Open Academic Journal of Advanced Science and Technology
ISSN 2577-7807
Vol. 10, No. 1, pp. 60-69.
2026
DOI: 10.33094/oajast.v10i1.2528
© 2026 by the author; licensee Online Academic Press, USA

Proposed approach to a chemical reaction that based on B+Tree

Suhad Malallah Kadhem

College of Computer Science, University of Technology, Baghdad, Iraq.
Email:
suhad.m.kadhem@uotechnology.edu.iq

Licensed:
This work is licensed under a Creative Commons Attribution 4.0 License.

Keywords:
Ambiguity problem
B+tree
Chemical reactions
Path
Unique bidirectional.

Received: *3 September 2026*
Revised: *13 October 2026*
Accepted: *3 November 2026*
Published: *10 November 2026*

Abstract

This research proposes a new methodology for organizing and retrieving chemical reactions by employing two B+ trees linked to a single chemical reaction database, aiming to achieve high efficiency in bidirectional searching and minimize data redundancy. The proposed methodology relies on using the first B+ tree to retrieve the products of the chemical reaction based on the input reactants, while the second B+ tree is used to retrieve the reactants corresponding to a specific chemical product. This structure provides unique, interconnected search paths that allow access to reaction information in both directions, while limiting the duplication of stored data. The expected outcomes include improved organization of chemical reaction databases, reduced memory requirements, and more efficient search and retrieval processes. The research concludes that the use of two B+ trees offers an effective mechanism for managing chemical reaction data, addressing both data redundancy and ambiguity during retrieval. This methodology can be applied in chemical database systems that require rapid and accurate retrieval of both reactants and products.

1. Introduction

The field of chemistry is fundamentally built upon a vast and complex network of chemical reactions. Managing and efficiently retrieving information from this extensive dataset presents a significant computational challenge, particularly in the era of high-throughput experimentation and chemical big data [1]. Traditional linear data structures often prove insufficient for these tasks, leading to performance bottlenecks as database scales increase [2].

In Gómez-Bombarelli, et al. [3] they proposes a novel methodology for organizing chemical reaction information using a B-Tree data structure. B-Trees are renowned for their efficiency in handling large-scale datasets in database systems and file management, offering robust performance for dynamic indexing operations. Alatas [4] achieved improved time complexity for essential operations such as search, insertion, and deletion by structuring a comprehensive set of reactions—encompassing de-composition, synthesis, single displacement, combustion, and double displacement —within a B-Tree.

In van Wijngaarden [5] they present a hybrid approach for computational completion of the equations of reaction. In particular, they combine a rule-based approach and a machine learning (ML) model for completing reactions. The rule-based approach constructs a balance of atoms and charge on either side of the reaction in an attempt to find missing molecules. They tailor the pre-trained transformer model on the chemical language domain to take partial re actions as inputs and predict missing molecules. Furthermore, They present a novel approach to measure the correctness of our model, which is useful when we apply it to the uncurated dataset and the ground-truth is unknown.

In Chen and Baldi [6] novel rule extensions have been proposed for enabling robust predictions and describing detailed mechanisms of reaction at electron flows' level in steps of elementary reaction, which ensures that all of the reactions are balanced properly then atom-mapped. The functionalities of core reaction prediction of this expert system have been depicted with applications that include: (i) prediction of detailed reaction mechanisms; (ii) retrosynthetic analyses; (iii) computer-based learning in organic chemistry; (iv) combinatorial library design.

The application of efficient data structures is particularly relevant for modern chemoinformatics and the development of automated chemical design platforms, which rely on rapid access to structured reaction data [7, 8]. This approach promises to enhance the performance of chemical databases and reaction retrieval systems, facilitating more efficient data management and knowledge extraction in both academic and industrial contexts [9, 10].

© 2026 by the author; licensee Online Academic Press, USA

In this research we will propose a new chemical reactions approach that provide an efficient search time and efficient memory usage by using two b+tree that index on the same chemical reactions database.

2. Chemical Reactions Categories

Because there are many chemical reactions taking place in chemical laboratories and industrial processes and life, it is therefore necessary to classify these reactions according to how they occur to the following classes:

2.1. Reactions of Synthesis

It is a chemical process of union between 2 or more reactants for producing one compound or product. In generic terms, Synthesis reactions appear as the examples below:

A + B → AB (General Example)

Fe + S → FeS

2.2. Decomposition Reactions

Which is the opposite of the process of the Synthesis and includes the fragmentation or disintegration of a particular compound to simpler compounds or elements, A decomposition reaction needs a source of energy, like light, heat, or electricity, to occur. In generic terms, Synthesis reactions can be represented as the following examples:

AB →A + B (General Example)

$CaCO_3 \xrightarrow{\Delta} CaO + CO_2 \uparrow$

2.3. Single Displacement Reactions

It replaces or displaces another element in a compound, such as reaction of zinc with diluted sulfuric acid, where zinc substitutes the hydrogen inside the acid structure as shown in the following examples:

A +BC → AC + B (General Example)

$Zn + H_2SO_4 \rightarrow ZnSO_4\downarrow + H_2 \uparrow$

2.4. Double Displacement Reactions

Which happen in the case where ions and cations of two separate molecules switch positions, forming 2 compounds that are completely different as shown in the following examples:

AB +CD → AD + BC

$AgNO_3 + KCl \rightarrow AgCl\downarrow + KNO_3$

2.5. Combustion Reactions

Is a reaction of a particular material with oxygen that releases large amounts of energy in a form of heat or light.

$C_3H_8 + 5O_2 \rightarrow 3\ CO_2 \uparrow + 4\ H_2O$

B + Tree [11, 12]

B + Tree is a type of tree or data structure That allowing a range of movable child node numbers that can be changed over time. In which every path from root to leaf is of identical length, and every non leaf node has between (n/2) and (n) children , where n is fixed for a tree . it consists of index pages and data pages.

B + tree is referred to as an index where all the records shall be indexed in the database and the key of such a record shall be displayed in B+ tree. In the case of requesting a given record, we therefore have to know its key so as to retrieve it out of B+ tree. Once we have the key of the record we can directly access any required record. B + tree is an ordered tree (Figure 1), which is simple to keep in balance, and the time used to search through it is less as compared to the majority of other types of trees.

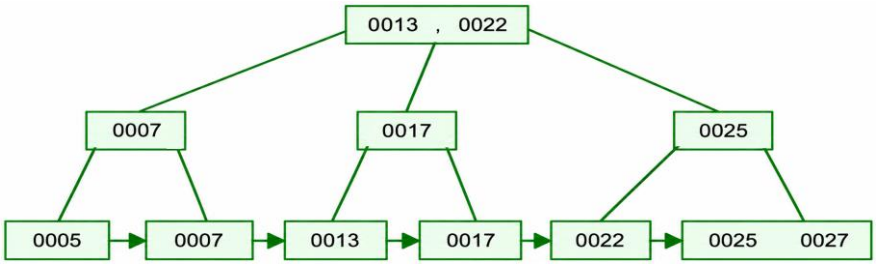


Figure 1. A B+ tree example.

© 2026 by the author; licensee Online Academic Press, USA

3. Description of the Proposed Method

In this paper we suggest a new mechanism for dealing with chemical reactions by using B+ tree to provide efficient access and take less time in processing also chemical reactions is stored in such a way that prevents redundancy of those reactions for providing efficient usage of memory.

The proposed method uses two binary plus trees (bpt1 and bpt2) that index on the same chemical reactants database (dbase). bpt1 is used for efficient access for the product of a certain chemical reaction by using the first reactant (highest priority) as a key. bpt2 is used for efficient access for the reactants of a certain chemical reaction by using the product as a key. In general if the chemical reaction is: Reactant1+Reactant2+…+Reactant$_n$ → Product Then it will be represented as shown in Figure 2.

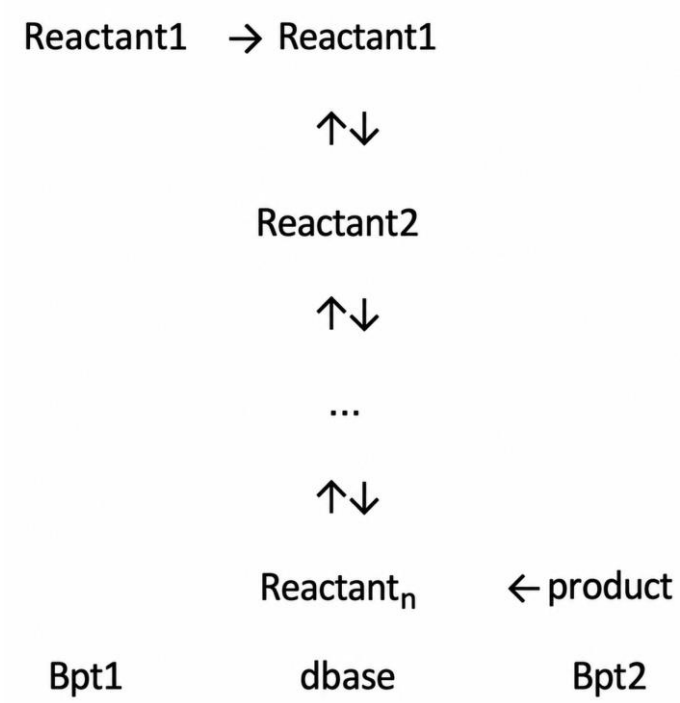


Figure 2. Two b+trees index on the same chemical reactions database.

The arrows in Figure 2 represent the database pointers such that we follow the down arrow (↓) to get the products for the reactants, and we follow the above arrow (↑) to retrieve the reactants that correspond to a certain product.

4. Algorithms of the Proposed Method

The proposed method for chemical reactions includes three chooses: (see Algorithm 1)

A. Storing a new chemical reaction.
B. Finding the product(s) for a chemical reaction when we have its reactants.
C. Finding the reactants for a chemical reaction when we have its product.

If the user wants to store a new chemical reaction in dbase, then the new chemical reaction is converted to two lists (reactants list and products list) these two lists will be sorted according to the priority of their elements. Then we must check if the first reactant (highest priority key) of the chemical reaction is not found in bpt1 (new reaction), then we will store it in bpt1 as key and store a record for each reactant in dbase and put the first product (with highest priority) as key in bpt2. (see Algorithm 2)

If the key is found in bpt1 then we have one of the following states:

1.All the reactants of the chemical reaction are found in dbase, and in this case we have:

- The reactants already have a product and this mean the chemical reaction is already stored.
- If the product of The reactants is not stored, then use the first product (with highest priority) as key in bpt2 with a pointer to a new record (that will contain the last reactant record reference) if it is not found in bpt2 or put this reference to the exists one that can be used for retrieved process .

2.All the reactants of the reaction are found in dbase except the last reactant, and in this case we store the last reactant in dbase with a pointer to the previous reactant record , then use the first product as key in bpt2.

3. if not all reactant is found in dbase then store a record for each remaining reactant with its next reactant (except the last reactant) in dbase such that each reactant is pointed to the next and previous reactant record. The last reactant record in dbase will contains a pointer to the previous reactant record, number of reactants and the sorted products list.

© 2026 by the author; licensee Online Academic Press, USA

To get the chemical reaction products we follow the pointer of the next reactant for n times (where n the number of reactant), when we reach to the last reactant record we retrieve its product list and balance the chemical reaction before producing to the user (see Algorithm 3).

On the other side if the user wants to know the reactants of a certain product we user the product as a key for bpt2 and return a record that contains all the references of chemical reactions that produce that product, and we use each reference to return the last reactant record that contains the number of reactants (n) and another details of that reaction, then we will follow the pointer of the previous reactant for n times, and when we collect all the reactants we balance the chemical reaction before producing to the user (see Algorithm 4).

Algorithm 1: *chemical reactions*

Input: user's choose

Process:

Step1: if the user chooses "adding new chemical reaction" then call Algorithm 2 after checking his authentication.

Step2: if the user chooses "finding the products for a chemical reaction" then call Algorithm 3.

Step3: if the user chooses "finding the reactants of a product for a chemical reaction" then call Algorithm 4.

End.

Algorithm 2: *adding new chemical reactions*

Input: List of reactants(L1), List of products(L2).

Process:

Step1: Sort L1 according to the reactants priority and sort L2 according to the products priority. /* the reactant priority will be computed according to the priority of each element of the reactant*/

Step2: Let N be the number of reactants.

Step3: If the reactions database (dbase) exist then open it with its two binary plus trees (bpt1 and bpt2) otherwise create them.

Step4: Search for K (first reactant of L1), if not found then

4.1: Put K as a key in the first binary plus tree (bpt1).

4.2: Store a record for the K (in reactions database) with the next reactant and a pointer to the next reactant's record.

4.3: Store a record for each remaining reactant of L1 with its next reactant (except the last reactant) in dbase such that each reactant is pointed to the next and previous reactant record.

4.4: Store the last reactant of L1 in reactions database with a pointer to the previous reactant record, N and L2.

4.5: Call Algorithm 5. /* take L2 and the pointer of the last reactant as input */

else

5: Repeat

5.1: Retrieve the record of the current reactant from dbase

5.2: Search for next reactant in the retrieved record

5.3: Follow the pointer of the next reactant record

Until the next reactant is not found or there is no next reactant

6: If all the reactants are found in dbase then

6.1: Check if the product L1 is already stored then write "this reaction is already found " and exit.

Else

6.2: Call Algorithm 5 and exit.

7: If the next reactant is not found then

7.1: Store a record for each remaining reactant of L1 with its next reactant (except the last reactant) in dbase such that each reactant is pointed to the next and previous reactant record.

7.2: Store the last reactant of L1 in dbase with a pointer to the previous reactant record, N and L2.

7.3: Call Algorithm 5.

End.

Algorithm 3: *finding the products of a chemical reaction*

Input: List of reactants (L1)

Output: List of products (L2)

Process:

Step1: Sort L1 according to the reactants priority.

Step2: Let N be the number of reactants.

Step3: Search in bpt1 for the first reactant

3.1: If not found then write "not found" and exit.

Else

3,2: i=1 , flag=true

3.3: while i< N and flag do

3.3.1: Retrieve the record of L1[i] from dbase

© 2026 by the author; licensee Online Academic Press, USA

3.3.2: search for next reactant in the retrieved record
3.3.3: if found then follow the pointer of the next reactant (L1⌈i+1⌉)
i=i+1
Else flag=false
End while
3.4: if not flag then write "not found" and exit.
3.5: Retrieve the record of L1⌈N⌉
3.6: Retrieve the list of products L3 from the term that has the same previous pointer.
3.7: Balance the chemical reaction (L1 and L3) to get balanced products list L2.
3.8: Return (L2).
End.
Algorithm 4: *Finding the reactants of a chemical reaction*
Input: product (P)
Output: list of reactants (L1)
Process:
Step1: search in bpt2 for P.
Step2: if not found then write "not found" and exit.
Else
Step3: Retrieve the record of this product to get a list of pointers (L) for all chemical reaction that this product comes from.
Step4: For each pointer in L do
4.1: Retrieve the record (R) that correspond to this pointer
4.2: Search for P in each list of product (L3) found in R to get the number of reactants (N)
4.3: i=1
4.4: while i< N do
4.4.1: Put this reactant in L2⌈i⌉
4.4.2: Follow the pointer of the previous reactant
4.4.3: i=i+1
End while
4.5: Put this reactant in L2 ⌈1⌉
4.6: Balance the chemical reaction (take L2 and L3) to get balanced reactants list L1.
4.7: Return (L1).
End.
Algorithm 5: *store product*
Input: list of products (L2), pointer of the last reactant record (P)
Process:
Step1: While L2 is not empty do
Step2: Remove product Pr from L2.
Step3: If Pr is found in bpt2 then retrieve its record R and store P in R.
else put Pr as a key in bpt2 with a pointer to a new record NR and store P in NR.
End while
End.

5. Implementation for the Proposed Method

We have several examples to describe the proposed method.
Example1:

If we want to get the product for a chemical reaction from reactions database (dbase) and suppose it is a new reaction, such as: "H2 + C → CH4", after that, we put the first reactant of the reaction as key in first binary plus tree (bpt1) , and we calculate the reactants' length, we use CH4 as a key for second binary plus tree (bpt2) as well. The reaction will be stored in reaction database as shown in Figure 3.

© 2026 by the author; licensee Online Academic Press, USA

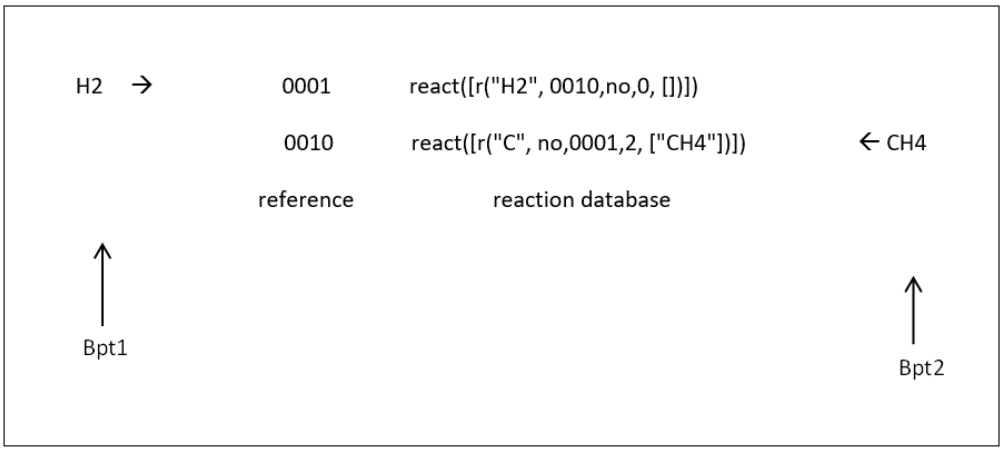


Figure 3. H2 + C → CH4 reaction is represented using the proposed method.

Example 2:

If we want to store a new chemical reaction such that some of its reactants are already found in dbase, like: "H2 + C + N2 → 2HCN", then we will store the remaining reactants in dbase only, and it will be as shown in Figure 4.

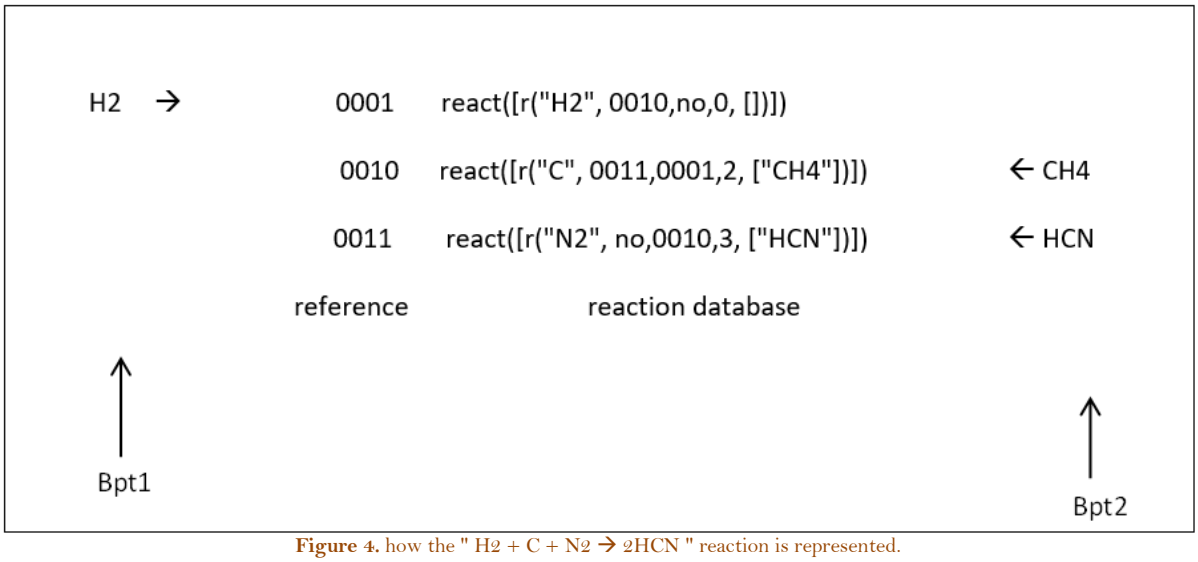


Figure 4. how the " H2 + C + N2 → 2HCN " reaction is represented.

Example 3:

If we want to get the product that its reactants is already found in reaction database but with no product, such as: "O2→ O3", then we store the product, and store it in second binary plus tree as a key, it will be as shown in Figure 5.

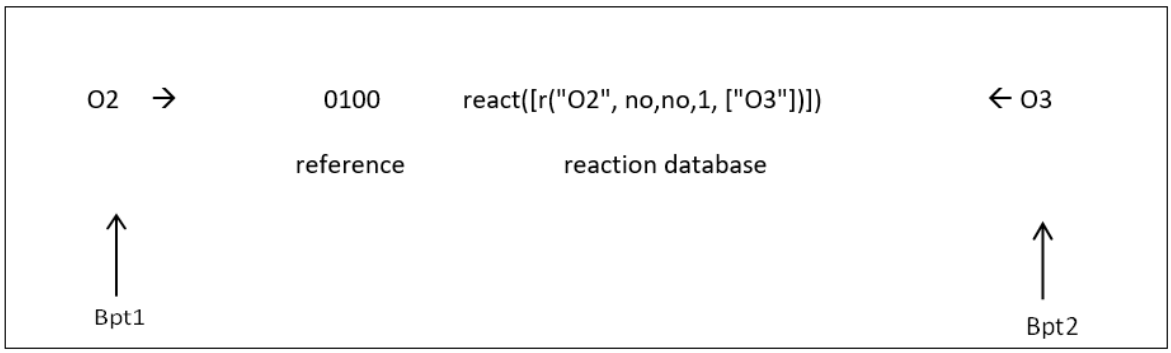


Figure 5. How the " O2 → O3 " reaction is represented.

© 2026 by the author; licensee Online Academic Press, USA

Example 4:

If we want to get the product for a reactants, such that, some of its middle reactants are found in reaction database, such as : "Fe+ O2⟶ Fe2O3" then only the not found reactants will be store in reaction database, and we will store the first reactant in first binary plus tree , and store the new product ("Fe2O3") in second binary plus tree, it will be as shown in Figure 6.

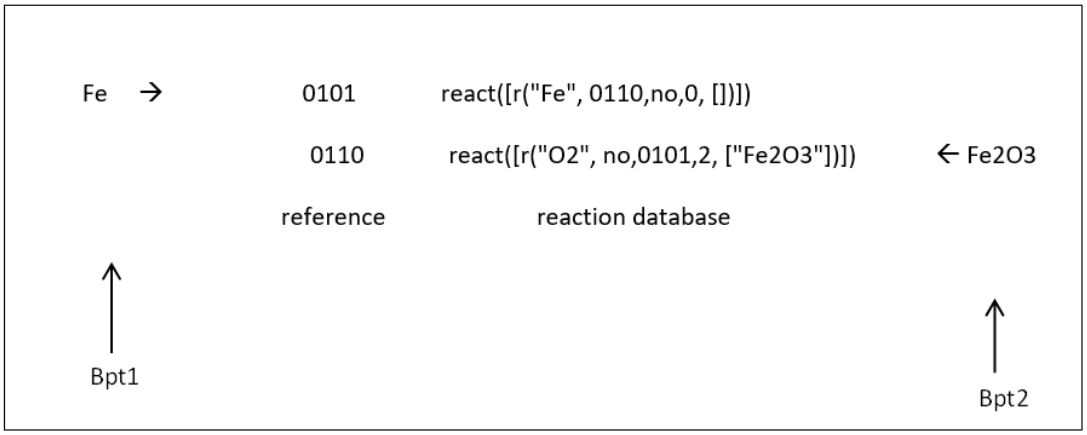


Figure 6. how the " Fe+ O2⟶ Fe2O3" reaction is represented.

6. Discussion

Some methods use production rules to build a chemical reaction system. Table 1 will compare these methods with our proposed method.

Table 1. Comparison between the suggested method and other rule base method.

Rule based methods	Proposed method
Increase the number of chemical reactions mean increase in program size	Increase the number of chemical reactions has no effect because B+ tree dealing with huge data
Search time for gating the products of a chemical reaction will be increased when chemical reactions are increased	Provide an efficient search time for gating the products of a chemical reactions or finding the reactants of a certain product
Redundant is exist since the reactants is restored at each reaction.	No redundant of chemical reaction or sub chemical reaction
Chemical reactions are represented as rules to find the chemical reaction products, and finding the reactants need additional rules.	Work in both sides (finding the products or finding the reactants) using the same representation

7. Conclusion

In the present study the following points can be concluded:

- Using B+ tree for building a chemical reaction system will provide a sufficient search time for gating products of a chemical reaction and for retrieving the reactants that correspond to a certain product.
- When we use B+ tree there is no ambiguity in finding products of a chemical reaction or in retrieving reactants of a particular product since we follow a unique bidirectional path for each chemical reaction.
- Using the proposed method to represent the chemical reactions will provide efficient memory usage since there is no redundant in chemical reactions or even sub of them.
- Using the same chemical reaction representation for gating the products (by follow the next reactant reference) or retrieving the reactants (by follow the previous reactant reference) provides an efficient chemical reactions system.

References

[1] A. M. Żurański, J. I. Martinez Alvarado, and A. G. Doyle, "A database of historically-observed chemical reactions," *Scientific Data*, vol. 8, no. 1, p. 293, 2021.

[2] Z. Luo and M. J. Carey, "Efficient data ingestion and query processing for LSM-based storage systems," *Proceedings of the VLDB Endowment*, vol. 12, no. 5, pp. 531–543, 2019.

[3] R. Gómez-Bombarelli *et al.*, "Automatic chemical design using a data-driven continuous representation of molecules," *ACS Central Science*, vol. 4, no. 2, pp. 268–276, 2018. https://doi.org/10.1021/acscentsci.7b00572

[4] B. Alatas, "ACROA: Artificial chemical reaction optimization algorithm for global optimization," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13170-13180, 2011. https://doi.org/10.1016/j.eswa.2011.04.126

© 2026 by the author; licensee Online Academic Press, USA

[5] M. van Wijngaarden, "Chemical reaction completion: A hybrid rule-based and language model-based approach," M.S. Thesis, Delft University of Technology, Delft, The Netherlands, 2023.
[6] J. H. Chen and P. Baldi, "No electron left behind: A rule-based expert system to predict chemical reactions and reaction mechanisms," *Journal of Chemical Information and Modeling*, vol. 49, no. 9, pp. 2034–2043, 2009. https://doi.org/10.1021/ci900157k
[7] S. C. Nayak, B. B. Misra, and H. S. Behera, "Hybridzing chemical reaction optimization and artificial neural network for stock future index forecasting," presented at the 2013 1st International Conference on Emerging Trends and Applications in Computer Science (ICETACS). IEEE, 2013, 130–134.
[8] A. Faulconbridge, "RBN-World: Sub-symbolic artificial chemistry for artificial life," Ph.D. Dissertation, University of York, York, U.K, 2011.
[9] A. Y. S. Lam and V. O. K. Li, "Chemical-reaction-inspired metaheuristic for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 381-399, 2010. https://doi.org/10.1109/TEVC.2009.2033580
[10] H. Li, Q. Li, and L. Zhang, "Multiple-replica cloud data integrity verification scheme based on B+ tree," presented at the 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2017, 2279–2282.
[11] H. C. V. Ngu and J.-H. Huh, "B+-tree construction on massive data with Hadoop," *Cluster Computing*, vol. 22, no. Suppl 1, pp. 1011-1021, 2019. https://doi.org/10.1007/s10586-017-1183-y
[12] S. M. Kadhem, "Using B+ tree to represent secret messages for steganography purpose," *Engineering and Technology Journal*, vol. 28, no. 15, pp. 5102–5112, 2010.

Appendix

$3O_2 \rightarrow 2O_3$
$CH_4(g) + 2O_2(g) \longrightarrow CO_2(g) + 2\ H_2O(g)$
$3\ Fe(s) + 4H_2O(g) \rightarrow Fe_3O_4(s) + 4\ H_2(g)$
$BaCl_2(aq) + Na_2CrO_4(aq) \longrightarrow BaCrO_4(s) + 2\ NaCl(aq)$
$2\ HgO(s) \dashrightarrow 2\ Hg(l) + O_2(g)$
$C_2H_4(g) + H_2(g) \rightarrow C_2H_6(g)$
$Na_2O(s) + H_2O(l) \longrightarrow 2NaOH$
$CS_2(l) + 3O_2(g) \longrightarrow CO_2(g) + 2\ SO_2(g)$
$Al_2(SO_4)_3(aq) + 3Ca(OH)_2(aq) \longrightarrow 2Al(OH)_3(s) + 3CaSO_4(s)$
$H_2(g) + Cl_2(g) \longrightarrow 2\ HCl(g)$
$NaCl(aq) + AgNO_3(aq) \longrightarrow NaNO_3(aq) + AgCl(s)$
$Zn(s) + 2HCl(aq) \longrightarrow ZnCl_2(aq) + H_2(g)$
$Al_4C_3(s) + 12H_2O(l) \longrightarrow 3CH_4(g) + 4Al(OH)_3(s)$
$2\ Mg(s) + O_2(g) \longrightarrow 2MgO(s)$
$16\ Rb(s) + S_8(s) \longrightarrow 8Rb_2S(s)$
$8Ba(s) + S_8(s) \longrightarrow 8\ BaS(s)$
$4Fe(s) + 3O_2(g) \longrightarrow 2\ Fe_2O_3(s)$
$2\ Na(s) + Cl_2(g) \longrightarrow 2\ NaCl(s)$
$S_8(s) + 8O_2(g) \longrightarrow 8SO_2(g)$
$C(s) + O_2(g) \longrightarrow CO_2(g)$
$2K(s) + I_2(g) \longrightarrow 2KI(s)$
$Mg(s) + F_2(g) \longrightarrow MgF_2(s)$
$2\ Co(s) + 3\ F_2(g) \longrightarrow 2\ CoF_3(s)$
$Sr(s) + Br_2(l) \longrightarrow SrBr_2(s)$
$2Na(s) + F_2(g) \longrightarrow 2NaF(s)$
$U(s) + 3F_2(g) \longrightarrow UF_6(g)$
$CaO(s) + H_2O(l) \longrightarrow Ca(OH)_2(s)$
$2\ H_2SO_3(aq) + O_2(g) \longrightarrow 2\ H_2SO_4(aq)$
$Ca(OH) \rightarrow 2(s)\ CaO(s) + H_2O(g)$
$SO_2(g) + H_2O(l) \longrightarrow H_2SO_3(aq)$
$CaO(s) + SO_2(g) \longrightarrow CaSO_3(s)$
$CaCO_3(s)\ CaO(s) + CO_2(g)$
$2\ KClO_3(s) \rightarrow 2KCl(s) + 3\ O_2(g)$
$Cl_2(g) + 2KBr(aq) \longrightarrow 2\ KCl(aq) + Br_2(l)$
$H_2CO_3(aq) \longrightarrow CO_2(g) + H_2O(l)$
$2\ Na(s) + 2\ H_2O(l) \longrightarrow 2\ NaOH(aq) + H_2(g)$
$H_2SO_4 \rightarrow (aq)\ SO_3(g) + H_2O(l)$
$2Al(s) + 3Pb(NO_3)_2(aq) \longrightarrow 3Pb(s) + 2Al(NO_3)_3(aq)$
$Mg(s) + 2HCl(aq) \longrightarrow H_2(g) + MgCl_2(aq)$

© 2026 by the author; licensee Online Academic Press, USA

$F_2(g) + 2NaCl(aq) \longrightarrow 2\ NaF(aq) + Cl_2(g)$
$Br_2(l) + KCl(aq) \longrightarrow$ no reaction
$HCl(aq) + NaOH(aq) \longrightarrow NaCl(aq) + H_2O(l)$
$2KI(aq) + Pb(NO_3)_2(aq) \longrightarrow PbI_2(s) + 2KNO_3(aq)$
$FeS(s) + 2HCl(aq) \longrightarrow H_2S(g) + FeCl_2(aq)$
$C_3H_8(g) + 5\ O_2(g) \longrightarrow 3CO_2(g) + 4H_2O(g)$
$2Li(s) + 2H_2O(l\ ) \longrightarrow 2LiOH(aq) + H_2(g)$
$N_2(g) + 3\ H_2(g) \longrightarrow 2\ NH_3(g)$
$2\ C_6H_{14}(l) + 19\ O_2(g) \longrightarrow 12CO_2(g) + 14\ H_2O(l\ )$
$2\ NaNO_3(s) \longrightarrow 2\ NaNO_2(s) + O_2(g)$
$2\ ZnS(s) + 3O_2(g) \longrightarrow 2ZnO(s) + 2\ SO_2(g)$
$CaH_2(s) + 2H_2O(l) \longrightarrow Ca(OH)_2(aq) + 2\ H_2(g)$
$AgNO_3(aq) + KI(aq) \longrightarrow AgI(s) + KNO_3(aq)$
$Li + O_2 \longrightarrow LiO_2$
$H_2 + Cl_2 \longrightarrow HCl$
$Al + Fe_2O_3 \longrightarrow Al_2O_3 + Fe$
$Pb(CH_3COO)_2 + H_2S \longrightarrow PbS + CH_3COOH$
$H_2 + Cl_2 \longrightarrow H_2Cl_2$
$NaI + Cl_2 \longrightarrow NaCl + I$
$CaCO_3 \rightarrow CaO!+!CO_2$
$2NH_3 + H_2SO_4 \rightarrow (NH_4)_2SO_4$
$MgCO_3 \longrightarrow MgO_2 + CO_2$
$3CuSO_4 + 2Fe \rightarrow Fe_2(SO_4)_3 + 3Cu$
$Li_3N + 3NH_4NO_3 \rightarrow 3LiNO_3 + (NH_4)_3N$
$NaCl(aq) + AgNO_3(aq)! \rightarrow AgCl(s) + !NaNO_3(aq)$
$2C_2H_6 + 7O_2 \rightarrow 4CO_2 + 6H_2O$
$S + O_2 \rightarrow SO_2$
$Al_4C_3 + H_2O \rightarrow Al(OH)_3 + CH_4$
$Al(ClO_3)_3 + MgCrO_4 \rightarrow Al_2(CrO_4)_3 + Mg(ClO_3)_2$
$C_2H_4 + O_2 \rightarrow CO_2 + H_2O$
$P_2O_5 + H_2O \rightarrow H_3PO_4$
$Na_2SiO_3 + HF \rightarrow H_2SiO_3 + NaF$
$NH_4NO_3 \rightarrow N_2 + O_2 + H_2O$
$Na_2O + H_2O \rightarrow NaOH$
$NaHCO_3 + H_3C_6H_5O_7 \rightarrow CO_2 + H_2O + Na_3C_6H_5O_7$
$C_3H_5N_3O_9 \rightarrow CO_2 + N_2 + O_2 + H_2O$
$KClO_3 \rightarrow KCl + O_2$
$Zn + H_2SO_4 \longrightarrow ZnSO_4 + H_2$
$N_2\ (g) + 3H_2\ (g) \longrightarrow 2\ NH_3\ (g$
$H_2SO_4\ (aq) + NaOH\ (aq) \longrightarrow Na_2SO_4\ (aq) + H_2O\ (l)$
$Al\ (s) + HCl\ (aq) \longrightarrow AlCl_3\ (aq) + H_2\ (g)$
$CaO\ (s) + H_2O\ (l) \longrightarrow Ca(OH)_2$
$Ca(OH)_2\ (aq) + CO_2\ (g) \longrightarrow CaCO_3\ (s) + H_2O\ (l)$
$C_6H_{12}O_6\ (aq) + 6O_2\ (aq) \longrightarrow 6CO_2\ (aq) + energy$
$CaCO_3\ (s) \longrightarrow CaO\ (s) + CO_2\ (g)$
$2\ FeSO_4\ (s) \longrightarrow Fe_2O_3\ (s) + SO_2\ (g) + SO_3\ (g)$
$2Pb(NO_3)_2\ (s)\ heat \longrightarrow 2\ PbO\ (s) + 4NO_2\ (g) + O_2\ (g)$
$Zn\ (s) + CuSO_4\ (aq) \longrightarrow ZnSO_4\ (aq) + Cu\ (s)$
$Fe\ (s) + CuSO_4\ (aq) \longrightarrow FeSO_4\ (aq) + Cu\ (s)$
$Na_2SO_4\ (aq) + BaCl_2\ (aq) \longrightarrow BaSO_4\ (s)\downarrow + 2NaCl\ (aq)$
$Pb\ (s) + CuCl_2\ (aq) \longrightarrow PbCl_2\ (aq) + Cu\ (s)$
$2\ Cu\ (s) + O_2\ (g)\ heat \longrightarrow 2CuO\ (s)$
$CuO\ (s) + H_2\ (g) \longrightarrow Cu\ (s) + H_2O\ (l)$
$ZnO\ (s) + C\ (s)\ heat \longrightarrow Zn\ (s) + CO\ (g)$
$MnO_2\ (s) + 4HCl\ (aq) \longrightarrow MnCl_2\ (aq) + 2H_2O\ (l) + Cl_2\ (g)$
$Cu\ (s) + I_2\ (s)\ heat \longrightarrow CuI_2\ (s)$
$Zn\ (s) + CuSO_4\ (aq) \longrightarrow ZnSO_4\ (aq) + Cu\ (s)$ or

© 2026 by the author; licensee Online Academic Press, USA

$Fe\ (s) + S\ (s)\ heat \longrightarrow FeS\ (s)$
$Fe_2O_3\ (s) + 2Al\ (s) \longrightarrow Al_2O_3\ (s) + 2\ Fe\ (s)$
$Zn\ (s) + Cu^{2+}\ (aq) \longrightarrow Zn^{2+}\ (aq) + Cu\ (s)$
$2\ Mg\ (s) + O_2\ (g) \longrightarrow 2\ MgO\ (s)$
$2Na\ (s) + Cl_2\ (g) \longrightarrow 2NaCl\ (s)$
$AgNO_3\ (aq) + HCl\ (aq) \longrightarrow AgCl\ (s) + HNO_3\ (aq)$
$4\ Na\ (s) + O_2\ (g) \longrightarrow 2\ Na_2O\ (s)$
$MnO_2\ (s) + 4HCl\ (aq) \longrightarrow MnCl_2\ (aq) + 2H_2O\ (l) + Cl_2\ (g)$
$H_2\ (g) + Cl_2\ (g) \longrightarrow 2HCl\ (g)$
$Zn\ (s) + 2AgNO_3\ (aq) \longrightarrow Zn(NO_3)_2\ (aq) + 2\ Ag\ (s)$
$H_2\ (g) + CuO\ (s) \longrightarrow Cu\ (s) + H_2O\ (l)$
$Ca(OH)_2 + HNO_3 \longrightarrow Ca(NO_3)_2 + H_2O$
$CuSO_4\ (aq) + Zn\ (s) \longrightarrow ZnSO_4\ (aq) + Cu\ (s)$
$BaCl_2\ (aq) + H_2SO_4\ (aq) \longrightarrow BaSO_4\ (s) + HCl\ (aq)$
$BaCl_2\ (aq) + Al_2(SO_4)_3\ (aq) \longrightarrow AlCl_3\ (aq) + BaSO_4\ (s)$
$H_2S\ (g) + SO_2\ (g) \longrightarrow S\ (s) + H_2O\ (l)$
$Pb\ (NO_3)_2\ (aq) + Fe_2(SO_4)\ (aq) \longrightarrow Fe(NO_3)_3\ (aq) + PbSO_4\ (s)$
$2\ KNO_3\ (s)\ heat \longrightarrow 2\ KNO_2 + O_2\ (g)$
$Zn\ (s) + 2\ AgNO_3\ (aq) \longrightarrow Zn(NO_3)_2 + 2\ Ag\ (s)$
$Ni\ (NO_3)_2\ (aq) + 2\ NaOH\ (aq) \longrightarrow Ni(OH)_2\ (s) + 2\ NaNO_3\ (aq)$
$MgO\ (s) + C\ (s) \longrightarrow CO\ (g) + Mg\ (s)$
$2\ KClO_3\ (s)\ heat \longrightarrow 2\ KCl\ (s) + 3O_2\ (g)$
$3MnO_2\ (s) + 4\ Al\ (s)\ heat \longrightarrow 3\ Mn\ (l) + 2Al_2O_3\ (s)$
$Ca\ (s) + Cl_2\ (g)\ heat \longrightarrow CaCl_2\ (s)$
$2\ HgO\ (s) \longrightarrow 2Hg\ (l) + O_2$
$H_2SO_4\ (aq) + 2\ NaOH\ (aq) \longrightarrow Na_2SO_4\ (aq) + 2\ H_2O\ (l)$
$Fe_2O_3\ (s) + 3\ CO\ (g)\ heat \longrightarrow 2Fe\ (s) + 3\ CO_2\ (g)$
$Zn(s) + 2\ HCl\ (aq) \longrightarrow ZnCl_2(aq) + H_2(g) + H_2$
$2\ Al\ (s) + 6\ HCl\ (aq) \longrightarrow 2\ AlCl_3\ (aq) + 3\ H_2\ (g)$
$H_2 + F_2 \boxed{\leftrightarrow} 2HF$

© 2026 by the author; licensee Online Academic Press, USA