I.J. Modern Education and Computer Science, **2025, 3, 1-14**
Published Online on June 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2025.03.01

Optimization of Balanced Academic Curriculum Problem in Educational Institutions Using Teaching Learning Based Optimization Algorithm

Mohd Fadzil Faisae Ab Rashid*
Faculty of Mechanical and Automotive Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Pahang, Malaysia
Email: ffaisae@umpsa.edu.my
ORCID iD: https://orcid.org/0000-0001-8573-3083
*Corresponding Author

Wasif Ullah
Faculty of Manufacturing and Mechatronic Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Pahang, Malaysia
Email: wasifmuno101@gmail.com
ORCID iD: https://orcid.org/0009-0007-9515-0341

Received: 23 August, 2024; Revised: 07 January, 2025; Accepted: 13 February, 2025; Published: 08 June, 2025

Abstract: The Balanced Academic Curriculum Problem (BACP) is a complex optimization problem in educational institutions, involving the allocation of courses across academic terms while satisfying various constraints. This study aims to optimize BACP using the Teaching-Learning Based Optimization (TLBO) algorithm, addressing the limitations of existing approaches and providing an efficient framework for curriculum balancing. The novelty lies in applying TLBO to BACP, offering a parameter-free, nature-inspired metaheuristic that balances exploration and exploitation effectively. The proposed method models BACP as a mathematical optimization problem and implements TLBO to minimize total load balance delay across academic terms. Computational experiments were conducted on 12 benchmark BACP instances, comparing TLBO against eight other metaheuristic algorithms. Results demonstrate TLBO's superior performance, achieving the best solutions in 75-83% of test problems across various indicators. Statistical analysis using the Wilcoxon rank-sum test confirms the significance of TLBO's improvements. The study concludes that TLBO is a robust and efficient tool for optimizing BACP, outperforming existing methods in solution quality and convergence speed. Future research could focus on enhancing TLBO through hybridization with other algorithms and applying it to real-world BACP scenarios in educational institutions.

Index Terms: Balanced Academic Curriculum Problem, Curriculum Optimization, Student Academic Load, Teaching Learning Based Optimization

1. Introduction

The Balanced Academic Curriculum Problem (BACP) is a complex optimization problem that involves academic institutions worldwide. At its core, BACP involves allocating courses to semesters or periods while minimizing course conflicts, balancing course loads, and satisfying degree requirements. This problem is particularly challenging due to the numerous constraints and variables that must be considered, including course prerequisites, teaching requirements, resource limitations, and student demand [1].

The significance of BACP in academic institutions cannot be overstated. A well-balanced academic curriculum is essential for ensuring that students receive a comprehensive education, while also allowing faculty members to manage their workloads effectively [2]. Efficient curriculum planning is critical for academic institutions, as it directly impacts student satisfaction, academic performance, and ultimately, the reputation of the institution. Moreover, a balanced curriculum can help institutions to optimize resource allocation, reduce costs, and improve overall efficiency [3].

In addition to its importance in academic institutions, BACP also has significant implications for students. A balanced curriculum can help students to avoid course conflicts, reduce their workload, and ensure that they can complete their degree requirements within the recommended timeframe [4]. This, in turn, can lead to improved academic performance, increased student satisfaction, and better job prospects. Furthermore, a well-balanced curriculum can also help to promote student engagement, motivation, and overall well-being

Despite its importance, balancing academic workloads is a daunting task that poses numerous challenges for academic institutions. One of the most significant challenges is managing course conflicts, which can occur when multiple courses with common prerequisites or target audiences are scheduled in the same semester. This can lead to overcrowded classrooms, overworked faculty members, and dissatisfied students [5]. Another challenge is allocating resources efficiently, as institutions must balance the need to offer a wide range of courses with the limited availability of faculty members, classrooms, and other resources [6].

Other common challenges associated with balancing academic workloads include ensuring that students can complete their degree requirements within the recommended timeframe, managing faculty workloads, and responding to changes in student demand. Moreover, the increasing complexity of academic curricula, combined with the growing need for flexibility and adaptability in curriculum planning, has created a pressing need for new optimization methods that can effectively address these challenges.

In recent years, the importance of BACP has been further underscored by the growing trend towards personalized learning, which requires institutions to offer a wide range of courses and programs tailored to individual student needs. This has created a need for more sophisticated curriculum planning systems that can take into account the diverse needs and preferences of students, while also ensuring that institutions can manage their resources effectively.

Overall, the Balanced Academic Curriculum Problem is a complex and challenging problem that has significant implications for academic institutions and students alike. Its importance cannot be overstated, and it is essential that institutions develop effective optimization methods to address the numerous challenges associated with balancing academic workloads.

Various approaches have been proposed to optimize the BACP effectively. The Genetic Algorithm, in particular, has garnered significant attention from researchers due to its effectiveness in solving the BACP. Several studies have explored different variations and hybridizations of GA to tackle this problem, including combinations with Constraint Programming (CP) and Simulated Annealing (SA). For instance, Lambert et al. (2005, 2006) compared CP and GA frameworks, introducing a hybrid approach that outperformed standalone GA or CP methods [7, 8]. Castro et al. (2009) developed a Genetic Local Search Algorithm (GLSA) that combined GA with Simulated Annealing to solve multiple BACPs simultaneously [9].

These GA-based approaches have shown promising results in solving the BACP across various test instances. Researchers have implemented different GA variants such as the Mutation-Only Genetic Algorithm (MGA) proposed by Sylejmani et al. (2017), which employed swap and shift mutation operators [10]. Chakradhar et al. (2019) used a standard GA with specific mutation operators to minimize the deviation of course load from mean credits [11]. Villalobos-Cid et al. (2019) applied GA to optimize a quadratic integer programming model of the BACP [12]. Comparative studies have demonstrated that GA-based methods often perform equally well or better than other approaches, especially for complex instances of the BACP. The effectiveness of these algorithms in terms of solution quality, computational efficiency, and ability to handle multiple constraints has made them valuable tools for addressing the challenges of academic curriculum optimization.

Tabu Search (TS) has emerged as an effective optimization algorithm for solving the BACP. Rosas-Tellez et al. (2012) employed TS to address modified BACP instances, successfully minimizing the range between minimum and maximum academic loads across periods [13, 14]. Their results demonstrated the algorithm's effectiveness in reaching optimal solutions, even in cases where formal methods failed. Di Gaspero and Schaerf (2008) combined simulated annealing, dynamic tabu search, and local search to optimize the Generalized BACP, introducing new challenging instances and a neighborhood relation for course movement [4]. They also proposed Generalized Local Search Machines (GLSM) templates as a framework for combining search components. Building on this work, Chiarandini et al. (2012) utilized dynamic tabu search and simulated annealing within the GLSM framework, incorporating various strategies such as random kicks for diversification and intensification kicks based on improving sequences [15]. These studies collectively highlight the versatility and effectiveness of Tabu Search-based approaches in solving complex BACP instances.

The Firefly Algorithm (FA), inspired by the behavior of fireflies, has been applied to optimize the BACP in several studies. Rubio et al. (2019) used FA to optimize a binary matrix representation of BACP in an integer linear programming model, emphasizing the importance of various parameters in obtaining high-quality solutions [16]. In 2021, Rubio et al. implemented the BACP mathematical model in Haskell and applied FA to enhance the optimization process, focusing on improving precedence constraint satisfaction [1]. Katal et al. (2023) proposed a Discrete Firefly Metaheuristic Algorithm (DFA) for BACP, incorporating a local search mechanism to prevent getting stuck in local minima and using a sigmoid function for discretization [2]. These studies demonstrate the versatility of FA in addressing the complexities of BACP, with researchers adapting and enhancing the algorithm to improve its performance in curriculum optimization tasks.

Researchers have employed a variety of approaches beyond the more common Genetic Algorithm, Tabu Search, and Firefly Algorithm methods. Hnich et al. (2002) combined Integer Linear Programming (ILP) and Constraint Programming (CP) models, finding that integrated models outperformed individual ones [17]. Monette et al. (2007) proposed a value ordering heuristic and explored the impact of dominance rules on the Branch-and-Bound search algorithm [18]. Rosas-Tellez (2011) utilized Evolutionary Strategies [19], while Rubio et al. (2013) applied Ant Colony Optimization, specifically the Best-Worst Ant System (BWAS) [20].

Other notable approaches include the use of Local Search with Simulated Annealing by Ceschia et al. [21], and a Mixed Integer Linear Programming model within the General Quadratic Assignment Problem framework by Ünal & Uysal [22]. Slim et al. (2015) introduced a lexicographic optimization technique and later in 2019 proposed a method using Non-Negative Matrix Factorization to extract latent features for recommending suitable academic programs [23]. These diverse approaches demonstrate the complexity of the BACP and the ongoing efforts to find efficient and effective solutions for optimizing academic curricula.

Despite the importance of the BACP, existing approaches to solving this problem have several limitations. Traditional optimization techniques, such as linear programming and integer programming, often struggle to handle the complexity and non-linearity of BACP [22]. These methods typically rely on simplifying assumptions and approximations, which can lead to suboptimal solutions that fail to capture the nuances of real-world academic curricula.

Moreover, many existing approaches focus on simplified versions of the problem, neglecting important constraints and variables. For example, some methods may ignore course prerequisites, teaching requirements, or resource limitations, which can lead to unrealistic and impractical solutions. Furthermore, existing approaches often rely on manual intervention and expert judgment, which can be time-consuming, labor-intensive, and prone to errors.

The limitations of existing approaches highlight the need for new optimization methods that can effectively address the complexities of BACP. With the increasing complexity of academic curricula and the growing need for flexibility and adaptability in curriculum planning, there is a pressing need for more sophisticated and robust optimization techniques. Specifically, there is a need for methods that can handle non-linear relationships, multiple objectives, and uncertain data, while also providing efficient and scalable solutions.

In this paper, we aim to address the limitations of existing approaches by modeling and optimizing BACP using the Teaching-Learning Based Optimization (TLBO) algorithm. The TLBO is designed to handle complex, dynamic, and non-linear optimization problems, making it an ideal candidate for solving BACP. Our objective is to demonstrate the effectiveness of the TLBO in solving BACP and to provide a robust and efficient optimization framework for academic institutions to balance their academic curricula.

The remainder of this paper is organized as follows: we begin with a comprehensive Literature Review of existing approaches to solving BACP, followed by a Problem Modelling section where we formulate BACP as a mathematical optimization problem. Next, we describe the application of the TLBO to BACP in the Optimization section. We then present the results of a Computational Experiment designed to evaluate the performance of the TTA in solving BACP, which are discussed in the Results and Discussions section. Finally, we summarize the main findings of the paper and provide recommendations for future research in the Conclusion section.

2. BACP Problem Description

The BACP is an optimization challenge in academic scheduling, where the goal is to assign courses to academic terms in a way that balances the student workload while satisfying various constraints, including prerequisites, curriculum requirements, and load limits. In modeling the BACP, the interaction between variables and constraints is crucial for achieving a balanced curriculum. The binary decision variable x_{cp} indicates weather a course C in assign to term p. This variable is influenced by constraints such as prerequisites in Eq. (4), which ensure that a course is only scheduled after its prerequisite courses have been completed. Furthermore, load limits in Eq. (6) and Eq. (7) maintain the total academic load $l_P$for each term within defined minimum and maximum bounds, thus ensuring that no term is overloaded. The objective function, which aims to minimize the academic load balance delay (ABD), is directly affected by the distribution of course loads across terms, highlighting the interdependencies of course assignments and their respective loads. This problem is vital for ensuring that students can progress smoothly through their curriculum without overloading any particular term, thus maintaining a balanced and manageable workload across their academic journey. The notations used in the BACP problem description are as in Table 1.

Table 1. BACP notations

Notation	Meaning
$C = \{c_1, c_2, \ldots, c_n\}$	Set of courses
$P = \{p_1, p_2, \ldots, p_m\}$	Set of academic terms
r_c	Load associated with course c
x_{cp}	Binary decision, where $x_{cp} = 1$ if course c is assigned to term p, and $x_{cp} = 0$ otherwise
m_p and M_p	Minimum and maximum allowable term p
L_p	Total load for term p
$Prereq(c)$	Set of prerequisite courses for course c

The BACP problem relies on several key assumptions to effectively model and solve the problem of distributing courses across academic terms while maintaining a balanced workload for students. Here are the main assumptions used in the BACP:

- The courses to be scheduled are predetermined and fixed across the academic terms.
- Each course has a fixed load, typically measured in credit hours, that does not vary.
- The number of academic terms is fixed and covers a specific period, such as semesters or quarters.
- Courses have a fixed set of prerequisites that must be respected in the scheduling process.
- Each term has set minimum and maximum limits on the number of courses or total workload.
- Each course must be assigned to only one term with no splitting or repetition across terms.
- The load for each course is uniform and applies equally to all students.
- All problem parameters are known in advance and remain unchanged during scheduling.

The objective of the optimization is to minimize the academic load balance delay (*ABD*). *ABD* is a metric used to assess the distribution of workload across an academic curriculum or program as in equation (1). It measures the degree of imbalance in course load distribution among different academic periods. A lower *ABD* indicates a more evenly distributed academic load, which can lead to improved student performance and well-being. Conversely, a higher *ABD* suggests an uneven distribution, potentially resulting in periods of academic overload or underutilization. This concept, adapted from the industrial notion of balanced delay in production lines, helps educational institutions and curriculum designers optimize course scheduling, ensuring a more consistent and manageable workload for students throughout their academic journey. By minimizing ABD, educators aim to create a more balanced learning experience that promotes better time management, reduces stress, and potentially improves overall academic outcomes.

$$ABD = \left(100 - \frac{\sum_{c=1}^{n} r_c}{\max L_p \times m} \times 100\right) \tag{1}$$

$$L_p = \sum_{c \in C} r_c \cdot x_{cp} \tag{2}$$

Therefore, the fitness function used in this work is as in Eq. (3).

$$f = \min \quad (ABD) \tag{3}$$

The fitness function Eq. (3) aims to minimize academic load balance delay, and each component is essential for evaluating workload balance. The first term measures the total load per course to prevent disproportionate burdens, while the second term penalizes deviations from ideal load distribution, promoting equitable allocation of responsibilities. Together, these elements guide the optimization process toward a balanced academic curriculum. We will clarify this function further in future revisions to enhance understanding and reproducibility.

Subjected to:

$$x_{cp} \leq \sum_{p\prime < p} x_{c\prime p\prime}, \quad \forall c \ \in C, \forall c' \in Prereq(c), \ \ \forall \in P \tag{4}$$

$$\sum_{p \in P} x_{cp} = 1, \quad \forall c \in C \tag{5}$$

$$m_p \leq \sum_{c \in C} x_{cp} \leq M_{p,} \ \ \forall p \in P \tag{6}$$

$$L_p = \sum_{c \in C} r_c \cdot x_{cp}, \ \ \forall p \in P \tag{7}$$

The prerequisite constraint in Eq. (4) guarantees that courses are only scheduled after their prerequisite courses have been completed. This is modeled by ensuring that if a course is scheduled in a term, all its prerequisites must have been scheduled in earlier terms. Eq. (5) shows course assignment constraint, ensures that every course is assigned to exactly one term. This is a fundamental requirement for any valid curriculum, as each course must be taken exactly once. Eq. (6) presents the load limits constraints, to maintain the number of courses in each term within predefined minimum and maximum bounds. These constraints help institutions manage student workload and ensure that terms are neither too light nor too heavy. Finally, Eq. (7) is the load limit constraint that maintains the number of courses assigned to each term within specified limits. The constraint ensures that this total falls between the minimum and maximum allowable number of courses for term p.

3. Optimization using TLBO Algorithm

Teaching-Learning-Based Optimization (TLBO) is a population-based optimization algorithm inspired by the teaching and learning process in a classroom, proposed by Rao in 2011 [24]. The TLBO algorithm works in two fundamental modes: the Teacher stage and the Learner stage. In the Teacher stage, the teacher tries to enhance the average knowledge of the students in the classroom, while in the Learner stage, the students interact with random partners to improve their knowledge. TLBO begins with a randomly initialized population of learners, each representing a potential solution to the optimization problem. In the Teacher stage, the best solution attempts to improve other solutions by moving their mean result toward its own position, using a Teaching Factor (F_t). The Learner stage involves random interactions between learners, where better solutions teach worse ones. TLBO updates solutions if new ones are superior and continues this process until a termination criterion is met. Notably, TLBO doesn't require algorithm-specific parameters except for population size and the number of iterations, making it easier to implement and reducing the need for parameter tuning. It balances exploitation in the Teacher stage with exploration in the Learner stage, preserves the best solution through elitism, and can handle constrained optimization problems. These features make TLBO a powerful, flexible, and widely applicable optimization technique.

TLBO's suitability for the BACP extends beyond its parameter-free nature; its dual-phase learning mechanism comprising the Teacher and Learner stages enables effective exploration and exploitation of the solution space, allowing for adaptive refinement of solutions, which is crucial for the complexities of academic scheduling. Additionally, TLBO effectively manages multiple constraints such as course prerequisites and load limits, overcoming challenges posed by the non-linearity of the BACP. Its straightforward implementation and minimal computational overhead position TLBO as an efficient choice for optimizing academic curricula. The TLBO algorithm pseudocode is depicted in Fig. 1.

```
Output: Academic load balance delay
Random initialization of population
define population size Sp, number of iterations Ni
While (Ni not reached yet)
    {teacher stage}
    determine average of each decision variable S_avg
    chose the best solution as teacher
    [S_best → S with y_max]
    for i=1 to n
        calculate Ft = round {(1+rand (0,1)}
        s_new,i = s_i +rand_(0,1) [s_best-Ft. S_avg]
        calculate y_new,i for s_new,i
        if y_new,i is better than y_old,i
        replace s_i with s_new,i and y_old,i with y_new,i
        end if
       {student stage}
        select a random partner s_p so that s ≠ i
        if y_i < y_p then
                s_new,i= s_old,i + rand_i (s_i-s_p)
          else
               s_new,i= s_old,i + rand_i (s_i-s_p)
               end if
           if y_new,i is better than y_old,i
          replace s_i with s_new,I and y_old,I with y_new,i
           end if
    end for
end while
```

Fig. 1. Pseudocode of TLBO

3.1 TLBO Algorithm Implementation

The TLBO algorithm is implemented to solve the BACP by mimicking the teaching-learning process. In this approach, each potential solution to the BACP is considered a "learner," representing a feasible allocation of courses across semesters while adhering to constraints like balance, prerequisites, and academic load. The algorithm progresses through two stages: the "Teacher stage," where the best solution (teacher) shares knowledge with other solutions (learners) to enhance their quality, and the "Learner stage," where learners interact with one another to exchange knowledge, further refining their solutions. Through iterative improvements, TLBO effectively searches for an optimal or near-optimal solution, balancing the academic curriculum across semesters.

Initialization

To generate an initial feasible solution for the BACP using the TLBO algorithm, the process begins with defining the problem constraints and parameters, such as course prerequisites, faculty availability, and classroom capacities. An initial population of feasible solutions is then created, typically through random assignment of courses to time slots while adhering to basic constraints like avoiding course conflicts and ensuring faculty and classroom availability. Each generated timetable is checked for feasibility, and adjustments are made if necessary to resolve any conflicts or constraints violations. The quality of these initial solutions is assessed using an objective function, which considers factors such as load balance and scheduling efficiency. This results in a set of feasible solutions that are evaluated and prepared for the subsequent iterative optimization steps of the TLBO algorithm.

Evaluation

In TLBO, the evaluation function for assessing the quality of solutions in the BACP focuses on academic load balance delay (ABD). The balance criterion ensures that courses are evenly distributed across different semesters, preventing any academic term from being overloaded with courses while others remain underutilized. For this purpose, the fitness function as equation (3) is applied.

Teacher stage

In the Teaching stage, TLBO leverages the concept of a "teacher," which is the best-performing solution in the current population. The primary goal of this stage is to improve the solutions based on the knowledge obtained from the teacher. During this stage, each solution, referred to as a student, is updated using the teacher's solution. The update process is guided by the following formula:

$$S_{new} = S_{old} + r\,(S_{best} - F_t * S_{mean}) \tag{8}$$

Where S_{old} is current candidate solution, r is the random number (0,1). The S_{best} is the s value corresponding to the best fitness value in the initial population. Where F_t is teaching factor of value 1 or 2 and calculated using $F_t = round\ (1 + rand)$. The S_{mean} is the average of design variables for all candidate solutions. Once S_{new} is found within the specified range or bounds, the corresponding fitness value will be calculated. Subsequently, the newly computed fitness value will be compared with the existing one. If the updated fitness is better than the old one, the older fitness value will be replaced with the new one, otherwise the old value for design variable and fitness will be kept as it is.

Learner stage

The Learning stage involves updating solutions based on interactions between pairs of solutions within the population. This phase facilitates exploration of the solution space by having solutions learn from each other. The newly generated solution from the teacher stage is than passed through the learner stage before going to proceed to the next candidate solution. In the learner's stage, the search for the new solution S_{new} depends on the choice of a random partner S_p. The comparison of fitness values is crucial in this stage, and the following expressions are employed to calculate S_{new} based on the fitness values comparison.

If $y > yp$, then

$$S_{new} = S_{old} - r * (S_{old} - S_p) \tag{9}$$

If $y < yp$, then

$$S_{new} = S_{old} + r * (S_{old} - S_p) \tag{10}$$

Where S_{old} represents the current solution, S_p is the partner value chosen randomly, and r is a random number. The corresponding fitness value is calculated once the new value of S_{new} is generated based on these expressions. The Greedy selection is conducted to look for new fitness value improvement.

To avoid local optima and ensure a diverse search, TLBO employs several key mechanisms. It maintains population diversity by incorporating random variations in both the teaching and learning stages, using random factors r in the update rules to explore different regions of the solution space. The algorithm balances exploration and exploitation through the combined use of the teaching phase, which refines solutions based on the best-performing solution, and the learning phase, which fosters exploration through solution interactions. Additionally, the adaptive teaching factor can be dynamically adjusted to fine-tune the balance between exploration and exploitation, thereby reducing the risk of premature convergence to local optima. These strategies collectively enhance TLBO's ability to navigate the solution space effectively.

4. Computational Experiment

A computational experiment was conducted to evaluate the performance of the TLBO algorithm in optimizing the BACP. The dataset defined by Brahim Hnich was utilized during the experiment [25]. This dataset is typically used to evaluate BACP problems and consists of 12 test problems with varying course load distributions. In each problem, the number of courses is 50, and the number of periods is 10. The upper and lower bounds for load per period were set at 100 and 2, respectively, while the bounds for courses per period were set at 10 and 2. Some of the comparative algorithms were selected to evaluate the effectiveness of the TLBO algorithm in the optimization of the BACP problem. The selection of algorithms was based on popularity in literature, well-established and recent algorithms. A total of nine algorithms were chosen for comparison and listed below.

- Ant colony optimization (ACO) [26]
- Crayfish optimization algorithm (COA) [27]
- Evolutionary algorithm (EA)
- Firefly algorithm (FA) [28]
- Goose algorithm (GOO) [29]
- Greywolf optimizer (GWO) [30]
- Particle swarm optimization (PSO) [31]
- Real-coded Simulated annealing (SA)

The population size was set to 50, with 500 iterations repeated 10 times for all algorithms. All algorithms were run on MATLAB using a Laptop Dell Latitude model 7480 with 8 GB of RAM, a 64-bit operating system, and an Intel(R) Core (TM) i5-6300U CPU @ 2.40 GHz.

The evaluation of the TLBO algorithm for the BACP problem focuses on key performance metrics. Solution quality, encompassing minimized maximum load, variance, and feasibility of constraints, is prioritized to ensure balanced and feasible outcomes. Computation time is assessed to gauge the algorithm's efficiency, particularly in large-scale problems. The convergence rate, reflecting the speed at which an optimal or near-optimal solution is reached, indicates the practicality and efficiency of the algorithm. Lastly, robustness and stability are evaluated to ensure the consistency and reliability of solutions across varying conditions, ensuring broad applicability.

Table 2 presents a sample of the BACP-1 benchmark problem used in the computational experiment. The problem consists of 50 courses that need to be assigned to 10 study periods. The lower and upper load limits per period are 2 and 100, respectively. The model must strictly follow the prerequisite and load limitation. In this problem, the load refers to the credit hours assigned to a course, representing the effort or workload required to complete it. A prerequisite is a course that must be completed before registering for a particular course. For instance, according to Table 2, a student must complete Course 1 before registering for Course 3.

Table 2. Sample of BACP benchmark problem (BACP-1)

Course	Load	Prerequisite	Course	Load	Prerequisite
1	6	-	26	10	21
2	3	-	27	10	21,23
3	5	1	28	1	-
4	3	1	29	7	-
5	7	1	30	3	22
6	8	1,2	31	4	-
7	1	1	32	2	-
8	9	2	33	7	25,29
9	4	-	34	9	27,29
10	9	-	35	7	27,28,30
11	8	3,4,6	36	4	31
12	8	-	37	6	33
13	4	9	38	7	31,33,34
14	5	6,9	39	2	31
15	6	-	40	2	31,32,33
16	3	4,5,8	41	5	34,35
17	2	11,12,14	42	9	35
18	1	13,14	43	9	31
19	3	11,12	44	10	36,40,41,43
20	1	-	45	4	36,37,40,41,43
21	1	19	46	6	41,42
22	2	-	47	4	40,42
23	6	17	48	5	47
24	7	23	49	6	46
25	6	23	50	6	47

5. Results and Discussions

The computational experimental results after applying various metaheuristics on the BACP are presented in Table 3. It should be noted that a smaller total academic load balance delay value in the context of the BACP indicates a more balanced distribution of the course load across periods. In other words, the differences in workload between different periods are minimized, leading to a more evenly distributed course load over the entire academic schedule.

In Table 3, the values represent the academic load balance delay (ABD) as defined in Eq. (1). As explained in Section 4, each problem and algorithm was run 10 times for optimization. To assess overall performance, the average, minimum, and maximum fitness values, along with the standard deviation, were calculated. In general, the ABD indicates the degree of load distribution imbalance. A smaller ABD signifies a better load distribution.

Table 3. Results of the computational experiment

Problem	Indicator	ACO	COA	EA	FA	GOO	GWO	PSO	SA	TLBO
BACP-1	Average	12.33	12.33	12.33	12.33	190.25	44.78	50.47	12.33	12.33
	Min	12.33	12.33	12.33	12.33	15.16	12.33	12.33	12.33	12.33
	Max	12.33	12.33	12.33	12.33	250.71	203.03	203.03	12.33	12.33
	SD	0	0	0	0	64.59	69.7	80.4	0	0
BACP -2	Average	11.27	27.39	10.71	30.28	193.42	63.63	80.23	36.76	9.73
	Min	8.7	8.7	8.71	11.56	170.08	8.7	8.7	4.71	4.71
	Max	11.56	170.08	11.56	196.02	196.02	196.02	186.78	142.42	11.56
	SD	0.9	50.15	1.38	58.24	8.2	83.51	88.25	55.73	2.26
BACP -3	Average	10	10.29	10	10	171.32	24.65	46.98	10	10
	Min	10	10	10	10	12.9	10	10	10	10
	Max	10	12.9	10	10	208.21	153.6	196.42	10	10
	SD	0	0.91	0	0	57.49	45.31	75	0	0
BACP -4	Average	18.59	19.34	18.76	18.99	23.05	19.85	20.35	19.83	18.6
	Min	18.54	18.54	18.54	18.54	21.30	13.92	18.98	18.55	18.5
	Max	18.98	23.48	18.98	21.29	23.48	21.30	23.48	23.48	19.0
	SD	0.13	1.46	0.22	0.83	0.92	2.34	1.90	1.60	0.2
BACP -5	Average	10.55	11.85	10.16	11.8	96.1	12.1	12.77	6.47	9.5
	Min	9.25	9.25	5.77	5.77	12.5	9.25	9.25	5.77	5.76
	Max	12.5	12.5	12.5	15.52	231.97	15.51	15.51	9.26	12.5
	SD	1.67	1.36	3.19	2.58	105.97	2.74	1.76	1.47	2.94
BACP -6	Average	8.07	9.41	8.08	9.39	77.29	10.41	10.72	9.66	8.07
	Min	8.07	8.07	8.08	8.08	11.48	8.07	8.07	9.66	8.07
	Max	8.07	14.64	8.08	14.64	195.28	14.64	14.64	9.66	8.07
	SD	0	2.31	0	2.77	82.06	2.71	3.02	0	0
BACP -7	Average	9.65	9.65	9.66	10.26	57.81	9.95	9.95	9.66	9.65
	Min	9.65	9.65	9.66	9.66	9.65	9.65	9.65	9.66	9.65
	Max	9.65	9.65	9.66	12.67	178.68	12.66	12.66	9.66	9.65
	SD	0	0	0	1.27	72.85	0.95	0.95	0	0
BACP -8	Average	15	15	15	15	57.45	15	15	15	15
	Min	15	15	15	15	15	15	15	15	15
	Max	15	15	15	15	227.27	15	15	15	15
	SD	0	0	0	0	89.5	0	0	0	0
BACP -9	Average	8.17	9.92	7.87	8.18	19.42	12.49	11.89	9.22	7.879
	Min	7.87	7.87	7.87	7.88	7.88	7.88	7.88	7.88	7.879
	Max	10.85	13.63	7.87	10.85	27.62	16.25	16.25	15.56	7.879
	SD	0.93	2.38	0	0.94	5.16	2.36	2.99	2.87	0.000
BACP -10	Average	12.06	32.09	12.03	13.24	237.13	108.85	86.75	63.67	11.12
	Min	12.06	12.06	8.93	12.07	200.62	15	12.06	8.93	8.92
	Max	12.06	200.62	15	15	267.24	227.27	212.96	200.63	12.06
	SD	0	59.23	2.02	1.51	21.19	99.85	95.05	84.1	1.51
BACP -11	Average	9.33	9.33	9.33	9.33	127.14	25.97	25.97	9.33	9.33
	Min	9.33	9.33	9.33	9.33	9.33	9.33	9.33	9.33	9.33
	Max	9.33	9.33	9.33	9.33	227.27	175.75	175.75	9.33	9.33
	SD	0	0	0	0	101.04	52.62	52.62	0	0
BACP -12	Average	7.33	7.33	7.33	7.33	67.42	7.63	7.33	7.33	7.33
	Min	7.33	7.33	7.33	7.33	7.33	7.33	7.33	7.33	7.33
	Max	7.33	7.33	7.33	7.33	184.75	10.32	7.33	7.33	7.33
	SD	0	0	0	0	79.97	0.94	0	0	0

Based on the results obtained, none of the single algorithms dominated all test problems and all indicators. The implemented metaheuristics on the BACP model reveal some interesting insights into their performance. Key indicators, including average fitness (Average), minimum fitness from 10 optimization runs (Min), maximum fitness across 10 optimization runs (Max) and standard deviation (SD) were assessed to evaluate the strength and efficiency of the algorithms in optimizing BACP benchmark problems.

The mean fitness (Average) is the average fitness value across all optimizations runs. This metric reflects the overall performance of the algorithm throughout the entire optimization process. Regarding the average indicator, the TLBO algorithm consistently produced the best results in 75% of the test problems. As shown in Fig. 2, the TLBO (in red line) consistently outperformed other algorithms. The second algorithm which performed well in average indicator is ACO achieving the best mean fitness results in 66% of the test problems, followed by EA and SA, which performed best in 50% of the test instances. The FA and COA ranked fifth and obtained best average fitness in 41% of the problems. In this context, the lowest-performing algorithms were GWO, GOO and PSO.

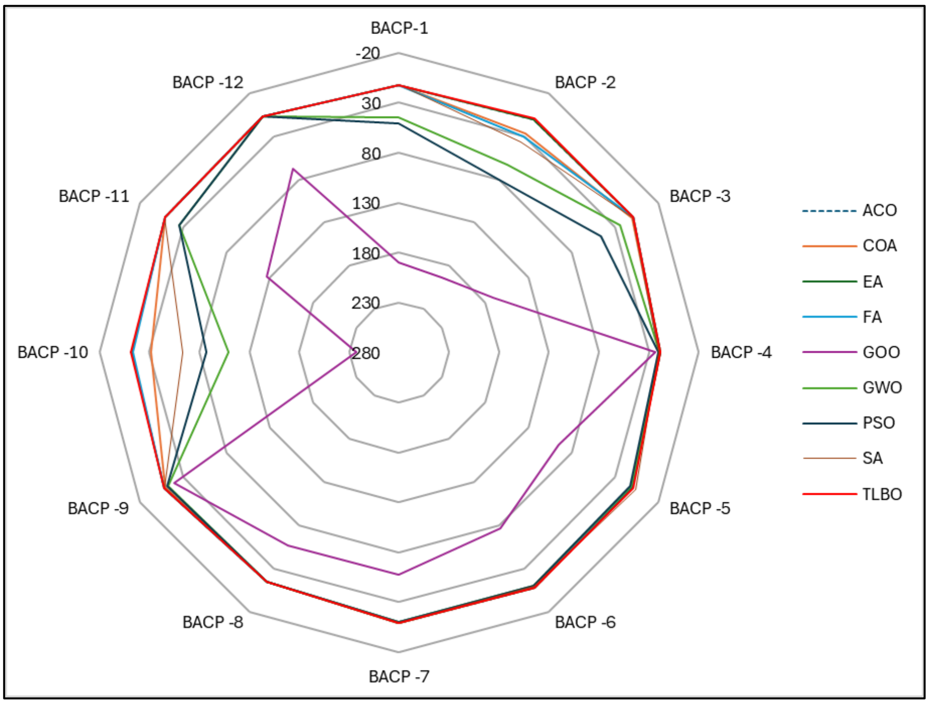


Fig. 2. Average Fitness of Different Optimization Algorithmss

The Min indicator is the best solution achieved by an algorithm among all optimization runs. This indicator shows the extent to which an algorithm is capable to optimize the concern problem. TLBO secured first position to obtain the best solution for Min indicator in 83% of test instances follows by ACO, COA and GWO in 66% of test problems. Afterward, PSO discovered the best solution regarding the Min indicator in 58% of test problems, while EA and SA achieved in 50% respectively. In this indicator the lowest performing algorithm were denoted FA and GOO.

The Max indicator represents the worst solution obtained by an algorithm within all optimization runs. In this scenario, TLBO obtained the best solution in 75% test problems regarding Max indicator, while EA in 66% and SA in 50%. The COA and FA were on fourth position securing the best solution for the Max indicator in 41% of test problems. The algorithms GOO, GWO and PSO performed lower regarding this indicator. Another interesting fact was observed that, in some problems (i.e.: BACP-2), the Max indicator value is smaller than the average fitness of other algorithms, hence helping the TLBO obtained better solution in majority of the problems.

The SD indicator shows the spread of obtained solution for fitness in all optimizations runs. This indicator shows, how far the obtained solution is from the average fitness. The smaller SD indicates the worse exploration capability of an algorithm. In this category, ACO achieved the smallest SD value in 75% of the problems, EA and TLBO did so in 66%, while SA in 50% of test problems. Regarding this indicator, COA and FA secured its fourth and fifth position to achieve the best solution for SD in 50% and 41% respectively. The GWO and PSO struggled to achieve better solution for SD while GOO contributed by zero percent.

While TLBO's performance is quantitatively compared with several other algorithms, its superior outcomes can be attributed to its unique balance of exploration and exploitation. This characteristic enables TLBO to escape local optima more effectively than many traditional algorithms. Additionally, TLBO's adaptive refinement of solutions based on the best candidates enhances its ability to converge toward optimal solutions. Furthermore, its efficiency in managing multiple constraints aligns well with the complex requirements of the BACP, contributing to its overall performance advantage.

Table 4 shows the optimal result for the BACP-1 problem as a sample benchmark problem. As indicated in the table, the academic load was distributed across 10 semesters. The minimum total load was 17 in semester 10, while the maximum load was 30 in semesters 4 and 7. Note that the course assignments fully adhere to the prerequisite and load constraints.

Table 4. Optimal course assignment for BACP-1

Semester	Course	Load	Cumulative Load	Semester	Course	Load	Cumulative Load
1	1	6	6	6	25	6	6
	2	3	9		26	10	16
	9	4	13		27	10	26
	10	9	22				
	20	1	23				
	22	2	25				
	28	1	26				
	32	2	28				
2	3	5	5	7	24	7	7
	4	3	8		33	7	14
	6	8	16		34	9	23
	7	1	17		35	7	30
	12	8	25				
3	8	9	9	8	37	6	6
	11	8	17		38	7	13
	14	5	22		40	2	15
	15	6	28		41	5	20
					42	9	29
4	5	7	7	9	44	10	10
	13	4	11		45	4	14
	17	2	13		46	6	20
	19	3	16		47	4	24
	29	7	23				
	30	3	26				
	31	4	30				
5	16	3	3	10	48	5	5
	18	1	4		49	6	11
	21	1	5		50	6	17
	23	6	11				
	36	4	15				
	39	2	17				
	43	9	26				

For this solution, the academic load balance delay (*ABD*) was calculated as follows, which is exactly the fitness obtained in Table 3.

$$ABD = \left(100 - \frac{(28+25+28+30+26+26+30+29+24+17)}{(30\times10)} \times 100\right) \tag{11}$$
$$ABD = 12.33$$

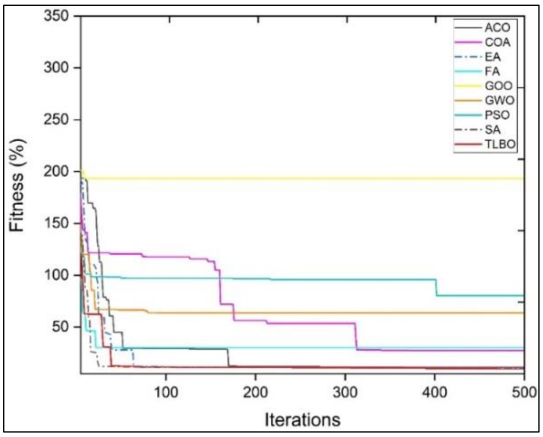
To evaluate the significance of the computational experiment's results, the Wilcoxon rank-sum test, a non-parametric statistical approach designed for analysing non-parametric data was employed. This test aimed to determine whether there was a significant difference between the fitness values obtained from TLBO and those from the compared algorithms. The null hypothesis asserts that no significant difference exists between the two data sets, while the alternative hypothesis indicates the opposite. If the resulting p-value exceeds the predefined significance level (alpha, typically set at 0.05), we accept the null hypothesis. However, if the p-value is below alpha, we reject the null hypothesis in favour of the alternative.

The p-values generated by the Wilcoxon rank-sum test, as presented in Table 5, revealed that 36% of the p-values were below 0.05. This indicates that in 36% of the cases, there was a significant difference between the fitness values of TLBO and the other proposed algorithms. Algorithms that struggled to achieve similar fitness levels as ITLBO in certain problems are marked with an asterisk ('*'). In 31% of the cases, the p-values were above the alpha threshold (0.05), suggesting that the fitness of these comparative algorithms closely matched TLBO, and arose from PSO and SA. Another interesting observation was that in 33% of the test problems, TLBO's fitness exactly matched that of the other comparative algorithms, resulting in infinite p-values, denoted by (∞).

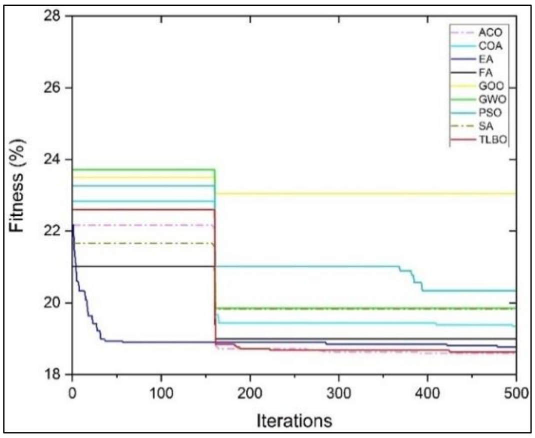
Table 5. Wilcoxon rank sum test (p-value)

Prob	ACO	COA	EA	FA	GOO	GWO	PSO	SA
BACP-1	∞	∞	∞	∞	0.0001	0.1681*	0.1675	∞
BACP -2	0.0593*	0.0313	0.3347*	0.0085	0.0001	0.0094	0.0058	0.0593*
BACP -3	∞	0.3681*	∞	∞	0.0001	0.1681*	0.0057	0.3681*
BACP -4	0.5828*	0.0093	0.1851*	0.1614*	0.0001	0.0038	0.0004	0.0021
BACP -5	0.4842*	0.0521*	0.5923*	0.0726*	0.0002	0.0884*	0.0124	0.0168
BACP -6	∞	0.0776*	∞	0.1675*	0.0001	0.0146	0.0146	0.00002
BACP -7	∞	∞	∞	0.1675*	0.0002	0.3681*	0.3681*	∞
BACP -8	∞	∞	∞	∞	0.1675	∞	∞	∞
BACP -9	0.3681*	0.0146	∞	0.3681*	0.0002	0.0002	0.0007	0.1681*
BACP -10	0.0767*	0.0060	0.3024*	0.0109	0.0001	0.0001	0.0016	0.0242*
BACP -11	∞	∞	∞	∞	0.0022	0.3681*	0.3681*	∞
BACP -12	∞	∞	∞	0.0012	0.0149	0.3681*	∞	∞

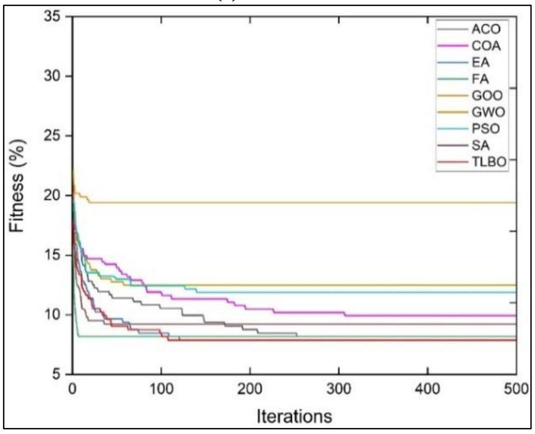
Fig. 3 illustrates the convergence profiles for various sizes and dimensions of BACP test problems. These curves highlight key characteristics of the different metaheuristic algorithms applied to the BACP problems. In this study, nine algorithms were tested across three different problem variants, with their convergence patterns plotted for comparison.



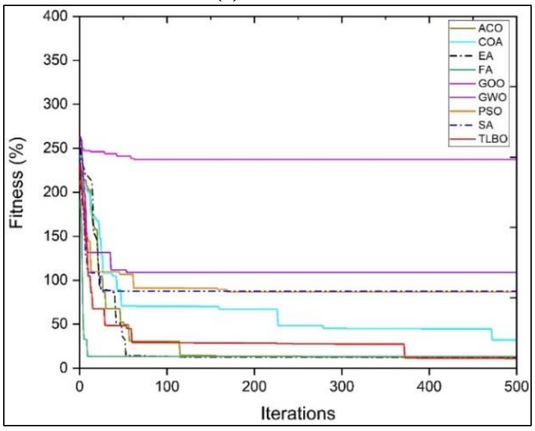
(a) BACP-2



(b) BACP-4



(c) BACP-9



(d) BACP-10

Fig. 3. Convergence Plot of BACP Optimization

The TLBO algorithm demonstrated consistent convergence across all three test problems, outperforming other algorithms throughout the iterations. The FA algorithm, in contrast, experienced a sudden drop-in fitness before 50 iterations and then navigated through an optimal trap region. SA, EA, GWO, and ACO also exhibited strong convergence, indicating a well-balanced trade-off between exploration and exploitation capabilities. The sharp decline before 100 iterations followed by a steady path suggests these algorithms' robust refinement and exploration abilities in searching for better solutions. These convergence profiles provide valuable insights for researchers optimizing various problems, helping them select the most stable and efficient algorithm. The plot effectively visualizes convergence across different algorithms, with iteration counts ranging from 300 to 500 and optimization runs from 10 to 20.

The TLBO algorithm stands out among other optimization techniques due to its unique, straightforward mechanism inspired by the teaching and learning process. Unlike other algorithms that require extensive parameter tuning, TLBO operates with minimal parameters, primarily focusing on the mean learning behaviour in a population.

This results in faster convergence and reduces computational complexity. Additionally, TLBO's dual-phase learning process, which includes the teacher and learner phases, ensures a robust exploration and exploitation of the search space, leading to higher-quality solutions. The simplicity, efficiency, and adaptability of TLBO make it a preferred choice in various optimization problems, particularly in scenarios where computational resources are limited.

The ACO algorithm emerges as the second-best performer due to its effective exploration and exploitation mechanisms, modeled after the foraging behaviour of ants, which balance the search for new solutions with the refinement of known good ones through pheromone trails. This adaptability allows ACO to consistently achieve high-quality solutions, reflected in its best mean fitness results in 66% of test instances. Conversely, the GOO algorithm performed the worst amongst all algorithms. The reason is premature convergence and is sensitive to parameter settings, which can affect its performance. It can be computationally expensive, especially for large-scale problems, and lacks a well-established theoretical foundation. Additionally, GOO's balance between exploration and exploitation can be challenging, and its complex implementation may deter practitioners.

The findings from the application of the TLBO algorithm in solving the BACP problem carry significant practical implications for educational institutions and curriculum planning. By optimizing course allocations while considering various constraints, educational institutions can achieve more balanced and efficient curricula, ultimately improving students' academic experiences and outcomes. The TLBO approach can also be extended to other complex optimization problems beyond academic scheduling, such as resource allocation, staff scheduling, and even broader logistical challenges in various industries, offering a versatile tool for decision-makers.

6. Conclusions

This paper presents an effective optimization approach for the Balanced Academic Curriculum Problem (BACP) using the Teaching-Learning Based Optimization (TLBO) algorithm. The model focuses on minimizing the total academic load balance delay, addressing a gap in the literature regarding robust algorithms for BACP. Computational experiments demonstrate that TLBO outperforms other algorithms in 83.3% of the cases tested.

However, the study has several limitations. Firstly, the performance of TLBO may vary depending on specific institutional contexts and constraints not captured in this analysis. There are also concerns regarding computational complexity; as the size of the dataset and the number of constraints increase, the computational time may rise significantly, affecting the algorithm's efficiency. Scalability poses a challenge, as TLBO may not perform optimally in extremely large or complex scheduling scenarios, which could limit its applicability in real-world settings.

Practical implementation of the TLBO algorithm in educational institutions presents additional challenges. Collecting real-world data can be inconsistent and difficult due to varying institutional policies. Moreover, parameter tuning may be necessary to adapt TLBO to specific contexts, requiring expertise and resources that may not always be available. Institutions should consider effective data management strategies and collaborative approaches for parameter optimization to leverage TLBO's full potential in practical scenarios.

Future research should enhance or hybridize TLBO with other algorithms to address real-world BACP scenarios in educational institutions, while expanding the dataset to include diverse academic settings for validation and robustness. Investigating TLBO's adaptability to varying constraints and objectives can provide valuable insights. Additionally, our study's assumption of fixed parameters may limit its applicability to dynamic scheduling scenarios, necessitating the exploration of adaptive models to better reflect real-world complexities. To strengthen our statistical analysis, employing tests like the Kruskal-Wallis test could offer a comprehensive understanding of performance differences among algorithms. Lastly, while this study focuses on minimizing academic load balance delay, it may overlook critical factors such as course diversity and student satisfaction; integrating these objectives into the optimization framework and utilizing multi-objective optimization techniques could enhance the assessment of curriculum effectiveness.

Acknowledgments

The authors would like to acknowledge Universiti Malaysia Pahang Al-Sultan Abdullah for funding this research under grant [PPU230113].

References

[1] J. M. Rubio, C. Vidal-Silva, L. Carter, and M. Tupac-Yupanqui, “Balanced Academic Curriculum: Looking For An Optimal Solution With Metaheuristics And Functional Programming,” Turkish Journal of Computer and Mathematics Education, vol. 12, no. 6, pp. 1181–1188, 2021.

[2] A. Katal, V. K. Singh, T. Choudhury, F. Imran, and M. S. Hossain, “Balancing Academic Curriculum Problem Solution: A Discrete Firefly-Based Approach,” Educ Res Int, vol. 2023, 2023, doi: 10.1155/2023/4300472.

[3] L. V Rosas-Tellez, V. Zanella-Palacios, and J. L. Mart ńez-Flores, “Solution of a Modified Balanced Academic Curriculum Problem Using Evolutionary Strategies,” in Computational Intelligence, K. Madani, A. Dourado, A. Rosa, and J. Filipe, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 49–58.

[4] L. Di Gaspero and A. Schaerf, “Hybrid Local Search Techniques for the Generalized Balanced Academic Curriculum Problem,” in Hybrid Metaheuristics in Lecture Notes in Computer Science, M. J. Blesa, C. Blum, C. Cotta, A. J. Fernández, J. E. Gallardo, A. Roli, and M. Sampels, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 146–157.
[5] N. Huntington-Klein and A. Gill, “Semester Course Load and Student Performance,” Res High Educ, vol. 62, no. 5, pp. 623–650, 2021, doi: 10.1007/s11162-020-09614-8.
[6] R. F. Szafran, “The Effect of Academic Load on Success for New College Students: Is Lighter Better?,” Res High Educ, vol. 42, no. 1, pp. 27–50, 2001, doi: 10.1023/A:1018712527023.
[7] T. Lambert, C. Castro, E. Monfroy, and F. Saubion, “Solving the Balanced Academic Curriculum Problem with an Hybridization of Genetic Algorithm and Constraint Propagation,” in Artificial Intelligence and Soft Computing – ICAISC 2006, L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, and J. M. Żurada, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 410–419.
[8] T. Lambert, C. Castro, E. Monfroy, M. C. Riff, and F. Saubion, “Hybridization of Genetic Algorithms and Constraint Propagation for the BACP,” in Logic Programming, M. Gabbrielli and G. Gupta, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 421–423.
[9] C. Castro, B. Crawford, and E. Monfroy, “A Genetic Local Search Algorithm for the Multiple Optimisation of the Balanced Academic Curriculum Problem,” in Cutting-Edge Research Topics on Multiple Criteria Decision Making, Y. Shi, S. Wang, Y. Peng, J. Li, and Y. Zeng, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 824–832.
[10] K. Sylejmani, A. Halili, and A. Rexhepi, “Balancing academic curricula by using a mutation-only genetic algorithm,” in 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017, pp. 1189–1194. doi: 10.23919/MIPRO.2017.7973604.
[11] M. Chakradhar, M. S. Charan, R. U. Sai, M. Kunal, Y. V. S. Murthy, and G. K. Shashidhar, “Academic Curriculum Load Balancing using GA,” in 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1–5. doi: 10.1109/ICCCNT45670.2019.8944897.
[12] M. Villalobos-Cid, M. Orellana, O. C. Vásquez, E. Pinto-Sothers, and M. Inostroza-Ponta, “Dealing with the Balanced Academic Curriculum Problem considering the Chilean Academic Credit Transfer System,” in 2019 38th International Conference of the Chilean Computer Science Society (SCCC), 2019, pp. 1–7. doi: 10.1109/SCCC49216.2019.8966411.
[13] L. V Rosas-Téllez, J. L. Martínez-Flores, and V. Zanella-Palacios, “Optimal Solution to the Problem of Balanced Academic Curriculum Problem Using Tabu Search,” in Proceedings of the International Conference on Genetic and Evolutionary Methods (GEM), The Steering Committee of The World Congress in Computer Science, 2012.
[14] L. V Rosas-Téllez, J. L. Martinez-Flores, and V. Zanella-Palacios, “Solution to the Balanced Academic Curriculum Problem Using Tabu Search,” Comput Technol Appl, vol. 3, no. 9, pp. 630–635, 2012.
[15] M. Chiarandini, L. Di Gaspero, S. Gualandi, and A. Schaerf, “The Balanced Academic Curriculum Problem Revisited,” Journal of Heuristics, vol. 18, no. 1, pp. 119–148, 2012, doi: 10.1007/s10732-011-9158-2.
[16] J. M. Rubio, C. L. Vidal-Silva, R. Soto, E. Madariaga, F. Johnson, and L. Carter, “Applying FireFly Algorithm to Solve the Problem of Balancing Curricula,” International Journal of Advanced Computer Science and Applications, vol. 10, no. 1, pp. 68–75, 2019, [Online]. Available: www.ijacsa.thesai.org
[17] B. Hnich, Z. Kiziltan, and T. Walsh, “Modelling a Balanced Academic Curriculum Problem,” in Fourth International Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimisation Problems , Le Croisic, France , Mar. 2002.
[18] J.-N. Monette, P. Schaus, S. Zampelli, Y. Deville, and P. Dupont, “A CP Approach to the Balanced Academic Curriculum Problem,” in The Seventh International Workshop on Symmetry and Constraint Satisfaction Problems, USA, Sep. 2007.
[19] L. V Rosas-Tellez, J. L. Martínez-Flores, and V. Zanella-Palacios, “Evolutionary Strategies for the Academic Curriculum Balanced Problem,” in International Conference on Evolutionary Computation Theory and Applications (FEC-2011), SciTePress, 2011, pp. 534–538.
[20] J. M. Rubio et al., “Solving the Balanced Academic Curriculum Problem Using the ACO Metaheuristic,” Math Probl Eng, vol. 2013, p. 793671, 2013, doi: 10.1155/2013/793671.
[21] S. Ceschia, L. Di Gaspero, and A. Schaerf, “The generalized balanced academic curriculum problem with heterogeneous classes,” Ann Oper Res, vol. 218, no. 1, pp. 147–163, 2014, doi: 10.1007/s10479-013-1358-8.
[22] Y. Z. Ünal and Ö. Uysal, “A new mixed integer programming model for curriculum balancing: Application to a Turkish university,” Eur J Oper Res, vol. 238, no. 1, pp. 339–347, 2014, doi: https://doi.org/10.1016/j.ejor.2014.03.015.
[23] A. Slim, G. L. Heileman, E. Lopez, H. Al Yusuf, and C. T. Abdallah, “Crucial based curriculum balancing: A new model for curriculum balancing,” in 2015 10th International Conference on Computer Science & Education (ICCSE), 2015, pp. 243–248.
[24] R. V. Rao, V. J. Savsani, and D. P. Vakharia, “Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems,” CAD Computer Aided Design, vol. 43, no. 3, pp. 303–315, Mar. 2011, doi: 10.1016/j.cad.2010.12.015.
[25] B. Hnich, Z. Kiziltan, and T. Walsh, “Problem 030: Balanced Academic Curriculum Problem (BACP),” CSPLib: A problem library for constraints. [Online]. Available: http://www.csplib.org/Problems/prob030
[26] Marco Dorigo, Vittorio Maniezzo, and Albert0 Colorni, “Ant System: Optimization by a Colony of Cooperating Agents,” vol. 26, no. 1, pp. 29–41, 1996, doi: 10.1109/3477.484436.
[27] H. Jia, H. Rao, C. Wen, and S. Mirjalili, “Crayfish optimization algorithm,” Artif Intell Rev, vol. 56, pp. 1919–1979, Nov. 2023, doi: 10.1007/s10462-023-10567-4.
[28] N. F. Johari, A. M. Zain, N. H. Mustaffa, and A. Udin, “Firefly algorithm for optimization problem,” in Applied Mechanics and Materials, 2013, pp. 512–517. doi: 10.4028/www.scientific.net/AMM.421.512.
[29] R. K. Hamad and T. A. Rashid, “GOOSE Algorithm: A Powerful Optimization Tool for Real-World Engineering Challenges and Beyond.”
[30] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” Advances in Engineering Software, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.
[31] F. Marini and B. Walczak, “Particle swarm optimization (PSO). A tutorial,” Chemometrics and Intelligent Laboratory Systems, vol. 149, pp. 153–165, Dec. 2015, doi: 10.1016/j.chemolab.2015.08.020.

Authors' Profiles

Mohd Fadzil Faisae AB Rashid received the bachelor's degree in mechanical (Industry) from Universiti Teknologi Malaysia in 2003, M. Eng (Manufacturing) from Universiti Malaysia Pahang in 2007 and Ph.D from Cranfield University, United Kingdom in 2013. During the beginning of his career, he worked in a multinational company as a Production Engineer. Later, he was appointed as a Tutor and Lecturer in Universiti Malaysia Pahang. During that period, he was awarded scholarships to pursue Master and Doctoral degrees. Currently, he is an Associate Professor in the Department of Industrial Engineering, College of Engineering, Universiti Malaysia Pahang. He is also a Chartered Engineer under the Institution of Mechanical Engineers. His research interests are in engineering optimization, particularly focusing on manufacturing systems, metaheuristics and discrete event simulation techniques.

Wasif Ullah is currently a full-time master's research student at Universiti Malaysia Pahang Al-Sultan Abdullah, within the Faculty of Manufacturing and Mechatronics Engineering Technology. He received his bachelor's degree in mechanical engineering from the University of Engineering and Technology Peshawar, Pakistan, in 2022. During his academic career, he worked as a trainee in various multinational production companies and was appointed as a teaching assistant during his master's program. His research interests include Artificial intelligence, Metaheuristics and computational optimization particularly in manufacturing systems.

How to cite this paper: Mohd Fadzil Faisae Ab Rashid, Wasif Ullah, "Optimization of Balanced Academic Curriculum Problem in Educational Institutions Using Teaching Learning Based Optimization Algorithm", International Journal of Modern Education and Computer Science(IJMECS), Vol.17, No.3, pp. 1-14, 2025. DOI:10.5815/ijmecs.2025.03.01