Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

KASDI Merbah University - Ouargla
Faculty of New Technologies of Information and Communication
Department of Computer Science and Information Technology

ACADEMIC Thesis

Presented for the graduation degree in Computer Science

Domain: Computer Science and Information technology
Branch: Computer science
Specialty: Information System

Incompatible products management

Presented by:

DJEZZAR Moncef
SADOUDI Abdessamad
AMRI Taki

Supervised by:

Dr. Ouafa CHAMA

Academic year: 2022/2023

ACKNOWLEDGMENTS

We would like to begin by expressing our gratitude to the Almighty God, Allah, for granting us the strength, patience, and courage to pursue our studies and complete this work successfully. We also sincerely thank our supervisor, Dr. Ouafa CHAMA, for her invaluable guidance, insightful advice, and unwavering support throughout the year.

Furthermore, we would like to express our heartfelt gratitude to the members of the jury for demonstrating their keen interest in evaluating and enriching our work with their insightful criticisms and constructive proposals. Additionally, we would like to acknowledge all those who have directly or indirectly contributed to the successful completion of this work.

Finally, we would like to express our deep appreciation to our family for their unwavering support and unconditional love throughout our studies. This work is the culmination of a challenging journey spanning three years, which required extensive efforts, perseverance, and continuous learning.

ABSTRACT

Graph coloring is a powerful mathematical technique applied in various fields, including computer science and operations research. Graph coloring algorithms have been used to solve complex problems by assigning colors or labels to different elements or objects. In the context of storing products, graph coloring algorithms can be used to build a program that minimizes the risk of accidents and ensures personal safety.

This research aims to explore the potential of these algorithms in the field of storing products. Our program developed using Welsh Powell algorithm would analyze the hazardous properties of different products and determine the best way to store them in order to minimize the risk of accidents. This research aims to find an effective solution to stock products problem while ensuring personnel safety by avoiding hazards.

KEYWORDS: Graph coloring, chemical products, minimize risk, Welsh Powell Algorithm.

ملخص:

تلوين الرسم البياني هو تقنية رياضية قوية تستخدم في مختلف المجالات، بما في ذلك علوم الكمبيوتر وبحوث العمليات. يتم استخدام خوارزميات تلوين الرسم البياني لحل المشكلات المعقدة التي تتضمن تعيين ألوان أو تسميات لعناصر أو كائنات مختلفة. فلتخزين المنتجات، يمكن استخدام خوارزميات تلوين الرسم البياني لبناء برنامج يقلل من مخاطر الحوادث ويضمن السلامة الشخصية.

يهدف هذا البحث إلى اكتشاف إمكانيات هاته الخوارزميات في مجال تخزين المنتجات. يتيح برنامجنا الذي قمنا بتطويره باستخدام خوارزمية Welsh Powell إمكانية تحليل الخصائص الخطرة للمنتجات المختلفة وتحديد أفضل طريقة لتخزينها من أجل تقليل مخاطر الحوادث. يهدف هذا البحث الى إيجاد حل فعال لمشكلة تخزين المنتجات مع ضمان سلامة الموظفين من خلال تجنب المخاطر.

الكلمات المفتاحية: تلوين الرسوم البيانية، المنتجات الكيميائية، تقليل المخاطر، خوارزمية Welsh Powell.

RÉSUMÉ

La coloration des graphes est une technique mathématique puissante appliquée dans divers domaines, y compris l'informatique et la recherche opérationnelle. Les algorithmes de coloration de graphes ont été utilisés pour résoudre des problèmes complexes en attribuant des couleurs ou d'étiquettes à différents éléments ou objets. Dans le contexte du stockage de produits, des algorithmes de coloration de graphes peuvent être utilisés pour construire un programme qui minimise les risques d'accidents et assure la sécurité personnelle.

Cette recherche vise à explorer le potentiel de ces algorithmes dans le domaine du stockage de produits. Notre programme développé à l'aide de l'algorithme Welsh Powell permet d'analyser les propriétés dangereuses de différents produits et de déterminer la meilleure façon de les entreposer afin de minimiser les risques d'accidents. Cette recherche vise à trouver une solution efficace au problème du stockage des produits tout en assurant la sécurité personnelle en évitant les dangers.

MOTS CLÉS : Coloration de graphes, produits chimiques, minimiser le risque, algorithme de Welsh Powell.

Table of Contents

Chapter 3: Optimizing Management with Graph Coloring

Chapter 4: Software Design and Development

List of Figures

LIST OF TABLES

General Introduction

In one hand, the safe storage and transportation of hazardous materials is a critical concern in many industries. The risk of accidents and injuries to personnel can be significant if products are not handled and stored properly. On the other hand, graph coloring algorithms have been used in various fields to assign colors or labels to different elements or objects to classify them. Research has shown that they can also be effective in storing products, but there are not enough applications in this field.

Our research aims to explore the potential of graph coloring algorithms in product stocking. We have developed a program that can determine the best storage of hazardous products based on their properties. Specifically, we will use the Welsh Powell algorithm as the core graph coloring algorithm in our application.

Our question is, "How can graph coloring algorithms be used to minimize the risk of accidents and ensure the personnel safety in product stocking?" To answer this question, we will provide a comprehensive solution to manage hazardous materials effectively, in other words, minimize the risk of accidents and ensure the safety of practitioners and policymakers use our program.

This work is composed of four main chapters. The first chapter discusses the types of chemical products, their hazards, safe storage and transportation practices. In the second chapter, we will cover essential concepts in graph theory, such as graph representation, bases concepts, and graph representation in machine. Chapter three explores graph coloring and its usefulness in product stocking. Finally, the forth chapter presents the development of our program and tests it thought real-world case.

Our research will make a significant contribution by providing a practical and effective solution to the problem of hazardous product stocking. It will also expand the application of graph coloring algorithms and contribute to the advancement of graph theory.

Chapter 1: Chemical products

1. Introduction

Chemical products have become an integral part of our daily lives, from the ink in our pens to the medicines that heal us. However, the use of chemicals poses potential hazards to human health and the environment. To ensure the safe use, transport, and disposal of chemicals around the world, chemical industry have developed laws and regulations.

In this chapter, we will present the hazards of chemical products and the importance of managing their risks. We will discuss the guidelines and regulations for the safe handling, storage, and transportation of it.

2. Definition

According to [1] Chemical Products means products that contain preparations or mixtures of chemical substances, including but not limited to inks, cleaning solvents, maintenance fluids, and media coatings; but excluding raw materials that are embedded in the Company's printers.

3. Types of chemical products[2]

The chemical industry involves processing or changing raw materials obtained from mines and agriculture into other useful materials that can be used in our daily lives or as raw materials for other industries.

Figure 1: Different chemical types

3.1. Agricultural Chemicals

The agricultural chemical industry supplies farmers and home gardeners with fertilizers, herbicides, pesticides, and other agricultural chemicals.

3.2. Consumer Products:

Consumer products include packaged products often referred to as “household products”. This includes everything from soaps and detergents to oral hygiene and hair and skin care products to personal care products (e.g. cosmetics, and deodorants).

3.3. Basic Chemicals:

The basic chemicals include both inorganic and organic chemicals. Organic chemicals are used in the production of other chemicals and to make products such as dyes, plastics, and petrochemical products. Inorganic chemicals are used to make industrial gases (sodium, sulfuric acid, and chlorine are some of the most common). It also serves as a catalyst in the manufacture of chemicals (used to speed up or aid a reaction).

3.4. Pharmaceuticals:

The pharmaceutical industry includes the manufacture, extraction, processing, purification, and packaging of chemical materials to be used as medications.

Pharmaceutical facilities manufacturing biological and medical products; the processing of botanical drugs and herbs; isolating active medical principles from botanical drugs and herbs; manufacturing of pharmaceutical products intended for internal and external consumption in forms such as tablets, capsules, ointments, powders, and solutions.

3.5. Specialty Chemicals:

Specialty chemicals are individual molecules or mixtures of molecules (i.e., formulations) that are manufactured based on a unique performance or function. Many other sectors rely on specialty chemicals for their products, including automotive, aerospace, agriculture, cosmetics, and food, among others.

4. Managing chemical products

In the words of [3] The use of chemicals to enhance and improve life is a widespread practice worldwide. But alongside the benefits of these products, there is also the potential for adverse effects on people or the environment. As a result, several countries or organizations have developed laws or regulations over the years that require information to be prepared and transmitted to those using chemicals.

Given the reality of the extensive global trade in chemicals, and the need to develop national programs to ensure their safe use, transport, and disposal, it was recognized that an internationally- harmonized approach to classification and labeling would provide the foundation for such programs. Once countries have consistent and appropriate information on the chemicals they import or produce in their own countries, the infrastructure to control chemical exposures and protect people and the environment can be established comprehensively manner.

So, when managing chemicals in the workplace or even in the transportation process, it is important to know the hazards of the chemical, assess the risk of chemical exposure, and control the risk of the chemical hazard so you can safely manage any chemical you work with.

4.1. Storing chemical products[4]

The safe storage of hazardous chemicals is an essential part. Chemical storage is complex (there is no one-size-fits-all plan to store chemicals) but there are regulations, campus requirements, and best practices that can guide the process.

To fully understand the hazards associated with stored chemicals you first need to know what chemicals are being stored. Safe storage begins with an up-to-date inventory of chemicals and knowledge of the hazards posed by each chemical.

4.2. Transporting chemical products[4]

If you need to safely transport chemicals, it's vital to be aware of the international directives and regulations that must be followed, the hazards of chemical transportation (and how to avoid them), and best practices for moving chemicals both on and off-site.

Transporting hazardous substances poses significant financial, environmental, and health risks, making it essential to establish a robust legislative framework to mitigate the potential for accidents during transportation. In our work, we have taken a keen interest in this critical phase.

4.3. Hazards of Chemical transportation [4]

Several possible hazards can result from chemical transportation accidents. The following list covers the nine broad classifications of hazards:

- Explosives: naturally, such substances pose the threat of explosion when in transport.
- Gases: dangerous gases can either be flammable and cause a fire or toxic and cause poisoning.
- Flammable liquids: such liquids can easily catch fire, damaging buildings and/or the environment, or causing injuries/fatalities.
- Flammable solids: certain solids are flammable, others can spontaneously combust, and other solids can form a flammable gas if in contact with water.
- Oxidizing substances: oxidizing chemicals can speed up the development of a fire, make a fire more intense, or cause combustible materials to burn spontaneously without an obvious ignition source present.
- Toxic substances: such chemicals are split into both toxic chemicals and infectious chemicals.
- Radioactive materials: the spillage of radioactive materials can have an enormously damaging impact on ecosystems and people.
- Corrosive substances: corrosive materials can present health hazards, such as burning the skin and eyes or damaging the respiratory tract.

4.4. How to avoid chemical hazards[4]

As a general rule, chemicals need to be physically segregated from incompatible chemicals; some key requirements are listed below.

- Store flammable liquids or combustible liquids in approved safety containers.

- Keep oxidizers away from flammables and combustibles.
- Keep corrosives away from substances that they may react with and release corrosive, toxic, or flammable vapors.
- Never place incompatible chemicals together in the same container during transport—you need to prevent unwanted reactions in the event of a leak or spill.
- Ensure you appropriately segregate classes of hazardous substances to avoid the risk of a reaction while they are in transport.
- Do not store chemicals alphabetically unless they are compatible.

Some best practices for the transport of hazardous substances are:

- Anyone involved in the transportation of hazardous goods on-site should wear appropriate PPE for the chemicals they are transporting.
- Always use secondary containment by placing bottles, jars, or other chemical containers in a tray or other carrier when moving chemicals on-site.
- Bring a spill management kit with you when transporting hazardous goods for a rapid response to any accident.
- Never attempt to clean up a spill without assistance if you are unsure what to do, you feel it is unsafe, or you don’t know what materials have been spilled.
- Fully secure hazardous substances on the vehicle or other mode of transport so that they can’t move or fall.

Figure 2: Chemical Transport Box Designed for safe transporting in Vehicles

5. Conclusion

Chemical products are widely used in our daily lives and various industries. However, their use introduces potential hazards that must be properly managed. So, we must understand the different types of chemicals, their hazards, and transportation requirements. Proper storage, segregation, labeling, and transportation procedures are essential to ensure safely and responsibly managed chemicals.

Chapter 2: Graph theory

1. Introduction

Initially, In the words of [5] the utility of Euler's concepts of graph theory was limited to solving puzzles and analyzing games and other forms of entertainment. However, during the mid-19th century, people began to recognize the potential of graphs as models for a diverse range of phenomena relevant to society. In the 20th century, the field of graph theory expanded considerably as the versatility of graphs as modeling tools became more apparent. This trend continues to the present day, with an increasing number of specific applications and a growing scope for their use.

The current chapter provides a comprehensive introduction to graph theory, including the definition of graphs and how they can be represented in machine-readable formats. We will also explore the basic concepts and properties of graphs, as well as the different types of graphs.

2. Graphs[5]

In mathematics, Graphs can be used to model a large number of problem areas, including social networking, chemistry, scheduling, parcel delivery, satellite navigation, electrical engineering, and computer networking.

We can define a graph as a diagram consisting of small circles, called vertices, and curves, called edges, where each curve connects two of the circles.

Mathematically, a graph G consists of two finite sets, $V(G)$ and $E(G)$. Each element of V is called a *vertex* (plural *vertices*). The elements of E, called *edges*, are unordered pairs of vertices. For instance, the set $V(G)$ might be $\{0,1,2,3,4,5,6,7\}$, and $E(G)$ might be $\{\{0,3\}, \{0,4\}, \{1,2\}, \{1,4\}, \{1,6\}, \{2,5\}, \{3,5\}, \{3,6\}, \{6,7\}\}$. Together, $V(G)$ and $E(G)$ are a graph G.

Graphs have natural visual representations (Figure 3). Notice that each element of $V(G)$ is represented by a small circle and that each element of $E(G)$ is represented by a line drawn between the corresponding two elements of $V(G)$.

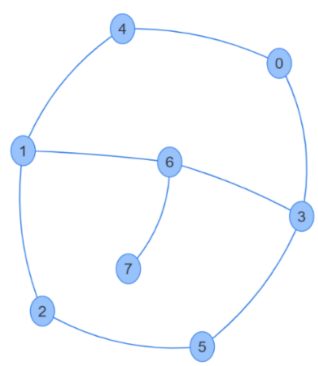

Figure 3: A visual representation of graph G.

3. Basic concepts[5]

For notational convenience, instead of representing an edge as $\{u, v\}$, we denote this simply by uv.

The order of a graph G denoted by $|V(G)|$ is the cardinality of its vertex set, and the size of a graph denoted by $|E(G)|$ is the cardinality of its edge set.

Given two vertices u and v, if $uv \in E(G)$, then u and v are said to be adjacent. In this case, u and v are said to be the *end vertices* of the edge uv. if $uv \notin E(G)$, then u and v are nonadjacent. Furthermore, if an edge e has a vertex v as an end vertex, we say that v is incident with e.

The neighborhood (or open neighborhood) of a vertex v, denoted by $N(v)$, is the set of vertices adjacent to v:

$$N(v) = \{x \in V(G) \mid vx \in E(G)\}.$$

The closed neighborhood of a vertex v, denoted by $N[v]$, is simply the set $\{v\} \cup N(v)$. Given a set S of vertices, we define the neighborhood of S, denoted by $N(S)$, to be the union of the neighborhoods of the vertices in S. Similarly, the closed neighborhood of S, denoted $N[S]$, is defined to be $S \cup N(S)$.

The degree of v, denoted by $deg(v)$, is the number of edges incident with v. In simple graphs, this is the same as the cardinality of the (open) neighborhood of v. The maximum degree of a graph G, denoted by $\Delta(G)$, is defined to be:

$$\Delta(G) = max\{deg(v) \mid v \in V(G)\}.$$

Similarly, the minimum degree of a graph G, denoted by $\delta(G)$, is defined to be:

$$\delta(G) = min\{deg(v) \mid v \in V(G)\}.$$

The degree sequence of a graph of order n is the n-term sequence (usually written in descending order) of the vertex degrees.

Example*:* Let's use the graph G in Figure 3 to illustrate some of these concepts:

G has order $|V(G)| = 8$ and size $|E(G)| = 9$; vertices 0 and 4 are adjacent while vertices 0 and 1 are nonadjacent; $N(3) = \{0,5,6\}$, $N[d] = \{0,3,5,6\}$; $\Delta(G) = 3$, $\delta(G) = 1$; and the degree sequence is $3, 3, 3, 2, 2, 2, 2, 1$.

4. Machine representation

When we want to store a graph on a computer, we can't always use the common method of drawing. There are different ways to represent graphs on a machine, and each method has its advantages and disadvantages. Depending on what we want to do with the graph, we might need to use a different representation. In this discussion, we will present three different ways for representing [6].

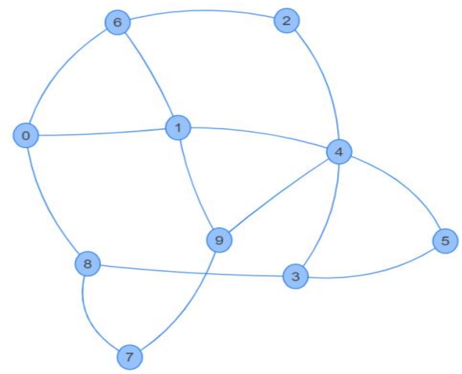

Figure 4: A Graph with an order of |V(G)| = 10.

4.1. Edge Lists [6]

A straightforward approach to graph representation is by using an edge list, which is essentially an array of edges. Each edge can be represented by an array of two vertex numbers or an object that contains numbers of the vertices that the edge connects. As an example, let's consider the graph shown in figure 4. Its corresponding edge list would be:

[0,1]	**[0,6]**	**[0,8]**	**[1,4]**	**[1,6]**	**[1,9]**	**[2,4]**	**[2,6]**	**[3,4]**	**[3,5]**	**[3,8]**	**[4,5]**	**[4,9]**	**[7,8]**	**[7,9]**

Table 1: The Edge List Representation of the Graph in Figure 4.

4.2. Adjacency Matrices [7]

An adjacency matrix of a graph G is a square matrix of size $|V(G)|\times|V(G)|$. It is represented using 0s and 1s, where each entry (i,j) in the matrix is 1 if and only if there exists an edge from vertex I to vertex j in the graph. Otherwise, the entry is 0. To illustrate, let us consider the graph in Figure 4. The adjacency matrix for this graph is given as follows:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	1	0	0	0	0	1	0	1	0
1	1	0	0	0	1	0	1	0	0	1
2	0	0	0	0	1	0	1	0	0	0
3	0	0	0	0	1	1	0	0	1	0
4	0	1	1	1	0	1	0	0	0	1
5	0	0	0	1	1	0	0	0	0	0
6	1	1	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	1
8	1	0	0	1	0	0	0	1	0	0
9	0	1	0	0	1	0	0	1	0	0

Table 2: The Adjacency Matrix Representation of the Graph in Figure 4.

4.3. Adjacency Lists [6]

An adjacency list of graph G is a data structure that combines the concepts of adjacency matrices and edge lists. Specifically, for each vertex I in a graph, an adjacency list stores an array of vertices that are adjacent to it, in other words, connected by an edge. In practice, this is implemented as an array of $|V(G)|$ adjacency lists.

To illustrate, let us consider the graph in Figure 4. The adjacency list representation for this graph is as follows:

Vertex 0	→	[1, 6, 8]
Vertex 1	→	[0, 4, 6, 9]
Vertex 2	→	[4, 6]
Vertex 3	→	[4, 5, 8]
Vertex 4	→	[1, 2, 3, 5, 9]
Vertex 5	→	[3, 4]
Vertex 6	→	[0, 1, 2]
Vertex 7	→	[8, 9]
Vertex 8	→	[0, 3, 7]
Vertex 9	→	[1, 4, 7]

Table 3: The Adjacency List Representation of the Graph in Figure 4.

4. Conclusion

This chapter has provided an introduction to the core concepts and components of graph theory. We explored the definition of graphs as a collection of vertices and edges, as well as key properties of graphs like order, size, and degree. We also examined three key ways to represent graphs in a machine-readable format such as edge lists, adjacency matrices, and adjacency lists. Each representation has distinct advantages and disadvantages, and the appropriate choice depends on the particular characteristics and uses cases of a given graph.

Chapter 3: Optimizing Management with Graph Coloring

1. Introduction

This chapter explores how graph coloring algorithms can be used to optimize the transportation of products with safety constraints. The graph coloring problem involves assigning colors to the vertices of a graph such that no adjacent vertices share the same color. The chapter shows how this problem can be applied to managing the stocking of chemical products that cannot be combined in the same wagon. It presents three graph software tools that implement graph coloring algorithms to find solutions: Graph Online, graph-coloring by amirdeljouyi, and Grin software (GRaph INterface). These tools can help determine how to safely transport incompatible products while minimizing the number of vehicles required.

2. Graph coloring problem

According to [8] The graph coloring problem is one of the most famous problems in the field of graph theory and has a long and illustrious history. In a nutshell, it asks, given any graph, how might we go about assigning "colors" to all of its vertices so that (a) no vertices joined by an edge are given the same color and (b) the number of different colors used is minimal?

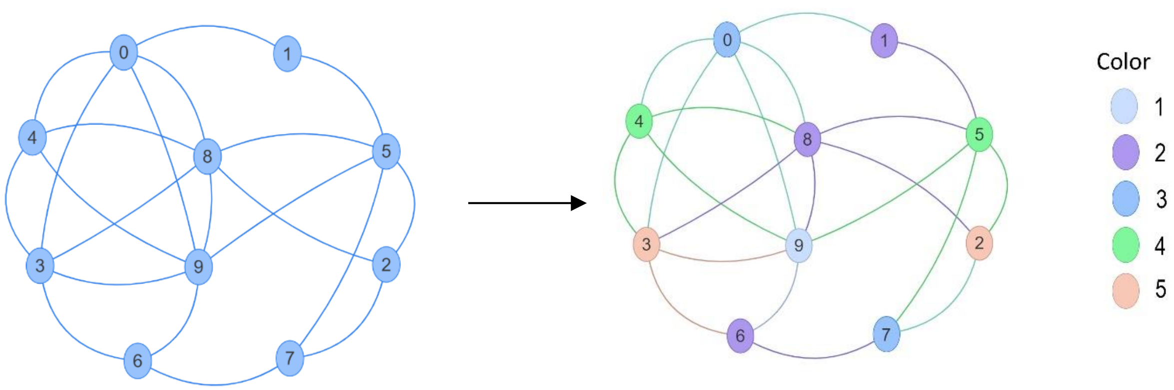

Figure 5: An example of graph coloring.

Figure 5 shows a graph G with order $|V(G)| = 10$, and size $|E(G)| = 21$. We have colored this graph using five different colors. We can call this solution a "proper" coloring because all pairs of vertices joined by edges have been assigned to different colors, as required by the problem. Specifically, one vertex has been assigned to color 1, three vertices to color 2, two vertices to color 3, two vertices to color 4, and two vertices to color 5.

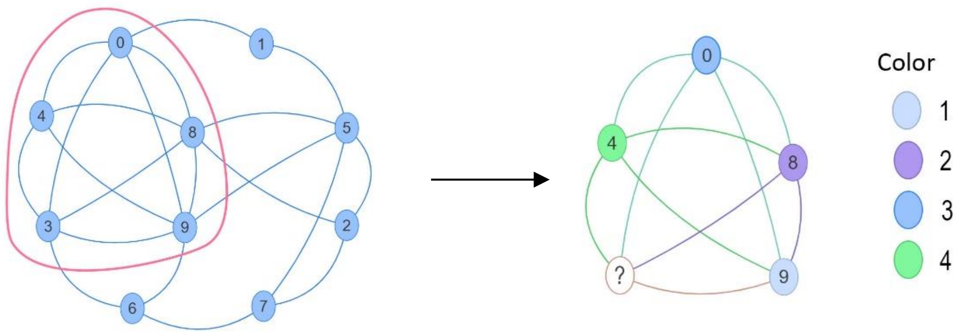

Figure 6: Proof of the optimal solution shown in figure 5 by a subgraph [8].

While there are different ways to color the example graph using five or more colors, it is not possible to color it with fewer than five colors. To prove this, we can consider a subgraph that contains only five vertices (figure 6), where each pair of vertices has an edge between them. If we had only four colors available, we would not be able to properly color this subgraph because all five vertices would need to be assigned a different color. Therefore, the solution in Figure 5 that uses five colors is the optimal solution, and no other solution exists that uses fewer than five colors.

3. Why Graph Coloring?

To understand the relationship between the transportation problem of chemical products and the concept of graph coloring, consider the example from [9] of a company that must transport six chemical products, denoted as 0, 1, 2, 3, 4, and 5, by train. Due to safety reasons, the products cannot be transported together in the same wagon if they are incompatible with each other, which mean that they could cause explosive reactions. More precisely:

- 0 cannot be transported with 1, 2, or 3.
- 1 cannot be transported with 0, 2, or 4.
- 2 cannot be transported with 0, 1, or 3.
- 4 cannot be transported with 1 or 5.

To determine the minimum number of wagons required to transport the products safely, we can represent the incompatibilities between the products as a graph of incompatibility. In this graph, the vertices represent the chemical products, and an edge connects two products if they are incompatible (see figure 7).

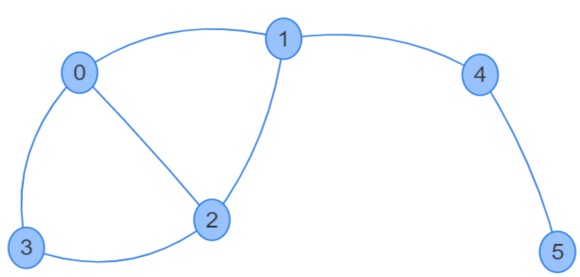

Figure 7: Incompatibility graph for our example.

The problem of determining the minimum number of wagons required to transport the products safely is equivalent to determining the chromatic number of the graph. The chromatic number of a graph is the minimum number of colors required to color their vertices.

In the example graph from [9], it is possible to color the vertices using three colors (Figure 8), which implies that three wagons are needed to transport the products safely. However, finding the chromatic number of a graph is not always straightforward, and no formula can determine the chromatic number of a graph.

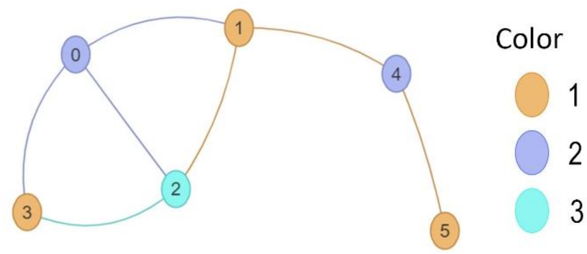

Figure 8: Coloring Graph for our Example.

4. Welsh and Powell Algorithm [9]

As mentioned above, determining the chromatic number of a graph is a challenging problem, and there is no exact formula that can provide the solution. While the Welsh and Powell coloring algorithm is a known algorithm that can be used to assign colors to the vertices of a graph in a way that minimizes the number of colors used, it cannot guarantee the minimum number of colors required for chromaticity (the chromatic number).

The algorithm consists of three steps (figure 9):

- Step 1:
  Sort the vertices of the graph in decreasing order of their degree, and assign to each vertex its order number in the obtained list. We obtain an ordered list of vertices $X1, X2, \ldots Xn$ such that: $deg(X1) \geq deg(X2) \ldots \geq deg(Xn)$.
- Step 2:
  By traversing the list in order, assign an unused color to the first uncolored vertex, and assign the same color to each uncolored vertex that is not adjacent to a vertex of that color.
- Step 3:
  If there are still uncolored vertices in the graph, go back to Step 2. Otherwise, the coloring is complete.

Algorithm Welsh-Powell

```
function WELSH_POWELL_ALGORITHM(graph)
    Create an empty list of colors: colors = []
    Sort the vertices by their degree in decreasing order: sorted_vertices = sort_vertices_by_degree(graph)
    for vertex in sorted_vertices do
        if vertex is uncolored then
            Find the first color from colors that is not used by any adjacent vertex
            if no such color exists then
                create a new color and add it to colors list
            end if
            Assign the found (or newly created) color to the vertex
        end if
    end for
    Return the colored graph
end function
function SORT_VERTICES_BY_DEGREE(graph)
    Calculate the degree of each vertex in the graph
    Sort the vertices in descending order of their degree
    Return the sorted list of vertices
end function
```

Figure 9: Welsh Powell Algorithm [9]

To gain a better understanding of the Welsh Powell coloring algorithm, let us consider an example of a graph G as shown in Figure 10.

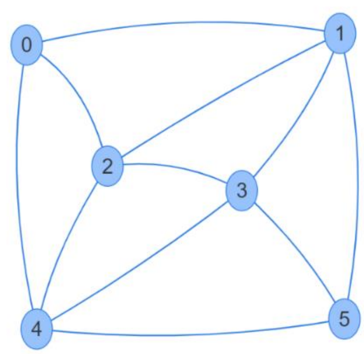

Figure 10: Example Graph illustration.

The first step of the algorithm involves calculating the degree of each vertex in the graph. In this example, the degrees of the vertices are as follows: *deg(0)=3, deg(1)=4, deg(2)=4, deg(3)= 4, deg(4)= 4,* and *deg(5)= 3*.

Once we have calculated the degree of each vertex, we can sort the vertices in descending order of their degree. In this case, the sorted list of vertices would be *1, 2, 3, 4, 0, and 5*.

Moving to step 2, we start by selecting the first vertex in the list, which is *1*. Then, we assign it the first available color, which color A (figure 11). We proceed to color all uncolored vertices that are not adjacent to an A-colored vertex with the same color (in this case, vertex *4)*.

Next, we move to the next vertex in the list, that is *2*. This vertex is adjacent to *1*, which is already colored with color A. Therefore, we assign *2* a new color, say, color B. We then color all uncolored vertices that are not adjacent to a B-colored vertex with the same color (vertex *5)*. This process ends when all vertices have been colored.

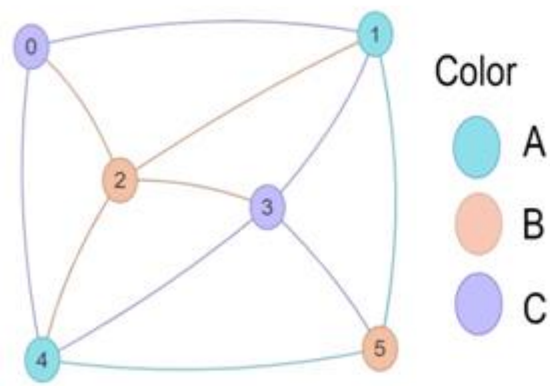

Figure 11: Implementation of the Welsh Powell Algorithm

5. Graph coloring based Tools

5.1. Graph Online

Graph Online [10] is a website that provides users a variety of tools for creating, editing, and visualizing graphs. The website supports a wide range of graph types, including directed and undirected graphs, weighted and unweighted graphs, and more.

- ***Functionalities***
  - ✓ Graph creation: Users can create new graphs by adding nodes and edges using a simple drag-and-drop interface. They can also upload a file in a supported format.
  - ✓ Graph editing: Users can edit existing graphs by adding, removing, or modifying nodes and edges. They can also change the properties of nodes and edges, such as their weight, and labels.
  - ✓ Graph visualization: Users can visualize graphs. They can also apply different graph algorithms such as graph coloring, shortest path, etc.
  - ✓ File import and export: Users can import graph data from a GraphML file. They can also export their graphs in various file formats, such as PNG and SVG.
- ***Pros***
  - ✓ User-friendly interface: Graph Online provides an intuitive interface, simplifying graph creation and customization for users without specialized technical skills.
  - ✓ Graph manipulation: The platform allows easy adjustments to graphs, including adding/removing nodes and edges.
  - ✓ Cost-effective: The website is free, making it an attractive option for those unwilling to invest in costly visualization tools.
  - ✓ Data import: Users can effortlessly import data.

- ✓ No account needed: Graph creation and customization can be done without creating an account or sharing personal information.
- ✓ Multilingual support: The website supports multiple languages, such as English, Russian, French, German, and Spanish, catering to an international user base.
- ✓ Responsive design: The platform's graphs are optimized for various devices and screen sizes, including mobile devices.
- ✓ Interactive features: Interactive elements, like zooming in on graph areas, are available.

- ***Cons***

- ✓ Limited customization: Some restrictions apply to graph customization.
- ✓ Limited data analysis tools: The website lacks advanced statistical analysis or data manipulation features, which may not suffice for some users.
- ✓ Online-only: Access to data and graphs requires an internet connection, as the platform is web-based.

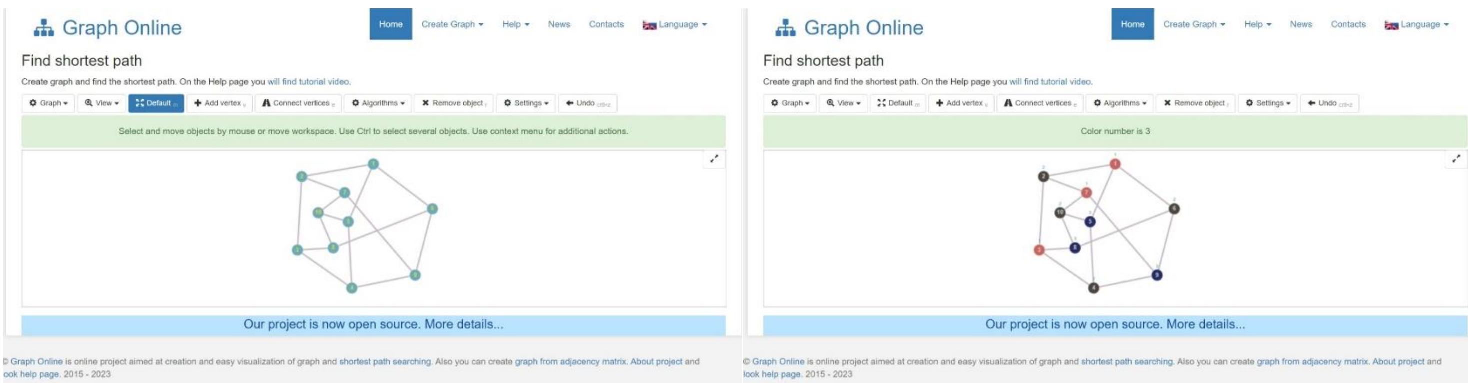

Figure 12: Screenshots from Graph Online.

5.2. Graph-coloring by amirdeljouyi

Graph-coloring by amirdeljouyi [11] is a program that provides an interactive platform for solving graph coloring problems.

- ***Functionalities***

- ✓ Graph input: Users can input a graph by manually adding nodes and edges, or by uploading a file in a supported format.
- ✓ Graph editing: Users can edit existing graphs by adding, removing, or modifying nodes and edges.
- ✓ Graph coloring algorithms: Users can find a valid coloring of the graph using different algorithms such as the Welsh Powell algorithm, genetic algorithm, and harmony search algorithm
- ✓ Visual representation: The program provides a visual representation of the graph.

- ✓ Comparison of algorithms: The program allows users to compare the results of different algorithms, by showing the number of colors used by each algorithm and the time it took to find the solution.
- ✓ File Import and Export: The user can export the graph in a supported format such as DIMACS, or JSON and also can import it in JSON or DIMACS format.

- ***Pros***

- ✓ Customizable inputs: Users can input graphs in various ways.
- ✓ Multiple algorithms: The tool offers a variety of graph coloring algorithms, useful for finding optimal solutions.
- ✓ Interactive visualization: An interactive visualization of graph coloring results is provided.
- ✓ User-friendly interface: The easy-to-use interface simplifies navigation and usage.
- ✓ Open-source: The tool is open-source; allowing users to access and modify the source code and contribute to its development.
- ✓ Versatile application: Graph coloring has numerous applications across fields such as computer science, mathematics, operations research, and more.
- ✓ Comprehensive documentation: Detailed documentation explains algorithms, inputs, and outputs, and provides examples and tutorials.
- ✓ Multi-platform support: The tool is compatible with Windows, macOS, and Linux.
- ✓ Export options: Graphs can be exported.

- ***Cons***

- ✓ Limited graph types: Not all graph types are supported.
- ✓ Limited customization: Customization options are restricted.
- ✓ Performance issues: The tool may struggle to process large graphs or use certain algorithms.
- ✓ Accuracy issues: Results' accuracy may vary depending on the algorithm used, and the tool may not always find the optimal solution.

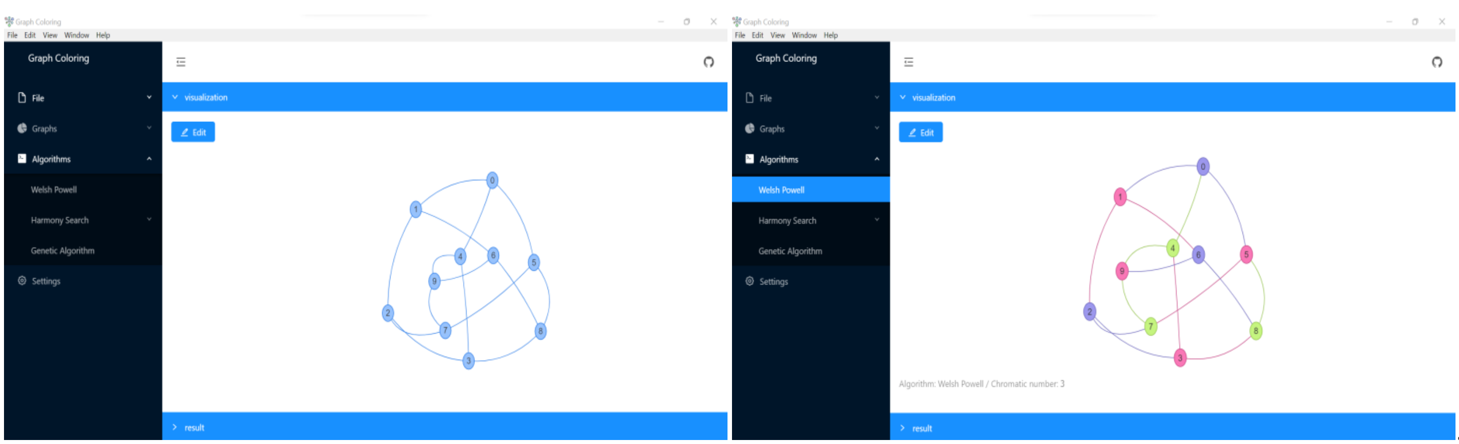

Figure 13: Screenshots from graph coloring by amirdeljouyi.

5.3. Grin software (GRaph INterface)

Grin software (GRaph INterface) [12] is a program that provides a variety of tools for creating, editing, and visualizing graphs.

- ***Functionalities***
  - ✓ Graph creation: tools to create graphs, such as nodes and edges, and to specify their attributes.
  - ✓ Graph editing: tools to modify existing graphs, adding, removing, and updating nodes and edges.
  - ✓ Graph visualization: features for visualizing graphs, including node and edge styles, labels, and layouts.
  - ✓ Data import and export: the ability to import data into the software from external sources and to export graph data to a variety of file formats.
  - ✓ Graph analysis: tools for analyzing graph data, including connectivity, centrality measures, and shortest paths.

Note: The specific features and capabilities of Grin software may vary depending on the version and implementation.

- ***Pros***
  - ✓ Customizable inputs: Users can input graphs in different ways.
  - ✓ Multiple algorithms: The tool offers a variety of graph algorithms.
  - ✓ Diverse graph types: Support for directed and undirected graphs, as well as weighted graphs, is available.
  - ✓ Graph manipulation: Users can manipulate graphs, calculate graph metrics, and more.
  - ✓ Export options: Graphs can be exported.

- ***Cons***
  - ✓ Steep learning curve: GRIN's user interface may be challenging for users new to graph theory or visualization.
  - ✓ Performance issues: GRIN is unsuitable for large-scale graphs due to potential performance problems.
  - ✓ Limited support: Developer or community support for GRIN is limited.
  - ✓ Compatibility issues: The software is only compatible with Windows.
  - ✓ Language barriers: French is the only supported language.

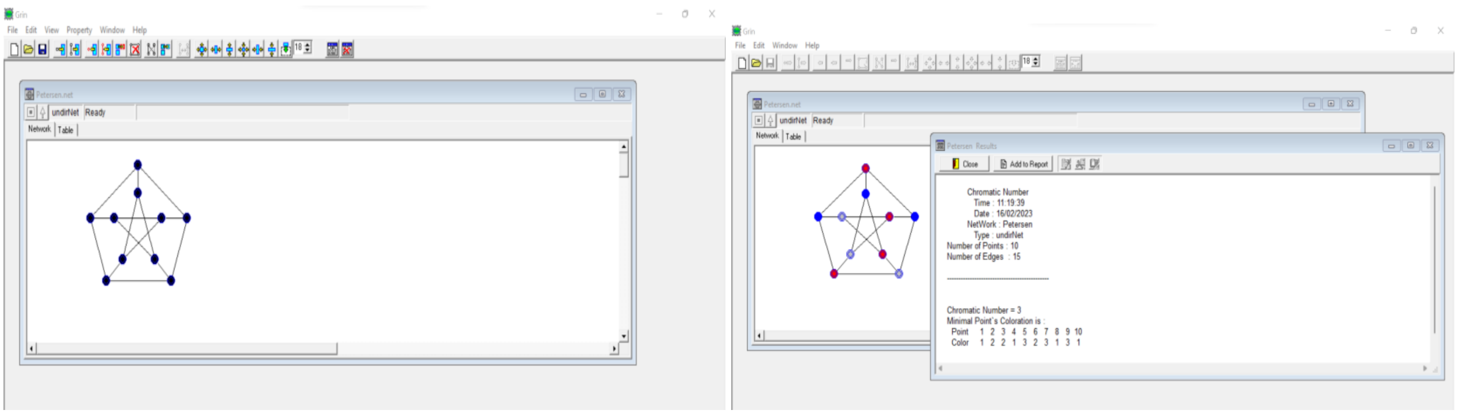

Figure 14: Screenshots from Grin software (GRaph INterface).

5.4. Why these tools may not give the best Solution?

The software tools mentioned earlier may pose difficulty in terms of accessibility and usability due to several factors. Firstly, they require a comprehensive understanding of graph theory, which may not be readily available to all users. Additionally, some of these tools are only available online, which means they cannot be used offline. Moreover, certain tools are unable to handle large volumes of data or complex graphs. Consequently, they may not be suitable for solving real-world problems, such as separating chemical products.

6. Conclusion

This chapter demonstrated how graph coloring algorithms can optimize the transportation of products with safety constraints. The graph coloring was used to model the problem of transporting chemical products that cannot be combined in the same vehicle. By representing product incompatibilities as a graph and applying graph coloring algorithms, the minimum number of vehicles needed for safe transport can be determined. The chapter presented three graph software tools—Graph Online, graph-coloring by amirdeljouyi, and Grin software—that implement graph coloring algorithms. Using these tools could help companies efficiently and safely transport incompatible products while reducing costs.

Chapter 4: Software Design and Development

1. Introduction

Software design is a crucial step in the development process that translates client requirements into a technical blueprint that developers can implement. This chapter discusses the design and development of the software application, including tools, libraries, and techniques used. Specifically, it covers the use case and sequence diagrams employed to model the system and its functionality, the programming languages and libraries selected, and the integrated development environments.

2. Modulization using UML

2.1. Use case Diagram

Use case diagram provides a simple visual model of system functionality to facilitate clear communication and guide implementation across the software development lifecycle. In other words it represents the functionality of a system from the user's perspective. Use case models how users interact with the system to achieve specific goals.

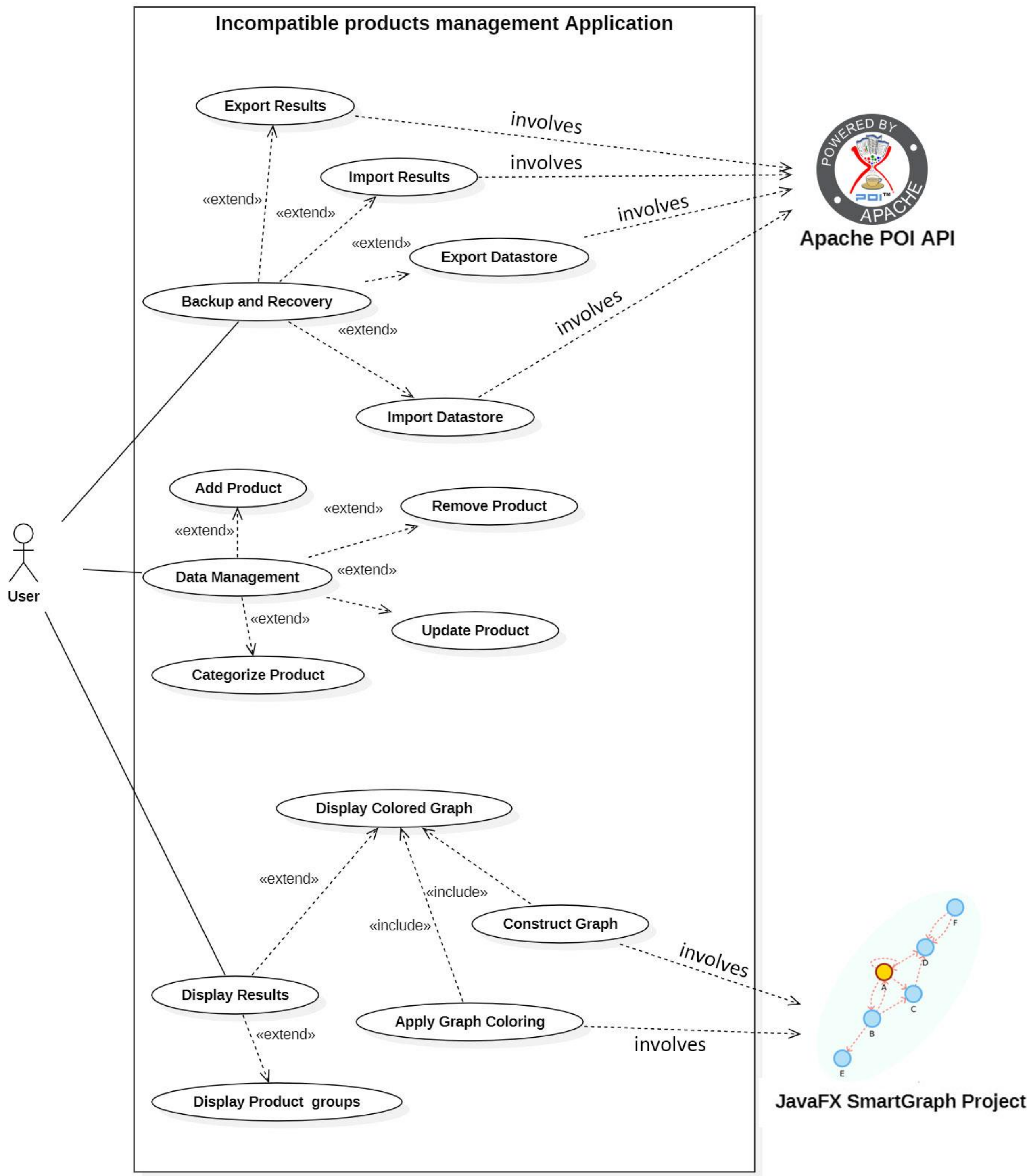

Figure 15: Use case Diagram.

2.2. Sequence Diagrams

Sequence diagram is an important tool in software design that provides a visual representation of the interactions between components in a system. It allows a clear and concise overview of interactions' chronological order, the components involved in a procedure, and the messages exchanged between them.

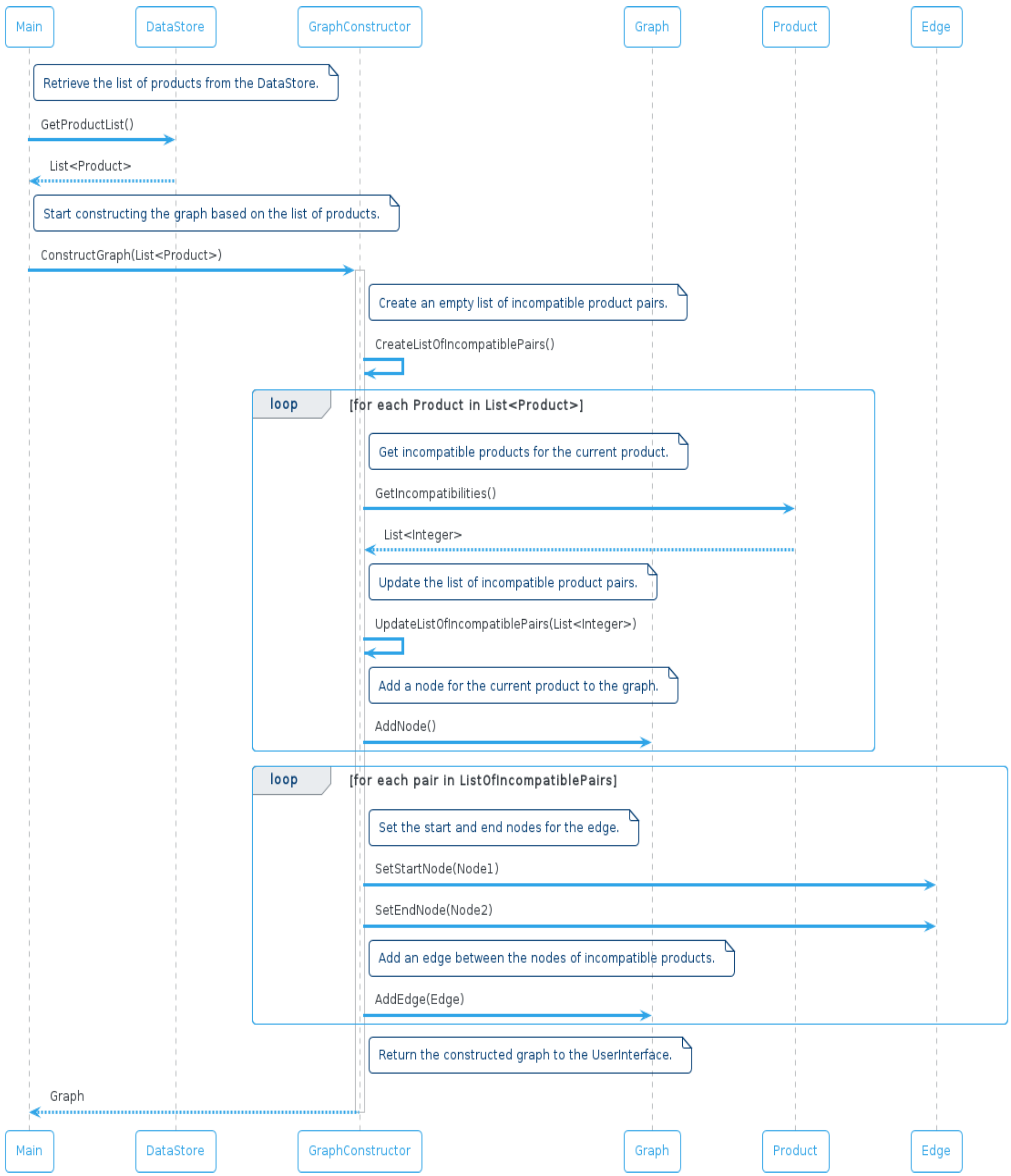

Figure 16: Sequence Diagram Graph construction

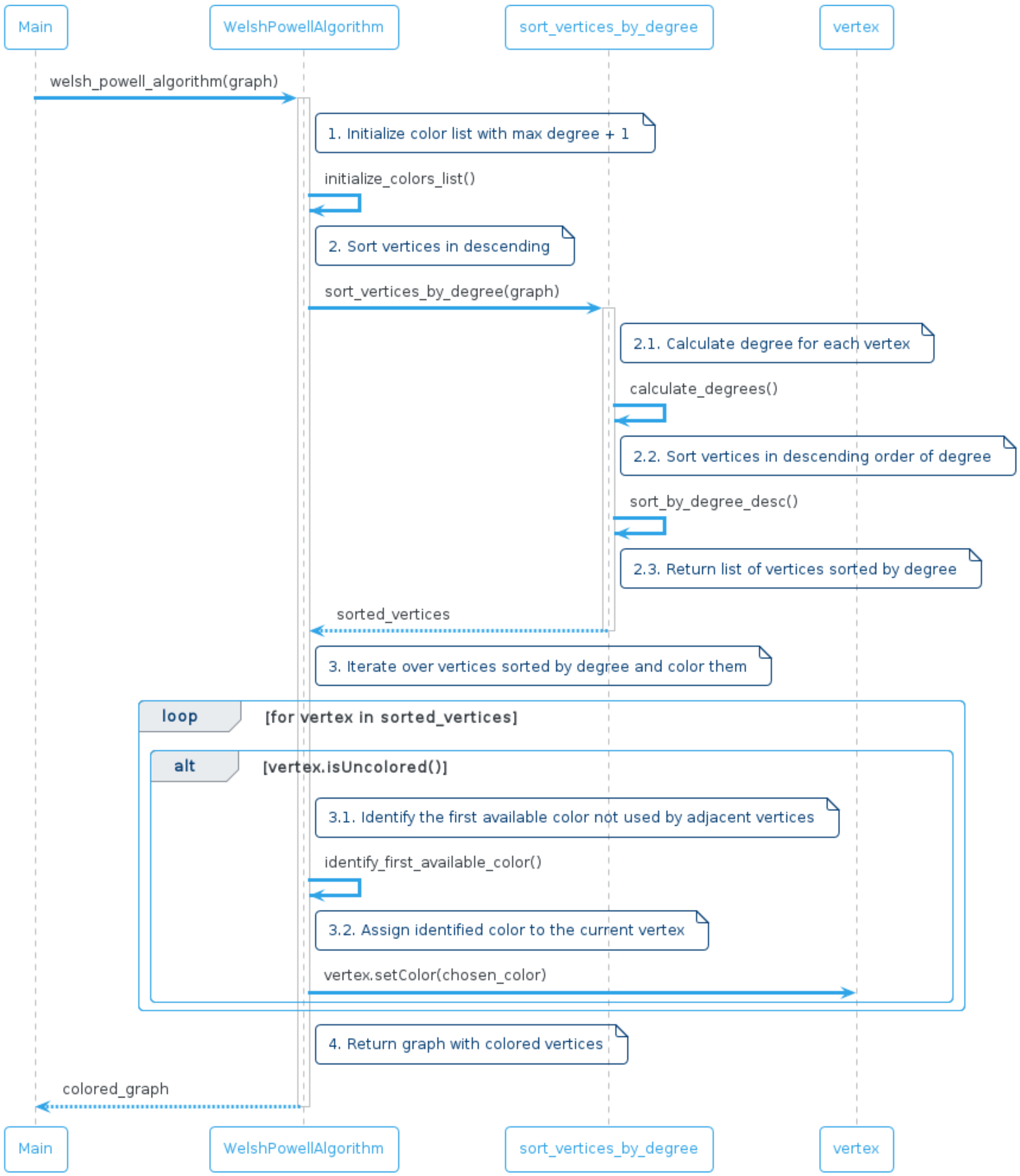

Figure 17: Sequence Diagram Graph coloring

3. Programming Tools and Libraries

3.1. Why Java and JavaFX?

- In the words of [13] Java is a powerful, general-purpose programming environment. It is one of the most widely used programming languages in the world, and has been exceptionally successful in business and enterprise computing by providing a stable, solid base for companies to develop business-critical applications.
- It is class based and object oriented. It is considered to be relatively easy to read and write.
- According to [14] JavaFX is an opensource framework for developing rich client applications written in Java, which enables you to take advantage of all Java features, it supports data binding through its libraries, it provides a rich set of multimedia support such as playing back audios and videos and out-of-the-box support for applying effects and animations, which are important for developing gaming applications.

3.2. **Apache POI API** [15]

- The Apache POI Project's mission is to create and maintain Java APIs for manipulating various file formats based upon the Office Open XML standards (OOXML) and Microsoft's OLE 2 Compound Document format (OLE2).
- You can read and write MS Excel files using Java. In addition, you can read and write MS Word and MS PowerPoint files using Java.

Note that our program uses the Apache POI API to save, load, recover, and backup data.

3.3. **JavaFX Smart Graph** [16]

- (Java FX) graph visualization library that can automatically arrange the vertices' locations through a force-directed algorithm in real-time.
- You can statically place the vertices according to other algorithms.
- Vertices and edges can be styled through a CSS style sheet or programmatically.

Note that our project uses this library to construct and build graphs.

3.4. **MaterialFX** [17]

MaterialFX is an open-source Java library that provides material design components for JavaFX. The aim of the project is to bring components that follow as much as possible the Google’s material design guidelines to JavaFX.

3.5. **Eclipse** [18]

Eclipse is an IDE for “anything, and nothing at all,” meaning that it can be used to develop software in any language, not just Java. It started as a proprietary replacement for Visual Age for Java from IBM, but was open sourced in November 2001. Eclipse is now controlled by an independent non-profit organization called the Eclipse Foundation. Since 2001, it has been downloaded over 50 million times; it is now being used by thousands of developers worldwide.

It also has a sizable following in the university community, where it is used in classes on programming and object-oriented design.

3.6. Gluon Scene Builder [19]

Gluon Scene Builder is a drag and drop UI designer tool allowing rapid desktop and mobile app development. Scene Builder separates design from logic, allowing team members to quickly and easily focus on their specific aspect of application development.

Scene Builder works with the JavaFX ecosystem – official controls, community projects, and Gluon offerings including Gluon Mobile, Gluon Desktop, and Gluon CloudLink.
Scene Builder is open source, and it is freely licensed under the BSD license. Gluon can provide custom consultancy, training, and opensource commercial support.

4. User Interface

We decided to give Our Application the name IPM and the following figures represent the user interface of IPM:

4.1. Logo

Figure 18: Logo

4.2. Overview

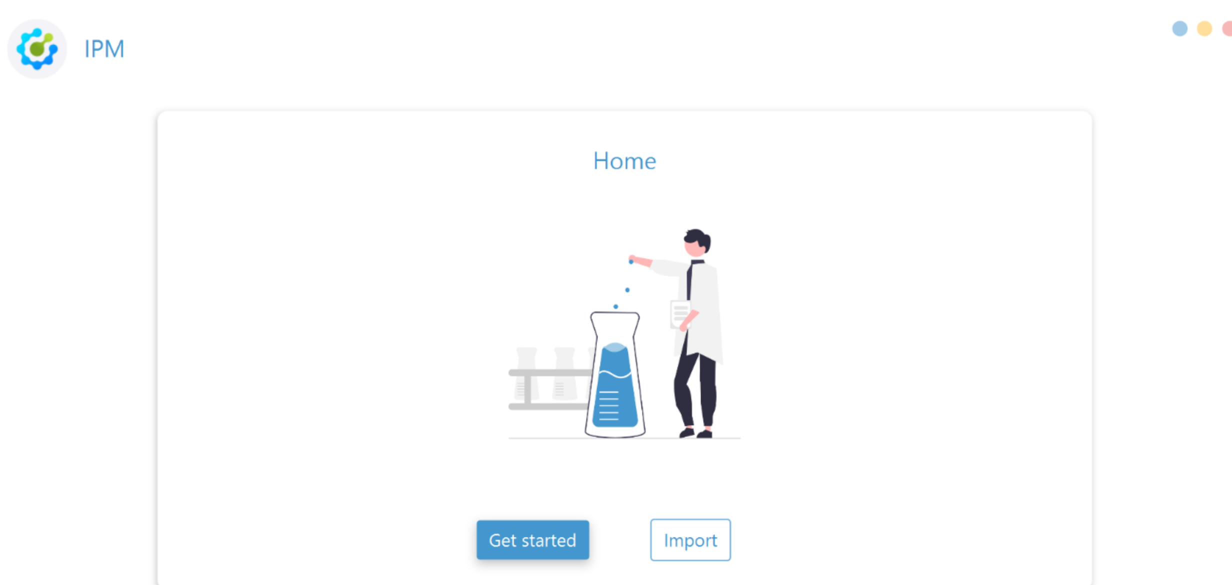

Figure 19: Home Page

4.2.1. Get Started

This button provides access to the product list page, allowing you to manage your products.

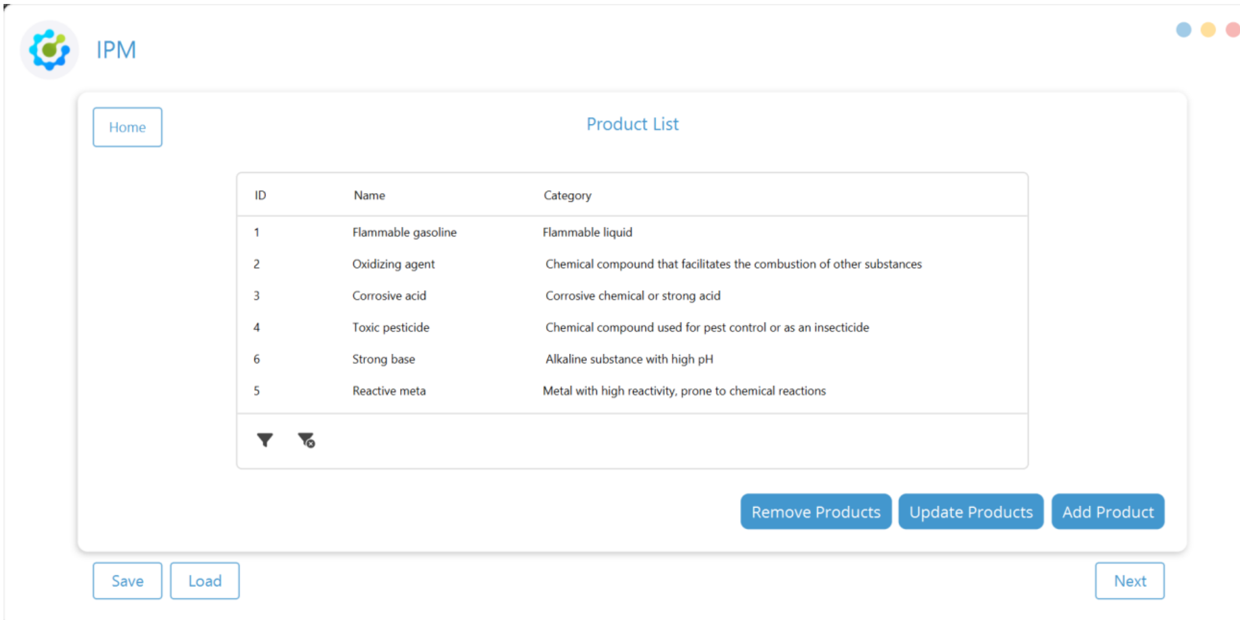

Figure 20: Product List Page

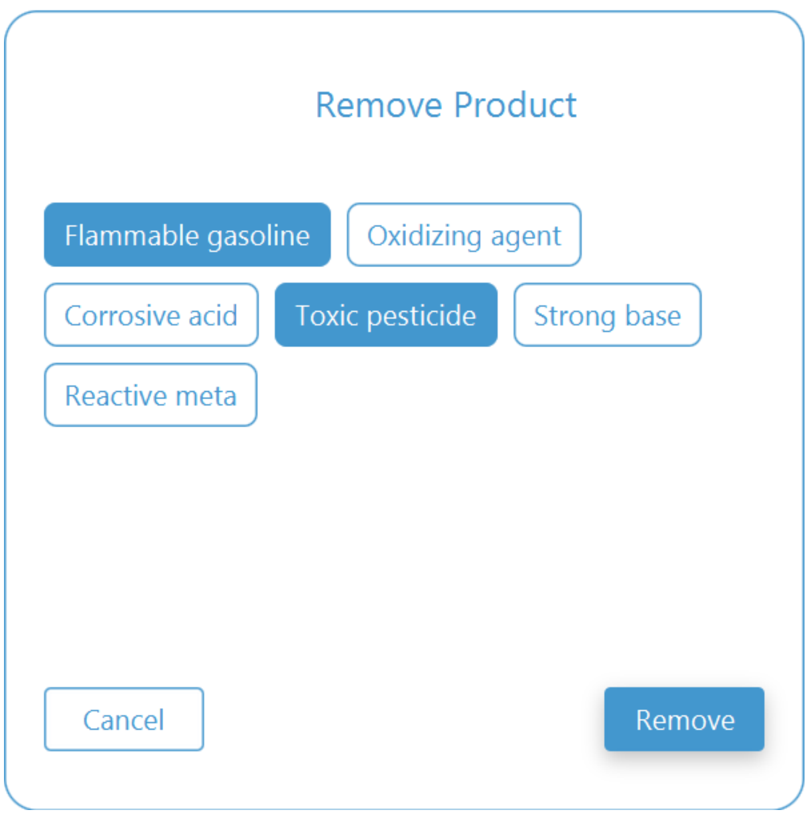

Figure 21:Remove Products Page

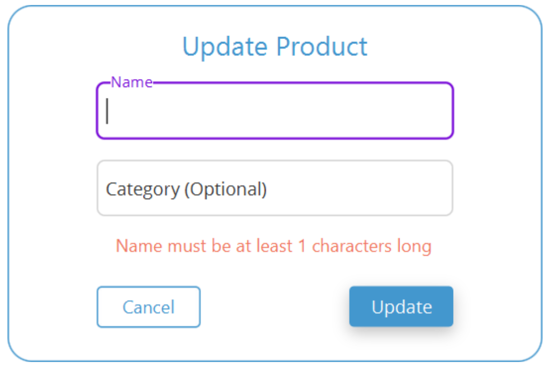

Figure 22 Update Product Page

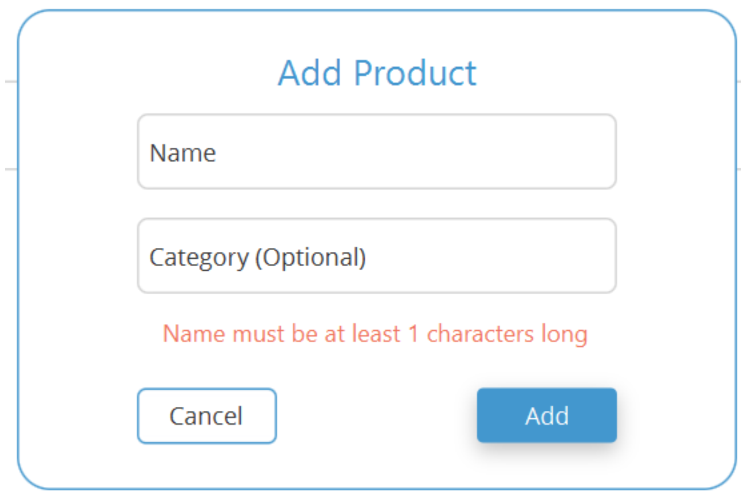

Figure 23: Add Product

- *Save:* This button allows you to save your product list into an excel file.
- *Load:* This button allows you to load your product list from an excel file.
- *Next:* This button directs you to the following page, where you can choose incompatible products for each product.

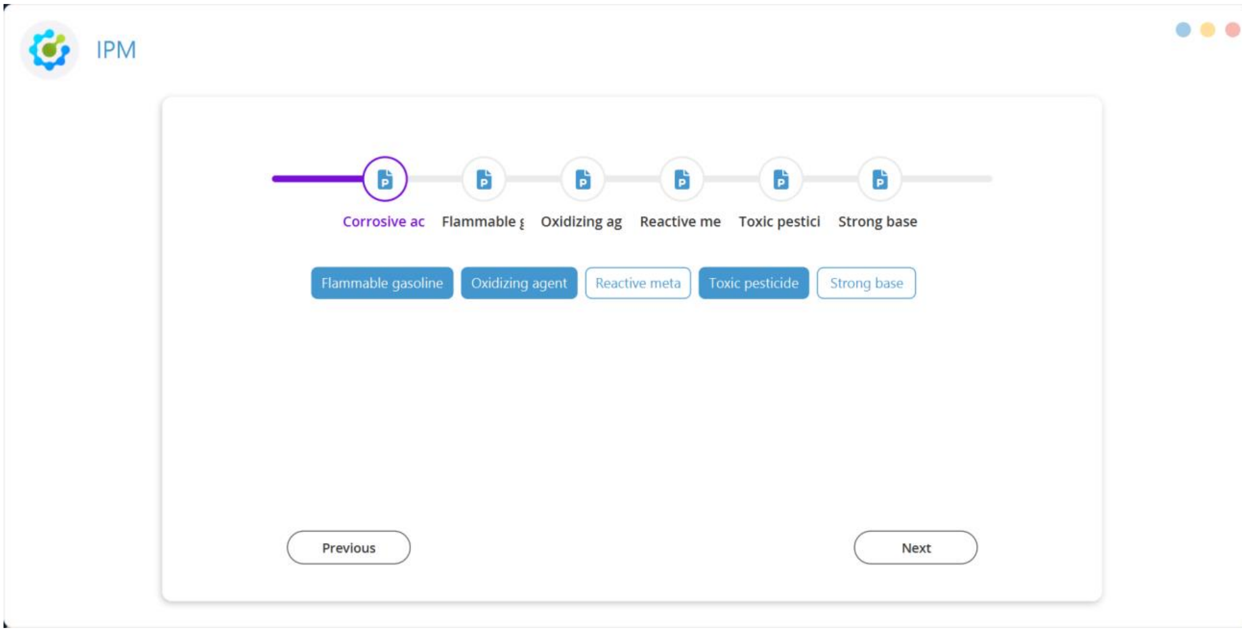

Figure 24:Selecting Incompatibles Page

- *Previous:* This button allows you to go back to the previous page.
- *Next:* This button directs you to the following page, where you can view the colored graph.

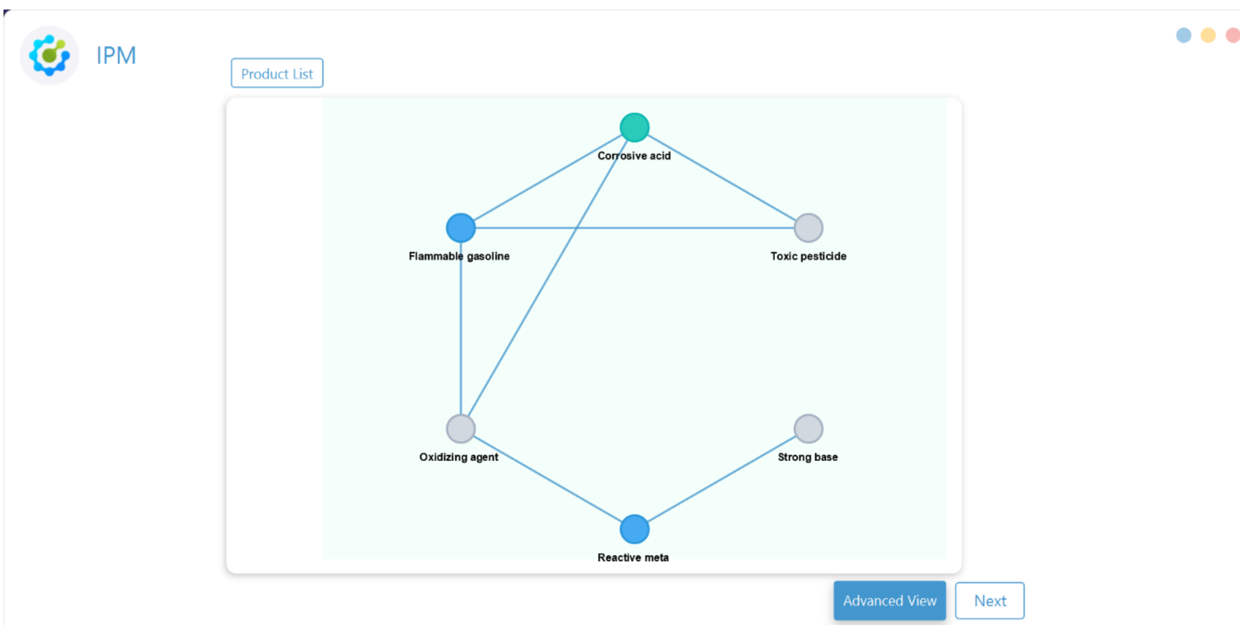

Figure 25: Colored Graph Page

- *Product List:* This button takes you back to the product list page.
- *Advanced View:* This button shows a view that provides zooming capability and automatic layout enabling.
- *Next:* This button directs you to the results page, where you can view the chemical products in separated groups.

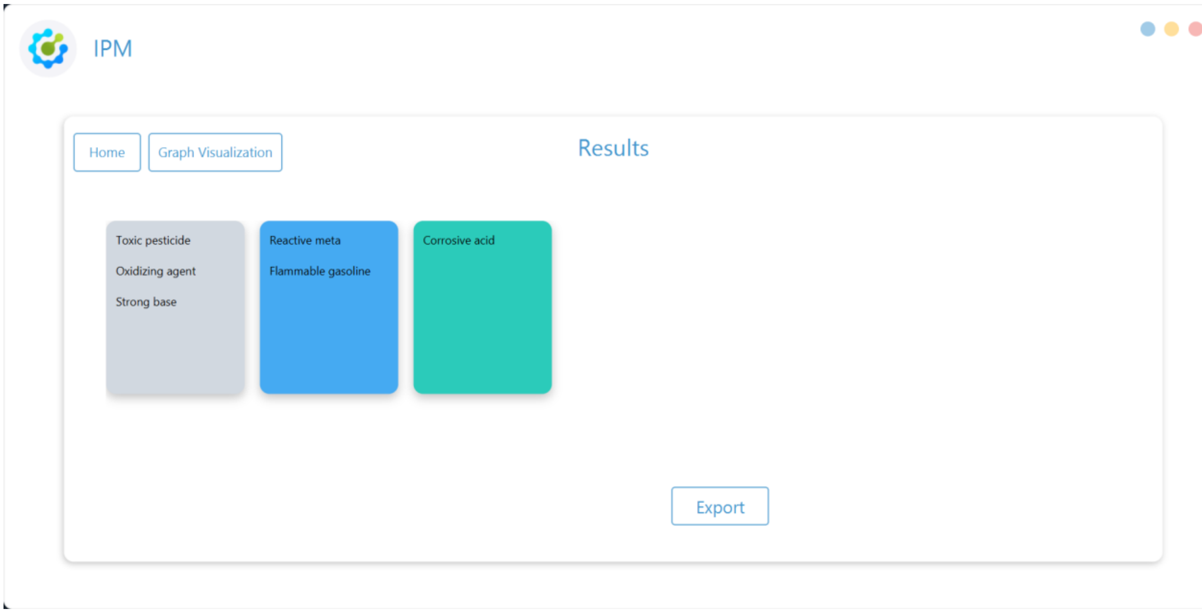

Figure 26:separated groups page

- *Home:* This button directs you to the home page.
- *Graph Visualization:* This button directs you to the colored graph page.
- *Export:* This button allows you to export your results into an excel file.

4.2.2. Import

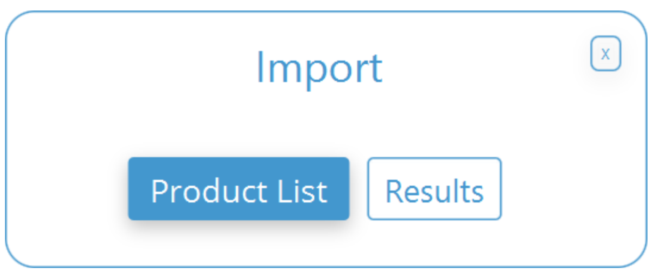

Figure 27:Import page

- *Product List:* This button allows you to Import your product list from an excel file.
- *Results:* This button allows you to Import your results from an excel file.

5. Testing

We conducted a thorough evaluation of IPM's performance in three distinct situations: one involving 10 colors, another utilizing 30 products, and a third representing an unconnected graph.

5.1. The 10 Colors Situation:

IPM successfully identified all colors in the 10-color situation, as demonstrated in Figure 28.

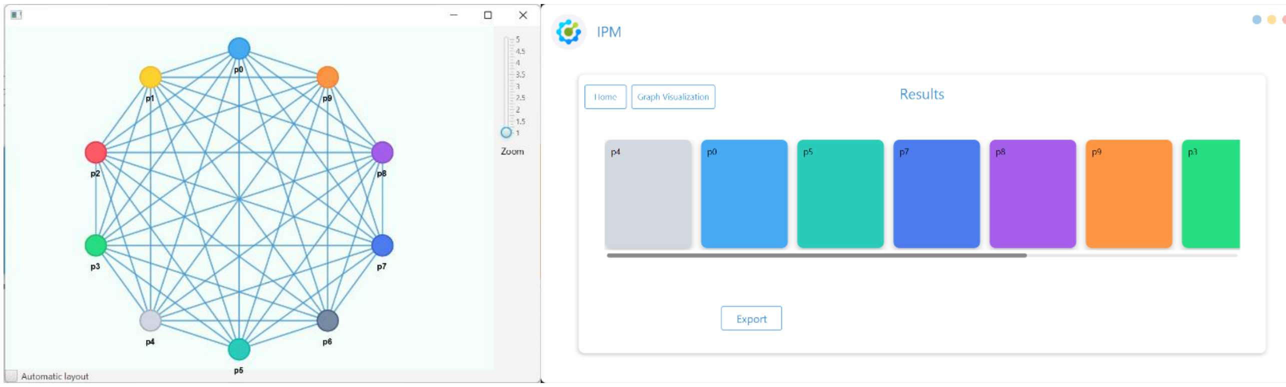

Figure 28: Performance of IPM in 10-Color Situation

5.2. The 30 Product Situation:

IPM correctly identified the desired solution in the 30-product situation, as shown in Figure 29.

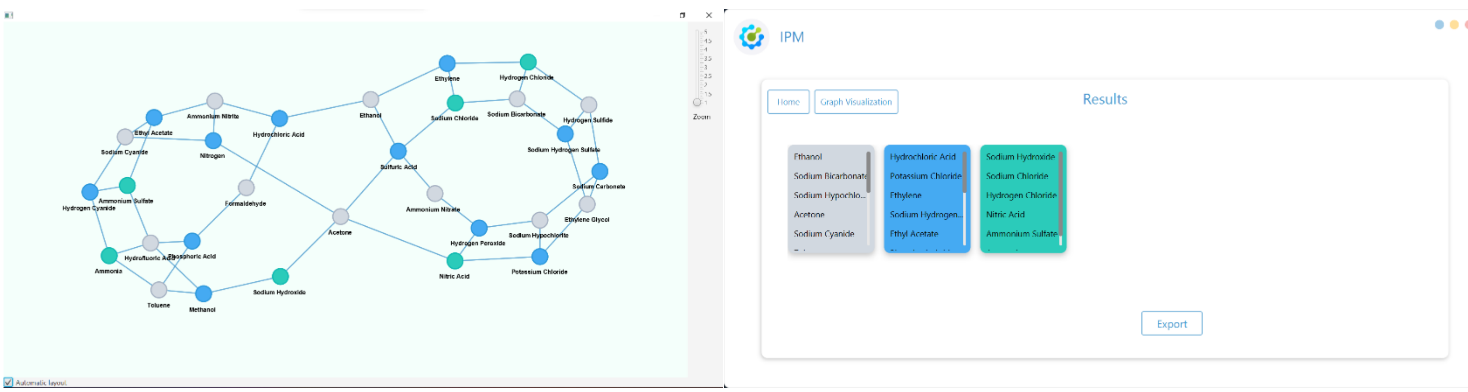

Figure 29: Performance of IPM in 30-Product Situation

5.3. The Unconnected Graph Situation:

IPM accurately identified the desired solution in the unconnected graph situation, as demonstrated in Figure 30.

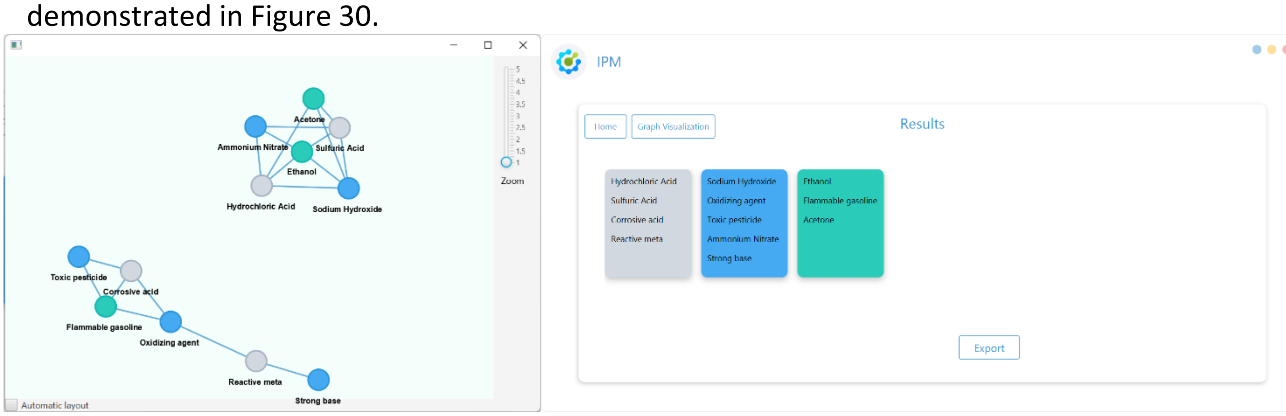

Figure 30: Performance of IPM in Unconnected Graph Situation

Overall, our comprehensive testing results indicate that IPM performs well across varying sizes and colors of situations.

6. Conclusion

The design and development of software is a complex process that requires careful planning and best selection of appropriate tools and techniques. For our project, Java and JavaFX were selected due to their power, flexibility, and multimedia capabilities. Useful libraries like Apache POI and JavaFX SmartGraph were incorporated to help implement important functionality. Eclipse and Gluon Scene Builder aided in the development and user interface design. And use case and sequence diagrams were created to visually represent the system's functionality and interactions.

General Conclusion

Our research has proved that the graph coloring algorithm, and specifically the Welsh Powell algorithm, has significant potential in enhancing the safety of product stocking. So, we have developed a program that can determine the safest way to store hazardous products based on their properties. Our program provided a comprehensive solution to the problem of minimizing the risk of accidents and ensuring the personnel safety.

Our research has also contributed to the advancement of graph theory by expanding the application of graph coloring algorithms. We have demonstrated the usefulness of graph coloring in real-world applications and we have shown that it can be used by practitioners and policymakers to manage hazardous materials effectively.

Going forward, there is still room for further research on the application of graph coloring algorithms in other areas of product stocking and management. However, our thesis provides a solid foundation for future studies, and our program can serve as a starting point for developing more advanced solutions.

Overall, our research has made a significant contribution to the field of product stocking and can make a positive impact on the safety of personnel in various industries. We hope that our findings will be used to enhance the management of hazardous materials and ultimately contribute to a safer working environment for all.

BIBLIOGRAPHY

- [1] Chemical Products. Law Insider, Consulted in 14 Mar. 2023, https://www.lawinsider.com/dictionary/chemical-products.

- [2] Chemical Sector Profile. Cybersecurity and Infrastructure Security Agency, Consulted in 14 Mar. 2023, https://www.cisa.gov/sites/default/files/publications/Chemical-Sector-Profile_Final%20508.pdf.

- [3] Globally Harmonized System of Classification and Labeling of Chemicals (GHS). United Nations, 2019.

- [4] Chemical Transportation: How Do You Safely Transport Chemicals?" Safety Storage Systems, Consulted in 14 Mar. 2023, https://safetystoragesystems.co.uk/chemical-transportation-how-do-you-safely-transport-chemicals/.

- [5] Combinatorics and Graph Theory. 2nd ed, West. D. B, Prentice Hall, 2001.

- [6] Representing Graphs. Algorithms, Khan Academy, Consulted in 14 Mar. 2023, https://www.khanacademy.org/computing/computer-science/algorithms/graph-representation/a/representing-graphs.

- [7] Introduction to Graph Theory. 5th ed. Wilson, Robin J., Pearson Education Limited, 2010.

- [8] A Guide to Graph Coloring Algorithms and Applications. Lewis, Andrew. Springer International Publishing, 2015.

- [9] Chapter 7: Coloration de graphes (Graph Coloring). Option spécifique – JtJ 2016, edited by French Ministry of National Education, 2016.

- [10] Graph Online. GraphOnline, Consulted in 14 Mar. 2023, https://graphonline.ru/en/.

- [11] Graph Coloring. Deljouyi, Amir. GitHub,Consulted in 14 Mar. 2023, https://amirdeljouyi.github.io/graph-coloring/.

- [12]. Logiciel Grin. Apprendre en Ligne, Consulted in 14 Mar. 2023, https://www.apprendre-en-ligne.net/crypto/grin/.

- [13]. Evans, Benjamin J & David Flanagan. Java in a Nutshell. 7th ed., O'Reilly Media, Inc., 2018.

- [14] Learn JavaFX 17. Sharan & Kishori &Peter Spath. Apress, 2022.

- [15] Apache POI. Apache Software Foundation, Consulted in 14 Mar. 2023, https://github.com/apache/poi.

- [16] JavaFX SmartGraph. Silva, Bruno. GitHub, Consulted in 14 Mar. 2023, https://github.com/brunomnsilva/JavaFXSmartGraph.

- [17]. Palexdev/MaterialFX: A library of material components for JavaFX. GitHub, Consulted in 14 Mar. 2023, https://github.com/palexdev/MaterialFX.

- [18] Eclipse IDE Pocket Guide: Using the Full-Featured IDE. McKellar, Ed. O'Reilly Media, Inc., 2005.

- [19] Scene Builder. Gluon, Consulted in 14 Mar. 2023, https://gluonhq.com/products/scene-builder/.