ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **17**, Number 3, 2014, 252–264

A Binary Coded Firefly Algorithm that Solves the Set Covering Problem

Broderick CRAWFORD[a,b,c], Ricardo SOTO[a,d,e],
Miguel OLIVARES-SUÁREZ[a], Wenceslao PALMA[a],
Fernando PAREDES[f], Eduardo OLGUÍN[b], Enrique NORERO[g]

[a] Pontificia Universidad Católica de Valparaíso, Chile
[b] Universidad San Sebastián, Chile
[c] Universidad Central de Chile, Chile
[d] Universidad Autónoma de Chile, Chile
[e] Universidad Cientifica del Sur, Lima, Perú
[f] Escuela de Ingeniería Industrial, Universidad Diego Portales, Santiago, Chile
[g] Facultad de Ingeniería, Universidad Santo Tomás, Viña del Mar, Chile

E-mail: `wenceslao.palma@ucv.cl`

Abstract. This work presents a study of a new binary coded firefly algorithm. The firefly algorithm is a novel nature-inspired metaheuristic, inspired by the social behavior of fireflies, which is being applied to solve many optimization problems. We test the proposed binary coded firefly algorithm solving the non-unicost set covering problem which is a well-known NP-hard discrete optimization problem with many practical applications. To tackle the mapping from a continuous search space to a discrete search space we use different transfer functions which are investigated in terms of convergence speed and accuracy of results. The experimental results show the effectiveness of our approach where the binary coded firefly algorithm produce competitive results solving a portfolio of set covering problems from the OR-Library.

Key words: Set Covering Problem, Binary Firefly Algorithm, Metaheuristic.

1. Introduction

The Set Covering Problem (SCP) is a class of representative combinatorial optimization problem that has been applied to many real world problems, such as crew scheduling in airlines [20], facility location problem [27], and production planning in industry [28]. The SCP is a well-known NP-hard in the strong sense [18]. Many algorithms have been developed to solve it and has been reported to literature. Exact algorithms are mostly based on branch-and-bound and branch-and-cut [2, 16]. However, these algorithms are rather time consuming and can only solve instances of very limited size. For this reason, many research efforts have been focused on the development of heuristics to find good or near-optimal solutions within a reasonable period of time. Classical greedy algorithms are very simple, fast, and easy to code in practice, but they rarely produce high quality solutions for their myopic and deterministic nature [9]. Compared with classical greedy algorithms, heuristics based on Lagrangian relaxation with subgradient optimization are much more effective. The most efficient ones are those proposed in [8, 6]. As top-level general search strategies, metaheuristics such as genetic algorithms [3], simulated annealing [5], tabu search [7], evolutionary algorithms [10], ant colony optimization (ACO) [24, 13], electromagnetism (unicost SCP) [23], gravitational emulation search [1] and cultural algorithms [12] have been also successfully applied to solve the SCP.

In this paper, we propose a binary Firefly Algorithm to solve the SCP. The Firefly Algorithm (FA) is a recently developed, population-based metaheuristic [30, 29] where the objective function of a given optimization problem is based on differences of light intensity. Thus, fireflies are characterized by their light intensity which helps fireflies to change their position iteratively towards more attractive locations in order to obtain optimal solutions. The canonical FA algorithm is developed to tackle continuous optimization problems [17, 31]. However, the effectiveness of the FA algorithm to solve discrete NP-hard problems such as image compression and processing [19], shape and size optimization [21] and manufacturing cell problem [25] encourage researchers to design novel FAs for discrete optimization problems. To the best of our knowledge, this is the first work proposing a binary coded FA to solve the SCP.

The rest of this paper is organized as follows. In Section 2, we give a formal definition of the SCP. The Section 3, describes the FA and the Section 4 describes the proposed approach. In Section 5, we present experimental results obtained when applying the algorithm for solving the 65 instances of SCP contained in the OR-Library. Finally, in Section 6 we conclude and highlight future directions of research.

2. Problem Description

The Set Covering Problem (SCP) can be formally defined as follows. Let $A = (a_{ij})$ be an m-row, n-column, zero-one matrix. We say that a column j covers a row i if $a_{ij} = 1$. Each column j is associated with a nonnegative real cost c_j. Let $I = \{1, ..., m\}$ and $J = \{1, ..., n\}$ be the row set and column set, respectively. The SCP calls for a minimum cost subset $S \subseteq J$, such that each row $i \in I$ is covered by at least one column $j \in S$. A mathematical model for the SCP is

$$Minimize \quad f(x) = \sum_{j=1}^{n} c_j x_j \tag{1}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1, \quad \forall i \in I \tag{2}$$

$$x_j \in \{0, 1\}, \qquad \forall j \in J \tag{3}$$

The goal is to minimize the sum of the costs of the selected columns, where $x_j = 1$ if the column j is in the solution, 0 otherwise. The restrictions ensure that each row i is covered by at least one column.

3. Overview of Firefly Algorithm

Nature-inspired methodologies are among the most powerful algorithms for optimization problems. The Firefly Algorithm (FA) is a novel nature-inspired algorithm inspired by the social behavior of fireflies. By idealizing some of the flashing characteristics of fireflies, a firefly-inspired algorithm was presented in [30, 29]. The canonical FA was developed using the following three idealized rules:

- All fireflies are unisex and are attracted to other fireflies regardless of their sex.
- The degree of the attractiveness of a firefly is proportional to its brightness, and thus for any two flashing fireflies, the one that is less bright will move towards to the brighter one. More brightness means less distance between two fireflies. However, if any two flashing fireflies have the same brightness, then they move randomly.
- The brightness of a firefly is determined by the value of the objective function. For a maximization problem, the brightness of each firefly is proportional to the value of the objective function.

As the attractiveness of a firefly is proportional to the light intensity seen by adjacent fireflies, the attractiveness β of a firefly is defined as follows:

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \ \ m \geq 1 \tag{4}$$

where r is the distance between two fireflies, β_0 is the attractiveness at $r = 0$ and γ is a fixed light absorption coefficient. The distance r_{ij} between two fireflies i and j at positions x_i and x_j is determined by

$$r_{ij} = ||x_i - x_j|| = \sqrt{\sum_{k=1}^{d} (x_i^k - x_j^k)^2} \tag{5}$$

where x_i^k is the current value of the k_{th} dimension of the i^{th} firefly and d is the number of dimensions. The movement of a firefly i is attracted to another more attractive (brighter) firefly j is determined by

$$x_i^k(t+1) = x_i^k(t) + \beta_0 e^{-\gamma r_{ij}^2} (x_j^k(t) - x_i^k(t)) + \alpha(rand - \frac{1}{2}) \tag{6}$$

where the first term $x_i^k(t)$ is the current value (current position) of the k_{th} dimension of the firefly i at iteration t. The second term denotes the firefly attractiveness where γ characterizes the variation of the attractiveness typically varying from 0.1 to 10 [30], and the last term introduces randomization, with $\alpha \in [0, 1]$ being the randomization parameter and *rand* is a random number generator uniformly distributed between 0 and 1.

4. Description of the proposed approach

In this section, a discrete FA is proposed to solve the SCP.

Step 1. Initialize the firefly parameters (γ, β_0, size for the firefly population and the maximum number of generations for the termination process).

Step 2. Initialization of firefly position. Initialize randomly $M = [X_1; ...; X_m]$ of m solutions or firefly positions in the multi-dimensional search space, where m represents the size of the firefly population. Each solution of X is represented by a d-dimensional binary vector.

Step 3. Evaluation of fitness of the population. For this case the function of fitness is equal to the objective function of the SCP model (Eq. 1).

Step 4. Modification of firefly position. A firefly produces a modification in its position based on the brightness w.r.t other fireflies. Using Eq. 6 the new position is determined by modifying the value of each dimension of a firefly. To move from a continuous search space to a discrete one we work with the following update rules individually:

$$x_i^k(t+1) = \begin{cases} 1 & \text{if } rand < T(x_i^k(t+1)) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$x_i^k(t+1) = \begin{cases} x^{-1} & \text{if } rand < T(x_i^k(t+1)) \\ x_i^k(t) & \text{otherwise} \end{cases} \tag{8}$$

$$x_i^k(t+1) = \begin{cases} x_*^k & \text{if } rand < T(x_i^k(t+1)) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where *rand* is a uniform random number between 0 and 1, $x_i^k(t)$ is the value of the k^{th} dimension of the firefly i at iteration t, $x_i^k(t+1)$ is the value resulting from the formula 6 (movement of the firefly), x^{-1} is the complement of $x_i^k(t)$ (i.e. if the vale of the k^{th} dimension is 0 then it is set to 1 and vice versa), x_*^k is the best firefly so far, and $T(x)$ is the binary transfer function [22].

The transfer functions force the values of the dimensions of fireflies to move in a binary space. Due to the shapes of these curves, they are named s-shaped and v-shaped transfer functions [22] (Fig. 1). These transfer functions are presented in Table 1.

Step 5. The new solution is evaluated and if it is not a feasible solution then it is repaired. In order to make feasible solutions we determine which rows have not yet been covered and choose the columns needed for coverage. The criteria used to choose these columns is based in the cost of a column/number of rows not covered that cover the column j. Once the solution has become feasible we apply an optimization step in order to eliminate those redundant columns. A column is redundant when it is removed and the solution remains feasible.

Step 6. Memorize the best solution achieved so far and increment the counter of generations.

Step 7. Stop the process and display the result if the termination criteria is satisfied. Termination criteria used in this work is the maximum number of generations. Otherwise, go to step 3.

Algorithm 1 shows the pseudo code of the steps proposed. Line 21 is written to a better understanding, but in practical terms it is omitted.

Algorithm 1: Pseudo code of Binary FA for SCP

```
1  Begin
2    Initialize parameters
3    Evaluate the light intensity I determined by f(x) Eq. 1
4    while t < MaxGeneration do
5      for i = 1 : m (m fireflies) do
6        for j = 1 : m (m fireflies) do
7          if (I_j < I_i) then
8            movement = calculates value according to Table ??
9            if (rand() < T(movement)) then
10             switch rule do
11               case (7)
12                 fireflies[i][j] = 1
13               case (8)
14                 fireflies[i][j] = complement(fireflies[i][j])
15               case (9)
16                 fireflies[i][j] = bestFirefly[j]
17             end switch
18           else
19             switch rule do
20               case (8)
21                 fireflies[i][j] = fireflies[i][j]
22               otherwise
23                 fireflies[i][j] = 0
24             end switch
25           end if
26         end if
27         Repair solutions
```

```
            Update attractiveness
            Update light intensity
        end for j
    end for i
    t = t + 1
end while
Output the results
End
```

Table 1. S-shaped and v-shaped transfer functions

S-shaped family		V-shaped family	
Name	Transfer function	Name	Transfer function
S1	$T(x) = \frac{1}{1+e^{-2x}}$	V1	$T(x) = \left\lvert \mathrm{erf}\left(\frac{\sqrt{2}}{\pi}x\right)\right\rvert = \left\lvert \frac{\sqrt{2}}{\pi}\int_o^{\frac{\sqrt{2}}{\pi}x} e^{-t^2}dt \right\rvert$
S2	$T(x) = \frac{1}{1+e^{-x}}$	V2	$T(x) = \lvert\tanh(x)\rvert$
S3	$T(x) = \frac{1}{1+e^{-x/2}}$	V3	$T(x) = \left\lvert \frac{x}{\sqrt{1+x^2}} \right\rvert$
S4	$T(x) = \frac{1}{1+e^{-x/3}}$	V4	$T(x) = \lvert \frac{2}{\pi}\arctan(\frac{\pi}{2}x)\rvert$

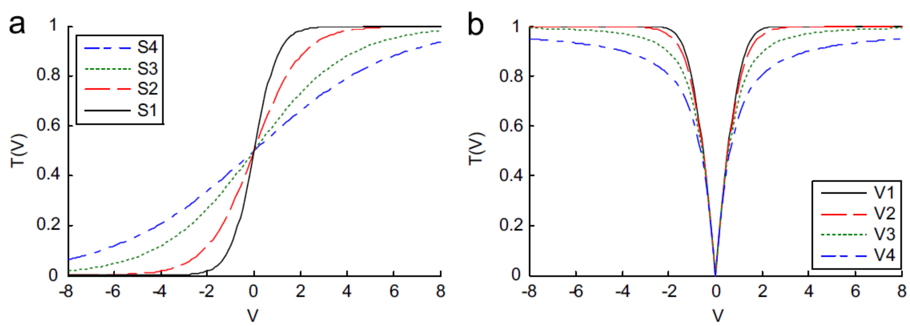

Fig. 1. (a) s-shaped and (b) v-shaped transfer functions.

5. Experimental evaluation

In order to test the effectiveness of our proposal, the binary FA was tested using the 65 SCP test instances from OR-Library [4]. These instances are divided into 11 groups and each group contains 5 or 10 instances. Table 2 shows their detailed information where "Density" is the percentage of non-zero entries in the SCP matrix. The algorithm was implemented using C language and conducted on a 1.8 GHz Intel Core 2 Duo T5670 CPU with 3GB RAM running Windows 8.

In all experiments, the binary FA is executed 30 times over each SCP instance and the maximum number of generations is set to 50. We used a population of 25 fireflies and the values of γ, β_0 are initialized to 1. These parameters were selected empirically after a large number of tests over all the SCP instances.

Table 2. Details of the test instances

Instance set	No. of instances	m	n	Cost range	Density (%)	Optimal solution
4	10	200	1000	[1, 100]	2	Known
5	10	200	2000	[1, 100]	2	Known
6	5	200	1000	[1, 100]	5	Known
A	5	300	3000	[1, 100]	2	Known
B	5	300	3000	[1, 100]	5	Known
C	5	400	4000	[1, 100]	2	Known
D	5	400	4000	[1, 100]	5	Known
NRE	5	500	5000	[1, 100]	10	Unknown
NRF	5	500	5000	[1, 100]	20	Unknown
NRG	5	1000	10000	[1, 100]	2	Unknown
NRH	5	1000	10000	[1, 100]	5	Unknown

Figure 2 shows the convergence of the instance SCP4.1 using the rule (7) with the v-shape transfer functions. Figure 3 shows the convergence of the instance SCP4.1 using the rule (8) with the v-shape transfer functions, In this figure, the convergence of V2 and V3 are not appreciated because the values of the solutions are very similar to the values of the transfer function V4.

The Figs. 2 and 3 do not consider the s-shape transfer functions because the solutions do not improve through the iterations. In these cases, the s-shape transfer functions generate repeatedly more 1s than 0s increasing the value of the objective function of the SCP model.

Figure 4 shows the convergence of the instance SCP4.1 using the rule (9) and the s-shape and v-shape transfer functions. The results show that the v-shape transfer functions have a fast convergence rate.

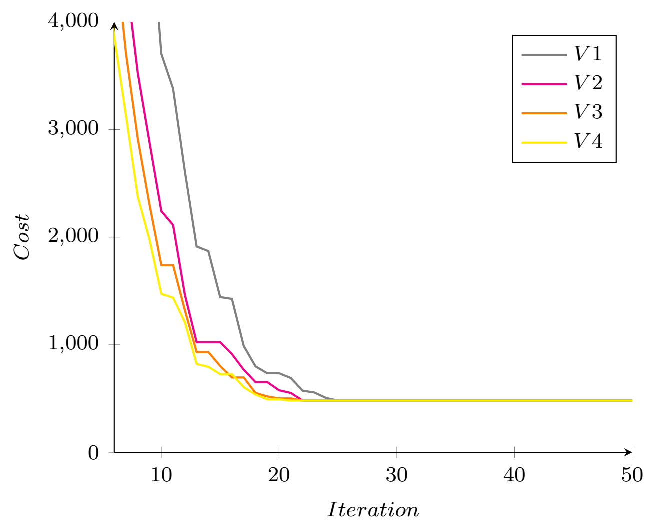

Fig. 2. Different evolutions for SCP4.1 using rule (7).

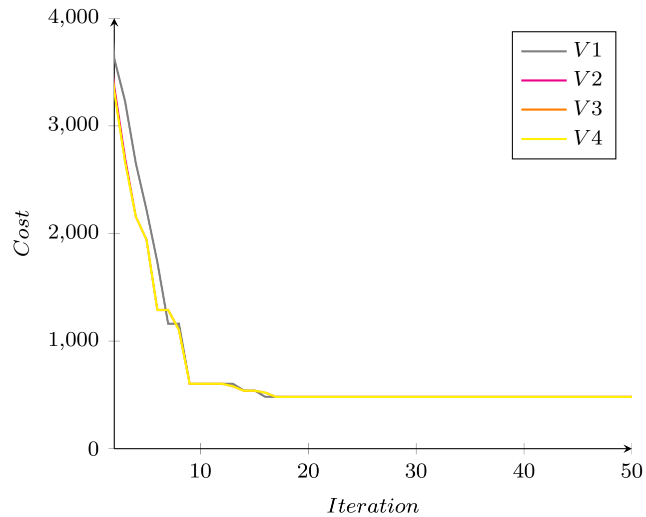

Fig. 3. Different evolutions for SCP4.1 using rule (8).

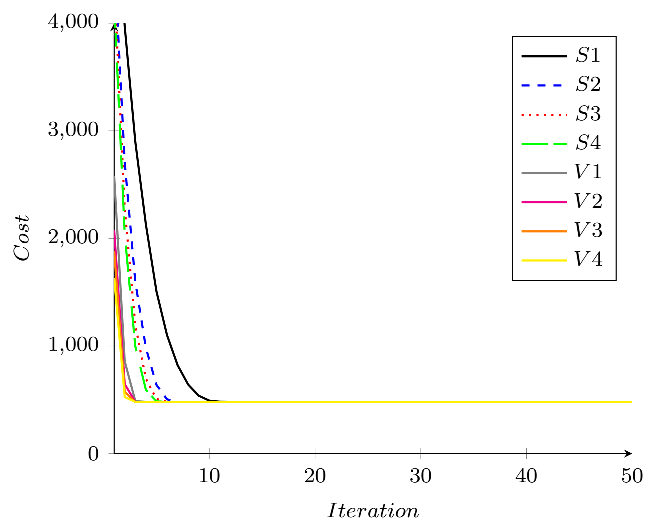

Fig. 4. Different evolutions of mean best values for SCP4.1 using rule (9) with s-shape and v-shape transfer functions.

Tables 3 and 4 show the results obtained over the 65 instances using the V4 function (the best performance transfer function) using the rule (9). The quality of a solution is evaluated using the relative percentage deviation (RPD). The RPD value quantifies the deviation of the objective value Z from Z_{opt} which in our case is the best known cost value for each instance (see the second column). We report the minimum, maximum, and average of the obtained solutions. To compute RPD we use $Z = Min.$. This measure is computed as follows:

$$RPD = (Z - Z_{opt})/Z_{opt} \times 100 \tag{10}$$

The results expressed in terms of the RPD show the effectiveness of our approach. It provides high quality near optimal solutions and it has the ability to generate them for a variety of instances.

Table 3. Experimental results over the 65 instances of SCP (using V4 and rule (9)). Part 1

Instance	Opt.	Min.	Max.	Avg.	RPD
4.1	429	481	482	481.03	12.12
4.2	512	580	580	580.00	13.28
4.3	516	619	620	619.03	19.96
4.4	494	537	537	537.00	8.7
4.5	512	609	609	609.00	18.94
4.6	560	653	653	653.00	16.6
4.7	430	491	492	491.07	14.18
4.8	492	565	565	565.00	14.83
4.9	641	749	750	749.03	16.84
4.10	514	550	550	550.00	7.0
5.1	253	296	297	296.03	16.99
5.2	302	372	372	372.00	23.17
5.3	226	250	250	250.00	10.61
5.4	242	277	278	277.07	14.46
5.5	211	253	253	253.00	19.9
5.6	213	264	265	264.03	23.94
5.7	293	337	337	337.00	15.01
5.8	288	326	326	326.00	13.19
5.9	279	350	350	350.00	25.44
5.10	265	321	321	321.00	21.13
6.1	138	173	174	173.03	25.36
6.2	146	180	181	180.07	23.28
6.3	145	160	160	160.00	10.34
6.4	131	161	161	161.00	22.9
6.5	161	186	186	186.00	15.52
A.1	253	285	285	285.00	12.64
A.2	252	285	286	285.07	13.09
A.3	232	272	272	272.00	17.24
A.4	234	297	297	297.00	26.92
A.5	236	262	262	262.00	11.01
B.1	69	80	81	80.03	15.94
B.2	76	92	92	92.00	21.05
B.3	80	93	93	93.00	16.25

Table 4. Experimental results on 65 instances of SCP (using V4 and rule (9)). Part 2

Instance	Opt.	Min.	Max.	Avg.	RPD
B.4	79	98	99	98.03	24.05
B.5	72	87	87	87.00	20.83
C.1	227	279	279	279.00	22.90
C.2	219	272	272	272.00	24.20
C.3	243	288	288	288.00	18.51
C.4	219	262	262	262.00	19.63
C.5	215	262	263	262.07	21.86
D.1	60	71	71	71.00	18.33
D.2	66	75	75	75.00	13.63
D.3	72	88	88	88.00	22.22
D.4	62	71	71	71.00	14.51
D.5	61	71	71	71.00	16.39
NRE.1	29	32	33	32.03	10.34
NRE.2	30	36	36	36.00	20
NRE.3	27	35	35	35.00	29.62
NRE.4	28	34	34	34.00	21.42
NRE.5	28	34	34	34.00	21.42
NRF.1	14	17	18	17.03	21.42
NRF.2	15	17	17	17.00	13.33
NRF.3	14	21	21	21.00	50
NRF.4	14	19	19	19.00	35.71
NRF.5	13	16	16	16.00	23.07
NRG.1	176	230	231	230.03	30.68
NRG.2	154	191	191	191.00	24.02
NRG.3	166	198	198	198.00	19.27
NRG.4	168	214	214	214.00	27.38
NRG.5	168	223	223	223.00	32.73
NRH.1	63	85	86	85.07	34.92
NRH.2	63	81	82	81.03	28.57
NRH.3	59	76	76	76.00	28.81
NRH.4	58	75	75	75.00	29.31
NRH.5	55	68	68	68.00	23.63

6. Conclusion

In this paper, a binary FA has been proposed to solve the SCP. The effectiveness of the proposed approach was tested on benchmark problems and the obtained results show that the binary FA is a good alternative to solve the SCP.

The results show that the v-shape transfer functions with the rules (7), (8) and (9) of updating position vectors converge faster than the s-shape transfer functions which generate repeatedly more 1s than 0s increasing the value of the objective function of the SCP model.

An interesting research direction of future work is the integration of autonomous search (AS) in the solving process, which in many cases has demonstrated excellent results [14, 26, 15, 11]. AS provides a framework to design systems that are able to autonomously self-tune their performance while effectively solving problems. Thus, problem solvers can now perform self-improvement operations based on analysis of the performances of the solving process.

Acknowledgements. The author Broderick Crawford is supported by grant CONICYT/FONDE-CYT/REGULAR/1140897, Ricardo Soto is supported by grant CONICYT /FONDECYT/INICIACION/11130459 and Fernando Paredes is supported by grant CONICYT/FONDECYT/REGULAR/1130455.

References

[1] BALACHANDAR S. R., KANNAN K., *A meta-heuristic algorithm for set covering problem based on gravity*, International Journal of Computational & Mathematical Sciences, **4** (7), pp. 937–942, 2010.

[2] BALAS E., CARRERA M. C., *A dynamic subgradient-based branch-and-bound procedure for set covering*, Operations Research, **44** (6), pp. 875–890, 1996.

[3] BEASLEY J., CHU P., *A genetic algorithm for the set covering problem*, European Journal of Operational Research, **94** (2), p. 392–404, 1996.

[4] BEASLEY J. E., *A lagrangian heuristic for set-covering problems*, Naval Research Logistics (NRL), **37** (1), pp. 151–164, 1990.

[5] BRUSCO M., JACOBS L., THOMPSON G., *A morphing procedure to supplement a simulated annealing heuristic for cost- and coverage-correlated set-covering problems*, Annals of Operations Research, **86** (0), pp. 611–627, 1999.

[6] CAPRARA A., FISCHETTI M., TOTH P., *A heuristic method for the set covering problem*, Operations Research, **47** (5), pp. 730–743, 1999.

[7] CASERTA M., *Tabu search-based metaheuristic algorithm for large-scale set covering problems*, in: Doerner, K., Gendreau, M., Greistorfer, P., Gutjahr, W., Hartl, R., Reimann, M. (Eds.), *Metaheuristics*. Vol. **39** of Operations Research/Computer Science Interfaces Series. Springer US, pp. 43–63, 2007.

[8] CERIA S., NOBILI P., SASSANO A., *A lagrangian-based heuristic for large-scale set covering problems*, Mathematical Programming, **81** (2), pp. 215–228, 1998.

[9] CHVATAL V., *A greedy heuristic for the set-covering problem*, Mathematics of Operations Research, **4** (3), pp. 233–235, 1979.

[10] CRAWFORD B., LAGOS C., CASTRO C., PAREDES F., *A evolutionary approach to solve set covering*, in: Cardoso, J., Cordeiro, J., Filipe, J. (Eds.), ICEIS (2), pp. 356–363, 2007.

[11] CRAWFORD B., SOTO R., CASTRO C., MONFROY E., *Extensible cp-based autonomous search*, Proceedings of HCI International, Vol. **173** of CCIS. Springer, pp. 561–565, 2011.

[12] CRAWFORD B., SOTO R., MONFROY E., *Cultural algorithms for the set covering problem*, in: Tan, Y., Shi, Y., Mo, H. (Eds.), ICSI (2). Vol. **7929** of Lecture Notes in Computer Science, Springer, pp. 27–34, 2013.

[13] CRAWFORD B., SOTO R., MONFROY E., CASTRO C., PALMA W., PAREDES F., *A hybrid soft computing approach for subset problems*, Mathematical Problems in Engineering, Article ID 716069, pp. 1–12, 2013.

[14] CRAWFORD B., SOTO R., MONFROY E., PALMA W., CASTRO C., PAREDES F., *Parameter tuning of a choice-function based hyperheuristic using particle swarm optimization*, Expert Systems with Applications, **40** (5), pp. 1690–1695, 2013.

[15] CRAWFORD B., SOTO R., MONTECINOS M., CASTRO C., MONFROY E., *A framework for autonomous search in the* ecl^ips^e *solver*, Proceedings of the 24th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE), Vol. **6703** of LNCS. Springer, pp. 79–84, 2011.

[16] FISHER M. L., KEDIA P., *Optimal solution of set covering/partitioning problems using dual heuristics*, Management Science, **36** (6), pp. 674–688, 1990.

[17] FISTER I., FISTER I. JR., YANG X.-S., BREST J., *A comprehensive review of firefly algorithms*, Swarm and Evolutionary Computation, **13** (0), pp. 34–46, 2013.

[18] GAREY M. R., JOHNSON D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA, 1990.

[19] HORNG M.-H., *Vector quantization using the firefly algorithm for image compression*, Expert Syst. Appl., **39** (1), pp. 1078–1091, 2012.

[20] HOUSOS E., ELMROTH T., *Automatic optimization of subproblems in scheduling airline crews*, Interfaces, **27** (5), pp. 68–77, 1997.

[21] MIGUEL L. F. F., MIGUEL L. F. F., *Shape and size optimization of truss structures considering dynamic constraints through modern metaheuristic algorithms*, Expert Systems with Applications, **39** (10), pp. 9458–9467, 2012.

[22] MIRJALILI S., LEWIS A., *S-shaped versus v-shaped transfer functions for binary particle swarm optimization*, Swarm and Evolutionary Computation, **9** (0), pp. 1–14, 2013.

[23] NAJI-AZIMI Z., TOTH P., GALLI L., *An electromagnetism metaheuristic for the unicost set covering problem*, European Journal of Operational Research, **205** (2), pp. 290–300, 2010.

[24] REN Z.-G., FENG Z.-R., KE L.-J., ZHANG Z.-J., *New ideas for applying ant colony optimization to the set covering problem*, Computers & Industrial Engineering, **58** (4), pp. 774–784, 2010.

[25] SAYADI M. K., HAFEZALKOTOB A., NAINI S. G. J., *Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation*, Journal of Manufacturing Systems, **32** (1), pp. 78–84, 2013.

[26] SOTO R., CRAWFORD B., MONFROY E., BUSTOS V., *Using autonomous search for generating good enumeration strategy blends in constraint programming*, Proceedings of the 12th International Conference on Computational Science and Its Applications (ICCSA), Vol. **7335** of LNCS, Springer, pp. 607–617, 2012.

[27] VASKO F. J., WILSON G. R., *Using a facility location algorithm to solve large set covering problems*, Operations Research Letters, **3** (2), pp. 85–90, 1984.

[28] VASKO F. J., WOLF F. E., STOTT K. L., *Optimal selection of ingot sizes via set covering*, Oper. Res., **35** (3), pp. 346–353, Jun. 1987.

[29] YANG X.-S., *Firefly algorithms for multimodal optimization*, Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications, SAGA'09, Springer-Verlag, Berlin, Heidelberg, pp. 169–178, 2009.

[30] YANG X.-S., *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2010.

[31] YANG X.-S., HE X., *Firefly algorithm: Recent advances and applications*, CoRR abs/1308.3898, 2013.