Optimizing Chemical Reaction Yields Using The Branch and Bound Algorithm

Naomi Risaka Sitorus - 13523122
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: naomi.risaka@gmail.com, 13523122@std.stei.itb.ac.id

***Abstract*—Chemicals are used in nearly every aspect of modern life. However, chemical synthetization remains complex and resource-intensive, as a compound can often be synthesized through multiple pathways, each with varying time and resource costs. Experimentally testing each possibility is impractical; therefore, certain compounds remain difficult to produce. This study explores the application of the Branch and Bound algorithm to optimize chemical reaction yields. Each node in the state-space tree represents a partial solution, where decisions have been made for a subset of available reactions, and each edge corresponds to a specific decision regarding reaction execution. A bounding function estimates the upper bound of the yield that can be produced based on current decisions. Time constraints, available compounds, and stoichiometric balance act as limiting factors alongside the bounding function to enable pruning. The developed program processes the available chemical reactions and compounds, ranks reactions based on their effectiveness, and selects the most optimal sequence to maximize yield according to the defined parameters. This study offers valuable insights into reaction planning and optimization using the Branch and Bound algorithm. By automating the selection of reaction pathways, the program reduces the time, effort, and expertise typically required to determine the most efficient synthesis route.**

Keywords—chemical reaction optimization; yield maximation; chemical sythesis; reaction planning; Branch and Bound algorithm.

I. Introduction

Chemicals are currently used in nearly every aspect of modern life, from medicine and agriculture to materials, manufacturing, and energy. Despite their widespread use, the process of producing chemicals remains complex and resource-intensive. In many cases, the same compound can be produced through multiple reaction paths, each with its own trade-offs in time, resources, and yield. However, choosing the best combination of reactions is often still based on trial-and-error or inherited habits, rather than systematic analysis.

As there are usually many ways to synthesize a compound, with each reaction consumes time and inputs, finding the optimal sequence of reactions is not trivial. Researchers often face a combinatorial explosion of possible choices, and testing them all experimentally is impractical. As a result, some valuable compounds remain difficult to produce efficiently. This is not due to chemical impossibility, but due to the difficulty of identifying the best synthesis path under specific conditions such as limited available compounds or time.

This paper explores how the Branch and Bound algorithm can be applied to optimize the yield of a specific target compound given a set of reaction rules, starting compounds, and a fixed time budget. By modeling each state of the system as a node in a decision tree, and each reaction as a branch, the algorithm can efficiently prune unpromising paths and focus only on those likely to increase the final yield. This approach can help automate and optimize the planning of chemical production processes, reduce wasted resources, and importantly ease the synthetization of currently hard-to-make chemicals by revealing reaction paths that may not be obvious through manual planning alone.

II. Theoretical Basis

A. Branch and Bound Algorithm

Branch and Bound is an algorithm designed for solving combinatorial optimization problems, where the objective is to maximize or minimize a given function subject to a set of constraints. The algorithm systematically explores the solution space using a state-space tree, with a worst-case time complexity of $O(b^d)$, where b denotes the branching factor (i.e., the number of sub problems generated at each level), and d represents the depth of the tree. The exploration strategy employed is similar to Depth-First Search (DFS), but it incorporates a least-cost search mechanism. This mechanism estimates the minimum possible cost required to reach the goal from a given node, thereby guiding the search towards more promising solutions. Nodes that cannot possibly yield a better solution than the best one found so far are pruned, significantly reducing the number of nodes that must be evaluated [1].

A key component of the Branch and Bound algorithm is the bounding function. This function provides an optimistic estimate of the best possible solution that can be achieved from a particular node. If this estimate is worse than the current best solution, the node and all its descendants are excluded from further consideration. Pruning is also performed when a node violates a constraint or when no feasible choices remain and the node's solution does not improve the current optimum. This

selective pruning contributes to the efficiency of the algorithm by avoiding the evaluation of suboptimal or infeasible paths.

An example of a problem solvable by Branch and Bound is the Integer Knapsack Problem, also known as the 0/1 Knapsack Problem. Given a set of *n* items, each with a specific weight and profit, along with a knapsack that has a maximum weight capacity *K*, the objective is to select a subset of items such that the total profit is maximized without exceeding the weight limit as shown in (1).

$$\sum_{i=1}^{n} w_i x_i \leq K \, , x_i \in \{0,1\}, i = 0,1,2,\dots,n \qquad (1)$$

In the context of Branch and Bound, each node in the search tree represents a decision—whether to include or exclude a specific item. Items are often sorted in descending order based on their profit-to-weight ratio to improve efficiency [2]. The bounding function for this problem calculates an upper bound on the total achievable profit from a node. It does so by summing the accumulated profit at the current node with the estimated maximum additional profit that could be obtained by greedily filling the remaining knapsack capacity with the next items, including fractional values solely for estimation purposes, as expressed in (2). This upper bound helps determine whether it is worthwhile to continue exploring a particular path.

$$\hat{c} = F + (K - W)p_{i+1}/w_{i+1} \qquad (2)$$

An example of a Branch and Bound state-space tree used for solving the Integer Knapsack problem is shown in Fig. 2.1. The tree demonstrates how the algorithm systematically explores different combinations of items while pruning branches that cannot lead to optimal solutions. Each node shows the current weight *W*, accumulated profit *F*, and bound value. The red 'B' letters indicate pruned branches where the bound is worse than the current best solution.

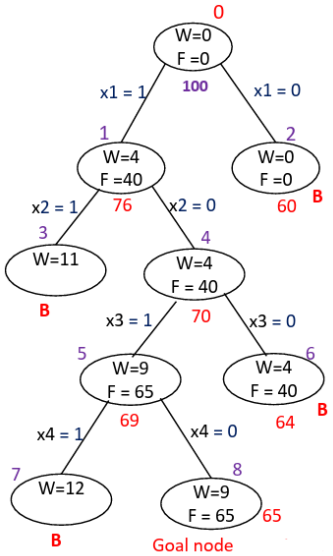

Fig. 2.1. Branch and Bound Tree for Solving An Integer Knapsack Problem (Source: [2])

B. *Stoichiometry*

Stoichiometry is a foundational concept in chemistry that focuses on the quantitative relationships between substances involved in a chemical reaction. It provides a mathematical framework to determine how much of each reactant is required to produce a desired amount of product, based on a balanced chemical equation. The stoichiometric relationships are derived from the Law of Conservation of Mass, which states that matter is neither created nor destroyed in a chemical reaction. As a result, the total mass of the reactants must equal the total mass of the products [3].

A balanced chemical equation is essential in stoichiometry, as it defines the molar ratios between reactants and products. These ratios are determined by the stoichiometric coefficients, which are the numbers placed in front of chemical formulas in a reaction. These coefficients indicate the relative number of moles of each substance involved.

Central to stoichiometry is the concept of mole (mol), which is the SI unit for measuring the amount of a substance. One mole corresponds to 6.022×10^{23} entities, known as Avogadro's number, which may be atoms, molecules, or ions [3]. The molar mass or molecular weight *M* of a compound, expressed in grams per mole (g/mol), is calculated by summing the atomic masses of all the atoms in the molecule, based on the periodic table. The conversion between mass and moles can be seen in (3), where *n* is the number of moles and *m* is the mass of the substance.

$$n = \frac{m}{M} \qquad (3)$$

In reactions involving more than one reactant, it is common for one of the reactants to be consumed completely before the others. This substance is referred to as the limiting reactant, as it determines the maximum possible amount of product that can be formed. Any other reactants that remain after the reaction has completed are considered to be in excess.

The theoretical yield refers to the maximum amount of product that can be generated from the limiting reactant under ideal conditions. In practice, the actual yield is often lower due to inefficiencies or side reactions. The effectiveness of a reaction is measured using the percent yield, defined as the percentage of the actual yield compared to the theoretical yield.

C. *Chemical Reactions*

A chemical reaction is a process where one or more substances, referred to as reactants, are transformed into different substances known as products. This transformation occurs through the rearrangement of atoms, involving the breaking of existing chemical bonds and the formation of new ones. Observable phenomena such as color changes, temperature shifts, gas release, and the formation of precipitates often accompany these changes, indicating that a chemical transformation has taken place [4].

Chemical reactions are formally represented using balanced chemical equations, which define the stoichiometric relationships between the participating substances. These equations ensure compliance with the Law of Conservation of Mass. The stoichiometric coefficients in the equation indicate

the exact molar ratios in which reactants combine and products form. Under ideal conditions, such equations allow chemists to predict the theoretical yield of products from given amounts of reactants with precision.

However, in practical applications, chemical reactions do not occur in isolation. They are influenced by a variety of kinetic and thermodynamic factors. Kinetic factors include the reaction rate, which defines how quickly reactants are converted into products. Variables such as temperature, pressure, concentration of reactants, surface area, and the presence of catalysts play crucial roles in determining the rate at which a reaction proceeds. Thermodynamic factors, on the other hand, manage the feasibility and equilibrium position of the reaction, determining how far the reaction will proceed and the maximum yield that can theoretically be achieved [4].

Another critical aspect of real-world reactions is the concept of reaction time, which is the duration required for a chemical transformation to reach completion or equilibrium under specific conditions. Reaction time can range from seconds to several hours depending on the reactants and the environment. In batch processing systems, commonly used in laboratory and industrial synthesis, it is useful to define reaction time as a residence time per batch. A batch in this context refers to one complete execution of a chemical reaction according to its stoichiometric coefficients. The total reaction time is modeled to scale approximately linearly with the number of batches. For example, executing the reaction twice would take roughly twice the time. While this linearity serves as a practical approximation, actual reaction dynamics may introduce minor deviations from perfect linearity due to intermediate product accumulation, diffusion limits, and heat transfer [4].

III. Methodology

The chemical reaction yield optimization problem is modeled as a combinatorial optimization problem solved using the Branch and Bound algorithm, where the goal is to select an optimal subset of available chemical reactions to maximize the production of a target compound within given constraints. Each state in the search space is characterized by the total yield of the target product in moles, a set of reactions already executed, the remaining available reactants and their quantities, and the total time used in the reaction sequence. The objective function aims to maximize the total yield of a specified target product while adhering to constraints. The optimization is subject to several critical constraints, including time limitations that reflect real-world batch processing scenarios, available reactant quantities that must be conserved according to mass balance principles, and stoichiometric constraints that ensure reactions maintain chemical equilibrium according to balanced equations.

The implementation of the source code in this study is conducted using the Python programming language, as its comprehensive built-in libraries significantly simplify the development process while maintaining computational efficiency. Python's heapq library enables efficient priority queue operations with logarithmic time complexity, which are essential for implementing the Branch and Bound algorithm. The priority queue ensures that nodes with the highest potential for improvement are explored first, thereby enhancing the algorithm's pruning effectiveness. Additionally, dictionaries are used to represent chemical compounds along with their corresponding molar quantities, providing constant-time lookup operations that are crucial for real-time stoichiometric calculations. Python's robust string manipulation capabilities, including its regular expression (regex) library, are extensively leveraged to extract detailed information from chemical equations and analyze compound formulas, including complex molecular structures with nested parentheses and multi-element compositions.

The Branch and Bound algorithm is designed to systematically explore the solution space while eliminating unpromising branches to improve computational efficiency compared to exhaustive enumeration approaches like the brute-force algorithm. Each node in the state-space tree represents a partial solution where decisions have been made for a subset of available reactions, while each edge represents the specific decision taken regarding reaction execution. The algorithm maintains a structured approach to decision-making by considering both the immediate impact of each reaction and its long-term implications for the overall optimization objective. The following data structure is used to represent a node in the search tree:

```
class Node:
    level: int
    total_yield: float
    total_time: float
    available_moles: Dict
    bound: float           # upper bound estimate
    taken: List[bool]
```

This node representation enables efficient state tracking and facilitates the comparison of different solution paths during the search process. The level attribute indicates the depth of the decision tree, while the taken list maintains a complete history of decisions made. This allows for solution reconstruction once the optimal path is identified.

The input required for this program contains information about the chemical system under optimization, including the chemical reactions available for use along with their per-batch reaction time measured in minutes, the inventory of available compounds with their initial quantities, and optimization parameters such as the maximum total reaction time and the specific target compound whose yield is to be maximized. Input can be provided either via a command-line interface (CLI) that or using a structured text file with the .txt extension for batch processing scenarios. The file-based input method requires adherence to a specific format to ensure proper parsing and data integrity. The following format is used for file-based input:

```
REACTIONS:
reactants -> products | reaction time per batch

COMPOUNDS:
compound amount unit

PARAMETERS:
MAX_TIME: time in minutes
TARGET: compound
```

Chemical equations are parsed using regular expressions to accurately extract reactants and products along with their

stoichiometric coefficients, handling complex molecular formulas that may include nested structures and multi-element compounds. The parsing algorithm is designed to handle compounds containing parentheses by implementing a depth-tracking mechanism that calculates the nesting level of each opening parenthesis '(', which must be properly closed with a matching closing parenthesis ')'. This approach ensures that complex molecular structures, such as $Ca(OH)_2$ or $Al_2(SO_4)_3$ are correctly interpreted according to their chemical composition. Multi-element compounds are connected using an underscore '_' to maintain consistency in naming conventions and avoid ambiguity in compound identification. The following regex patterns are used to parse chemical compounds:

```
# Pattern for parsing singular elements
r'([A-Z][a-z]?)(\d*)'

# Pattern for parsing more complex compounds
r'(\d*\.?\d*)\s*([A-Za-z0-9()_]+)'
```

Furthermore, the separation of compounds within reactions is performed using the plus symbol '+' as a delimiter. On the other hand, the distinction between reactants and products in a chemical equation is based on the arrow symbol '->' which serves as the reaction direction indicator. To ensure data integrity and prevent computational errors, all reaction times are validated to confirm that they are non-negative values.

Chemical formulas are internally represented using dictionary data structures, where each compound serves as a unique key mapped to its corresponding molar quantity as the associated value. This representation enables fast and efficient lookup operations and easy manipulation of compound quantities throughout the algorithm's execution. These dictionaries play a critical role in determining the feasibility of reactions by checking reactant availability, computing intermediate yields during multi-step synthesis pathways, and calculating final product yields. The dictionary-based approach also facilitates the implementation of mass balance constraints, ensuring that the total mass of reactants equals the total mass of products in accordance with the Law of Conservation of Mass.

In addition to accepting compound quantities directly in moles, the program incorporates functionality to process inputs provided in mass units such as grams or kilograms; therefore, enhancing its practical applicability in real-world scenarios where laboratory measurements are typically performed using mass-based units. This conversion capability is achieved by calculating the molar mass of each compound based on its elemental composition, utilizing a comprehensive built-in database of atomic masses. For any unknown or synthetic elements not present in the database, the user is prompted to manually input the molar mass, ensuring that the system can handle specialized compounds that may be relevant to specific research applications. This molar mass information is then used to convert mass-based inputs into molar quantities following established stoichiometric principles. It is important to note that the program does not perform automatic chemical equation balancing, so all inputted reactions are assumed to be already balanced, and users are responsible for ensuring the chemical validity of their input equations.

A priority queue is utilized to manage the exploration order of nodes in the search space, implementing a traversal strategy that prioritizes the most promising solution paths. Each node represents a partial solution and is systematically evaluated based on an upper bound estimate of the achievable yield, calculated using bounding functions that consider both current state and future potential. The node with the highest bound is consistently prioritized for exploration, ensuring that the algorithm focuses computational resources on the branches most likely to yield optimal solutions. This approach significantly reduces the search space compared to breadth-first or depth-first strategies that do not consider solution quality during traversal.

The bounding function calculates the upper bound estimate by performing a forward simulation that considers the application of all remaining reactions that can potentially be performed with the available resources and time constraints. Before evaluating each reaction, a feasibility checking function verifies that every required reactant is present in the dictionary of available moles and that the available quantity meets or exceeds the stoichiometric requirement for at least one complete reaction cycle. For each feasible reaction in the remaining set, the function identifies the limiting reactant that will be exhausted first, calculates the maximum number of applicable reaction batches based on this constraint, updates the projected total time and available compound quantities accordingly, and accumulates the potential yield contribution. If the total projected time remains within the specified time constraint, the potential yield from this forward simulation is added to the current accumulated yield to form the upper bound estimate. Otherwise, when time constraints would be violated, the bound is conservatively set to the current yield, indicating that no further improvements are feasible from this state.

The Branch and Bound algorithm explores the decision tree by systematically making binary choices at each node: to apply the current reaction under consideration ($x_i = 1$) or to skip it entirely ($x_i = 0$). When choosing to apply the reaction, the algorithm calculates the maximum reaction extent based on available reactants, computes the corresponding increase in total reaction time and yield improvement for the target product, and updates the molar quantities of all affected compounds. A new node is then created to reflect this updated state, and its upper bound is computed to estimate its potential for further improvement.

If this new yield exceeds the best yield discovered so far during the search process, the algorithm updates and stores it as the current best solution, maintaining a record of the optimal reaction sequence. The new node is added to the priority queue only if its computed bound suggests that it could potentially lead to a solution superior to the current best, implementing an aggressive pruning strategy that eliminates unpromising branches early. Conversely, when the algorithm chooses to skip the current reaction, it creates a new node that is identical to the current state but marks the specific reaction as unused in the decision history. The bound for this alternative node is also calculated, and if the bound indicates promising potential, the node is queued for subsequent exploration, ensuring that all viable solution paths are considered.

During the search process, nodes are retrieved from the priority queue and for each node extracted from the queue, a comprehensive pruning step is applied to eliminate branches that cannot possibly lead to improved solutions. If the node's upper bound is less than or equal to the current best-known yield, it is immediately discarded, as it cannot possibly lead to a better solution regardless of the subsequent decisions made. Furthermore, if the node corresponds to the last level in the reaction list—indicating that decisions have already been made for all available reactions—it is also skipped since no further improvements are possible. Nodes that violate either time constraints or material availability constraints are immediately eliminated to prevent the exploration of infeasible solution paths. Additional pruning occurs when the remaining time is insufficient to complete any of the remaining reactions, effectively terminating branches that cannot contribute to the optimization objective.

This pruning strategy plays a crucial role in reducing the combinatorial explosion typically encountered in brute-force approaches, which would otherwise require the exploration of all possible reaction combinations, resulting in exponential time complexity of $O(2^n)$ where n represents the number of available reactions. By systematically discarding unpromising branches based on mathematical bounds and constraint violations, the algorithm narrows the search space and focuses its effort exclusively on paths with potential to improve the current solution. This targeted approach makes the optimization process significantly more efficient and practical for real-world applications involving complex chemical systems with numerous reaction pathways. The algorithm's efficiency is further enhanced by the ordering of reactions based on their efficiency ratios, which considers both yield potential and time requirements, ensuring that high-impact reactions are considered early in the search process.

The algorithm terminates when one of several stopping conditions is met: the priority queue becomes empty—indicating that all potentially beneficial branches have been explored—or all remaining nodes have upper bounds inferior to the current best solution, confirming that the optimal solution has been identified. Upon termination, the algorithm reconstructs the optimal reaction sequence from the decision history maintained in the best solution node, providing a complete specification of which reactions should be executed to achieve maximum yield. The reconstruction process traces back through the boolean decision array to identify the selected reactions, calculates their optimal execution sequence based on reactant dependencies and time constraints, and validates the final solution against all specified constraints to ensure feasibility.

The solution is displayed to the terminal with detailed information regarding the optimization results, including the maximum yield of the target product expressed in moles, the total time consumed for the complete reaction sequence, the efficiency metric calculated as the rate of product creation in moles per minute, the number of nodes explored during the search process as an indicator of computational efficiency, the time utilization percentage relative to the maximum allowed time, and the algorithm processing time. An option to save the complete solution to a text file, which includes all relevant parameters, constraints, and results, is offered to the user, enabling the preservation of results for future reference.

The complete source code for this study is available at: https://github.com/naomirisaka/Branch-and-Bound-Chemical-Yield-Optimizer

IV. Results and Analysis

Testing was conducted on the source code to evaluate its functionality and generate results. These results were then analyzed to provide insights into the application of the Branch and Bound algorithm for optimizing chemical reaction yields. The test case comprises 16 chemical reactions involving simple and complex compounds. There are 14 available compounds with varying units such as kilograms, grams, and moles. In this test case, the objective is to maximize NaCl production within a 200-minute time limit using the available reactions. The test case specifications are shown in Fig. 4.1 and Fig. 4.2.

```
REACTIONS:
2 NaCl + H2SO4 -> Na2SO4 + 2 HCl | 8.0
NaCl + AgNO3 -> AgCl + NaNO3 | 7.5
Na2SO4 + BaCl2 -> BaSO4 + 2 NaCl | 8.2
CaCl2 + Na2CO3 -> CaCO3 + 2 NaCl | 7.8
2 NaCl + Ca(OH)2 -> CaCl2 + 2 NaOH | 8.5
NaOH + HCl -> NaCl + H2O | 7.2
2 NaOH + H2SO4 -> Na2SO4 + 2 H2O | 8.3
3 NaOH + AlCl3 -> Al(OH)3 + 3 NaCl | 7.9
2 NaCl + MgSO4 -> Na2SO4 + MgCl2 | 8.1
MgCl2 + 2 NaOH -> Mg(OH)2 + 2 NaCl | 7.6
Na2CO3 + 2 HCl -> 2 NaCl + H2O + CO2 | 7.4
NaHCO3 + HCl -> NaCl + H2O + CO2 | 7.3
2 NaHCO3 -> Na2CO3 + H2O + CO2 | 8.4
Na2CO3 + CaCl2 -> CaCO3 + 2 NaCl | 7.7
NaCl + KNO3 -> KCl + NaNO3 | 8.6
2 NaCl + ZnSO4 -> Na2SO4 + ZnCl2 | 8.0
```

Fig. 4.1. Available Chemical Reactions (Source: Author)

```
COMPOUNDS:
NaCl 100 g
H2SO4 0.5 kg
AgNO3 250 g
BaCl2 300 g
Na2CO3 400 g
Ca(OH)2 350 g
HCl 200 g
AlCl3 250 g
MgSO4 20 mol
NaOH 150 g
NaHCO3 450 g
CaCl2 280 g
KNO3 220 g
ZnSO4 320 g

PARAMETERS:
MAX_TIME: 200.0
TARGET: NaCl
```

Fig. 4.2. Available Compounds and Optimization Parameters (Source: Author)

Based on the test case, the program converts the available quantity of each compound to moles using atomic mass data

from the periodic table. The molar quantities of compounds serve as constraints in the optimization process. This conversion is essential for stoichiometric calculations, as chemical reactions are governed by molar ratios rather than mass ratios. The calculated compound quantities are illustrated in Fig. 4.3.

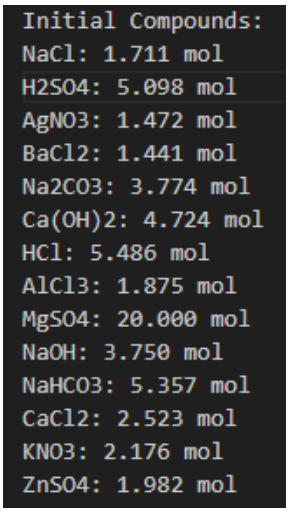

Fig. 4.3. Available Compound Quantities in Moles
(Source: Author)

Subsequently, the efficiency of each reaction is calculated based on the potential yield per unit time. These calculations guide the search tree exploration according to the Branch and Bound algorithm, where reactions with higher efficiency ratios are prioritized during the branching process. Each reaction is evaluated for its maximum feasible extent based on the limiting reactant principle, ensuring optimal utilization of available compounds. The optimal solution identified by the algorithm is presented in Fig. 4.4.

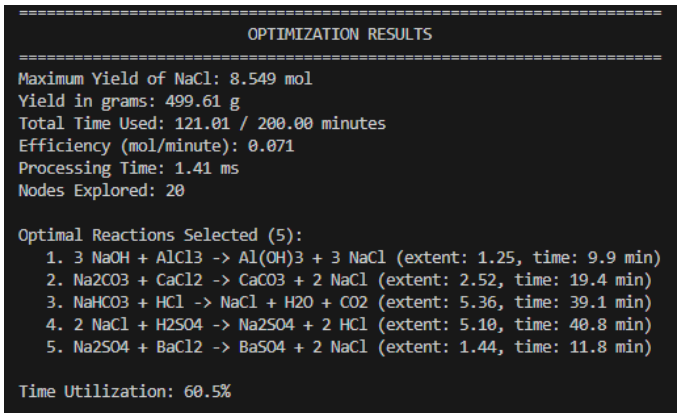

Fig. 4.4. Chemical Yield Optimization Result Using The Algorithm
(Source: Author)

The algorithm selected five specific reactions from the available set, each executed with different extents and time requirements. The Branch and Bound algorithm prioritized reactions based on their efficiency while considering stoichiometric constraints and available reactants. Each selected reaction operates at its maximum feasible extent, determined by the limiting reactant availability. The relatively low time utilization indicates that the optimization is constrained by reactant availability rather than time limits. Although time remains available for additional reactions, insufficient reactant quantities prevent further synthesis, which is the primary limiting factor in this test case.

The Branch and Bound algorithm demonstrated excellent computational efficiency by exploring only 20 nodes out of potentially 2^{16} or 65,536 possible combinations, completing the optimization in 1.41 milliseconds. This performance proves substantially more efficient than alternative approaches. A brute force approach would require exploring all 65,536 possible reaction combinations, potentially taking exponentially longer execution time and overwhelming computational resources. The algorithm's effective pruning through bound calculations significantly reduces the search space while maintaining optimality.

The results demonstrate several important practical implications for chemical synthesis optimization, where chemical synthesis is often limited by reactant availability rather than processing time. The millisecond-level execution time enables real-time optimization in automated chemical synthesis systems. The algorithm's performance scales well with the problem size, demonstrating potential for larger industrial applications with hundreds of potential reactions.

Additional test cases were conducted with varying parameters to validate the algorithm's robustness, including different numbers of available reactions, tighter time constraints, and alternative target products. These tests, documented in the source code repository, confirm the algorithm's reliability across diverse chemical synthesis scenarios and demonstrate consistent optimal performance. The Branch and Bound algorithm successfully provides an efficient solution for chemical reaction yield optimization while maintaining computational feasibility for real-world applications.

V. Conclusion

This study successfully demonstrates the use of the Branch and Bound algorithm to optimize chemical reaction yields. The developed program converts all compound quantities into moles, ranks the available reactions based on their effectiveness, and selects the most optimal sequence while considering time constraints, compound availability, and stoichiometric relationships. Testing has shown that the algorithm scales efficiently with increasing numbers of reactions, enabling practical performance even in larger problem spaces.

Future development of the program could include support for additional units of measurement, such as volume, and integration of external factors that influence reaction rates and yields—such as temperature, pressure, or catalysts—for improved accuracy. Incorporating economic factors, like the cost of compounds, would also make the optimization more applicable to real-world industrial scenarios, where both yield and cost-efficiency are critical.

Ultimately, applying the Branch and Bound algorithm in this context offers valuable insights into reaction planning and optimization. By automating the selection of reaction pathways, the program reduces the time, effort, and domain expertise typically required to identify the most efficient synthesis route.

This simplification is particularly beneficial for producing compounds that are currently difficult or expensive to synthesize. The findings from this study support the advancement of safer, more efficient, and more accessible chemical synthesis practices, with potential applications across a wide range of chemical and pharmaceutical industries.

VIDEO LINK AT YOUTUBE

https://youtu.be/brrNlVvfYwQ?si=dpG-TmlDuJJJZsQF

ACKNOWLEDGMENT

The author would like to begin by expressing sincere gratitude to God Almighty for the strength, guidance, and perseverance that made the completion of this study possible. The author is also profoundly thankful to Dr. Ir. Rinaldi Munir, M.T., whose expertise and mentorship in Algorithm Strategies provided essential direction and inspiration throughout the development of this project. Appreciation is also extended to the author's family and friends, whose constant support, motivation, and encouragement served as a source of strength during the research and writing process.

REFERENCES

[1] R. Munir, "*Algoritma Branch and Bound (Bagian 1)"*, IF2211 Strategi Algoritma, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/17-Algoritma-Branch-and-Bound-(2025)-Bagian1.pdf. [Accessed: June 21, 2025].

[2] R. Munir, "*Algoritma Branch and Bound (Bagian 4)"*, IF2211 Strategi Algoritma, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/20-Algoritma-Branch-and-Bound-(2025)-Bagian4.pdf. [Accessed: June 21, 2025].

[3] University of North Georgia, "Chapter 4 Stoichiometry of Chemical Reactions", University of North Georgia. [Online]. Available: https://web.ung.edu/media/chemistry/Chapter4/Chapter4-StoichiometryOfChemicalReactions.pdf. [Accessed: June 22, 2025].

[4] NCERT, "Chemical Kinetics", NCERT. [Online]. Available: https://ncert.nic.in/textbook/pdf/lech103.pdf. [Accessed: June 22, 2025].

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Juni 2025

Naomi Risaka Sitorus – 13523122