

Applying FireFly Algorithm to Solve the Problem of Balancing Curricula

Jose' Miguel Rubio^{1*}, Cristian L. Vidal-Silva^{2*}, Ricardo Soto³, Erika Madariaga⁴, Franklin Johnson⁵, Luis Carter⁶

¹ Area Académica de Informática y Telecomunicaciones, Universidad Tecnológica de Chile INACAP, Santiago, Chile

²Ingeniería Civil Informática, Escuela de Ingeniería, Universidad Viña del Mar, Viña del Mar, Chile

³Escuela de Ingeniería Informática, Facultad de Ingeniería, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile

⁴Ingeniería Informática, Facultad de Ingeniería, Ciencia y Tecnología, Universidad Bernardo O'Higgins, Santiago, Chile

⁵Depto. Disciplinario de Computación e Informática, Facultad de Ingeniería, Universidad de Playa Ancha, Valparaíso, Chile

⁶Ingeniería Civil Industrial, Facultad de Ingeniería, Universidad Autónoma de Chile, Talca, Chile

Abstract—The problem of assigning a balanced academic curriculum to academic periods of a curriculum, that is, the balancing curricula, represents a traditional challenge for every educational institution which look for a match among students and professors. This article proposes a solution for the balancing curricula problem using an optimization technique based on the attraction of fireflies (FA) meta-heuristic. We perform a set of test and real instances to measure the performance of our solution proposal just looking to deliver a system that will simplify the process of designing a curricular network in higher education institutions. The obtained results show that our solution achieves a fairly fast convergence and finds the optimum known in most of the tests carried out.

Keywords—Balanced Academic Curriculum; Attraction of Fire-flies Meta-heuristic; Optimization

I. INTRODUCTION

At the time of designing curricular meshes for a study program in higher education, we consider factors such as the number of subjects for the career, the number of periods to assign those courses, the minimum and maximum acceptable academic load, and the number of courses per semester. We construct the curriculum using all this information as well as restrictions or academic regulations related to the curriculum through a trial and error approach until we achieve an adequate mesh.

If we measure the degree of effort required to pass a subject in credits, the academic success that students may achieve is directly related to the academic load they face in each period. The academic load corresponds to the number of credits per semester. It is for this reason that the curricular meshes must be “balanced,” that is, the number of credits for each period must be similar so that the load that the students face is the minimum possible. Therefore, it is of interest to minimize this cost by designing a study plan using an algorithm that performs this effort automatically and without errors. This problem is known in the literature as the Balanced Academic Curriculum Problem (BACP), and it is of the CSP (Constraint Satisfaction Problem) type. In a CSP, it is sought to satisfy all the associated constraints and then to optimize the quality of the solution found. Several models that solve the BACP have been studied, where this problem has generally been approached using

the paradigm of programming with restrictions and hybrid algorithms using genetic algorithms, collaboration schemes, and local searches, among other techniques.

The present work focuses on solving the test instances recognized by the CSPLib [1], and a few real situations of the problem in the study programs of the computer area of two Chilean universities. To solve the BACP, we use the optimization meta-heuristics based on the behavior of fireflies or Firefly Algorithm (FA) proposed by Xin-She Yan [2].

In the present investigation, we apply the FA algorithm to a set of solutions previously found by the whole linear programming method and represented in a binary matrix. After obtaining a set of valid solutions, considering that each one expresses a firefly, we proceed to optimize the space of initial solutions through FA to be able to find an optimal solution.

The rest of the article follows the next structure: Section 2 presents the development of the theoretical analysis of the BACP problem, Section 3 describes a math model for the BACP and gives ideas of the firefly optimization, Section 4 describe the classic firefly algorithm and ideas about how to apply it on the BACP, Section 5 gives application results of the firefly optimization on test and real cases of the BACP problem, Section 6 summarizes related work, and Section 7 gives final ideas and conclusions of our research work.

II. BACP BACKGROUND THEORY

The BACP was initially introduced and developed by Castro and Manzano in [3] who proposed a whole linear programming model to considers the following entities and restrictions:

- Courses: The curriculum considers a set of mandatory courses, that is, non-optional, which have credits assigned.
- Periods: The curricular mesh composes a curriculum that corresponds to a fixed number of time intervals (academic periods). Each academic period includes courses to teach. For example, a curricular mesh of 4 years contains 8 academic periods, and each year consists of 2 periods (semesters).
- Maximum load: For each period there is a maximum academic load allowed, that is, a maximum number of credits allowed.

* Corresponding author

- Minimum load: For each period there is a minimum academic load allowed.
- Prerequisites: The curriculum contains a defined order in the courses, that is, some courses must be taught and approved before than others. These courses are called prerequisites which permit generating ordered pairs of courses in which the restriction is that a student must pass the first course before taking the second course.
- Balanced distribution of the load: The curricular mesh should be balanced, that is, the number of credits of each academic period should be similar, ideally equal.

The work of [4] define an structural and behavioral models for the BACP problem whereas the work [3] appreciate it as a constraints problem.

The main objective of this research is to find an allocation of courses for each period that satisfies all the mentioned entities and restrictions in an optimal way.

We use the test instances of the CSPLib (BACP8, BACP10, and BACP12) to validate our proposed algorithm. For example, considering the BACP8, this curriculum is made up of 46 courses with a total of 133 credits to be taught in 8 academic periods. Consequently, the simple arithmetic average of credits per period is equivalent to $133/8 = 16,625$, which implies that the lower limit of the maximum number of credits per period is 17. Therefore, solutions with value 17 are optimal.

III. BACP MATH MODEL AND FIREFLY OPTIMIZATION

First, we describe a math model of the BACP to appreciate it as an optimization problem: and second, we present a background about the Firefly meta-heuristic.

A. BACP Math Model

We propose an integer linear programming model based on [3]. This model uses a decision variable of one dimension for the resolution of the problem and considers the following parameters:

- m : Number of courses.
- n : Number of academic periods.
- α_i : Number of credits of course i , where $i = 1, \dots, m$.
- β : Minimum academic load per period.
- γ : Maximum academic load per period.
- δ : Minimum number of courses per period.
- ϵ : Maximum number of courses per period.

The decision variables correspond to:

- A vector with the periods assigned to each course:

$$x_i = j, \forall i = 1, \dots, m \quad (1)$$

- The maximum academic load for all periods is c :

$$c = \max\{c_1, \dots, c_n\} \quad (2)$$

- The academic load for a period j is defined by:

$$c_j = \sum_{i=1}^m \alpha_i \zeta_i, \forall i = 1, \dots, m; \forall j = 1, \dots, n \quad (3)$$

$$\text{where: } \zeta_i = \begin{cases} 1, & \text{if } x_i = j \\ 0, & \text{if } x_i \neq j \end{cases}$$

Thus, the objective function globally minimizes the academic load:

$$\min c \quad (4)$$

We define the following restrictions:

- Every course i must be assigned to a period j :

$$\sum_{j=1}^n x_{ij} = 1, \forall i = 1, \dots, m \quad (5)$$

- Course b of a period j has a prerequisite:

$$x_{bj} \leq \sum_{r=1}^{j-1} x_{ar}, \forall j = 2, \dots, n \quad (6)$$

- The maximum academic load is defined by Eq. (2) the one that obeys the following set of linear constraints:

$$c_j \leq c, \forall j = 1, \dots, n \quad (7)$$

- The academic load of period j must be greater than or equal to the minimum required:

$$c_j \geq \beta, \forall j = 1, \dots, n \quad (8)$$

- The academic load of period j must be less than or equal to the maximum required:

$$c_j \leq \gamma, \forall j = 1, \dots, n \quad (9)$$

- The number of courses of period j must be greater than or equal to the minimum required:

$$\sum_{i=1}^m \zeta_i \geq \delta, \forall j = 1, \dots, n \quad (10)$$

$$\text{where: } \zeta_i = \begin{cases} 1, & \text{if } x_i = j \\ 0, & \text{if } x_i \neq j \end{cases}$$

- The number of courses of period j must be less than or equal to the maximum required:

$$\sum_{i=1}^m \zeta_i \leq \epsilon, \forall j = 1, \dots, n \quad (11)$$

$$\text{where: } \zeta_i = \begin{cases} 1, & \text{if } x_i = j \\ 0, & \text{if } x_i \neq j \end{cases}$$

B. Optimization based on Fireflies

Optimization based on fireflies is one of the newest heuristics inspired by natural behaviors for optimization problems. For the work of [5] [6] [7], we know that fireflies possess an unmistakable characteristic glow, and even people who have not seen one in their life know that they emanate a light.

For the work of [7], the foundation of the algorithm focuses on the brightness of fireflies which, in their need to mate, approach to a other fireflies releasing attractive light. It is this behavior that has given rise to the optimization algorithm which considers as an objective function the brightness of the fireflies and their need to approach the most brilliant or optimal firefly.

There are different types of insects and animals in nature and, depending on each species, present different organization, communication, and skill which make each element of the colony can fulfill an objective looking for a common good. A firefly presents an example of communication in which it combines the absorbed oxygen by individual cells with a substance called luciferin and reacts by producing light without hardly generating heat. The light of the firefly is usually intermittent and shines in a specific way in each species. Such as [2] argue, each approach to shine is an optical signal that helps the fireflies to find possible pairs.

Firefly Algorithms (FA) especially solves multi-modal optimization problems. The work of [8] details that there exist records of continuous FA applications in optimization problems, traveler's problem (TSP), segmentation (clustering) tasks, image processing, and feature selection problems.

According to the work of [5] and [6], three essential properties of the behavior of the FA are identified:

- All fireflies are unisexual and are attracted to other fireflies, regardless of their sex.
- The value of the objective function determines the brightness of a firefly, that is, for a maximization problem, the brightness of each firefly is proportional to the value of the objective function and vice versa.
- The degree of attraction of a firefly is proportional to its brightness, and therefore for any pair of blinking fireflies, the one that is less bright will move towards the brighter one. More brightness means less distance between two fireflies. However, if the two flickering fireflies have the same brightness, they randomly move.

We obtain the formulas and procedures for the operation of FA meta-heuristics from the analysis of these properties which the following section explains in details.

IV. FA ON THE BACP

We must consider the brightness proportional to the value of the objective function to apply the algorithm FA in the optimization problems. In the case of genetic algorithms, we can define the brightness in the same sense as the objective function.

Since the attractiveness of a firefly is proportional to the light emanating, we can define this attraction as:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (12)$$

where β_0 is the attractiveness at a distance $r = 0$. We calculate the distance r_{ij} between two fireflies using the Cartesian distance method, and γ represents the light absorption coefficient that is associated with the scale and nature of the problem.

A firefly i is attracted to a brighter firefly j and its movement is determined by:

$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha(\text{rand} - 1/2) \quad (13)$$

Where x_i and x_j are the current position of the fireflies, the second term corresponds to their attraction, and the third term introduces a random component in which α is a randomization parameter and rand a uniformly distributed random number between 0 and 1 .

Different authors have already demonstrated the ability of the FA algorithm to solve optimization problems. On the other hand, in [5] the effectiveness of this algorithm has been shown to solve problems with binary representation.

The main idea to solve the BACP proposed in this paper is based on representing the problem through a binary arrangement because evidence exists that raising the solution in this way is useful and we can say undoubtedly that the fireflies algorithm adapts perfectly to a binary representation like the one that is presented later in this article.

Algorithm 1 [9] shows the general scheme of the FA solution for which the objective function defines the objective of the problem. Besides, and it is necessary to initialize the firefly parameters: γ , β_0 , the size of the firefly population n , and the maximum number of generations *MaxGeneration*.

Algorithm 1 Pseudocode of Algorithm FA

```
Objective function  $f(x)$ ,  $x = (x_1, \dots, x_d)^t$ 
Generate an initial population of  $n$  fireflies  $x_i$  ( $i = 1, 2, \dots, n$ )
The intensity of light  $I_i$  in  $x_i$  is determined by  $f(x_i)$ .
Define the light absorption coefficient  $\gamma$ 

while  $t < \text{MaxGeneration}$  do
  for  $i \leftarrow 1$  to  $n$  do ▷ for all the  $n$  fireflies
    for  $j \leftarrow 1$  to  $n$  do ▷ for all the  $n$  fireflies
      if  $I_i < I_j$  then
        to move firefly  $i$  to firefly  $j$ 
      end if
    end for
  end for
  To classify the fireflies and find the best global optimum  $g$ 
end while
To Process and visualize
```

We propose to apply the algorithm FA to solve the whole linear programming model of the BACP previously described. Fig. 1 shows the three simple steps necessary for our

solution: first, an instance of the problem is loaded; second, the algorithm FA is applied in a binary representation of the problem; and third, finally a representation of the best solution found by the algorithm FA is obtained.

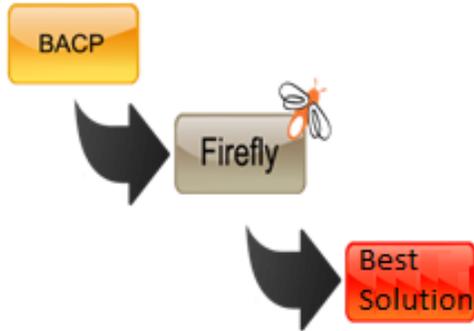


Fig. 1: Scheme of proposed solution.

A set of constraints defines each instance of the problem, that is, the set of parameters and decision variables mentioned at the beginning of this section. The courses and academic periods form a curricular mesh (or curriculum) that can be represented by a binary matrix on which we can apply the FA algorithm.

In addition to the characteristics of the problem, we need to take into account the following steps: (1) to define the absorption coefficient, (2) to generate an initial population of fireflies, (3) to define the attractiveness coefficient, and (4) to determine the maximum number of generations.

When we want to solve a problem, according to [9] [10], an important aspect is to look for a simple, logical and useful representation on which we are capable of solving the problem efficiently and effectively.

To solve the BACP, we have proposed a binary matrix representation of dimensions $m \times n$, that is, rows with periods and columns with courses, with the purpose of that the matrix has squares with possible values 0 or 1. This representation will indicate if a course i is assigned to a period j (value 1), or, on the contrary, that the course i does not correspond to that period j (value 0). Fig. 2 shows the base representation for a possible solution to the problem.

		Periods							
Courses	1	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	0	0
	0	0	0	0	0	0	0	0	1
	0	0	1	0	0	0	0	0	0
	0	0	0	0	1	0	0	0	0
	0	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1

Fig. 2: Scheme of proposed solution.

V. RESULTS

The proposed solution was implemented in the Java programming language using the Netbeans IDE 8.0 programming environment. Besides, our solution ran in a computer using Windows 7 Ultimate 64-bit operating system. Our computer to run the tests has a 3.4 GHz Phenom II X4 processor, and 8 GB of RAM.

The algorithm FA must be modified so that it can operate with binary representations. When applying the formula of motion to a firefly, the algorithm generates values that do not meet the conditions defined in the proposed binary matrix to represent the solution mainly due to the fact that, when applying the movement formula in each of the matrix dimensions, we will obtain a real value that must be transformed into binary by means of a transfer function.

A transfer function is a mathematical model that, through a quotient, relates the modeled response of a system to an also modeled input or excitation signal.

In control theory, transfer functions are often used to characterize the input and output relationships of components or systems that are described by linear differential equations and time-invariants. In this paper, we use the following transfer function [5]:

$$\text{Tanh}(|X_p|) = \frac{\exp(2|X_p| - 1)}{\exp(2|X_p| + 1)} \quad (14)$$

To pass the values from a continuous search space to a discrete one, the rule used in [5] follows:

$$x_i^k(t+1) = \begin{cases} 1, & \text{if rand} < T(x_i^k)(t+1) \\ 0, & \text{in other case} \end{cases} \quad (15)$$

Where rand is a random number between 0 and 1 evenly distributed, $x_i^k(t)$ is the value of the dimension k of a firefly i at the iteration t .

TABLE I: PARAMETERS USED IN EXPERIMENTS

Parameter	Value
Iterations number	1000
Fireflies number	30
α	0.5
β	1
γ	1

To validate the results of the system, and its performance at the time of generating solutions, 50 tests were performed on each of the instances available in the CSPLib. The parameters used to solve these test instances are those presented in Table 1. Those test instances serve to measure the behavior of the algorithm and are also the basis of comparison with the work of other authors who have solved the same problem. For the BACP problem, there are 3 test instances which are analyzed and independently solved.

A. Test Cases

- BACP8: The BACP8 corresponds to the smallest instance of the problem, with 46 subjects distributed in 8 academic periods. In Fig. 3 you can see how the results obtained are distributed.

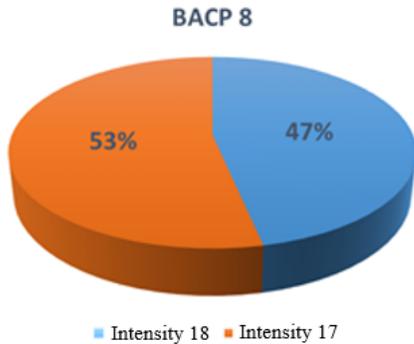


Fig. 3: BACP8 results.

The convergence graph of the algorithm (see Fig. 4) shows how the system finds an optimal initial solution of 18 credits and changes rapidly to 17 in the first iterations, staying at that optimum until the execution of the algorithm completes.

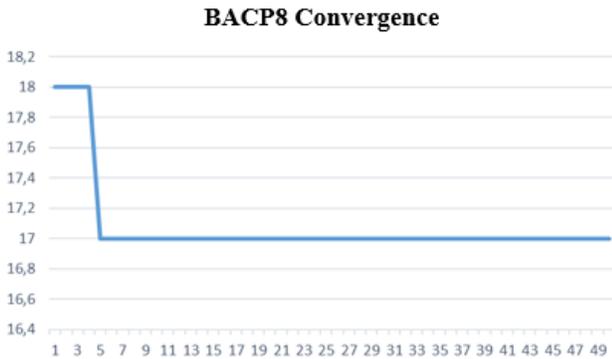


Fig. 4: BACP8 convergence.

- BACP10: This instance is composed of 42 subjects assigned in 10 academic periods, so the maximum academic load is lower than in the case of BACP8. In this test instance, the system had greater difficulty in finding the known optimum. Fig. 5 displays the obtained results.
- BACP12: The instance of 12 periods is the most complex and has 66 subjects to assign, so the computational effort required to find good solutions is greater than in the previous cases. For this instance of the problem, the system managed to find the optimal

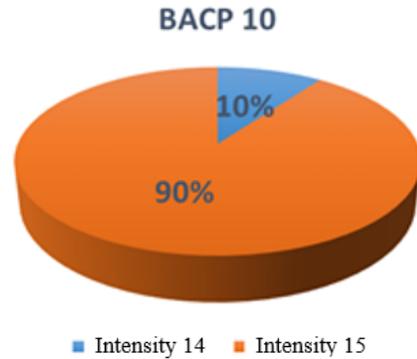


Fig. 5: BACP10 results.

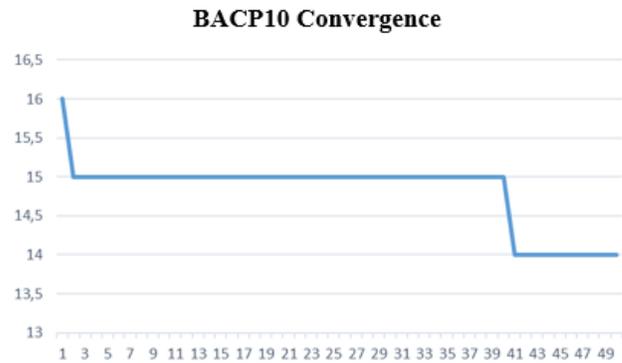


Fig. 6: BACP10 convergence.

solution known only in 23% of the executions. Fig. 7 shows the distribution of the obtained results.

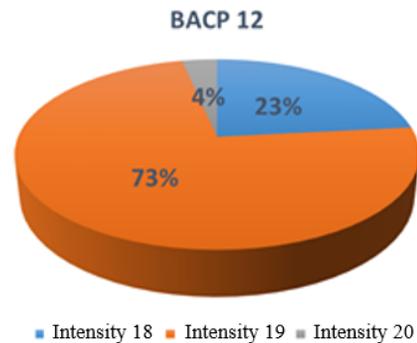


Fig. 7: BACP12 results.

Fig. 8 shows how the algorithm converges during execution for BACP12 instance, finding a solution of 20 credits in the initial iterations, then 19 and finally converging to 18 credits in iteration 13.

B. Real Cases

This section shows the results obtained by applying the system developed in the optimization of different real curricula of 2 Chilean universities. The instances have been named

TABLE II: RESULTS SYNTHESIS

Solution	BACP8	BACP10	BACP12	REAL8	REAL10	REAL12
Best	17	14	18	17	13	17
Medium	17.5	14.9	18.8	18.1	13.9	19
Worst	18	15	20	19	15	20
σ	.509	.305	.484	.681	.860	.831
Optimum	17	14	18	18	13	18

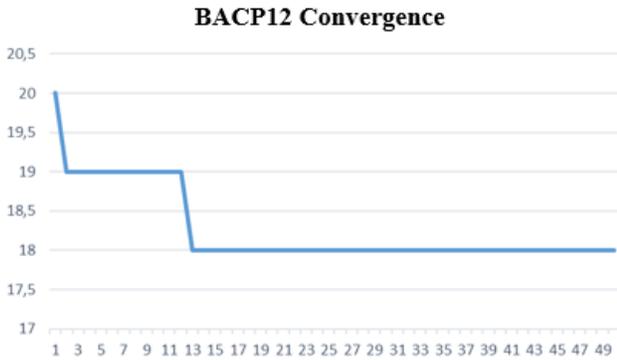


Fig. 8: BACP12 convergence.

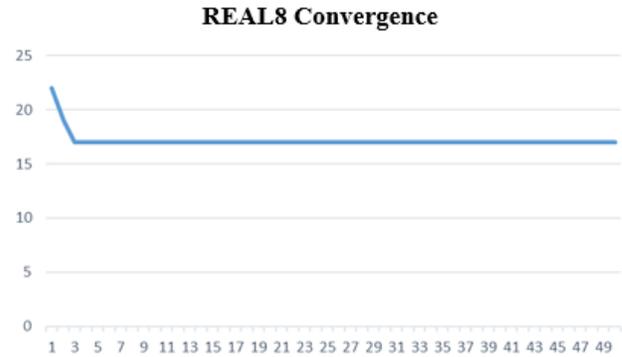


Fig. 10: REAL8 convergence.

REAL8 (8 periods), REAL10 (10 periods) and REAL12 (12 periods).

- REAL8: This instance has 34 subjects planned in 8 periods. Besides, 27 of these subjects are compulsory, 2 subjects are for general study, and 5 subjects are elective. For this case, it was possible to find an optimal value of 17 credits, while in the current curriculum the value is 18 credits. Fig. 9 displays these results.

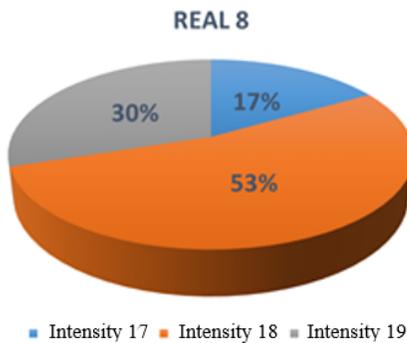


Fig. 9: REAL8 results.

The convergence graph of Fig. 10 shows that the system converges from 22 credits to 17 credits quickly.

- REAL10: The second solved real instance corresponds to a curriculum that lasts 5 years that plans 49 subjects in 10 academic periods. This instance obtained an optimal value of 13 which coincides with the current value of the study plan. Fig. 11 presents the obtained results. Fig. 12 displays the behavior of the algorithm which is similar to the previous solved instances and showing

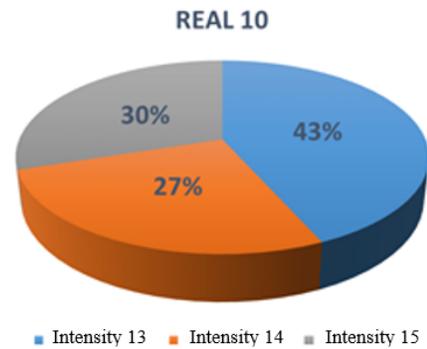


Fig. 11: REAL10 results.

a fast convergence towards the known optimum for the problem.



Fig. 12: REAL10 convergence.

- REAL12: The last real instance of the problem is the one that presents the greatest complexity and is composed of 53 subjects, of which 49 are compulsory

and 4 elective, all of them in 12 periods. For this instance, in 3% of the executions, an optimum value of 17 credits was obtained which improves the optimum of the current plan whose value is 18. Fig. 13 displays the distribution of the obtained results.

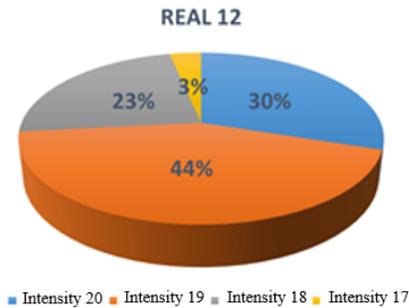


Fig. 13: REAL12 results.

Fig. 14 illustrates how the system converges to the optimum in iteration 15. Once again, the rapid convergence of the proposed algorithm is demonstrated, both when solving the test instances and the real instances of the problem.



Fig. 14: REAL12 convergence.

In general terms, the obtained results by the system are considered encouraging. Table 2 summarizes these results.

Table 2 shows that in all of the resolved instances it is possible to obtain the optimum for the problem and even to improve that optimum in some cases (instances REAL8 and REAL12). There is also a high dispersion expressed in the value of the standard deviation of the solutions found for the REAL10 and REAL12 instances. That can occur because those instances present a higher resolution complexity.

VI. RELATED WORK: OPTIMIZATION MODELS BASED ON THE BEHAVIOR OF FIREFLIES

The FA algorithm has many variants already applied in almost all areas of science [11]. Here we briefly explain a few of them:

- Discrete FA (DFA): This algorithm can be applied directly to solve discrete optimization problems [12] [13] [14] [15] [9].

- Multiobjective FA: This method has the purpose of solving problems with multiple objectives, that is, it operates with several objective functions by combining all the objective functions into one. Thus, the algorithm does not change in a big way [16].
- Lagrangian FA: This algorithm is proposed to solve the problem of optimization of commitment of units in unregulated power systems [17].
- Chaotic FA: This algorithm seeks to solve problems related to systems that behave unpredictably. This property does not mean that they are systems with a complete absence of order, but systems that have perfect order which include random factors [18].
- Hybrid Algorithms: This is a combination of the firefly algorithm and the ant algorithm [19].
- Parallel FA with predation (pFAP): This is an implementation for shared memory environments with an aggregate predation mechanism that helps the method escape from the local optimum [20].
- Modified FA: This is a mechanism used by different authors that seek to optimize the performance of the FA algorithm by adding methods or variants to the algorithm, where you can even change the update process to maintain the best result through the iterations [21].

VII. CONCLUSIONS

This work proposed an algorithm based on the behavior of natural fireflies to solve the Balanced Academic Curriculum Problem (BACP) problem. The experimental evaluation shows the effectiveness of artificial fireflies to solve this type of problems.

All the tests carried out show that it is possible to find good solutions in most of the executions of the proposed algorithm. In the test instances, the quality of the solutions is satisfactory, and the best-known value of each instance is obtained in most cases. In real instances of the problem, the best-known value is obtained and it is even possible to improve such value in 2 of the 3 instances in this investigation.

It will be interesting to investigate in the future how this algorithm behaves when faced with other real instances of the problem and to evaluate other variants of the algorithm.

It is also of interest for the authors to evaluate the behavior of the proposed system for instances of the generalized version of the problem, known as GBACP. These instances have greater complexity than the original problem and have been proposed initially in [22].

ACKNOWLEDGMENT

Franklin Johnson Parejas is supported by Grant "Dirección General de Investigación de la Universidad de Playa Ancha, Concurso Regular 2017, Clave ING 04-181".

REFERENCES

- [1] I. P. Gent and T. Walsh, "Csplib: A benchmark library for constraints," in *Principles and Practice of Constraint Programming – CP'99*, J. Jaffar, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 480–481.
- [2] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.
- [3] C. Castro and S. Manzano, "Variable and value ordering when solving balanced academic curriculum problems," *CoRR*, vol. cs.PL/0110007, 2001.
- [4] B. Hnich and Z. Kiziltan, "Modelling a balanced academic curriculum problem," in *Proceedings of CP-AI-OR-2002*, 2002, pp. 121–131.
- [5] B. Crawford, R. Soto, M. Olivares-Suárez, and F. Paredes, "A binary firefly algorithm for the set covering problem," in *Modern Trends and Techniques in Computer Science*, R. Silhavy, R. Senkerik, Z. K. Oplatkova, P. Silhavy, and Z. Prokopova, Eds. Cham: Springer International Publishing, 2014, pp. 65–73.
- [6] S. L. Tilahun and J. M. T. Ngnotchouye, "Firefly algorithm for discrete optimization problems: A survey," *KSCE Journal of Civil Engineering*, vol. 21, no. 2, pp. 535–545, Feb 2017. [Online]. Available: <https://doi.org/10.1007/s12205-017-1501-1>
- [7] J. M. Rubio, R. Soto, H. Jorquera, J. Aguilera, and C. Vidal, "Solving the balanced academic curriculum problem using firefly algorithm," *Ingeniare, Revista Chilena de Ingeniera*, vol. 26, no. special 2018, pp. 102–112, November 2018.
- [8] K. N. Krishnanand and D. Ghose, "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions," *Swarm Intelligence*, vol. 3, no. 2, pp. 87–124, Jun 2009.
- [9] K. Durkota, "Implementation of a discrete firefly algorithm for the gap problem within the sage framework," in *Bachelor Thesis, Czech Technical University*, 2011.
- [10] S. Arora and S. Singh, "Article: The firefly optimization algorithm: Convergence analysis and parameter selection," *International Journal of Computer Applications*, vol. 69, no. 3, pp. 48–52, May 2013, full text available.
- [11] N. J. Cheung, X.-M. Ding, and H.-B. Shen, "Adaptive firefly algorithm: Parameter analysis and its application," *PLOS ONE*, vol. 9, no. 11, pp. 1–12, 11 2014. [Online]. Available: <https://doi.org/10.1371/journal.pone.0112634>
- [12] M. K. Sayadi, R. Ramezani, and N. Ghaffarinasab, "A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems," in *International Journal of Industrial Engineering Computations*, vol. 1, no. 1, July 2010.
- [13] M. K. Marichelvam and M. Geetha, "A hybrid discrete firefly algorithm to solve flow shop scheduling problems to minimise total flow time," *Int. J. Bio-Inspired Comput.*, vol. 8, no. 5, pp. 318–325, Jan. 2016. [Online]. Available: <https://doi.org/10.1504/IJBIC.2016.079572>
- [14] F. Zhang, J. Hui, B. Zhu, and Y. Guo, "An improved firefly algorithm for collaborative manufacturing chain optimization problem," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 0, no. 0, p. 0954405418789981, 0. [Online]. Available: <https://doi.org/10.1177/0954405418789981>
- [15] T. Hassanzadeh, H. Vojodi, and A. Eftekhari-Moghadam, "An image segmentation approach based on maximum variance intra-cluster method and firefly algorithm," in *Seventh International Conference on Natural Computation, ICNC 2011, Shanghai, China, 26-28 July, 2011*, 2011, pp. 1817–1821. [Online]. Available: <https://doi.org/10.1109/ICNC.2011.6022379>
- [16] T. Apostolopoulos and A. Vlachos, "Application of the firefly algorithm for solving the economic emissions load dispatch problem," in *International Journal of Combinatorics*, vol. 2011, no. 523806.
- [17] B. Rampriya, K. Mahadevan, and S. S. Kannan, "Unit commitment in deregulated power system using lagrangian firefly algorithm," *2010 International Conference on Communication Control and Computing Technologies*, pp. 389–393, 2010.
- [18] L. dos Santos Coelho, D. L. de Andrade Bernert, and V. C. Mariani, "A chaotic firefly algorithm applied to reliability-redundancy optimization," *2011 IEEE Congress of Evolutionary Computation (CEC)*, pp. 517–521, 2011.
- [19] G. Giannakouris, V. Vassiliadis, and G. Dounias, "Experimental study on a hybrid nature-inspired algorithm for financial portfolio optimization," in *Artificial Intelligence: Theories, Models and Applications*, S. Konstantopoulos, S. Perantonis, V. Karkaletsis, C. D. Spyropoulos, and G. Vouros, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 101–111.
- [20] E. F. P. Luz, H. F. C. Velho, and J. C. Becceneri, "Firefly algorithm with predation: A parallel implementation applied to inverse heat conduction problem," *Proc. of 10th World Congress on Computational Mechanics (WCCM 2012)*, 2012.
- [21] S. L. Tilahun and H. C. Ong, "Modified firefly algorithm," *Journal of Applied Mathematics*, no. 467631.
- [22] 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, Feb 2012. [Online]. Available: <https://doi.org/10.1007/s10732-011-9158-2>