Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th June 2025. Vol. 4. No. 3; e-ISSN: 2808-4519

Analysis of the Use of Backtracking Algorithm in Course Scheduling

Surizky Ananda[1*], M. Khalil Gibran [2], Fajar Syakbani[3], Maria Ulfa[4], Maya Sari Hasibuan [5]

[1,2,3,4,5]*State Islamic University of North Sumatra (UIN North Sumatra)*
surizkyananda@gmail.com[1*], m.khalil1100000202@uinsu.ac.id[2], fajarsyakbani2004@gmail.com[3], mulfa7903@gmail.com [4], mayasarihasibuanaa@gmail.com[5]

Abstract

In a university study program, course scheduling is an essential procedure to guarantee the effective use of resources, including lecturers and classrooms. The goal of this study is to examine how the Constraint Satisfaction Problem (CSP) Method and the Backtracking Algorithm are used in the scheduling of courses in the State Islamic University of North Sumatra's Computer Science Study Program. Through the assignment of courses to the appropriate time slots, classrooms, and lecturers, this study seeks to maximize the timetable while meeting predefined limits. such include the length of the course, the availability of lecturers, and the size of the classroom.The NetworkX library is used to show the graph-based approach that is being used. utilizing the NetworkX library, in which classrooms, lecturers, and courses are represented by vertices, and the connections between them are represented by edges. The technique By removing incompatible scheduling configurations, this technique effectively lowers the quantity of erroneous configurations. The findings of the study demonstrate that a workable and ideal schedule that minimizes conflicts and satisfies all requirements may be created by employing CSP and backtracking. disputes. This study offers a possible model and advances knowledge of the practical applications of computing techniques like CSP and backtracking in scheduling issues.

Keywords: *Graphic Visualization, NetworkX, Matplotlib, Lecturer, Class, Backtracking Algorithm, CSP, Course, Scheduling*

1. Introduction

Every university's lecture procedure needs to be organized to ensure smooth operation. Setting aside time and location is essential to achieving the best possible course schedule and avoiding lecture conflicts. When the ideal combination of lecturers and courses is found, together with enough room for students and in line with lecture time, the course schedule is considered optimal. Some things that are considered in the scheduling process include students, lecturers, room capacity, working days, working hours, and so on (Handayani et al., 2019). Problems in scheduling can be solved using the Constraint Satisfaction Problem (CSP) which is a problem approach by meeting a number of constraints or criteria (constraints). One algorithm that can be used in solving CSP is a backtracking algorithm based on DFS (Depth- First Search) (Russell & Norvig, 2023). Nevertheless, a variant of the backtracking approach is employed in this work, which allows one to skip levels that don't provide answers while looking for solutions in the tree and move on to the next level expansion (Ginsberg, 2023). Thus, this study uses a modified version of the backtracking technique to investigate the course scheduling in the State Islamic University of North Sumatra's Computer Science StudyProgram. The goal of this study is to create a tree-shaped course schedule for the State Islamic University of North Sumatra's Computer Science Study Program. The tree is then traced using a modified backtracking algorithm until the best possible course schedule is found. Course data is used in this study in the. information gathered, including professor and course statistics. Additionally, a research issue with choice variables, objective functions, and restrictions is developed using these data. The root node is at level 0 of the tree and is treated as a variable that has no value yet (the first node). This is where tree formation begins. The tree's levels are regarded as variables. The tree's edges are regarded as pupils, and its nodes as domains. Constraints affect the relationship between nodes and. levels, as well as between nodes at the previous or next level. Limitations include the number of credits in the course, the time allotted for it, the distance between the course and the lodging, and many more. The node can either retrace to the closest node or become a dead node if it is unable to expand. The nodes at that level are removed and a new level node is created if it is still unable to identify a solution. A path is created from the root node to the leaf nodes on the tree, where the course schedule will be located, if the search for a solution is successful or there is no more node expansion.

2. Literature Review

2.1. Tree

According to Munir (2020), trees are connected graphs devoid of circuits. Although there is no set root for trees or free trees, one vertex may be designated as the root (Wulandari, 2019). In trees, the following terms are frequently used:

1. One of the nodes chosen and positioned at level zero of the tree is the root node.
2. A sub-tree, also known as an up-tree, is a section of the tree in which S is a sub-tree of tree G if ( ) ( ) ( ) ( )
3. A path is made up of several interconnected edges.
4. A node of degree one is a leaf or dead node.
5. The number of edges next to a vertex is called its degree.
6. Levels exist within

2.2. Constraint Satisfaction Problem (CSP)

The goal of the constraint satisfaction problem (CSP) technique is to solve a problem by satisfying a variety of constraints or criteria (Rosmasari et al., 2019). It is necessary to understand the following elements of CSP difficulties (Abidin et al., 2019):

1. Usually populated with domains that already have restrictions, variables are a set that can have a variety of values. The formula for variables is x = {x 1,x 2,x_3,...,x_n}.
2. The rule or condition c = {c 1,c 2,c_3,...,c_n' is a constraint.
3. The values of variables can be controlled via constraints. There are two categories into which all constraints can be divided: Ex: an ordered list of elements in the case of a finite domain, to use an extension. Specifically, for example, mathematical equations. A domain is a collection of fillable values.

2.3. Algoritma Backtracking

Backtracking methodology is based on a search method that methodically looks through all potential solution spaces for a solution. When this method is used, only the search that results in the answer is always taken into consideration; there is no need to look at every potential solution (Adiguna & Swanjaya, 2019). This algorithm's search is typically based on DFS (Depth-First Search), in which the process begins at the root node and proceeds to the next level in the solution space, which is represented by a tree structure, with the left child node (Rosen, 2019). The difficulty is solved by the path that is created from the root to the leaves. There are steps involved in using the backtracking algorithm to determine the solution (Ferdinal, 2019):

1. A path from the root node to the leaf nodes is formed to acquire the solution. The enlarged live node is referred to as an E-node (Expand-node), while the born node is known as a live node.
2. The E-node becomes a dead node that can no longer be enlarged if the trajectory that results from growing it does not lead to a solution.
3. The search procedure is completed by creating a new node if the path's final place is a dead node. Additionally, the search is conducted by going back to the closest vertex if no vertex is produced.
   If a solution is identified or no more vertices are enlarged, the search is terminateds.

2.4. Schedulling

In order to decide when and where each task as part of the overall project should be completed with restricted resources, scheduling is a planning activity. Scheduling, according to Baker (2021), is the process of completing tasks at a predetermined time with the resources at hand. Scheduling can aid in decision-making, decrease the buildup of semi-finished goods in the production line, and improve the efficacy and efficiency of resource usage (Munarto, 2017).

1. Every university's lecture procedure needs to be organized to ensure smooth operation. Setting aside time and location is essential to achieving the best possible course schedule and avoiding lecture conflicts. When the ideal combination of lecturers and courses is found, together with enough room for students and in line with lecture time, the course schedule is considered optimal. The scheduling procedure takes into account a number of factors, such as students, lecturers, room capacity, working days, working hours, and so forth.

2.5. Analysis of Algorithm implementation

In this study, the backtracking algorithm itself serves as a solution-finding technique for combinatorial issues by progressively attempting potential solutions and reverting if the answer does not satisfy specific requirements. Backtracking helps find schedules that don't conflict while taking into account different limitations such lecturer teaching hours and classroom availability in the context of college scheduling (Wulandari, 2022). This case study explains how the backtracking algorithm is used in lecture scheduling as a real-world application. The backtracking algorithm is utilized in this work to create a lecture schedule that takes into account the length of the course, the class, and the availability of the lecturers. There are steps involved in using the backtracking technique to find the solution:

1. A path from the root node to the leaf nodes is formed to acquire the solution. The enlarged live node is known as the E-node (Expand-node), while the born node is known as the live node.
2. The E-node becomes a dead node that can no longer be enlarged if the trajectory that results from growing it does not lead to a solution.
3. The search procedure is completed by creating a new node if the path's final place is a dead node. Additionally, the search is conducted by going back to the closest vertex if no vertex is produced.
4. If a solution is found or not, the search is terminated. Backtracking is therefore crucial for optimizing course scheduling. The system can be methodically investigated until the optimal solution that satisfies all requirements is found. Furthermore, the process's efficacy can be raised by combining heuristic search with backtracking. The system can conduct a more intelligent search and steer clear of unproductive avenues when looking for answers thanks to heuristic search (Lumbantoruan.2022).

3. Research Method

3.1. Dataset Acquisition

Gathering information on classes, instructors, classrooms, and the length of the scheduled lecture time is the first stage. The State Islamic University of North Sumatra's Computer Science Study Program is the source of this information. Among the necessary information are:

3.2. List of Classes Offered

The State Islamic University of North Sumatra's Computer Science Study Program offers the following courses. You can view the list of courses taught in this study at Tabel 2.

Table 1: List of Courses Taught

No.	Course Code	Course Name	Session Duration (Hours)	Number of Sessions
1	MK001	Matematika Diskrit	2	2
2	MK002	Struktur Data	2	2
3	MK003	Algoritma dan Pemrograman	3	1
4	MK004	Basis Data	2	2
5	MK005	Sistem Operasi	3	1
6	MK006	Kecerdasan Buatan	2	2
7	MK007	Rekayasa Perangkat Lunak	2	2
8	MK008	Jaringan Komputer	3	1
9	MK009	Komputasi Numerik	2	1
10	MK010	Pemrograman Web	2	2

3.3. List of lecturers and time availability

This is a list of lecturers and the availability of time for the Computer Science Study Program at the State Islamic University of North Sumatra. The list of lecturers and the availability of time in this study can be seen in Table 2. Tabel 2. Daftar dosen beserta ketersediaan waktu

Tabel 2: List of names of lecturers and courses taught

No	Nama Dosen	Mata Kuliah yang Diampu
1	Muhammad Siddik, M.Kom	Sistem Operasi, Kecerdasan Buatan
2	Armansyah, M.Kom	Algoritma dan Pemrograman, Manajemen Proyek TI
3	Abdul Halim Hasugian, M.Kom	Jaringan Komputer, Struktur Data
4	Muhammad Ikhsan Rifki, ST, MT	Pemrograman Web, Pemrograman Mobile
5	Ilka Zufria, M.Kom	Matematika Diskrit, Basis Data

3.4. A list with capacity of each classroom

This is a list of classrooms and their capacities for the State Islamic University of North Sumatra's Computer Science Study Program. Table 3 shows the list of classrooms and their capacities used in this investigation.

Tabel 3: List of classrooms and their capacities

No	Kode Kelas	Nama Kelas	Jumlah Mahasiswa
1	KLS01	FST 1	30
2	KLS02	FST 2	28
3	KLS03	FST 3	32
4	KLS04	FST 4	27

3.5. A list with capacity of each classroom

This is a list of classrooms and their capacities for the State Islamic University of North Sumatra's Computer Science Study Program. Table 3 shows the list of classrooms and their capacities used in this investigation.

Table 4. List of time for each course

Slot 1 (08.00-11.00)	Slot 2 (11.00-13.00)	Slot 3 (13.00-15.00)	Slot 4 (15.00-17.00)	Slot 5 (17.00-19.00)
Kosong	Kosong	Kosong	Kosong	Kosong
Kosong	Kosong	Kosong	Kosong	Kosong
Kosong	Kosong	Kosong	Kosong	Kosong
Kosong	Kosong	Kosong	Kosong	Kosong
Kosong	Kosong	Kosong	Kosong	Kosong

3.6. Formulation of the Problem

Using a modified backtracking technique, the State Islamic University of North Sumatra's Computer Science Study Program creates class schedules in the manner described below:

1. Variable of Decision The study's decision variables are y_i=(classrooms, lecturers, and courses, time list of each course).The purpose of the objective This study's

2. objective function is to fill the variable x_i with y_i, yielding the objective function 〖f(y_i)=x_i.
3. Limitations These are the limitations that were used in this study: Course schedules that call for the same classroom at the same time must not conflict. Lecturers are not permitted to teach more than one course concurrently. Classrooms cannot hold more people than is specified.

4. Results and Discussion

4.1. Scheduling Tree Crration

The State Islamic University of North Sumatra's Computer Science Study Program's course schedule preparation is the subject of this study. Courses, professors, weekly meetings, and the amount of time lecturers spend teaching make up the domain that is utilized. The table lists the distinct codes and course names for each course presented in the study program. The professors on the list teach each course, and some of them teach more than one. How many meetings are held for each course is indicated by the number of meetings held each week. Depending on the course's credit requirements and material requirements, some courses call for multiple weekly meetings.The number of credits in the course determines how long lecturers teach at each meeting, with the duration breakdown being as follows: One-credit courses are single (50 minutes), two-credit courses are double (100 minutes), and three-credit courses are triple (150 minutes). For instance, because MK001 (Discrete Mathematics) has two credits and is scheduled for two meetings each week, each meeting lasts double that amount of time, or. 100 minutes. In contrast, the course MK003 (Algorithms and Programming) contains three credits and only meets once a week, meaning that each meeting lasts 150 minutes. The list of domains used in this study can be found at.

Table 5: Domain of each course

No	Mata Kuliah	Dosen	Jumlah Pertemuan Tiap Minggu	Jumlah Sesi
1	MK001	Muhammad Siddik, M.Kom	2	Double
2	MK002	Armansyah, M.Kom	2	Double
3	MK003	Abdul Halim Hasugian, M.Kom	1	Double
4	MK004	Armansyah, M.Kom	2	Double
5	MK005	Muhammad Ikhsan Rifki, ST, MT	1	Double
6	MK006	Ilka Zufria, M.Kom	2	Double
7	MK007	Muhammad Siddik, M.Kom	1	Double
8	MK008	Muhammad Ikhsan Rifki, ST, MT	1	Double
9	MK009	Ilka Zufria, M.Kom	1	Double
10	MK010	Abdul Halim Hasugian, M.Kom	2	Double

4.2. Modification of Backtracking Algorithm

Thrashing, or tracing branches without a solution, and tracing nodes not just at the expansion node but also at other points in the tree are intended to be eliminated by the backtracking method change [3]. As a result, the backtracking algorithm can be modified to eliminate at the level that doesn't yield a solution before moving on to the next level expansion. The modified backtracking algorithm's solution to the problem is then broken down into its general parts:

a. Decision variables are those that provide a complete description of the choices that need to be taken.
b. Objective Function: an equation or function that joins variables and creates a unity of what needs to be accomplished in order to provide an optimal objective function.
c. Constraints: these limitations are applied to a decision variable in order to limit its value. All of the constraints must be met by the value of the variable that optimizes the objective function.

To create a solution with the modified backtracking algorithm, follow these steps:

a. A path from the root node to the leaf nodes is formed to acquire the solution. The enlarged live node is known as the E-node (Expand-node), while the born node is known as the live node.
b. The E-node becomes a dead node if the trajectory that results from expanding it does not lead to a solution.
c. The search procedure is completed by creating a new node if the trajectory's final position is a dead node, Additionally, the search is conducted by going back to the closest vertex if no vertex is produced.
d. The vertices at that level are removed if there is still no solution, and new vertices are created at

4.3. Visualization and Evaluation

a. Scheduling Tree

Based on the domain and constraints that have been determined, a tree using a modified backtracking algorithm based on DFS and CSP can be seen in Figure 1.

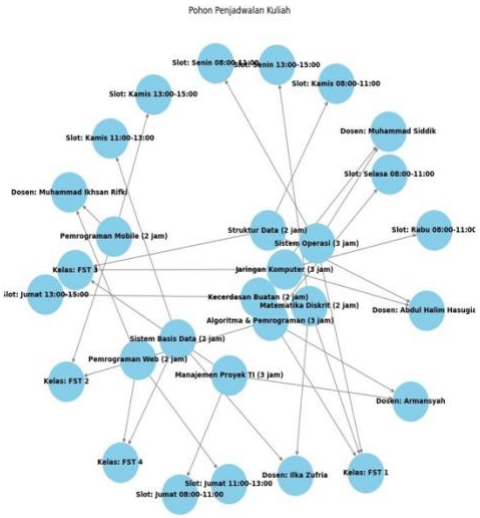

Fig. 1: Course Scheduling Tree

The relationship between time slots, courses, lecturers, and classes is depicted graphically in this tree, which is a representation of course scheduling. An essential component, including the time slot, course name, lecturer name, and class, is represented by each blue circle (node). The relationship between these elements is depicted by the connecting lines, or edges. For instance, the course "Discrete Mathematics (2 hours)" is associated with a time slot like "Monday 08:00-11:00."
It is also associated with "Class: FST 1" and "Lecturer: Muhammad Siddik." By linking each course to the implementation time, lecturer, and subsequent class, this tree illustrates how the lecture calendar is set up. Easy schedule identification, avoiding scheduling conflicts between classes, lecturers, or courses, and improving schedule management are all made possible by this depiction. This type of tree makes it easy to observe how resources are allocated and how time is distributed. Table 6 displays the outcomes of the lecture schedule for the Computer Science Study Program.

```
# Data mata kuliah default courses = [
    {"kode": "MK005", "nama": "Sistem Operasi", "durasi": 3, "dosen": "Muhammad Siddik", "kelas": "FST 1", "waktu": "Senin 08:00–11:00"},
    {"kode": "MK003", "nama": "Algoritma dan Pemrograman", "durasi": 3, "dosen": "Armansyah", "kelas": "FST 2", "waktu": "Selasa 08:00–11:00"},
    {"kode": "MK008", "nama": "Jaringan Komputer", "durasi": 3, "dosen": "Abdul Halim Hasugian", "kelas": "FST 3", "waktu": "Rabu 08:00–11:00"},
     {"kode": "MK010", "nama": "Pemrograman Web", "durasi": 2, "dosen": "Muhammad Ikhsan Rifki", "kelas": "FST 4", "waktu": "Jumat 11:00–
13:00"},
    {"kode": "MK001", "nama": "Matematika Diskrit", "durasi": 2, "dosen": "Ilka Zufria", "kelas": "FST 1",
   waktu": "Senin 13:00–15:00",
       {"kode": "MK002", "nama": "Struktur Data", "durasi": 2, "dosen": "Abdul Halim Hasugian", "kelas": "FST 3", "waktu": "Kamis 08:00–11:00"},
        {"kode": "MK004", "nama": "Sistem Basis Data", "durasi": 2, "dosen": "Ilka Zufria", "kelas": "FST 4",  "waktu": "Kamis 11:00–13:00"},
        {"kode": "MK009", "nama": "Pemrograman Mobile", "durasi": 2, "dosen": "Muhammad Ikhsan Rifki", "kelas": "FST 2", "waktu": "Kamis 13:00–
   15:00"},
       {"kode": "MK007", "nama": "Manajemen Proyek TI", "durasi": 3, "dosen": "Armansyah", "kelas": "FST 3", "waktu": "Jumat 08:00–11:00"},
        {"kode": "MK006", "nama": "Kecerdasan Buatan", "durasi": 2, "dosen": "Muhammad Siddik", "kelas": "FST 1", "waktu": "Jumat 13:00–
   15:00"},
   ]
   # Hari dalam seminggu
   days = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat"]

   # Fungsi untuk menyusun jadwal per hari def
   organize_schedule(courses):
       weekly_schedule = {day: [] for day in days} for course in courses:
           day = course["waktu"].split()[0]  # Ambil nama hari dari waktu weekly_schedule[day].append(course)
       return weekly_schedule
   # Fungsi untuk mencetak jadwal
   def print_schedule(weekly_schedule):
       print("Jadwal Mata Kuliah Selama Seminggu:\n") for day in days:
           print(f"{day}:")
           if weekly_schedule[day]:
               for course in weekly_schedule[day]:
                   print(f"  - {course['nama']} ({course['durasi']} jam)") print(f"Dosen:
                   {course['dosen']}")
                   print(f"       Kelas: {course['kelas']}")
                   print(f"       Waktu: {course['waktu']}")
           else:
               print("  Tidak ada jadwal.") print()
```

The program using pyhton programming language with google colab organizes and displays the lecture schedule by combining important elements such as time, course, lecturer, and class in a tree-based structure. Using a dictionary, the organize_schedule (courses) function groups courses by day, and print_schedule (weekly_schedule) displays the schedule based on the relationships that have been built. In addition to preventing schedule conflicts, the presentation facilitates time management and resource allocation.

```
Data Mata Kuliah:
    kode                     nama  durasi                dosen  kelas  \
0  MK005           Sistem Operasi       9       Muhammad Siddik  FST 1
1  MK003  Algoritma dan Pemrograman       9             Armansyah  FST 2
2  MK008         Jaringan Komputer       9  Abdul Halim Hasugian  FST 3
3  MK010           Pemrograman Web       8  Muhammad Ikhsan Rifki  FST 4
4  MK001       Matematika Diskrit       8           Ilka Zufria  FST 1
5  MK002            Struktur Data       8  Abdul Halim Hasugian  FST 3
6  MK004        Sistem Basis Data       8           Ilka Zufria  FST 4
7  MK009       Pemrograman Mobile       8  Muhammad Ikhsan Rifki  FST 2
8  MK007      Manajemen Proyek TI       9             Armansyah  FST 3
9  MK006        Kecerdasan Buatan       8       Muhammad Siddik  FST 1
```

```
Kamis:
  - Struktur Data (2 jam)
    Dosen: Abdul Halim Hasugian
    Kelas: FST 3
    Waktu: Kamis 08:00-11:00
  - Sistem Basis Data (2 jam)
    Dosen: Ilka Zufria
    Kelas: FST 4
    Waktu: Kamis 11:00-13:00
  - Pemrograman Mobile (2 jam)
    Dosen: Muhammad Ikhsan Rifki
    Kelas: FST 2
    Waktu: Kamis 13:00-15:00

Jumat:
  - Pemrograman Web (2 jam)
    Dosen: Muhammad Ikhsan Rifki
    Kelas: FST 4
    Waktu: Jumat 11:00-13:00
  - Manajemen Proyek TI (3 jam)
    Dosen: Armansyah
    Kelas: FST 3
    Waktu: Jumat 08:00-11:00
  - Kecerdasan Buatan (2 jam)
    Dosen: Muhammad Siddik
    Kelas: FST 1
    Waktu: Jumat 13:00-15:00


Data terbaru telah disimpan ke Excel.
```

```
                waktu
0   Senin 08:00-11:00
1  Selasa 08:00-11:00
2    Rabu 08:00-11:00
3   Jumat 11:00-13:00
4   Senin 13:00-15:00
5   Kamis 08:00-11:00
6   Kamis 11:00-13:00
7   Kamis 13:00-15:00
8   Jumat 08:00-11:00
9   Jumat 13:00-15:00
Jadwal Mata Kuliah Selama Seminggu:

Senin:
  - Sistem Operasi (3 jam)
    Dosen: Muhammad Siddik
    Kelas: FST 1
    Waktu: Senin 08:00-11:00
  - Matematika Diskrit (2 jam)
    Dosen: Ilka Zufria
    Kelas: FST 1
    Waktu: Senin 13:00-15:00

Selasa:
  - Algoritma dan Pemrograman (3 jam)
    Dosen: Armansyah
    Kelas: FST 2
    Waktu: Selasa 08:00-11:00

Rabu:
  - Jaringan Komputer (3 jam)
    Dosen: Abdul Halim Hasugian
    Kelas: FST 3
    Waktu: Rabu 08:00-11:00
```

Table 6: Final Results of Computer Science Study Program Course Schedule, State Islamic University of North Sumatra

HARI	RUANG			
	KLS01	KLS02	KLS03	KLS04
Senin	Sistem Operasi - Muhammad Siddik (08.00-11.00) 3 Jam			
		Matematika Diskrit – Ilka Zufria (13.00-15.00) 2 Jam		
Selasa	Algoritma dan Pemrograman - Armansyah (08.00-11.00) 3 Jam			
Rabu	Jaringan Komputer – Abdul Halim Hasugian (08.00-1.00) 3 Jam			
Kamis	Struktur Data – Abdul Halim Hasugian (08.00-11.00) 2 Jam			
				Basis Data – Ilka Zufria (11.00-13.00) 2 Jam
		Pemrograman Mobile – Muhammad Ikhsan Rifki (13.00-1500) 2 Jam		
Jumat			Manajemen Proyek TI – Armansyah (08.00-11.00) 3 Jam	
				Pemrograman Web – Muhammad Ikhsan Rifki (11.00-13.00) 2 Jam
	Kecerdasan Buatan – Muhammad Siddik (13.00-15.00) 2 Jam			

The balance between academic and practical courses has been taken into consideration when creating this lecture program. Each course has a sufficient amount of time, however certain days—like Monday and Friday— are busier than others. Although there are difficulties in managing time, particularly on days when there are many lecture sessions, this schedule enables students to thoroughly examine a variety of topics. Although it still necessitates effective time management, the timetable generally supports students' academic needs rather well.

5. Conclusion

This lecture schedule's conclusion demonstrates how a well-planned time allocation can help students meet their academic objectives. Students can acquire a thorough understanding of a range of computer science topics with a range of courses that include both theory and practice, as well as well-balanced schedules. Overall, this plan enables students to maximize their learning experience, despite the fact that there are some days with strict restrictions.

References

[1] Abidin, M. Z., Wiranto, & Setiadi, H. (2019). Course scheduling using dynamic-order backjumping (Case study: STIKA Madiun). *ITSMART, 7*(2), 101-107.

[2] Adiguna, Y., & Swanjaya, D. (2020). Implementasi algoritma backtracking untuk mencari jalan keluar labirin. Dalam Sucipto (Ed.), *Pengembangan sains dan teknologi untuk pembangunan yang berkelanjutan* (Prosiding SEMNAS INOTEK, 4(3), 131-136). Kediri: Prosiding SEMNAS INOTEK.

[3] Ferdinal, R. (2021). Penerapan algoritma backtracking dan metode constraint satisfaction untuk penjadwalan. Ginsberg, M. L. (2020). Dynamic backtracking. *Journal of Artificial Intelligence Research, 1*, 25-46.

[4] Handayani, D., Rosely, E., & Mayadewi, R. P. (2016). Aplikasi penjadwalan mata pelajaran dengan pewarnaan graf menggunakan algoritma Welch-Powell studi kasus: X MIPA SMA Negeri 8 Bandung. *E-Proceeding of Applied Science, 2*(3), 933-935.

[5] Lumbantoruan, R. (2022). Penjadwalan kuliah dengan algoritma backtracking. *Politeknik Informasi, 12*. Makarim, F. (2024). Analisis penggunaan algoritma backtracking dalam penjadwalan kuliah. *Jurnal Informatika*

[6] *Institut Teknik Bandung*, 5-6.

[7] Munarto, R. (2019). Perancangan sistem penjadwalan kuliah di jurusan teknik elektro FT. UNTIRTA menggunakan teknik pewarnaan graph algoritma backtracking Welch-Powell. *Seminar Nasional Inovasi Teknologi*, 278. Munir, R. (2020). *Matematika diskrit*. Informatika Bandung.
[8] Putra, D. N. (2023). Penerapan dan implementasi algoritma backtracking. *Departemen Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Bandung (ITB)*, 10-12.
[9] Rosmasari, D., Dengen, N., & Chandra, F. (2018). Implementasi algoritma constraint satisfaction problems pada
[10] sistem penjadwalan mata kuliah. *Jurnal Ilmu Pengetahuan dan Teknologi Komputer, 3*(2), 169-176.
[11] Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach*. Pearson Education. Rosen, K. (2020). *Discrete mathematics and its applications* (7th ed.). McGraw-Hill.
[12] Sirait, R. B. (2020). Perancangan aplikasi game labirin dengan. *Pelita Informatika Budi Darma*, 100.
[13] Siregar, N. (2024). Penerapan algoritma backtracking dalam penyelesaian masalah. *JUSINFO (Jurnal Sains dan Informatika), 2*.
[14] Wulandari, D. Y. (2020). *Representasi pohon dari graf kordal bipartisi* (Tesis, Universitas Sumatera Utara). Medan.
[15] Wulandari, R. (2022). Penjadwalan mata kuliah dengan modifikasi. *Buletin Ilmiah Math. Stat. dan Terapannya (Bimaster)*, 209.