A Teaching-Learning-Based Optimization Algorithm for the Weighted Set-Covering Problem

Broderick CRAWFORD, Ricardo SOTO, Wenceslao PALMA, Felipe ABALLAY, Gino ASTORGA*, José LEMUS-ROMANI, Sanjay MISRA, Carlos CASTRO, Fernando PAREDES, José-Miguel RUBIO

Abstract: The need to make good use of resources has allowed metaheuristics to become a tool to achieve this goal. There are a number of complex problems to solve, among which is the Set-Covering Problem, which is a representation of a type of combinatorial optimization problem, which has been applied to several real industrial problems. We use a binary version of the optimization algorithm based on teaching and learning to solve the problem, incorporating various binarization schemes, in order to solve the binary problem. In this paper, several binarization techniques are implemented in the teaching/learning based optimization algorithm, which presents only the minimum parameters to be configured such as the population and number of iterations to be evaluated. The performance of metaheuristic was evaluated through 65 benchmark instances. The results obtained are promising compared to those found in the literature.

Keywords: combinatorial optimization; metaheuristics; set-covering problem (SCP); teaching-learning-based optimization algorithm (TLBO)

1 INTRODUCTION

The Set-Covering Problem (SCP) is a combinatorial problem included in the list by Karp of 21 NP-complete problems [1]. The SCP has many applications, such as cutting stock problems, balancing production and airline crew scheduling [2, 3].

We propose to use an algorithm to solve the SCP that could find a global optimum or a closer value in a shorter amount of time. The proposed algorithm is a metaheuristic optimization algorithm known as the Teaching-Learning-Based Optimization (TLBO) algorithm, which is characterized by no additional parameter requirements.

The implementation of metaheuristic techniques to solve highly complex problems is an open research field in constant evolution. Several metaheuristics have been used to solve complex problems, such as: Classification Subsystem in a Series Hybrid Electric Vehicle Using Particle Swarm Optimization modified by Fuzzy [4], The problem of Optimal Power Flow (OPF) Using Reactive Power Optimization [5], Optimization of bridges Reinforcement with Tied-Arch Using Moth Search Algorithm [6].

The TLBO is used for real-world problems; an example of this is multi-level production planning in the petrochemical industry [7] or the combinatorial optimization problem in the foundry industry [8], in the problem of Metal Inert Gas (MIG) welding process optimization [9], Optimization of Surface Roughness in Plasma Arc Cutting [10], in Methods for GPS Receiver Position Estimation [11] and solar photovoltaic parameter estimation [12] among others.

In order to solve a binary problem such as SCP, we have incorporated a binary process in the TLBO metaheuristics, which consists of transferring the continuous values of the MH to binary values in order to solve the SCP. This is achieved by using a transfer function in conjunction with a binary function [13].

To evaluate the performance of the binary version of TLBO (bTLBO), we use the OR-library benchmark [14], where we evaluated 65 instances, using the different binary schemes composed by 8 transfer functions and 5 binary functions, giving a total of 40 evaluated combinations. The best results are compared with results from the literature through the Wilcoxon statistical test.

Although it is true that the optimum results were not obtained in all or most instances, the performance of bTLBO metaheuristics obtains promising results, statistically competitive with results from the literature, after not finding significance under the Wilcoxon test, which opens a research opportunity for the incorporation of other techniques such as hybridization, Autonomous Search (AS), techniques inspired by Machine Learning, among others.

This document is organized as follows, section 2 presents a description of the SCP, section 3 presents the TLBO algorithm, together with the transfer and binarization functions, while section 4 presents the results and section 5 the working conclusions.

2 SET COVERING PROBLEM

The problem is mathematically represented as follows: Let $A = (a_{ij})$, a binary matrix $(0, 1)$ of dimensions $m \times n$; $C = (c_j)$, a vector $n$ dimensional; $I = \{1, \ldots, m\}$ is a set of rows; $J = \{1, \ldots, n\}$ is a set of columns. Given $c_j > 0$ for $(c_j \in C, j \in J)$, representing the cost of selecting column $j$ of matrix $A$. If $a_{ij} \in A$ is equal to 1, we say that column $j$ covers row $i$; otherwise, row $i$ is not covered. In the SCP, the objective is to find a minimum cost subset of columns in matrix $A$ such that each row is covered by at least one column in subset $S$. We can formulate the SCP mathematically as follows:

$$
\text{minimize } Z = \sum_{j=1}^{n} c_j x_j
$$

Subject to:

$$
\sum_{j=1}^{n} a_{ij} x_j \geq 1 \forall i \in I
$$

$$
x_j \in \{0,1\} \forall j \in J
$$
Other metaheuristics that have solved the SCP are the Cuckoo Search (CS) [15], Simulated Annealing (SA) [16], Black Hole (BH) [17], Ant Colony Optimization (ACO) [18], Intelligent Water Drops (IWD) [19] Particle Swarm Optimization (PSO) [20], Artificial Bee Colony (ABC) [21] and Firefly Algorithms (FA) [22].

2.1 Pre-Processing

The SCP is considered a NP-Hard problem because there is no polynomial time solution available, and the application of the exact algorithms to the large-scale instances is very time-consuming. For this reason, we use preprocessing to reduce the size of the instances [23].

Column Domination [14]: If column \( j \) has rows \( I_j \) covered by other columns, with a total cost lower than \( c_j \), then column \( j \) is dominated and is deleted from the solution. Algorithm 1 shows how Column Domination works.

Column Inclusion [14]: If a row is covered by a single column after applying the previous point, it must be incorporated into the solution.

Algorithm 1 Column Domination
1: Order all the columns by cost, ascending
2: If two or more columns have the same cost, then
3: Order those columns by the amount of rows \( I_j \) covered by
Column \( j \), descending
4: Check if rows \( I_j \) can be covered by a set of other columns with a cost lower than \( c_j \)
5: If the cost is lower, then
6: Column \( j \) is dominated and then removed
7: end if
8: end if

3 TEACHING LEARNING BASED OPTIMIZATION ALGORITHM

TLBO is a population method based on the knowledge that a teacher in a classroom that is shared with students improves the knowledge level of the class. Moreover, the students are evaluated by the value of the qualification interaction between students.

The algorithm converges to a solution when the mean of a class increases from \( M_A \) to \( M_B \) depending upon the strength of the teacher. A good teacher tries to bring their pupils to their own knowledge level. In practice, this is not possible, and a teacher can only move the mean of a class up to a certain extent depending on the capability of the class. This method follows a random process depending on many factors.

Let \( M_t \) be the mean and \( T_t \) be the teacher at any iteration \( i \). \( T_t \) will try to move \( M_t \) towards its own level; thus, the new mean will be \( T_{new} \), designated \( M_{new} \). The solution is updated according to the difference between the existing and the new mean given in Eq (4):

\[
D_{Mean_i} = \text{rand}(X_{new} - T_p M_i)
\]

where \( T_p \) is a teaching factor that decides the value of the mean to be changed, and rand is a random value between 0 and 1. The value of \( T_p \) can be either 1 or 2, which is again a heuristic step and is randomly decided with equal probability using the following Eq. (5) [24]:

\[
T_p = \text{round}\left(1 + \text{rand}[0,1]\right)
\]

This difference modifies the existing solution by means of Eq. (6) [25]:

\[
X_{new} = (X_{old} + D_{Mean_i})
\]

3.2 Learner Phase

The students increase their knowledge by interactions among themselves. A solution is randomly interacted to learn something new with other solutions in the population. A solution will learn new information if the other solutions have more knowledge than him or her. The learning phenomenon of this phase is expressed by Eq. (7):

\[
x^d_{new} = x^d_i + \text{rand} \left( x^d_j - x^d_i \right), \text{ if } f(x_i) > f(x_j)
\]

\[
x^d_{new} = x^d_i + \text{rand} \left( x^d_j - x^d_i \right), \text{ if } f(x_i) \leq f(x_j)
\]

At any solution \( x \), considering two different learners \( x_i \) and \( x_j \), where \( i \neq j \). Consequently, we accept \( x_{new} \) if it gives a better function value. After a number of sequential teaching learning cycles, the teacher passes on knowledge to the learners, and those levels increase towards his or her own level; the randomness distribution within the search space decreases, close to the point of being considered a teacher. The algorithm converges to a solution when the knowledge level of a class shows smoothness. The term criterion can be the number of evaluations or can reach a maximum number of iterations as previously established.

Algorithm 2 TLBO
1: \( i = 1 \)
2: Objective Function \( f(x) \), \( x_i = (x^1, x^2, \ldots, x^d)^T \) \( d = n^o \). of design variables or dimension
3: Generate initial classroom students at random \( (x_i) \)
4: For all the students, \( f(x_i) \) is calculated
5: while is not the stop condition, do
6: \#teacher
7: Compute the mean of each design variable \( x_{mean} \)
8: Find the best solution
3.4 bTLBO

To calculate the velocity of each student (in both phases), the following Eq. (9) and Eq. (10) are used:

\[ V_i^d(t+1) = \text{rand}(X_i^d, T_F X_i^d) \]  
\[ V_i^d(t+1) = \text{rand}(X_i^d, -X_i^d) \]  

A binary vector represents the position as well as a floating point vector of the velocity, using the velocity for the change in the probability from 0 to 1 when the position is updated. In the binary vector, the version is replaced with Eq. (6) by Eq. (9) for the teacher phase, and Eq. (7) is replaced by Eq. (10) for the learner phase. In both equations, random represent a random number between 0 and 1.

3.5 Binarization Schemes

The problems are within the search space in the real variables and can continuously be converted into binary problems. A binary space search has similar structure limitations and can be considered similar to a hypercube. The agents of the binary optimization algorithm can only pass from a near nook to the furthest area in the hypercube by changing several bits.

| Transfer functions used | $S_i$ | $T_i(x_i)$ | $V_i$ | $T_i(x_i)$ |
|------------------------|------|------------|------|------------|
| $S_1$ | $V_i^{\alpha}$ | $T_i(x_i) = 1 + \frac{1}{1 + e^{-\beta x_i}}$ | $V_i$ | $T_i(x_i) = \arctan \left( \frac{\pi}{2} \right)$ |
| $S_2$ | $V_i^{\beta}$ | $T_i(x_i) = 1 + \frac{1}{1 + e^{-\alpha x_i}}$ | $V_i$ | $T_i(x_i) = \arctan \left( \frac{\pi}{2} \right)$ |
| $S_3$ | $V_i^{\gamma}$ | $T_i(x_i) = \sqrt{\frac{1}{1 + e^{\alpha x_i}}}$ | $V_i$ | $T_i(x_i) = \arctan \left( \frac{\pi}{2} \right)$ |
| $S_4$ | $V_i^{\delta}$ | $T_i(x_i) = \sqrt{\frac{1}{1 + e^{-\alpha x_i}}}$ | $V_i$ | $T_i(x_i) = \arctan \left( \frac{\pi}{2} \right)$ |

### Algorithm 3 Repair Operator

1. Begin

\[
	ext{Cost of a column} = \frac{\text{number of uncovered rows which it covers}}{(8)}
\]

When a column is deleted and the solution remains feasible, it is said that this column is redundant. For that reason, it is necessary to remove the column using Algorithm 3, where $S$ is the set of columns of the solution, $U$ is the set of rows not covered and $w_i$ is the number of columns that cover the row $i$, $\forall i \in I$ in the solution.
The transfer functions (TF) represent a probability to change an element of the solution from 1 to 0, or vice versa. These functions lead the motion in a binary space. We solve these problems with 8TF, also used by Mirjalili in [26], by separating them into the following 2 groups: the function type S-shape and the function type V-Shape. (Tab. 1, Fig. 1).

Therefore, the TF and five discretization methods were used, as described below.

$$X_i^d (t+1) = \begin{cases} 1 & \text{if } \text{rand} \leq T(V_i^d (t+1)) \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (11)

**3.5.2 Elitist**

In this Eq. (12), the value of the best learner is allocated if the aleatory value is contained in the probability of the aforementioned variable; if not, 0 is allocated.

$$X_i^d (t+1) = \begin{cases} x_{best}^d & \text{if } \text{rand} \leq T(V_i^d (t+1)) \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (12)

**3.5.3 Complement**

If the random value is in the range of the probability that is allocated, the complement of the value is as follows Eq. (13):

$$X_i^d (t+1) = \begin{cases} \text{complement} (X_i^d) & \text{if } \text{rand} \leq T(V_i^d (t+1)) \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (13)

**3.5.4 Static Probability**

In this Eq. (14), the variable is assigned if the probability is less than $\alpha$, where $\alpha$ is a parameter that should be considered. The variable is assigned to the best case if the probability is in the range between $\alpha$ and $1 = 2(1 + \alpha)$; in any other case, the variable remains.

$$X_i^d (t+1) = \begin{cases} x_{best}^d & \text{if } T(V_i^d (t+1)) \leq \alpha \\ x_i^d & \text{if } \alpha \leq T(V_i^d (t+1)) \leq \frac{1}{2}(1 + \alpha) \\ \frac{1}{2}(1 + \alpha) & \text{if } \frac{1}{2}(1 + \alpha) \leq T(V_i^d (t+1)) \end{cases}$$  \hspace{1cm} (14)

**3.5.5 Elitist Roulette**

This method is similar to the Elitist Roulette MonteCarlo Selection. Each population individual is allocated a proportional part to the roulette setting so that the sum of all the parts becomes the unit. The best individuals will receive a higher roulette portion than the worst individuals. Generally, the population is arranged based on the largest portion size at the beginning of the process. An individual is chosen by generating a random roulette number, and the individual located in that position is returned as follow Eq. (15):

$$P_i = \frac{f_i}{\sum_{j=1}^{k} f_j}$$  \hspace{1cm} (15)

where $f_i$ is the fitness of a candidate solution represented by a learner, and $k$ is the total number of learners.

**4 EXPERIMENTAL RESULTS**

The 65 instances from Beasley's OR-Library are used in the experiments in this study [27]. For each instance, the algorithm was executed 30 times to use the average of the results obtained. Each instance was pre-processed by using the methods defined in Section 2 with 20 students and 100 iterations.

To implement the algorithm, the Java Programming Language and a computer with a Windows 10 operating system, 16 GB of RAM, and an i7-6700 processor are used.

We compared our binary learning and teaching based optimization algorithm (bTLBO) with the global optimal (Zopt) in [27], as well as with recent work solving SCP based on bio-inspired metaheuristics. The metaheuristics are: Binary Cat Swarm Optimization (BCSO) [28], Binary Firefly Optimization (BFO) [22], Binary Shuffled Frog Leaping Algorithm (BSFLA) [29], Binary Electromagnetism - Like Algorithm (BELA) [30], and Binary Artificial Bee Colony (BABC) [31]. The results are summarized in Tab. 2, where the first column details the instances to be solved, the next one the Zopt value that corresponds to the optimal value of each instance, while the following columns present the best fitness obtained, while the last rows detail the average values and p-value corresponding to each column.

To perform the statistical analysis in this study, we use the following tests:

Kolmogorov-Smirnov-Lilliefors [32] is used to determine the independence of samples.

Wilcoxon's Signed Rank [33] is used to verify superiority of the strategy of resolution using bTLBO in relation to the other metaheuristics.
|Instances| Zopt| bTLBO| BCSO| BFO| BSFLA| BELA| BABC|
|---|---|---|---|---|---|---|---|
|4.1| 429| 430| 439| 429| 430| 447| 430|
|4.2| 512| 523| 570| 517| 516| 559| 513|
|4.3| 516| 526| 590| 519| 520| 537| 519|
|4.4| 494| 501| 547| 495| 501| 527| 495|
|4.5| 512| 518| 545| 514| 514| 527| 514|
|4.6| 560| 564| 637| 563| 563| 607| 561|
|4.7| 430| 431| 462| 430| 431| 448| 431|
|4.8| 492| 501| 546| 497| 497| 509| 493|
|4.9| 641| 660| 711| 655| 656| 682| 649|
|4.1| 514| 519| 537| 519| 518| 571| 517|
|5.1| 253| 258| 279| 257| 254| 280| 254|
|5.2| 302| 311| 339| 309| 307| 318| 309|
|5.3| 226| 228| 247| 229| 228| 242| 229|
|5.4| 242| 244| 251| 242| 242| 251| 242|
|5.5| 211| 212| 230| 211| 211| 225| 211|
|5.6| 213| 214| 232| 213| 213| 247| 214|
|5.7| 293| 293| 332| 298| 297| 316| 298|
|5.8| 288| 293| 320| 291| 291| 315| 289|
|5.9| 279| 281| 295| 284| 281| 314| 280|
|5.1| 265| 268| 285| 268| 265| 280| 267|
|6.1| 138| 140| 151| 138| 140| 152| 142|
|6.2| 146| 149| 152| 147| 147| 160| 147|
|6.3| 145| 148| 160| 147| 147| 160| 148|
|6.4| 131| 131| 138| 131| 131| 140| 131|
|6.5| 161| 167| 169| 164| 166| 184| 165|
|A.1| 253| 257| 286| 255| 255| 261| 254|
|A.2| 252| 260| 274| 259| 260| 279| 257|
|A.3| 232| 239| 257| 238| 237| 252| 235|
|A.4| 234| 238| 248| 235| 235| 250| 236|
|A.5| 236| 238| 244| 236| 241| 236| 236|
|B.1| 69| 71| 79| 71| 70| 86| 70|
|B.2| 76| 80| 86| 78| 76| 88| 78|
|B.3| 80| 80| 85| 80| 80| 85| 80|
|B.4| 79| 83| 89| 80| 79| 84| 80|
|B.5| 72| 72| 73| 72| 72| 78| 72|
|C.1| 227| 234| 242| 230| 229| 237| 231|
|C.2| 219| 223| 240| 223| 223| 237| 222|
|C.3| 243| 260| 277| 253| 253| 271| 254|
|C.4| 219| 228| 250| 225| 227| 246| 231|
|C.5| 215| 220| 243| 217| 217| 224| 216|
|D.1| 60| 62| 65| 60| 60| 62| 60|
|D.2| 66| 69| 70| 68| 67| 71| 68|
|D.3| 72| 77| 79| 75| 75| 79| 76|
|D.4| 62| 64| 64| 62| 63| 67| 63|
|D.5| 61| 63| 65| 63| 63| 66| 63|
|E.1| 29| 30| 29| 29| 29| 30| 29|
|E.2| 30| 33| 34| 32| 31| 35| 32|
|E.3| 27| 29| 31| 29| 28| 34| 29|
|E.4| 28| 32| 32| 29| 29| 33| 29|
|E.5| 29| 29| 30| 29| 28| 30| 29|
|F.1| 14| 16| 17| 15| 15| 17| 14|
|F.2| 15| 17| 18| 16| 15| 18| 16|
|F.3| 14| 17| 17| 16| 16| 17| 16|
|F.4| 14| 16| 17| 15| 15| 17| 15|
|F.5| 13| 15| 15| 15| 15| 16| 15|
|G.1| 176| 193| 190| 195| 182| 194| 183|
|G.2| 154| 164| 165| 161| 161| 176| 162|
|G.3| 166| 178| 187| 175| 173| 184| 174|
|G.4| 168| 181| 179| 176| 173| 196| 175|
|G.5| 168| 185| 181| 177| 174| 198| 179|
|H.1| 63| 72| 70| 69| 68| 70| 60|
|H.2| 63| 67| 67| 66| 66| 71| 69|
|H.3| 59| 68| 68| 65| 62| 68| 66|
|H.4| 58| 66| 66| 63| 63| 70| 64|
|H.5| 55| 60| 61| 59| 59| 69| 60|
|Avg.| 196.34| 201.49| 214.98| 199.51| 199.15| 212.42| 199.32|
|p-value| 0.29| 0.61| 0.63| 0.27| 0.60|
For both tests, we use a significance level of 0.05, that is, values smaller than 0.05 indicate that the corresponding hypothesis cannot be assumed. For the first test, the following hypothesis is used:

\[ H_0 = \text{Data follow a normal distribution.} \]
\[ H_1 = \text{Data do not follow a normal distribution.} \]

Given the \( p \)-values obtained in the tests, the hypothesis is rejected.

The Wilcoxon-Mann-Whitney [33] test is then applied. To verify superiority of the resolution bTLBO in relation to other techniques, the following hypotheses are defined:

\[ H_0 = \text{bTLBO} \geq \text{Other MH.} \]
\[ H_1 = \text{Other MH} < \text{bTLBO.} \]

The statistical program R [34] is used to obtain the \( p \)-value, if this value is less than 0.05, we can affirm that the hypothesis \( H_0 \) is rejected and \( H_1 \) is accepted, which implies that bTLBO would provide better results. This procedure is repeated with all the techniques to be compared.

5 CONCLUSIONS

In this work, bTLBO was applied to resolve the SCP, and various transfer functions were used along with binary functions to resolve 65 instances of OR-library. For the obtained solutions, 4 optimums were reached in the instances 5.7, 6.4, B.3 and B.5. Other results are very close to the optimal values, but others, like NRG.1 or C.3, were not. This MH has a promising performance considering on one hand the few iterations made to obtain the presented results, together with a lack of parameters to configure, as it is the case of most MH techniques, which constitutes in itself an optimization problem. With techniques like this, where the configuration and implementation does not present major difficulties, the use of these optimization techniques is facilitated for real world problems, where they are usually implemented by users who are not experts in these techniques.

TLBO, being a global stochastic algorithm, which bases its behavior on imitating the relationship between learners and a teacher, has advantages over other techniques due to its faultless structure of understanding and implementation, along with its high speed in obtaining results.

In future works, we propose to use other discretization techniques, such as those shown in the work of Crawford et al. [13], in addition to using techniques inspired by Machine Learning such as [35, 36].

Acknowledgements

Broderick Crawford is supported by Grant CONICYT/FONDECYT/REGULAR/1171243. Ricardo Soto is supported by Grant CONICYT/FONDECYT/REGULAR/1190129. This work was funded by the CONICYTPFCHA/DOCTORADO BECAS NACIONAL/2019 - 21191692.

6 REFERENCES

[1] Garey, M. R. & Johnson, D. S. (1990). A Guide to the Theory of NP-Completeness. Computers and Intractability, 37-79.
[2] Klocker, B. & Raidl, G. R. (2017). Solving a weighted set covering problem for improving algorithms for cutting stock problems with setup costs by solution merging. International Conference on Computer Aided Systems Theory, 355-363. https://doi.org/10.1007/978-3-319-74718-7_43
[3] Mesquita, M. & Paías, A. (2008). Set partitioning/covering-based approaches for the integrated vehicle and crew scheduling problem. Computers & Operations Research, 35(5), 1562-1575. https://doi.org/10.1016/j.cor.2006.08.001
[4] Johanyák, Z. C. (2017). A modified particle swarm optimization algorithm for the optimization of a fuzzy classification subsystem in a series hybrid electric vehicle. Tehnički vjesnik, 24(2), 295-301. https://doi.org/10.17559/TV-20151021202802
[5] Üney, M. Ş. & Cetinkaya, N. (2019). New Metaheuristic Algorithms for Reactive Power Optimization. Tehnički vjesnik, 26(5), 1427-1433. https://doi.org/10.17559/TV-2018120513116
[6] Carrasco, O., Crawford, B., Soto, R., Lemus-Romani, J., Astorga, G., & Salas-Fernández, A. (2019). Optimization of Bridges Reinforcements with Tied-Arch Using Moth Search Algorithm. De International Work-Conference on the Interplay between Natural and Artificial Computation, Cham. https://doi.org/10.1007/978-3-030-19651-6_24
[7] Kadambur, R. & Kotecha, P. (2015). Multi-level production planning in a petrochemical industry using elitist Teaching–Learning-Based-Optimization. Expert Systems with Applications, 42(1), 628-641. https://doi.org/10.1016/j.eswa.2014.08.006
[8] Ji, X., Ye, H., Zhou, J., Yin, Y., & Shen, X. (2017). An improved teaching-learning-based optimization algorithm and its application to a combinatorial optimization problem in foundry industry. Applied Soft Computing, 57, 504-516. https://doi.org/10.1016/j.asoc.2017.04.029
[9] Jogi, B. F., Awale, A., Nirantar, S., & Bhussare, H. (2018). Metal inert gas (MIG) welding process optimization using teaching-learning based optimization (TLBO) algorithm. Materials Today: Proceedings, 5(2), 7086-7095. https://doi.org/10.1016/j.matpr.2017.11.373
[10] Patel, P., Nakam, B., Abhishek, K., Kumar, V. R., & Kumar, A. (2018). Optimization of Surface Roughness in Plasma Arc Cutting of AISI52 Steel Using TLBO. Materials Today: Proceedings, 5(9), 18927-18932. https://doi.org/10.1016/j.matpr.2018.06.242
[11] Bagadi, L., Rao, G. S., & Kumar, N. A. (2018). Firefly, Teaching Learning Based Optimization and Kalman Filter Methods for GPS Receiver Position Estimation. Procedia computer science, 143, 892-898. https://doi.org/10.1016/j.procs.2018.10.365
[12] Chen, X., Xu, B., Mei, C., Ding, Y., & Li, K. (2018). Teaching-learning-based artificial bee colony for solar photovoltaic parameter estimation. Applied energy, 212, 1578-1588. https://doi.org/10.1016/j.apenergy.2017.12.115
[13] Crawford, B., Soto, R., Astorga, G., Garcia, J., Castro, C., & Paredes, F. (2017). Putting continuous metaheuristics to work in binary search spaces. Complexity. https://doi.org/10.1155/2017/840431
[14] Beasley, J. E. & Chu, P. C. (1996). A genetic algorithm for the set covering problem. European journal of operational research, 94(2), 392-404. https://doi.org/10.1016/0377-2217(95)00159-X
[15] Soto, R., Crawford, B., Olivares, R., Barraza, J., Figueroa, I., Johnson, F., Paredes, F., & Olguín, E. (2017). Solving the non-uniform set covering problem by using cuckoo search and black hole optimization. Natural Computing, 16(2), 213-229. https://doi.org/10.1007/s11047-016-9609-7
[16] Thompson, G. M. (1996). A simulated-annealing heuristic for shift scheduling using non-continuously available employees. Computers & Operations Research, 23(3), 275-288. https://doi.org/10.1016/0305-0483(95)00127-7
[17] García, J., Crawford, B., Soto, R., & Garcia, P. (2017). A multi dynamic binary black hole algorithm applied to set
covering problem. *International Conference on Harmony Search Algorithm*, 42-51. https://doi.org/10.1007/978-981-10-3728-3_6

[18] Cotta, C., Sevaux, M., & Sörensen, K. (2008). Adaptive and multilevel heuristics. *Springer*. https://doi.org/10.1007/978-3-540-79438-7

[19] Crawford, B., Soto, R., Astorga, G., Lemus-Romani, J., Misra, S., & Rubio, J.-M. (2019). An Adaptive Intelligent Water Drops Algorithm for Set Covering Problem. *De 2019 19th International Conference on Computational Science and Its Applications (ICCSA)*, Saint Petersburg. https://doi.org/10.1007/978-3-030-20919-0_30

[20] Crawford, B., Soto, R., Monfroy, E., Palma, W., Castro, C., & Paredes, F. (2013). Parameter tuning of a choice-function based hyperheuristic using particle swarm optimization. *Expert Systems with Applications*, 40(5), 1690-1695. https://doi.org/10.1016/j.eswa.2012.09.013

[21] Zhang, Y., Wu, L., Wang, S., & Huo, Y. (2011). Chaotic artificial bee colony used for cluster analysis. *International Conference on Intelligent Computing and Information Science*, 205-211. https://doi.org/10.1007/978-3-642-18129-0_33

[22] Crawford, B., Soto, R., Suárez, M. O., Paredes, F., & Johnson, F. (2014). Binary firefly algorithm for the set covering problem. *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-5. https://doi.org/10.1109/CISTI.2014.6877090

[23] Fisher, M. L. & Kedia, P. (1990). Optimal solution of set covering/partitioning problems using dual heuristics. *Management Science*, 36(6), 674-688. https://doi.org/10.1287/mnsc.36.6.674

[24] Rao, R. V. & Patel, V. (2013). Multi-objective optimization of two stage thermoelectric cooler using a modified teaching-learning-based optimization algorithm. *Engineering Applications of Artificial Intelligence*, 26(1), 430-445. https://doi.org/10.1016/j.engappai.2012.02.016

[25] Rao, R. V. & Savi, V. J. (2012). Mechanical design optimization using advanced optimization techniques. https://doi.org/10.1007/978-1-4471-2748-2

[26] Mirjalili, S. & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, 9, 1-14. https://doi.org/10.1016/j.swevo.2012.09.002

[27] Beasley, J. E. (1990). A Lagrangian heuristic for set-covering problems. *Naval Research Logistics (NRL)*, 37(1), 151-164. https://doi.org/10.1002/1520-6750(199002)37:1<151::AID-NNAV3220071103>3.0.CO;2-2

[28] Crawford, B., Soto, R., Berrios, N., Johnson, F., Paredes, F., Castro, C., & Norero, E. (2015). A binary cat swarm optimization algorithm for the non-unicost set covering problem. *Mathematical Problems in Engineering*. https://doi.org/10.1155/2015/578541

[29] Crawford, B., Soto, R., Peña, C., Palma, W., Johnson, F., & Paredes, F. (2015). Solving the set covering problem with a shuffled frog leaping algorithm. *Asian Conference on Intelligent Information and Database Systems*, 41-50. https://doi.org/10.1007/978-3-319-15705-4_5

[30] Soto, R., Crawford, B., Munoz, A., Johnson, F., & Paredes, F. (2015). Pre-processing, repairing and transfer functions can help binary electromagnetism-like algorithms. *Artificial intelligence perspectives and applications*, 89-97. https://doi.org/10.1007/978-3-319-18476-0_10

[31] Cuesta, R., Crawford, B., Soto, R., & Paredes, F. (2014). An artificial bee colony algorithm for the set covering problem. *Modern Trends and Techniques in Computer Science*, 53-63. https://doi.org/10.1007/978-3-319-06740-7_5

[32] Shapiro S. S. & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4), 591-611. https://doi.org/10.1093/biomet/52.3.4.591

[33] Mann, H. B. & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 50-60. https://doi.org/10.1214/aoms/1177730491

[34] Ripley, B. D. (2001). The R project in statistical computing. *MSOR Connections. The newsletter of the LMS Maths, Stats & OR Network*, 1(1), 23-25. https://doi.org/10.11120/msor.2001.010022

[35] García, J., Crawford, B., Soto, R., Castro, C., & Paredes, F. (2018). A k-means binarization framework applied to multidimensional knapsack problem. *Applied Intelligence*, 48(2), 357-380. https://doi.org/10.1007/s10489-017-0972-6

[36] García, J., Moraga, P., Valenzuela, M., Crawford, B., Soto, R., Pinto, H., Peña, A., Altimiras, F., & Astorga, G. (2019). A Dh-Scan Binarization Algorithm Applied to Matrix Covering Problems. *Computational intelligence and neuroscience*. https://doi.org/10.1155/2019/3238574

[37] Marx, A., Backes, C., Meso, E., Lenhof, H.-P., & Keller, A. (2016). EDISON-WWM: exact dynamic programming solution of the Wilcoxon--Mann–Whitney test. *Genomics, proteomics & bioinformatics*, 14(1), 55-61. https://doi.org/10.11016/j.pgb.2015.11.004

[38] Rodriguez-Tello, E., Hao, J.-K., & Romero-Monsivais, H. (2015). Boosting the performance of metaheuristics for the MInLA problem using a more discriminating evaluation function. *Technical Gazette*, 22(1), 11-24. https://doi.org/10.17559/TV-20130905130612

Contact information:

- **Broderick CRAWFORD**, PhD, Professor
  Universidad de Valparaíso, Avenida España 1680, Valparaíso, Chile
  E-mail: broderick.crawford@uv.cl

- **Ricardo SOTO**, PhD, Professor
  Universidad Técnica Federico Santa María, Fermin Rodriguez Sur 415, Santiago, Chile
  E-mail: ricardo.soto@udp.cl

- **Felipe ABALLAY**, PhD, Professor
  Pontificia Universidad Católica de Valparaíso, Avenida Brasil 2241, Valparaíso, Chile
  E-mail: felipe.aballay@mail.pucv.cl

- **José LEMUS-ROMANI**, PhD student
  Universidad de Valparaíso, Avenida España 1680, Valparaíso, Chile
  E-mail: josemiguel.rubio@ubo.cl

- **Sanjay MISRA**, PhD, Professor
  Department of Computer and Information Science, Covenant University, Nigeria
  KM 10 Idiroko Rd, Ota, Nigeria
  E-mail: smisra@atilim.edu.ng

- **Gino ASTORGA**, PhD (Corresponding author)
  Universidad de Valparaíso, Prat 856, Valparaíso, Chile
  E-mail: gino.astorga@uv.cl

- **Fernando PAREDES**, PhD, Professor
  Escuela de Ingeniería Industrial, Universidad Diego Portales, Manuel Rodríguez Sur 415, Santiago, Chile
  E-mail: fernando.paredes@udp.cl

- **José-Miguel RUBIO**, PhD student, Professor
  Universidad Bernardo O’Higgins, Av. Vial 1497, Santiago, Chile
  E-mail: josemiguel.rubio@ubo.cl