] S. Sharma, M. R. Harper, and W. H. Green. "Modeling of 1,3-hexadiene, 2,4-hexadiene and 1,4-hexadiene-doped methane flames: Flame modeling, benzene and styrene formation." *Combustion and Flame* 157 (7), July 2010, pp. 1331–1345. ISSN: 0010-2180. DOI: 10.1016/j.combustflame.2010.02.012. URL: http://www.sciencedirect.com/science/article/pii/S0010218010000581.

2

Memory Usage Reduction with Pruning algorithm

To achieve high model fidelity, a very large number of possible reactions, intermediates, and byproducts must be considered when constructing the reaction mechanism. When dealing with complicated reactive systems (e.g., higher carbon number fuels, higher equivalence ratios), automated model generators like RMG are often restricted by hardware limitations; numerous species and their reactions quickly fill up the computer memory [1]. Because the number of possible bimolecular reactions approximately scales as the square of the number of species in the model, the memory usage increases superlinearly (see Figure 2.1). For instance, generating a model by RMG with more than 230 species on a computer with 8GB RAM leads to memory allocation shortage. Although this issue will eventually be relieved by future improvement of RAM size, it usually takes time; historically it took the industry almost 5 years to increase standard RAM size by a factor of 4. Thus this chapter aims to provide solutions to reduces RAM requirements for RMG.

To mitigate memory limitation, there have been several previous attempts to develop software that combines model generation and model reduction. Among them is Klinke and Broadbelt's work, which incorporated into their reaction mechanism generation algorithm in NetGen a radical lumping strategy that groups radicals based on their similarity in reactivity, and on-the-fly sensitivity analysis that evaluates the importance of a certain species based on its impact on fluxes towards IN (Important and Necessary) species [1]. This integration allowed NetGen to create more accurate kinetic models.

However, the Klinke and Broadbelt approach cannot be implemented in RMG, due to different model generation strategies: 1) RMG is designed to distinguish all the radicals so that all the thermo-chemistry and kinetic parameters can be calculated from first principles; and 2) certain species in RMG (called edge species, see

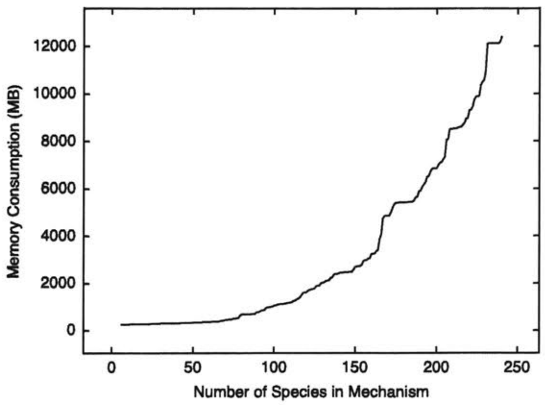


Figure 2.1: Memory (RAM) usage by RMG-Py grows super linearly as the reaction mechanism is enlarged to improve fidelity. This example is partial oxidization of natural gas.

Subsection 2.1.1) temporarily do not react, having no impact on the fluxes towards core species (equavalent IN species in RMG), which makes sensitivity evaluation ineffective. Consequently, this chapter presents an alternative memory-efficient approach for on-the-fly model reduction during mechanism generation. By identifying and pruning unimportant species based on flux analysis in early stages, RMG was able to generate a model of over 400 species without memory shortage.

2.1 Method

2.1.1 The original algorithm (no pruning)

RMG works by a "core-edge model" approach [2]. The model "core" collects all the important species selected by a rate-based algorithm, while the model "edge" collects all the other species appearing as products of reactions of the core species. The "edge" serves as a species pool for future selections of important species. At each iteration one of the edge species is moved into the "core", and new species are added to the "edge". At the end of model generation, the "core" model will be the final model to be exported.

Typical RMG model generation workflow is illustrated in Figure 2.2. User input will be translated to initial core-edge model, which is further transformed into an ODE system. Simulation starts from t=0 , along which edge species' fluxes will be monitored at each time point. If an edge species's flux becomes greater than a predefined threshold, that species will be selected to the core, next the new core species will be reacted with other core species so that model is enlarged. Updated model will trigger a new ODE simulation. This process continues iteratively until the model integrates to the specified final time. All reactions involving only core species are output as the final model.

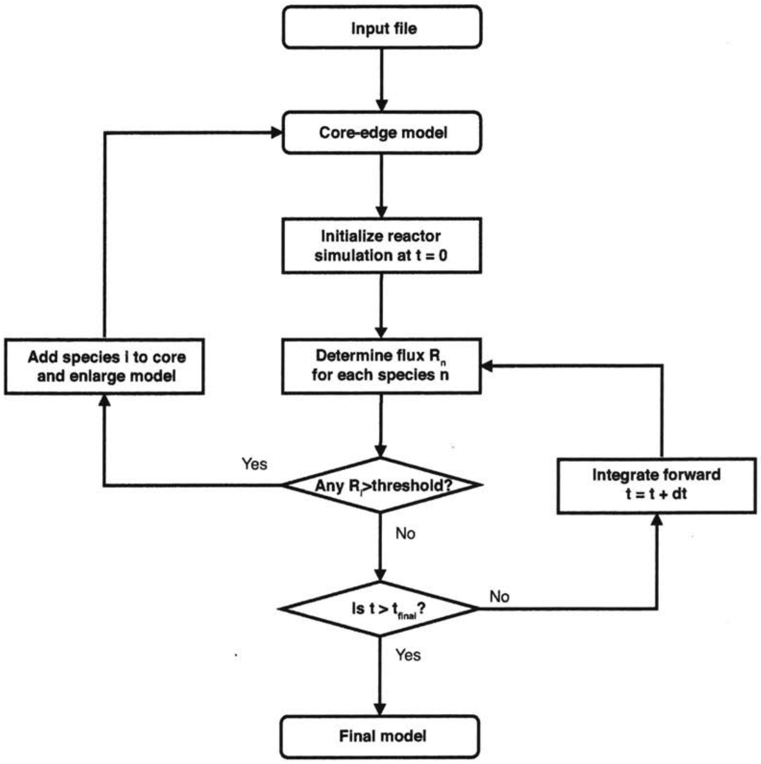

Input file
Core-edge model
Initialize reactor simulation at t = 0
Determine flux R_n for each species n
Add species i to core and enlarge model
Yes
Any R_i>threshold?
No
Integrate forward t = t + dt
Is t > t_{final}?
No
Yes
Final model

Figure 2.2: Model generation workflow of original algorithm

As Figure 2.3 shows in more detail, an RMG iteration starts with a pool of species (species A, B, C in core, D, E, F, and G in edge), and solves the ODE system corresponding to reactions within the set of core species. From the resulting concentrations of core species, reacting fluxes towards each edge species are computed as follows, and then compared to a flux threshold.

$$r_{species_i}(t) = \sum_j \nu^{i,j} r_j(t)$$

where $\nu^{i,j}$ is stoichiometric coefficient of species i in reaction j and r_j is reaction rate for reaction j usually written in Arrhenius form $r_j(t) = k_j T(t)^{n_j} exp(-\frac{E_a^j}{RT(t)}) \prod_m c_m^{\nu^{m,j}}(t)$. For instance, at some time the flux towards edge species D is found to be significant, i.e., $r_D(t) > flux\ threshold$, then the computation is halted, and D is moved to the "core". Specifically, the threshold is calculated as $flux\ threshold =$

R_{char} * `toleranceMoveToCore`, where R_{char} is the root sum square of core species fluxes and `toleranceMoveToCore` is specified by the user acoording to his/her preference of final mechanism accuracy (see more detail on `toleranceMoveToCore` in 2.1.2.1). RMG will later enlarge the core-edge model by exploring reactions between D and other core species. After model "core" and "edge" are updated, the simulator solves the system again from t = 0 to the point when next important species is discovered. The whole iterative process terminates when the user-specified goal time/conversion is reached and no additional important species is identified.

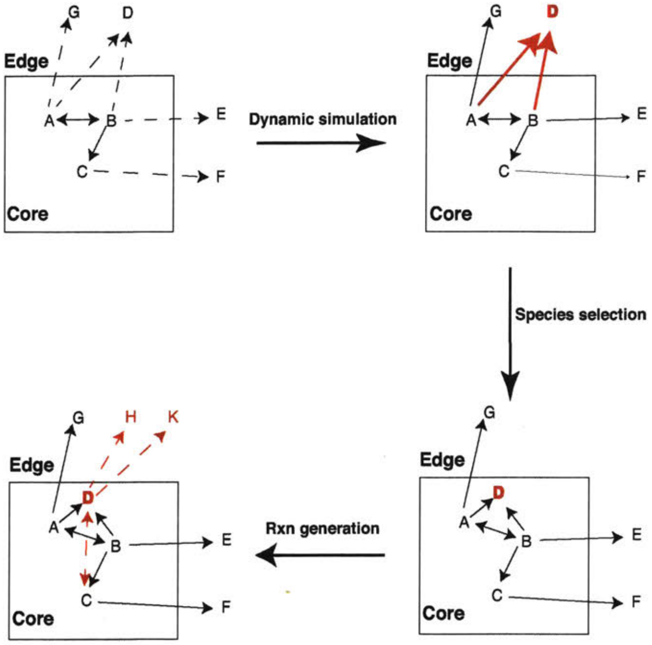


Figure 2.3: one iteration of model generation by RMG (original algorithm)

2.1.2 The new algorithm (with pruning)

Ultimately RMG produces a final kinetic model with only core species and reactions. However, RMG has to store both edge and core throughout simulation. Since the number of edge species is much larger than that of core species (typically by over 2 orders of magnitude), most memory is consumed by the model "edge". Furthermore, among the edge species are many minor species unlikely to become core species due to structural unstability and usually have low $r_{species_i}$. Thus, pruning those minor edge species is helpful to mitigate RAM limitation and still keeps model accuracy. In order to achieve that, pruning module should first identify unimportant edge species and then delete them and their reactions to minimize the impact on model accuracy.

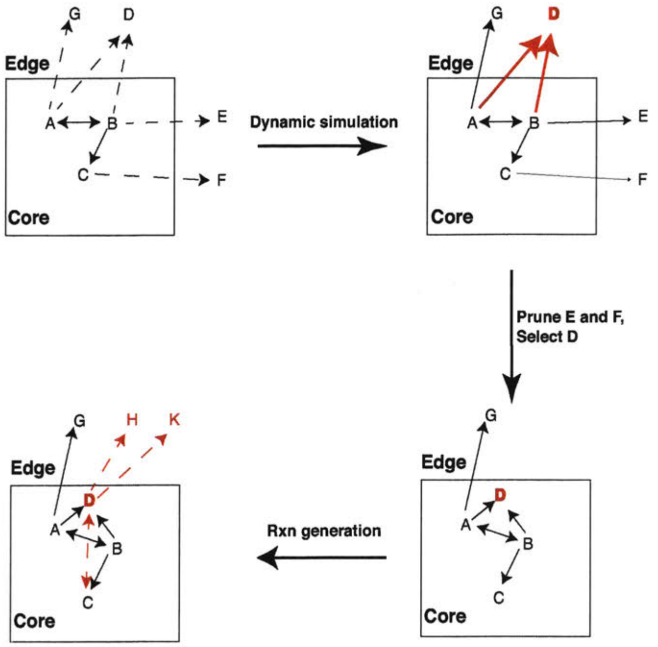


Figure 2.4: one iteration of model generation by RMG with pruning

The pruning module is integrated into the original RMG package illustrated in Figure 2.5; the main difference from the original workflow is, after identifying new core species, simulation will continue to final time to figure out **maximum normalized flux** for each edge species:

$$max_{t\in[0,t_{final}]}\left(\frac{flux_i(t)}{R_{char}(t)}\right)$$

where t_{final} is set by user for simulation termination, see Figure 2.2.

Those species with $max_{t\in[0,t_{final}]}\left(\frac{flux_i(t)}{R_{char}(t)}\right) \leq$ toleranceKeepInEdge will be pruned (e.g. in Figure 2.4, edge species E and F having relatively small flux are pruned). The simulation will only stop if some edge flux exceeds threshold2 (computed as $threshold2 = R_{char} *$ toleranceInterruptSimulation, see more detail in 2.1.2.2), in which case pruning won't be executed.

Four parameters are critical to make pruning functionality work properly.

2.1.2.1 toleranceMoveToCore: for threshold1

This tolerance is inherited from the previous algorithm. Users get final kinetic models with desired accuracy by setting an appropriate toleranceMoveToCore. For instance, toleranceMoveToCore = 0.1 means all the edge species with fluxes ≥ 10%

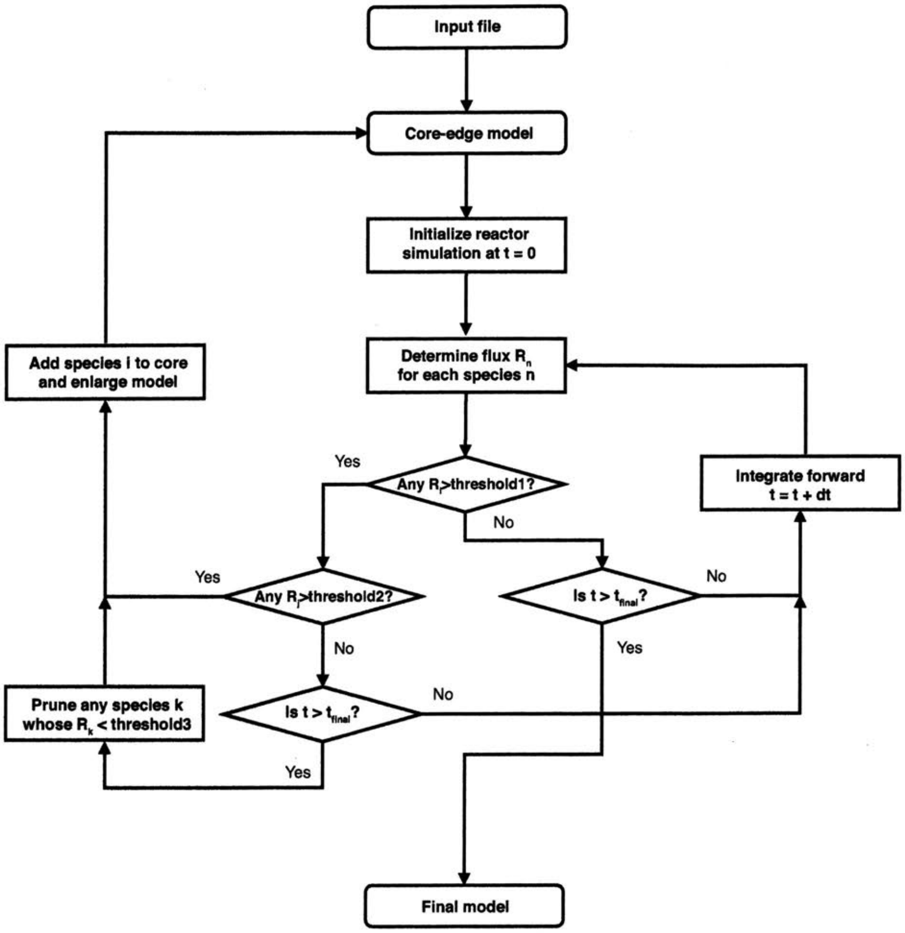

Input file
Core-edge model
Initialize reactor simulation at t = 0
Determine flux R_n for each species n
Add species i to core and enlarge model
Yes
Any R_i>threshold1?
No
Integrate forward t = t + dt
Yes
Any R_i>threshold2?
Is t > t_{final}?
No
No
Yes
Prune any species k whose R_k < threshold3
Is t > t_{final}?
No
Yes
Final model

Figure 2.5: Workflow of pruning algorithm

of R_{char} will be moved into the final model (model "core"). Normally, lower (tighter) `toleranceMoveToCore` leads to a larger, more detailed final model.

2.1.2.2 toleranceInterruptSimulation: for threshold2

RMG has to run a complete dynamic simulation (from time=0 to goal time/conversion) to get $\max_{t\in[0,t_{final}]}\left(\frac{flux_i(t)}{R_{char}(t)}\right)$ for edge species i. However, in cases where any of the edge species has an unrealistically high flux, the species is clearly important and must be included in the core first instead of conducting pruning. Thus, `toleranceInterruptSimulation` is defined to decide if a flux is beyond certain limit; kinetic simulation will be halted when some flux is higher than `toleranceInterruptSimulation`$* R_{char}$. Small values of `toleranceInterruptSimulation` have the effect of turning off pruning.

2.1.2.3 `toleranceKeepInEdge`: for threshold3

As a metric for **importance** of an edge species i , $\max_{t\in[0,t_{final}]}\left(\frac{flux_i(t)}{R_{char}(t)}\right)$ is calculated in each iteration. Any edge species with this metric larger than `toleranceKeepInEdge` at any time t will be kept in the model edge, i.e., it will not be pruned. Thus, the lower `toleranceKeepInEdge` is, the closer the pruned model will be to a non-pruning model due to fewer edge species being removed.

Naturally, the lower bound for `toleranceKeepInEdge` is 0, indicating no edge species will be deleted (non-pruning scenario); the upper bound is the value of `toleranceMoveToCore`, since `toleranceKeepInEdge` defines the boundary to identify unimportant species while `toleranceMoveToCore` selects the important ones.

2.1.2.4 `maxEdgeSpecies`

`maxEdgeSpecies` sets an upper bound for total number of edge species. Once it exceeds `maxEdgeSpecies`, RMG will start removing edge species with lowest $\max_{t\in[0,t_{final}]}\left(\frac{flux_i(t)}{R_{char}(t)}\right)$, regardless of being higher or lower than `toleranceKeepInEdge`, which will greatly help avoid memory crash but probably harm final model accuracy. It's always recommended to set `maxEdgeSpecies` as large as the computer can handle. In our experience, with 8 Gb of RAM an appropriate choice for `maxEdgeSpecies` is 100,000.

2.2 Results

Two high-temperature combustion systems were chosen to test the pruning algorithm (see Table 2.1); one has natural gas (a combination of methane, ethane and propane) as the fuel and O2 as the oxidizer while the other has n-heptane and O2. Both of them are in high equivalence ratios that usually lead to combination of oxidation and pyrolysis, generating significant amounts of intermediates, overflowing RAM with the usual algorithm. However, the two systems have very different chemistries: first system (hereafter, `NG`) goes through a chemical process where small molecules form large ones on the way to soot formation, while the second (hereafter, `C7`) has large molecules decomposing into small ones at the chosen reaction conditions (see Figure 2.6).

The new RMG with pruning was able to reduce the quadratic dependence of RAM on core size to nearly linear dependence, as Figure 2.7 shows.

For various jobs converged to a range of `toleranceMoveToCore` (which generates final models of various sizes), the pruning algorithm not only can generate identical models, but also requires much less memory than the original algorithm does (Figure

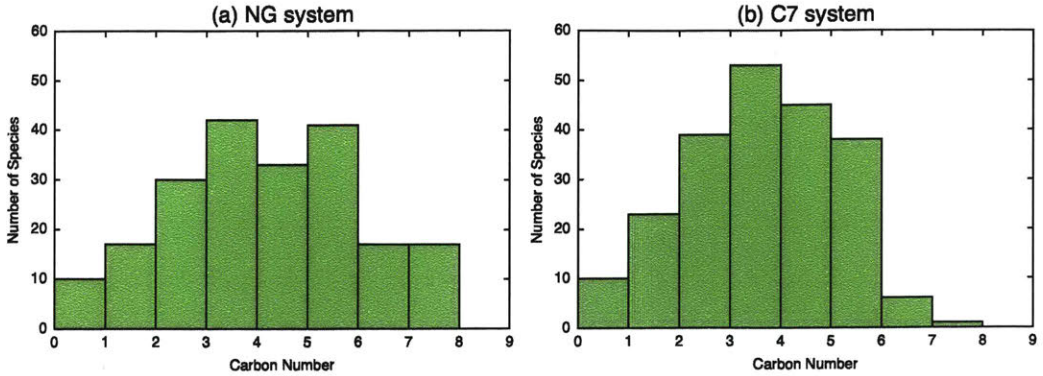


Figure 2.6: Species size (carbon number) distribution in NG and C7 systems in "core" model at convergence

system	fuel composition	fuel/O_2 equivalence ratio	T and P
NG	CH_4: 90 mol% C_2H_6: 8 mol% C_3H_8: 2 mol%	3.4	1400K, 20 atm
C7	C_7H_{16}: 100 mol%	11	1400K, 20 atm

Table 2.1: Specification of test chemistry

2.8). Pruning saves around 75% of memory for highly complicated systems (low `toleranceMoveToCore`), still generating exactly the same final models as the original algorithm.

On the other hand, the pruning algorithm makes it possible to generate large models which the original RMG was not able to; a model with 200 core species can quickly fill up 8Gb RAM using the original algorithm, while the pruning algorithm can easily generate models of 400 core species with memory usage no more than 5Gb, increasing the RMG modeling capability by at least a factor of two (see Table 2.2).

Table 2.2: Modeling capacity is greatly extended by pruning algorithm for both natural gas and n-heptane systems

system	maximum species number in model (RAM requirement) with original algorithm	maximum species number in model (RAM requirement) with pruning algorithm
NG	220 (12 Gb)	$\geq$ 433 (5 Gb)
C7	230 (8 Gb)	$\geq$ 531 (4 Gb)

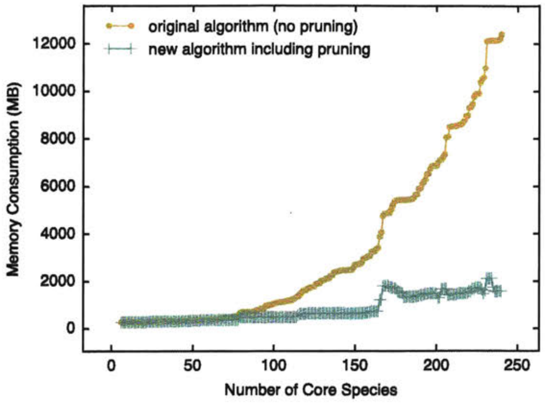


Figure 2.7: Pruning significantly reduces RAM requirements for the NG system of Figure 2.1

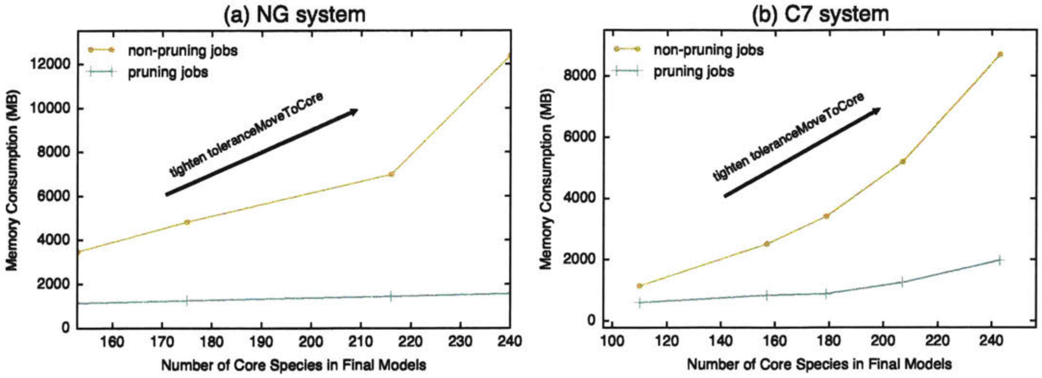


Figure 2.8: Comparison of RAM usage between original algorithm and pruning algorithm in two testing systems

In order to provide a more concrete example for pruning, phenyl dodecane (hereafter, PDD, see molecule structure Figure 6.1) thermal decomposition, an heavy oil-to-gas application with which RMG originally fails due to high memory cost, illustrates how pruning can make RMG keep searching and discovering important decomposition pathways.

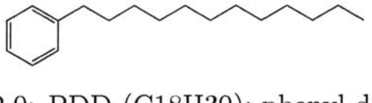


Figure 2.9: PDD (C18H30); phenyl dodecane

PDD with eighteen carbons, is one of the largest systems RMG has modelled; compared with smaller molecules, it has much more reacting sites, higher number of species RMG has to keep track of and therefore higher memory consumption. Consistent with previous testing cases, PDD job usually crashes around 200 species (hitting 8 Gb RAM limit) using RMG original algorithm (see Figure 2.10).

The mechanism misses some decomposition pathways, leading to slower conversion compared with experiment observations (see Figure 2.12); one important missing pathway (see Figure 2.11) is PDD decomposes to pentadiene after more than five consecutive steps, which can actively react with styrene and create radicals through reverse disproportionation reactions.

Original RMG (no pruning) has to explore and store whole species space with radius of > 5 steps to be able to find pentadiene but fails in the middle with memory filled up. With pruning turned on, RMG focuses on the most significant decomposition pathways, explores bigger species space and selects relevant pathways more efficiently, eventually discovers pentadiene pathway (Figure 2.11) with less than 3 Gb memory consumption, making the prediction of PDD conversion greatly improved (Figure 2.12).

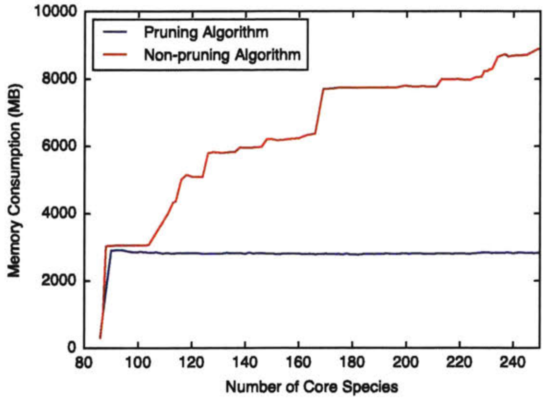


Figure 2.10: Pruning vs. Non-pruning memory consumption for phenyl dodecane decomposition

2.3 Discussion

2.3.1 Trade-off between effectiveness and accuracy

We are concerned about two aspects of the pruning algorithm: its **effectiveness** in reducing memory demands and its **accuracy** regarding final mechanisms generated. Quantitatively **effectiveness** and **accuracy** defined as follows:

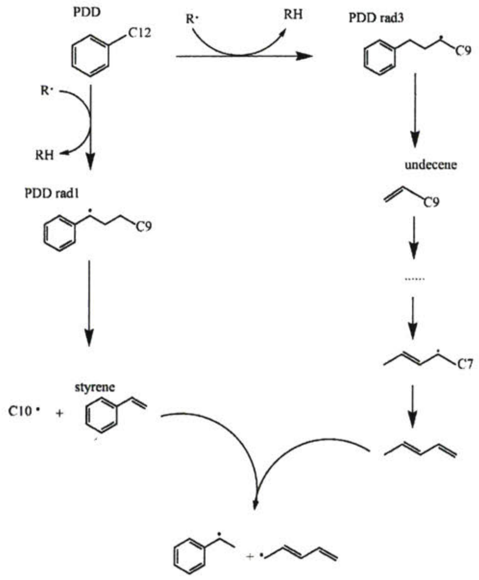


Figure 2.11: Important PDD decomposition acceleration pathway found by pruning

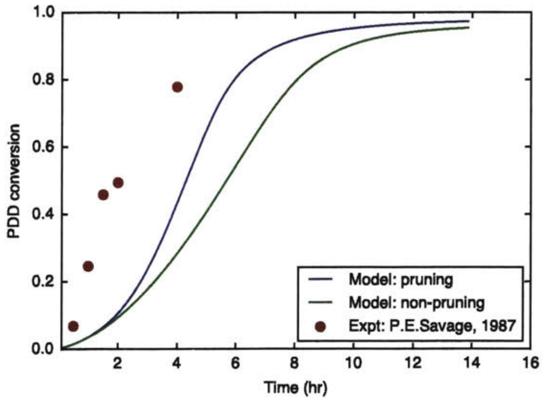


Figure 2.12: Pruning vs. Non-pruning phenyl dodecane conversion prediction

$$memory\ effectiveness = \frac{RAM_{NP}}{RAM_P}$$

$$accuracy = \frac{|Species_P \cap Species_{NP}|}{|Species_{NP}|}$$

where RAM_{NP} and RAM_P are memory requirements for building a model using the non-pruning model and pruning algorithms respectively, $Species_{NP}$ and $Species_P$ are species sets in non-pruning model and pruning model respectively.

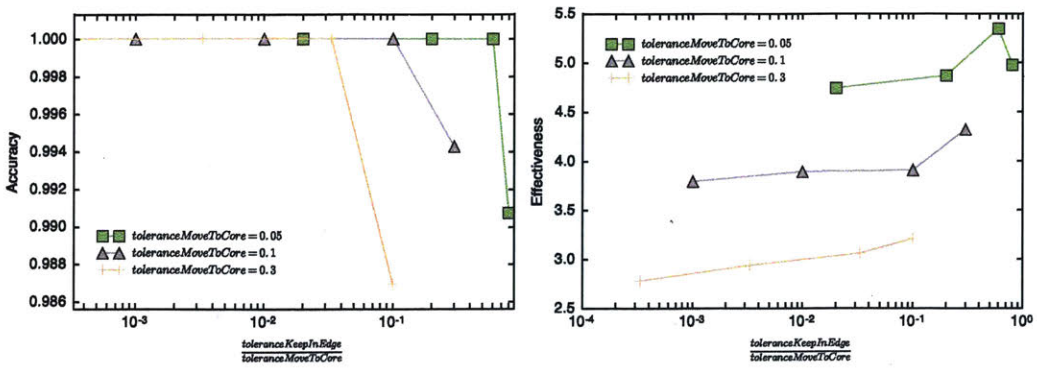


Figure 2.13: Effect of toleranceKeepInEdge in balancing pruning effectiveness and accuracy for NG system. Accuracy = 1 if pruning algorithm generates exactly identical model as non-pruning one does. Memory effectiveness is RAM reduction factor.

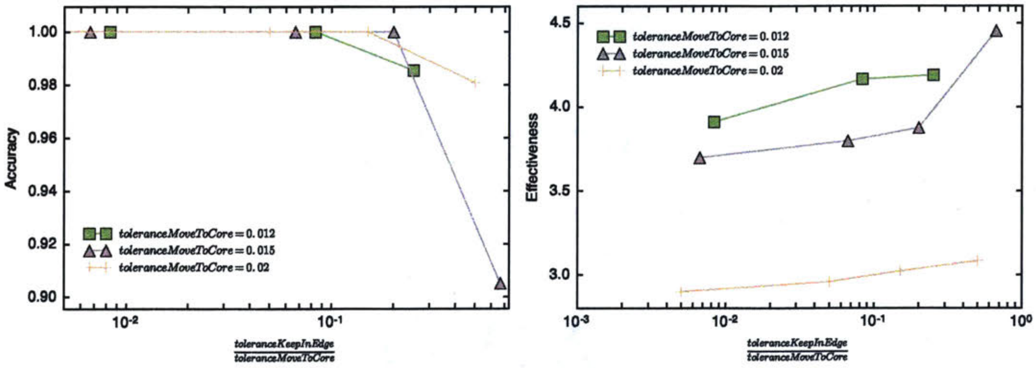


Figure 2.14: Effect of toleranceKeepInEdge in balancing pruning effectiveness and accuracy for C7 system. Accuracy = 1 if pruning algorithm generates exactly identical model as non-pruning one does. Memory effectiveness is RAM reduction factor.

In the cases presented in Figure 2.8, the final models produced by RMG-Py are exactly the same whether or not pruning is used, i.e. they have maximal accuracy = 1. In other extreme cases, pruning reaches maximal memory effectiveness by deleting all the edge species, leading to a very different final model from the non-pruning case. In pruning algorithms, `toleranceKeepInEdge` is the key handle balancing these two aspects: a loose `toleranceKeepInEdge` reduces accuracy, but also reduces RAM requirement.

Thus determining the value of `toleranceKeepInEdge` is crucial for pruning performance. It turns out for complicated NG and C7 systems (with small `toleranceMoveToCore`) that `toleranceKeepInEdge` being smaller than 1/10 of `toleranceMoveToCore` usu-

Scenario	Core species in final model	Core reactions in final model
non-pruning	153	5491
minCoreSizeForPrune=0	157	5840
minCoreSizeForPrune=50	153	5491

Table 2.3: Premature pruning impact. This example is NG system using toleranceMoveToCore=0.3 and toleranceKeepInEdge=0.01

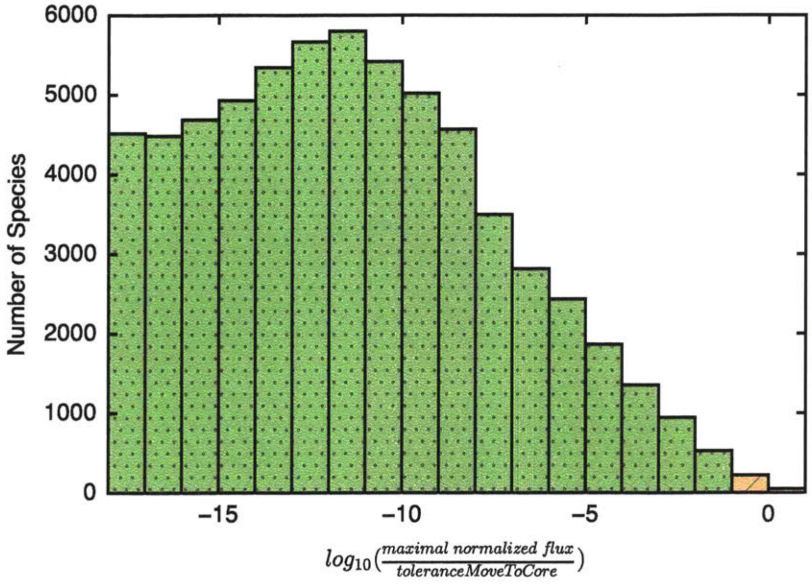


Figure 2.15: Maximal normalized flux ($max_t\left(\frac{flux_i(t)}{R_{char}(t)}\right)$) distribution of edge species at 180^{th} iteration in non-pruning scenario of C7 system. With pruning most of the species on the left would be deleted, freeing a lot of memory

ally gives good memory effectiveness and maintains the same models (accuracy=1), as Figure 2.13 and Figure 2.14 indicate.

By applying this rule of thumb (choose 1/10 of `toleranceMoveToCore` for `toleranceKeepInEdge`) to phenyl dodecane study, the pruning model not only keeps all the important pathways originally included in non-pruning model, but also discovers the pentadiene pathway that we discussed earlier.

Two additional components were designed to secure pruning **accuracy**. During early iterations, incomplete models usually result in inaccurate edge flux estimation, making the algorithm more likely to prune important species than it is when models are close to completeness (see Table. 3). It is observed that pruning at an early stage usually loses important species. Once an important species is lost, the model often grows in strange directions, including species which are not really important in the physical system. To prevent such undesired pruning, user-specified `minCoreSizeForPrune` is added to the pruning algorithm; no pruning is performed if the core species number is less than `minCoreSizeForPrune`. In both tested systems, `minCoreSizeForPrune` $= 50$ is a good choice.

A second consideration is edge species eligibility to be pruned. For a newly generated edge species, its associated reaction network is usually not fully developed, and the consequent low flux usually misleads pruning. To avoid that, RMG records the age of each edge species ("age" = number of iterations since the species is first identified) so that only those with age larger than the user-specified `minExistIterationForPrune` are eligible to be pruned. We recommend setting `minExistIterationForPrune` to 3.

2.3.2 Flux evaluates species importance

Pruning algorithms rely on accurate evaluation of edge species' importance. From Figure 2.15, we can clearly see by ranking all the edge species using maximal normalized flux $max_t\left(\frac{flux_i(t)}{R_{char}(t)}\right)$, that most species (colored in green) have small fluxes ($\leq$ 10% of `toleranceMoveToCore`). Our pruning algorithm removes them (dotted bars), based on the assumption that the fluxes to these minor byproducts are unlikely to change by orders of magnitude as the model is refined.

The original non-pruning algorithm [2] is designed conservatively, retaining all the edge species so if the flux towards any of the species gets large enough, that species will be added to the model. Our experience is that the reacting fluxes in the core and towards the edge species sometimes change significantly early in the mechanism-generation process, but after the major species have been included in the core, most of the major species concentrations and computed fluxes stabilize. It is unlikely that any of the fluxes will change by orders of magnitude because a few more minor species have been included in the model.

As shown in Figure 2.15, the fluxes towards most of the edge species are tiny, ten or more orders of magnitude smaller than the core species fluxes. It is therefore safe to delete these negligible species. Note that if a new reaction pathway towards a deleted edge species is discovered, the edge species will be resurrected, and maintained on the edge for at least `minExistIterationForPrune` to allow a fair re-assessment of its kinetic significance. We've observed that with reasonable tolerance values, the original rate-based algorithm and this pruned version yield exactly the same final models, but with very different memory requirements.

2.4 Conclusion

This chapter introduces pruning algorithm that largely mitigates the memory limitation associated with RMG. By first selecting unimportant edge species and pruning them and their reactions, the algorithm reduces memory consumption by a factor of 4 for cases where the original RMG algorithm runs into memory shortage; if with fixed computer memory it enables reaction mechanisms to contain about twice as many species as previously possible. Several special considerations regarding pruning eligibility were applied to reduce the risk of mistaken pruning. With the new pruning algorithm, it is practical to generate converged models for more complicated systems (high carbon number, high equivalence ratio, or low `toleranceMoveToCore`) than was possible in the past; it was able to construct a more complete mechanism for a real application (phenyl dodecane thermal decomposition) with important pathways that original RMG missed due to memory limitation.

The new pruning algorithm performed well with two typical combustion systems having distinct chemical behaviors. In order to save a significant amount of memory with minimal loss in final mechanism accuracy, only negilible edge species should be pruned. Follow-up study of the trade-off between effectiveness and accuracy suggests an appropriate ranges of the tolerances for RMG users to employ to execute the pruning algorithm properly.

2.5 References

[1] D. J. Klinke and L. J. Broadbelt. "Mechanism reduction during computer generation of compact reaction models." en. *AIChE Journal* 43 (7), July 1997, pp. 1828–1837. ISSN: 1547-5905. DOI: `10.1002/aic.690430718`.

[2] R. G. Susnow, A. M. Dean, W. H. Green, P. Peczak, and L. J. Broadbelt. "Rate-Based Construction of Kinetic Models for Complex Systems." *The Journal of Physical Chemistry A* 101 (20), May 1997, pp. 3731–3740. ISSN: 1089-5639. DOI: `10.1021/jp9637690`.

3

Mechanism Generation Speedup with Scalable Algorithms

Besides the memory usage issue covered in Chapter 2, RMG faces another challenge from superlinear increase of simulation time as reaction-network size grows; It can take weeks to months to complete network generation for large systems. On top of that, RMG's add-on features such as sensitivity analysis [1], on-the-fly quantum chemistry calculations for kinetics [2, 3] and uncertainty quantification [4] significantly increases the computational burden. In order to facilitate RMG's evolution towards modeling increasingly large systems and support many value-added features which are otherwise unaffordable in practice, it becomes a must that we speedup RMG simulation.

There are three primary aspects of RMG's slowness:

1) Application: Chemical systems with more components, more isomers and more reactive sites need much larger reaction networks to capture relevant chemistry. For instance, number of possible bimolecular reactions, whose generation is an inherently combinatorial computation problem, exponentially increases with system complexity.

2) Language: RMG uses Python as its main language, whose slowness caused by its flexibilty (e.g., dynamic typing) is widely known.

3) Algorithm: RMG simulation follows an iterative algorithm where all the steps are implemented in a serial execution fashion.

As extending RMG's modeling capability to large complex systems is the central goal of this thesis, we mainly focuse on the non-application aspects. For language, we have been introducing code optimization techniques throughout its codebase to reduce the computational penalty of programming in Python. Performance-critical code, such as the graph isomorphism algorithms has been converted to Cython [5], which compiles Python code to C through static typing, leading to over an order of magnitude or higher speed up for numerically intensive code. We also hook standard computational operations to compiled libraries. For instance, Ordinary Differential Equations (ODEs) simulation, as one key step in RMG workflow (see Section 3.1), is performed by DASSL [6] or DASPK [7], Fortran-based differential algebraic system solvers. Many of these strategies were already practiced in early versions of RMG and will not be further discussed in this chapter.

This chapter mainly summarizes the attempts that we explored to improve algorithm scalability.

3.1 Reaction network generation workflow

As detailed in Chapter 2, RMG uses the rate-based algorithm [8] to iteratively grow ("enlargement") the reaction mechanism by adding one species into it at a time. To obtain all information needed to decide which species to add, the enlargement procedure passes through four consecutive key phases: the reaction generation phase, the synchronization phase, the chemical data computation phase and finally the reaction system simulation phase (ODE solving) in which the reaction rate and species concentrations are evaluated, as shown in Figure 3.1.

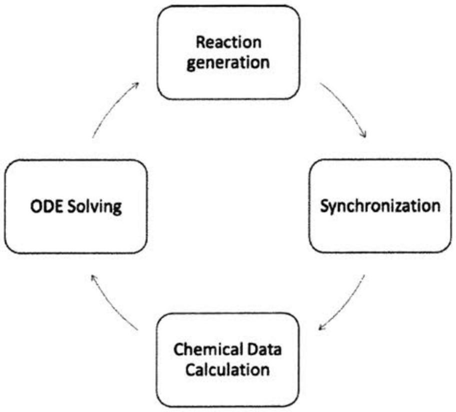


Figure 3.1: The enlargement step that integrates a species from the edge into the core consists of four consecutive key components: reaction generation, synchronization, calculation of chemical data such as thermodynamic properties of species and kinetics of reactions and finally solving the system of ODEs and calculating species concentrations and fluxes.

In the reaction generation phase, the species that was added in the previous enlargement iteration is used as a reactant to generate new reactions and species. This is done by matching the species to the reactant templates (e.g., pre-defined reactive sites) of a series of built-in reaction families. For bimolecular reaction families, this species is reacted with itself or with a species already present in the network. Reactants are transformed into products by applying reaction family-specific recipes with instructions to break or form bonds, gain or lose electrons, etc. The reaction generation step results in the creation of a large number of new reactions and species. These newly created product structures and reactions may be identical to those previously generated. Therefore, the synchronization phase is designed to consolidate the lists of new species and reactions and only allow the unique new ones to exist in memory; if a newly generated species is identical to a previous species, RMG proceeds by deleting the new structure and makes a reference to the object in memory that corresponds to that species. A similar mechanism is in place for reactions.

The consolidated lists of new species and reactions from the synchronization phase are further sent to the chemical data computation phase, i.e. the calculation of the thermodynamic properties of the species or the kinetic parameters of the reaction.

RMG selects edge species to the mechanism based on flux they draw. The ODE solving step facilitates the process by providing flux evaluation. RMG relies on the differential equation solver DASSL accessed through the PyDAS Python interface.

3.2 Performance bottlenecks

This section presents the identification of performance bottlenecks in RMG simulations, which helps tailor solutions toward the reduction of the time spent in these steps and hence overall wall clock time.

As explained in the previous section, RMG's iterative algorithm is composed of a loop that grows the mechanism by one species at a time. As an iteration uses the results computed in previous iterations, it is not possible to execute different iterations concurrently. Therefore, we focuse on performance improvements of the individual phases within one iteration.

A number of profiling studies were devised to identify the characteristics of the performance bottlenecks of RMG and aim at profiling the time spent in the four phases of a single enlargement iteration as a function of the state of the simulation and on the nature of the simulated process.

Three test RMG simulations are used to serve as benchmark cases (Table 3.1). They represent a diverse and complementary collection of simulations that make use of a large part of the features and settings of RMG. The first test application consists of a diesel oxidation simulation. Its initial fuel mixture consists of five large linear alkanes and n-decylbenzene giving rise to large number of generated reactions, even at early stages of the simulation. In the second application, the pyrolysis chemistry of a 1,3-hexadiene doped methane diffusion flame is modeled, which could be employed in studies of soot formation. On-the-fly quantum chemistry calculation (QMTP) is turned on to estimate thermodynamic properties for polycyclic species, which is critical in soot formation modeling. The third application is fuel-rich natural gas combustion containing methane, ethane, propane and oxygen as the initial reactants. Because the reactant molecules are small in size, a low number of reactions are created per iteration early in the simulation, in contrast to test problem one.

Table 3.1: Overview of the simulation specifications of the test cases used for the performance benchmarking.

Test application	Diesel oxidation	Hexadiene pyrolysis	Natural gas combustion
Initial mixture	diesel + air	hexadiene, CH_4, H_2, N_2	Natural gas + air
Condition	500K, 200 bar	1350K, 1 bar	1400K, 20 bar
Thermo estimator	GA	QMTP	GA

GA = Benson group additivity, QMTP = on-the-fly quantum chemistry for thermodynamic properties of species using PM7 from MOPAC 2012 [9]. Initial molar ratio for diesel oxidation: n-C11: n-C13: n-C16: n-C19: nC21: n-decylbenzene: O_2 = 1.00 : 1.27 : 1.67 : 1.20 : 0.67 : 0.80 : 0.33, hexadiene pyrolysis: 1,3-hexadiene : methane : H_2 : N_2 = 1 : 152 : 23 : 1288, natural gas combustion: CH_4 : C_2H_6 : C_3H_8 : O_2 : N_2 = 1.00 : 0.09 : 0.02 : 0.71 : 0.33.

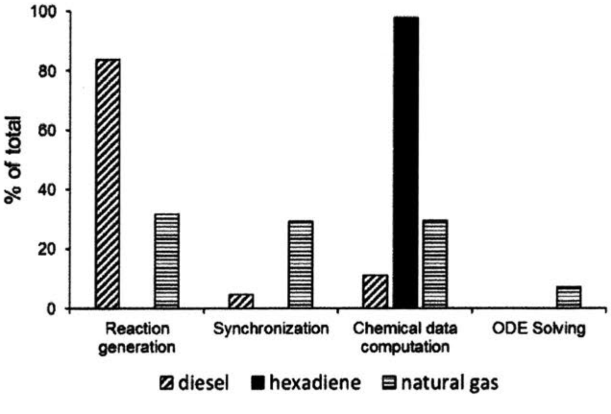


Figure 3.2: Relative contributions of the four phases of the enlargement procedure to the simulation time for the three test applications

Figure 3.2 shows the relative contributions of the four phases of the enlargement procedure to the simulation time, averaged over all enlargement iterations of the three test problems.

It can be observed in Figure 3.2 that the reaction generation phase is the performance bottleneck for simulations that involve reactants such as long n-alkanes with multiple reacting sites. In test case two, which employs the computationally expensive on-the-fly quantum chemistry method for estimating thermodynamic properties, the chemical data computation phase becomes limiting. Finally, as RMG simulations advance to larger mechanism sizes of several thousands of species and reactions on the edge, a significant amount of time is spent in the synchronization phase, verifying the uniqueness of newly generated species and reactions, as is the case for the natural gas combustion problem. The profiling results indicate that the computational load of RMG simulations does not follow a single static pattern, but rather heavily depends on the type of simulation. As a result, multi-faceted strategies that focus on the computational performance across the wide spectrum of simulations are considered to reduce the wall clock time.

3.3 Speedup strategies

This section summarizes the strategies we followed to speedup reaction generation phase (Subsection 3.3.1) and chemical data computation phase (Subsection 3.3.2)

3.3.1 Speedup reaction generation

As generally dominated by bimolecular reaction generation, reaction generation phase scales on the order of $m^2_{\text{reaction sites}} \cdot n^2_{\text{core species}} \cdot l^2_{\text{resonance isomers}} \cdot f_{\text{bimolecular reaction families}}$ where n is the number of total core species, m and l are average numbers of reaction sites and resonance isomers in a core species respectively and f is the number of bimolecular reaction families. That makes reaction generation step usually the performance bottleneck for large systems, which is also confirmed by the diesel test application.

We approached this bottleneck with two speedup strategies:

1) Reaction filtering: use heuristic to filter out reactions with low expected fluxes and only generate high-flux reactions. This strategy aims to reduce reaction generation workload.

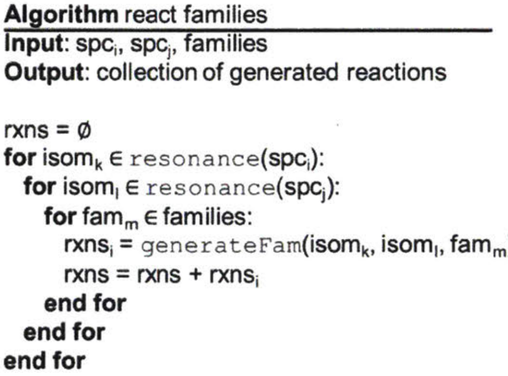
2) Parallelization: reaction generation is a perfectly parallel problem which can be divided into independent small tasks. This strategy aims to spread reaction generation workload across multiple CPUs.

3.3.1.1 Reaction filtering

This algorithm implemented by Dr. Gao [4], is similar in nature to the pruning algorithm (discussed in Chapter 2) which prunes species with low incoming fluxes. Since reactions with high fluxes are deemed important in RMG's rate-based enlargement scheme, the concentration of a species, greatly affecting reaction fluxes, can be treated as an *importance* indicator for the reactions to be generated. By choosing an upper bound for rate constant, the algorithm avoids generating unimportant reactions for a species or a pair of species if the highest achievable reaction rates are below a threshold.

3.3.1.2 Parallelization

Instead of reducing workload heuristically, the parallelization strategy attempts to discretize original computation into independent tasks and spread them across multiple CPUs. Fortunately, reaction generation is a perfectly parallel problem, as shown in Figure 3.3 where `generateFam` is an independent and atomic task in the three nested loops.



```
**Algorithm** react families
**Input**: $spc_i$, $spc_j$, families
**Output**: collection of generated reactions

rxns = ∅
for $isom_k \in$ resonance($spc_i$):
  for $isom_l \in$ resonance($spc_j$):
    for $fam_m \in$ families:
      $rxns_i$ = generateFam($isom_k$, $isom_l$, $fam_m$)
      rxns = rxns + $rxns_i$
    end for
  end for
end for
```

Figure 3.3: Algorithm outline for the generation of bimolecular reactions for a given pair of species. The algorithm accepts two species and iterates over the resonance isomers of each species in the two outermost loops. The innermost loop iterates over the reaction families available to RMG. The algorithm for unimolecular reactions is not depicted but follows a similar pattern.

The first two outermost loops iterate over all combinations of resonance isomers of the chosen species pair (every species consists of one or more resonance isomers). The innermost loop iterates over the reaction families that contain the family-specific recipes to transform reactant structures into product structures. The task `generateFam` returns zero or more reactions, depending on the number of reacting sites identified in the reactant structures.

The two outermost loops are parallelized, cf. Figure 3.4. Given N species to react, this results in the creation of N and $N \cdot (N+1)/2$ tasks for uni- and bimolecular reaction families respectively, if one resonance isomer per species is assumed. It is opted not to parallelize across the reaction families with the aim of creating tasks with a computational load that is more uniformly distributed. In most applications explored by RMG, the majority of the reactions are generated through only a limited number of reaction families.

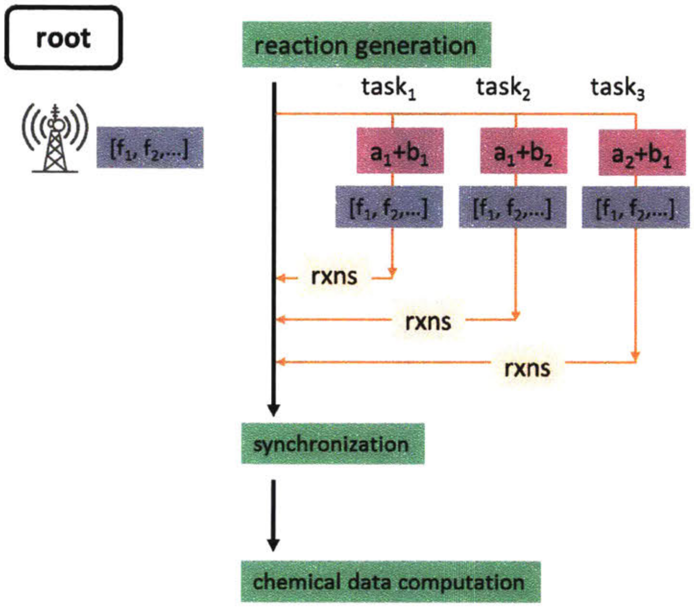


Figure 3.4: The parallelization scheme for the "reaction generation" phase as part of one enlargement iteration in RMG. The scheme distributes the generation of reactions by creating one task per molecule-molecule (a_i, b_i) pair and subsequently creates reactions through application of the list of reaction families [f_1, f_2, ...]. The generated reactions created per task are sent to the root process, bundled together and further processed in the synchronization phase.

Because CPython has the Global Interpreter Lock (GIL) that prevents the simultaneous execution of multiple threads (shared memory), the parallel execution of tasks is achieved through multiple processes with separate memory spaces. Data communication between processes occurs through message passing.

A task scheduler library, Scalable COncurrent Operations in Python (SCOOP) v0.7.1 [10], is used to dynamically load-balance the tasks across available computational resources. In SCOOP, a central "broker" process mediates communications between worker processes that each run a distinct Python interpreter. Submitted tasks are added to a broker queue through function calls such as `map` and `submit` and are subsequently distributed across the pool of available worker processes. For data communication between processes, SCOOP uses ZeroMQ, a lightweight library for message passing.

3.3.2 Speed up chemical data calculation

The computation of chemical data becomes a performance bottleneck when computationally demanding features such as on-the-fly quantum chemistry methods are used to estimate thermodynamic properties of species rather than computationally light group additive methods. Furthermore, the chemical data computing phase may become more demanding in the future if more advanced quantum chemical methods replace the faster but less accurate semi-empirical methods currently supported. To reduce the contribution of the chemical data computation phase within the overall simulation time, a scheme is devised that allows overlapping synchronization with chemical data calculation, cf. Figure 3.5.

In this scheme, a concurrent task is spawned and sent to a worker that calculates the required chemical data whenever the synchronization component discovers a new species or reaction. Since the creation of such chemical data tasks is non-blocking, the synchronization of newly generated species and reactions continues while independent tasks are processed on the available pool of workers. Whenever the chemical data is required, e.g. for solving the system of ODEs, the task responsible for the calculation of the data is retrieved and the data is requested through a blocking call. Similar schemes can be devised for other computationally intensive components of chemical data computation, such as the estimation of pressure-dependent rate coefficients.

3.4 Results and Discussion

To get better assessments on scalability improvements offered by the concurrent strategies, we carried out isolated experiments (Subsection 3.4.1) to evaluate improvements for specific bottleneck and RMG simulations (Subsection 3.4.2) for overall improvements.

3.4.1 Isolated experiments

In the isolated experiments, i.e. not embedded in a RMG simulation, we measured the elapsed wall clock time and parallel efficiency (defined as Eq. 1) under strong scaling setting (scaling with the number of cores when the problem size is fixed) for

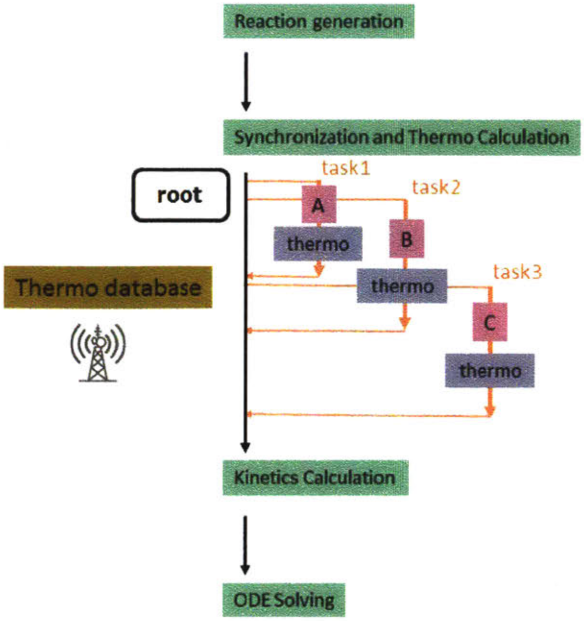


Figure 3.5: Asynchronous calculation of thermodynamic properties of species during the synchronization phase of an enlargement iteration. Whenever a new species is handed off from the synchronization phase, a task is spawned on a remote worker in which the thermodynamic properties of the species is calculated. When the task is finished, the thermodynamic properties are sent back to the original species object. Databases used to estimate thermodynamic properties are broadcasted to all workers at the startup of the simulation.

the calculation of thermodynamic properties (470 unique hydrocarbons with MOPAC 2012 [9] at the PM3 level) and generation of reactions (500 reactions by the application of 24 reaction families on 100 molecule pairs).

$$E = \frac{T_1}{m \cdot T_m} \tag{1}$$

with T_1 and T_m the elapsed wall clock time of the experiment using 1 and m workers respectively.

Figure 3.6A shows how the elapsed wall clock time to calculate the thermodynamic properties of 470 hydrocarbons decreases from approximately 3600 seconds using one core to 240 seconds using 24 cores, which translates in a speed-up factor of 15. Figure 3.6B shows how the elapsed wall clock time to generate approximately 500 reactions for 100 molecule-molecule pairs by the application of 24 reaction families requires 35 seconds using one core, but only 1.4 seconds with 48 cores, which translates in a speed-up factor of 25.

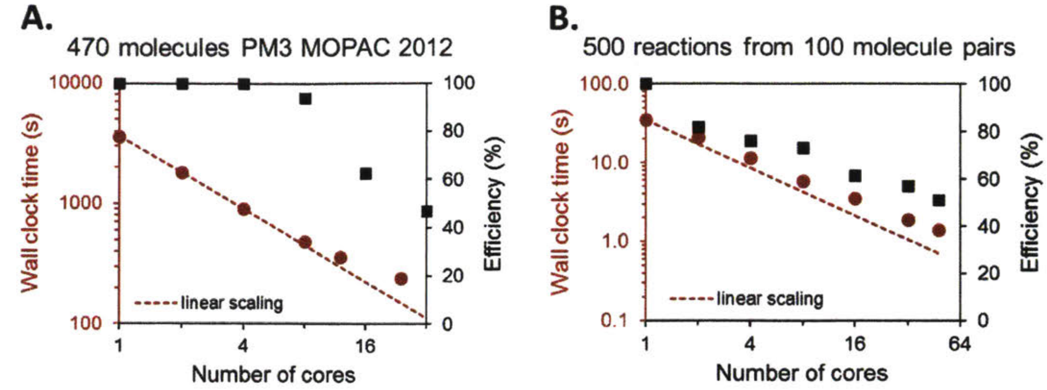


Figure 3.6: Strong scalability characteristics for A. the calculation of thermodynamic properties and B. the generation of reactions. Left vertical axis shows the elapsed wall clock time, dotted line denotes the ideal, linear scaling. Right vertical axis shows parallel efficiency

We also observed parallel efficiency gradually decrease for both cases from initial 100% to approximately 50% after using 48 cores. The decreased parallel efficiency is possibly caused by communication between the master and the slave CPUs, contention for shared resources and unbalanced tasks.

3.4.2 RMG simulations

Given the scalability characteristics of the individual components, we now assess the impact of the new algorithmic scalability improvements on RMG simulations. The wall clock time is compared for two versions of RMG: RMG v2.0.0 containing the modified algorithms for improved scalability is compared against the benchmark RMG v1.0.0 without these improvements. Figure 3.7 shows the wall clock time as a function of the number of core species for the three test cases. The individual contributions of the concurrent computing and reaction filtering algorithm to the speed-up are indicated.

It can be seen that the new version of RMG is significantly faster than v1.0. In the diesel oxidation case, a speed-up factor of 20 with the new, parallel version using 1 worker is calculated after 50 species are added to the core. For instance, the reaction filtering heuristic results in a drastically smaller edge containing 1218 species and 10043 reactions, while the original version contains 10361 species and 47879 reactions at the same point of the simulation with 30 species added to the core. This acceleration effect becomes more amplified at later stages of the simulation. The new version of RMG using 8 workers results in an additional speed-up factor of two. The

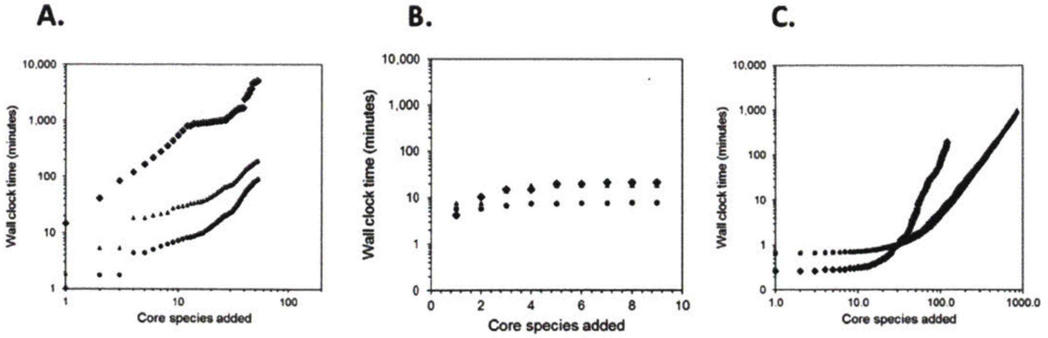


Figure 3.7: Wall clock time as a function of the number of species added to the core. ♦: RMG v1.0 (benchmark) ▲: RMG v2.0 with 1 worker ●: RMG v2.0 with 8 concurrent workers A. Diesel oxidation (test application 1) B. 1,3-hexadiene doped methane pyrolysis (test application 2) C. Natural gas combustion (test application 3).

speed-up through concurrent computing may seem underwhelming compared to the scalability metrics shown for the isolated experiments. However, in earliest stages of the RMG simulation only a small number of species are present in the core, resulting in fewer tasks than available workers. In addition, as Amdahl's law [11] dictates, the maximum achievable speed-up in an enlargement iteration is governed by the time spent in the serial parts of the code. The 1,3-hexadiene pyrolysis case shows that significant speed-ups can be observed even for very small mechanisms when the calculation of chemical data is expensive. As can be observed from Figure 3.7B, most of the speed-up originates from the concurrent workers, which are employed in the calculation of thermodynamic properties. Figure 3.7C highlights that the scalability improvements within RMG also significantly impact wall clock times for systems of smaller size reactants such as natural gas combustion. The higher wall clock time at the initial stages of the natural gas combustion simulation with 8 workers reflect the start-up costs associated with setting up the concurrent environment and broadcasting the static databases to the available workers. The new version using 8 workers is only 20% faster than the version using only 1 worker, which is attributed to the increasingly large contribution of the reaction network synchronization phase in the overall simulation time.

In addition to the test cases, the new version of RMG was also applied to the main application in this thesis (Chapter 6): thermal decomposition of phenyl-dodecane (hereafter, PDD, Figure 6.1). The C18 hydrocarbon is by far one of the largest systems RMG has modelled. Figure 3.8 shows new RMG has significantly improved scalability for PDD application, where it nearly halves the trend line slope from 6.8 to 3.5.

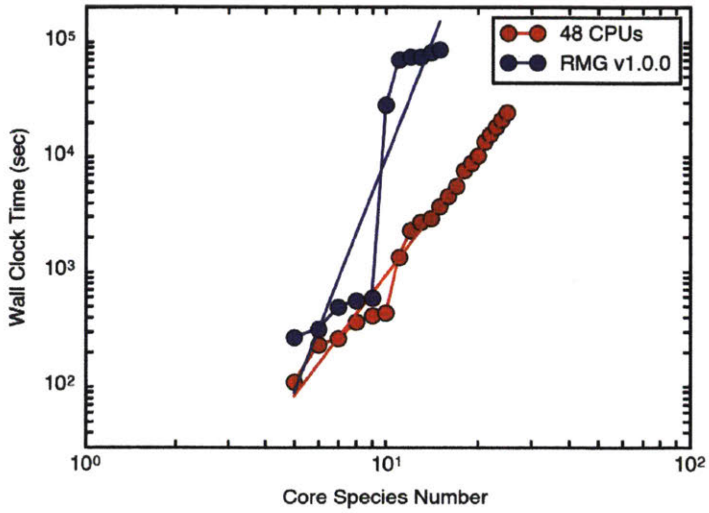


Figure 3.8: Wall clock time as a function of the number of species added to the core for early PDD thermal decomposition. RMG v.1.0 (benchmark) is colored blue, RMG v2.0 with 48 workers is colored red. Linear trend lines are added to show scalability improvements: blue line has slope of 6.8, while red line 3.5

3.5 Conclusion

Achieving efficient generation of reaction networks for complex systems has been a long standing challenge. With the concerted efforts outlined in this chapter, we have identified critical bottlenecks and provided scalable solutions to make computer-aided reaction mechanism construction a tool of practical usefulness. It was shown that bottlenecks for the overall performance of RMG simulations are dynamic in nature and depend on the type of the simulation as well as methods used for the calculation of chemical data. Code optimization, concurrent computing and algorithm heuristics are three key strategies to accelerate the essential components of the rate-based enlargement procedure.

Despite the iterative nature of the rate-based enlargement of RMG, integrated simulations are now significantly faster, even for small mechanisms, relative to the original version. Speed-up tests that measured individual scalability characteristics have shown speed-up factors with respect to the use of a single core of 15 and 25 for the calculation of thermodynamic properties and generation of reactions respectively.

Overall, this chapter and Chapter 2 together demonstrate how the new version of RMG opens up new opportunities for the construction of more comprehensive and more accurate mechanisms of chemical processes and creates avenues for modeling real world processes that previously were too complex to model. The scalability improvements described in this chapter are implemented in the latest version of RMG found at http://reactionmechanismgenerator.github.io/RMG-Py/.

3.6 References

[1] C. W. Gao, J. W. Allen, W. H. Green, and R. H. West. "Reaction Mechanism Generator: Automatic construction of chemical kinetic mechanisms." *Computer Physics Communications* 203, June 2016, pp. 212–225. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2016.02.013.

[2] Y. V. Suleimanov and W. H. Green. "Automated Discovery of Elementary Chemical Reaction Steps Using Freezing String and Berny Optimization Methods." *Journal of Chemical Theory and Computation* 11 (9), 2015. PMID: 26575920, pp. 4248–4259. DOI: 10.1021/acs.jctc.5b00407. eprint: https://doi.org/10.1021/acs.jctc.5b00407. URL: https://doi.org/10.1021/acs.jctc.5b00407.

[3] P. L. Bhoorasingh and R. H. West. "Transition state geometry prediction using molecular group contributions." en. *Physical Chemistry Chemical Physics* 17 (48), Dec. 2015, pp. 32173–32182. ISSN: 1463-9084. DOI: 10.1039/C5CP04706D. URL: http://pubs.rsc.org/en/content/articlelanding/2015/cp/c5cp04706d.

[4] C. W. Gao. "Automatic reaction mechanism generation: High Fidelity Predictive Modeling of Combustion Processes." eng. Thesis. Massachusetts Institute of Technology, 2016. URL: http://dspace.mit.edu/handle/1721.1/104205.

[5] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith. "Cython: The Best of Both Worlds." *Computing in Science Engineering* 13 (2), Mar. 2011, pp. 31–39. ISSN: 1521-9615. DOI: 10.1109/MCSE.2010.118.

[6] L. R. Petzold. *Description of DASSL: a differential/algebraic system solver.* English. Tech. rep. SAND-82-8637; CONF-820810-21. Sandia National Labs., Livermore, CA (USA), Sept. 1982. URL: https://www.osti.gov/biblio/5882821.

[7] S. Li and L. Petzold. *Design of New Daspk for Sensitivity Analysis.* Tech. rep. Santa Barbara, CA, USA: University of California at Santa Barbara, 1999.

[8] R. G. Susnow, A. M. Dean, W. H. Green, P. Peczak, and L. J. Broadbelt. "Rate-Based Construction of Kinetic Models for Complex Systems." *The Journal of Physical Chemistry A* 101 (20), May 1997, pp. 3731–3740. ISSN: 1089-5639. DOI: 10.1021/jp9637690.

[9] J. J. Stewart. “Mopac2012.” *Stewart Computational Chemistry, Colorado Springs, CO, USA*, 2012.

[10] Y. Hold-Geoffroy, O. Gagnon, and M. Parizeau. “Once You SCOOP, No Need to Fork.” *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment.* XSEDE ’14. New York, NY, USA: ACM, 2014, 60:1–60:8. ISBN: 978-1-4503-2893-7. DOI: 10.1145/2616498.2616565. URL: http://doi.acm.org/10.1145/2616498.2616565.

[11] G. M. Amdahl. “Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities.” *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference.* AFIPS ’67 (Spring). New York, NY, USA: ACM, 1967, pp. 483–485. DOI: 10.1145/1465482.1465560. URL: http://doi.acm.org/10.1145/1465482.1465560.

4

Thermochemistry Estimation for Polycyclics: Heuristic Method

During kinetic mechanism generation, RMG has to explore a large space of molecules; it typically scans $10,000 \sim 1,000,000$ of species during a single run for chemical systems with average size. As the modelled system complexity grows, we can expect an even greater number. Such large scale of screening for which even cheap DFT methods are not affordable, requires fast estimation of molecular properties, such as thermochemistry. Group additivity method has served as a backend of RMG's main thermochemistry estimator with major advantages of convenience and speed. An enthalpy estimation scheme is depicted via Eq. 4.1.

$$H_f(298K) = \sum_{i=1}^{N_{atom}} GAV_i \tag{4.1}$$

where GAV_i is group additivity value for i^{th} atom centered group.

However, due to its underlying assumption that each atom-based group is independent and their contributions are additive, group additivity methods have difficulty estimating the thermochemistry of cyclic molecules, since ring strain is a joint effect among many ring atoms that is beyond single-atom-based scope. The inaccurate estimation is actually a hidden cause to RMG's convergence difficulty; an error in thermo estimation for a critical species can mislead RMG to explore unnecessary pathways, indirectly wasting computer power and memory.

To improve estimation accuracy for cyclics, Benson [1], Constantinou and Gani [2] proposed ring corrections on top of the normal atom-based group additivity scheme (Eq. 4.2).

$$H_f(298K) = \sum_{i=1}^{N_{atom}} GAV_i + \sum_{j=1}^{N_{ring\ cluster}} (ring\ correction)_j \tag{4.2}$$

where (*ring correction*)$_j$ is additional strain contributed by ring cluster j as a whole.

Note a ring cluster may consist of several individual rings that share at least one atom with at least one other individual ring in the cluster. To make accurate predictions, Eq. 4.2 requires correction data for every ring cluster in each molecule.

Since each ring cluster structure has its specific ring correction, and there are an extremely large number of possible fused ring clusters, this group additivity method only gives accurate predictions for molecules whose ring structures have been studied in the past. Estimation accuracy drops significantly when dealing with molecules with ring cluster structures not included in the database.

The root problem is that one cannot list the infinite number of possible ring clusters and prepare all the ring corrections. Due to the difficulty in acquiring data from ab initio calculations or experimental measurements, less data is available for molecules with larger ring clusters than for those with smaller ones. However, as cluster size increases more possible structural variations exist, which worsens the situation for estimating large polycyclics.

Therefore, we divide the problem of accurately estimating the thermochemistry of a polycyclic into two sub-problems based on the size of the ring cluster (number of smallest rings in the cluster, using Fan's algorithm [3] of Smallest Set of Smallest Rings) in the molecule:

- small cyclics ($\leq$ 2-ring molecules) and

- large cyclics ($\geq$ 3-ring molecules)

For the former problem, we calculate and organize the available ring corrections into a functional group tree that can find similar matches for any new small cyclics. For the latter problem, we develop a bicyclic-decomposition model which estimates large polycyclic ring cluster corrections by decomposing them into smaller ones and adding up the contributions from the fragments. Overall, we managed to bring down group additivity thermo prediction error from over 60 kcal/mol in some cases (original group additivity method in cases where the ring cluster structure of interest had not been studied previously) to 5 kcal/mol for both small cyclics and large cyclics as judged using the dataset of Ramakrishnan, et al. [4].

In this chapter, we discuss our similarity match approach in Section 4.2 and our bicyclic-decomposition approach in Section 4.3. Additionally, to power these algorithms, we organize and precalculate ring corrections for a list of frequently seen ring cluster structures, with more details in Section 4.1.

4.1 Pre-calculation

The precalculated list of hydrocarbon molecules covers molecules with mostly small ring cluster structures (1-ring and 2-ring clusters, see Supporting Information). Some example molecules are shown in Figure 4.1.

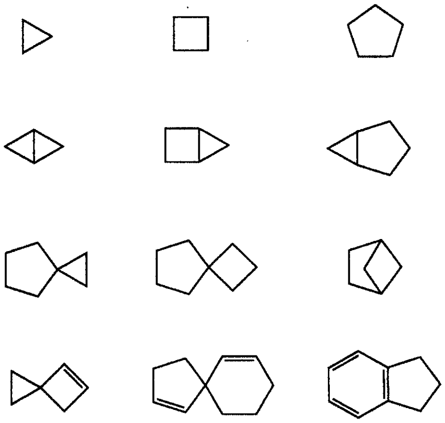


Figure 4.1: Example small cyclics in our database

To automate the data preparation process, a 3-step scheme was used as shown in Figure 4.2. Firstly, molecular identifiers are fed into RDKit Chem module [5] to generate initial XYZ coordinates. A GAUSSIAN job creator receives the molecular coordinates, composes GAUSSIAN 09 [6] job inputs and launches quantum chemistry jobs. To optimize the geometry and to compute vibrational frequencies at the optimized geometry XYZ_{opt}, we used the DFT method at M06-2X/cc-pVTZ level of theory [7]. Once the quantum chemistry calculation finishes, RMG's Cantherm module [8] (for more detail on cantherm information, visit `http://reactionmechanismgenerator.github.io/RMG-Py/users/cantherm/index.html`) parses the output GAUSSIAN log file and calculates the thermochemical parameters such as H_f(298 K), $C_p$and S(298 K) using the Rigid Rotor Harmonic Oscillator (RRHO) approximation. At each step, molecular representations (SMILES, XYZ, and optimized XYZ coordinates) are converted into RMG species objects and RMG's isomorphism check ensures that they still represent the same molecule.

Note that the single structure RRHO approach employed here only considers one conformer, and ignores anharmonicity, so it is expected to underestimate S and C_p for floppy rings.

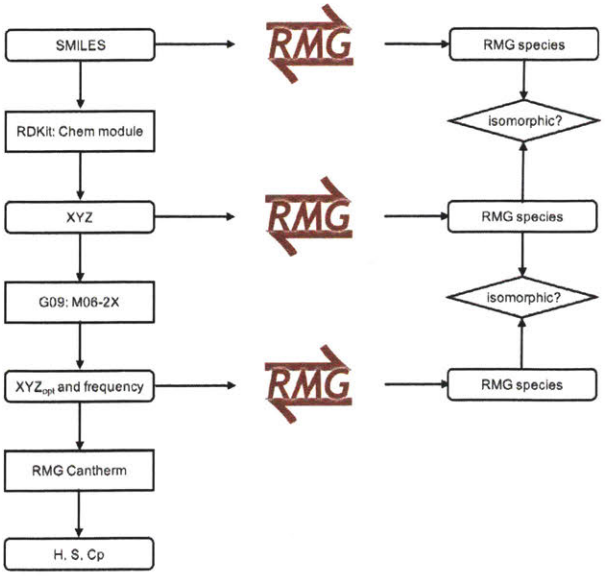


Figure 4.2: Quantum calculation scheme for small cyclic thermochemistry

4.2 Similarity Match

As discussed in the previous section, Eq. 4.2 needs exact matches of target ring clusters (otherwise no correction is applied at all), and thus requires extensive data to ensure high prediction accuracy.

Here we propose a similarity match algorithm that can find a similar ring with similar thermochemistry for situations where no exact matches are available.

4.2.1 Cyclic trees

The similarity algorithm greatly relies on data organization. The ring structures are organized into trees (we have two trees so far: monocyclic tree and polycyclic tree), see a sub-tree example in Figure 4.3. Nodes further down the tree have more specific structural details. The top layer defines the skeleton frame, for instance, s1_3_6 represents a bicyclic consisting of 3-member ring and 6-member ring with 1 atom shared, and the next layer defines categories such as alkane, alkene, diene or aromatics. Finally, the bottom layer lists the most specific ring structures.

With this design, if a new molecule does not exactly match a known ring cluster, it can be classified as similar to some other nodes in the tree, and assigned the average of their values. For instance the molecule in Figure 4.4 most closely matches the second-layer node s1_3_6_diene in Figure 4.3. Since there is no exact match, the algorithm will use the average of the ring corrections of the children of s1_3_6_diene

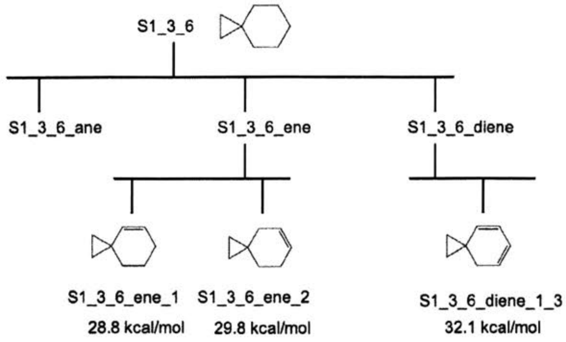


Figure 4.3: Example sub-tree that organizes polycyclic ring corrections with derived ring correction for enthalpy of formation

as its estimated correction.

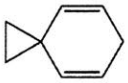


Figure 4.4: Example molecule which does not exactly match any node in the tree in Figure 4.3. The tree gives enthalpy estimation of 32.1 kcal/mol, while its real enthalpy of formation from quantum calculation in this study is 27 kcal/mol, leading to around 5 kcal/mol error

4.2.2 Model test

To evaluate the performance of this similarity match algorithm, an external large quantum calculation dataset [4] is used. This dataset contains 134,000 molecules and has enabled several interesting big data studies [9, 10] that connect machine learning models to molecular property estimation. This chapter selects cyclic molecules in that dataset as the test dataset (named `polycyclic_2954_table`) and categorizes the cyclics into small cyclics (1-ring and 2-ring molecules, see example in Figure 4.5(a)), large linear cyclics (at least 3-ring molecules, and atoms are at most shared by two rings, see example in Figure 4.5(b)) and large fused cyclics (at least 3-ring molecules, rings are heavily fused (having atoms shared by at least 3 rings), see example in Figure 4.5(c)).

With the polycyclic tree (mostly small cyclics), the similarity match algorithm can successfully reduce the mean absolute error of H_f(298 K) from 32 kcal/mol (original group additivity method in cases where the ring cluster structure of interest had not been studied previously) to 3 kcal/mol for small cyclics in the test dataset by Ramakrishnan, et al. (see Table 4.1 and Figure 4.6) [4], which is expected since the tree includes many pre-calculated small cyclic corrections.

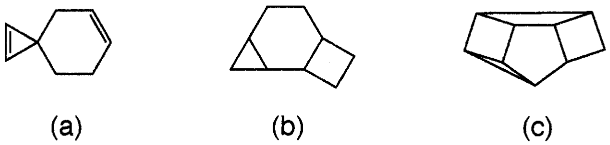


Figure 4.5: Example cyclics in each category: (a) small cyclics, (b) large linear cyclics, and (c) large fused cyclics

Table 4.1: Mean absolute error (kcal/mol) of H_f(298 K) for each category in validataion dataset

Method	small cyclics	large linear cyclics	large fused cyclics
Group Additivity Method	32	65	80
+ Similarity Match	3	29	40
+ Bicyclic Decomposition	3	4.9	9.8

Even though there are no large polycyclics in the tree, this similarity match approach also improves the predictions for large polycyclics by sub-molecule isomorphism (see an example in the Supporting Information for how a 3-ring molecule matches a 2-ring node if no 3-ring node available in tree), cutting the mean absolute error by about a factor of two (Table 4.1).

To further improve the accuracy of large cyclics thermochemistry prediction, obvious approaches would require pre-calculated data on large cyclics. However, the number of possible large polycyclics increases rapidly with the number of rings. To avoid this poor scaling, we built a model that estimates ring corrections of polycyclics from known corrections for bicyclics, as discussed in Section 4.3.

4.3 Bicyclic Decomposition

4.3.1 Method development

Having available ring correction data mostly for small cyclics, a model that estimates the thermochemistry of large cyclics from small cyclic building blocks is needed. We tried three methods, as shown in Figure 4.7.

Method (a) simply sums up ring correction contributions from single rings that make up the targeted large cyclics. One main drawback of this method is that it only counts the ring strain contributions from individual rings and overlooks the extra strain from the fused part of the bicyclic ring AB in Figure 4.8. That is

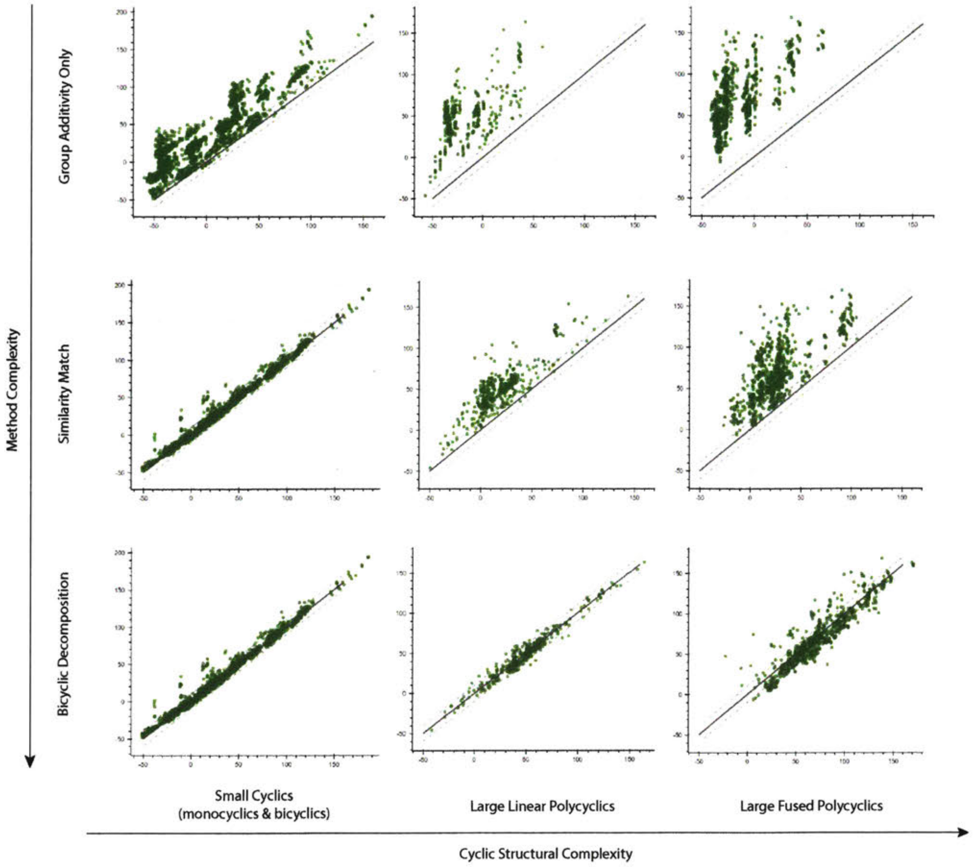


Figure 4.6: Predictions of enthalpy of formation (x axis) vs. quantum mechanics values (y axis, calculated by Ramakrishnan using DFT method B3LYP/6-31G(2df,p)) with various models and cyclic types, unit: kcal/mol, dataset: `polycyclic_2954_table`

the reason method (a) underestimates the ring corrections by over 60 kcal/mol in some cases (Figure 4.7). Significant discrepancy is observed between actual ring corrections needed to accurately estimate bicyclics and the sums of individual ring strain corrections (Figure 4.9).

Method (b) in Figure 4.7 divides a large cyclic into bicyclic components (called bicyclic decomposition), which automatically captures thermo contributions from fused parts. For instance, it decomposes a tricyclic into two bicyclics and estimates its ring corrections by taking the sum of bicylic corrections.

Method (b) reduces the error in predicted enthalpy of formation of tricyclo-octane to 27 kcal/mol by adding ring strain contributions from two fused parts in the tricyclic

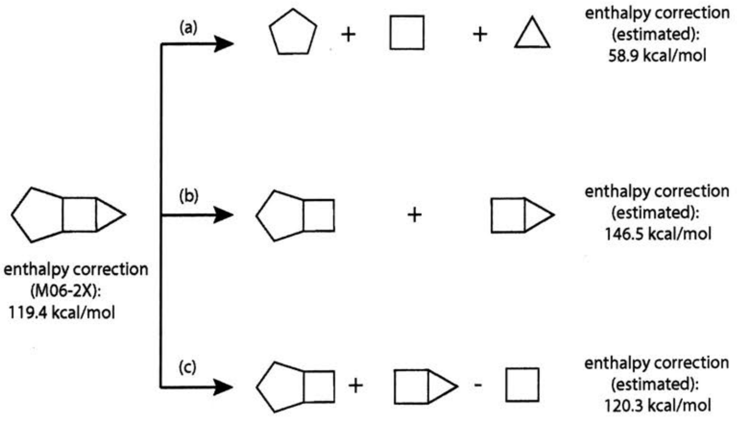

(a)
+
+
enthalpy correction (estimated): 58.9 kcal/mol
(b)
+
enthalpy correction (estimated): 146.5 kcal/mol
enthalpy correction (M06-2X): 119.4 kcal/mol
(c)
+
-
enthalpy correction (estimated): 120.3 kcal/mol

Figure 4.7: Large cyclic corrections estimation method evolution: (a) sum of individual single ring corrections, (b) sum of bicyclic corrections, and (c) sum of bicyclic corrections with overlapping ring correction subtraction

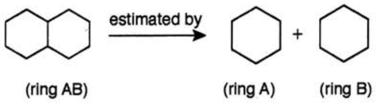


Figure 4.8: bicyclic AB correction estimated as sum of single ring A and B's corrections

(Figure 4.7). However, it always over-predicts the enthalpy corrections, due to the fact that method (b) double counts the contribution of the middle 4-member ring. By eliminating the overlapped ring correction, method (c) shown in Figure 4.7 calculates a ring strain that agrees well with "true" ring correction (here we use our M06-2X calculations as "true" values). In this particular example, the prediction error is remarkably reduced to 0.9 kcal/mol by adopting the bicyclic decomposition approach. In Subsection 4.3.3, we conducted a more thorough test of the performance of method (c) .

4.3.2 Bicyclic correction estimation

We attempted to calculate commonly seen bicyclics and store them in the database. But polycylics may have bicyclic components that are not registered in our database. Often these are highly strained (e.g., consecutive double bonds in a ring) such as the examples in Figure 4.10.

To maintain high accuracy, adding relevant bicyclic clusters into the database would be a long term solution. In this study we developed an additional layer (Figure

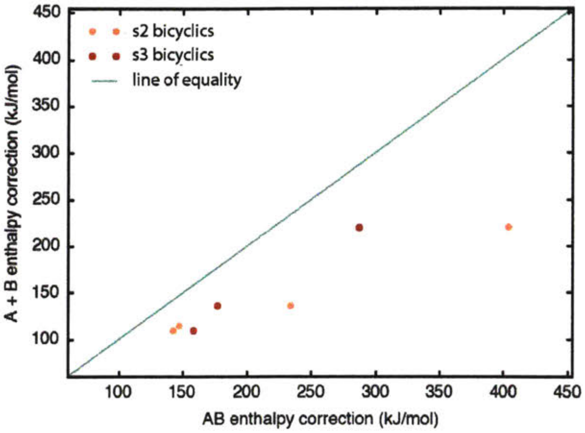


Figure 4.9: Ring corrections from bicyclics are very different from sum of corrections of individual single rings that make up the bicyclics. Note s2 bicyclics are those with 2-atom bridges, and s3 bicyclics are those with 3-atom bridges.

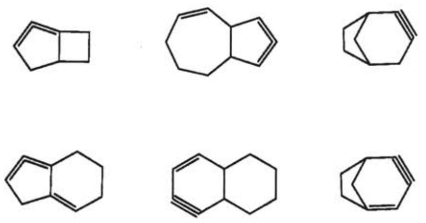


Figure 4.10: Bicyclic structures like these with high ring strain are usually not recorded in databases, but they can be formed as intermediates during reaction network exploration

4.11) in the original bicyclic decomposition method for cases where requested bicyclic components are not available or matched nodes are not similar enough to the target bicyclics.

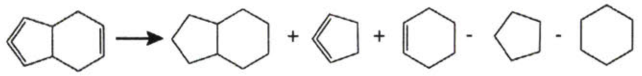


Figure 4.11: Bicyclic correction estimation scheme for bicyclics missing from the database

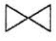
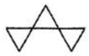
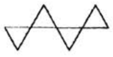
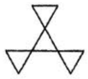
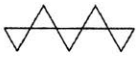
4.3.3 Model test

Using the same polycylic dataset, the bicyclic decomposition method significantly reduces prediction error for both small and large cyclics (Figure 4.6).

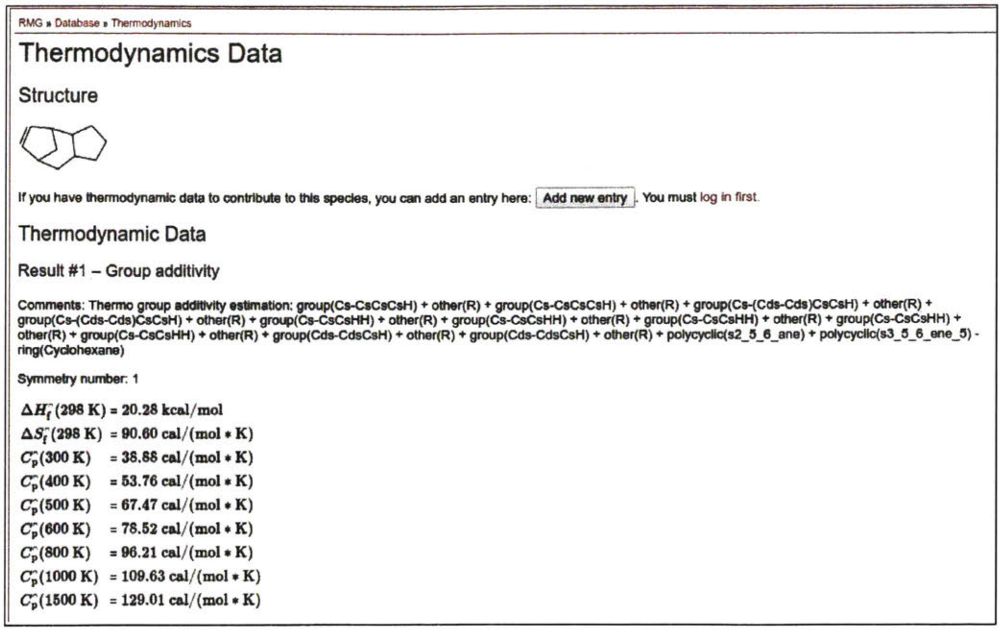
For small cyclics ($\leq$ 2-ring molecules), bicyclics decomposition automatically falls back to the similarity match method, which guarantees the prediction accuracy achieved by similarity match, see Table 4.1. For large polycyclics, bicyclic decomposition outperforms similarity match, bringing the error down to 4.9 kcal/mol and 9.8 kcal/mol for large linear cyclics and large fused cyclics, respectively.

Our algorithm was aslo tested against the data set by Osmont, et al. [11] who used B3LYP/6-31g(d,p) to calculate enthalpy of formation for propellanes. Our bicyclic decomposition algorithm, without running any further quantum chemistry calculations, was able to get enthalpies of dispiro[2.0.2.1]heptane, trispiro[2.0.2.0.2.0]nonane, trispiro[2.0.0.2.1.1]nonane and tetraspiro[2.0.0.0.2.1.1.1]undecane with DFT accuracy, as shown in Table 4.2.

Table 4.2: Experimental and calculated enthalpy of formation at 298K (kcal/mol) for spiropentane related polycyclic compounds

Structure	Experimental value	Osmont [11] DFT data	This work
	44.3	39.0	44.3
	72.4	67.7	69.0
	102.7	96.5	100.1
	101.2	96.5	97.1
	130.0	125.3	131.3

For users that are interested in using this method to estimate polycylic thermochemistry, a web application (`http://rmg.mit.edu/molecule_search`) is made available to allow users to input molecules (with elements of C, H, O) using species identifiers such as SMILES, InChI, CAS number or species name and to compute that molecule's thermochemistry. A screenshot illustrating the output from this web tool is shown in Figure 4.12.



RMG » Database » Thermodynamics

Thermodynamics Data

Structure

If you have thermodynamic data to contribute to this species, you can add an entry here: [Add new entry]. You must log in first.

Thermodynamic Data

Result #1 – Group additivity

Comments: Thermo group additivity estimation: group(Cs-CsCsCsH) + other(R) + group(Cs-CsCsCsH) + other(R) + group(Cs-(Cds-Cds)CsCsH) + other(R) + group(Cs-(Cds-Cds)CsCsH) + other(R) + group(Cs-CsCsHH) + other(R) + group(Cs-CsCsHH) + other(R) + group(Cs-CsCsHH) + other(R) + group(Cs-CsCsHH) + other(R) + group(Cs-CsCsHH) + other(R) + group(Cds-CdsCsH) + other(R) + group(Cds-CdsCsH) + other(R) + polycyclic(s2_5_6_ane) + polycyclic(s3_5_6_ene_5) - ring(Cyclohexane)

Symmetry number: 1

$\Delta H_f^\circ(298\ \mathrm{K}) = 20.28\ \mathrm{kcal/mol}$

$\Delta S_f^\circ(298\ \mathrm{K}) = 90.60\ \mathrm{cal/(mol * K)}$

$C_p^\circ(300\ \mathrm{K}) = 38.88\ \mathrm{cal/(mol * K)}$

$C_p^\circ(400\ \mathrm{K}) = 53.76\ \mathrm{cal/(mol * K)}$

$C_p^\circ(500\ \mathrm{K}) = 67.47\ \mathrm{cal/(mol * K)}$

$C_p^\circ(600\ \mathrm{K}) = 78.52\ \mathrm{cal/(mol * K)}$

$C_p^\circ(800\ \mathrm{K}) = 96.21\ \mathrm{cal/(mol * K)}$

$C_p^\circ(1000\ \mathrm{K}) = 109.63\ \mathrm{cal/(mol * K)}$

$C_p^\circ(1500\ \mathrm{K}) = 129.01\ \mathrm{cal/(mol * K)}$

Figure 4.12: Example polycyclic thermochemistry estimation in web application at rmg.mit.edu

4.4 Discussion

The new thermochemistry estimator based on group addivity, which combines similarity match and the bicyclic decomposition method, predicts polycyclic thermochemistry more accurately. This extension makes the group additivity method generalizable for polycyclics without requiring much pre-calculated data.

4.4.1 Large fused cyclics

For bicyclics and linear polycyclics, the typical error of 3 $\sim$ 5 kcal/mol is generally acceptable as a first approximation, especially when dealing with molecules with more than 10 carbons. The underlying reason that such a simple model performs well is that the decomposed bicyclic components act relatively independently; the thermodynamic contribution from the inter-bicyclic interaction is small compared with the ring strain contributions from the bicyclic itself.

For heavily fused cyclics, there are atoms shared by more than two rings. For instance, tricyclic A in Figure 4.13 has one such atom (atom 1). To account for all the fused parts (bond 1-2, 1-3 and 1-4), the bicyclic decompostion algorithm has to decompose tricyclic A into 3 bicyclics; bicyclic B is for contribution from bond 1-4, bicyclic C for 1-2 and bicyclic D for 1-3. In this case, the thermodynamic contribution from the inter-bicyclic interaction is more important than in the large linear cyclic

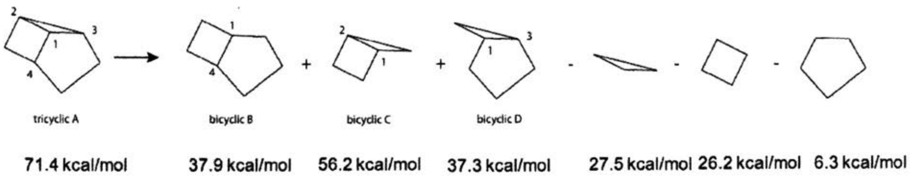


Figure 4.13: Bicyclic decompostion of an example heavily fused cyclic. The algorithm estimatates its ring correction for enthalpy of formation to be 71.4 kcal/mol using data from small rings, while direct quantum calculation gives 65 kcal/mol, leading to around 6 kcal/mol error

cases. That explains why the prediction error increases to 10 kcal/mol for those heavily fused cyclics.

4.4.2 Hetero-atom polycyclics

In many real applications, hetero-atoms such as O, S, and N are embedded in polycylics. There are usually fewer data available on heteropolycyclics than on polycyclic hydrocarbons. However if ring corrections depend more on ring structures than atom types, one can use the same database created in this chapter to estimate heteroatom polycyclic thermochemistry.

Predictions made in this way for oxygen-embedded polycyclics agree well with the quantum mechanically calculated values for over 18,000 oxygen-embedded polycyclics [4] (Table 4.3). Predictions for hetero-atom polycyclics achieve similar accuracy to hydrocarbon polycyclics.

Table 4.3: Mean absolute error (kcal/mol) of H_f(298 K) for each category of oxygen-embeded polycyclics

Method	small cyclics	large linear cyclics	large fused cyclics
Group Additivity Method	44	78	84
+ Similarity Match	5	34	40
+ Bicyclic Decomposition	5	6.6	10.6

4.4.3 Heat capacity and standard entropy predictions

Besides H_f(298 K), the methods proposed by this chapter also improve heat capacity prediction accuracy by a factor of $6 \sim 10$ for all three categories of polycyclics (Table 4.4). This can be crucial for chemical systems operated at temperatures other than $298K$. For standard entropy S(298 K) predictions (Table 4.5), we also observed a similar accuracy boost using the bicyclic decomposition method. The good agreement

Table 4.4: Mean absolute error (cal/mol/K) of C_p(298 K) for each category of polycyclics

Method	small cyclics	large linear cyclics	large fused cyclics
Group Additivity Method	6.1	10.0	10.5
+ Similarity Match	1.1	2.0	2.7
+ Bicyclic Decomposition	1.1	0.7	1.7

seems to suggest the contributions to entropy and heat capacity from ring strains are also additive (although we note the entropy prediction for large fused cyclics has large uncertainties).

Table 4.5: Mean absolute error (cal/mol/K) of S(298 K) for each category of polycyclics

Method	small cyclics	large linear cyclics	large fused cyclics
Group Additivity Method	44.5	93.3	103.5
+ Similarity Match	3.6	36.7	47.6
+ Bicyclic Decomposition	3.6	4.9	11.9

4.5 Conclusion

The similarity match algorithm combined with the bicyclic decomposition model can estimate unknown ring corrections and can be applied to various kinds of polycyclics. By assuming bicyclic ring strain contributions are independent and additive, the proposed method is both interpretable and effective (mean absolute error of H_f(298 K): 3 $\sim$ 5 kcal/mol) for bicyclics and linear polycyclics. Its accuracy starts dropping (mean absolute error of H_f(298 K): 10 kcal/mol) for large heavily fused cyclics, which motivates us to develop a novel thermocehmistry estimator discussed in Chapter 5. Besides formation enthalpy, the method also shows good performance in heat capacity and entropy predictions and is applied well to some heteroatomic polycyclics (tested on oxygen-embedded polycyclics).

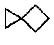
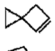
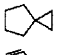
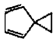
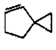
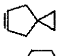
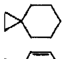
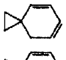
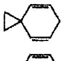
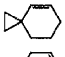
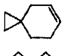
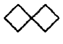
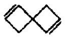
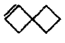
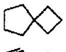
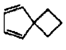
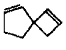
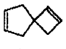
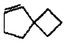
Overall, this method provides a quick and moderately accurate way of estimating themochemistry of large unknown polycyclics where quantum mechanical calculation may be significantly more expensive. We have implemented this new method in RMG and a web service is freely accessible via `http://rmg.mit.edu/molecule_search`.

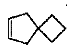
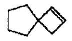
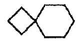
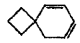
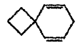
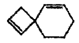
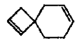
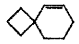
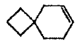
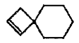
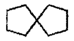
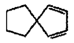
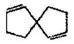
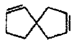
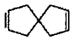
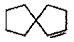
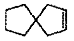
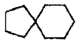
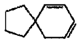
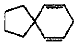
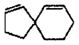
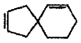
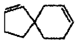
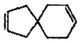
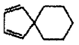
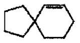
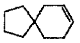
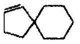
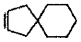
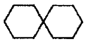
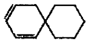
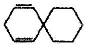
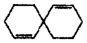
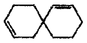
4.6 Appendix I: computed thermochemistry of bicyclics

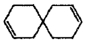
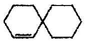
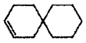
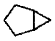
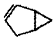
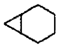
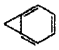
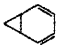
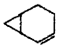
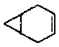
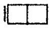
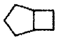
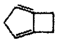
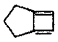
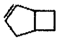
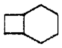
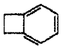
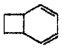
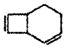
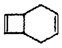
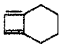
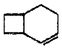
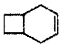
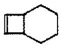
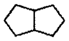
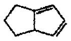
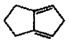
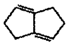
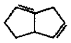
The 190 pre-calculated molecules are mostly bicyclics. Table 4.6 records the full collection. M06-2X/cc-pVTZ is employed with RRHO partition functions. Note the values listed here are thermochemistry for molecules.

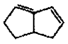
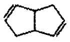
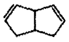
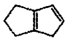
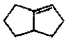
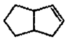
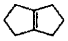
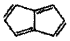
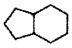
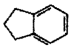
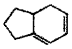
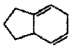
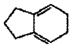
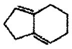
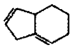
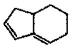
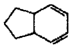
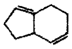
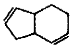
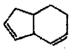
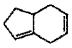
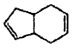
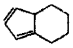
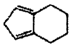
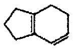
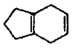
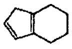
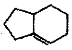
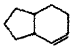
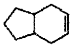
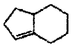
Ring corrections derived from these row values and Benson group values used for the derivation are recorded in RMG-database, which is hosted on Github. Specifically, ring corrections are stored at `https://github.com/ReactionMechanismGenerator/RMG-database/blob/master/input/thermo/groups/polycyclic.py` and Benson group values are at `https://github.com/ReactionMechanismGenerator/RMG-database/blob/master/input/thermo/groups/group.py`.

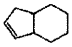
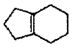
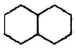
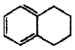
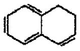
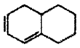
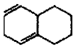
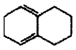
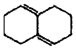
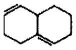
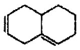
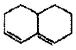
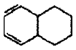
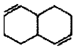
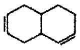
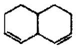
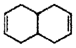
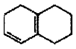
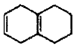
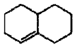
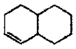
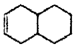
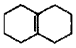
Table 4.6: Pre-calculated thermo-properties of 190 bycyclics

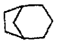
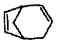
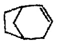
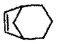
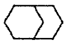
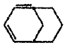
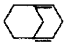
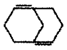
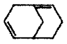
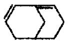
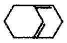
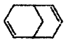
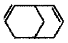
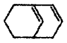
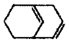
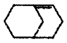
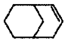
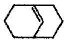
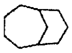
Structure	Label in RMG tree	$\Delta_f H$(298K) kcal/mol	S(298K) cal/mol/K	C_p(300K) cal/mol/K	C_p(1000K) cal/mol/K	C_p(1500K) cal/mol/K
	s1_3_3_ane	40.87	67.23	21.09	53.04	61.46
	s1_3_3_ene	95.61	67.64	20.20	45.94	52.32
	s1_3_4_ane	31.71	75.52	25.26	65.68	76.42
	s1_3_4_ene	64.25	73.35	23.59	58.31	67.12
	s1_3_5_ane	8.12	79.72	29.10	78.09	91.19
	s1_3_5_diene_1_3	55.75	74.45	25.61	63.37	72.54
	s1_3_5_ene_1	33.86	78.67	27.63	70.79	81.91
	s1_3_5_ene_2	33.86	78.68	27.63	70.79	81.91
	s1_3_6_ane	-3.92	82.94	33.38	90.63	106.03
	s1_3_6_diene_1_3	49.92	81.16	30.36	75.97	87.44
	s1_3_6_diene_1_4	48.56	79.91	30.29	75.88	87.40
	s1_3_6_ene_1	23.18	82.32	31.90	83.26	96.72
	s1_3_6_ene_2	23.18	82.32	31.90	83.26	96.72
	s1_4_4_ane	26.49	78.15	29.14	78.23	91.32
	s1_4_4_diene_1_5	91.71	75.80	26.19	63.58	72.80
	s1_4_4_ene_1	58.98	78.38	27.69	70.94	82.08
	s1_4_5_ane	3.81	85.56	33.58	90.71	106.14
	s1_4_5_diene_1_3	55.69	80.94	29.97	76.09	87.55
	s1_4_5_diene_1_6	62.88	81.79	30.22	76.05	87.58
	s1_4_5_diene_2_6	63.38	81.12	30.12	76.10	87.62
	s1_4_5_ene_1	30.23	82.76	31.71	83.42	96.87

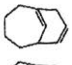
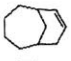
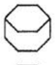
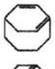
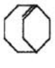
	s1_4_5_ene_2	30.53	82.26	31.52	83.42	96.88
	s1_4_5_ene_6	36.82	84.58	32.08	83.39	96.85
	s1_4_6_ane	-7.63	87.69	37.76	103.35	121.02
	s1_4_6_diene_1_3	47.30	85.37	34.63	88.63	102.25
	s1_4_6_diene_1_4	46.94	86.59	34.75	88.62	102.29
	s1_4_6_diene_1_7	53.01	85.77	34.70	88.62	102.32
	s1_4_6_diene_2_7	52.80	85.32	34.63	88.63	102.32
	s1_4_6_ene_1	20.39	86.92	36.24	95.95	111.69
	s1_4_6_ene_2	20.38	86.97	36.23	95.94	111.69
	s1_4_6_ene_7	25.39	86.30	36.09	95.90	111.69
	s1_5_5_ane	-17.99	88.86	37.57	103.18	120.97
	s1_5_5_diene_1_3	33.47	86.09	34.21	88.50	102.32
	s1_5_5_diene_1_6	35.22	86.50	34.33	88.56	102.38
	s1_5_5_diene_1_7	35.17	85.55	34.13	88.56	102.40
	s1_5_5_diene_2_7	206.39	88.46	37.12	87.89	100.95
	s1_5_5_ene_1	8.54	89.98	35.82	95.88	111.66
	s1_5_5_ene_2	8.10	87.37	35.88	95.91	111.69
	s1_5_6_ane	-28.08	92.09	41.62	115.67	135.76
	s1_5_6_diene_1_3	25.34	90.33	38.84	101.06	117.06
	s1_5_6_diene_1_4	26.21	92.85	38.81	101.01	117.07
	s1_5_6_diene_1_7	25.76	90.07	38.77	101.06	117.09
	s1_5_6_diene_1_8	25.68	89.64	38.71	101.13	117.12
	s1_5_6_diene_2_7	24.78	89.46	38.69	101.09	117.10
	s1_5_6_diene_2_8	25.41	89.34	38.56	101.09	117.11
	s1_5_6_diene_7_9	23.54	88.66	38.29	100.99	117.12
	s1_5_6_ene_1	-0.61	93.01	40.37	108.41	126.42
	s1_5_6_ene_2	-1.37	91.03	40.28	108.34	126.38
	s1_5_6_ene_7	-2.01	90.65	40.26	108.43	126.47
	s1_5_6_ene_8	-2.01	90.64	40.26	108.43	126.47
	s1_6_6_ane	-37.61	94.71	46.05	128.31	150.63
	s1_6_6_diene_1_3	15.68	93.57	43.27	113.71	131.96
	s1_6_6_diene_1_4	16.45	94.96	43.15	113.52	131.88
	s1_6_6_diene_1_7	17.41	93.09	43.21	113.55	131.90
	s1_6_6_diene_1_8	17.05	94.02	43.21	113.61	131.92

	s1_6_6_diene_2_8	15.41	91.61	43.12	113.65	131.96
	s1_6_6_ene_1	-9.99	94.70	44.61	120.89	141.21
	s1_6_6_ene_2	-10.87	93.89	44.50	120.90	141.24
	s2_3_3_ane	49.72	61.87	15.50	40.24	46.60
	s2_3_3_ene	113.84	62.54	14.33	32.88	37.40
	s2_3_4_ane	37.61	66.26	19.20	52.73	61.40
	s2_3_4_ene_1	79.26	66.23	17.88	45.58	52.21
	s2_3_5_ane	13.73	73.13	23.58	65.33	76.29
	s2_3_5_ene_1	38.45	70.48	21.72	57.99	66.97
	s2_3_6_ane	8.53	79.47	28.42	77.88	91.12
	s2_3_6_ben	88.52	73.33	22.72	55.40	63.14
	s2_3_6_diene_1_3	50.99	74.00	24.63	63.08	72.45
	s2_3_6_ene_1	29.65	75.58	26.35	70.42	81.75
	s2_3_6_ene_2	29.87	75.52	26.60	70.46	81.82
	s2_4_4_ane	34.40	72.46	23.74	65.51	76.44
	s2_4_4_ene_1	30.00	70.98	22.48	57.70	66.89
	s2_4_5_ane	9.44	76.07	27.63	77.91	91.20
	s2_4_5_diene_0_3	73.79	75.63	25.29	63.34	72.69
	s2_4_5_diene_4_6	119.13	74.98	24.40	61.45	70.74
	s2_4_5_ene_1	34.05	75.66	25.88	70.53	81.89
	s2_4_6_ane	0.56	81.37	32.09	90.41	106.03
	s2_4_6_ben	50.11	77.61	26.83	68.03	78.02
	s2_4_6_diene_1_3	51.90	79.35	28.92	75.70	87.41
	s2_4_6_diene_1_6	58.62	79.41	29.05	75.73	87.46
	s2_4_6_diene_2_6	59.96	80.39	29.41	75.83	87.53
	s2_4_6_diene_5_7	97.66	81.98	30.89	76.06	87.51
	s2_4_6_ene_1	25.59	80.44	30.50	83.03	96.71
	s2_4_6_ene_2	29.14	82.16	30.60	83.04	96.74
	s2_4_6_ene_6	47.27	79.70	30.39	83.00	96.72
	s2_5_5_ane	-14.47	79.90	31.54	90.31	105.96
	s2_5_5_diene_0_2	40.26	79.67	28.81	75.72	87.45
	s2_5_5_diene_0_3	36.35	79.52	29.22	75.74	87.45
	s2_5_5_diene_0_4	37.29	81.85	29.47	75.53	87.37
	s2_5_5_diene_0_5	42.58	79.04	28.72	75.64	87.43

	s2_5_5_diene_0_6	39.38	78.91	28.62	75.62	87.42
	s2_5_5_diene_1_5	37.56	79.21	28.38	75.64	87.41
	s2_5_5_diene_1_6	38.78	79.53	28.49	75.70	87.45
	s2_5_5_diene_m_2	37.66	80.30	29.23	75.59	87.41
	s2_5_5_ene_0	14.24	81.61	30.65	82.99	96.73
	s2_5_5_ene_1	11.24	81.29	30.15	83.04	96.74
	s2_5_5_ene_m	12.38	81.49	31.07	82.79	96.67
	s2_5_5_tetraene_0_2_4_6	96.36	76.11	25.38	60.81	68.72
	s2_5_6_ane	-23.45	87.64	36.72	102.94	120.89
	s2_5_6_ben	18.42	81.95	30.95	80.50	92.82
	s2_5_6_diene_0_2	27.17	84.14	33.47	88.20	102.27
	s2_5_6_diene_0_3	29.41	85.93	33.67	88.16	102.25
	s2_5_6_diene_0_4	26.86	84.97	33.95	88.17	102.25
	s2_5_6_diene_0_5	23.69	85.82	33.83	88.07	102.20
	s2_5_6_diene_0_6	29.83	84.03	33.39	88.17	102.27
	s2_5_6_diene_0_7	26.08	84.00	33.33	88.15	102.25
	s2_5_6_diene_1_3	29.95	86.71	33.21	88.15	102.21
	s2_5_6_diene_1_5	32.37	85.27	33.77	88.25	102.30
	s2_5_6_diene_1_6	30.28	83.85	33.17	88.21	102.27
	s2_5_6_diene_1_7	31.10	83.86	33.11	88.20	102.26
	s2_5_6_diene_2_5	29.36	86.20	33.76	88.18	102.28
	s2_5_6_diene_2_6	32.24	83.19	33.10	88.24	102.31
	s2_5_6_diene_5_7	25.30	83.49	33.19	88.22	102.26
	s2_5_6_diene_5_8	24.67	83.45	33.84	88.28	102.31
	s2_5_6_diene_m_1	26.13	85.93	34.06	88.15	102.21
	s2_5_6_diene_m_2	25.06	86.21	33.94	88.03	102.18
	s2_5_6_diene_m_7	23.24	84.68	33.83	88.24	102.31
	s2_5_6_ene_0	1.75	86.31	35.29	95.54	111.57
	s2_5_6_ene_1	4.36	85.92	34.83	95.49	111.52
	s2_5_6_ene_2	6.22	85.10	34.74	95.54	111.53
	s2_5_6_ene_5	6.01	86.68	35.45	95.54	111.55

	s2_5_6_ene_6	6.61	85.77	34.95	95.59	111.58
	s2_5_6_ene_m	-1.05	87.16	35.58	95.38	111.56
	s2_6_6_ane	-35.51	89.21	41.18	115.60	135.82
	s2_6_6_ben	10.52	87.14	35.56	93.04	107.67
	s2_6_6_ben_ene_1	34.45	85.01	33.78	85.66	98.34
	s2_6_6_diene_0_2	21.86	89.50	38.24	100.73	117.00
	s2_6_6_diene_0_3	17.87	89.18	37.97	100.70	117.10
	s2_6_6_diene_0_4	18.90	88.14	38.38	100.72	117.02
	s2_6_6_diene_0_5	14.26	87.86	38.14	100.62	116.97
	s2_6_6_diene_0_6	72.90	88.59	38.55	101.21	117.21
	s2_6_6_diene_0_7	20.57	89.64	38.16	100.69	117.02
	s2_6_6_diene_0_8	17.87	88.86	37.95	100.67	117.07
	s2_6_6_diene_1_3	19.90	86.51	37.77	100.83	117.16
	s2_6_6_diene_1_6	20.92	87.02	37.80	100.73	117.09
	s2_6_6_diene_1_7	129.14	86.94	38.44	101.49	117.44
	s2_6_6_diene_1_8	21.64	88.10	37.74	100.76	117.11
	s2_6_6_diene_2_7	20.67	87.98	37.79	100.79	117.13
	s2_6_6_diene_m_1	16.93	90.17	38.45	100.65	117.03
	s2_6_6_diene_m_2	16.49	89.51	38.44	100.59	117.05
	s2_6_6_ene_0	-8.08	90.03	39.58	108.04	126.38
	s2_6_6_ene_1	-0.62	89.55	39.37	108.12	126.39
	s2_6_6_ene_2	-7.55	88.77	39.71	108.20	126.48
	s2_6_6_ene_m	-9.82	88.32	39.87	107.91	126.32
	s3_4_4_ane	49.94	62.34	17.96	52.71	61.48
	s3_4_4_diene_0_2	152.09	63.37	16.06	38.12	42.95
	s3_4_4_ene_0	112.10	66.61	18.17	45.60	52.28
	s3_4_5_ane	18.81	68.86	22.23	65.27	76.30
	s3_4_5_diene_0_2	118.33	67.48	20.06	50.66	57.77
	s3_4_5_diene_0_3	127.78	73.35	22.67	51.04	57.85
	s3_4_5_diene_1_3	125.49	69.87	20.62	50.94	57.91
	s3_4_5_diene_3_4	145.71	69.18	21.10	50.90	57.89
	s3_4_5_ene_0	130.15	70.23	22.32	58.53	67.19

	s3_4_5_ene_1	60.44	67.42	20.81	58.08	67.05
	s3_4_5_ene_3	99.29	73.46	23.12	58.44	67.27
	s3_4_6_ane	9.37	75.59	26.99	77.75	91.13
	s3_4_6_diene_0_2	115.97	72.72	24.41	63.48	72.74
	s3_4_6_diene_0_3	82.15	73.20	24.32	63.26	72.47
	s3_4_6_diene_0_4	93.02	77.39	26.87	63.54	72.57
	s3_4_6_diene_1_4	110.24	72.61	24.23	63.42	72.72
	s3_4_6_diene_1_5	116.53	73.06	24.42	63.41	72.74
	s3_4_6_ene_0	80.33	73.56	25.91	70.88	81.86
	s3_4_6_ene_1	35.62	73.59	25.21	70.43	81.83
	s3_4_6_ene_4	80.56	74.66	26.02	70.67	81.96
	s3_5_5_ane	-6.59	73.65	26.40	77.69	91.07
	s3_5_5_diene_1_4	64.50	70.04	23.44	63.30	72.57
	s3_5_5_ene_1	26.24	72.98	24.81	70.45	81.79
	s3_5_6_ane	-17.27	79.25	31.03	90.24	105.94
	s3_5_6_diene_1_5	42.66	76.43	28.15	75.81	87.27
	s3_5_6_ene_1	10.51	78.43	29.42	82.84	96.61
	s3_5_6_ene_5	14.24	77.61	29.35	82.89	96.62
	s3_6_6_ane	-22.57	82.55	35.34	102.51	120.64
	s3_6_6_diene_0_2	49.78	81.32	32.60	88.17	102.22
	s3_6_6_diene_0_3	65.54	80.99	32.94	88.37	102.27
	s3_6_6_diene_0_4	60.78	80.07	33.13	88.35	102.26
	s3_6_6_diene_0_5	43.85	81.27	32.55	88.19	102.22
	s3_6_6_diene_0_6	110.77	80.51	32.82	88.32	102.20
	s3_6_6_diene_0_m	125.35	81.56	33.56	88.78	102.52
	s3_6_6_diene_1_5	26.80	80.01	32.33	87.99	102.15
	s3_6_6_diene_1_6	30.44	83.28	32.60	87.98	102.13
	s3_6_6_diene_1_8	77.43	81.93	33.06	88.36	102.32
	s3_6_6_diene_1_m	78.57	81.61	33.00	88.42	102.36
	s3_6_6_ene_0	54.35	82.41	34.43	95.71	111.60
	s3_6_6_ene_1	0.10	82.73	34.04	95.44	111.51
	s3_6_6_ene_4	95.55	86.73	37.12	96.46	111.86
	s3_6_7_ane	-23.91	88.98	40.63	115.40	135.68

	s3_6_7_diene_6_9-0	118.04	86.85	38.44	101.29	117.32
	s3_6_7_ene_6	28.52	86.49	38.49	107.94	126.24
	s4_6_8_ane	26.56	91.69	42.02	115.68	135.62
	s4_6_8_diene_7_9	41.54	86.07	37.66	100.86	117.12
	s4_6_8_ene_7	15.63	88.48	39.52	108.21	126.43

4.7 Appendix II: Similarity match example

Here is an example where a big cyclic molecule (see Figure 4.15) that does not have exact match in tree (see Figure 4.14). The similarity match algorithm will still match an existing node in the tree for the molecule.

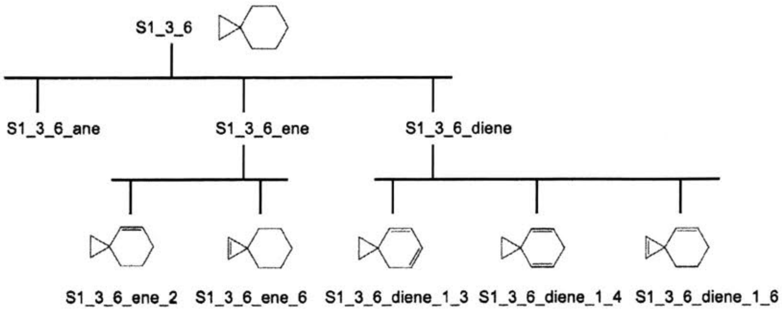


Figure 4.14: Example sub-tree that organizes polycyclic ring corrections

It uses sub-graph isomorphism check to find which nodes are contained by the big molecule and selects the correction of the first matched node. In this case, both s1_3_6_ene_2 and s1_3_6_diene_1_4 are contained in the example big molecule, but the correction of first match s1_3_6_ene_2 will be applied. This explaines necessity of bicyclic decomposition algorithm.

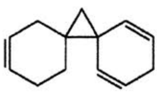


Figure 4.15: Example tricyclic that does not have exact match in the tree

4.8 References

[1] S. W. Benson, F. R. Cruickshank, D. M. Golden, G. R. Haugen, H. E. ONeal, A. S. Rodgers, R. Shaw, and R. Walsh. "Additivity Rules for the Estimation of Thermochemical Properties." *Chem. Rev.* 69, 1969, pp. 279–324. URL: http://kinetics.nist.gov/kinetics/Detail?id=1969BEN/CRU279-324:0.

[2] L. Constantinou and R. Gani. "New group contribution method for estimating properties of pure compounds." en. *AIChE Journal* 40 (10), Oct. 1994, pp. 1697–1710. ISSN: 1547-5905. DOI: 10.1002/aic.690401011. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/aic.690401011.

[3] B. T. Fan, A. Panaye, J. P. Doucet, and A. Barbu. "Ring perception. A new algorithm for directly finding the smallest set of smallest rings from a connection table." *Journal of Chemical Information and Computer Sciences* 33 (5), Sept. 1993, pp. 657–662. ISSN: 0095-2338. DOI: 10.1021/ci00015a002. URL: http://dx.doi.org/10.1021/ci00015a002.

[4] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. v. Lilienfeld. "Quantum chemistry structures and properties of 134 kilo molecules." en. *Scientific Data* 1, Aug. 2014, p. 140022. ISSN: 2052-4463. DOI: 10.1038/sdata.2014.22. URL: http://www.nature.com/articles/sdata201422.

[5] G. Landrum et al. *RDKit: Open-source cheminformatics.* URL: http://www.rdkit.org.

[6] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Rob, J. R. Cheeseman, et al. *Gaussian 09 Revision C.01.* Gaussian Inc. Wallingford CT 2016.

[7] Y. Zhao and D. G. Truhlar. "The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: two new functionals and systematic testing of four M06-class functionals and 12 other functionals." *Theoretical Chemistry Accounts* 120 (1), May 1, 2008, pp. 215–241. ISSN: 1432-881X, 1432-2234. DOI: 10.1007/s00214-007-0310-x. URL: https://link.springer.com/article/10.1007/s00214-007-0310-x.

[8] C. W. Gao, J. W. Allen, W. H. Green, and R. H. West. "Reaction Mechanism Generator: Automatic construction of chemical kinetic mechanisms." *Computer Physics Communications* 203, June 2016, pp. 212–225. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2016.02.013.

[9] M. Rupp, A. Tkatchenko, K.-R. Muller, and O. A. von Lilienfeld. "Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning." *Physical Review Letters* 108 (5), Jan. 2012, p. 058301. DOI: 10.1103/PhysRevLett.108.058301. URL: http://link.aps.org/doi/10.1103/PhysRevLett.108.058301.

[10] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld. “Big Data Meets Quantum Chemistry Approximations: The Machine Learning Approach.” *Journal of Chemical Theory and Computation* 11 (5), May 2015, pp. 2087–2096. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.5b00099. URL: http://dx.doi.org/10.1021/acs.jctc.5b00099.

[11] A. Osmont, L. Catoire, and I. Gökalp. “Physicochemical Properties and Thermochemistry of Propellanes.” *Energy & Fuels* 22 (4), July 2008, pp. 2241–2257. ISSN: 0887-0624. DOI: 10.1021/ef8000423. URL: http://dx.doi.org/10.1021/ef8000423.

5

Adaptive Thermochemistry Estimator using Molecular Convolutional Neural Networks

Heuristic method is effective in estimating thermochemistry of large polycyclics from small polycyclic data, as shown in Chapter 4. In H_f(298 K) prediction, it achieves moderately accurate results (MAE$\sim$ 5 kcal/mol from Table 4.1 and Table 4.3) for small cyclics and large linear cyclics. As polycylics become more fused and complicated, its accuracy starts to drop (MAE$\sim$ 10 kcal/mol with large fused polycyclic). It is because heuristic method assumes bicyclic components act independently without accounting for inter-bicyclic interaction, as described in Section 4.4.1. To account for that, tricyclic decomposition (similar to bicyclic decomposition) can be designed to help capture the interaction. It requires precalculation of possible kinds of tricyclics, of which the total number is much larger than that of bicyclics.

Instead of creating more heuristics, this chapter focuses on a machine learning approach to further improve thermochemistry estimator. A general machine learning framework for molecular property prediction consists of two steps: featurization and regression (Figure 5.1). Featurization converts a molecule (e.g., node-edge graphs in 2D representation) to a feature vector with fixed length, and the regression maps the vector to property (scalar).

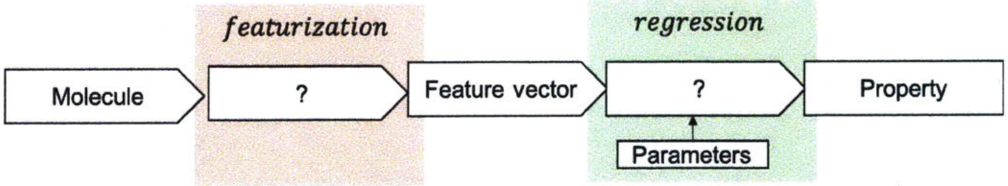


Figure 5.1: A general machine learning framework for molecular property prediction

5.1 Previous work revisited

The thermochemistry estimators previously developed for RMG can also fit into this framework. Group additivity method (see Figure 5.2) featurizes a molecule by counting pre-defined groups present in the molecule; the size of pre-defined group list defines the final feature vector size. Its regression part is linear regression model with coefficients being group contributions.

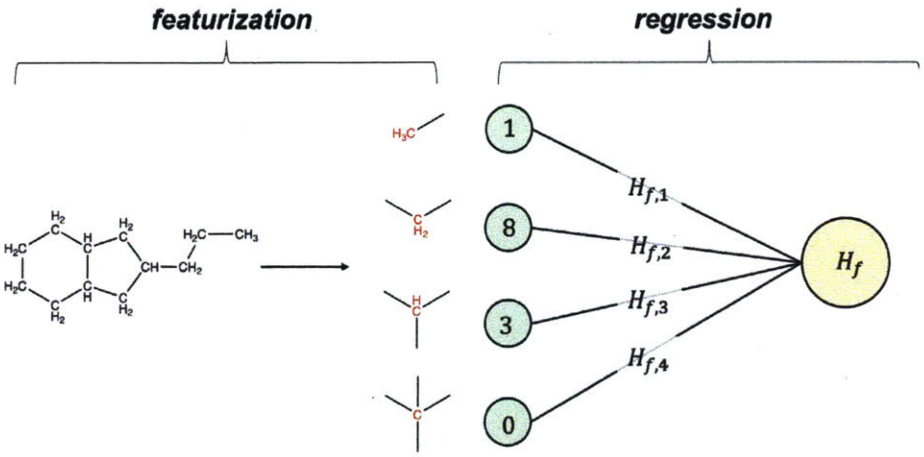


Figure 5.2: Group additivity method with featurization and regression

Occasionally representation collision (Figure 5.3) takes place where multiple molecules are mapped to a same feature vector. It's because pre-defined group list isn't comprehensive enough; usually more specific groups with larger defining neighborhood can be added to the list to resolve collision, as shown in Figure 5.4. The widely used Extended-Connectivity Fingerprint (ECFP) also originates from the similar idea [1]; its framework allows users to choose radius of a center atom's defining neighborhood.

The idea of adding correction for ring strain can be regarded as addition of ring groups to the list (Figure 5.5). Since there's an infinite number of ring groups, practically only those that have appeared in the training data are added. As expected such estimator makes much less accurate predictions for molecules with ring groups outside training data.

Heuristic method reveals the fact that the ring strain of a large group (e.g., tricyclic groups) can be expressed by those of small ones (e.g., monocyclic and bicyclic groups) and utilizes it to reduce the infinite feature space dimension to approximately the total number of possible monocyclic and bicyclic groups (see Figure 5.6).

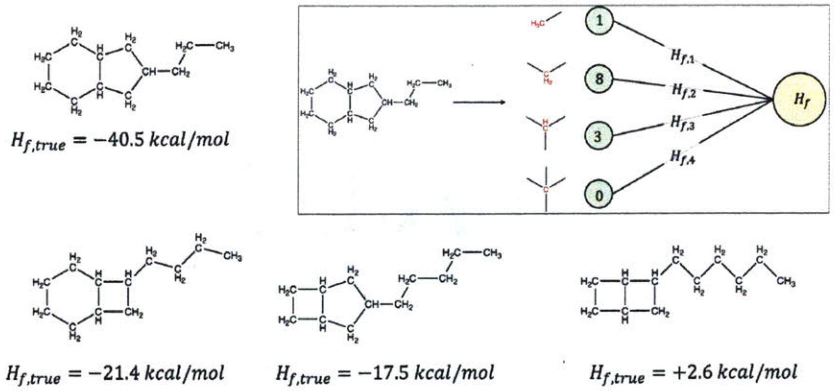

$H_{f,true}$ = −40.5 kcal/mol
1
8
3
0
$H_{f,1}$
$H_{f,2}$
$H_{f,3}$
$H_{f,4}$
H_f
$H_{f,true}$ = −21.4 kcal/mol
$H_{f,true}$ = −17.5 kcal/mol
$H_{f,true}$ = +2.6 kcal/mol

Figure 5.3: Representation collision: multiple molecules are mapped to a same feature vector

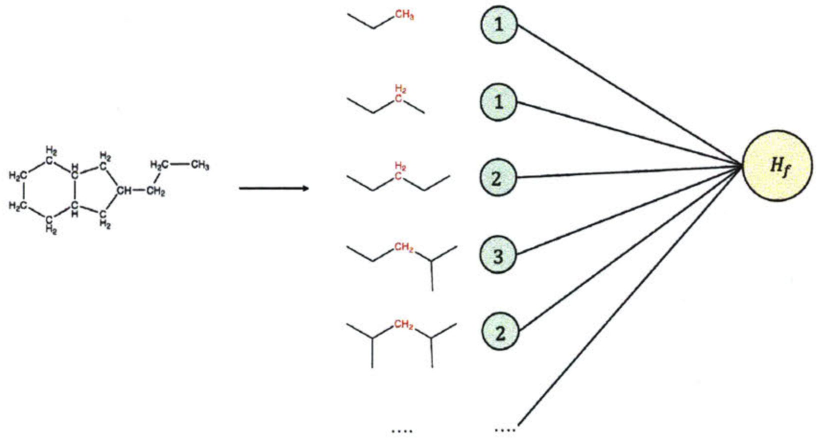


Figure 5.4: More specific groups with larger defining neighborhood compared with Figure 5.2

5.2 New estimator highlights

The previous estimators share a common part: human-designed featurization. When encountering a new molecule domain with unsatisfactory estimation accuracy, the estimators need new chemical insights from a human to modify featurization (e.g., adding more specific groups). When the insights are not available, estimators experience poor predictive performance and researchers tend to include as many features as possible.

In this study, we created a new estimator with learnable featurization to reduce the requirement of human chemical expertise. The learnable featurization, enabled by molecular convolutional neural networks (Section 5.3), achieves better performance than all the previous methods on thermochemistry predictions for polycyclics (Sec-

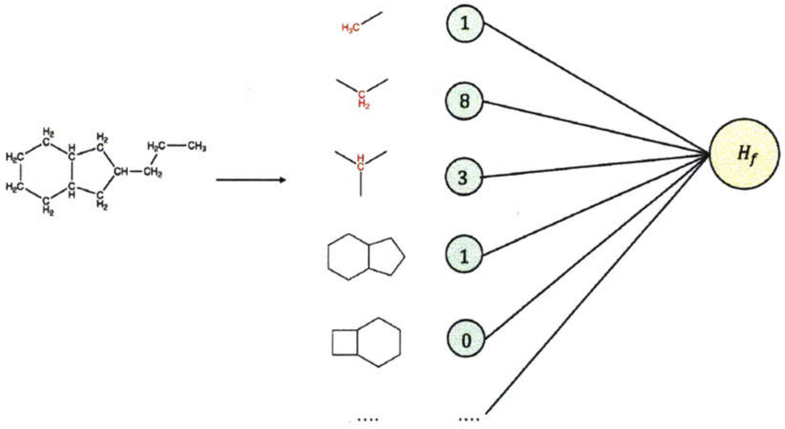


Figure 5.5: Ring correction method expands the pre-defined group list with infinite number of ring groups. Practically it only add those covered by training data

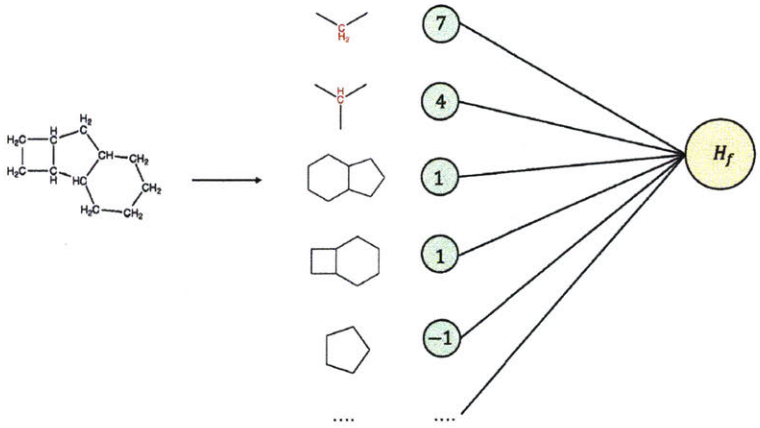


Figure 5.6: Heuristic method utilizes the ring strain dependence between large ring groups and small ones and reduces feature space dimension by only pre-defining monocyclic and bicyclic groups

tion 5.5); in particular, this estimator was able to gain higher accuracy for large fused polycyclics than heuristic method without needing any human insights. Through a self-evolving pipeline (Section 5.6), the estimator shows good extensibility; its prediction capability has expanded in three dimensions as below.

- molecule shape: from cyclics to non-cyclics
- heteroatom: from C,H,O-based to nitrogen-containing molecules
- prediction task: from formation enthalpy to entropy and heat capacity

5.3 Molecular convolutional neural networks

To make learnable featurization for molecules, Aspuru-Guzik, et al. designed convolutional neural networks for molecular graphs and got promising results for solubility, drug efficacy and photovoltaic efficiency applications [2]. Coley, et al. further applied it to predictions on melting point, toxicity, etc [3]. This is the first time that molecular convolutional neural network (thereafter MCNN) is applied to thermochemistry prediction (architecture in Figure 5.7).

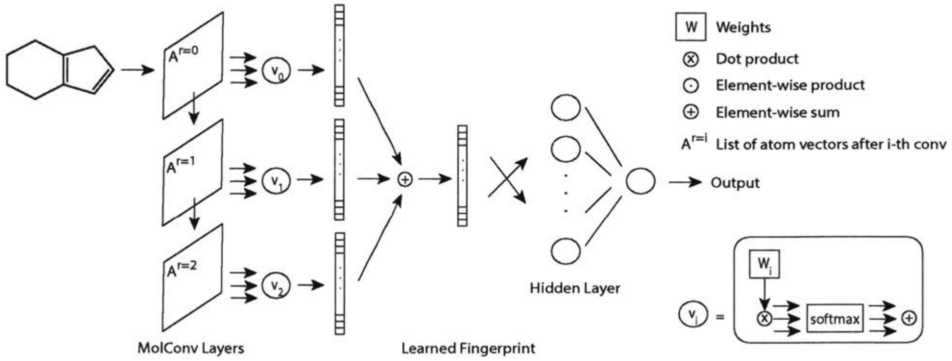


Figure 5.7: New thermochemistry estimator parameterizes featurization module via MCNN and uses fully-connected neural network with one hidden layer as regression module

Each molecule fed to MCNN is represented by three inputs: atom fingerprint matrix (denoted by A), bond fingerprint tensor (denoted by B), connectivity matrix (denoted by C). A has dimensionality of $n_a \times f_a$, B $n_a \times n_a \times f_b$, and C $n_a \times n_a$, where n_a is the number of atoms, f_a the size of atom fingerprint and f_b the size of bond fingerprint. The output of MCNN is molecular fingerprint vector with size of f_m.

5.3.1 atom fingerprint matrix A

For a given molecule, A stores the information at atom level; each atom has a fingerprint vector therefore A has size of $n_a \times f_a$. Atom fingerprint includes basic atomic information such as nuclear charge, number of hydrogens attached, appearance in n-member ring, etc.

5.3.2 bond fingerprint matrix B

For a given molecule, B stores the information at bond level; each bond has a fingerprint vector therefore B has size of $n_a \times n_a \times f_b$. Bond fingerprint includes basic information such as bond order, whether is in a ring, etc.

5.3.3 connectivity matrix C

For a given molecule, each possible pair of atoms has an entry in C; 1 indicates there's a bond between the pair, zero otherwise. C has size of $n_a \times n_a$. For algorithm implementation purpose, we set all diagonal entries as 1.

5.3.4 molecular convolution

In molecular convolution, we loop over each atom in the molecule and add its direct neighbors' atom fingerprints to its own fingerprint with certain weights, which gives a new convoluted atom fingerprint for each atom. Consecutive convolution is made possible by the iterative algorithm below and it captures neighborhood information with increasing radius r:

$$A^{r=0} = A$$

$$A^{r=m+1} = \tanh\Big([C \cdot A^{r=m}, B_{aggr}] \cdot W + b\Big)_{n_a \times f_a}, \quad m = 1, ..., R$$

where B_{aggr} is a matrix $(n_a \times f_b)$ by summing tensor B along the second dimension, namely $B_{aggr,ij} = \sum_k B_{ikj}$, R is the maximum radius being considered and W and b are weights and offsets to be trained.

Each atom fingerprint matrix has its own contribution to the final molecule fingerprint:

$$\text{fingerprint} = \begin{pmatrix} 1 \\ 1 \\ \dots \\ 1 \end{pmatrix}_{1 \times n_a} \cdot \Big(\sum_{m=0}^{R} \text{softmax}\big(A^{r=m} \cdot W' + b'\big) \Big)_{n_a \times f_m}$$

where W and b are another set of weights and offsets to be trained.

5.4 Datasets

In order to train and test the new estimators, two datasets were created from an earlier paper by Ramakrishnan [4]; one named `polycyclic_2954_table` has 2954 cyclic hydrocarbons and the other named `cyclic_O_only_table` has 25620 cyclic oxygenates. Three thermochemistry properties are included in the datasets: formation enthalpy at 298 K (unit: kcal/mol), standard entropy at 298 K (unit: cal/mol/K) and constant-pressure heat capacity at 300 K (unit: cal/mol/K). Both datasets are currently hosted by a `MongoDB` instance on RMG server (hostname: `rmg.mit.edu`) and support free accessibility.

We held out 20% of the datasets as test sets (`polycyclic_2954_table_test` and `cyclic_O_only_table_test`). The remaining 80% (`polycyclic_2954_table_train` and `cyclic_O_only_table_train`) are used in 5-fold cross-validation; 4 folds for training the neural networks and 1 fold for validation. Early stopping technique (using 10% of 4 folds training data) is applied to avoid ovefitting for each round of cross-validation.

5.5 Results and Discussion

The MCNN-based thermochemistry estimator is implemented via Keras [5]. Source code is available on Github: `https://github.com/KEHANG/RMG-Py/tree/cnn_framework_concise2`.

In order to conduct comparison study, two baseline models are established:

1) RMG's current thermochemistry estimator based on heuristic method, and

2) ECFP model using Extended-Connectivity Fingerprint (thereafter ECFP) operation as featurization and one hidden layer neural network as regression module (architecture shown in Figure 5.8).

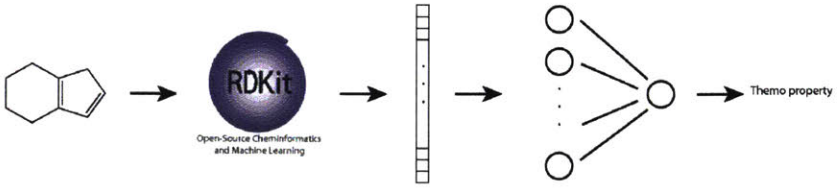


Figure 5.8: Architecture of second baseline model using ECFP and fully-connected neural network as regression module

5.5.1 Performance

Similar to the observation in Chapter 4, the heuristic method gives moderately accurate predictions with overall MAE $\sim$ 6 kcal/mol. Its accuracy drops significantly as the cyclic structural complexity increases; for large fused polycyclics, the MAE reaches beyond 10 kcal/mol (see Table 5.1).

Using off-the-shelf featurization, ECFP model performs worst among the three with overall MAE $\sim$ 10 kcal/mol. Additional study shows more complex non-linear regression module (e.g., from one hidden layer to multiple layers) doesn't futher improve prediction performace, suggesting ECFP isn't sufficiently effective in extracting features essential for formation enthalpy prediction.

Table 5.1: Mean absolute error (kcal/mol) of $H_f(298\text{ K})$ for cyclic hydrocarbons and oxygenates. Test Datasets are `polycyclic_2954_table_test` and `cyclic_O_only_table_test`

Method	small cyclics	large linear cyclics	large fused cyclics	overall
Heuristic Method	5.0	6.7	10.6	6.2
2048-bit ECFP	10.0	9.8	11.7	10.3
MCNN	1.5	2.0	2.5	1.6

MCNN-based estimator remarkably outperforms the other two, achieving MAE of 1.6 kcal/mol overall. Without asking for human insights, it reduces MAE for large fused polycyclics from $\sim$ 10 kcal/mol to 2.5 kcal/mol (Table 5.1 and Figure 5.10). Its effectiveness is probably due to the fact that the featurization is optimized together with regression; compared with precoded sub-structure features used by ECFP, MCNN may be able to find higher-level features that are more direct and essential to formation enthalpy prediction. This can be partially supported by the observation that MCNN fingerprint with around 200 entries reaches performance plateau (Figure 5.9), but still has better performance than ECFP with 2048 entries.

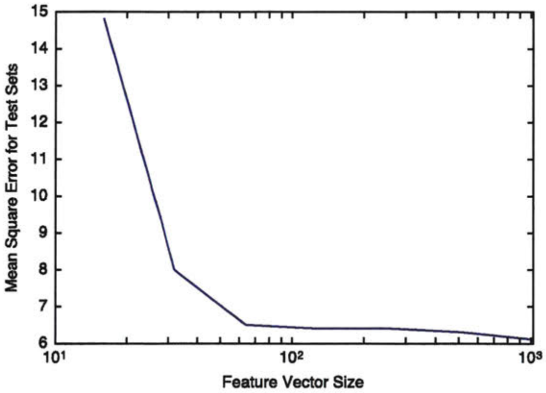


Figure 5.9: Effect of MCNN fingerprint length on test error (y axis: mean square error, unit: kcal/mol). Test Datasets are `polycyclic_2954_table_test` and `cyclic_O_only_table_test`

5.5.2 Model interpretaion

We carried out examination of the learned fingerprints to qualitatively evaluate convergence quality and interpret the embedded meaning. This provides us with several interesting observations.

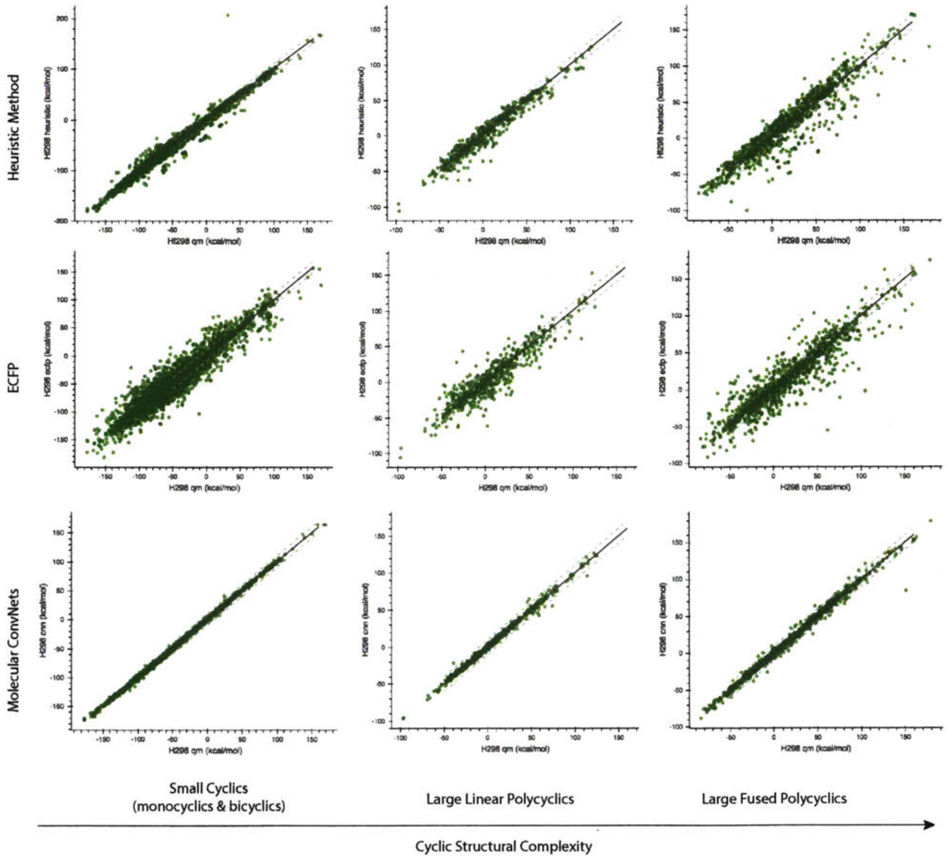


Figure 5.10: Predictions of enthalpy of formation (y axis) vs. quantum mechanics values (x axis, using DFT method B3LYP/6-31G(2df,p)) with various models and cyclic types, unit: kcal/mol, dataset: `polycyclic_2954_table_test` and `cyclic_0_only_table_test`

5.5.2.1 Similar molecules have similar fingerprints

In Figure 5.11, we show the first three cyclic hydrocarbons have fingerprints sharing major high peaks (e.g., the three entries in the beginning section of the fingerprints). Those peaks are not shared by the linear molecule (last molecule) with same number of carbons.

5.5.2.2 Simple algebraic property is preserved

As inspired by a famous interpretation example in word embedding [6] that vector(King) - vector(Queen) results in a vector that is very close to the vector difference produced by vector(Man) - vector(Woman), we demonstrates similar behavior

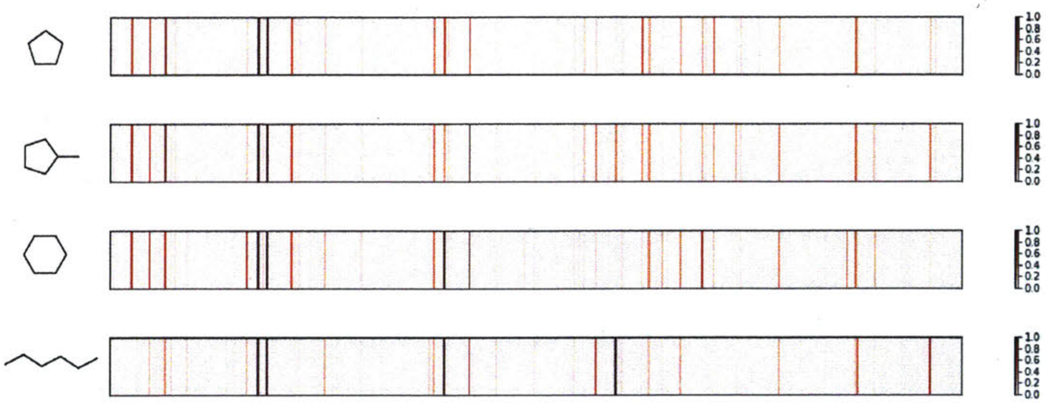


Figure 5.11: Fingerprints generated by MCNN for 4 example molecules

in MCNN featurization (Figure 5.12).

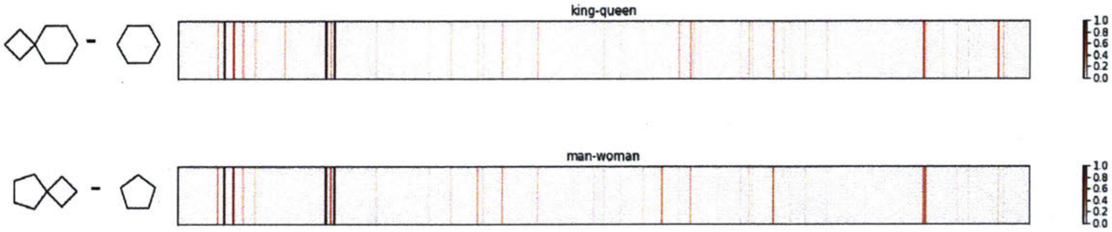


Figure 5.12: Simple arithmatic check for molecule fingerprints

Graphically, the differences from the two pairs of molecules are similar; for each pair, the first molecule differs from the second by a 4-member ring. In fingerprint form, we do see the vector differences are learned to be similar.

5.5.2.3 Prediction on internal molecule stability

With a model interpretation technique used by Coley et al. [3], we are able to learn what part of molecule MCNN model predicts to be stable or unstable in enthalpy formation task. Figure 5.13 shows MCNN model is able to identify carbons in strained configurations (colored red, e.g., in a 3-member ring, or in the fused part) as the unstable parts of the molecules and carbons in free configurations (colored blue, e.g., in 5-member ring or in side-chain) as the stable parts.

5.5.3 Uncertainty

It's extremely difficult or even impossible to prepare a training dataset sampling molecules uniformly from entire molecule space. In general, estimators usually make accurate predictions for molecules similar to those in the training set, but are less

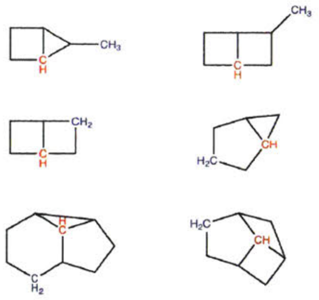


Figure 5.13: MCNN model suggests red atoms are the most unstable parts of molecules while blue atoms the most stable

reliable for molecules that are very different. Thus, a single reported performance metric is not sufficient to indicate prediction error in real applications since a thermochemistry estimator, once trained, allows any input molecules. It is always desired to have molecule-specific uncertainty along with prediction.

The neural network literature presents a large amount of work on uncertainty estimation [7, 8]. In this thesis, we provide two approaches: a intuitive distance-based method[3] and a more rigorous ensemble-based [9] method.

5.5.3.1 Distance-based uncertainty estimation

Intuitively, prediction uncertainty is associated with distance between query molecule and training molecules in feature space. We create a working curve (Figure 5.14) which reveals that relationship quantitatively: prediction uncertainty increases with distance and after distance > 3.5, the uncertainty reaches over 10 kcal/mol.

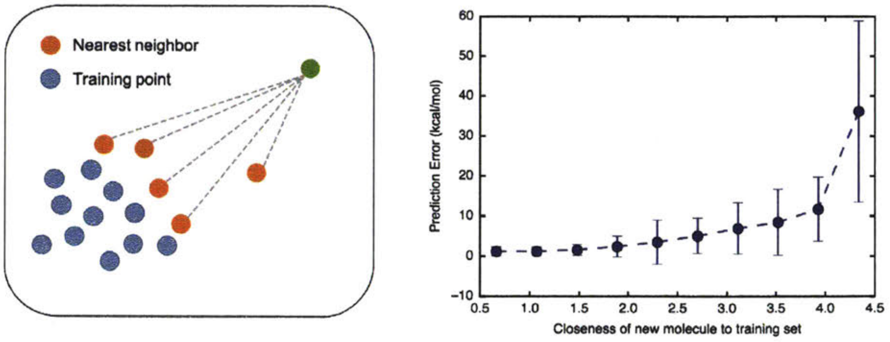


Figure 5.14: Relation between absolute prediction error for formation enthalpy and distance to training data (defined as average distance to 5 nearest neighbors in training set). Test datasets are `polycyclic_2954_table_test` and `cyclic_O_only_table_test`

5.5.3.2 Ensemble-based uncertainty estimation

In addition to the intuitive approach, a more rigourious non-parametric ensemble method motivated by bootstrap sampling was designed. The bootstrap principle is to approximate a population distribution by a sample distribution. In its most common form, bootstrap generates k sets of samples D_0,..., D_k from a given data set D by resampling uniformly with replacement [10]. Each bootstrap data set D_i is expected to have a fraction of the unique samples of D and the rest being duplicates. If the original data set is a good approximation of the population of interest, one can derive the sampling distribution of a particular statistic from the collection of its values arising from the k data sets generated by bootstrapping. Similarly, one can train a committee of k models using the bootstrap data sets and derive ensemble outputs for a query, which is known as bagging or bootstrap aggregating [9, 11]. Since the diversity of the outputs implies the uncertainty in the prediction, one can calculate the standard deviation of the outputs to quantify the uncertainty and evaluate the potential benefits of obtaining an accurate value for that molecule, e.g., by performing a quantum chemistry calculation.

In this study, the ensemble models were implemented using dropout training with neural networks. That is, instead of building multiple MCNN models, we trained the original MCNN model from scratch (both MCNN featurization and regression) with multiple dropout masks, as shown in Figure 5.15.

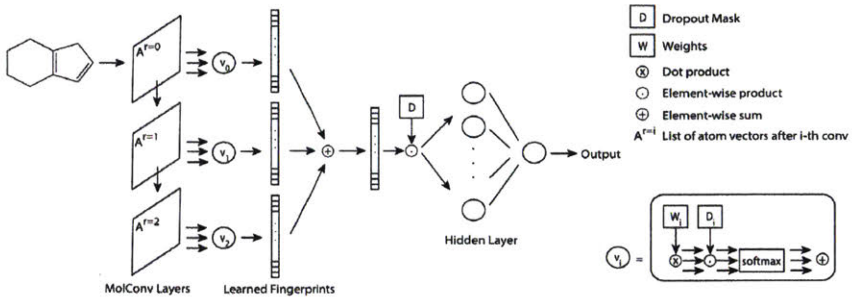


Figure 5.15: Dropout training with MCNN model enables uncertainty estimation

Unlike the standard dropout procedure in which the mask is generated on-the-fly during training, we randomly generated a set of masks before training and saved them along with the weights of the networks as part of the model. Since applying dropout masks removes non-output units from a fully connected network [12], a standard neural net with k dropout masks can be viewed as an ensemble of k sub-networks that share weights. For each training step, one of the sub-networks was randomly selected and optimized with one example (mini-batch of size one). Therefore, each of the sub-networks is expected to see some duplicated examples and only a fraction of the training data just as training ensemble models with bootstrap data sets. The ensemble prediction and estimated uncertainty were derived by averaging and taking standard deviation of the sub-network outputs.

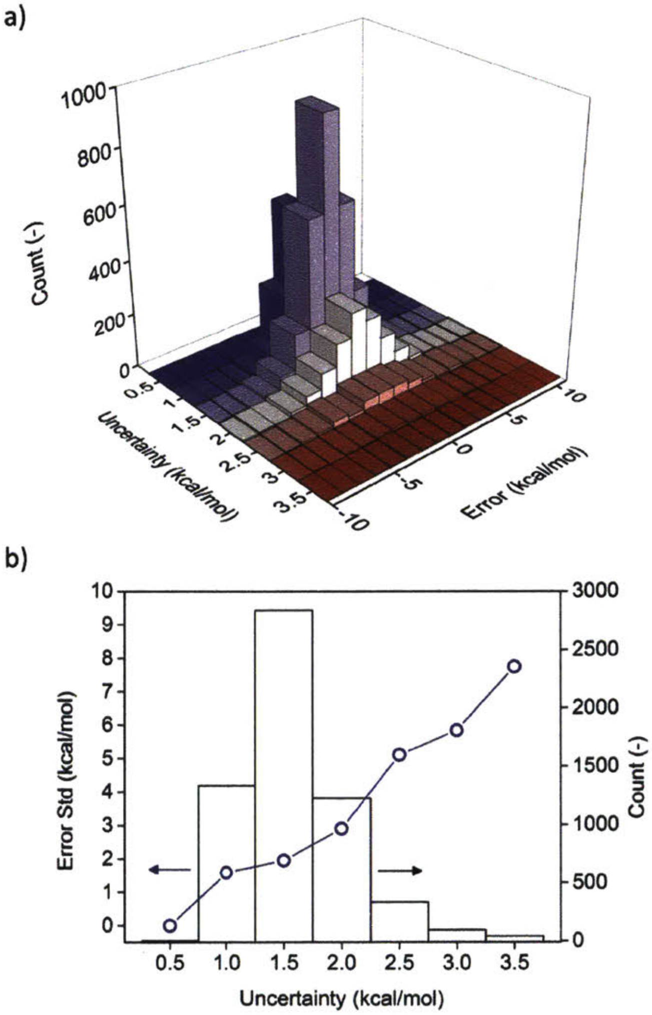


Figure 5.16: Errors and uncertainties in the predicted enthalpy of formation. The first panel, (a), shows that the predictions with higher uncertainties tend to have a broader true error distribution. This observation can be confirmed by the second panel, (b), which shows a clear positive correlation between the estimated uncertainties and the standard deviations of the true error distributions.

As shown in Figure 5.16a, the error distribution is bell-shaped and centered at the origin at each uncertainty level. Instead of directly interpreting the estimated uncertainty (the square root of variance in the committee predictions) as a quantitative estimation of the true error in a prediction, one should view this uncertainty as a descriptor of the error distribution to which the prediction belongs. Figure 5.16b shows a clear positive correlation between the estimated uncertainty and the standard deviation of the true error distribution at each uncertainty level. Therefore, if the uncertainty in a prediction is small, the error distribution the prediction belongs to should be narrow, and the probability of having a large error in that prediction should be low. Moreover, because the estimated uncertainty correlates with the standard deviations of the error distributions, one can divide the errors by the associated uncertainties to derive a "standardized" error distribution as shown in Figure 5.17.

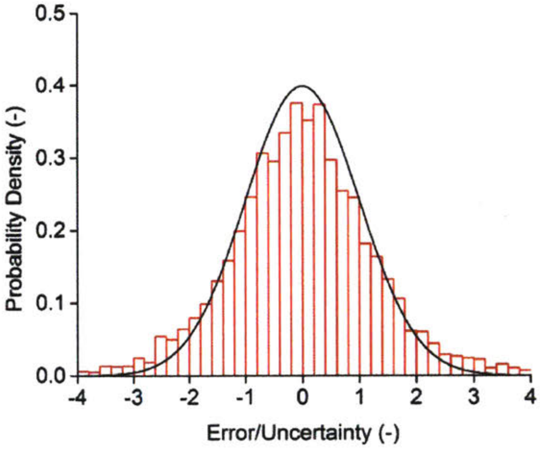


Figure 5.17: Distribution of standardized error (error/uncertainty). The black curve shows a standard normal distribution ($\sigma = 1$).

Though the standardized error distribution is not strictly normal (slightly broader than a normal distribution), it provides a sense of prediction quality, and can be viewed as a working curve for this method. For instance, given the estimated uncertainty of 1 kcal/mol, the probablity of having true error > 3 kcal/mol is less than 5%.

5.6 Self-evolution

As RMG models increasingly complex systems, thermochemistry estimator needs to expand its applicability domain. Collecting new training data usually requires intensive computation (e.g., quantum mechanics calculation), so it becomes a practical issue how to suggest new training data that is valuable for the estimator to improve its performance.

In this section we demonstrate the MCNN estimator's ability of suggesting new training data via a numerical experiment in Subsection 5.6.1. That greatly facilitates the construction of a pipeline that enables self-evolution of the thermochemistry estimator over time (Subsection 5.6.2, 5.6.3). Subsection 5.6.4 shows the expanded prediction capability of our MCNN estimator in three dimensions.

5.6.1 Experiment of new data selection

In this numerical experiment, we trained a MCNN model with dropout masks (Figure 5.15) on `polycyclic_2954_table_train` and `cyclic_O_only_table_train` and tested it against `polycyclic_2954_table_test`, `cyclic_O_only_table_test`, exactly same as we did for the MCNN model in Section 5.4. In addition, we tested it on `N_cyclics_table_test`, a test dataset with 9,995 nitrogen-containing molecules.

Since the model has only seen molecules composed of C, H, and O atoms, the MAE on `N_cyclics_table_test` (18.1 kcal/mol) is much higher than MAEs (~ 2 kcal/mol) of `polycyclic_2954_table_test` and `cyclic_O_only_table_test`. The model correctly assigned higher uncertainties to molecules in `N_cyclics_table_test` than those in `polycyclic_2954_table_test`, `cyclic_O_only_table_test`; Figure 5.18 shows about 1.5% of the test hydrocarbons and oxygenates have uncertainties higher than 3 kcal/mol, while 9% of the test nitrogen-containing molecules exceed this level of uncertainty. Therefore, with cutoff of 3 kcal/mol, the model would suggest picking nitrogen-containing molecules as additional training data six times more likely than picking hydrocarbons or oxygenates.

Moreover, once the model is trained on a few nitrogen-containing molecules, it starts to recognize this new type of molecule and make better predictions with more accurate uncertainty estimations, which further increases the chance of identifying most beneficial training data. As shown in Figure 5.18, if one adds 100 nitrogen-containing molecules to the old training data, the percentage of the test nitrogen-containing examples that exceed the 3 kcal/mol uncertainty level jumps up to 22%, suggesting 14 times higher chance of picking less known examples (nitrogen-containing molecules) than the better known ones (hydrocarbons and oxygenates).

5.6.2 Pipeline for self-evolving estimator

Using the ability to suggest new training data effectively, a pipeline was developed to achieve self-evolution of the estimator. As shown in Figure 5.19, Users query

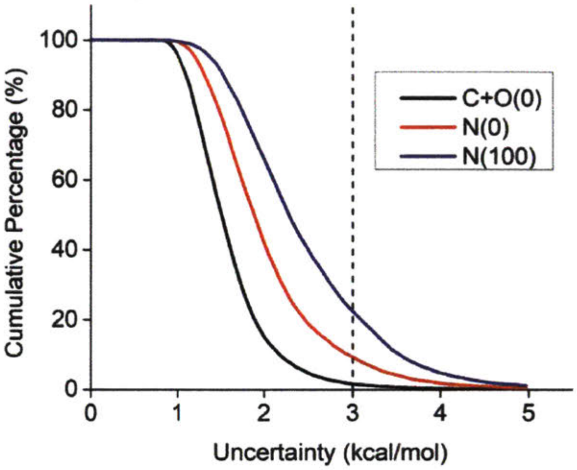


Figure 5.18: Cumulative percentage of test molecules above certain uncertainty level. The black curve, C+O(0), is the combined result of two test datasets: `polycyclic_2954_table_test`, `cyclic_0_only_table_test`. The red and blue curves, N(0) and N(100), are results of a test set composed of 9,995 nitrogen-containing species. The numbers in the parentheses are the number of nitrogen-containing species in the training data.

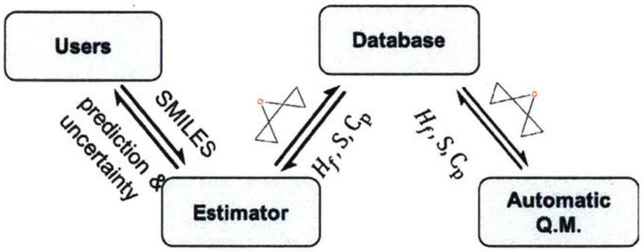


Figure 5.19: Adaptive pipeline consists of four components: users, estimator, database, automatic quantum mechanics calculator

for thermochemistry from the Estimator which returns both prediction and uncertainty. If the uncertainty is higher than a pre-defined threshold, the corresponding query molecule gets registered into Database as an unlabeled data point (i.e., without thermochemistry data). The Database is responsible for analyzing and prioritizing the unlabeled molecules as well as storing the ones already labeled with their thermochemical parameters. In the meanwhile, it communicates with the Automatic Quantum Mechanics Calculator (autoQM, implemented by the author) to launch ab initio calculations for the highly uncertain molecules. When the calculations are finished, the thermochemistry data are sent back to the Database waiting to serve as additional training examples in the next update of the estimator.

Table 5.2: Data distribution and statistical errors during the first three interations in estimator self-evolution described in Subsection 5.6.3

		Model 1	Model 2	Model 3
Number of labeled examples	Hydrocarbons + Oxygenates	23,582	23,582	23,582
	Nitrogen-containing molecules	0	3,846	9,973
Number of unlabeled examples[a]	High Uncertainty[b]	3,846	6,127	1,049
	Low Uncertainty[b]	36,135	30,008	28,959
Statistical errors (kcal/mol)	MSE	-15.12	0.48	0.08
on `N_cyclics_table_test`	MAE	18.09	5.21	3.21
(9,995 test molecules)	RMSE	23.56	6.84	4.30

[a] All the unlabeled examples are nitrogen-containing species. Identified high-uncertainty examples will be labeled and added to the training set to get a new generation of model.

[b] The cutoff between low and high uncertainty is 3 kcal/mol.

Table 5.3: Prediction performance (mean absolute error) of latest MCNN estimator

Molecule domain	Shape	H_f at 298 K (kcal/mol)	S at 298 K (cal/mol/K)	C_p at 300 K (cal/mol/K)	Training/test set size (ratio: 4:1)
C, H, O	non-cyclic	1.36	1.08	0.24	4,160/1,040
	cyclic	1.81	0.55	0.15	23,584/5,896
N-containing	non-cyclic	1.61	0.93	0.21	5,808/1,452
	cyclic	1.82	0.62	0.18	39,980/9,995
S-containing	non-cyclic	4.52	1.83	1.32	104/26
	cyclic	19.66	4.65	1.7	12/3

5.6.3 Self-evolving case study

To demonstrate the self-evolving process, we registered 39,981 unlabeled nitrogen-containing molecules and 23,582 labeled hydrocarbons and oxygenates to the database (Table 5.2). The process starts with training the model with the labeled examples in the database. Since only the hydrocarbons and oxygenates have labels at the beginning, the initial model (Model 1) is a C,H,O-based model which has not been exposed to any nitrogen-containing molecules. The uncertainties of the unlabeled examples, i.e., all the nitrogen-containing molecules, are calculated by Model 1 and those high-uncertainty ones (uncertainty is higher than a certain cutoff value) will

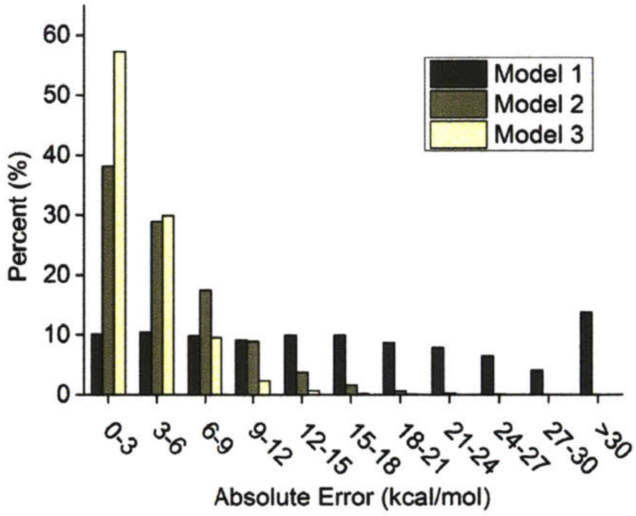


Figure 5.20: Error distributions of the low-uncertainty molecules identified by Model 1, Model 2 and Model 3, the three generations of estimators from self-evolution case study

be labeled and incorporated into the training data to update the model. Since in practice, high-uncertainty examples have to be subjected to ab initio calculations to derive their labels, the cutoff for high and low uncertainties is a parameter that needs to be chosen based on the available computational resources and the requirements of accuracy in predictions. If the cutoff is too low, a large fraction of the unlabeled examples will be subjected to ab initio calculations. For this demonstration, the cutoff was set to 3 kcal/mol.

Similar to experiment result in Section 5.6.1, Table 5.2 shows Model 1 identifies roughly 10% of the nitrogen-containing molecules as high-uncertainty molecules. Incorporating all of these into the training set significantly improves uncertainty estimation quality so the model generated by the second round of training (Model 2) found 6,127 high-uncertainty examples from the low-uncertainty species of Model 1. One might expect more high-uncertainty species to be found after the next round of training (Model 3). But in fact, the number of high uncertainty species determined by Model 3 is significantly lower than the previous two models because not only the quality of uncertainty estimates but also the accuracy of prediction have been improved for nitrogen-containing species in this active learning process.

As shown in Figure 5.20, most of the low-uncertainty species determined by Model 3 indeed have small errors (< 6 kcal/mol), suggesting that the model has successfully expanded its scope from just C,H,O to includes nitrogen-containing molecules.

5.6.4 Expanded prediction capability

The MCNN estimator has significantly expanded its prediction capability since it was first trained on enthalpy formation data of hydrocarbon polycyclics; now it is able to predict for a wide range of molecules for the tasks of enthalpy formation, entropy and heat capacity, as shown in Table 5.3.

C,H,O,N-based molecules have been predicted well, with MAE for H_f: ~ 2 kcal/mol, for both non-cyclic and cyclic categories. Even with very limited training data for S-containing molecules ($\sim$ 100 data points), the MCNN estimator is able to predict reasonably well for the non-cyclic category across all the three tasks. With new training data flows in database, we can expect the estimator to further effectively improve its performance in the S molecule domain. Performance metrics are updated daily through the web dashboards:

- Enthalpy: `kehangsblog.com/thermo_predictor/overall_performance/Hf298/`
- Entropy: `kehangsblog.com/thermo_predictor/overall_performance/S298/`
- Heat capacity: `kehangsblog.com/thermo_predictor/overall_performance/Cp/`

5.7 Conclusion

In previous attempts to estimate thermochemistry, we greatly relied on human chemical insights to design effective featurization, which often comes with hidden assumptions (e.g., heuristic method assumes inter-bicyclic interaction is negligible). However, as RMG models increasingly complex chemical systems, the thermochemistry estimator encounters new molecule domains which may break the hidden assumptions (e.g., large fused polycyclics breaking the assumption of heuristic method) and require new insights for featurization. When such new chemical insights are unavailable or hard to be converted to straighforward and generalizable estimation formula, one can hardly improve prediction performance in the target molecule domain even when data is available.

To let featurization learn from data rather than depend on human input, this chapter presents a new approach via molecular convolutional neural networks (MCNN). The MCNN estimator successfully learnt an effective, compact and meaningful featurization that helps gain higher accuracy for large fused polycyclics than heuristic method, without asking for human insights. Its performance on other types of polycyclics is also superior to those of the previous methods.

We also designed the uncertainty estimation scheme for the MCNN estimator, which eventually leads to the construction of a pipeline that makes MCNN estimator self-evolve over time. The MCNN estimator has significantly expanded its prediction capability since its first generation trained on hydrocarbon cyclics' formation enthalpy data; it is now able to predict enthalpy, entropy and heat capacity for C,H,O,N,S-based molecules (both cyclics and non-cyclics). A web service returning the estimated values for any input C,H,O,N,S molecules is freely accessible via http://kehangsblog.com/thermo_predictor/thermo_estimation. Besides automatic mechanism generation tools, we also recommend using it in general applications where a large molecular space has to be scanned/explored with limited time such as high throughput virtual screening, automatic transition state search, etc.

5.8 References

[1] D. Rogers and M. Hahn. "Extended-Connectivity Fingerprints." *Journal of Chemical Information and Modeling* 50 (5), May 2010, pp. 742–754. ISSN: 1549-9596. DOI: 10.1021/ci100050t. URL: http://dx.doi.org/10.1021/ci100050t.

[2] A. Aspuru-Guzik, D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguire, R. Gomez-Bombarelli, T. D. Hirzel, and R. P. Adams. "Convolutional Networks on Graphs for Learning Molecular Fingerprints." en_US. Neural Information Processing Systems Foundation, Inc., 2015. URL: https://dash.harvard.edu/handle/1/24873720.

[3] C. W. Coley, R. Barzilay, W. H. Green, T. S. Jaakkola, and K. F. Jensen. "Convolutional Embedding of Attributed Molecular Graphs for Physical Property Prediction." *Journal of Chemical Information and Modeling* 57 (8), Aug. 2017, pp. 1757–1772. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.6b00601. URL: https://doi.org/10.1021/acs.jcim.6b00601.

[4] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. v. Lilienfeld. "Quantum chemistry structures and properties of 134 kilo molecules." en. *Scientific Data* 1, Aug. 2014, p. 140022. ISSN: 2052-4463. DOI: 10.1038/sdata.2014.22. URL: http://www.nature.com/articles/sdata201422.

[5] F. Chollet. *keras*. https://github.com/keras-team/keras. 2015.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient Estimation of Word Representations in Vector Space." *arXiv:1301.3781 [cs]*, Jan. 2013. arXiv: 1301.3781. URL: http://arxiv.org/abs/1301.3781.

[7] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.” *arXiv:1506.02142 [cs, stat]*, June 2015. arXiv: 1506.02142. URL: http://arxiv.org/abs/1506.02142.

[8] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. “Weight Uncertainty in Neural Networks.” *arXiv:1505.05424 [cs, stat]*, May 2015. arXiv: 1505.05424. URL: http://arxiv.org/abs/1505.05424.

[9] A. Bhaskara, M. Ghadiri, V. S. Mirrokni, and O. Svensson. “Linear Relaxations for Finding Diverse Elements in Metric Spaces.” 2016, pp. 4098–4106. URL: https://papers.nips.cc/paper/6500-linear-relaxations-for-finding-diverse-elements-in-metric-spaces.pdf.

[10] R. T. Bradley Efron. *An Introduction to the Bootstrap.* en. May 1994. URL: https://www.crcpress.com/An-Introduction-to-the-Bootstrap/Efron-Tibshirani/p/book/9780412042317.

[11] L. Breiman. “Bagging Predictors.” en. *Machine Learning* 24 (2), Aug. 1996, pp. 123–140. ISSN: 0885-6125, 1573-0565. DOI: 10.1023/A:1018054314350. URL: https://link.springer.com/article/10.1023/A:1018054314350.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting.” *Journal of Machine Learning Research* 15, 2014, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

6

Heavy System Application: Thermal Decomposition of Heavy Oils

Thermal decomposition of heavy oils forming light petroleum and gases has been very crucial in many energy applications. One important example is crude oil refinery and upgrading (especially the produced crude oils have become heavier over the recent years), which provides the energy support for most human and industrial activities around the world.

However, even today we have very limited chemical knowledge of the decomposition process. One of the major difficulties in the investigation is that crude oils have hundreds of heavy components and many contain heteroatoms (e.g., nitrogen and sulfur). Due to the overwhelming complexity associated with the heavy systems, lumping strategy is widely employed to simplify kinetic models.

However, it greatly hampers models' extrapolation potential, which becomes a key issue for applications where direct experiments are impractical (such as geological formation with timescale in the order of millions of years) and models are fitted at other experimentally feasible conditions.

As discussed in Chapter 1, reaction mechanisms at level 3 distinguish molecules and model chemical process through elementary reaction steps, which retain most fundamental chemistry and thus have greater potential to extrapolate. The construction of detailed kinetic mechanisms by hand for such complex systems is extremely difficult. Thus, automatic reaction generation software RMG has been used.

In this study, we chose PDD (phenyldodecane, Figure 6.1) as the model compound to study thermal decomposition of heavy oils. Literature has also presented extensive experimental and mechanistic investigations on PDD pyrolysis [1–5]. On the experimental side, PDD and many of its liquid and gaseous products are readily recoverable and quantifiable by gas chromatography.

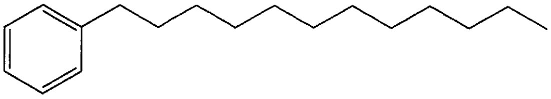


Figure 6.1: PDD ($C_{18}H_{30}$); phenyldodecane; 246 amu

6.1 Methods

6.1.1 Reaction model generation

RMG has been used to model a variety of processes and give predictions consistent with combustion and pyrolysis experiments for a wide range of applications at high temperature [6–10]. An attempt before this thesis was made to generate a PDD pyrolysis model with early version of RMG [11]; however, it lacked critical species found in experiment and never ran to completion. In this study, we used the new version of RMG software introduced by this thesis.

The PDD model was generated at 35 MPa and temperatures of 250, 350, and 450°C, with a tolerance of $\epsilon = 0.2$ and termination PDD conversions of 0.2, 0.6, and 0.8 respectively. The number of radical electrons was restricted to 1 to aid model convergence. The final PDD mechanism contains 344 species and 9204 reactions.

The completion of this model largely relies on recent RMG advances such as pruning [12], reaction filtering [13], polycyclic thermochemistry estimation methods [14] and improved thermo / kinetics data and estimation rules [15].

Reaction simulations were carried out in a ideal-mixture homogenous batch reactor module of CHEMKIN-PRO [16].

6.1.2 Quantum chemical calculations

Many reaction rate constants and thermochemical parameters from previous work [11, 13–15] were added to RMG database and available for use in this study. Additionally, we identified reactions that are important for certain predictions and refined their kinetic parameters using *ab initio* methods.

Specifically, we used the computational chemistry program Gaussian 03 [17] to optimize geometries and calculate vibrational frequencies for reactants, products and transition states at the CBS-QB3 level of theory. The CanTherm (open-source tool, bundled within RMG) was used to translate quantum mechanic calculations to rate constants and thermochemical parameters using transition state theory. Modified Arrhenius constants were then derived with the kinetic parameters added to RMG as rate rules to allow similar reactions to take place.

6.2 Results and Discussion

Low temperature modeling has an inherent challenge where uncertainties in kinetics parameters derived from *ab initio* calculations are greatly amplified [13]. For instance, a typical rate coefficient with activation energy E_a uncertainty $\delta E = 2$ kcal/mol potentially contributing a factor of 2.7 error at 1000 K can contribute a factor of $2.7^2 = 7.3$ error at 500 K. Thus, the model should be validated extensively. A collection of various experimental data for PDD pyrolysis at different conditions [2, 4, 5] was obtained from literature and used for model validation. Additionally, Dr. Reeves conducted PDD confined pyrolysis experiments at 350°C for this study.

6.2.1 PDD conversion prediction

The RMG model for PDD pyrolysis is simulated at various conditions and compared against experimental data in Figure 6.2 where a shaded error bar reflects a factor of 2 uncertainty. The model predicts the conversion of PDD with moderate accuracy. There is general agreement between these datasets except for the Lewan experiments, which were conducted at atmospheric pressure, in contrast to the Behar and Reeves experiments conducted at 14 MPa and 35 MPa, respectively. Further investigation may be necessary to determine the full extent of the effects of low pressure on PDD decomposition. At the temperature and pressure conditions of the Reeves experiment, confined pyrolysis occurs in a single, liquid phase, but the system is thought to be multiphase at the conditions of Lewan's experiment.

We currently use RMG's gas phase kinetics of high-pressure-limit rate coefficients to model this, approximating the system as an ideal-mixture homogenous batch reactor with inert wall. Constant-volume was assumed with adjusted pressure to reflect liquid PDD density. An investigation into the error caused by these approximations might be necessary to distinguish the chemistry error from the physics error. In the future, using an equation of state to constrain the volume profile in CHEMKIN may improve the approximation.

6.2.2 Products from β-scission

Additional validation of the model was performed through comparisons with species profiles from the 400°C experiments of Savage and Klein [2] in Figures 6.3 to 6.4. The model well predicts the PDD conversion and major products such as toluene, decane and ethylbenzene yields at these conditions, while it slightly overpredicts styrene and underpredicts undecane.

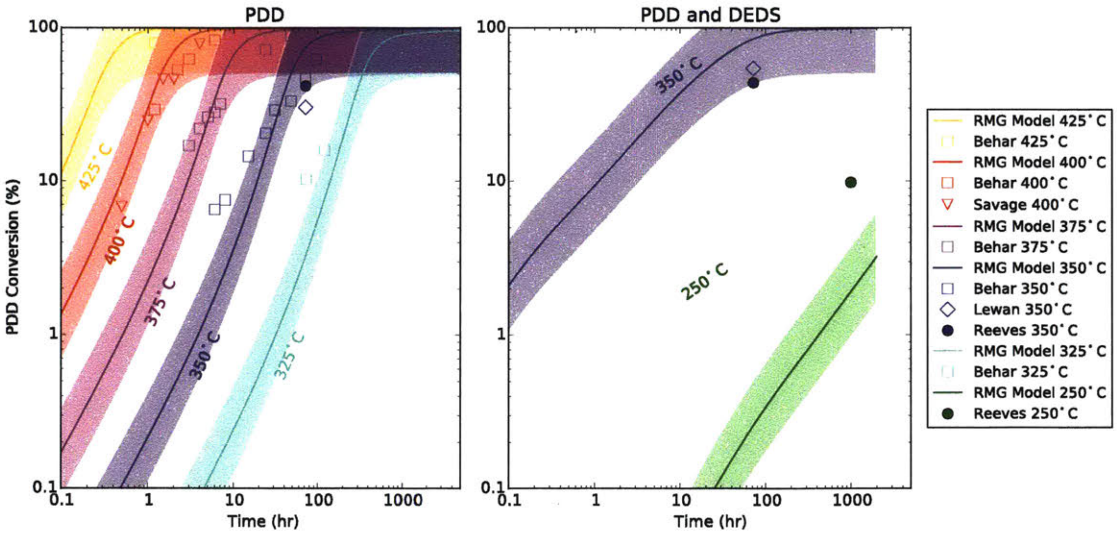


Figure 6.2: PDD conversion in neat pyrolysis and in the presence of DEDS with respect to temperature. The RMG model is simulated at P = 35 MPa and at different temperatures with an shaded error bar reflecting a factor of 2 uncertainty. The model is plotted against various experimental data by Lewan, Behar, Savage and Reeves

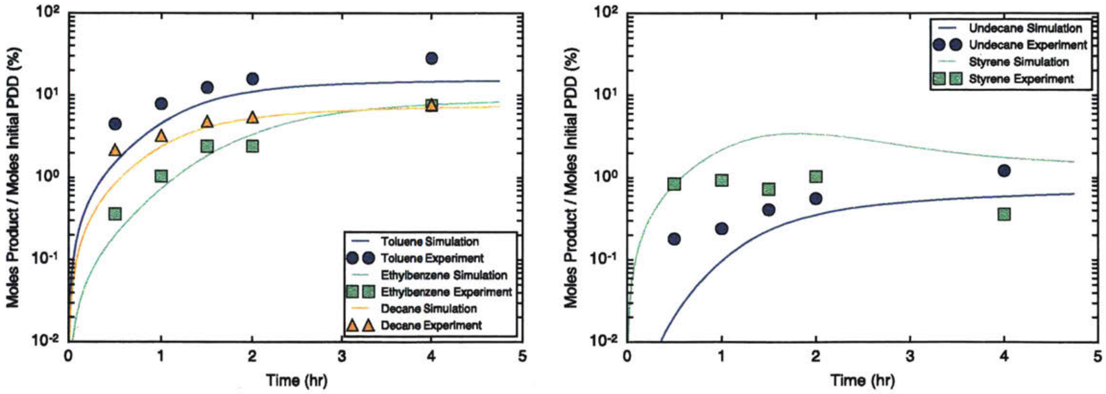


Figure 6.3: Simulated PDD neat pyrolysis molar yields of major species compared against Savage and Klein experiments conducted at 400°C.

Figure 6.5 shows flux analysis of the reaction network where the major products predicted in PDD neat pyrolysis are toluene, undecene, ethylbenzene, and decane. Toluene and undecene arise primarily from PDDrad3, the radical formed by hydrogen abstraction from the 3rd carbon for the phenyl group in PDD. Styrene, ethylbenzene, and decane are formed primarily from PDDrad1, the radical formed by hydrogen

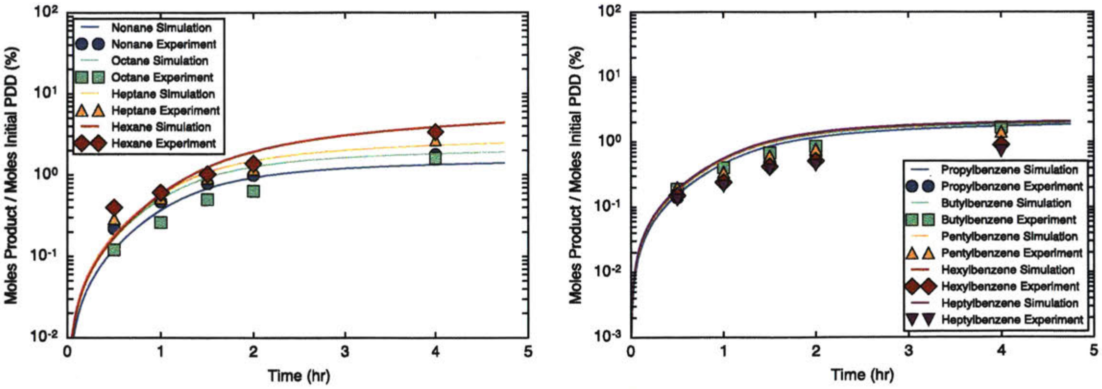


Figure 6.4: Simulated PDD neat pyrolysis molar yields of alkane and aromatic species compared against Savage and Klein experiments conducted at 400°C.

abstraction from the benzylic carbon adjacent to the phenyl group in PDD. This radical is resonantly stabilized by the presence of the aromatic ring, therefore more stable than all other PDD radicals formed from hydrogen abstraction on the aliphatic chain.

No styrene was observed in the Reeves experiments. The Savage 400°C [2] experiments also show little styrene formation. This suggests that styrene formed from PDDrad1 apparently reacts rapidly at these conditions, probably most of it converts to ethylbenzene. This pathway is found by RMG: styrene can participate in a reverse disproportionation reaction with PDD to form two resonantly stabilized radical species, leading to the formation of ethylbenzene. The kinetic rate of the reverse disproportionation reaction can largely affect the predicted styrene concentration. While the model suggests reverse disproportionation is the dominant pathway consuming styrene, other pathways are also possible, e.g., any radical can add to styrene.

The undecane formation in the RMG model is primarily through the reverse disproportionation reactions where undecene receives a hydrogen from PDD or toluene to form undecyl radicals. The undecyl radical then abstracts another hydrogen to form undecane. These reverse disproportionation reactions have very low branching fraction, as shown in Figure 6.5, but are very important for undecane production. The fact that current model estimates those parameters from non-exact reaction analogs might explain the undecane underprediction. Future refinements of the reverse disproportionation rates could be important to further improve the prediction of undecane.

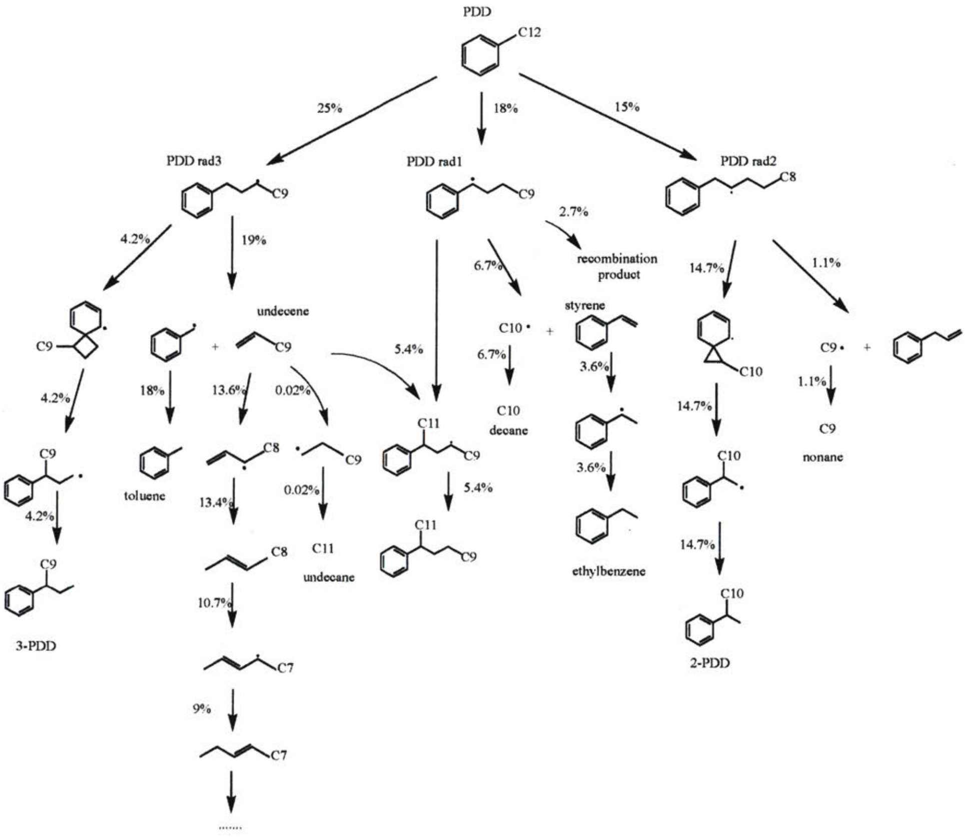


Figure 6.5: Fluxes of PDD decomposition at 350°C and 35 hours.

6.2.3 Products from *ipso*-isomerization

Besides products derived from β-scission of initial PDD radicals, a number of PDD isomers are found in Reeves experiments, as depicted in Figure 6.6. Those various carbon-shifted isomers are from another major decomposition pathway: *ipso*-isomerization of PDD radicals, shown in Figure 6.7. These reactions are analogous to the reactions previously published in the phenyldecane Burklé-Vitzthum mechanism. [18]. With the recent advances in polycyclic thermochemistry estimation methods [14] and quantum mechanic calculations of the `Intra_R_Add_Exocyclic` and `Intra_R_Add_Endocyclic` family kinetics carried out by Dr. Khanniche, RMG was able to find these reaction pathways.

Experimentally *beta*-scission and *ipso*-isomerization are the two major competing pathways for PDD decomposition at 350°C, as summarized in Table 6.1. Figure 6.8 shows the RMG model well captures both of the pathways. But the branching ratio of *beta*-scission to *ipso*-isomerization is underpredicted. That may be caused by the fact that the rate parameters are estimated from inconsistent sources. Further high quality calculations with a consistent level of theory is recommended to refine the branching ratio.

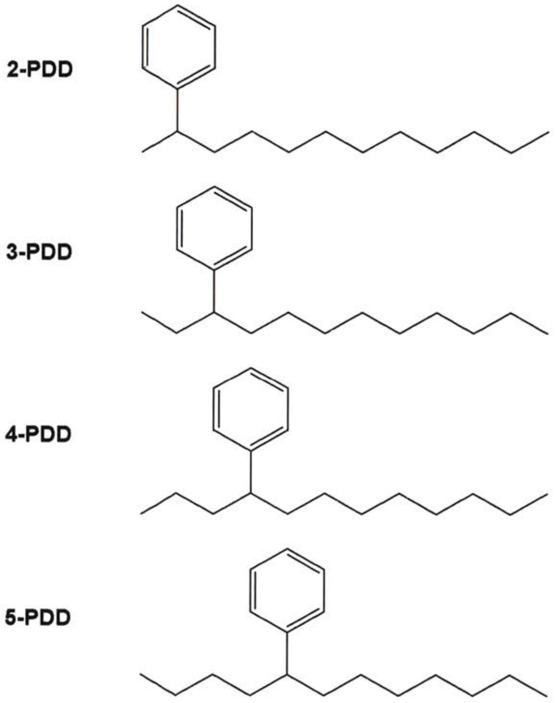


Figure 6.6: PDD isomers identified in experiment.

6.2.4 Heavy products

Higher molecular weight products were detected in the experiments of the present work although their chemical structures could not be identified. The model suggests most of these products are formed from the recombination of resonantly stabilized radicals (Figure 6.9). Further experimental probing using analytical chemistry techniques that can identify the structures of these recombination products will be useful.

A general product trend in neat pyrolysis in this temperature range is that saturated hydrocarbons are selectively formed over their unsaturated counterparts, i.e. there are higher concentrations of butane vs. butene, hexane vs. hexene, octane vs octene, etc. This result is also corroborated in the Savage [2] data. Note that the simple process where large alkylaromatic converts to small alkylaromatic and alkane

Table 6.1: The β-scission and *ipso*-isomerization reactions of the PDD initial radicals leading to products detected in experiment. PDDrad1 has a radical closest to the phenyl group, while PDDrad12 has a radical furthest away from the phenyl group.

Precursor	Reaction Type	Products
PDDrad1	β-scission	styrene + decane
PDDrad2	β-scission	propenylbenzene + nonane
	ipso-isomerization	2-PDD
PDDrad3	β-scission	toluene + undecene
		butenylbenzene + octane
	ipso-isomerization	3-PDD
PDDrad4	β-scission	ethylbenzene + decene
		pentenylbenzene + heptane
	ipso-isomerization	4-PDD
PDDrad5	β-scission	propylbenzene + nonene
		hexenylbenzene + hexane
	ipso-isomerization	5-PDD
PDDrad6	β-scission	butylbenzene + octene
		heptenylbenzene + pentane
	ipso-isomerization	6-PDD
PDDrad7	β-scission	pentylbenzene + heptene
		octenylbenzene + butane
	ipso-isomerization	5-PDD
PDDrad8	β-scission	hexylbenzene + hexene
		nonenylbenzene + propane
PDDrad9	β-scission	heptylbenzene + pentene
		decenylbenzene + ethane
PDDrad10	β-scission	octylbenzene + butene
		undecenylbenzene + methane
PDDrad11	β-scission	nonylbenzene + propene
PDDrad12	β-scission	decylbenzene + ethene

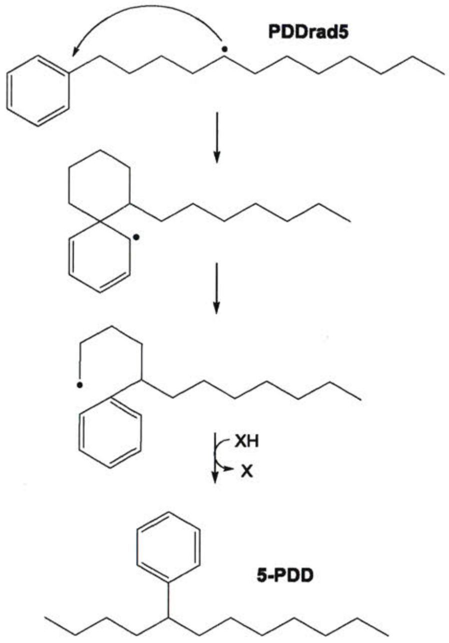


Figure 6.7: *Ipso*-isomerization reaction of a PDD radical leading to the formation of a PDD isomer. Several related isomerizations also have high rates.

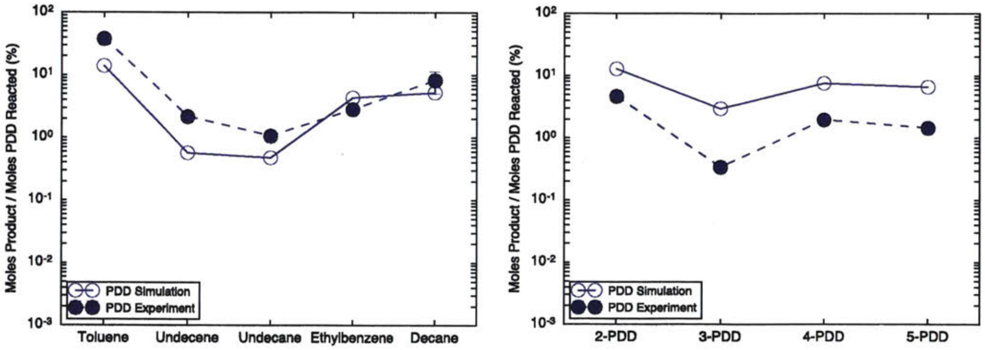


Figure 6.8: Simulated and experimental data of major species selectivities (moles produced / moles PDD reacted) at 350°C after 72 hours of reaction time. Left panel shows the major products from β-scission pathway, while right panel shows the major products from *ipso*-isomerization pathway.

requires some other process to occur in concert to provide the needed H atoms. What is that process? One hypothesis is that at longer timescales, unsaturated hydrocarbons begin to cyclize via diels-alder type reactions, which give up H atoms as they form polycyclic and polyaromatic ring structures eventually leading to coke. However,

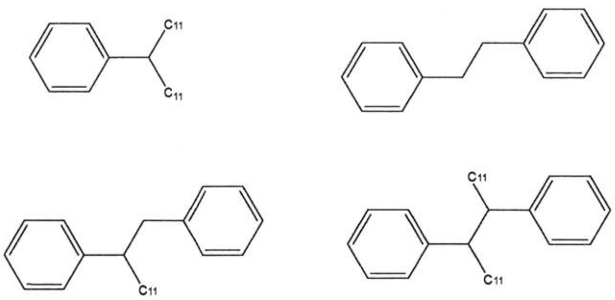


Figure 6.9: Recombination products in neat PDD pyrolysis.

RMG is currently not able to estimate these reactions due to our lack of elementary reaction families for polyaromatic hydrocarbon (PAH) formation.

6.3 Conclusion

In this chapter, we conducted mechanistic investigation on the thermal decomposition of heavy oils. Aimed to better understand the chemical process and provide a reaction model with high extrapolation potential, we generated a detailed mechanism using RMG. PDD is chosen as the model compound for this study. Although it was too complex to model earlier, the new version of RMG was able to generate a complete model for PDD pyrolysis.

The model predicts PDD conversion and the major products of PDD pyrolysis experiments found in literature and in the new experiments by Dr. Reeves except that there is an overprediction of styrene and underprediction of undecane. These differences could be investigated by looking into the kinetics of related reverse disproportionation reactions to verify that the branching ratios used in this model are correct. It is also possible that the present model omits some pathways which consume styrene.

For the underprediction of the branching ratio between β-scission and *ipso*-isomoerization pathways, further high quality calculations with more consistent level of theory is recommended.

6.4 References

[1] B. Blouri, F. Hamdan, and D. Herault. "Mild cracking of high-molecular-weight hydrocarbons." *Ind. Eng. Chem. Proc. DD* 24 (1), 1985, pp. 30–37. DOI: 10.1021/i200028a005.

[2] P. E. Savage and M. T. Klein. "Discrimination between molecular and free-radical models of 1-phenyldodecane pyrolysis." *Ind. Eng. Chem. Res.* 26 (2), 1987, pp. 374–376. DOI: 10.1021/ie00062a034.

[3] F. Billaud, P. Chaverot, M. Berthelin, and E. Freund. "Thermal decomposition of aromatics substituted by a long aliphatic chain." *Ind. Eng. Chem. Res.* 27 (8), 1988, pp. 1529–1536. DOI: 10.1021/ie00080a030.

[4] M. D. Lewan. "Sulphur-radical control on petroleum formation rates." *Nature* 391 (6663), 1998, pp. 164–166.

[5] F. Behar, F. Lorant, H. Budzinski, and E. Desavis. "Thermal Stability of Alkylaromatics in Natural Systems: Kinetics of Thermal Decomposition of Dodecylbenzene." *Energy & Fuels* 16 (4), 2002, pp. 831–841. DOI: 10.1021/ef010139a.

[6] K. M. Van Geem, M. F. Reyniers, G. B. Marin, J. Song, W. H. Green, and D. M. Matheu. "Automatic reaction network generation using RMG for steam cracking of n-hexane." *Aiche J.* 52 (2), 2006, pp. 718–730. DOI: 10.1002/Aic.10655.

[7] K. M. Van Geem, S. P. Pyl, G. B. Marin, M. R. Harper, and W. H. Green. "Accurate High-Temperature Reaction Networks for Alternative Fuels: Butanol Isomers." *Ind. Eng. Chem. Res.* 49 (21), 2010, pp. 10399–10420. DOI: 10.1021/ie1005349.

[8] A. G. Vandeputte, M. K. Sabbe, M.-F. Reyniers, and G. B. Marin. "Modeling the Gas-Phase Thermochemistry of Organosulfur Compounds." *Chem-Eur. J.* 17 (27), 2011, pp. 7656–7673. DOI: 10.1002/chem.201002422.

[9] S. S. Merchant, C. F. Goldsmith, A. G. Vandeputte, M. P. Burke, S. J. Klippenstein, and W. H. Green. "Understanding low-temperature first-stage ignition delay: Propane." *Combustion and Flame* 162 (10), Oct. 2015, pp. 3658–3673. ISSN: 0010-2180. DOI: 10.1016/j.combustflame.2015.07.005. URL: http://www.sciencedirect.com/science/article/pii/S0010218015002060.

[10] C. W. Gao, A. G. Vandeputte, N. W. Yee, W. H. Green, R. E. Bonomi, G. R. Magoon, H.-W. Wong, O. O. Oluwole, D. K. Lewis, N. M. Vandewiele, and K. M. Van Geem. "JP-10 combustion studied with shock tube experiments and modeled with automatic reaction mechanism generation." *Combustion and Flame* 162 (8), Aug. 2015, pp. 3115–3129. ISSN: 0010-2180. DOI: 10.1016/j.combustflame.2015.02.010. URL: http://www.sciencedirect.com/science/article/pii/S0010218015000528.

[11] C. A. Class. "Predicting Organosulfur Chemistry in Fuel Sources." PhD thesis. Massachusetts Institute of Technology, 2015.

[12] K. Han, W. H. Green, and R. H. West. "On-the-fly pruning for rate-based reaction mechanism generation." *Computers & Chemical Engineering* 100, May 2017, pp. 1–8. ISSN: 0098-1354. DOI: 10.1016/j.compchemeng.2017.01.003. URL: http://www.sciencedirect.com/science/article/pii/S0098135417300030.

[13] C. W. Gao. "Automatic reaction mechanism generation: High Fidelity Predictive Modeling of Combustion Processes." eng. Thesis. Massachusetts Institute of Technology, 2016. URL: http://dspace.mit.edu/handle/1721.1/104205.

[14] K. Han, A. Jamal, C. A. Grambow, Z. J. Buras, and W. H. Green. "An Extended Group Additivity Method for Polycyclic Thermochemistry Estimation." en. *International Journal of Chemical Kinetics* 50 (4), Apr. 2018, pp. 294–303. ISSN: 1097-4601. DOI: 10.1002/kin.21158. URL: http://onlinelibrary.wiley.com/doi/10.1002/kin.21158/full.

[15] L. Lai, S. Gudiyella, M. Liu, and W. H. Green. "Chemistry of Alkylaromatics Reconsidered." *Energy & Fuels*, Feb. 2018. ISSN: 0887-0624. DOI: 10.1021/acs.energyfuels.8b00069. URL: https://doi.org/10.1021/acs.energyfuels.8b00069.

[16] Reaction Design. *CHEMKIN-PRO 15131.* 2013.

[17] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery Jr., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, and J. A. Pople. *Gaussian 03.* Gaussian, Inc., Wallingford, CT. 2004.

[18] V. Burklé-Vitzthum, R. Michels, G. Scacchi, and P.-M. Marquaire. "Mechanistic Modeling of the Thermal Cracking of Decylbenzene. Application to the Prediction of Its Thermal Stability at Geological Temperatures." *Ind. Eng. Chem. Res.* 42 (23), 2003, pp. 5791–5808. DOI: 10.1021/ie030086f.

7

Extension of RMG: A Fragment-based Kinetic Modeling Framework

Full-detailed molecule representation used in RMG allows very high fidelity to the true chemistry, and so has superior extrapolation potential. Unfortunately full-detail quickly becomes intractable for large molecules. To illustrate the poor scalabilty, consider an example feedstock with M carbons: up to L functionalized carbons (e.g., carbonyl, carboxyl, etc.) and remaining CH_2 carbons. Suppose there are S different carbon types for each functionalized carbon to choose from. The number of distinct molecules is

$$N_{full-detail} = \sum_{N=0}^{L} \binom{M}{N} S^N \tag{7.1}$$

The largest term in the summation is $\binom{M}{L}S^L$, which gives a lower bound for $N_{full-detail}$.

$$\ln N_{full-detail} \geq \ln\left(\binom{M}{L}S^L\right) = \ln\left(\frac{M!S^L}{(M-L)!L!}\right) \tag{7.2}$$

If we choose $M = 30$,$L = 15$ and $S = 8$, $\ln N_{full-detail} \geq 50$: it would require $e^{50} \approx 10^{22}$ distinct molecules to represent the feedstock. Even 1 byte per each molecule requires total memory of 10^{22} bytes $= 10^{10}$ terabytes, which is 30 times of entire Internet data of 2011 [1]. Unfortunately, many applications that have significant impact on energy, environment and society are complex chemical processes which may have even larger feedstocks with more than 30 carbons such as crude oil, kerogen, coal etc.

To model them, less-detailed representation methods are needed. Over the past decades, people have tried to understand those systems and model the essential chemistry. Lumping strategy is widely employed where in early practice molecules were lumped based on molecular weights (or boiling points). In lumped models, the reactions are often assumed first-order and irreversible. See an example of 3-lump model for the catalytic cracking process in Figure 7.1 [2]. Despite simplicity, the lumping strategy introduces an intrinsic error since the lumps contain molecules with different reactivities; they don't all react at the same rate as assumed in the lumped model. Consequently, the composition of a lump changes as the reaction proceeds (e.g. certain molecules get enriched) so experimentally the lump does not have fixed properties nor follow first order kinetics. The kinetic parameters are often fitted from a very limited set of pilot experiments, resulting in poor accuracy in extraploation. When the feed or the operating condition changes, new experiments are usually required and model parameters refitted (sometimes even model structure needs to be revised to account for new chemistry).

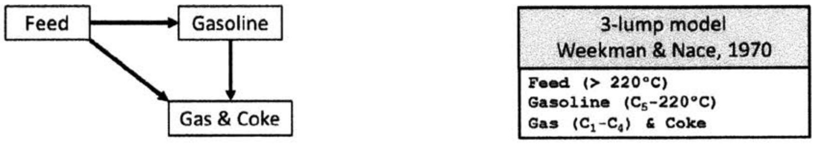


Figure 7.1: A simple lumped model for catalytic cracking process by Weekman and Nace. From Oliveira *et al.*

That greatly reduces the predictive capability of a lumped model. Structure-oriented lumping (SOL) [3, 4] was invented to suggest a different lumping strategy: lumping by functional groups. It predefines a list of functional groups, and represents a molecule using a vector of which each entry corresponds to a group and records the count of that group existing in the molecule. It assumes the functional groups are chemically independent in a molecule and ignores the connectivity between them. Its scalability can be illustrated in the same framework set by the aforementioned example: we have S pre-defined functional groups and up to $L = M/2$ carbons are functionalized. The number of distinct molecules equals to the number of distinct vectors that satisfy:

$$\sum_{i=0}^{S} v_i \leq L \tag{7.3}$$

where v_i is the i^{th} entry of the vector (length S) that represents a molecule in SOL.

The number of physically meaningful SOL vectors can be approximated by the volume of the positive hypersphere quadrant with dimension S and radius L. Assuming S is even, we have

$$N_{SOL} \approx \frac{\pi^{S/2}(L)^S}{(S/2)!} \cdot \frac{1}{2^S} \tag{7.4}$$

which can be further simplified by Stirling's approximation into:

$$N_{SOL} \approx (L\sqrt{\frac{\pi e}{2S}})^S \tag{7.5}$$

If we choose $M = 30$,$L = 15$ and $S = 8$, $N_{SOL} \approx 200$ millions, which is much less than the full-detail representations but still too large for practical simulations; From Eq. 7.5 we also see the number of species in SOL grows rapidly with the number of functional groups allowed per molecule (L, which scales with the molecular weight) as well as the number of distinct types of functional groups (S).

In this chapter, we propose a fragment-based strategy that supports mechanistic modeling for large systems with better scalability (see Figure 7.2):

- We designed a fragment-based framework with two innovative components: fragmentation of model compounds and reattachment of fragments (see details in Section 7.1), which speeds up modeling process by focusing on kinetics between key parts of large molecules instead between molecules themselves.

- We demonstrate its accuracy using a case study of pyrolysis of a C18 hydrocarbon: phenyldodecane (PDD, Section 7.2). By comparing its results against those from full-detail representation method (RMG), we get good agreements in feedstock's conversion and product molecular weight distribution (Section 7.3).

- We have built a modeling package `AutoFragmentModeling` based on this framework. The source code is made available at **`https://github.com/KEHANG/AutoFragmentModeling`**.

7.1 Fragment modeling framework

This section presents our fragment-based modeling framework. It starts with definition of fragment and fragmentization, and proceeds with a case study detailing how to generate fragment reactions, and estimate fragment-based kinetics and thermochemistry.

7.1.1 Fragment

In fragment modeling, we see a large molecule as a combination of molecular fragments. Each molecular fragment is defined as an entity which consists of multiple functional groups and preserves their original connectivity in the molecule. In that sense, a molecular fragment can be viewed as a middle layer entity between molecule and functional groups. Unlike SOL assuming little spacial interaction between functional groups, the fragment framework accounts for the interaction by accommodating related functional groups within the fragment, which may help improve model fidelity.

This concept makes a much more compact representation for feedstock than aforementioned methods. Using the previous example, suppose fragments have K carbons on average, the total number of distinct species in the fragment representation is

$$N_{fragment} = S^K \tag{7.6}$$

Since fragments are much smaller than molecules, we choose $K = 5$ with same parameter setup as previous ($S = 8$), $N_{fragment} = 33,000$. Eq. 7.6 also shows $N_{fragment}$ doesn't increase when dealing with larger molecules as long as modellers keep fragment size and functional groups fixed. Scalability comparison is made in Figure 7.2. 5-carbon fragments provide a smaller number of species than SOL for C10 or larger feeds. For heavy feeds, e.g., >C25, even an 8-carbon fragment is a more compact representation than SOL.

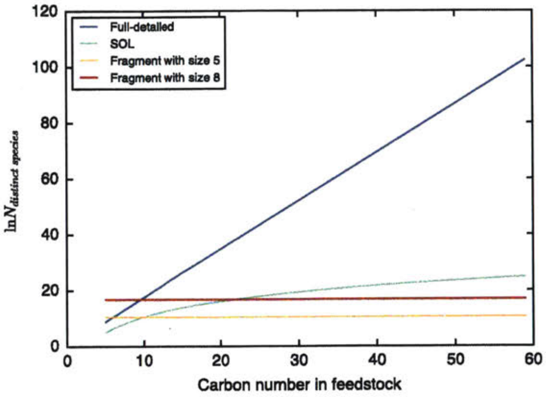


Figure 7.2: Scaling behaviors of number of distinct species with feedstock size for three different methods, assuming 8 distinct carbon arom types, and that at least half of the carbons in the large molecules are not functionalized (i.e., $L = M/2$).

7.1.2 Fragmentization

The fragmentization is a modeling step of converting feedstocks to fragments. It has three working scenarios.

- A) systems where individual molecule structures are currently not possible to characterize but molecular weight distribution can be obtained as well as concentrations of functional groups.
- B) systems where individual molecule structures can be characterized
- C) systems where modellers have selected model compounds (with known molecular structures)

In scenario A, one needs to predefine fragment types based on characterized functional groups and solve for initial fragment concentrations by minimizing the deviations between model and characterization data, which is very similar to regular feedstock reconstruction procedure. [4–6]

In scenario B and C, with original molecule structure at hand, we fragmentize the feedstock molecules with two competing considerations: making fragments small to reduce computation cost, while enlarging fragments to keep important reactivities locally. This chapter includes a case study of phenyldodecane pyrolysis, of which the fragmentation step is shown in Figure 7.3.

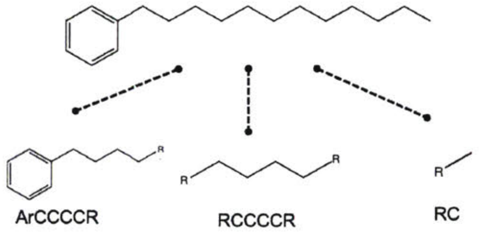


Figure 7.3: In the case study, phenyldodecane is fragmentized into three types of fragments

7.2 Case study

In this section we designed a case study to demonstrate the fragment-based modeling workflow. In addition, by comparing prediction results of a fragment-based model with those of a detailed model by RMG, we were able to assess feasibility of the proposed framework.

The target large system was purposefully chosen: low-temperature pyrolysis of phenyldodecane (thereafer PDD), which is one of the largest systems ever modeled using the full-detail representation by RMG (Chapter 6).

7.2.1 fragmentation

In a PDD molecule, the carbons on the long alkyl chain experience aromatic influence, which decreases with distance from the benzene ring. The first four carbons, α, β, γ and δ carbon, are most influenced, which leads us to create the first fragment ArCCCCR that contains these carbons shown in Figure 7.3. Second fragment RCCCCR represents the remaining alkyl carbons that behave as regular carbons on a long alkyl chain. The terminal carbon is separately represented by the third fragment RC.

7.2.2 fragment reaction generation

In order to capture the key chemistry of PDD pyrolysis, we defined four elementary reaction families (listed below) for the case study.

- bond-fission / radical-recombination, Figure 7.4
- beta-scission / multiple-bond-addition, Figure 7.5
- hydrogen-abstraction, Figure 7.6
- disproportionation, Figure 7.7

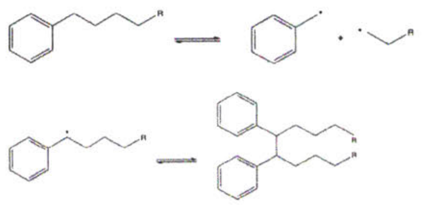


Figure 7.4: Example reactions for reaction type: bond-fission / radical-recombination

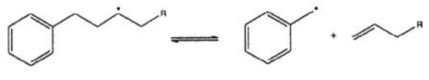


Figure 7.5: Example reaction for reaction type: β-scission / multiple-bond-addition

Cross-reaction generation currently is achieved semi-automatically via `AutoFragmentModeling`; modellers have to input a list of fragments and their reactions (see the input format below) to `AutoFragmentModeling`, which facilitates the model construction process

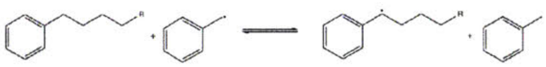


Figure 7.6: Example reaction for reaction type: hydrogen-abstraction

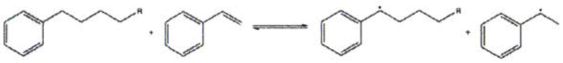


Figure 7.7: Example reaction for reaction type: disproportionation

by automatic bookkeeping, duplicate checking, thermochemistry and kinetics estimation.

In particular, `AutoFragmentModeling` has designed SMILES identifier for fragment (see example fragments defined in input file below) and implemented SMILES parser which creates fragment graph structure from text representation. It has also implemented fragment isomorphism algorithm, enabling fragment bookkeeping and duplicate identification.

```
# This is a fragment input file
# 1st part is stable fragments
# label: fragment SMILES
ArC(CCCR)CCCR: c1ccccc1C(CCCR)CCCR
ArCC: c1ccccc1CC
ArCCC: c1ccccc1CCC
H2: [H][H]
RCC: RCC
RCCC: RCCC
RCCCCC__CC: RCCCCC=CC
...
# 2nd part is radical fragments
# label: fragment SMILES
RCCC*: RCC[CH2]
ArC*: c1ccccc1[CH2]
RCC*: RC[CH2]
ArCC*: c1ccccc1C[CH2]
RC*: R[CH2]
ArCCC*: c1ccccc1CC[CH2]
...
```

On the other hand, reaction string parser has been created to help `AutoFragmentModeling` understand and process human readable fragment reaction representations (see example reactions below), which provides equivalent service for reaction bookkeeping. In addition, it also checks if a certain fragment reaction is a viable elementary step.

```
# This is a fragment reaction input file
# Beta-scissions: R_Addition_MultipleBond

ArC*CCCR == ArC__C + RCC*

ArCC*CCR == ArCC__C + RC*

# Disproportionations: Disproportionation

ArC__C + C__CC__CC == ArC*C + C__CC__CC*

ArC__C + ArCCCCR == ArC*C + ArC*CCCR
...
```

Future functionalities such as automatic fragment reaction generation and selection are desired to be added.

7.2.2.1 thermo and kinetics estimation

`AutoFragmentModeling` estimates thermodynamic properties and kinetic parameters based on first principle data to gain high transferability. In practice, it creates for query fragment a representative molecule and sends the molecule to RMG's thermochemistry estimator; the returned thermochemistry of the representative molecule will be regarded as the fragment's. A similar scheme is in place for fragment kinetic estimation.

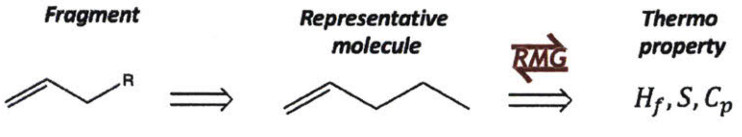


Figure 7.8: Fragment thermochemistry estimation scheme in `AutoFragmentModeling`

7.3 Results and Discussions

A final model was generated using `AutoFragmentModeling v1.0.0`. It has 76 fragment species and 528 reactions. Compared with molecule-based model (ref to PDD paper), it reduces species and reactions by 5 and 17 times, respectively.

We conduct kinetic simulations in Cantera's [7] homogeneous batch reactor module at 673 K and 350 bar using both models: one is molecule-based model by `RMG-Py v2.0.0`, the other is fragment-based model by `AutoFragmentModeling v1.0.0`.

7.3.1 feedstock conversion

The feedstock composition for both simulations is 100% PDD. Since the fragment-based model doesn't have PDD as a whole molecule representation, we set initial composition equivalently to 26.7 mol% ArCCCCR, 46.7 mol% RCCCCR, and 26.7 mol% RC. When calculating feedstock conversion, we choose ArCCCCR's conversion as an approximation for PDD's conversion since most ArCCCCR belongs to PDD especially in early conversion.

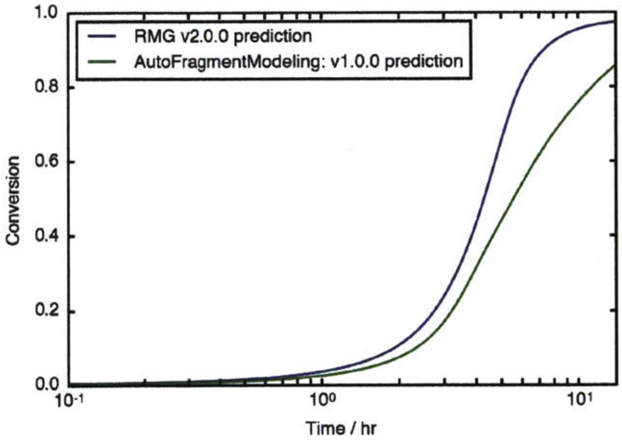


Figure 7.9: Agreement between feedstock conversions predicted by molecule-based model (RMG v2.0.0) and fragment-based model (this work)

As Figure 7.9 shows, feedstock conversions predicted by these two models agree very well with each other when conversion < 0.6, after which the two predictions start to deviate (by around 0.15 eventually). One major source causing the discrepancy is that ArCCCCR appears in many other products in late conversion of PDD, which makes it no longer an accurate proxy for PDD.

7.3.2 selected products comparison

Molar yields (defined as moles produced per initial PDD mole) of a few selected products are also compared between the two models (Figure 7.10). Toluene and ethylbenzene are the two major products from PDD pyrolysis, both of which are well captured by the fragment-based model. It also predicts heavy products, which are typically generated from radical recombination (e.g., bottom right in Figure 7.10, RArArR) or multiple bond addition followed by hydrogen abstraction (e.g., bottom left in Figure 7.10, ArC(CCCR)CCCR). RArArR from the fragment-based model is compared with c1ccccc1C(CCCCCCCCCCC)C(CCCCCCCCCCC)c1ccccc1 (SMILES) from molecule-based model, and ArC(CCCR)CCCR from the fragment-based model is compared with c1ccccc1C(CCCCCCCCCC)CCCCCCCCCCC (SMILES) from molecule-based model.

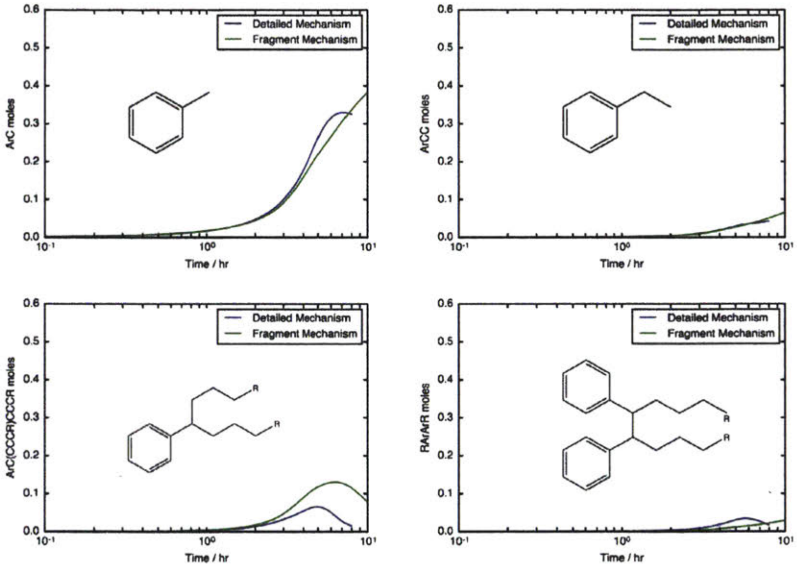


Figure 7.10: Major light products (toluene and ethylbenzene) and heavy products are predicted by molecule-based model via RMG v2.0.0 and fragment-based model via AutoFragmentModeling v1.0.0 with their molar yields compared.

7.3.3 molecular weight distribution

Sole ODE / PDE simulation in fragment space predicts fragment distribution which makes limited predictions on molecule level, as shown in Figure 7.9 and 7.10, as a fragment can appear in multiple molecules. To link fragment distribution back to molecule distribution, we designed a reattachment algorithm that merges fragments into molecules.

However, the fragmentation step has lost connectivity information between fragments in the original feedstock molecule. If allowing fragments to merge randomly, reattachment would lead to frequent nonphysical reattachment, e.g., reattachment between ArCCCCR and itself, impairing molecular prediction accuracy. Thus, we modified the fragmentation step (Figure 7.3) to retain information of fragment connectivity , as shown in Figure 7.11 by annotating fragments with pairing fragmentation labels; if two fragments are originally connected, one of them has R-label and the other L-label.

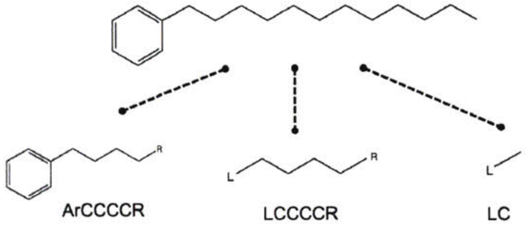


Figure 7.11: Fragmentation of PDD with two fragmentation label types: R-label and L-label

Most of the nonphysical reattachments can be easily prevented with the the new fragmentation by requiring two merging fragments to have compatible label types; reattachment is only allowed when a fragment contains R-label and the other L-label (Figure 7.12).

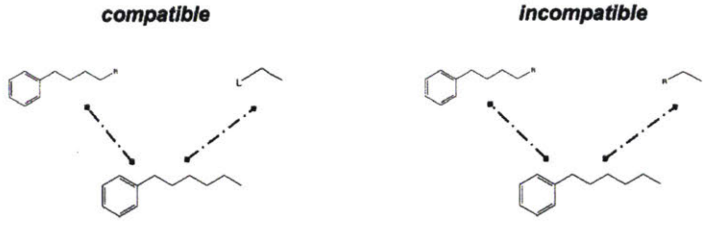


Figure 7.12: Compatible reattachment example (left panel) and incompatible reattachment example (right panel)

It should be noted that the reattachment provides one possible realization of product distribution. Statistically, the variance of all possible realizations can be reduced by introducing more pairs of fragmentation labels, as it helps retain more information of fragment connectivity in original molecules. But it also creates more distinct fragments, increasing computation workload during cross-reaction generation. To balance with model construction cost, in this case study we choose to have one pair of fragmentation labels. Its reconstruction effectiveness is shown in Figure 7.13; the fragment-based model preserves most of prediction accuracy on products' molecular weight distribution from detailed model by RMG.

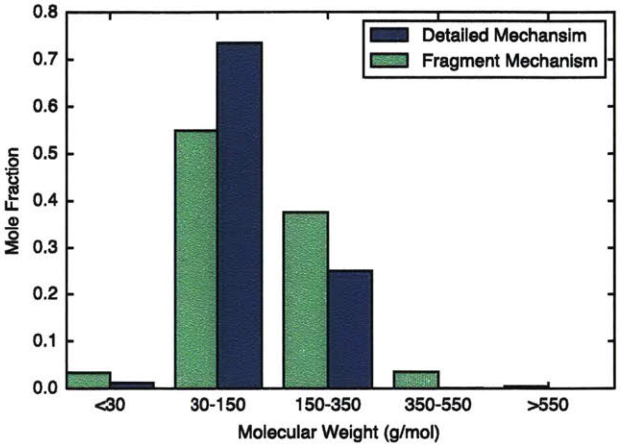


Figure 7.13: molecular weight distributions of PDD pyrolysis agree reasonably well between molecule-based model by RMG v2.0.0 and fragment-based model by AutoFragmentModeling v1.0.0.

7.4 Conclusion

In this chapter, we designed a new kinetic modeling framework with fragment concept. Two key components in the framework make it scalable to model large kinetic systems: the fragmentation of feedstock molecules reduces the number of species and reactions needed to describe the chemical systems, and the reattachment of fragments allows predictions at molecule level.

A proof-of-concept case study using C18 pyrolysis has been carried out. Compared with RMG's detailed model, the fragment-based model has 17 times fewer reactions, but gives similar predictions on feedstock conversion, major product molar yields and molecular weight distribution. This demonstrates the promising potential of the fragment-based framework in modeling large systems which RMG cannot handle well.

7.5 References

[1] *How much data is on the Internet?* `https://www.quora.com/How-much-data-is-on-the-Internet`. Accessed: 2018-04-16.

[2] L. P. d. Oliveira, D. Hudebine, D. Guillaume, and J. J. Verstraete. "A Review of Kinetic Modeling Methodologies for Complex Processes." en. *Oil & Gas Science and Technology – Revue d'IFP Energies nouvelles* 71 (3), May 2016, p. 45. ISSN: 1294-4475, 1953-8189. DOI: `10.2516/ogst/2016011`. URL: `https://ogst.ifpenergiesnouvelles.fr/articles/ogst/abs/2016/03/ogst150117/ogst150117.html`.

[3] R. J. Quann and S. B. Jaffe. "Structure-oriented lumping: describing the chemistry of complex hydrocarbon mixtures." *Industrial & Engineering Chemistry Research* 31 (11), Nov. 1992, pp. 2483–2497. ISSN: 0888-5885. DOI: `10.1021/ie00011a013`. URL: `https://doi.org/10.1021/ie00011a013`.

[4] S. B. Jaffe, H. Freund, and W. N. Olmstead. "Extension of Structure-Oriented Lumping to Vacuum Residua." *Industrial & Engineering Chemistry Research* 44 (26), Dec. 2005, pp. 9840–9852. ISSN: 0888-5885. DOI: `10.1021/ie058048e`. URL: `https://doi.org/10.1021/ie058048e`.

[5] M. Neurock, A. Nigam, D. Trauth, and M. T. Klein. "Molecular representation of complex hydrocarbon feedstocks through efficient characterization and stochastic algorithms." *Chemical Engineering Science*. Chemical Reaction Engineering: Science & Technology 49 (24, Part A), Jan. 1994, pp. 4153–4177. ISSN: 0009-2509. DOI: `10.1016/S0009-2509(05)80013-2`. URL: `http://www.sciencedirect.com/science/article/pii/S0009250905800132`.

[6] M. I. Ahmad, N. Zhang, and M. Jobson. "Molecular components-based representation of petroleum fractions." *Chemical Engineering Research and Design* 89 (4), Apr. 2011, pp. 410–420. ISSN: 0263-8762. DOI: `10.1016/j.cherd.2010.07.016`. URL: `http://www.sciencedirect.com/science/article/pii/S0263876210002145`.

[7] D. G. Goodwin, H. K. Moffat, and R. L. Speth. *Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes.* Version 2.2.1. 2016. URL: `http://www.cantera.org`.

8

Recommendations for Future Work

This thesis has made several contributions to enabling automated construction of high fidelity kinetic models for large complex chemical systems. Among them, are memory usage reduction, concurrent model construction, improved cyclic thermochemistry estimations. With these improvements integrated into RMG, a subsurface oil-to-gas application was chosen and modelled, whose success highlights RMG's modeling capability expanding toward very complex systems. Additionally, a novel modeling strategy was designed to complement RMG's detailed modeling strategy, which opens new opportunities to constructing accurate models for extremely large systems. This chapter presents several challenges that become important and possible to solve with recent advances in the automatic kinetic modeling community.

8.1 Data-driven estimation

With recent advances in machine learning and data science, many estimation tasks in RMG can obtain great benefit. One example in this thesis is the data-driven MCNN estimator for thermochemistry, which outperforms many traditional methods. During traditional estimator development process, researchers have to collect data, look for patterns, gain insights (e.g., our heuristic model) and eventutally translate them to mathematic terms (model formulation). Data-driven estimation accelerates the process by employing automatic pattern recognition and relating patterns directly to prediction targets. It has significant advantage in boosting accuracy over traditional estimations especially when insights are difficult to generate or generalize but data collection is relatively easy. In order to pursue that within RMG scope, there are at least two subjects requiring future research attention.

8.1.1 Central database

We need a central database dedicated to raw data storage to support development of future data-driven estimator. The main database in our group is `RMG-database`. However, it is not designed for training and testing purposes. Instead its data practically serves as the parameters of many RMG's estimators such as group-additivity-based estimator, library-based estimator, rate-rule-based estimator. Its text-based nature enables easy version control, but also prevents it from benefitting from recent advances in database management, e.g., quick query, easy insertion/deletion, status reporting, etc.

In this thesis, I have done some preliminary work in establishing a central database to store raw thermochemistry data via widely used database framework `MongoDB`. Each data point is associated with its meta data, e.g., level of theory, timestamp. The database is hosted by RMG server, allowing free accessibility. The standardized central database frees RMG developers from carrying raw data around. Since the data has been already cleaned, it enables us to mainly focus on creating, training, testing and comparing data-driven estimators with exactly same base.

Several aspects of the central database still need futher work:

- Easy online tools for data query, insertion, visualization. Currently a series of ipython notebooks have been in place to serve the purposes for developers. Online web tools will be more desirable for general users outside the group.

- Massive data generation. I have implmented an automatic thermochemistry calculator `autoQM` which takes a molecule identifier (e.g., SMILES), creates 3D initial geometry, launch a quantum mechanics job and convert the job result to thermodynamic properties. It currently only supports Guassian with SLURM scheduler. Future work may extend that to other quantum mechanics softwares, job schedulers as well as kinetic parameter calculations via automatic transition state search.

- Set up kinetic central database for development of future kinetic estimators. As an extension of thermo central database, kinetic central database can start by addng existing elementary reactions stored in RMG's libraries and training reaction libraries. At next stage of further enlarging reaction domain, `autoQM` kinetic jobs can be launched to supply necessary kinetic data.

8.1.2 Interpretation research

As discussed in Chapter 5, MCNN-based estimator successfully learnt patterns in large fused polycyclics. Although we have done some qulitative interpretation analysis for the learned fingerprints, more rigorous interpretation analysis can be conducted. Gomez-Bombarelli *et al.*[1] developed an auto-encoder whose decoder was able to map fingerprint space back to molecule space. It would be worthwhile pairing our MCNN estimator with an appropriately selected decoder, which can be used to visualize each learned fingerprint entry in molecule space.

8.2 Quality control

As RMG turns into a large scale scientific software, quality control becomes an increasingly important subject as any small code modification may introduce unexpected chain effect leading to execution failure or performance loss. Since RMG development follows modularized philosophy, it is straightforward to pair individual functionality with one or more unit tests. That has already widely adopted among RMG developers, which effectively prevents or early detects functionality breakdown.

For performance loss, we have built our own continuous integration test platform: `RMG-tests`. It is designed to automatically compare model construction results for a collection of applications, between previous version and pull-requested version with new changes. `RMG-tests` currently has two types of performance monitoring: estimator prediction, RMG model generation.

- Add kinetic prediction tests. For estimator performance, `RMG-tests` currently only tests thermodynamic property prediction. Kinetic performance is not monitored mainly due to lack of test data. The situation can be improved by the establishment of kinetic central database.

- For RMG model generation tests, the selection of applications should be updated with RMG development status. Currently it only reflects a fraction of chemistry RMG is able to model. On the other hand, most of the applications have little experimental data. An effort should be made to create a structured dataset containing application level data e.g., conversion and speciation. That will help the RMG team to decide if a code change is beneficial and accelerates new feature development.

8.3 Fragment modeling

This thesis presents a fragment-based modeling strategy, which has much better scalability than previous methods in dealing with extremely large chemical systems. The proof-of-concept case study demonstrates its possibility of making an accurate kinetic model. The basic framework has been implmented in `AutoFragmentModeling` package. There are several aspects listed below for future improvements.

- Automatic reaction generation for fragments should be implemented in `AutoFragmentModeling`. Ideally, the related methods in RMG could be modified to accept fragment format.
- Currently we need user inputs to divide molecules into fragments, which could be made more intelligent and automated.
- It is an undesirable behavior during model generation that the range of fragment sizes increases over time (heavier fragments get formed via recombination, smaller ones via decomposition). The heavier fragments increasing modeling complexity need to be further divided, while smaller ones need to be reattached to reclaim reactivity. Thus, integration of more frequent fragmentation and reattachment into model generation may be valuable.

8.4 References

[1] R. Gómez-Bombarelli, D. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. “Automatic chemical design using a data-driven continuous representation of molecules.” *arXiv:1610.02415 [physics]*, Oct. 2016. arXiv: 1610.02415. URL: `http://arxiv.org/abs/1610.02415` (visited on 02/28/2017).