Research

A Meta-Optimization Approach to Solve the Set Covering Problem.

Un enfoque de Meta-Optimización para Resolver el Problema de Cobertura de Conjunto

Broderick Crawford[1]***, Ricardo Soto***[1]***, Eric Monfroy***[2]***, Gino Astorga***[*,1,3]***, José García***[1,4]***, Enrique Cortes***[1,5]

[1]Pontificia Universidad Católica de Valparaíso. Chile, [2]Universite de Nantes. Francia, [1,3]Universidad de Valparaíso. Chile, [1,4]Centro de Investigación y Desarrollo Telefónica, [1,5]Universidad de Playa Ancha. Valparaiso - Chile

[*]Correspondence email: gino.astorga@uv.cl

Received: 16-04-2018. Modified: 09-08-2018. Accepted: 14-09-2018

Abstract

Context: In the industry the resources are increasingly scarce. For this reason, we must make a good use of it. Being the optimization tools, a good alternative that it is necessary to bear in mind. A real-world problem is the facilities location being the Set Covering Problem, one of the most used models. Our interest, it is to find solution alternatives to this problem of the real-world using metaheuristics.

Method: One of the main problems which we turn out to be faced on having used metaheuristic is the difficulty of realizing a correct parametrization with the purpose to find good solutions. This is not an easy task, for which our proposal is to use a metaheuristic that allows to provide good parameters to another metaheuristics that will be responsible for resolving the Set Covering Problem.

Results: To prove our proposal, we use the set of 65 instances of OR-Library which also was compared with other recent algorithms, used to solve the Set Covering Problem.

Conclusions: Our proposal has proved to be very effective able to produce solutions of good quality avoiding also have to invest large amounts of time in the parametrization of the metaheuristic responsible for resolving the problem.

Keywords: Artificial Bee Colony Algorithm, Meta-Optimization, Set Covering Problems.

Language: English

Cite this paper as: B. Crawford, R. Soto, E. Monfroy, G. Astorga, J. García and E. Cortes. "A Meta-Optimization Approach to Solve the Set Covering Problem", Ingeniería, vol. 23, no.3, pp. 274-288, Sep-Dic 2018.
© The authors; reproduction right holder Universidad Distrital Francisco José de Caldas.
DOI:https://doi.org/10.14483/23448393.13247

Resumen

Contexto: En la industria los recursos son cada vez más escasos, por esta razón se debe hacer un buen uso de ellos y las herramientas de optimización son una buena alternativa que se debe tener presente. Un problema del mundo real lo contituye la ubicación de instalaciones, siendo el problema de cobertura de conjuntos uno de los modelos más utilizados. El presente interés es encontrar alternativas de solución a este problema de la vidareal, utilizando metaheurísticas.
Método: Uno de los principales problemas que se enfrentan al utilizar metaheurísticas es la dificultad de realizar una correcta parametrización con el objetivo de encontrar buenas soluciones. Esta no es una tarea fácil, por lo cual la propuesta es utilizar una metaheurística que permita proporcionar buenos parametros a otra metaheurística que será la encargada de resolver el problema de cobertura de conjuntos.
Resultados: Para probar la propuesta, se utiliza el set de 65 instancias de OR-Library, el cual fue comparado con otros recientes algoritmos que son usados para resolver el problema de cobertura de conjuntos.
Conclusiones: La propuesta ha demostrado ser muy efectiva, logrando producir soluciones de buena calidad y evitando, además, que se tenga que invertir gran cantidad de tiempo en la parametrización de la metaheurística encargada de resolver el problema.
Palabras clave: Algoritmo colonia de abejas artificiales, metaoptimización, problema de cobertura.
Language: Inglés

1. Introduction

The Set Covering Problem (SCP), introduced in [1], is an important problem NP-Hard present in the current industry. The following applications for covering problems were mentioned in [2]: Bus stop location, Fire equipment allocation, Fire company relocation, Fire service sitting and Terrain visibility. Also, in [3] were presented some general applications of the gradual covering problem: The delivery problem; Competitive location; Dense competition; The radio, TV, or cellular transmitter problem and Medical facility location problem.

Mathematically SCP can be defined as: Let $A = (a_{ij})$ be an m-row, n-column, zero-one matrix. We say that a column j can cover a row i if $a_{ij} = 1$. Each column j is associated with a non-negative 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 stated in the following:

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

subject to

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

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

If the costs c_j are equal for each $j \in J$, the problem is referred to as the unicost SCP, otherwise, the problem is called the weighted or non-unicost SCP, where $J_i = \{j \in J : a_{ij} = 1\}$: the subset of columns covering row i and $I_j = \{i \in I : a_{ij} = 1\}$: the subset of rows covered by column j.

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 constraints ensure that each row i is covered by at least one column.

Different solving methods have been proposed in the literature for the Set Covering Problem. The use of metaheuristics is a good alternative to tackle this problem as can be swarm intelligence continuous metaheuristic. Because they are continuous metaheuristics and SCP is a combinatorial problem, these metaheuristics must be accompanied by a binarization mechanism. In the literature, we find the main binarization technique used to solve SCP corresponds to transfer functions, for more details on binarization techniques see gas [4], [5]. Among the main algorithms that use this technique we found a cat swarm [6], a binary Firefly Optimization [7], a Binary Cuckoo Search (BCS) [8] and artificial bee colony [9]. Specific binarization techniques have also been developed to solve SCP, among the most efficient are: a Teaching-learning binarization [10], a Binary Black Hole (BBH) [11], and a specific Jumping Particle Swarm Optimization (JPSO) method [12].

Depending on the algorithm that has been used, the quality of the solution wanted and the complexity of the SCP chosen, it is defined the amount of customization effort required. Being of paramount importance the determination of the values that are given to the parameters. Conveniently, this work proposes a meta-optimization approach where the task of customization is transferred to another metaheuristic (a “high level”metaheuristic) which can handle the task of parameters adjustment for a low level metaheuristic [13].

Our proposal considers a Genetic Algorithm (GA) for parameter setting and an ABC algorithm at a lower level using an Automatic Parameter Tuning approach. The Automatic Parameter Tuning is carried by the GA which searches for the best parameters in the parameter space in order to tune the solver automatically.

This approach is considered as meta-optimization since there are two metaheuristics covering tasks of parameter setting, for the former, and problem solving, for the latter [14]. This work is an extension of [15] where emphasis is given to the percentage of improvement of the different instances.

The rest of this paper is organized as follows. In Section 2, we briefly survey the ABC algorithm. In section 3 we present the meta-optimization approach. In Section 4, we present the experimental results obtained. The analysis of the results of comparing our proposal with a constructive metaheuristic is presented in section 5. Finally, in Section 6 we conclude the paper and give some perspectives for further research.

2. Artificial Bee Colony Algorithm

ABC is one of the most recent algorithms in the domain of the collective intelligence [16]. Created by Dervis Karaboga in 2005, who was motivated by the intelligent behavior observed in the domestic bees to take the process of foraging [17]. ABC is an algorithm of combinatorial optimization based on populations, in which the solutions of the problem of optimization, the sources of food, are modified by the artificial bees, that fungen as operators of variation. The aim of these bees is to discover the food sources with major nectar.

In the ABC algorithm, the artificial bees move in a multidimensional search space choosing sources of nectar depending on its past experience and its companions of beehive or fitting his position. Some bees (exploratory) fly and choose food sources randomly without using experience. When they find a source of major nectar, they memorize his position and forget the previous one. Thus, ABC combines methods of local search and global search, trying to balance the process of the exploration and exploitation of the space of search. The Flow chart of Artificial Bee Colony is showed in Fig. 1.

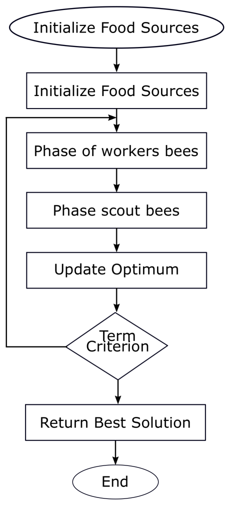

Figure 1. Flow Chart of ABC Algorithm.

The procedure for determining a food source in the neighbourhood of a particular food source which depends on the nature of the problem. Karaboga [18] developed the first ABC algorithm for continuous optimization. The method for determining a food source in the neighbourhood of a particular food source is based on changing the value of one randomly chosen variable while keeping other variables unchanged. This is done by adding to the current value of the variable the product of a uniform value in [-1, 1] and the difference in values of this variable for this food source and some other randomly chosen food source. This approach can not be used for discrete optimization problems for which it generates at best a random effect.

Singh [19] subsequently proposed a method, which is appropriate for subset selection problems. In his model, to generate a neighbouring solution, an object is randomly dropped from the solution and in its place another object, which is not already present in the solution is added. The object to be added is selected from another randomly chosen solution. If there are more than one candidate objects for addition then ties are broken arbitrarily. In this work we use the ABC algorithm described in [20] and extending the work presented in [14].

3. A Meta-Optimization Approach to Solve the SCP

Metaheuristics, in their original definition, are solution methods that orchestrate an interaction between local improvement procedures and higher level strategies. Thus, metaheuristics create a process capable of escaping from local optima and performing a robust search of a solution space.

Over time, these methods have also come to include any procedures that employ strategies for overcoming the trap of local optimality in complex solution spaces. The use of one or more neighborhood structures as a means of defining admissible moves to transition from one solution to another, or to build or destroy solutions in constructive and destructive processes are examples of such procedures.

3.1. Parameter setting

The selection of an adequate set of values for parameters improves the performance of metaheuristic methods. This configuration can be realized of two ways:

3.1.1. Offline configuration

It consists of finding the appropriate configuration of parameters of the algorithm before the algorithm is executed. It is mostly done in the form of trial and error. This makes tuning process a very high consuming time task. There depends much of the intuition and the experience of the creator of the algorithm. They are typically undocumented and therefore not reproducible driving often to an unequal adjustment of different algorithms. An alternative to find good values for the parameters shown in [21]. Within this group are the racing methods where it is evaluated iteratively different candidate configurations for a certain number of instances [22];Also this Sequential Model-Based Optimisation, this approach, consists of improving the initial values of the parameters alternating the experiment design and the identification of the parameters [23] and Graphic Radial Method, in this approach radar chart curve is used to define the best configuration [5].

3.1.2. Online configuration

It is an important research area, since the algorithms can adapt themselves better to the characteristics of a particular instance. In the search process, it is essential to identify the phases of exploration and exploitation of the algorithm since the adjustment of parameters can be different in each stage allowing to achieve a better performance. Can improve results in cases of algorithms that are used in situations which are very different to those that were built.

Different techniques exist, being the most simple to define the rule variation of parameters before executing the algorithm.

The on-line approaches can be classified in dynamic and adaptive. Dynamic approaches are those where the updating of the parameters is performed in a random or deterministic way. On the other hand in adaptive approaches the memory is used and the change of value is made according to the progress in the search process. One special case is the autonomous search concept where internal and external information is used for adjustment. Find a correct configuration of parameters constitutes an optimization problem for which we can use a meta-Level optimizer.

3.2. Meta-Optimization

A meta-optimization approach can be considered as two or more metaheuristics where a higher level metaheuristic controls the parameters of a lower level one, which is at charge of dealing more directly to the problem. Our ABC algorithm employs four control parameters which are: number of food sources, the value of limit, % of Columns to add, and % of Columns to eliminate. On a higher level GA allows to evaluate different parameter configurations avoiding manual configuration. Each GA individual encodes the parameters of an ABC algorithm generating an ABC instance. The detail of the proposed approach is presented below:

3.3. Higher level metaheuristic

GA has been successfully used in various algorithm configurations, a particular case is the presented by Grefenstette [24], that used it to find the parameter values of another genetic algorithm. Another example is [25] where several subpopulations are used and a specialized cross-over operator to generate new candidate configurations. On the other hand it was recently used by Senthilkumar in [26] to find the parameters of the arc welding process with flux core. Using a proven algorithm in the search for configurations, allows us to validate our proposal.

In the GA component, the chromosome genes are: "*Food sources*", it is the number of initial solutions for ABC (which is equal to the number of workers or onlookers bees), it will take values between 50 and 500. The second gene, "*Limit*", it takes values between 0 and 100. Similarly, the third and fourth genes, " *% Columns to Add*" and " *% Columns to Eliminate*", they take values between 0.01 and 10.

During each generation of the GA algorithm, in order to produce offsprings, a tournament selection method is used to select a set of individuals from the population as follows. Given the size k of the tournament group and a probability p, the k individuals are sorted using their fitness value and a random value r is generated. If $r \leq p$, the best individual is choosen, otherwise the probability p is incremented, a new random value is generated and the tournament group is reduced by one. This process is repeated at most k times. If no individual is chosen after k attempts, the worst individual belonging to the tournament group is selected. To select n individuals this selection procedure is carried out n times.

The operator randomly selects two chromosomes from the population and performs the crossover as follows. It generates randomly a binary crossover mask with the same length as the chromosomes. If the value of a bit is 1, chromosome information is copied from the first parent. If the value is 0, the genes from the second parent are used and vice-versa for the second offspring.
The mutation operator is applied with a certain rate replacing the value of a gene with a value drawn uniformly from its domain.

3.4. Lower level Metaheuristic

In the ABC component, a first step is performed in order to initialize the parameters of ABC as size of the colony, number of workers and onlookers bees, limit of attempts and maximum number of cycles (iterations). To generate the initial population we cross every row of the counterfoil of constraints and by every row a column is selected at random. This column is part of the solution which is represented by means of an entire vector. This vector considers the columns chosen in one solution. To complete this vector a procedure is performed for all the rows in such a way that the generated solution complies with all the constraints. Then, the evaluation of the population fitness is performed using the objective function, see (1). Afterwards, the modification of the position and selection of sites for worker bees is performed as follows. A hard-working bee modifies its current position selecting a food source randomly. If a hard-working bee duplicates a solution, it is transformed to an explorer bee. Otherwise, it proceeds to add a certain random number of columns between 0 and the maximum number of columns to be added. Then, it proceeds to eliminate a certain random number of columns between 0 and the maximum number of columns to be eliminated. If the new solution does not meet the constraints, it is repaired.

We use repair method where all rows not covered are identified and the columns required are added. So in this way all the constraints will be covered. The search of these columns are based in the relationship showed in the Equation 4.

$$\frac{cost\ of\ one\ column}{amount\ of\ columns\ not\ covered} \tag{4}$$

Once the columns are added and the solution is feasible, a method is applied to remove redundant columns of the solution. A redundant column are those that are removed, the solution remains a feasible solution.

After this, the fitness of the solution is evaluated by means of the objective function of the SCP and if the fitness is minor that the solution previously obtained, the solution is replaced. Otherwise, the number of attempts for improving this solution is increased and the algorithm continues evaluating another hard-working bee.

3.5. Integrating Components

The Figure 2 shows the meta-optimization approach developed to solve the SCP. Once the GA population is generated, each individual is taken to run the ABC algorithm until a certain cut-off. Then, the genetic operators are applied and a termination criterion is evaluated in order to stop the parameter setting. Once the termination criterion is achieved, the best individual from the GA contains the best parameter set which is selected to run the ABC algorithm.

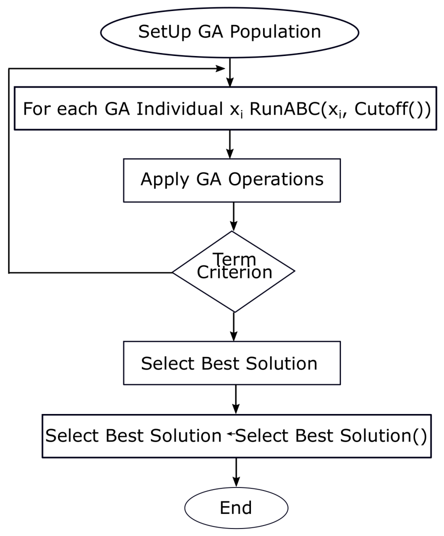
SetUp GA Population
For each GA Individual x_i RunABC(x_i, Cutoff())
Apply GA Operations
Term Criterion
Select Best Solution
Select Best Solution ← Select Best Solution()
End

Figure 2. *Tuning ABC()*.

4. Results

In this section we detail the behaviour of our approach. To solve the different SCP instances, a Computer with Windows 7, 2.5 GHz Dual Core processor and 4GB in RAM was used.

The ABC algorithm was executed 30 times for each instance, where the main features are shown in the Table I,where the density corresponds to the percentage of non-zero in the matrix.

In the top level Java Package was used [1] (jgap) 3,5 version to implement the genetic algorithm using the parameters shown in the Table II. The GA implementation has 3 main phases: configuration, initial population and evolution of the population.

[1]http://jgap.sourceforge.net

Table I. Features of the 65 instances [20].

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	Known
NRF	5	500	5000	[1, 100]	20	Known
NRG	5	1000	10000	[1, 100]	2	Unknown
NRH	5	1000	10000	[1, 100]	5	Unknown

Table II. Parameters used in GA

Parameter	Value	Parameter	Value
Number of generations	20	Mask probability	0.5
Population size	30	Mutation rate	0.025
Crossover type	Uniform crossover	Selector tournament size	3
Crossover rate	0.4	Tournament selector parameter (p)	0.75

The best parameters settings for each instance were found using GA, since these may vary because the search space can change very much of one instance to another. In the table III shows the values of the parameters of the best chromosomes obtained.

Table III. Parameters values from best chromosome

Instance set	Food sources	Limit	% Columns to add	% Columns to eliminate
4	83	30	0.4	1
5	77	40	0.7	1.2
6	106	37	0.6	1.3
A	93	53	0.6	1.1
B	85	50	0.2	1.7
C	100	70	0.5	1.4
D	112	66	0.2	1.5
NRE	98	38	0.3	1.5
NRF	200	51	0.3	1.7
NRG	103	70	0.3	1.6
NRH	107	53	0.4	2

In order to compare our proposal with other works and given information available, we use the relative percentage deviation (RPD) which quantifies the deviation of the target value Z of $Zopt$. We report the optimal value, the best found value using our proposed and its average. The results are shown in the Tables V, VI and VII.

To validate our proposal were resolved the 65 instances of OR-Library and were compared with published results of recent approaches: Binary Cat Swarm Optimization (BCSO) [6]; Binary Firefly Optimization (BFO) [27]; Binary Shuffled Frog Leaping Algorithm (BSFLA) [28]; Binary Electromagnetism - Like Algorithm (BELA) [29]; and Binary Artificial Bee Colony (BABC) [30]. The comparison is done using relative percentage deviation (RPD).

The table V show the results obtained for instances from group 4, where ABC had the best optimal values compared to the results by the previous approaches, only BFO exhibit a good performance for this data with two global optimums.

Table IV. Results obtained for IWD - Meta-Optimization

		IWD			M-OPT		
Instance	**Optimal**	**Min**	**Avg**	**RPD**	**Min**	**Avg**	**RPD**
4.1	429	454	468.73	5.83	430	430.5	0.2
5.1	253	294	302.00	12.65	254	255	0.4
6.1	138	177	217.96	28.26	140	140.5	1.4
A.1	253	310	350.68	22.53	254	254	0.4
B.1	69	92	124.44	33.33	69	69	0.0
C.1	227	227	351.74	30.84	230	231	1.3

In the group 5, ABC, BFO, BSFLA AND BABC obtained optimal values, detailed as follows: ABC, four optimal values; BFO, three optimal values; BSFLA, four optimal values; BABC, two optimal values. BELA and BCSO did not achieve some optimum. The results are shown in the Table VI.

The results of groups B, C and H are shown in the Table VII where it is visualized that ABC achieves all the optimals for groups B and H, also achieving 2 optimal for group C. In relation to the other techniques used in the comparison, BFO finds two optimal values, BSFLA finds four optimal values. In the case of BABC, there are two optimal values. BCSO not get optimal as well as BELA.

Additionally, our proposal was compared with the constructive Metaheuristic called Intelligent Water Drops presented by [31]. For this comparison was used the first instances of the groups 4, 5, 6, A, B y C. The results are shown in the table IV.

Table V. Results obtained for Set 4

Instance		**4.1**	**4.2**	**4.3**	**4.4**	**4.5**	**4.6**	**4.7**	**4.8**	**4.9**	**4.10**
Z_{opt}		**429**	**512**	**516**	**494**	**512**	**560**	**430**	**492**	**641**	**514**
New approach											
ABC	Z_{min}	430.0	512.0	516.0	494.0	512.0	561.0	430.0	493.0	643.0	514.0
	Z_{avg}	430.5	512.0	516.0	494.0	512.0	561.7	430.0	494.0	645.5	514.0
	RPD	0.2	0.0	0.0	0.0	0.0	0.2	0.0	0.2	0.3	0.0
Previous approaches											
BCSO	Z_{min}	459	570	590	547	545	637	462	546	711	537
	Z_{avg}	480	594	607	578	554	650	467	567	725	552
	RPD	7	11.3	14.3	10.7	6.4	13.8	7.4	11	10.9	4.5
BFO	Z_{min}	429	517	519	495	514	563	430	497	655	519
	Z_{avg}	430	517	522	497	515	565	430	499	658	523
	RPD	0	0.97	0.58	0.2	0.39	0.53	0	1.01	2.18	0.97
BSFLA	Z_{min}	430	516	520	501	514	563	431	497	656	518
	Z_{avg}	430	518	520	504	514	563	432	499	656	519
	RPD	0.23	0.78	0.78	1.42	0.39	0.54	0.23	1.02	2.34	0.78
BELA	Z_{min}	447	559	537	527	527	607	448	509	682	571
	Z_{avg}	448	559	539	530	529	608	449	512	682	571
	RPD	4.20	9.18	4.07	6.68	2.93	8.39	4.19	3.46	6.40	11.09
BABC	Z_{min}	430	513	519	495	514	561	431	493	649	517
	Z_{avg}	430	513	521	496	517	565	434	494	651	519
	RPD	0.23	0.20	0.58	0.20	0.39	0.18	0.23	0.20	0.93	0.58

In order to demonstrate that our approach helps to improve the overall time to solve a problem, to contribute to the search for a good configuration parameters in less time, we use the table VIII it shows the times used for the configuration of each instance. Manually we used 5 hours, in contrast with GA we use 720 seconds for each group of instances.

In the Table VIII we can see that the biggest instances the percentage of progress of the time diminishes what it makes necessary to incorporate in the future other techniques to improve this behavior.

Table VI. Results obtained for Set 5

Instance		**5.1**	**5.2**	**5.3**	**5.4**	**5.5**	**5.6**	**5.7**	**5.8**	**5.9**	**5.10**
Z_{opt}		**253**	**302**	**226**	**242**	**211**	**213**	**293**	**288**	**279**	**265**
New approach											
ABC	Z_{min}	254.0	309.0	228.0	242.0	211.0	213.0	296.0	288.0	280.0	266.0
	Z_{avg}	255.0	310.2	228.5	242.0	211.0	213.0	296.0	288.0	279.2	267.0
	RPD	0.4	2.3	0.9	0.0	0.0	0.0	1.0	0.0	0.4	0.4
Previous approaches											
BCSO	Z_{min}	279	339	247	251	230	232	332	320	295	285
	Z_{avg}	287	340	251	253	230	243	338	330	297	287
	RPD	10.3	12.3	9.3	3.7	9	8.9	13.3	11.1	5.7	7.5
BFO	Z_{min}	257	309	229	242	211	213	298	291	284	268
	Z_{avg}	260	311	233	242	213	213	301	292	284	270
	RPD	1.58	2.31	1.32	0	0	0	1.7	1.04	1.79	1.13
BSFLA	Z_{min}	254	307	228	242	211	213	297	291	281	265
	Z_{avg}	255	307	230	242	213	214	299	293	283	266
	RPD	0.4	1.66	0.88	0	0	0	1.37	1.04	0.72	0
BELA	Z_{min}	280	318	242	251	225	247	316	315	314	280
	Z_{avg}	281	321	240	252	227	248	317	317	315	282
	RPD	10.67	5.30	7.08	3.72	6.64	15.96	7.85	9.38	12.54	5.66
BABC	Z_{min}	254	309	229	242	211	214	298	289	280	267
	Z_{avg}	255	309	233	245	212	214	301	291	281	270
	RPD	0.40	2.32	1.33	0	0	0.47	1.71	0.35	0.36	0.75

Table VII. Results obtained for Set B, C and H

Instance		**B.1**	**B.2**	**B.3**	**B.4**	**B.5**	**C.1**	**C.2**	**C.3**	**C.4**	**C.5**	**H.1**	**H.2**	**H.3**	**H.4**	**H.5**
Z_{opt}		**69**	**76**	**80**	**79**	**72**	**227**	**219**	**243**	**219**	**215**	**63**	**63**	**59**	**58**	**55**
New approaches																
ABC	Z_{min}	69.0	76.0	80.0	79.0	72.0	230.0	219.0	244.0	220.0	215.0	63.0	63.0	59.0	58.0	55.0
	Z_{avg}	69.0	76.0	80.0	79.0	72.0	231.0	219.0	244.5	224.0	215.0	63.0	63.0	59.0	58.0	55.0
	RPD	0.0	0.0	0.0	0.0	0.0	1.3	0.0	0.4	0.5	0.0	0.0	0.0	0.0	0.0	0.0
Previous approaches																
BCSO	Z_{min}	79	86	85	89	73	242	240	277	250	243	70	67	68	66	61
	Z_{avg}	79	89	85	89	73	242	241	278	250	244	71	67	70	67	62
	RPD	14.5	13.2	6.3	12.7	1.4	6.6	9.6	14	12.3	13	11.1	6.3	15.3	13.8	10.9
BFO	Z_{min}	71	78	80	80	72	230	223	253	225	217	69	66	65	63	59
	Z_{avg}	72	78	80	81	73	232	224	254	227	219	70	66	67	65	60
	RPD	2.89	2.63	0	1.26	0	1.32	1.82	4.11	2.73	0.93	9.52	4.76	10.16	6.77	7.27
BSFLA	Z_{min}	70	76	80	79	72	229	223	253	227	217	68	66	62	63	59
	Z_{avg}	70	77	80	80	73	231	225	253	228	218	69	66	63	64	61
	RPD	1.45	0	0	0	0	0.88	1.83	4.12	3.65	0.93	7.94	4.76	5.08	8.62	7.27
BELA	Z_{min}	86	88	85	84	78	237	237	271	246	224	70	71	68	70	69
	Z_{avg}	87	88	87	88	81	238	239	271	248	225	71	71	70	72	69
	RPD	24.64	15.79	6.25	6.33	8.33	4.41	8.22	11.52	12.33	4.19	11.11	12.70	15.25	20.69	25.45
BABC	Z_{min}	70	78	80	80	72	231	222	254	231	216	70	69	66	64	60
	Z_{avg}	70	79	80	81	74	233	223	255	233	217	71	72	67	64	61
	RPD	1.45	2.63	0	1.27	0	1.76	1.37	4.53	5.48	0.47	11.11	9.52	11.86	10.34	9.09

Table VIII. Comparison of ABC with manually and automatically tuned parameters

Instance	Manual tuned	Auto tuned	%	Instance	Manual tuned	Auto tuned	%	Instance	Manual tuned	Auto tuned	%
4.1	282.0	77.1	72.66	6.3	287.2	154.3	46.27	D.5	344.3	211.4	38.60
4.2	281.7	76.8	72.74	6.4	286.0	153.1	46.47	NRE.1	366.0	233.1	36.31
4.3	281.8	76.9	72.71	6.5	290.0	157.1	45.83	NRE.2	375.7	242.8	35.37
4.4	282.3	77,4	72.58	A.1	288.8	155.9	46.02	NRE.3	381.8	248.9	34.80
4.5	282.5	77.6	72.53	A.2	288.1	155.1	46.16	NRE.4	370.1	237.2	35.90
4.6	282.0	77.1	72.66	A.3	287.7	154.7	46.23	NRE.5	374.6	241.7	35.47
4.7	281.9	77.0	72.69	A.4	288.5	155.6	46.07	NRF.1	607.3	474.4	21.88
4.8	282.3	77.4	72.58	A.5	287.6	154.7	46.21	NRF.2	559.5	426.5	23.77
4.9	281.2	76.3	72.87	B.1	316.3	183.4	42.02	NRF.3	585.4	452.5	22.70
4.10	282.5	77.6	72.53	B.2	314.6	181.6	42.28	NRF.4	602.0	469.1	22.07
5.1	283.9	79.0	72.17	B.3	311.7	178.8	42.64	NRF.5	601.2	468.2	22.12
5.2	283.0	78.1	72.40	B.4	318.6	185.7	41.71	NRG.1	374.5	241.5	35.51
5.3	284.3	79.4	72.07	B.5	312.9	179.9	42.51	NRG.2	373.8	240.9	35.55
5.4	283.1	78.2	72.38	C.1	294.1	161.2	45.19	NRG.3	371.3	238.4	35.79
5.5	283.5	78.6	72.28	C.2	294.2	161.3	45.17	NRG.4	369.4	236.5	35.97
5.6	283.6	78.7	72.25	C.3	295.7	162.8	44.94	NRG.5	369.7	236.8	35.94
5.7	284.2	79.3	72.10	C.4	294.5	161.5	45.16	NRH.1	809.3	676.4	16.42
5.8	283.0	78.1	72.40	C.5	294.6	161.6	45.15	NRH.2	810.8	677.9	16.39
5.9	283.0	78.1	72.40	D.1	355.5	222.6	37.38	NRH.3	834.1	701.1	15.94
5.10	283.1	78.1	72.41	D.2	351.2	218.3	37.84	NRH.4	842.7	709.8	15.77
6.1	288.5	155.6	46.07	D.3	351.5	218.6	37.81	NRH.5	816.0	683.1	16.28
6.2	287.6	154.7	46.21	D.4	343.1	210.1	38.76				

In tables IX to XI we analysed the difference of RPD of the different techniques with respect to the RPD obtained by ABC. We can observe that for Set 4 (Figure IX)in average the lowest percentage of improvement corresponds to 36,41 %, however low to 27.84 % for set 5. For the most complicated groups, our proposal also presents important improvement with respect to the other techniques.

Table IX. Comparison RPD Set 4 with ABC

	4.1	**4.2**	**4.3**	**4.4**	**4.5**	**4.6**	**4.7**	**4.8**	**4.9**	**4.10**
BCSO	97,14	100	100	100	100	0,00	100	98,18	97,25	100
BFO	0,00	100	100	100	100	96,88	-	80,20	86,24	100
BSFLA	13,04	100	100	100	100	48,72	100	80,39	87,18	100
BELA	95,24	100	100	100	100	48,72	100	94,22	95,31	100
BABC	13,04	100	100	100	100	78,49	100	0,00	67,74	100
AVG	36,41	100	100	100	100	54,56	100	70,60	86,74	100

Table X. Comparison RPD Set 5 with ABC

	5.1	**5.2**	**5.3**	**5.4**	**5.5**	**5.6**	**5.7**	**5.8**	**5.9**	**5.10**
BCSO	96,12	81,30	90,32	100	100	100	92,48	100	92,98	94,67
BFO	74,68	0,43	31,82	0,00	0,00	0,00	41,18	100	77,65	64,60
BSFLA	0,00	0,00	0,00	0,00	0,00	0,00	27,01	100	44,44	0,00
BELA	96,25	56,60	87,29	100	100	100	87,26	100	96,81	92,93
BABC	0,00	0,86	32,33	0,00	0,00	100	41,52	100	0,00	46,67
AVG	53,41	27,84	48,35	40,00	40,00	60,00	57,89	100	62,38	59,77

Table XI. Comparison RPD Set B, C and H with ABC

	B.1	**B.2**	**B.3**	**B.4**	**B.5**	**C.1**	**C.2**	**C.3**	**C.4**	**C.5**	**H.1**	**H.2**	**H.3**	**H.4**	**H.5**
BCSO	100	100,00	100,00	100	100	80,30	100,00	97,14	95,93	100	100	100	100	100	100
BFO	100	100,00	0,00	100	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
BSFLA	100	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
BELA	100	100	100,00	100	100	70,52	100,00	96,53	95,94	100	100	100	100,00	100	100
BABC	100	100	0,00	100	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
AVG	100	80,00	40,00	80,00	40,00	30,16	40,00	38,73	38,38	40,00	40,00	40,00	40,00	40,00	40,00

5. Statistical Analysis

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.
- Wilcoxons Signed Rank [33] is used to verify superiority of the strategy of resolution using Meta-optimization in relation to the IWD algorithm.

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:

$H_0 =$ The data follow a normal distribution.
$H_1 =$ The 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 the superiority of the resolution strategy using Meta-Optimization in relation to the IWD Algorithm, fitness is used, and the following hypotheses are defined:
$H_0 =$ IWD algorithm $\geq$ Meta-Optimization

Table XII. Statistical Analysis Results.

	IWD	M-Opt	
Instance	**Min**	**Min**	**p-value**
4.1	454	430	6.273690e-12
5.1	285	254	2.203342e-11
6.1	177	140	2.660479e-11
A.1	310	254	7.643806e-12
B.1	92	69	1.677019e-12
C.1	227	231	1.530219e-11

H_1 = IWD algorithm < Meta-Optimization

It is obtained p-value < 0.05; therefore, the hypothesis H_0 is rejected, and H_1 is accepted, which implies that Meta-optimization provides better results. This procedure extends to each instance of the benchmark (Table XII). The results of this analysis are coincident with those verified by RPD.

6. Conclusions

In our work we use a metaheuristics with a meta-optimization approach to solve the SCP. The results revealed, once again, the importance of the parametrization of the algorithms. The results were very good in comparison with the other proposals used. Clearly the use of the automation in the parameters, allows to significantly improve the results in relation to the manual parameterization. In theis work, was possible to observe that the time of instances of the groups 4 and 5, considering the time used in the process of configuration, was greater than 50 percent. As future work, we propose the use of other metaheuristics with different parameters with the order to improve the results and also to facilitate the use by simplifying the parametrization of the metaheuristic of top level. In addition you can use this approach to other problems of the real world as is the problem of routing of school buses with time windows [34].

Acknowledgment

Broderick Crawford is supported by grant CONICYT / FONDECYT / REGULAR 1171243 and Ricardo Soto is supported by Grant CONICYT / FONDECYT / REGULAR / 1160455, Gino Astorga is supported by Postgraduate Grant, Pontificia Universidad Catolica de Valparaíso, 2015 and José García is supported by INF-PUCV 2016. This research was partially funded by CORFO Program Ingeniería 2030 PUCV - Consortium of Chilean Engineering Faculties.

Referencias

[1] C. Toregas, R. Swain, C. ReVelle, and L. Bergman, "The location of emergency service facilities," *Operations Research*, vol. 19, no. 6, pp. 1363–1373, 1971. https://doi.org/10.1287/opre.19.6.1363↑. 275

[2] D. Schilling, V. Jayaraman, and R. Barkhi, "A review of covering problem in facility location," *Location Science*, vol. Vol. 1, no. 1, pp. 25–55, 1993↑. 275

[3] T. Drezner, Z. Drezner, and Z. Goldstein, "A stochastic gradual cover location problem," *Naval Research Logistics (NRL)*, vol. 57, no. 4, pp. 367–372, 2010. https://doi.org/10.1002/nav.20410↑. 275

[4] B. Crawford, R. Soto, G. Astorga, J. García, C. Castro, and F. Paredes, "Putting continuous metaheuristics to work in binary search spaces," *Complexity*, vol. 2017, no. 2, 2017. https://doi.org/10.1155/2017/8404231↑. 276

[5] J. García, B. Crawford, R. Soto, C. Castro, and F. Paredes, "A k-means binarization framework applied to multidimensional knapsack problem," *Applied Intelligence*, vol. 48, no. 2, pp. 357–380, 2017↑. 276, 278

[6] B. Crawford, R. Soto, N. Berríos, F. Johnson, F. Paredes, C. Castro, and E. Norero, "A binary cat swarm optimization algorithm for the non-unicost set covering problem," *Mathematical Problems in Engineering*, vol. 2015, no. 2, 2015. https://doi.org/10.1155/2015/578541↑. 276, 281

[7] B. Crawford, R. Soto, M. Olivares-Suárez, F. Paredes, and F. Johnson, "Binary firefly algorithm for the set covering problem," in *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–5, june 2014. https://doi.org/10.1109/CISTI.2014.6877090. https://doi.org/10.1007/978-3-319-06740-7_6↑. 276

[8] R. Soto, B. Crawford, R. Olivares, J. Barraza, I. Figueroa, F. Johnson, F. Paredes, and E. Olguín, "Solving the non-unicost set covering problem by using cuckoo search and black hole optimization," *Natural Computing*, pp. 1–17, 2017. https://doi.org/10.1007/s11047-016-9609-7↑. 276

[9] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," tech. rep., Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005↑. 276

[10] Y. Lu and F. J. Vasko, "An or practitioner's solution approach for the set covering problem," *International Journal of Applied Metaheuristic Computing (IJAMC)*, vol. 6, no. 4, pp. 1–13, 2015. https://doi.org/10.4018/IJAMC.2015100101↑. 276

[11] J. García, B. Crawford, R. Soto, and P. García, "A multi dynamic binary black hole algorithm applied to set covering problem," in *Harmony Search Algorithm*, pp. 42–51, Springer, 2017↑. 276

[12] S. Balaji and N. Revathi, "A new approach for solving set covering problem using jumping particle swarm optimization method," *Natural Computing*, vol. 15, no. 3, pp. 503–517, 2016. https://doi.org/10.1007/s11047-015-9509-2↑. 276

[13] B. Crawford, C. Valenzuela, R. Soto, E. Monfroy, and F. Paredes, "Parameter tuning of metaheuristics using metaheuristics," *Advanced Science Letters*, vol. 19, no. 12, pp. 3556–3559, 2013. https://doi.org/10.1166/asl.2013.5236↑. 276

[14] B. Crawford, R. Soto, W. Palma, F. Johnson, F. Paredes, and E. Olguín, "A 2-level approach for the set covering problem: Parameter tuning of artificial bee colony algorithm by using genetic algorithm," in *Advances in Swarm Intelligence* (Y. Tan, Y. Shi, and C. Coello, eds.), vol. 8794, pp. 189–196, Springer International Publishing, 2014↑. 276, 277

[15] B. Crawford, R. Soto, E. Monfroy, G. Astorga, J. García, and E. Cortes, "A meta-optimization approach for covering problems in facility location," in *4th Workshop on Engineering Applications*, pp. 565–578, Springer, 2017. https://doi.org/10.1007/978-3-319-66963-2_50↑. 276

[16] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 3, pp. 21–57, 2014. https://doi.org/10.1007/s10462-012-9328-0↑. 276

[17] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007. https://doi.org/10.1007/s10898-007-9149-x↑. 276

[18] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical Report TR06. Computer Engineering Department, Erciyes University, Turkey*, 2005. 277

[19] A. Singh, "An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem," *Applied Soft Computing*, vol. 9, no. 2, pp. 625–631, 2009. https://doi.org/10.1016/j.asoc.2008.09.001↑. 277

[20] B. Crawford, R. Soto, R. Cuesta, and F. Paredes, "Application of the artificial bee colony algorithm for solving the set covering problem," *The Scientific World Journal*, vol. 2014, no. 1, 2014. https://doi.org/10.1155/2014/189164↑. 277, 281

[21] M. López-Ibánez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Stützle, "The irace package: Iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43 – 58, 2016. https://doi.org/10.1016/j.orp.2016.09.002↑. 278

[22] M. Birattari, "*The problem of tuning metaheuristics as seen from a machine learning perspective*," *Amsterdam: IOS Press*, 2004↑. 278

[23] G. Franceschini and S. Macchietto, "Model-based design of experiments for parameter precision: State of the art," *Chemical Engineering Science*, vol. 63, no. 19, pp. 4846–4872, 2008. https://doi.org/10.1016/j.ces.2007.11.034↑. 278

[24] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on systems, man, and cybernetics*, vol. 16, no. 1, pp. 122–128, 1986. https://doi.org/10.1109/TSMC.1986.289288↑. 279

[25] C. Ansótegui, M. Sellmann, and K. Tierney, "A gender-based genetic algorithm for the automatic configuration of algorithms," in *International Conference on Principles and Practice of Constraint Programming*, pp. 142–157, Springer, 2009. https://doi.org/10.1007/978-3-642-04244-7_14↑. 279

[26] B. Senthilkumar, T. Kannan, and R. Madesh, "Optimization of flux-cored arc welding process parameters by using genetic algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 93, no. 1-4, pp. 35–41, 2017. https://doi.org/10.1007/s00170-015-7636-7↑. 279

[27] B. Crawford, R. Soto, M. Olivares-Suárez, and F. Paredes, "A binary firefly algorithm for the set covering problem," in *3rd Computer Science On-line Conference 2014 (CSOC 2014)*, vol. 285 of *Advances in Intelligent Systems and Computing*, pp. 65–73, Springer International Publishing, 2014. ↑. 281

[28] B. Crawford, R. Soto, C. Peña, W. Palma, F. Johnson, and F. Paredes, "Solving the set covering problem with a shuffled frog leaping algorithm," in *7th Asian Conference, ACIIDS 2015, Bali, march 23-25, 2015*. 281

[29] R. Soto, B. Crawford, A. Muñoz, F. Johnson, and F. Paredes, "Pre-processing, repairing and transfer functions can help binary electromagnetism-like algorithms," in *Artificial Intelligence Perspectives and Applications*, vol. 347 of *Advances in Intelligent Systems and Computing*, pp. 89–97, Springer International Publishing, 2015. https://doi.org/10.1007/978-3-319-18476-0_10↑. 281

[30] R. Cuesta, B. Crawford, R. Soto, and F. Paredes, "An artificial bee colony algorithm for the set covering problem," in *Modern Trends and Techniques in Computer Science:3rd Computer Science On-line Conference 2014 (CSOC 2014)*, Springer International Publishing, 2014. https://doi.org/10.1007/978-3-319-06740-7_5↑. 281

[31] H. Shah-Hosseini, "Intelligent water drops algorithm: A new optimization method for solving the multiple knapsack problem," *International Journal of Intelligent Computing and Cybernetics*, vol. 1, no. 2, pp. 193–212, 2008. https://doi.org/10.1108/17563780810874717↑. 282

[32] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965. https://doi.org/10.2307/2333709.https://doi.org/10.1093/biomet/52.3-4.591↑. 284

[33] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947. https://doi.org/10.1214/aoms/1177730491↑. 284

[34] L. Santana, E. Ramiro, and J. Romero, "A hybrid column generation and clustering approach to the school bus routing problem with time windows," *Ingeniería*, vol. 20, no. 1, pp. 101–117, 2015↑. 285

Broderick Crawford

Currently is adjunct professor at the Pontificia Universidad Católica de Valparaíso, Chile.Informatics Engineering, Universidad Técnica Federico Santa María, Chile; Bachelor of Arts in Engineering Science, Universidad Técnica Federico Santa María, Chile; Master of Business Administration, Universidad de Chile; Doctor in Informatics Engineering, Universidad Técnica Federico Santa María, Chile.

Ricardo Soto

Currently is adjunct professor at the Pontificia Universidad Católica de Valparaíso , Chile. Civil Engineer in Computer Science, Pontificia Universidad Católica de Valparaíso, Chile; Bachelor of Science in Engineering, Pontificia Universidad Católica de Valparaíso, Chile; PhD in Computer Science, Universit de Nantes, France.

Eric Monfroy

Currently working as a Professor in the Department of Science and Technology, France. His research interests includes Digital Sciences . Is serving as an editorial member and reviewer of several international reputed journals and member of many international affiliations. Is has authored of many research articles/books related to Digital Sciences.

Gino Astorga

Currently is Phd Student in Computer Engineering, Pontificia Universidad Católica de Valparaíso, Chile. Computer Engineer, Universidad de Viña del Mar, Chile; Master in Computer Engineering, Pontificia Universidad Católica de Valparaíso, Chile; Master in Sciences of the Computer Engineering, Pontificia Universidad Católica de Valparaíso, Chile.

José García

Currently is Phd Student in Computer Engineering, Pontificia Universidad Católica de Valparaíso, Chile. Biochemical, Universidad de Chile, Chile; PhD in Mathematics, Universidad de Chile,Chile; Master in Computer Engineering, Pontificia Universidad Católica de Valparaíso, Chile; Master in Sciences of the Computer Engineering, Pontificia Universidad Católica de Valparaíso, Chile.

Enrique Cortes

Currently is professor at the Universidad de Playa Ancha, Chile and Phd Student in Computer Engineering. Chemical Engineering, Pontificia Universidad Católica de Valparaíso, Chile; ; Master of Science in Engineering with mention in Chemical Engineering; Master in Computer Engineering, Pontificia Universidad Católica de Valparaíso, Chile; Master in Sciences of the Computer Engineering, Pontificia Universidad Católica de Valparaíso, Chile.