Hindawi
Education Research International
Volume 2023, Article ID 4300472, 16 pages
https://doi.org/10.1155/2023/4300472

Research Article

Balancing Academic Curriculum Problem Solution: A Discrete Firefly-Based Approach

Avita Katal,[1] **Vijay K. Singh**,[2] **Tanupriya Choudhury**,[1] **Faisal Imran**,[3] **and Md. Sabbir Hossain**[4]

[1]*School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248007, India*
[2]*School of Law, University of Petroleum and Energy Studies, Dehradun 248007, India*
[3]*Department of Computer Science and Engineering, Daffodil International University, Daffodil Smart City, Bangladesh*
[4]*Department of ICT Education, Daffodil International University, Daffodil Smart City, Bangladesh*

Correspondence should be addressed to Md. Sabbir Hossain; sabbir.ict@diu.edu.bd

Received 5 April 2022; Revised 14 October 2022; Accepted 18 October 2022; Published 9 February 2023

Academic Editor: Shi Yin

The difficulty of allocating a balanced educational syllabus to academic periods of a curriculum, also known as curriculum balancing, has long been a source of consternation for any institution of higher education attempting to connect learners and teachers. The balanced academic curriculum challenge entails assigning courses to academic times while adhering to all load restrictions and prerequisite requirements. The balanced academic curriculum problem (BACP) includes assigning subjects to class hours that fulfill standards even while managing students' burden in terms of credits, course load, and perquisites that includes subjects covered in the previous semesters/periods. The number of credits every semester corresponds to the academic load. As a result, educational frameworks must be "balanced," which means the credits for each period should be equivalent in order for students to bear minimum work. As a result, it is desirable to reduce this cost by developing a study plan that employs an algorithm that conducts this work automatically. Using an optimization method, this article provides a solution to the challenge of curricula balancing based on the discrete firefly algorithm (DFA). In research, FA has already been used to solve the BACP problem. However, the basic FA is modified to DFA with a local search mechanism inbuilt that helps to reach optimum solution in less number of iterations. A series of tests on standard and real data instances are done to check the efficiency of the suggested approach, with the objective of producing a platform that would simplify the procedure of building a curriculum system at institutions of higher learning. The results show that the proposed solution obtained a rather rapid solution and hit the recognized optimum in most of the iterations.

1. Introduction

A balanced curriculum is one in which activities and subjects for various grades of students are chosen in accordance with the overall development of the personality. A curriculum covers all elements of human activity and growth. This style of curriculum incorporates the learner's evolving interests as well as the changing surroundings. A well-rounded curriculum should foster the development of all key areas of human capability. A balanced curriculum balances scientific and arts topics, vocational/technical and academic, and urban and rural learners. It suggests an order in its breadth and advances all of the educational objectives mentioned. According to Ornstein and Hunkins, "A balanced curriculum is one in which students have opportunities to master knowledge and to conceptualize it in ways that are appropriate for their personal, social, and intellectual goals" [1].

In the academic world, time scheduling is a prominent study topic. It deals with a variety of real-world issues, such as rostering and job planning. Curriculum development, test administration, and course planning are some of the most essential planning responsibilities for academic organizations. A university's study curriculum (program) is typically built around some professional academic topic (e.g., computer science, economics, medicine, etc.). Such study curricula are made up of many courses that are generally divided into as

many study times as feasible (semesters/periods). Since certain subjects are much harder to understand over others, each subject is awarded a number of credits based on the number of study hours required to master the relevant information. Furthermore, certain subjects may not be taught before others due to prerequisites. In such cases, an obvious goal would be to arrange the courses over semesters such that the students' burden (in terms of learning hours) is balanced among different semesters. This is referred to as the balanced academic curriculum problem (BACP), which is also a constraint satisfaction problem (CSP). The objective of a CSP is to satisfy all of the relevant restrictions while improving the solution quality produced [2]. Many ways to resolve the BACP have been investigated, with this problem typically handled utilizing the idea of programing with restrictions and hybrid methods based on evolutionary algorithms, cooperative systems, and local searches, among other methods.

Many real-world issues may be represented as CSPs. Solving a CSP entails finding acceptable values (as solutions) to problem variables from the domain of variables so that all constraints of the problem are fulfilled. Formally, any CSP may be defined as follows: (1) a collection of variables, (2) a set of finite values as the domain for each variable, and (3) a set of constraints limiting the values that variables can take at the same time. Dealing with CSPs in acceptable ways has been a focus of attention since practically all real-world issues include restrictions, and so finding proper ways to handle these restricted problems has been critical. CSPs have typically worked with classical systematic procedures, the most well-known of which being backtrack search. However, because the systematic technique has an exponential cost, academicians have been seeking other algorithms for solving CSPs problems, such as graph coloring. N-queen problems are among the most well-known constraint fulfillment problems, as are implementations such as planning and management, arrangement, timetables, gear train layout, tension/compression spring architecture, and pressure vessel design problems. CSPs are NP-complete problems, which implies that addressing them utilizing typical methods like search strategy takes an increasing amount of time. A CSP with parameters and a scope size d will, in the worst case, need a backtrack search algorithm with O(d) computation time. Backtrack search is the most well-known systematic strategy for resolving CSPs. However, this method has certain downsides, the most prominent of which is thrashing. That is, the algorithm is unable to detect and recall the true source of the conflict and hence fails again for the same reason. Constraint propagation has been proposed to alleviate this limitation and is achieved utilizing look-ahead techniques such as maintaining arc consistency (MAC) and forward checking (FC). These methods enable the program to prune branching that would otherwise fail. In addition to pruning early of those branches of the search tree that would lead to failure, FC ensures local consistency between the present constant (the variable being assigned) and the future variables (the nonassigned variables associated with the current variable via constraints). Full local coherence on current and future variables is computed using MAC (also known as full look ahead or FLA). MAC has the advantage of predicting future constant clashes and so cutting more branches than FC. This, nevertheless, takes much longer than FC. While reducing the size of the problem space by FC or MAC (by deleting some contradictory variables from the variable ranges) increases the backtrack search's time efficiency, it still struggles from its exponential time cost, especially for difficult-to-solve problem scenarios. This is the primary incentive for the academic community in this domain to build new algorithms and leverage current powerful algorithms such as metaheuristics to solve these challenges.

Many solutions can be developed for a single issue, each having its own problem description and solution technique, as well as a separate formulation of the restrictions. As a result, a model's performance may differ from that of the other models. It is feasible that different models have complementary skills. Various problem concepts can be combined in this case to create a new paradigm, which combines the drawbacks of one paradigm with the benefits of the other and conversely. Combining numerous models can significantly increase the domain pruning done in each model, culminating in a more robust model than any of the participating systems. However, due to the increasing number of variables and constraints, the run-time may be extended. This is performed by imposing channeling constraints on the parameters of the collaborating simulations.

Initially, the BACP was suggested with the objective of arranging all courses within a syllabus to their respective semesters by meeting necessary dependencies links and preserving appropriate term burdens. BACP includes assigning subjects to teaching hours that fulfill standards while also managing students' credit and semester burden [3]. The BACP planning system is split into academic years that are then subdivided into further terms. The BACP, as presented in Ref. [3], is significantly NP-complete. Additionally, BACP is a fascinating issue since it stands at the crossroads of several problem categories, including scheduling, bin packing, and balancing [4]. Because of restrictions of preliminary, for example, BACP appears to be a scheduling issue. A single commodity (the learner) is managed by accounting for unit time (periods), activity (courses), and prerequisites (courses to be covered before others).

This article focuses on addressing the test cases recognized by the CSPLib [5], as well as a few real-world datasets taken from the course grid of the School of Computer Science and School of Law at University of Petroleum and Energy Studies, Dehradun, India. Firefly algorithm (FA) developed by Yang [6] is modified to discrete firefly algorithm (DFA) with a local search mechanism (LSM) developed to find the optimal solution. In this paper, DFA method is used to build a collection of solutions found via the linear programing technique and recorded in a binary matrix. We use DFA to maximize the space of initial solutions after finding a collection of valid solutions, each of which represents a firefly. A discretization strategy and LSM are developed to obtain the best possible solution.

Since BACP is the optimization problem, FA gives consistent results in the optimization problems. An optimized

problem seeks the ideal method among all available methods. In other words, the goal of an optimization problem is to discover a feasible solution that has the lowest (or highest) value of the desired parameter. The mathematical relationships between the goal, restrictions, and choice variables in an optimization problem impact how complex it is to solve, as well as the optimization techniques or methods, which can be utilized to get the truly optimal solution. According to Apostolopoulos and Vlachos [7], FA is a moderately efficient approach that outperforms other conventional means in data analysis using standard randomized design parameters. The algorithm operates on the basis of global communication among fireflies. As a result, it can identify both global and local optimum solutions at the same time. According to Yang [8], FA primarily uses genuine random numbers. Different fireflies operate separately, allowing for parallel implementation. According to Zhang et al. [9], FA is one of the techniques that academics have lately employed to tackle optimization challenges in dynamic environments.

The rest of the article is organized as follows: Section 2 discusses the relevant work, Section 3 provides the problem statement, Section 4 discusses the details of the approach employed, Section 5 discusses the outcomes of the recommended strategy, and Section 6 describes the conclusion and future work.

2. Related Work

Several techniques in the literature address and solve the BACP problem using deterministic and nondeterministic algorithms.

Castro and Manzano [10] introduced the BACP while presenting and constructing a linear programing model that took into account the following entities and constraints: (1) courses that are not optional (courses with their credits according to the academic curriculum), (2) educational periods relate to a defined amount of temporal periods, as per the syllabus, (3) the highest academic load permitted, (4) allowable academic load (minimum number of credits and courses for each academic period), (5) each course's approval criteria (students must take and approve some courses before others). The prerequisites and next courses enable the creation of ordered pairings of courses, (6) a balanced curriculum distribution, which means the number of credits of each semester should be equal. Thus, the BACP planning horizon divides a career into academic years, each of which is divided into course-taking periods. Previously, BACP was characterized as a variant of the generalized assignment issue, and programs were allocated to semesters while meeting prerequisites [11]. To summarize, BACP meets the prerequisite requirements; however, it does not have the goal of allocating relevant courses as near as feasible. It was not clear when to use what kind of heuristics for getting optimized results.

In terms of performance and solution quality, Hnich et al. [12] and Castro and Manzano [10] compared many BACP models depending on restriction and integer coding, Lambert et al. [13] created hybrid BACP approaches that used evolutionary algorithms and CP.

Di Gaspero and Schaerf [14] created the generalized BACP (GBACP) model, which extends the BACP models by including a lecturer preferences criterion. The study's findings enable them to construct a sophisticated mix of dynamic tabu search, simulated annealing, and large-neighborhood search. Furthermore, they provide six additional cases collected from their institutions that are substantially larger and more difficult than the CSPlib instances.

Chiarandini et al. [3] proposed a hybrid local search-based integer programing model and heuristic solution approach for GBACP. They created, implemented, and assessed local search heuristics. They gathered computational results on all new cases using the provided methodologies and evaluated the quality of solutions in relation to integer linear programming lower limits on a relaxed and deconstructed issue. The results demonstrate that a chosen heuristic finds excellent answers at 9%–60% distance from the lower bound.

Lee and Ma [15] developed the first generalized quadratic assignment problems (GQAP) formulation. They used it to locate numerous pieces of equipment at four manufacturing sites while reducing overall shipping and installation expenses. They provided three distinct linearization strategies as well as a branch and bound algorithm to obtain the best GQAP solution. This problem emerges in a variety of everyday situations, including facility placement and logistics network design. They refer to the problem as the GQAP and demonstrate that this relaxation significantly increases the problem's complexity. To solve the GQAP optimally, they provide three linearization techniques as well as a branch-and-bound algorithm. Mathematical experiments have been conducted to demonstrate the efficacy of the suggested approaches.

To solve a Lagrange formulation of GQAP, Hahn et al. [16] developed a branch and bound methodology based on a reformulation linearization technique (RLT) and a dual ascent mechanism. The GQAP specifies a broad class of quadratic integer programing problems in which M pairwise linked objects are distributed to N destinations with restricted capacity to receive them. This unique method is based on the dual ascent technique of the RLT.

Pessoa et al. [17] created two hybrid branch and bound algorithms based on Hahn et al. [16] research. The strategies, one utilizing the volume approach separately and the other integrating the volume method with the transformative lower bounding procedure, generate significantly higher upper and lower limits for relaxed GQAP. They also use transformational lower bounding approaches to increase the new procedure's performance. They offer detailed experimental data demonstrating that 19 of 21 cases with up to 35 capabilities may be addressed in a matter of days. There were six of these instances open.

McKendall Jr. [18] has shown how to formulate the dynamic space allocation issue as a GQAP. He devised three separate tabu search techniques that distribute idle resources to storage facilities over many time periods while lowering overall preparatory and two-way transportation costs. Simple tabu search heuristic is the first heuristic. The next heuristic augments the first using variety and amplification tactics, while the third strategy is a randomized tabu search heuristic.

The research uses a collection of test issues from the investigation to evaluate the efficiency of the heuristics. The results show that the tabu search techniques may be used to solve the dynamic space allocation problem. More crucially, for one-half of all tests, the suggested tabu search heuristic with alternative investments revealed fresh best solution in less computing time.

Hahn et al. [19] created a multiobjective 3D GQAP framework for inter assignment challenges that distributes division to various floors of the building while decreasing cross travel and evacuate costs. They created a one-of-a-kind edition of the multistory assignment problem, which merely allocates uneven area divisions to different levels inside the building, but also considers occupant escape. A thorough branch-and-bound approach based on an RLT1 dual ascent procedure has been provided in addition to the complete mathematical understanding of the equation and its evolutionary history from quadratic assignment concerns (QAP).

Lambert et al. [13] concentrated on the creation of a hybrid resolution architecture that takes evolutionary algorithms and constraint propagation. They create a conceptual framework wherein hybrid resolution is achieved by calculating a fixpoint of arithmetic operations. They aim to successfully solve BACP by merging a genetic algorithm (GA) with CP methods while also providing a generalized architecture to correctly develop such hybrid resolution procedures and highlight its features and attributes. The proposed framework enables the creation and management of new and refined problem-solving techniques and extensions. CP is still not able to discover a viable resolution in 10 min of CPU time in the suggested approach. GA can determine the best value; however, it is ten times slower than the hybrid approach GA + CP.

Chiarandini et al. [3] provide a genetic local search method to tackle the problem of curriculum balance by utilizing two objectives. They created a basic genetic local search algorithm. The technique employs a mutation-like operator (MSA) to perform simulated annealing. Consequently, depending on the temperature measurements, the algorithm does both exploitation and exploration. The operator considers the limitations by performing a relocation that results in only feasible options. Given the intricacy of the restrictions, they have chosen to solely utilize a MSA with no crossover. The primary drawback is that it takes longer to find the best answer.

Castro et al. [20] propose a customized BACP in which the educational workloads and variety of courses can be the same or vary for each period, and certain courses can be taken at specified times. This is considered as an integer programing challenge, and the utilization of Tabu search with short-term memory is suggested as a solution because no solution can be found for all situations of this changed issue using an exact technique.

Apart from the approaches that are mentioned earlier, there are various other approaches like constrained programing, integer linear programing, integer programing, and hybrid local search approaches are examples of hybrid approaches based on evolutionary algorithms, and constraint transmission have been used to solve the problem of curriculum balancing.

Hnich et al. [12] investigate a BACP. The task was to build a course syllabus by allocating hours to classes in such a way in which the educational load of every session is fair. They demonstrate how BACP problem may be represented in several ways and discuss why each model is beneficial. They then recommend combining the models in order to capitalize on the complementary characteristics of each model. Experiment findings demonstrate that integration greatly improves domain trimming and even reduces runtime in several cases. The main advantage is that the run time decreases due to the integration of different models.

For the enhancement of the search results, nowadays many researchers solve the curriculum-balancing problem using a metaheuristics algorithm. The major benefit of using a heuristic method is that they provide quick and viable solutions to planning and scheduling difficulties.

Rubio et al. [21] proposed best–worst ant system (BWAS) to solve the BACP. BWAS integrates evolving computation concepts, enabling for a balanced exploitation and exploration of the search area provided by a pheromone update technique and pheromone matrix mutation, respectively. Furthermore, a restart mechanism prevents the algorithm from being stuck and executing superfluous iterations. The experimental results demonstrate that artificial ants are effective at addressing constraint fulfillment issues. All of the tests demonstrate that the margin of error is significantly lower than the correct solutions provided. The solution's quality was acceptable for standard instances, and the ideal settings of every case were achieved in most cases. In practice, the outcomes were also positive. The proposed strategy has not been tested for the complicated versions of BACP datasets.

Slim et al. [4] seek to design a syllabus that is more applicable to real circumstances not only just by shortening the gap between essential courses but also by shifting those to the next accessible term while adhering to the BACP restrictions. To accomplish this objective, they suggest a crucial-based curriculum balancing (CBCB) model that is enacted as a multiobjective optimization issue with linear objective functions that also have a benefit over the suggested relevance-based curriculum balancing model. One drawback is that the authors did not include the course difficulty criteria. The primary drawback is that the academic load limit is stated in an inefficient manner.

Rubio et al. [22] use an optimization algorithm on the basis of the firefly attraction (FA) meta-heuristic to solve the BACP. They perform a number of experiments and actual events to evaluate the efficacy of their solution proposal, with both the objectives of producing a technology that will make the procedure of building a pedagogical system at universities easier. The collected findings demonstrate that their approach achieves a relatively rapid convergence and reaches the known optimum in the majority of the tests performed. They do not apply the LSM, which leads to results stuck in the local optima.

The abovementioned literature is either incomplete or cannot be used in the present scenario because of their

disadvantages like inefficient load constraint, not being tested for complicated data sets, taking too many iterations to reach to optimal solution, etc. Many of the approaches do not apply the LSM that leads to results stuck in the local optima. In this work, a DFA is used along with the discretization strategy and LSM. The suggested method can locate solutions in a limited number of iterations. Table 1 shows the comparison of literature review.

3. Problem Statement

The balanced academic curriculum challenge entails assigning courses to academic times while adhering to all load restrictions and prerequisite requirements that include the courses to be covered before the allocated course in the current period/semester. We describe the creation of a resolution to the BACP using the discrete firefly metaheuristic algorithm in this work. To obtain the most optimized solution, the FA has been modified to DFA, which includes the local search mechanism to make the algorithm not stuck in local minima.

4. The BACP Formulation

The following entities and restrictions are engaged in the creation of the BACP concern:

(1) Courses: The curriculum is made up of mandatory or nonoptional subjects that have credits assigned to it.
(2) Periods: The curriculum grid constructs a course based on a predefined number of time intervals. Courses to teach are included in each academic session. A 4-year curricular mesh, for example, comprises eight academic periods, with each year consisting of two periods (semesters).
(3) Maximum load: There is a maximal educational load for each session or number of credits allowed.
(4) Minimum load: A minimum academic load is permitted for each session.
(5) Prerequisites: The syllabus determines the order wherein courses must be delivered and approved; in other words, some courses should be given and authorized before others. These classes are called preliminaries, and they enable the formation of ordered pairs of courses and follow a specific order.
(6) Balanced load distribution: The curriculum meshes must be equal, which implies that the number of credits earned throughout each academic term should be similar, if not identical.

4.1. BACP Math Model. For issue resolution, the model employs a 1D decision variable and takes into account the following parameters:

c = total number of courses
p = the number of periods
β_i = number of credits of course i, where $i = 1,\ldots, m$
γ = minimal periodical academic load
δ = maximum periodical academic load
ε = a minimal amount of courses every semester
α = maximum number of courses allowed in a given period

The following are the decision variables:

A vector containing the periods allotted to each course:

$$p_i = q,\ \forall i = 1, \ldots, c. \tag{1}$$

For all periods, the maximum academic load is k:

$$k = \max\{k_1, \ldots, k_d\}. \tag{2}$$

The educational workload for a particular period q is denoted by the following:

$$k_q = \sum_{i=1}^{c} \beta_i \rho_i,\ \forall i = 1, \ldots, c;\ \forall q = 1, \ldots, d, \tag{3}$$

$$\text{where } \rho_i = \begin{cases} 1, & \text{if } p_i = q \\ 0, & \text{if } p_i \neq q \end{cases}$$

As a result, the objective function reduces the educational burden globally:

$$\min k. \tag{4}$$

The constraints are defined as follows:

For each course, i must be allocated to a period q:

$$\sum_{q=1}^{d} xiq = 1,\ \forall i = 1, \ldots, c. \tag{5}$$

A prerequisite exists for course h of period q:

$$x_{hq} \leq \sum_{s=1}^{q-1} x_{\text{es},}\ \forall q = 2, \ldots, d. \tag{6}$$

Equation (2) defines the maximum academic load as the one that satisfies the following restrictions:

$$k_q \leq k,\ \forall q = 1, \ldots, d. \tag{7}$$

The educational load for period q has to be more than or equivalent to the minimal required:

$$k_q \leq \gamma,\ \forall q = 1, \ldots, d. \tag{8}$$

The educational load for period q should be lower than or equivalent to the maximum required:

$$k_q \leq \delta,\ \forall q = 1, \ldots, d. \tag{9}$$

The number of period q courses must be more than or equal to the bare minimum:

$$\sum_{i=1}^{c} \rho_i \geq \alpha,\ \forall q = 1, \ldots d, \tag{10}$$

$$\text{where } \rho_i = \begin{cases} 1, & \text{if } p_i = q \\ 0, & \text{if } p_i \neq q \end{cases}$$

Table 1: Comparison of literature review.

Reference	Work done	Advantages	Disadvantages
[14]	Build a clever combination of tabu search, simulated annealing, dynamic, and large-neighborhood search	They have larger instances to compare	They do not have any web page for importing data
[3]	GBACP uses a hybrid local search-based integer programing paradigm and heuristic solution technique	The chosen heuristic finds high-quality answers within 9%–60% of the lower limit	The current approach does not compensate for the workload distribution of professors who teach over one course
[15]	To find the optimal GQAP solution, they gave three unique linearization algorithms as well as a branch and bound approach. They look into a modification of the quadratic assignment problem (QAP) that permits many pieces of equipment to be allocated to a single place as long as the site's resources allow	Three regularization approaches and a branch and bound algorithm were utilized to resolve the GQAP efficiently	Not mentioned
[16]	Describe a novel algorithm for the generalized quadratic assignment problem (GQAP)	Approach is based on the dual ascent technique of the reformulation linearization technique (RLT)	They made no mention of the development of heuristic (most likely meta-heuristic) strategies for getting good results
[17]	The strategies, one utilizing the volume algorithm separately and the other integrating the volume algorithm with the transformative lower bounding process, generate significantly higher lower bounds for relaxed GQAP	They also use transformational lower bounding approaches to increase the new procedure's performance	Not mentioned
[18]	Built three separate tabu search methods that distribute available resources to storage locations throughout several time periods while reducing the total of preparation and two-way transit expenses	Less computing time	The parameters for heuristics are more
[19]	3D GQAP model for multistory assignment challenges	A thorough branch-and-bound strategy based on an RLT1 dual ascent procedure has been provided in addition to the approach's complete mathematics and evolutionary history from quadratic assignment difficulties (QAP)	Not mentioned
[13]	They hope to solve BACP by taking a genetic algorithm with CP approaches while also giving a generic foundation for developing such hybrid solution methods and emphasizing its features and characteristics	The proposed framework enables the creation and management of new and refined problem-solving techniques and extensions	Not mentioned
[3]	Provide a genetic local search method to tackle the problem of curriculum balance by utilizing two objectives. They created a basic genetic local search algorithm	The technique employs a mutation-like operator (MSA) to perform simulated annealing. Consequently, depending on the temperature measurements, the algorithm does both exploitation and exploration	The primary drawback is that it takes longer to find the best answer
[20]	Customized BACP in which the educational workloads and variety of courses can be the same or vary for each period and certain courses can be taken at specified times	Utilization of tabu search with short-term memory is suggested as a solution because no solution can be found for all situations of this changed issue using an exact technique	Not mentioned
[12]	Combining the models in order to capitalize on the complementary characteristics of each model	Improves domain trimming and even reduces run-time in several cases	Not mentioned
[21]	Proposed the best–worst ant system (BWAS) to solve the BACP	A restart mechanism prevents the algorithm from being stuck and executing superfluous iterations	The proposed strategy has not been tested for the complicated versions of BACP datasets
[4]	Suggest a crucial-based curriculum balancing (CBCB) model that is enacted as a multiobjective optimization issue with linear objective functions that also has a benefit over the suggested relevance-based curriculum balancing model	Design a syllabus that is more applicable to real circumstances, not only just by shortening the gap between essential courses but also by shifting those to the next accessible term while adhering to the BACP restrictions	One drawback is that the authors did not include the course difficulty criteria. The primary drawback is that the academic load limit is stated in an inefficient manner
[22]	Optimization algorithm based on the firefly attraction (FA) meta-heuristic to solve the BACP	Relatively rapid convergence and reaches the known optimum in the majority of the tests performed	They do not apply the local search mechanism, which leads to results stuck in the local optima

The number of period q courses has to be fewer than or equal to the maximum allowed:

$$\sum_{i=1}^{c} \rho_i \leq \alpha, \ \forall q = 1, \ldots d, \tag{11}$$

where $\rho_i = \begin{cases} 1, & \text{if } p_i = q \\ 0, & \text{if } p_i \neq q \end{cases}$

4.2. Discrete Firefly-Based Optimization. Optimization problems are focused on identifying variable values that will result in the objective function's optimum functional value. This sort of problem occurs outside of our typical activities. Based on the choice variables, a solution may be classified into three types: continuous variables, noncontinuous variables, and mixed variables. A mixed problem occurs when some of the objective functions can be allocated with continuous data and the remainder with noncontinuous values. There are several approaches to solving an optimization issue. Metaheuristic algorithms are one type of solution approach. These algorithms are a nondeterministic solution method that uses an informed guess and a "trial and error" approach based on a provided randomness term to explore the solution space. Nature has served as an inspiration for the development of numerous metaheuristic algorithms. It has solved difficulties without being instructed but rather by experience. Natural selection and survival of the fittest were primary motivators for early metaheuristic algorithms. Different animals communicate with one another in a variety of ways. Swarm-based algorithms are a form of metaheuristic optimization method inspired by animal's social behavior [11].

Xin-She Yang created the FA in 2008 [23], which is one of the most current swarm intelligence approaches. FA is a stochastic, nature-inspired meta-heuristic method for tackling the most difficult optimization problems. There are over 2,000 firefly species, the majority of which generate brief, repetitive flashes. Typically, each species has a distinct flashing pattern. Bioluminescence causes the flashing light. It is thought that such flashes serve two primary functions: attracting mating partners (communication) and attracting prospective prey. Furthermore, flashing may function as a protective warning system. To summarize, fireflies use stored energy to flash as a light to marry, hunt, or avoid predators. Fireflies provide appeal by emitting light.

It is usually recognized that the light intensity at a set distance from the source of light follows the inverse square law [24], which asserts that the intensity of light decreases with rising distance between an observer and the source of light. Additionally, light is absorbed by the air, which diminishes with increasing distance. As a result, fireflies can only be seen for a limited distance. The FA assumes that the flashing light may be represented so that it is related to the objective function of the optimization problem. Three idealized rules underpin the FA [25]:

(1) Even though all fireflies are unisex, the amount of light they generate determines their attraction regardless of male or female.

(2) Fireflies' attraction is proportionate to their brightness. Therefore, for any two lighting fireflies, the one with the fewest flashes will fly toward the one with the most flashes. Attractiveness is related to brightness; therefore, they both fade as their distance increases, and for any two flashing fireflies, the less brighter one will go closer to the brighter one. If no firefly is brighter than another, it will fly at random.

(3) A fitness function determines the firefly's brightness.

The FA is a simple and efficient algorithm. It can also be implemented in parallel. However, studies demonstrate that for multimodal issues, it is sluggish to convergence and quickly becomes locked in the local optimum. Furthermore, the changes are completely based on current performance, with no recollection of prior best solutions and performances preserved. This may result in the loss of superior options. Furthermore, because the parameters are constant, the search behavior is consistent between iterations for any circumstance. As a result, one of the research concerns has been to improve the typical FA's performance. Furthermore, the typical firefly approach is created for continuous optimization issues; hence, it must be modified and altered to be used for noncontinuous situations. The majority of metaheuristic optimization approaches rely on the development of a random starting population of viable candidate solutions. All population candidates are placed in the solution search area with the purpose of directing the search to the optimal site. The FA adheres to the same premise. The following are the key steps of FA. The first stage is to initialize a swarm of fireflies, each of which is specified by the intensity of its flashing light. The brightness and attractiveness of each firefly are calculated at each subsequent iterative phase. After comparing the brightness of each firefly to the brightness of all other fireflies, the locations of the fireflies are updated depending on knowledge about fireflies and their neighbors.

In the proposed approach, the basic FA proposed by Xin-She is modified to DFA with inbuilt LSM to reach the optimal solution in less time and to avoid the algorithm to stuck in the local minima. The details for the discretization strategy and LSM are explained in Section 4.3.

4.3. BACP with Discrete Firefly. When using the FA technique to optimize issues, it is believed that the brightness is proportionate to the value of the objective function. The brightness of GAs may be described in the same manner that the objective function can. Therefore, because the attraction of a firefly is related to the amount of light it produces, the attraction may be defined as follows [26]:

$$\beta_{ji} = \beta_0 \times e^{-\gamma r 2} ij, \tag{12}$$

where β_0 represents the attractiveness at a distance of $r = 0$. Using the Cartesian distance technique, we compute the distance r_{ij} between two fireflies using Equation (13), which indicates the coefficient of light absorption related with the magnitude and nature of the actual problem.

1. Define the objective function
2. Initialize n fireflies, $x_i = (x_{i1}, x_{i2}, \ldots\ldots, x_{id})$
3. while(t < Iterations)
4. Identify each fireflies absolute brightness according to equation
5. for i=1:n for n fireflies
6. for j=1:n for n fireflies
7. if ($I_i<I_j$)
8. Determine the distance among fireflies i and j using equation 13
9. Determine the attractiveness of firefly j attracting i using equation 12
10. Relocate firefly i to j using equation 14
11. end if
12. end j
13. Using a discretization method, scatter placement after movement. // Modified FA (Discretization using sigmoid function)
14. end i
15. if (t > 2)
16. Using the local search technique, optimize the local optimum solution. // Local Search Mechanism
17. end if
18. end while
19. Obtain the optimum placement of courses in different semesters

Algorithm 1: Modified discrete firefly algorithm with local search mechanism.

The distance is calculated as follows [27]:

$$r_{ij} = \sqrt{\sum_{k=1}^{d} \left(x_{ik} - x_{jk}\right)2}. \quad (13)$$

A firefly i is attracted to a brighter firefly j, and its behavior is governed by Wu et al. [28]:

$$x_i(t+1) = x_i(t) + \beta_{ji} \times \left(x_j(t) - x_i(t)\right) + \alpha \times (\text{rand} - 0.5 \times A), \quad (14)$$

where x_i and x_j are the current locations of the fireflies, the next component is their attractiveness, and the third term adds a randomized factor in which there is a randomized variable and rand is a continuously dispersed randomized number among 0 and 1.

The main idea for solving the proposed BACP in this study is based on portraying the issue in a binary setup since there is proof that trying to raise the solution in this manner is useful without a doubt that FA adapts properly to a binary depiction. Algorithm 1 depicts the suggested DFA solution, where the objective function describes the problem's objective. Furthermore, it is required to initialize the firefly variables: γ, β_0, the population size n, and the maximum number of generations Iterations.

To resolve the BACP, a binary matrix structure of size $(m \times n)$ is provided, consisting of columns with courses and rows with periods, each with a square with a value of 0 or 1. This illustration demonstrates whether a course i is allocated to a period j (value 1) or not (value 0). When the fireflies move in accordance with Equation (14), the binary values will be converted into real numbers, resulting in new locations. As a result, in order to continue the search for a solution to this discrete problem, these real numbers must be transformed into binary integers. Sigmoid function is used to restrict the location following the firefly's movement in 0–1 [29]:

$$S(x_{ik}) = \frac{1}{1 + e^{-x_{ik}}}. \quad (15)$$

$S(x_{ik})$ in Equation (15) denotes the probability of x_{ik} value being 1.

The FA's discretization method is as follows:

Replace the largest value in each row with 1 and the remainder with 0. Calculate the load for each period, and if the load does not exist in the upper and lower thresholds, then replace the second largest value with 1 in each row and repeat the procedure until the correct period is determined. Repeat the preceding steps until all of the courses have been assigned to the proper periods.

The local search method is presented to enhance the quality of the optimum solution in order to accelerate the solution search process. The following local search method is implemented [30].

We begin with the first course in the first position and analyze the swap of courses in the mth and $(m+1)$th places. If the swap enhances the optimal solution, it will be executed; otherwise, no action is required to proceed to the next phase of the algorithm. This is done in each iteration after two iterations.

5. Results

The proposed algorithm is written in Java and runs on the BlueJ programing environment. Furthermore, the proposed approach was tested on a machine running Windows 7

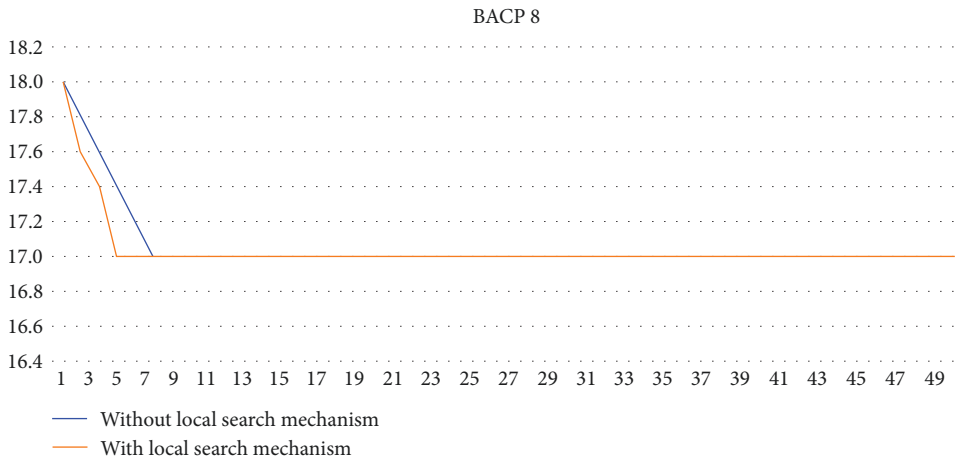

Figure 1: Number of iterations to reach optimal solution in BACP 8.

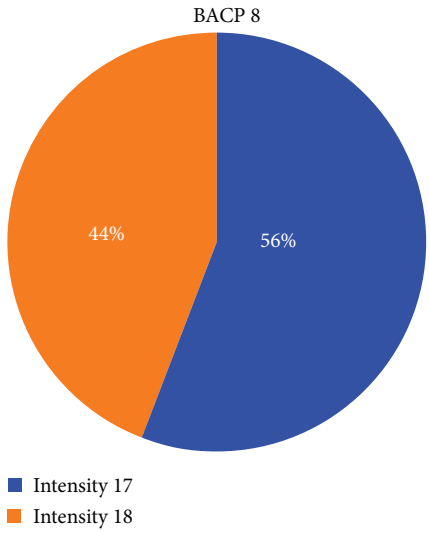

Figure 2: BACP 8 result distribution without local search mechanism.

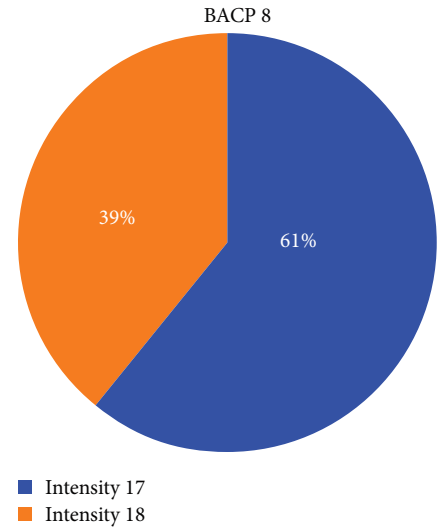

Figure 3: BACP 8 result distribution with local search mechanism.

Table 2: Variables utilized in the study.

Parameter	Value
Number of iterations	50
Number of fireflies	30
Alpha	0.5
Beta	1
Gamma	1

operating system with 64 bits. Our test PC features a 2.4 GHz Intel I3 CPU and 4 GB of RAM.

Each of the CSPLib events and the genuine occurrences were subjected to 50 tests. Real test data were taken from the course Grid of the School of Computer Science and School of Law at the University of Petroleum and Energy Studies in Dehradun, India, to validate the system's findings and performance when producing solutions. Table 2 shows the parameter values used to validate the new technique. These test samples are used to evaluate the algorithm's behavior and to compare it with the different proposed works that tackled the same problem. There are three test samples for the BACP issue, each of which is studied and solved individually.

5.1. Standard Test Cases. BACP 8: The BACP 8 is the smallest instance of the issue, with 46 courses spread throughout 8 academic periods. The results show that the optimal solution starts with 18 credits, and after performing six iterations, the optimal solution changes to 17 till the end of the execution when the LSM is not applied. After applying the LSM, the optimal solution of 17 is obtained by performing four iterations. Figure 1 depicts the number of iterations required in BACP 8 to find the best solution. Figures 2 and 3 show the distribution of results with and without LSM, respectively. Without LSM, 56% of iterations give the optimal solution, whereas with LSM, 61% of iterations give the optimal solution.

BACP 10: The maximum academic load in this instance is smaller than in BACP 8 since it consists of 42 courses distributed across 10 academic sessions. The system had a

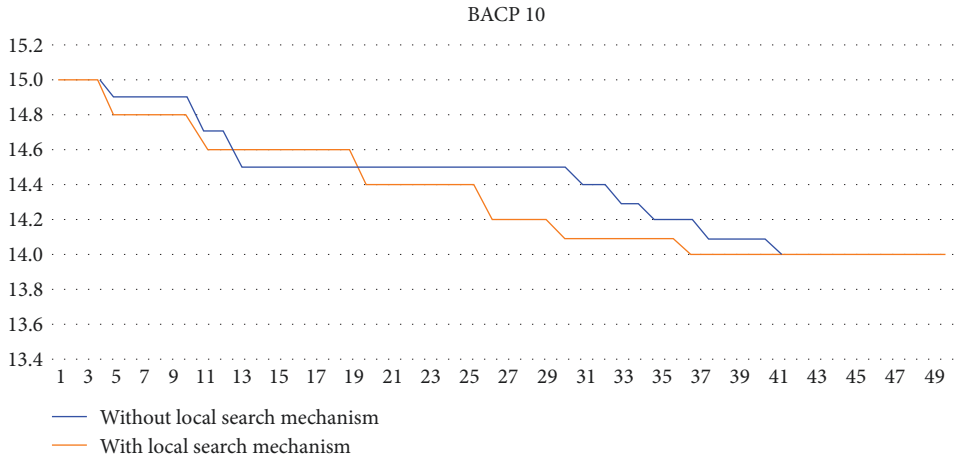

Figure 4: Number of iterations to reach optimal solution in BACP 10.

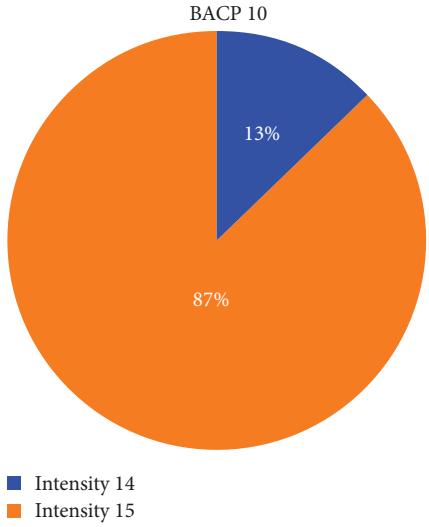

Figure 5: BACP 10 result distribution without local search mechanism.

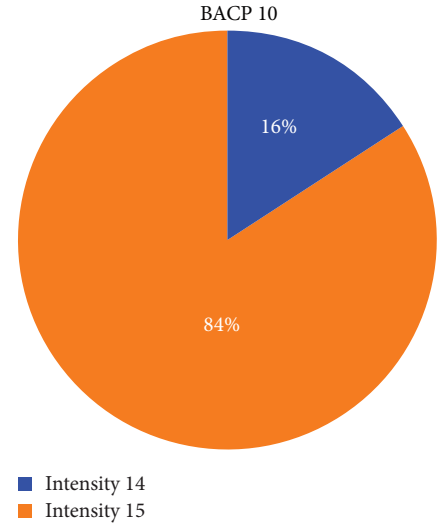

Figure 6: BACP 10 result distribution with local search mechanism.

harder time obtaining the known optimum in this test case. The results show that the optimal solution starts with 15 credits, and after performing 41 iterations, the optimal solution changes to 14 until the end of the execution when there is no LSM. After applying the LSM, the optimal solution of 14 comes by performing 36 iterations. Figure 4 depicts the number of iterations required in BACP 10 to find the best solution. Figures 5 and 6 show the distribution of results without and with the LSM, respectively. Without LSM, 13% of iterations give the optimal solution, whereas with LSM, 16% of iterations give the optimal solution.

BACP 12: The 12-period instance is the most complicated, with 66 courses to assign; therefore, the computing work necessary to discover appropriate answers is larger than in the other examples. The results show that the optimal solution starts with 20 credits, and after performing 12 iterations, the optimal solution changes to 18 until the end of the execution when there is no LSM. After applying the LSM, the optimal solution of 18 comes by performing 9 iterations. Figure 7 depicts the required number of iterations in BACP 12 to find the best solution. Figures 8 and 9 show the distribution of results without LSM and with LSM, respectively. Without LSM, 25% of iterations give the optimal solution, whereas with LSM, 29% of iterations give the optimal solution.

5.2. Real Test Cases. REAL 8: The REAL 8 instance consists of the 79 courses that should be distributed across the eight semesters/periods. It consists of the courses from the Bachelor of Technology in Computer Science with specialization in cloud computing and virtualization technology program that is taught at the University of Petroleum and Energy Studies. The results show that the optimal solution starts with 26 credits, and after performing eight iterations, the optimal solution changes to 23 till the end of the execution when there is no LSM. After applying the LSM, the optimal solution of 23 comes by performing 5 iterations. Figure 10 depicts the number of iterations required in REAL 8 to find the ideal solution. Figures 11 and 12 show the distribution of results without LSM and with LSM, respectively. Without LSM, 61% of iterations give the optimal solution, whereas with LSM, 67% of iterations give the optimal solution.

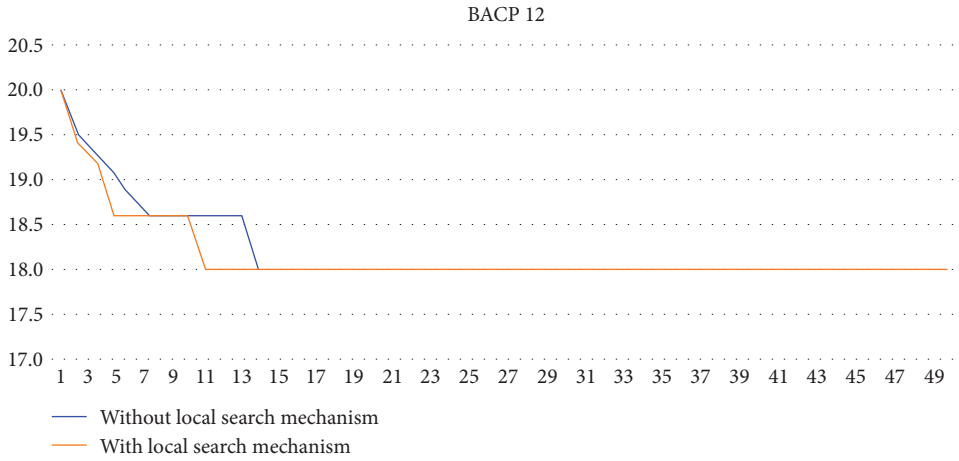

Figure 7: Number of iterations to reach optimal solution in BACP 12.

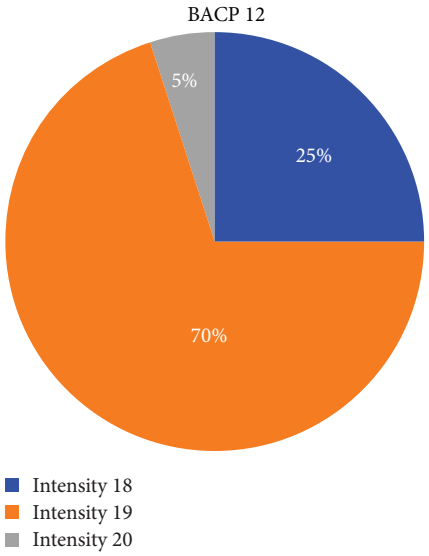

Figure 8: BACP 12 result distribution without local search mechanism.

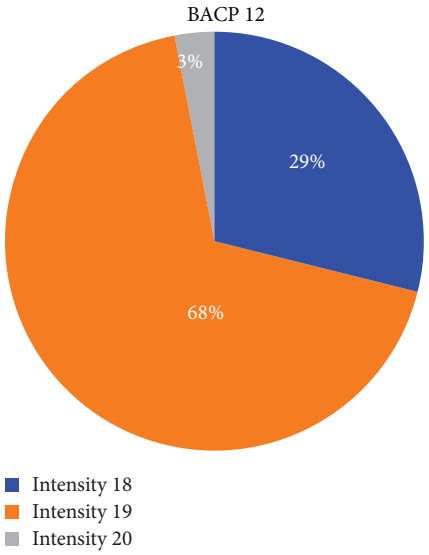

Figure 9: BACP 12 result distribution with local search mechanism.

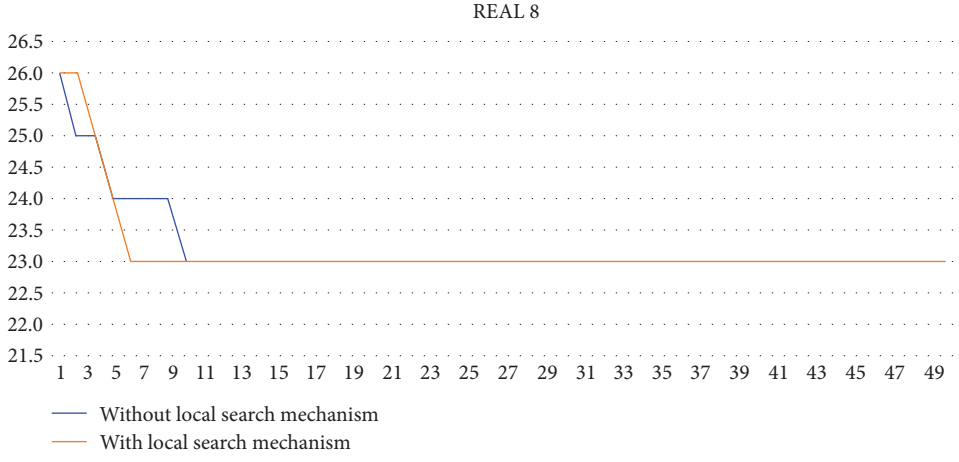

Figure 10: Number of iterations to reach optimal solution in REAL 8.

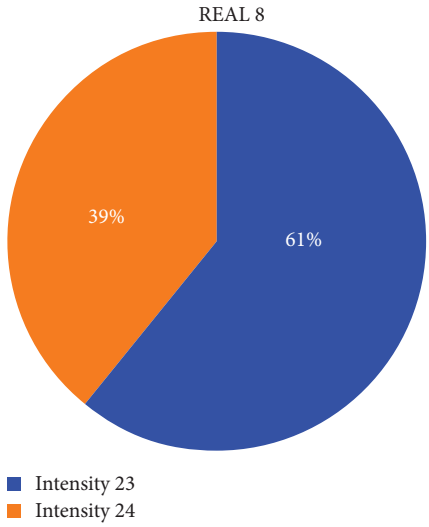

Figure 11: REAL 8 result distribution without local search mechanism.

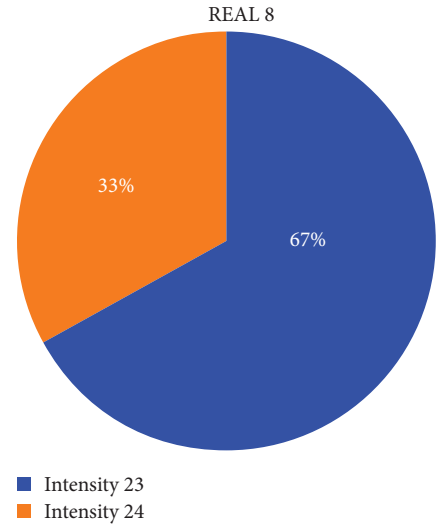

Figure 12: REAL 8 result distribution with local search mechanism.

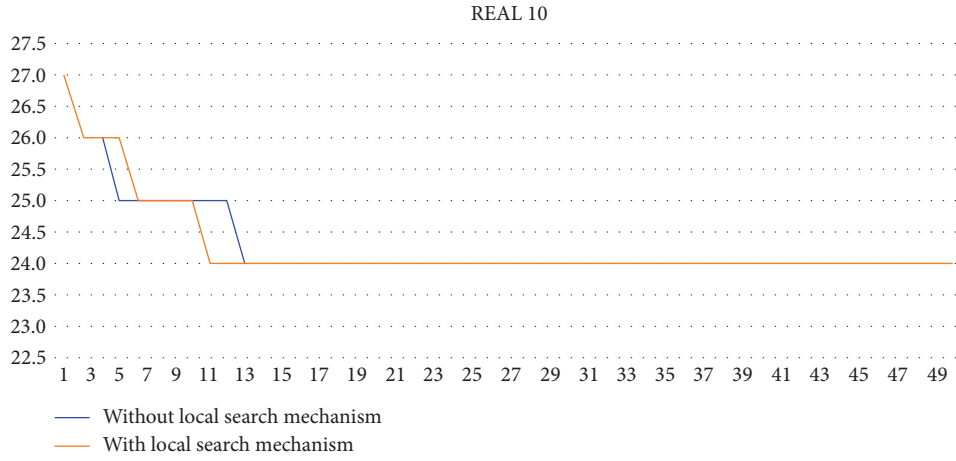

Figure 13: Number of iterations to reach optimal solution in REAL 10.

REAL 10: The REAL 10 instance consists of the 79 courses that should be distributed across the 10 semesters/periods. It consists of the courses from the BBA Bachelor of Laws (LLB) program that is taught at the University of Petroleum and Energy Studies. The results show that the optimal solution starts with 27 credits, and after performing 11 iterations, the optimal solution changes to 24 until the end of the execution when there is no LSM. After applying the LSM, the optimal solution of 24 comes by performing nine iterations. The number of iterations necessary in REAL 10 to get the optimal solution is depicted in Figure 13. Figures 14 and 15 show the distribution of results without LSM and with LSM, respectively. Without LSM, 51% of iterations give the optimal solution, whereas with LSM, 57% of iterations give the optimal solution.

REAL 12: The REAL 12 instance consists of 115 courses that should be distributed across the 12 semesters/periods. It consists of the courses from the Bachelor of Technology in Computer Science integrated with LLB program that is taught at the University of Petroleum and Energy Studies. The results show that the optimal solution starts with 30 credits, and after performing 14 iterations, the optimal solution changes to 26 until the end of the execution when there is no LSM. After applying the LSM, the optimal solution of 26 comes by performing 13 iterations. Figure 16 depicts the number of iterations required in REAL 12 to find the best solution. Figures 17 and 18 show the distribution of results without LSM and with LSM, respectively. Without LSM, 62% of iterations give the optimal solution, whereas with local search mechanism, 71% of iterations give the optimal solution.

Figure 19 depicts the comparison of all test instances with LSM and without LSM.

Table 3 shows the comparison of the optimal solution in discrete firefly algorithm with local search algorithm and discrete firefly algorithm without local search algorithm.

Table 3 shows that combining the local search strategy with the discrete firefly method minimizes the number of iterations required to obtain the best solution. Local optimizing is used to find the optimal solution for a small region of the search space, while global optimization is used for problems that have no local optima. A local optimization

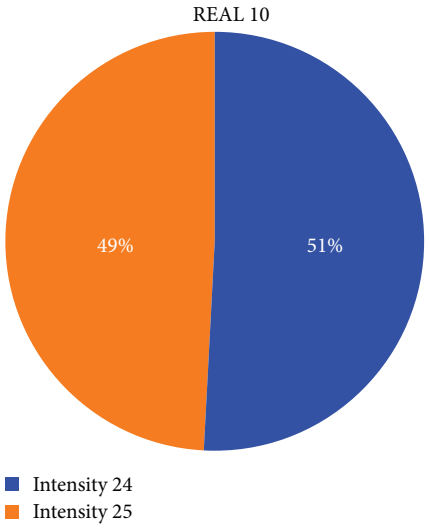

Figure 14: REAL 10 result distribution without local search mechanism.

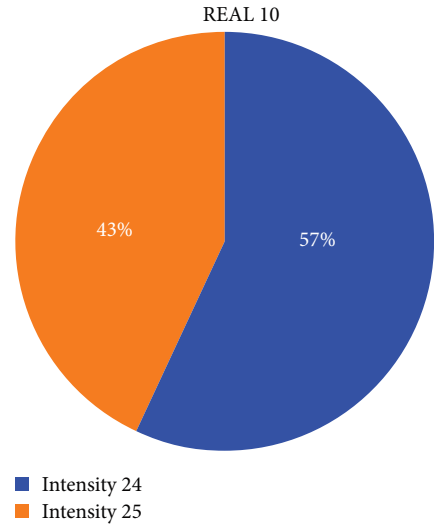

Figure 15: REAL 10 result distribution with local search mechanism.

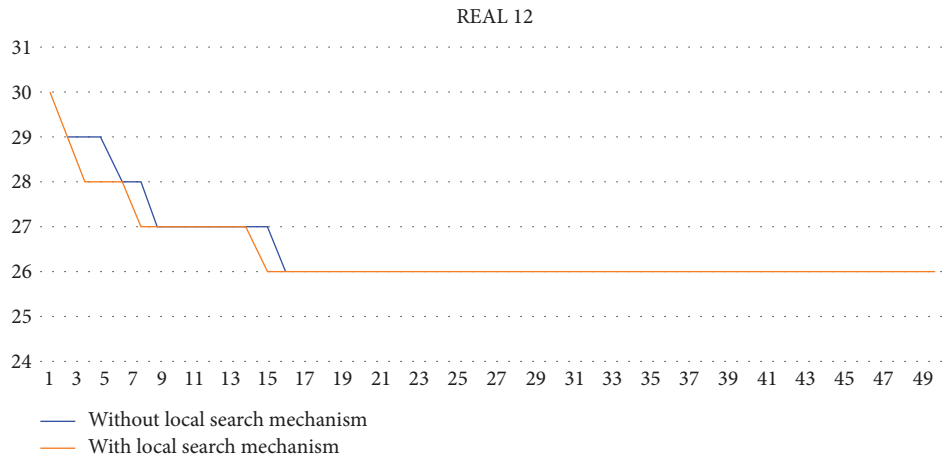

Figure 16: Number of iterations to reach optimal solution in REAL 12.

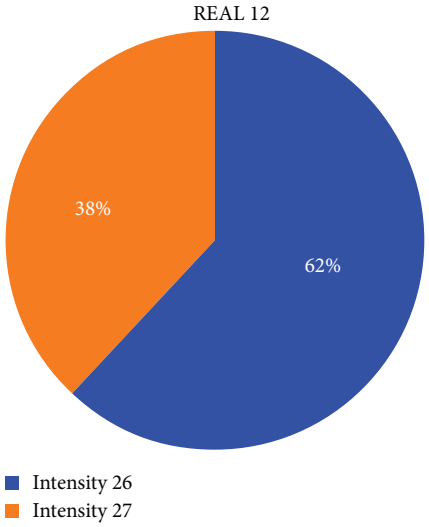

Figure 17: REAL 12 result distribution without local search mechanism.

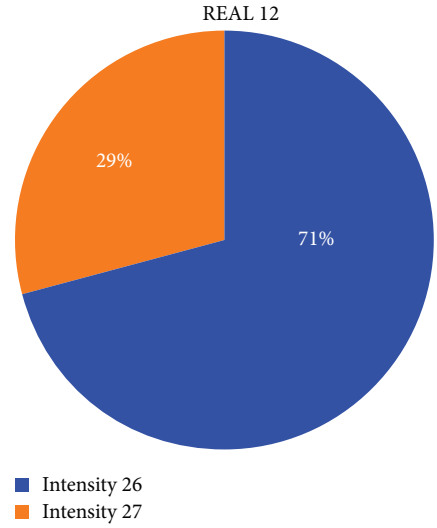

Figure 18: REAL 12 result distribution with local search mechanism.

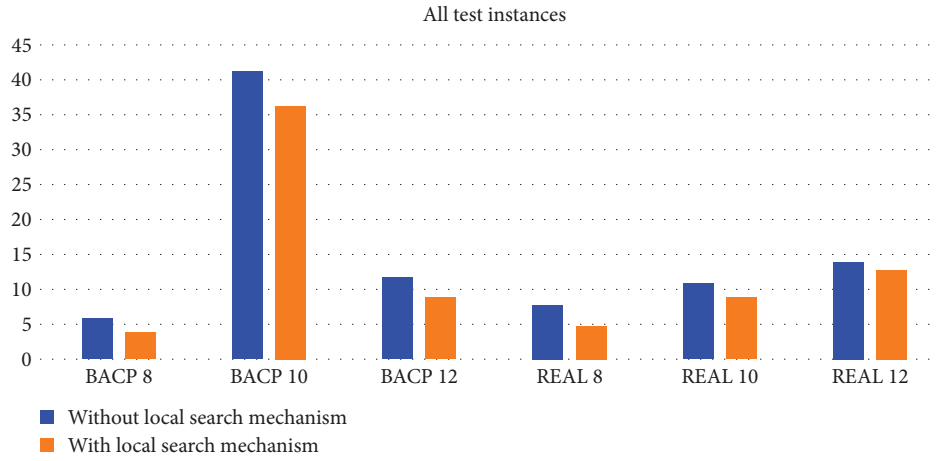

Figure 19: Comparison of all test instances with LSM and without LSM.

Table 3: Comparison of number of iterations required with local search mechanism and without local search mechanism.

Dataset	Number of iterations	
	With local search mechanism	Without local search mechanism
BACP 8	4	6
BACP 10	36	41
BACP 12	9	12
REAL 8	5	8
REAL 10	9	11
REAL 12	13	14

Table 4: Time execution with LSM and without LSM.

Dataset	Time of execution (ms)	
	With local search mechanism	Without local search mechanism
BACP 8	3,610	4,220
BACP 10	10,010	13,450
BACP 12	5,360	7,110
REAL 8	3,890	4,660
REAL 10	5,910	6,250
REAL 12	7,260	7,940

Table 5: Optimum values obtained after multiple iterations.

Solution	BACP 8	BACP 10	BACP 12	REAL 8	REAL 10	REAL 12
Best	17	14	18	23	24	26
Medium	17.4	14.6	18.6	24	25	27
Worst	18	15	20	26	27	30
Standard deviation	0.50	0.50	1.02	1.52	1.52	2.08
Optimum	17	14	18	23	24	26

algorithm, also known as a local search algorithm, is one that seeks out local optima. It is best suited for exploring a certain portion of the search space and coming near to (or finding exactly) the function's extrema in that region. Local search algorithms usually work on a single candidate solution, iteratively making modest modifications to the candidate solution and assessing the change to determine whether it results in an improvement and is adopted as the new candidate solution.

Table 4 shows the time execution with LSM and without LSM.

Table 5 shows the optimized load, medium load, and worst load. It also shows the standard deviation of all the values.

6. Conclusion

The BACP sought to organize all modules within a program at their relevant times while meeting the curriculum's standards, the necessary dependence linkages, and the maintenance of term workloads that are balanced. The BACP planned horizon is split into academic years, which are subdivided further into semesters. Every semester is a training process in which coursework can be assigned. The challenge is to locate a course assignment to a time that meets specific load limitations and requirements. In this paper, a DFA is used to find the optimal solution. The experimental assessment demonstrates the efficacy of artificial fireflies in solving such difficulties. All of the experiments conducted

on standard and real data set demonstrates that the DFA with inbuilt LSM helped to reach optimized solutions in maximum iterations. In addition, iterations number used to reach the optimum result are less as compared to the algorithm without LSM. The proposed algorithm DFA is thus computationally less expensive and faster. It will be fascinating to explore how this method performs when confronted with more complex real-world cases of the problem in the future, as well as to assess different variations of the approach. In the future, the complexity of the courses can also be considered for the proper allocation to the specific periods.

Data Availability

The open source data used to support the findings of this study is publicly available by CSPLIB (https://www.csplib.org/Problems/prob030/data/). The closed source data used to support the findings of this study is confidential to the organization and cannot be made public.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. C. Ornstein and F. P. Hunkins, *Curriculum Foundations, Principles, and Issues*, Pearson, 7th edition, 2018.

[2] J. Bulín, A. Krokhin, and J. Opršal, "Algebraic approach to promise constraint satisfaction," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019)*, pp. 602–613, Association for Computing Machinery, New York, NY, USA, 2019.

[3] M. Chiarandini, L. Di Gaspero, S. Gualandi, and A. Schaerf, "The balanced academic curriculum problem revisited," *Journal of Heuristics*, vol. 18, pp. 119–148, 2012.

[4] A. Slim, G. L. Heileman, E. Lopez, H. A. 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)*, pp. 243–248, IEEE, 2015.

[5] I. P. Gent and T. Walsh, "CSPlib: a benchmark library for constraints," in *Principles and Practice of Constraint Programming–CP'99*, J. Jaffar, Ed., vol. 1713 of *Lecture Notes in Computer Science*, pp. 480–481, Springer, Berlin, Heidelberg, 1999.

[6] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2010.

[7] T. Apostolopoulos and A. Vlachos, "Application of the firefly algorithm for solving the economic emissions load dispatch problem," *International Journal of Combinatorics*, vol. 2011, Article ID 523806, 23 pages, 2011.

[8] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms: Second Edition*, Luniver Press, 2022.

[9] Y. Zhang, L. Wu, and S. Wang, "Solving two-dimensional HP model by firefly algorithm and simplified energy function," *Mathematical Problems in Engineering*, vol. 2013, Article ID 398141, 9 pages, 2013.

[10] C. Castro and S. Manzano, "Variable and value ordering when solving balanced academic curriculum problems," October 2001.

[11] J.-N. Monette, P. Schaus, S. Zampelli, Y. Deville, and P. Dupont, "A CP approach to the balanced academic curriculum problem," 2007, https://api.semanticscholar.org/CorpusID:2663559.

[12] B. Hnich, Z. Kzfitan, and T. Walsh, "Modelling a balanced academic curriculum problem," *CiteSeer*, 2002, https://www.researchgate.net/publication/2521521_Modelling_a_Balanced_Academic_Curriculum_Problem, Accessed October 3, 2021.

[13] 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., vol. 4029 of *Lecture Notes in Computer Science*, pp. 410–419, Springer, Berlin, Heidelberg, 2006.

[14] L. Di Gaspero and A. Schaerf, "Hybrid local search techniques for the generalized balanced academic curriculum problem," in *Hybrid Metaheuristics*, vol. 5296 of *Lecture Notes in Computer Science*, pp. 146–157, Springer, Berlin, Heidelberg, 2008.

[15] C.-G. Lee and Z. Ma, "The generalized quadratic assignment problem," 2004, https://www.researchgate.net/publication/228575335, Accessed August 25, 2022.

[16] P. M. Hahn, B.-J. Kim, M. Guignard, J. MacGregor-Smith, and Y.-R. Zhu, "An algorithm for the generalized quadratic assignment problem," *Computational Optimization and Applications*, vol. 40, pp. 351–372, 2008.

[17] A. A. Pessoa, P. M. Hahn, M. Guignard, and Y.-R. Zhu, "Algorithms for the generalized quadratic assignment problem combining Lagrangean decomposition and the Reformulation–Linearization Technique," *European Journal of Operational Research*, vol. 206, no. 1, pp. 54–63, 2010.

[18] A. R. McKendall Jr., "Improved tabu search heuristics for the dynamic space allocation problem," *Computers & Operations Research*, vol. 35, no. 10, pp. 3347–3359, 2008.

[19] P. Hahn, J. MacGregor Smith, and Y.-R. Zhu, "The multi-story space assignment problem," *Annals of Operations Research*, vol. 179, pp. 77–103, 2010.

[20] 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., vol. 35 of *Communications in Computer and Information Science*, pp. 824–832, Springer, Berlin, Heidelberg, 2009.

[21] J.-M. Rubio, W. Palma, N. Rodriguez et al., "Solving the balanced academic curriculum problem using the ACO metaheuristic," *Mathematical Problems in Engineering*, vol. 2013, Article ID 793671, 8 pages, 2013.

[22] 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.

[23] G. Lindfield and J. Penny, "The firefly algorithm," in *Introduction to Nature-Inspired Optimization*, pp. 85–100, Academic Press, 2017.

[24] N. F. Johari, A. M. Zain, N. H. Mustaffa, and A. Udin, "Machining parameters optimization using hybrid firefly algorithm and particle swarm optimization," *Journal of Physics: Conference Series*, vol. 892, Article ID 012005, 2017.

[25] X.-S. Yang and X. He, "Firefly algorithm: recent advances and applications," *International Journal of Swarm Intelligence*, vol. 1, no. 1, pp. 36–50, 2013.

[26] X.-S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers*, vol. 29, pp. 175–184, 2013.

[27] R. B. Francisco, M. F. P. Costa, and A. M. A. C. Rocha, "Experiments with firefly algorithm," in *Computational Science and Its Applications*, vol. 8580 of *Lecture Notes in Computer Science*, pp. 227–236, Springer, Cham, 2014.

[28] J. Wu, Y.-G. Wang, K. Burrage, Y.-C. Tian, B. Lawson, and Z. Ding, "An improved firefly algorithm for global continuous optimization problems," *Expert Systems with Applications*, vol. 149, Article ID 113340, 2020.

[29] E. W. Weisstein, "Sigmoid function," From MathWorld–A Wolfram Web Resource, https://mathworld.wolfram.com/SigmoidFunction.html.

[30] M. K. Sayadi, R. Ramezanian, and N. Ghaffari-Nasab, "A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems," *International Journal of Industrial Engineering Computations*, vol. 1, no. 1, pp. 1–10, 2010.