UNIVERSITY TIMETABLE SCHEDULING

USING PSO

• OVERVIEW

01 — Project idea in detail

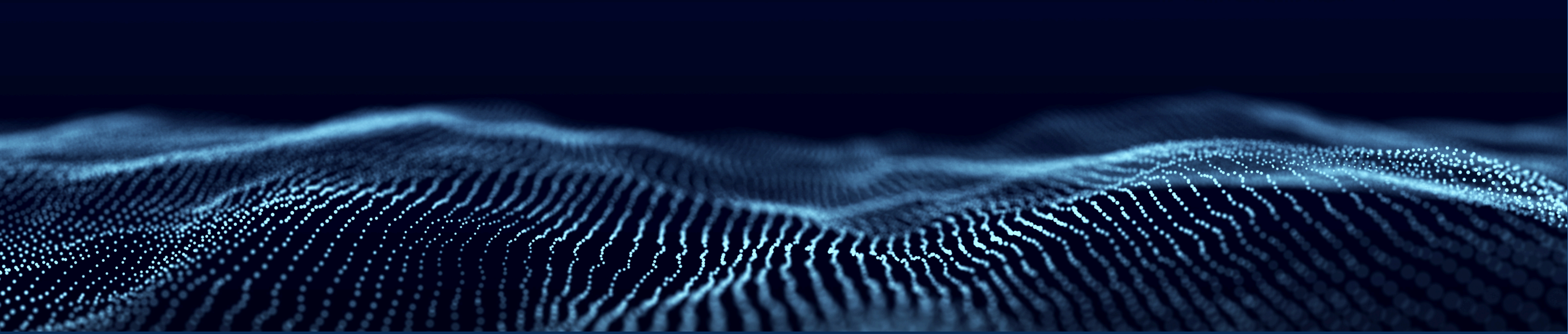
02 — Main functionalities

03 — Similar applications

04 — Academic publications

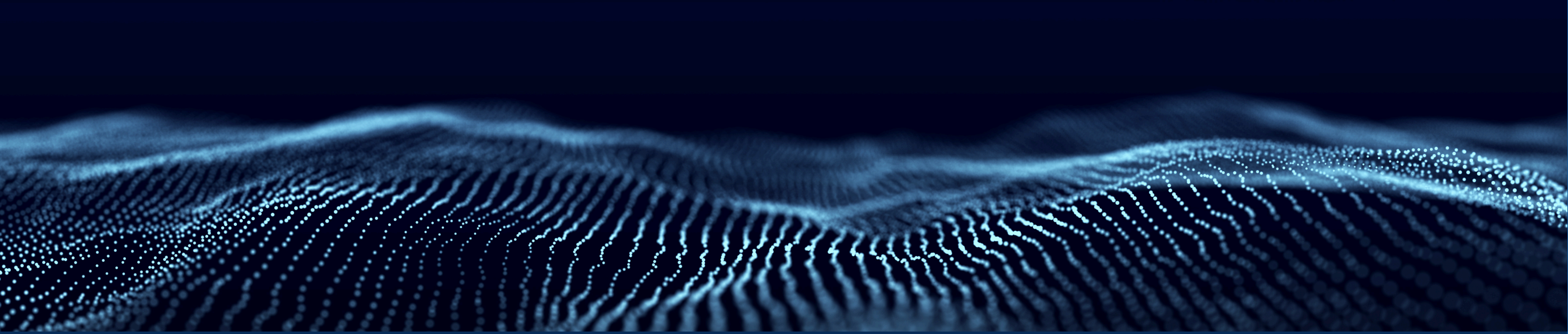
05 — Algorithm Details

06 — Development platform



1. PROJECT IDEA IN DETAIL

- The university timetable scheduling problem involves the complex task of efficiently assigning courses, classes, exams, and other academic activities to specific time slots and classrooms within a university's schedule. This task is compounded by various constraints, including room capacities, faculty availability, student preferences, and course prerequisites. The primary objective is to create a timetable that minimizes conflicts, maximizes resource utilization, and satisfies all constraints and requirements.



- Achieving this goal requires careful consideration of the diverse needs and preferences of students, faculty, and administrative staff, while also optimizing the allocation of limited resources such as classrooms and faculty time. Additionally, the problem often involves multiple stakeholders and competing objectives, necessitating a sophisticated optimization approach such as Particle Swarm Optimization (PSO) to navigate the vast search space and converge towards an optimal or near-optimal solution.

2.MAIN FUNCTIONALITIES

- **TIMETABLE CLASS**

1) Update current position:

   - Updates the current position based on the velocity.
   - Clips the position to ensure it stays within valid bounds.

2) Update Velocity:

   - Updates the velocity using the PSO formula with cognitive and social componen

3) Update Personal Best Position and Fitness:

- Updates the personal best position and fitness if a better solution is found.

4) Update Velocity:

- Calculates the fitness of the current position based on various constraints such as course conflicts, instructor conflicts, and department constraints

- **PSO FUNCTION:**

1) Timetable Initialization:

- Initializes the PSO algorithm with given population size and maximum iterations.
- Initializes random timetables and their velocities.

2) Optimization Loop:

- Iterates through the specified number of iterations.
- Updates velocity and position for each timetable.
- Updates personal and swarm best positions and fitness values.

3.SIMILAR APPLICATIONS IN THE MARKET

- **OPTIMOROUTE**

OptimoRoute is a routing and scheduling software designed to optimize delivery and service routes for businesses. It's particularly useful for companies with fleets of vehicles, delivery services, or field service operations

3.SIMILAR APPLICATIONS IN THE MARKET

- **TIMETABLER**

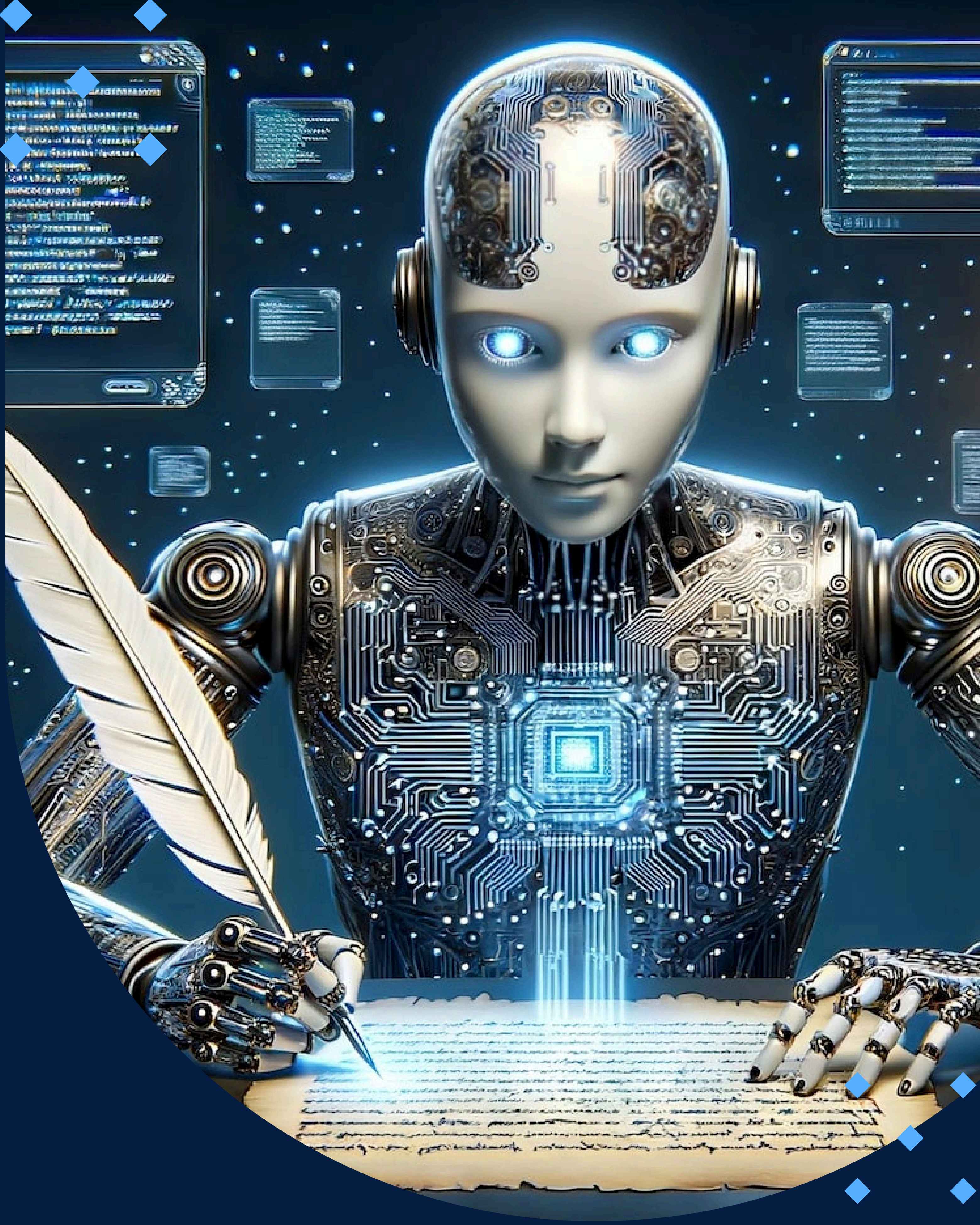
Modern software can relieve you of much of the donkey-work of timetabling, leaving you with more time to apply your skill and judgement where they are needed in order to produce a timetable of the highest quality. This page gives you an introduction to our widely-used timetabling program called TimeTabler. This extremely popular and proven computer software will schedule your timetable for you.

3.SIMILAR APPLICATIONS IN THE MARKET

- **QUICKSCHOOLS**

  Specify how many times each course is taught during the week, and with just one click, the automated scheduler will expertly distribute those classes into available time slots in your schedule. Completely conflict free!

4.LITERATURE REVIEW OF ACADEMIC PUBLICATIONS



Doctor of technical sciences, professor of Ukrainian State Academy of Railway Transport, Kharkov
Ref: Ukrainian State Academy of Railway Transport,

PAPER 1

THE THEORY OF NP-COMPLETE PROBLEMS

In this paper, we embark on an exploration of the intricate realm of computational complexity theory, focusing particularly on the enigmatic domain of NP-complete problems and their profound implications for optimization tasks. At the heart of our inquiry lies the fundamental question of the relationship between the complexity classes P and NP, a conundrum that has captured the imagination of mathematicians and computer scientists alike for decades.

The genesis of this inquiry can be traced back to the seminal works of Stephen Cook and Leonid Levin, who laid the groundwork for the theory of NP-completeness by introducing the concept of NP-complete problems and establishing the famed Cook-Levin theorem, which identified the satisfiability (SAT) problem as the prototypical NP-complete problem. Subsequent contributions by luminaries such as Richard Karp further elucidated the landscape of NP-completeness, revealing a rich tapestry of computational challenges with far-reaching implications.

PAPER 1 CONT...

However, recent developments, notably the works of Vineya Deolalikar and Anatoly D. Plotnikov, have cast doubt on the prevailing understanding of NP-completeness, challenging the validity of certain proofs and suggesting potential discrepancies between the classes P and NP. Our study seeks to critically examine these assertions in light of prior research, offering a nuanced perspective on the intricate interplay between polynomial reducibility, algorithmic solvability, and the classification of NP-complete problems.

lCentral to our analysis is a deep dive into the theory of NP-complete problems, where we unravel the intricacies of polynomial-time algorithms, nondeterministic computation, and the concept of "checkability." Through a meticulous examination of Cook's theorem and its implications for polynomial reducibility within NP-complete classes, we aim to shed light on the complexities inherent in these computational problems and the challenges they pose for algorithmic theory.

PAPER 1 CONT...

Moreover, our inquiry extends beyond theoretical abstractions to explore practical implications and real-world applications of NP-complete problems, particularly in the domains of discrete optimization and graph theory. By highlighting the ubiquity of NP-complete problems in various computational domains and their relevance to modern-day challenges, we underscore the importance of a robust understanding of computational complexity theory in addressing complex optimization tasks.

In conclusion, our paper advocates for a holistic approach to the study of NP-completeness, one that synthesizes theoretical insights with practical considerations and acknowledges the nuances and complexities inherent in these computational problems. By critically evaluating recent assertions and building upon prior research, we aim to contribute to a deeper understanding of the profound mysteries surrounding NP-completeness and its implications for algorithmic theory and practice.

Faculty of Computer Science and Information Technology,
Universiti Malaysia Sarawak, 94300, Sarawak, Malaysia
Ref:Content uploaded by Universiti Malaysia Sarawak

PAPER 2

Heuristic Algorithm for Multi-Location Lecture Timetabling

1. Detailed Algorithm Explanation: Provide a more in-depth explanation of the two-stage heuristic algorithm proposed in the paper. Break down each stage further, describing the specific steps involved, the rationale behind them, and how they contribute to generating a feasible timetable.
2. Comparison with Other Approaches: Expand the discussion on how the proposed two-stage heuristic algorithm compares with other existing methods for solving university timetabling problems. This could include a more comprehensive literature review covering various heuristic algorithms, meta-heuristics, and optimization techniques used in similar contexts.
3. Real-World Case Studies: Include additional real-world case studies or examples beyond those mentioned in the paper. These could involve different universities, faculties, or departments facing similar timetabling challenges, providing more diverse scenarios for evaluation and validation of the proposed algorithm.

PAPER 2 CONT...

4. Scalability and Efficiency Analysis: Discuss the scalability of the proposed algorithm in handling larger datasets and more complex timetabling constraints. Analyze its efficiency in terms of computational resources required and the time taken to generate timetables for different problem sizes.
5. Impact on Resource Utilization: Explore the impact of the proposed algorithm on resource utilization within the academic institution. This could include factors such as lecturer workload distribution, classroom allocation optimization, and overall cost-effectiveness of the timetabling process.
6. User Interface and Implementation Details: Provide insights into the practical implementation of the proposed algorithm, including considerations for developing a user-friendly interface or integrating it into existing timetabling software used by universities. Discuss any challenges encountered during implementation and potential solutions.
7. Future Research Directions: Identify potential avenues for future research in the field of university timetabling optimization. This could involve addressing specific constraints not covered by the current algorithm, exploring new heuristic techniques, or integrating machine learning approaches for improved performance.

PAPER 3

Ref:Content uploaded by Shu-Chuan Chu

TIMETABLE SCHEDULING USING PARTICLE SWARM OPTIMIZATION

In this paper, our focus revolves around harnessing the power of Particle Swarm Optimization (PSO) to tackle the intricate task of timetable scheduling. Traditional PSO methods are augmented in our proposed approach to incorporate self-mutated particles. These particles undergo adaptive changes wherein two slots are randomly altered, thus facilitating an enhanced exploration of both global and local solution spaces.

The methodology commences with the initialization of a particle population, each representing a candidate solution. These solutions are then evaluated based on their performance, following which the best positions are iteratively updated. Subsequently, timetables for the succeeding generation are formulated through a multi-step process that involves modifying particles based on the information gleaned from both local and global best solutions.

PAPER 3 CONT...

Throughout the experimentation phase, the performance of our PSO-based approach is meticulously evaluated. Penalty scores, indicative of clashes within the generated timetables, are computed alongside the time taken for scheduling varying numbers of exams. The results showcase the efficacy of our methodology in producing clash-free timetables, thus underscoring the utility of PSO in the context of scheduling optimization. Furthermore, this study serves as a testament to the continuous evolution and refinement of scheduling algorithms. By building upon prior research, we not only demonstrate the applicability of PSO but also hint at its potential for further enhancements. Specifically, we propose future endeavors that involve amalgamating PSO with tabu search techniques, aiming to bolster the efficiency and effectiveness of timetable scheduling.

In summary, this research contributes to the body of knowledge surrounding optimization methodologies, particularly within the realm of scheduling. By leveraging the inherent capabilities of PSO and exploring avenues for refinement, we pave the way for more robust solutions to the complex challenges inherent in timetable scheduling.

Department of Mathematics, Punjabi University, Patiala 147002, Punjab, India
2 University College, Chunni Kalan, Dist. Fatehgarh Sahib, Patiala 140307, Punjab

PAPER 4

Ref:Content uploaded by Mathematics, Punjabi University

AN OVERVIEW OF VARIANTS AND ADVANCEMENTS OF PSO ALGORITHM

In this paper, we aim to provide a comprehensive overview of the Particle Swarm Optimization (PSO) algorithm, its variants, and recent advancements in the field of swarm-based optimization techniques. PSO stands as a prominent method in optimization, drawing inspiration from the collective behavior of natural swarms, particularly bird flocks. Since its introduction in 1995, PSO has garnered widespread attention from researchers due to its simplicity, effectiveness, and versatility across various domains.

At its core, PSO involves a population of potential solutions, referred to as particles, which iteratively adjust their positions in the search space to optimize a given objective function. Each particle maintains its own best-known position (pbest) and the globally best-known position (gbest) discovered by any particle in the swarm. Through social interaction and self-experience, particles dynamically adapt their movements, seeking to strike a balance between exploration of new solution regions and exploitation of known promising areas.

PAPER 4 CONT...

The success of PSO hinges on careful parameter selection, including population size, neighborhood size, acceleration coefficients, velocity clamping, and inertia weight. These parameters profoundly influence the algorithm's convergence speed, solution quality, and ability to navigate complex optimization landscapes. As such, researchers have devoted considerable effort to fine-tuning these parameters and exploring their effects on PSO's performance.

In addition to parameter tuning, recent research has focused on advancing the PSO algorithm in several directions. Modifications of the original PSO have been proposed to address shortcomings such as premature convergence and poor exploration-exploitation balance. Extensions of PSO's applications have expanded its utility beyond traditional optimization problems to include diverse domains such as engineering design, data mining, and image processing.

Furthermore, theoretical analyses have delved into the underlying principles of PSO, providing insights into its convergence properties, stability, and scalability. These analyses have contributed to a deeper understanding of PSO's behavior and informed the development of more effective variants and hybridization strategies.

PAPER 4 CONT...

Hybridization, in particular, has emerged as a promising approach to enhance PSO's performance by combining it with other optimization techniques, such as evolutionary algorithms, simulated annealing, or local search methods. By leveraging the strengths of multiple algorithms, hybrid PSO approaches aim to achieve superior optimization results while mitigating each method's individual weaknesses.

Overall, this paper aims to offer a comprehensive exploration of PSO and its variants, elucidating fundamental concepts, parameter considerations, recent advancements, and future research directions. By synthesizing existing knowledge and identifying key research trends, we hope to contribute to the ongoing evolution and refinement of PSO as a powerful optimization tool in various fields.

Research Group (IASRG), School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310, Johor, Malaysia

PAPER 5

Ref:Content uploaded by Universiti Teknologi Malaysia

OPTIMIZATION TECHNIQUES IN UNIVERSITY TIMETABLING PROBLEM: CONSTRAINTS, METHODOLOGIES, BENCHMARKS, AND OPEN ISSUES

In this paper, we embark on a journey through the landscape of academic literature surrounding university timetabling problems. These problems, perennial challenges faced by educational institutions each semester, fall within the realm of NP-COP, characterized by their non-polynomial complexity and combinatorial optimization nature. The quest for an optimal timetable amidst a myriad of constraints necessitates the utilization of sophisticated optimization algorithms.

Our exploration begins by elucidating the essence of university timetabling, dissecting it into two distinct categories: University Course Timetabling and University Examination Timetabling. The former involves assigning courses to specific times and rooms, while the latter focuses on scheduling examinations to avoid conflicts and ensure logistical feasibility. Both domains present unique challenges and require tailored solutions.

PAPER 5 CONT...

As we delve deeper into the literature, we encounter a plethora of optimization techniques employed to tackle university timetabling problems. Among these, meta-heuristic methodologies emerge as prominent contenders, leveraging principles from evolutionary computation and swarm intelligence to navigate the vast solution space. The efficacy of meta-heuristics lies in their ability to strike a balance between exploration and exploitation, traversing the search landscape in search of optimal solutions.

Furthermore, our review sheds light on the intricate web of constraints that ensnare university timetabling endeavors. These constraints, ranging from room capacities to faculty preferences, delineate the boundaries within which feasible timetables must be crafted. Balancing hard constraints, which are non-negotiable, with soft constraints, which allow for some degree of flexibility, poses a formidable challenge to optimization algorithms.

PAPER 5 CONT...

Benchmarking emerges as a crucial aspect of evaluating the performance of optimization algorithms. By subjecting algorithms to standardized benchmark datasets, researchers can glean insights into their strengths and weaknesses. However, the dearth of comprehensive benchmark datasets tailored to university timetabling remains a pressing issue, hampering efforts to compare and assess different methodologies effectively.

Looking ahead, we identify several avenues for future research and development in the field of university timetabling. The exploration of novel optimization algorithms, such as those inspired by biologically-inspired paradigms like Biogeography-Based Optimization and Grey Wolf Optimizer, holds promise for unlocking new frontiers in timetabling efficiency. Moreover, the integration of multi-objective optimization approaches promises to yield robust solutions capable of balancing competing objectives.

PAPER 5 CONT...

In conclusion, our comprehensive review serves as a beacon guiding researchers, practitioners, and students through the labyrinthine landscape of university timetabling. By illuminating key concepts, methodologies, and challenges, we aim to catalyze further advancements in this vital area of academic research.

5.ALGORITHM DETAILS:

- Particle Swarm Optimization(PSO) Algorithm:

Initialization:

The algorithm starts by initializing a population of timetables with random positions, where each position represents a possible timetable arrangement.
Each timetable's position consists of course identifiers distributed across timeslots and classrooms.

Fitness Calculation:

The fitness of each timetable is calculated based on certain constraints and penalties associated with the timetable arrangement

Penalties are incurred for violations such as:

- A course being assigned more than once in a single timeslot.
- A course being assigned more than once in a week.
- Courses with similar departments being assigned in the same timeslot.
- Instructors being assigned to more than one class simultaneously in the same timeslot.

5.ALGORITHM DETAILS:

- Particle Swarm Optimization(PSO) Algorithm:

PSO Optimization:

The PSO algorithm iterates through a predefined number of iterations.
At each iteration, the velocity and position of each timetable (particle) are updated based on its current position, velocity, and the best positions found so far by itself and the swarm.
The best fitness value found by the swarm and the corresponding timetable position are recorded.

GUI Interface:

A GUI interface allows users to input PSO parameters such as population size, maximum iterations, inertia weight (w), cognitive coefficient (c1), and social coefficient (c2).
Upon running the PSO algorithm from the GUI, the algorithm executes with the provided parameters, and the results are displayed.

5.ALGORITHM DETAILS:

- Experiment Results:

- The PSO algorithm aims to minimize the total penalties associated with the timetable arrangements.
- The best fitness value found by the swarm is printed at the end of the optimization process.
- Additionally, the timetable arrangement corresponding to the best fitness value is displayed, showing the assigned courses and instructors for each timeslot and classroom
- A plot of the evolution of the best fitness value over iterations is also generated, providing insights into the optimization process's convergence behavior.

6.DEVELOPMENT PLATFORM:

Programming Language — Python

Pycharm — Editor

GUI Library — Tkinter

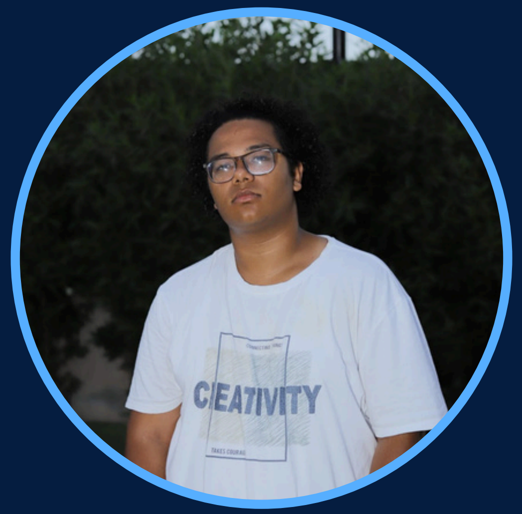
matplotlib — Plotting library

OUR TEAM

Mohab Mohamed

Yasir khaled

Abdelrahman Deraz



Youssef Said

Shehab Orban

Thanks...