A Multi-Swarm Particle Swarm Optimization Solution for the University Course Timetabling Problem

Gian Myrl D. Renomeron
CMSC 199.1: Research in Computer Science I
Division of Natural Sciences and Mathematics
University of the Philippines Tacloban College
gdrenomeron@up.edu.ph

A Multi-Swarm Particle Swarm Optimization Solution for the University Course Timetabling Problem

1 Introduction

Academic institutions face the significant challenge of efficient resource management in their administrations without compromising the quality of the educational experience in this dynamic world of academia. [1] Among the most complicated logistical problems that universities encounter is the so-called *University Course Timetabling Problem* (UCTP). [2] [3] Essentially, the scheduling of courses, instructors, and students into particular time slots and rooms will match a variety of constraints, including room capacities, faculty availability, and course prerequisites, among others. [4] It is quite a challenging task to prepare the perfect schedule, especially for larger institutions where manual methods often result in conflicts and inefficiencies. [2]

Latest advanced computational approaches emerged to address these problems, and among them *Swarm Intelligence* (SI) is one of the most promising directions. [5] Among the algorithms of this group, there is an attention-grabber - *Particle Swarm Optimization* (PSO) [6], distinguished by adaptability and efficiency in searching large solution spaces. [7] [8] The algorithm is represented by a set of particles, which are possible solutions; each particle moves in the search space, and its position changes according to individual and collective experiences. [9] Still, despite many strengths, PSO fails at times to solve problems efficiently within UCTP, often failing to find the best solution under tight constraints. [3]

Multi-Swarm Optimization (MSO) is designed to split up the main swarm into smaller, specialized sub-swarms that concurrently operate in exploiting different regions of the solution space simultaneously. [10] It aims to improve the capabilities of PSO by focusing on issues such as search diversity and the global optimizing ability of the algorithm. [11] One of such particular modifications of the above-mentioned technique is MSPSO: *Multi-Swarm Particle Swarm Optimization*, which dynamically varies the sub-swarms in the standard PSO in both exploration and exploitation for a good performance to provide capabilities of global and local search. [11] To make this approach balanced, mechanisms for exploration and exploitation are integrated appropriately such that different solution domains would be explored in detail while optimizing the potential solutions in a balanced manner. [12]

This paper introduces a new approach to solving the UCTP based on multi-swarm PSO only. MSPSO is specially designed for university course timetabling with diverse constraints, while search methods are efficient in finding optimal timetables. This approach promises to provide schedules that are conflict-free as well as resource-efficient and meet the specific needs of educational institutions. This effort aims to contribute towards scalable and realistic automation and optimization of timetabling, responding to one of the most complex problems in academic administration.

1.1 Background of the Study

University Course Timetabling Problem (UCTP)

The *University Course Timetabling Problem* (UCTP) is defined as a problem of making assignments for courses, instructors, and students to particular time slots and rooms according to numerous constraints, for example room capacities and instructor availability.[1] [3] [7] UCTP is a classic problem of academic optimization, and quality methods are demanded for efficient scheduling.[2] [13] [14]

Swarm Intelligence

Swarm Intelligence refers to bio-inspired algorithms based on the collective behavior of animals such as ants, bees, or birds. [15] [16] It is applied to optimization problems because it can efficiently and adaptively search very large solution spaces. [3] [15]

Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is the best known among SI algorithms. [17] It simulates the motions of particles, which denote solutions, in the search space and updates them based on individual and collective experiences. [6] [8] [18] PSO performs very well on many optimization problems, such as UCTP, but still struggles with tricky constraints. [3] [7]

Multi-Swarm Optimization (MSO)

Multi-Swarm Optimization (MSO) is an advanced extension of swarm intelligence algorithms. [19] The basic idea in MSO is to divide the primal swarm into multiple sub-swarms that scan concurrently the different areas of the solution space. A concurrent search makes the algorithm have a better opportunity to discover global optima with effective balancing between exploration and exploitation. [10].

Multi-Swarm Particle Swarm Optimization (MSPSO)

Multi-Swarm Particle Swarm Optimization (MSPSO) is a PSO algorithm developed to overcome the lack of diversity of search and local optima avoidance in the process of complex optimization problems, by splitting the main swarm into a considerable number of sub-swarms that explore different regions in parallel with the solution space, hence, it promotes diversity and the likelihood of finding the global optima. [19] The difference gives the algorithm an excellent improvement in searching for such solutions for complex problems, such as the UCTP, which balanced exploration and exploitation would provide. [11] [20]

1.2 Related Works

In the field of UCTP, metaheuristic approaches have been studied for extensive potential towards providing valid and effective solutions with respect to constraints. Metaheuristics aim at establishing quick but effective solutions by approximating optimal schedules rather than executing exhaustive searches, which can sometimes be very computationally expensive.

A two-stage very popular heuristic strategy by Bong Chia Lih et al. [13] gives the first stage to be the grouping of all courses that can be held at the same time, and the second stage to be the assignment of time slots and venues for these groups. Meanwhile, the approach lowers the complexity of this problem, and it has already been successfully applied using real university data and it can handle real-world constraints very efficiently. Similar to Zhang et al., [1] an efficient greedy heuristic algorithm handles room and time-slot assignments. Thus, the presented approach, emphasizing the predefined constraints, can meet the required flexibility by these timetables and institutional requirements.

Further, Yanming Yang et al. [14] proposed a genetic algorithm with self-adaptive crossover and mutation probabilities, which along with a preservation strategy, led to optimal schedules. Although technically in the category of genetic algorithms, a technique based on adaptive nature is close to metaheuristic techniques and thus highly efficient for timetabling in dynamic environments such as military academies. Alves et al. [21] further enhanced the genetic approach by developing a recursive GA that maximizes efficiency in course timetabling when real-world constraints and evolving scheduling demands are considered.

Other metaheuristic methods applied in solving UCTP include Particle Swarm Optimization (PSO). PSO is another optimization technique that is increasingly gaining acceptance in solving UCTP problems. The concept was derived from the social behavior a swarm of birds or of fish follows. This makes it possible to adopt the efficient exploration of large search spaces. Oswald and Durai [3] proposed a hybrid PSO with strategies for improving search applied particularly for UCTP. This was well balanced between exploration and exploitation so the robust timetabling solutions in reference to Oswald et al 2013. Similarly, Chen and Shih [7] created a constriction PSO model integrated with local search strategies to enhance both the gain in generation speed of solutions and the quality of the solution obtained. It has shown the effectiveness of PSO in solving challenging scheduling problems.

Generalizing the metaheuristic methods embraced here, such as greedy algorithms, adaptive genetic strategies, and PSO-based techniques, indicate that these are feasible for solving UCTP. In fact, these methods are not a global optimality guarantee; however, speed and flexibility in an ideally minimal number of computations make them highly valuable tools for educational institutions, especially for large-scale tasks where a large number of constraints are to be balanced.

2 Statement of the Problem

The UCTP problem is quite complex and key for any institution of learning because it considers assigning courses to specific rooms and time slots while considering constraints such as room capacities, course prerequisites, and faculty courses. [2] To this date, most universities still consider non-optimal manual or heuristic methods, and the inefficiencies include uneven teaching loads and a mismatch between qualifications and requirements in course teaching. [3] [9] While the complexity of the problem is increased, it happens to be highly infeasible to find the best solutions, especially when there are conflicting constraints that need to be fulfilled.

The existing work [3] [8] [7] used PSO-based techniques in solving UCTP but highly developed approaches are required to be found due to the complexity of the problem, such as MSPSO, this improves the original PSO technique by splitting the swarm into many sub-swarms that simultaneously start exploring different regions of solution space. This allows for a higher degree of diversity in search and avoids getting trapped in the local optimum for a better possibility of finding the global optimum.

This paper focuses on the development of a solution for UCTP using MSPSO, that is, to explore the vast solution space with much greater efficiency than before, paying attention to constraints like room capacities, course prerequisites, and scheduling conflicts. In terms of metrics like balance of workloads, course-faculty alignment, and computational efficiency, this proposed approach is supposed to bring about improvements in quality and fairness in the courses scheduled within institutions.

3 Objectives of the Study

3.1 General Objective

To design and develop a Multi-Swarm Particle Swarm Optimization (MSPSO) algorithm to solve the University Course Timetabling Problem (UCTP).

3.2 Specific Objectives

1. To examine how the MSPSO algorithm is used and how well it solves the UCTP.

2. To model the UCTP with crucial characteristics like room availability, course prerequisites, and faculty courses within the framework of MSPSO.

3. To perform a state-of-the-art analysis by contrasting the MSPSO approach with current techniques used in the UCTP

4. To deploy a web-based application that uses the MSPSO approach for generating university course timetabling.

4 Scope and Limitation

In the context of solving the UCTP problem, the focus is on the implementation of the MSPSO algorithm. The study will execute its experiment using the data instance of a particular UCTP, thereby limiting its generalizability to different timetabling scenarios, not necessarily on the exact data set. It does not include dynamic factors that may unexpectedly appear in the scheduling stage. Examples of such factors include the alteration of course schedule at the last minute and sudden availability of rooms. Other optimization models aside from the MSPSO used in the study were not included along with hybrid approaches. This is how the scope of the present research was set, and opportunities were presented for further consideration and adaptation to other school settings.

5 Significance of the Study

The importance of this paper is developing a new approach for solving the University Course Timetabling Problem using Multi-Swarm Particle Swarm Optimization. To be more specific, because numerous methods have been widely applied to solve UCTP, MSPSO is the first approach that explores more extensive areas of solutions through the use of more than one sub-swarm. This new technique aims to generate high-quality, conflict-free timetables that respect necessary constraints such as room capacities, course prerequisites, and faculty schedules, thereby advancing the field of university scheduling.

Besides the academic value, this research opens up possibilities for wider applications of MSPSO in fields requiring efficient scheduling and resource management. With the establishment of MSPSO as an efficient strategy for UCTP, this study opens avenues for its use in other complex scheduling scenarios related to healthcare, transportation, and manufacturing. These contexts are critical where optimum planning and resource utilization must occur. The research outcomes from this study might stimulate new studies and further applications in MSPSO for diverse industry fields.

Therefore, this paper not only develops a novel solution for scheduling at the university level but also lays down a foundation that may help in further studying MSPSO as a broad tool for automated scheduling in multiple industries.

6 Theoretical and Conceptual Framework

The theoretical and conceptual framework of this study consists of five separate yet connected components: Data Collection, Data Processing, Approach Construction, Approach Evaluation, and Deployment. The workflow starts with the Data Collection phase, collecting benchmark datasets and optimization problem instances. These datasets usually are publicly available and typically rep-

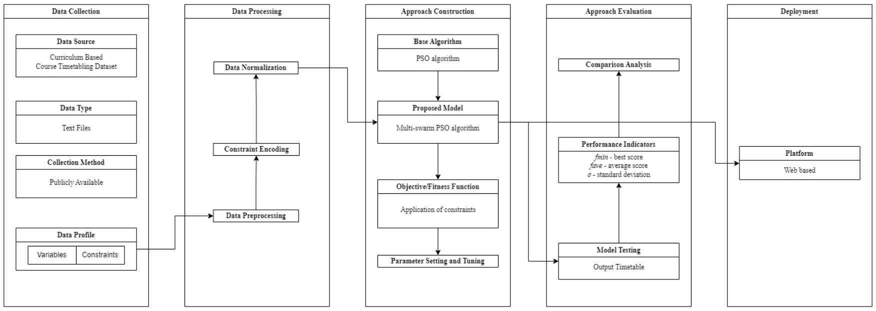

Figure 1: Theoretical and Conceptual Framework

resented in text or XML formats. This phase ensures that there is adequate sourcing and data variety to support the solid analysis of the optimization model against different conditions. Steps undertaken in the Data Processing step include cleaning the data to remove inconsistencies and make sure that all inconsistencies in constraints are encoded in apt forms of mathematics for solving before preparing the data for optimized use. Data cleaning removes errors or missing values that could have otherwise affected the model's performance while encoding constraints take problem-specific rules and transform them into a format that could be used by the model. Normalization and data categorization are also considered uniform since the optimization model can cope well with the data. All these processes, therefore, collectively provide standardized and reliable input for the optimization phase. This ensures that the information is valid and appropriate to the model in question. It will act as a foundation for later phases.

The construction of the approach deals with the creation of the model of multi-swarm optimization. This acts as a core method for the optimization problem solution. A part of this component deals with the base model setup and setting parameters that make the model capable of appropriately addressing the needs of the problem. The Approach Evaluation phase involves testing the performance of the model based on whether it satisfies any required objectives. This phase must be done to determine how well the proposed model is and make adjustments according to what is needed to increase its applicability. Lastly, the solution would be deployed by presenting an optimized solution via the use of a web-based interface, where users will have the ability to upload their problem instances and find real-time optimized outputs for their respective problems. All parts are included to be effective contributors to the solution development process, ensuring that data is handled properly, models built and evaluated appropriately, and the final solution is practically deployable.

6.1 Concepts, Theories, and Methodologies Reused

This paper employs current swarm intelligence theories and algorithms, mainly in Particle Swarm Optimization (PSO) [6] and Multi-Swarm Optimization (MSO) [19]. PSO is an established algorithm to efficiently search large solution spaces by a set of particles that are exploring a search area with individual and collective experiences. Parsopoulos and Vrahatis [22] first introduced neighborhood-based swarming in global and local variants in the introduction of the MSO approach. That paper set a solid basis for multi-swarm frameworks. The structure of the neighborhood and dynamic tuning of parameters discussed there provided important inspirations for later developments in multi-swarm optimization. One of the first spectacular applications of these multi-swarm approaches is given by Blackwell and Branke [19], who studied MSO in dynamic settings. These methodologies are crucial to establish the foundation for MSPSO.

6.2 Concepts, Theories, and Methodologies Modified

To improve existing PSO and MSO methodologies, the research here modifies the base methodologies to take on a multi-swarm structure particularly designed for UCTP, called MSPSO. The MSPSO algorithm divides the main swarm into smaller, specialized sub-swarms operating in parallel to explore the solution space more effectively at different parts. [11] [19] THe improvement is designed to circumvent the limitations of lack of search diversity and trapping into local optima through the encouragement of a greater exploration of solution possibilities along with the exploitation of promising regions. The modification improves better adaptability and efficiency in generating feasible course timetables.

6.3 Novel Concepts, Theories, and Methodologies Introduced

This research introduces a new optimization approach for UCTP via MSPSO, specifically designed to address complex timetabling constraints such as room availability, faculty preferences, and scheduling conflicts. Its novelty lies in the combination of dynamic swarm partitioning and targeted exploration, allowing the MSPSO algorithm to achieve workload balance and enhance resource use within educational institutions [10]. The proposed MSPSO-based methodology balances mechanisms for exploration and exploitation, ensuring that the different solution domains are discovered thoroughly to allow the provision of optimal scheduling solutions.

7 Methodology

This section outlines the methodology of using MSPSO to effectively solve the UCTP.

7.1 Data Collection

- Decompose the UCTP into exploratory and refinement components.

- Identify exploratory tasks suitable for MSPSO and refinement tasks for IP.
- Define the structure of the solutions represented by particles in MSPSO, including course, room, and time slot assignments.

7.2 Data Processing

7.3 Approach Construction

- Initialize multiple swarms to explore various potential timetabling solutions.
- Define the representation of solutions (particle encoding) for the assignment of courses to rooms and time slots.
- Develop a fitness function to evaluate solutions based on:
  - Minimizing scheduling conflicts (e.g., room clashes, instructor availability).
  - Maximizing resource usage (e.g., optimal room utilization and course distribution across time slots).
- Implement the MSPSO algorithm:
  - Update particle positions and velocities based on individual and global best solutions.
  - Perform iterations until a convergence criterion is met or a predefined number of iterations is reached.
  - Collect the best solutions generated by the swarms for refinement.

7.4 Approach Evaluation

- Post-process the final timetabling solutions to evaluate against key performance metrics:
  - Conflict minimization: Ensure no room or instructor conflicts in the final schedule.
  - Course-resource alignment: Maximize the alignment of course scheduling with available room resources and time slots.
  - Computational efficiency: Measure the overall performance of the hybrid optimization approach in terms of speed and accuracy.
- Select the best overall timetable based on the combination of fitness and objective evaluations.

7.5 Deployment

8 Schedule of Activities

OBJECTIVES	**TARGET ACTIVITIES**	**TARGET ACCOMPLISHMENTS** (quantify, if possible)	**YEAR 1**											
			1	2	3	4	5	6	7	8	9	10	11	12
Objective 1	To examine how the MSPSO algorithm is used and how well it solves the UCTP.	1. Gather relevant literature on MSPSO applications. 2. Analyze the performance of MSPSO on test datasets. 3. Evaluate the effectiveness of MSPSO in solving UCTP with various constraints.									■	■	■	■
Objective 2	To model the UCTP with crucial characteristics like room availability, course prerequisites, and faculty courses within the framework of MSPSO.	1. Identify and define critical constraints in UCTP. 2. Develop and model UCTP using MSPSO framework. 3. Test the model on synthetic and real data.										■	■	■

Objective 3	To perform a state-of-the-art analysis by contrasting the MSPSO approach with current techniques used in the UCTP.	1. Benchmark MSPSO against other UCTP methods like GA and PSO. 2. Conduct performance evaluations and analysis.												■

OBJECTIVES	**TARGET ACTIVITIES**	**TARGET ACCOMPLISHMENTS** (quantify, if possible)	**YEAR 2**											
			1	2	3	4	5	6	7	8	9	10	11	12
Objective 3	To perform a state-of-the-art analysis by contrasting the MSPSO approach with current techniques used in the UCTP.	1. Conduct a detailed analysis and comparison with existing UCTP techniques. 2. Publish results and make recommendations for improvements.	■	■	■	■	■							

References

[1] Dezhen Zhang et al. "A novel greedy heuristic algorithm for university course timetabling problem". In: (2014).

[2] Nancy Maribel Arratia-Martinez, Cristina Maya-Padron, and Paulina A Avila-Torres. "University course timetabling problem with professor assignment". en. In: *Math. Probl. Eng.* 2021 (Jan. 2021), pp. 1–9.

[3] Oswald C. and Anand Deva Durai C. "Novel hybrid PSO algorithms with search optimization strategies for a University Course Timetabling Problem". In: (Dec. 2013).

[4] Monica Torres, Kyla Kiela Villegas, and Maica Krizna Gavina. “Solving faculty-course allocation problem using integer programming model”. en. In: *Philipp J. Sci.* 150.4 (June 2021), p. 679.

[5] H. Algethami and W. Laesanklang. “A Mathematical Model for Course Timetabling Problem With Faculty-Course Assignment Constraints”. In: *IEEE Access* 9 (Aug. 2021), pp. 111666–111682. DOI: 10.1109/ACCESS.2021.3103495.

[6] James Kennedy and Russell Eberhart. “Particle swarm optimization”. In: *Proceedings of IEEE International Conference on Neural Networks* (1995), pp. 1942–1948.

[7] Ruey-Maw Chen and Hsiao-Fang Shih. “Solving University Course Timetabling Problems Using Constriction Particle Swarm Optimization with Local Search”. In: *Algorithms* 6 (Apr. 2013), pp. 227–244. DOI: 10.3390/a6020227.

[8] A I Ali and R Talal. “UCTP based on Hybrid PSO with Tabu Search Algorithm using Mosul university dataset”. In: *International Journal of Computer Applications* 975 (2014).

[9] K. M. Ng Aldy Gunawan and H. L. Ong. “A Genetic Algorithm for the Teacher Assignment Problem for a University in Indonesia”. In: *International Journal of Information and Management Sciences* 19.1 (Mar. 2008), pp. 1–16. DOI: 10.1000/ijims.2008.237502795.

[10] Nebojsa Bacanin et al. “Multi-Swarm Algorithm for Extreme Learning Machine Optimization”. In: *Sensors* 22.11 (2022), p. 4204. DOI: 10.3390/s22114204. URL: https://doi.org/10.3390/s22114204.

[11] Xuewen Xia, Ling Gui, and Zhi-Hui Zhan. “A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting”. In: *Applied Soft Computing* 67 (2018), pp. 126–140. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2018.02.042. URL: https://www.sciencedirect.com/science/article/pii/S1568494618301017.

[12] Xi Wang et al. “A Hybrid Particle Swarm Optimization for Parallel Machine Scheduling with Shared and Multi-mode Resources”. In: *IEEE Transactions on Computational Intelligence and Design* 9.4 (Dec. 2023), pp. 2473–2479. DOI: 10.1109/TGCN.2023.3283509.

[13] Bong Chia Lih, Sze San Nah, and Noor Alamshah Bolhassan. “A study on heuristic timetabling method for faculty course timetable problem”. In: (2018).

[14] Yanming Yang, Wanchun Gao, and Yang Gao. “Mathematical modeling and system design of timetabling problem based on improved GA”. In: (2017).

[15] Mingqiang Gao and Xu Yang. “APSO-SL: An Adaptive Particle Swarm Optimization with State-Based Learning Strategy”. In: *Processes* 12 (2024), p. 400. DOI: 10.3390/pr12020400.

[16] Saeed Fallahi and Mohamadreza Taghadosi. “Quantum-behaved Particle Swarm Optimization Based on Solitons”. In: *Scientific Reports* 12 (2022), p. 13977. DOI: 10.1038/s41598-022-18351-0.

[17] Tianyu Liu et al. “Quantum-behaved Particle Swarm Optimization with Collaborative Attractors for Nonlinear Numerical Problems”. In: *Com-*

munications in Nonlinear Science and Numerical Simulation 44 (2017), pp. 167–183. DOI: 10.1016/j.cnsns.2016.08.001.

[18] Zhi-Hui Zhan et al. "Adaptive Particle Swarm Optimization". In: *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics* 39.6 (Dec. 2009), pp. 1362–1375. DOI: 10.1109/TSMCB.2009.2015956.

[19] Tim Blackwell and Jürgen Branke. "Multi-swarm Optimization in Dynamic Environments". In: *Proceedings of the EvoWorkshops 2004, LNCS 3005*. Springer-Verlag Berlin Heidelberg, 2004, pp. 489–500.

[20] Qi Liu et al. "A Multi-Swarm PSO Approach to Large-Scale Task Scheduling in a Sustainable Supply Chain Datacenter". In: *IEEE Transactions on Green Communications and Networking* 7.4 (Dec. 2023), pp. 1667–1678. DOI: 10.1109/TGCN.2023.3283509.

[21] Shara S. A. Alves, Saulo A. F. Oliveira, and Ajalmar R. Rocha Neto. "A novel educational timetabling solution through recursive genetic algorithms". In: (2018).

[22] K.E. Parsopoulos and M.N. Vrahatis. "Recent approaches to global optimization problems through Particle Swarm Optimization". In: *Natural Computing* 1 (2002), pp. 235–306.