PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

A BINARY BLACK HOLE ALGORITHM TO SOLVE THE SET COVERING PROBLEM

UN ALGORITMO BINARIO INSPIRADO EN HOYOS NEGROS PARA RESOLVER EL PROBLEMA DE LA COBERTURA DE CONJUNTOS

ÁLVARO FERNÁN GÓMEZ RUBIO

TESIS DE GRADO
MAGÍSTER EN INGENIERÍA INFORMÁTICA

AGOSTO 2016

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

A BINARY BLACK HOLE ALGORITHM TO SOLVE THE SET COVERING PROBLEM

UN ALGORITMO BINARIO INSPIRADO EN HOYOS NEGROS PARA RESOLVER EL PROBLEMA DE LA COBERTURA DE CONJUNTOS

ÁLVARO FERNÁN GÓMEZ RUBIO

Profesor Guía: **DR. BRODERICK CRAWFORD**

Programa: **Magíster en Ingeniería Informática**

AGOSTO 2016

Contents

List of Figures

List of Tables

1 Abstract.

The Set Covering Problem (SCP) is one of the most representative combinatorial optimization problems and it has multiple applications in different situations of engineering, sciences and some other disciplines. It aims to find a set of solutions that meet the needs defined in the constraints having lowest possible cost.
In this thesis we use an existing binary algorithm inspired by Binary Black Holes (BBH) to solve multiple instances of the problem with known benchmarks obtained from the OR-library. The presented method emulates the behavior of these celestial bodies using several operators, such as rotation and collapse, to bring good solutions. Additionally, we implemented some improvements in certain operators, as well as added others also inspired by black holes physical behavior, to optimize the exploration during the search to optimum values.

2 Resumen.

El Problema de Cobertura de Conjuntos (SCP por sus siglas en inglés) es uno de los problemas más representativos de la optimización combinatoria, con múltiples aplicaciones en diferentes situaciones de la ingeniería, ciencias y otras disciplinas. Su objetivo es encontrar un conjunto de soluciones que satisfagan las necesidades definidas en las restricciones del problema al menor costo posible.
En esta tesis se utiliza un algoritmo binario inspirado en los agujeros negros (BBH por sus siglas en inglés) para resolver las instancias del problema definidas en la OR-Library. El método presentado emula el comportamiento de estos cuerpos celestes utilizando varios operadores, tales como la rotación y colapso, para encontrar buenas soluciones. Además, se implementaron algunas mejoras en ciertos operadores, así como también algún otro nuevo operador inspirado en el comportamiento físico de los agujeros negros, con el objetivo de optimizar la exploración durante la búsqueda de los valores óptimos.

3 Introduction.

The Set Covering Problem is one of 21 NP-Hard problems [26], representing a variety of optimization strategies in various fields and realities. Since its formulation in the 1970s has been used, for example, in minimization of loss of materials for metallurgical industry [40], preparing crews for urban transportation planning [14], safety and robustness of data networks [8], focus of public policies [18], construction structural calculations [4]. This problem was introduced in 1972 by Karp [29] and it is used to optimize problems of elements locations that provide spatial coverage such as community services [30], telecommunications antennas [15] and others.
The present work uses a strategy based on a binary algorithm inspired by black holes to solve the SCP, developing some operators that allow to implement an analog version of some characteristics of these celestial bodies to support the behavior of the algorithm and to improve the processes of searching for the optimum. This type of algorithm was presented for the first time by Abdolreza Hatamlou in 2012 [22], registering some later publications dealing with some applications and improvements. In this thesis it will be detailed the methodology, developed operators, experimental results and execution parameters and handed out some conclusions about them, before and after implementing the proposed improvements.
Considering a binary based matrix (zero-one) $A = a_{ij}$, of m rows and n columns and a vector c of n components containing the costs assigned to each matrix column, then we can define the SCP such as:

$$\text{Minimize} \sum_{j=1}^{n} c_j x_j \tag{1}$$

Where a:

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

$$x_j \in \{0, 1\}, \forall j \in J = \{1, ..., n\}$$

This ensures that each row is covered by at least one column and that there is a cost associated with it [36].

3.1 SCP Example.

Imagine a floor that is required to work on a series of pipes (represented in red) that are under 20 tiles, rising the least amount possible of them. The diagram of the situation would be as follows.

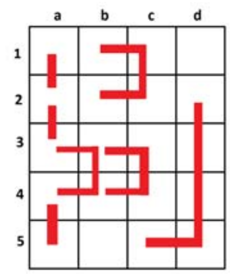

Fig. 1. SCP example

We define a variable X_{ij} which represents each tile by its coordinates in row i and column j. It will store a 1 if it is necessary to lift the back plate and a 0 if it is not. Then we will have:

$$\text{i} \in I = \{1, 2, 3, 4, 5\}\ ,\ \ \text{j} \in J = \{a, b, c, d\}$$

We will define the "objective function" as:

Min z= X_{1a} + X_{2a} + X_{3a} + X_{4a} + X_{5a}+ X_{1b} + X_{2b} + X_{3b} + X_{4b} + X_{5b}+ X_{1c} + X_{2c} + X_{3c} + X_{4c} + X_{5c} + X_{1d} + X_{2d} + X_{3d} + X_{4d} + X_{5d}

With $x_{ij} \in \{0, 1\}, \forall i \in I, \forall j \in J$. Then the following system of equations represent the constraints of the problem:

Table 1. Equations system

$$\begin{array}{ll}
X_{1a} + X_{2a} & \geq 1 \\
X_{2a} + X_{3a} & \geq 1 \\
X_{3a} + X_{3b} + X_{4a} + X_{4b} & \geq 1 \\
X_{4a} + X_{5a} & \geq 1 \\
X_{1b} + X_{1c} + X_{2b} + X_{2c} & \geq 1 \\
X_{3b} + X_{3c} + X_{4b} + X_{4c} & \geq 1 \\
X_{2d} + X_{3d} + X_{4d} + X_{5c} + X_{5d} & \geq 1
\end{array}$$

Then, each constrain corresponds to the tile on top of a main. It is only necessary to pick one per tube. One solution for the system is:

Table 2. Solution

$$\begin{array}{llll}
X_{1a} = 0, & X_{1b} = 1, & X_{1c} = 0, & X_{1d} = 0 \\
X_{2a} = 1, & X_{2b} = 0, & X_{2c} = 0, & X_{2d} = 0 \\
X_{3a} = 0, & X_{3b} = 3, & X_{4b} = 0, & X_{5b} = 0 \\
X_{4a} = 0, & X_{4b} = 0, & X_{4c} = 0, & X_{4d} = 0 \\
X_{5a} = 1, & X_{5b} = 0, & X_{5c} = 1, & X_{5d} = 0
\end{array}$$

Finally, it is only necessary to lift 5 tiles:

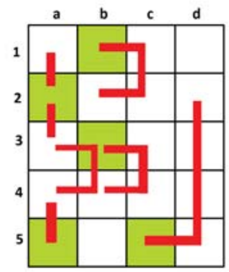

Fig. 2. SCP example solution

This type of strategy has been widely used in aerospace turbine design [43], timetabling design [11], probabilistic queuing [41], geographic analysis [13], services location [42], scheduling [10] and many others [39].
There are variety of algorithms ”bio-inspired” that mimic the behavior of some living beings [12] to solve problems, as well as others who are inspired by elements of nature [39], cultural and other types.
In this thesis we apply a strategy based on a binary algorithm inspired by black holes to solve the SCP, developing some operators that allow to implement an analog version of some characteristics of these celestial bodies to support the behavior of the algorithm and to improve the processes of searching of the optimum.

4 Goals

The general objective is to solve the Set Covering Problem (SCP) using an algorithm inspired by black holes, validating its effectiveness of this through the OR-library benchmarks resolutions. The specific objectives are the following:

- Apply different mechanisms of binarization and discretization and experimentally determine the best performance of different instances of black holes.
- Try with unfesibility adding operators to repair.
- Compair the results with others metaheuristics.

5 Definitions.

Operator: Unitary procedure for transforming information or implement the behavior of the algorithm.
Solution: Array of n columns containing a solution to the equations system of the problem.
Constraints: Conditions to be met by a viable solution.
Benchmark: Optimal set of known problem instances to validate the algorithm.
Objective Function: Implements the mathematical expression representing the cost of a solution.

Fitness: Resulting target value by applying the objective function to a vector.

Matrix of Costs: n columns vector containing the cost associated with each problem variables or columns of vectors.

I-Case: Artificial intelligence software used to generate source code according to a central specification.

Parameter: Initial values for algorithm starts.

Optimal Value: Solution with the best value found for the objective function.

Domain: Set of possible values for the variables.

Transfer function or discretization function: Method for carrying a real number to interval [0,1].

Binarization: Method for converting a decimal number to a binary equivalent.

Matrix A: Matrix containing the constraints for problem.

RPD: Relative Percentage Deviation.

6 Theoretical framework.

6.1 Black holes.

Black holes are the result of the collapse of a big star's mass that after passing through several intermediate stages is transformed in a so massively dense body that manages to bend the surrounding space because of its immense gravity. They are called "black holes" due to even light does not escape their attraction, and therefore is undetectable in the visible spectrum. They are also known as "singularities" because inside the traditional physics can not be applied. As consequence of its immense gravity, they tend to be orbited by other stars in binary or multiple systems consuming a little mass of bodies in its orbit [23]. When a star or any other body is approaching the black hole through what is called "event horizon", it collapses in its interior and it is completely absorbed without any possibility to escape, since all its mass and energy become part of singularity (Fig.3). This is because at that point the exhaust speed is the light one [23].

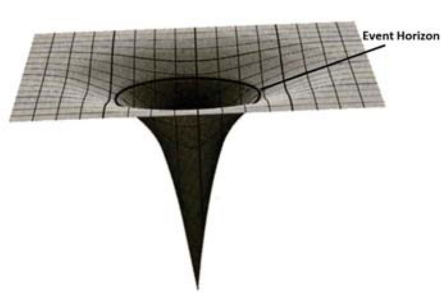

Fig. 3. Event horizon in a black hole

On the other hand, black holes also generate a type of radiation called "Hawking radiation", in honor of its discoverer. This radiation have a quantum origin and it implies transfer of energy from the event horizon of the black hole to its inmediate surroundings, causing a slight loss of mass of the dark body and an emission of additional energy to the nearby objects [24].

6.2 State of the art.

Although the term “algorithm inspired by black hole” was used for the first time in 2008 by Zhang, Liu and Tan [28], only in the 2012 year Abdolreza Hatamlou proposed a methodology and implementation specific of this [22], although only tangentially in a paper of swarm optimization. There are some that use it for optimization of varied elements of engineering, such as electrical circuits [9], problems in the aerospace industry [25] and for various other problems. Moreover , the author has some relevant publications in various conferences and scientific journals [3] [20] [2]. On the other hand, some authors have made some proposals for improvements to the algorithm. It is the case of Nemati et al. presenting the inclusion of gravitational and electric forces as part of the rotation operator [32], the incorporation of elements of fuzzy logic [34] and others suggest improvements in the calculation of distances [16]. It has also received some critics questioning its novelty, rather considering that it is one variation of existing ones [35].

6.3 Original algorithm.

The original algorithm presented by Hatamlou [22] faces the problem of determination of solutions through the development of a set of stars called “universe”, using an ”population” kind algorithm, similar to those used by genetic techniques [21] or particles swarm [37]. It proposes the rotation of the universe around the star that has the best fitness, i.e., which has the lowest value of a defined function, called “objective function”, in order to minimize results.

This rotation is applied by an operator of rotation that moves each one of the stars in each iteration of the algorithm and determines in each cycle if there is a new black hole, that will replace the previous one. In case the vector is not feasible, it is replaced by a new feasible one. The rotation operation is repeated until find the detention criteria, being the last black hole founded the proposed solution and the last of the black holes found corresponds to the final solution proposed.

Eventually, a star can ever exceed the defined by the radius of the event horizon [23]. In this case, the star collapses into the black hole and is removed from the whole universe being taken instead by a new star. Thus, stimulating the exploration of the space of solutions.

The following is the proposed flow chart and the corresponding operators according to the initial version of the method, searching the fitness minimization, according the SCP definition:

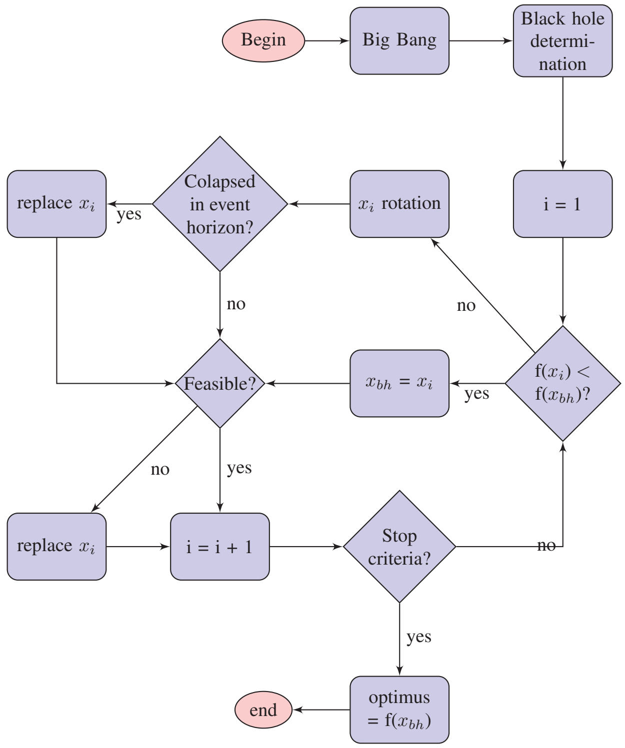

Fig. 4. Original black hole algorithm

6.3.1 Big bang. It consists of the creation of the initial random universe for the algorithm. The number of stars generated will remain fixed during the iterations, although many vectors (or stars) are replaced. The mechanism of creation of vectors is showed in algorithm 1 and also apply in the intermediate steps that require the generation of new stars:

Algorithm 1 Random initial generation of stars

1: $n \leftarrow Cols\ quantity$
2: **for** $row = 1$ to $Stars\ quantity$ **do**
3: $Star_{row}$ = StarGeneration(n)
4: **end for**

Where StarGeneration is the creation of a binary vector of n elements that comply with the constraints of A matrix.

6.3.2 Fitness evaluation. Each star x_j fitness is calculated by evaluating the objective function, according to the initial definition of the problem.

$$\sum_{j=1}^{n} c_j x_j \qquad (2)$$

The Algorithm 2 describes it:

Algorithm 2 Fitness evaluation

$Fitness \leftarrow 0$
for j=1 to Cols **do**
    $Fitness \leftarrow Fitness + x_j c_j$
end for

It should be remembered that c_j corresponds to the cost of j column in the vector of costs and x_j is the jth column of star. In other words, the fitness of a star is the evaluation or objective function for the star vector, considering the vector cost in each column. The black hole will be those who have minor fitness among all existing stars at the time of the evaluation.

6.3.3 Rotation operator.

The rotation operation occurs above all the universe of N stars x_i, with the exception of the black hole, which is fixed in its position. The operation sets the new t+1 position as follows:

$$X_i^d(t+1) = X_i^d(t) + random\ (X_{BH}^d - X_i^d(t)),\ \forall i \in \{1, 2, ..., N\} \tag{3}$$

Where X_i(t) and X_i(t+1) are the positions of the star X_i at t and t+1 iterations respectively, d is the array dimension, x_{BH} is the black hole location in the search space, random is a aleatory number in the range [0,1] and N is the number of stars that make up the universe (candidate solution). It should be noted that the only exception in the rotation is designated as black hole star, which retains the position.

6.3.4 Collapse into the black hole.

When a star is approaching to the black hole at a distance called event horizon is captured and permanently absorbed by it, being replaced by a new randomly generated one. In other words, it is considered when the collapse of a star exceeds the radius of Schawarzchild (R). In a 2015 publication, Farahmandian and Hatamlouy [16] intend to determine the distance of a star x_i to the radius R as:

$$R = \frac{f(x_{bh})}{\sum_{i=1}^{N} f(x_i)} \tag{4}$$

Where $f(x_{bh})$ is the fitness value of the black hole, N is the number of candidate solutions (stars) and $f(x_i)$ is the fitness value of the ith star. I.e. a star x_i will collapse when the star's distance with black hole is less than a defined radius [22].

Additionally, we incorporated an algorithm parameter s, where $s \in [0, 1]$ containing the minimum allowable proximity to the black hole measured on a percentage of its fitness. This, with the aim of managing the tolerance threshold calculating the event horizon. Finally, the star colapse into black hole if:

$$|f(x_{bh}) - f(x_i)| < sR \tag{5}$$

7 Algorithm implementation.

The algorithm implementation was carried out with a I-CASE tool [5], generating Java programs and using a relational database as a repository of the entry information and gathered one during executions. The parameters that will be presented are the result both of the needs of the original design of the algorithm and improvements made product of the tests performed. In particular, it was attempted to improve the capacity of exploration of the metaheuristic. It was contrasted with tables of known optimal values [36], in order to quantitatively estimate the degree of effectiveness of the presented metaheuristics.

The process begins with the random generation of a population of binaries vectors (Stars) in a step that we will call ”big bang”. With a universe of *N* stars formed by vectors of *d* binary digits, the algorithm must identify the star with better fitness value , i.e., the which one the objective function obtains a lower result. The next step is to rotate the other stars around the black hole detected until some other presents a better fitness and take its place.

The number of stars generated will remain fixed during the iterations, notwithstanding that many vectors (or star) will be replaced by one of the operators. The creation mechanism of vectors is listed in algorithm 3 and shall apply also in the intermediate steps that require the generation of new stars:

Algorithm 3 Star generation

```
1: m ← Cols quantity
2: for j = 1 to m do
3:     if random < 0.5 then
4:         Col_i = 0
5:     else
6:         Col_i = 1
7:     end if
8: end for
```

Where Col_i is the ith column of the star vector.

7.1 Discretization and binarization.

As mentioned before, the stars are represented by binary vectors, so there is needed to migrate the real values generated by some functions and operators to binary domain. For this purpose transfer and discretization or binarization functions are used. The transfer function aims to take values from the domain of reals to interval [0,1]. Two representatives functions of two different families was tested, the ”S-Shape” and ”V-Shape” ones [38]. The respective selected functions are:

$$\frac{1}{1+(e^{-x/3})} \tag{6}$$

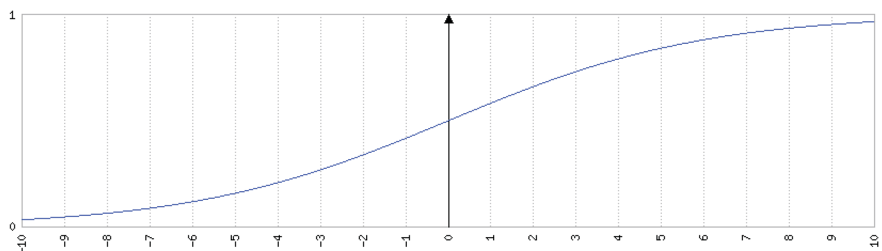

Fig. 5. S-Shape function

$$\left|\frac{x}{\sqrt{1+x^2}}\right| \tag{7}$$

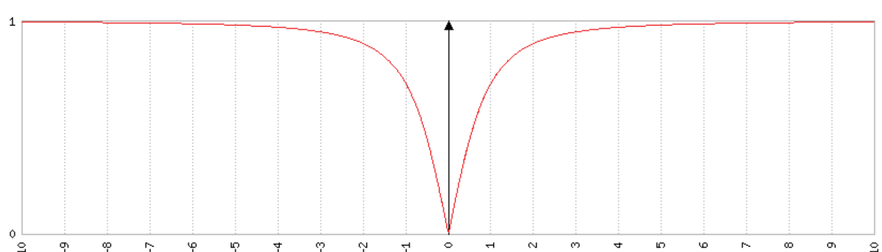

Fig. 6. V-Shape function

The following is a graph comparing both functions:

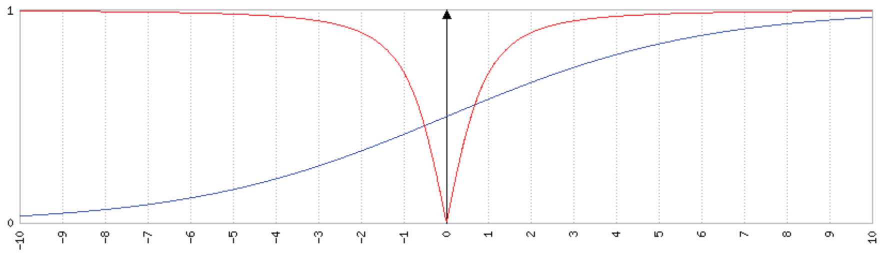

Fig. 7. Comparative graphic

The results using each of the functions can be reviewed in detail in the results section.

In addition, the binarization function is aimed at conveying the value obtained in the previous transformation in a binary digit. Therefore, experiments were performed for the algorithm 4 , the algorithm 5 and the 6 one, where ”random” is a random value between 0 and 1 inclusive. ($random \in [0, 1]$):

Algorithm 4 Standard binarization

---

if random $\leq$ value **then**
    Digit = 1
else
    Digit = 0
end if

Algorithm 5 Reverse binarization

---

if random $\leq$ value **then**
    Digit = 0
else
    Digit = 1
end if

The binarization best results have been achieved with standard binarization to be applied in the subsequent benchmarks.

7.2 Feasibility.

The viability of a star is conditional on compliance with each one of the constraints defined in the matrix A. To determine it was implemented the algorithm 7:

Algorithm 6 Complementary binarization

if random $\leq$ value **then**
  if x_j = 0 **then**
    Digit = 1
  else
    Digit = 0
  end if
else
  Digit = x_j
end if

Algorithm 7 Determining feasibility of a solution

$Feasible = Yes$
$n \leftarrow Matrix\ A\ rows\ quantity$
$c \leftarrow Star\ cols\ quantity$
for $i = 1$ to n **do**
  $Sum \leftarrow 0$
  for $j = 1$ to c **do**
    $Sum \leftarrow Sum + a_{ij}x_j$
  end for
  if Suma = 0 **then**
    $Feasible = No$
  else
    $Feasible = Yes$
  end if
end for

7.3 Parameters.

For the purpose of implementing all the features and operators that are detailed in this document in multiple configurations easily, a table of parameters was built for the algorithm.
The results tables presented in the following sections were obtained with equivalent parameter settings. The main ones are:

- 31 experiments for each configuration. This number of iterations was selected because is the recommended amount for the subsequent statistical analysis.
- 20,000 iterations for each experiment. It corresponds to an amount that keeps the execution times within an acceptable range, without degrading the quality of the results. It was estimated experimentally.
- Transfer function.
- Binarization function.
- R parameter for colapse. The factor was determinated experimentally. We started with a factor of 10% or tolerance and was decreasing in steps of 1%. Finally, it was determined in 5% (0.05).

- Factor for Hawking radiation. The predefined range for the mutation probability is between 10% and 60%, increasing as the iterations are progressing.
- Stars Quantity (size of universe). No improvements were achieved with a number greater than 50 stars, and less than the amount stated causes degradation in the results. Therefore, we worked with a universe of size 50.

8 Proposed improvements.

From the analysis of the experimental results of the original algorithm, estimated that improvements in the performance of the algorithm could be achieved making contributions and changes in some operators, maintaining the general strategy of the algorithm for minimization. The flow chart will be modified to incorporate some of these proposals.

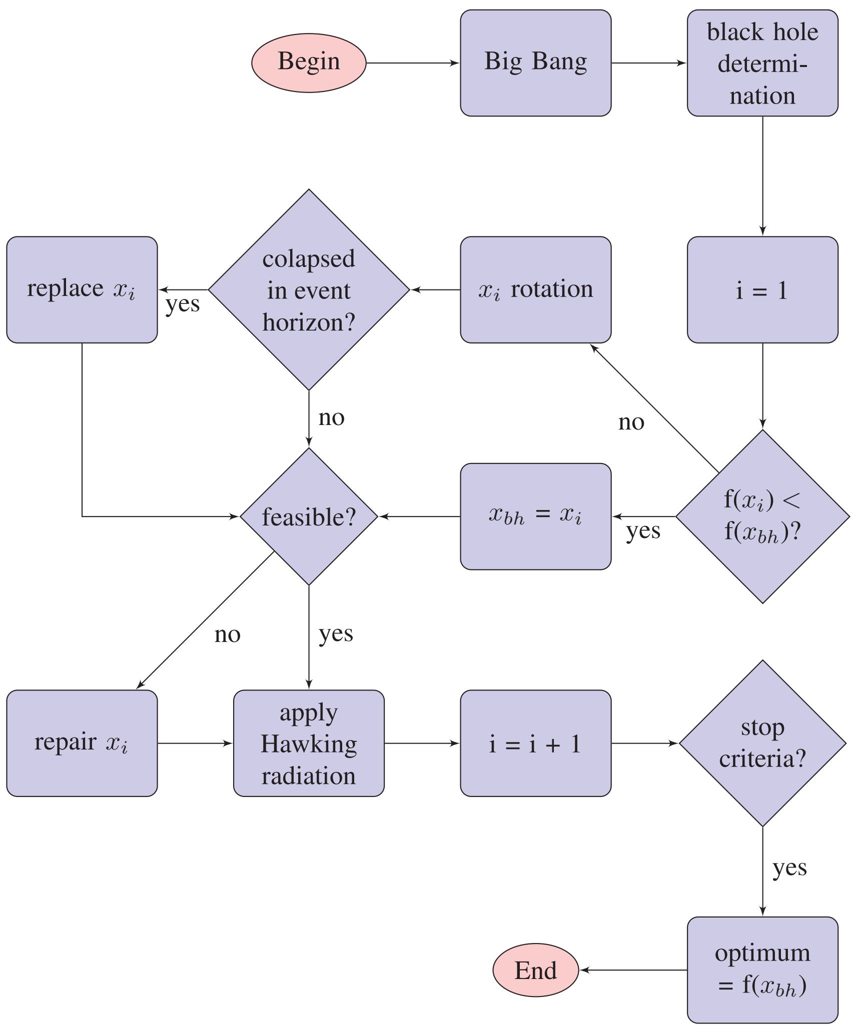

Fig. 8. Modified black hole algorithm

8.1 Repair operator.

In those cases which unfeasibility was detected, we opted for repair of the vector to make it complies with the constraints. We implemented a repair function in two phases, ADD and DROP, as way to optimize the vector in terms of coverage and costs. The first phase changes the vector in the column that provides the coverage at the lowest cost, while the second one removes those columns which only added cost and do not provide coverage. The repair operator was implemented as algorithm 8 [1], where:

- I is the set of all rows
- J is the set of all columns
- J_i is the set of columns that cover the row i, i $\in$ I
- I_j is the set of rows covered by the column j, j $\in$ J
- S is the set of columns associated to solution
- U is the set of columns not covered
- w_i is the number of columns that cover the row i, $\forall$ i $\in$ I in S

Algorithm 8 Repair operator

$w_i \leftarrow |S \bigcap J_i|, \forall i \in I$
$U \leftarrow w_i = 0, \forall i \in I$
for $i \in U$ **do**
    // Find the first column j in J_i that minimize $\frac{c_j}{|U \bigcap I_i|}$
    $S \leftarrow S \bigcap j$
    $w_i \leftarrow w_i + 1, \forall i \in I_j$
    $U \leftarrow U - I_j$
end for
for $j \in S$ **do**
    if $w_i \geq 2, \forall i \in I_j$ **then**
        $S \leftarrow S - j$
        $w_i \leftarrow w_i - 1, \forall i \in I_j$
    end if
end for

8.2 Discretization functions.

A new S-Shape transfer function is proposed to give greater diversity to the process:

$$\frac{1}{1 + (e^{-2x})} \tag{8}$$

Being its graphic:

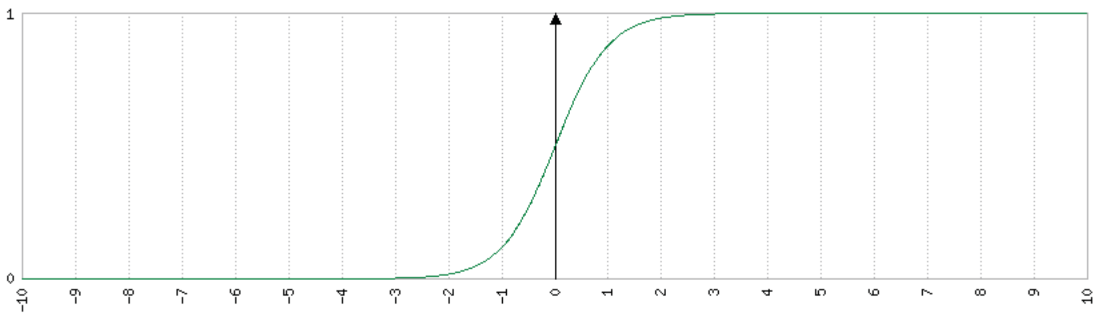

Fig. 9. New suggested transfer functions graphic

On the other hand, remain binarization functions.

8.3 Hawking radiation.

According to the Stephen Hawking theory [24], due to the uncertainty principle it is possible the creation of an anti-particle pair from the void in the exact location of the event horizon. These pairs disintegrate quickly each other returning energy paid for their creation. However, as this takes place in the events of the black hole horizon, the probability of one member of the pair is forming on the inside and the other outside is not null, so one of the components of the pair could escape from the black hole. This will produce a very small and brief radiation, but it is recurrent enough to be losing mass to the singularity. This radiation that affects the surrounding stars was taken as a model to produce some mutations that enhance the exploration of the algorithm. As this physical effect also is probabilistic and more affects some bodies that others due to its proximity, it was decided to implement the algorithm in a progressive roulette, i.e., there is a lower probability of mutation in the first stars but in the final iteration (with better solutions closer to the optimum) likely will be greater, according to a range defined in the parameters. For that, an increase is calculated by iteration, accumulating it in each iteration and gradually increasing the probability of mutation.This improvement is intended to prevent the algorithm locks in local optima. Is relevant notes that while the operator runs on all iterations, only a few iterations mutation is applied, due to the progressive roulette. Then, is established in the algorithm 9 way [31]:

Algorithm 9 Hawking radiation

```
Lower limit ⇐ parameter 1
Upper limit = parameter 2
Iterations max = parameter 3
Increase by iteration : (Upper limit − Lower limit) / (Iteration max)
Actual ⇐ 0
Iteration begin
Actual ⇐ Actual + Increase by iteration
...
if random ≤ Actual then
    for all x_j of star do
        if random ≥ parameter then
            if Bit = 1 then
                Bit ⇐ 0
            end if
            Feasible ⇐ Feasibility evaluation
            if Feasible = No then
                Revert operation
            end if
        end if
    end for
end if
Iteration end
```

9 Experimental results.

The algorithm was subjected to a test by running the benchmark 4, 5, 6, A, B, C, D, NRE, NRF, NRG and NRH from OR library [6]. The RPD column provides a measure of deviation of a isolation to the optimus known, in order to have an assessment of closeness to the best value. Its definition is:

$$RPD = 100\frac{(Z - Z_{opt})}{Z_{opt}} \tag{9}$$

9.1 Original algorithm.

The first results table corresponds to the execution of the original algorithm using the ”S-Shape” transfer function (5):

Table 3. Experimental results for S-Shape (5)

Instance	Z_{BKS}	Z_{min}	Z_{max}	Z_{avg}	RPD	Instance	Z_{BKS}	Z_{min}	Z_{max}	Z_{avg}	RPD
4.1	429	439	598	518.50	2.33	C.1	227	256	291	273.50	12.77
4.2	512	553	664	608.50	8.01	C.2	219	250	291	270.50	14.15
4.3	516	566	698	632.00	9.69	C.3	243	259	298	278.50	6.58
4.4	494	527	777	651,00	6,28	C.4	219	255	312	283.50	16.43
4.5	512	589	649	619.00	15.04	C.5	215	245	287	266.00	13.95
4.6	560	580	596	588.00	3.57	D.1	60	73	139	106.00	21.66
4.7	430	481	593	537.00	11.86	D.2	66	74	110.00	92.00	12.12
4.8	492	502	623	562.50	2.03	D.3	72	91	152	121.50	26.38
4.9	641	689	691	690.00	7,49	D.4	62	77	101	89.00	24.19
4.10	514	565	655	610.00	9.92	D.5	61	79	122	100.50	29.50
5.1	253	284	380	332.00	12.25	E.1	5	10	18	14.00	100.00
5.2	302	342	358	350.00	13.25	E.2	5	15	21	18.00	200.00
5.3	226	251	333	292.00	11.06	E.3	5	8	17	12.50	60.00
5.4	242	273	296	284.50	12.81	E.4	5	8	21	14,50	60.00
5.5	211	232	271	251.50	9.95	E.5	5	9	21	15.00	80.00
5.6	213	249	366	307,50	16.90	NRE1	29	77	123	100.00	165.51
5.7	293	339	368	353.50	15.70	NRE2	30	81	114	97.50	170.00
5.8	288	313	421	367.00	8.68	NRE3	27	82	99	91.00	207.40
5.9	279	311	450	380.50	11.47	NRE4	28	39	74	56.50	39.28
5.10	265	297	407	352.00	12.08	NRE5	28	37	69	53.00	32.14
6.1	138	151	218	185.50	10.86	NRF1	14	34	62	48.00	142.85
6.2	146	159	278	218.50	8.90	NRF2	15	23	77	50.00	53.33
6.3	145	163	239	201.00	12.41	NRF3	14	28	83	55.50	100.00
6.4	131	160	233	196.50	22.13	NRF4	14	39	56	47.50	178.57
6.5	161	187	261	224.00	16.14	NRF5	13	23	102	62.50	76.92
A.1	253	315	389	352.00	24.50	NRG1	176*	220	290	255.00	25.00
A.2	252	303	402	352.50	20.23	NRG2	154*	233	281	257.00	54.30
A.3	232	259	412	335.50	11.63	NRG3	166*	318	422	370.00	91.56
A.4	234	289	364	326.50	23.50	NRG4	168*	285	366	325.50	69.64
A.5	236	271	384	327.50	14.83	NRG5	168*	187	225	206.00	11.30
B.1	69	87	115	101.00	26.08	NRH1	63*	98	155	126.50	55.55
B.2	76	101	107	104.00	32.89	NRH2	63*	99	115	107.00	57.14
B.3	80	98	151	124.50	22.50	NRH3	59*	96	111	103.50	62.71
B4	79	91	137	114,00	15.18	NRH4	58*	99	130	114.50	67.79
B.5	72	92	117	104,50	27.77	NRH5	55*	88	133	108.50	52.72

With the transfer function ”V-Shape” (6) the following results were obtained:

Table 4. Experimental results for V-Shape (6)

Instance	Z_{BKS}	Z_{min}	Z_{max}	Z_{avg}	RPD	Instance	Z_{BKS}	Z_{min}	Z_{max}	Z_{avg}	RPD
4.1	429	435	489	462	1.40	C.1	227	251	287	269.00	10.57
4.2	512	547	601	574.00	6.84	C.2	219	248	289	273.00	13.24
4.3	516	561	677	619.00	8.72	C.3	243	273	399	336.00	12.34
4.4	494	525	594	559.50	6.28	C.4	219	248	301	274.50	13.24
4.5	512	522	635	578.50	1.95	C.5	215	247	295	271.00	14.88
4.6	560	578	666	622.00	3.21	D.1	60	67	157	112.00	15.49
4.7	430	472	495	483.50	9.77	D.2	66	75	177	126.00	13.63
4.8	492	534	615	574.50	8.54	D.3	72	81	120	100.50	12.50
4.9	641	688	767	727.50	7.33	D.4	62	70	135	102.50	12.90
4.10	514	555	674	614.50	7.98	D.5	61	78	120	99.00	27.86
5.1	253	290	388	339.00	14.62	E.1	5	7	18	12.50	40.00
5.2	302	329	330	329.50	8.94	E.2	5	12	61	36.50	140.00
5.3	226	241	278	259.00	6.64	E.3	5	7	10	8.50	40.00
5.4	242	261	287	274.00	7.85	E.4	5	11	76	43.50	120.00
5.5	211	222	270	246.00	5.21	E.5	5	6	10	8.00	20.00
5.6	213	240	351	294.50	7.98	NRE1	29	74	169	121.50	155.17
5.7	293	333	355	344.00	13.65	NRE2	30	97	156	126.50	233.33
5.8	288	310	400	355.00	7.64	NRE3	27	74	152	113.00	174.07
5.9	279	298	400	349.00	6.81	NRE4	28	55	81	68.00	96.42
5.10	265	291	399	345.00	9.81	NRE5	28	39	56	47.50	39.28
6.1	138	149	215	182.00	9.42	NRF1	14	21	38	29.50	50.00
6.2	146	155	198	175.00	6.16	NRF2	15	18	92	55.00	20.00
6.3	145	160	222	191.00	10.31	NRF3	14	17	22	19.50	21.42
6.4	131	156	228	192.00	19.08	NRF4	14	28	74	51.00	100.00
6.5	161	177	278	227.50	9.93	NRF5	13	17	99	58.00	30.76
A.1	253	301	444	372.50	18.97	NRG1	176*	214	333	273.50	21.59
A.2	252	301	412	356.50	19.44	NRG2	154*	215	333	274.00	42.38
A.3	232	254	378	316.50	9.48	NRG3	166*	312	413	323.50	87.95
A.4	234	270	333	301.50	15.38	NRG4	168*	289	398	343,50	72.02
A.5	236	263	397	330.00	11.44	NRG5	168*	211	569	390.00	25.59
B.1	69	93	162	127,50	34.78	NRH1	63*	77	99	88.00	22.22
B.2	76	95	151	123.00	25.00	NRH2	63*	115	197	156.00	82.53
B.3	80	99	146	122,50	23.75	NRH3	59*	213	311	262.00	261.01
B4	79	87	139	113.00	10.12	NRH4	58*	88	122	105.00	49.15
B.5	72	84	124	104.00	16.66	NRH5	55*	78	119	98.50	41.81

* = Best results found in literature [19]

Where Z_{BKS} is the optimal for instance, Z_{min} is the minimum value found, Z_{max} is the maximum value found , Z_{avg} is the average value and RPD is the percentage of deviation from the optimum.

The detailed results can be reviewed in the Appendices section.

9.2 Proposed algorithm.

After implementing the proposed improvements and configure the algorithm to consider the new transfer function and the parameterization of new operators, the following results were obtained:

Table 5. Experimental results for proposed function (9) and improvements

Instance	Z_{BKS}	Z_{min}	Z_{max}	Z_{avg}	RPD	Instance	Z_{BKS}	Z_{min}	Z_{max}	Z_{avg}	RPD
4.1	429	432	491	461.50	0.70	C.1	227	252	287	269.50	9.92
4.2	512	512	589	550.50	0.00	C.2	219	245	289	267.00	10.61
4.3	516	522	606	564.00	1.16	C.3	243	266	399	332.50	8.65
4.4	494	514	609	561.50	4.05	C.4	219	252	301	276.50	13.10
4.5	512	520	635	577.50	1.56	C.5	215	232	250	241.00	7.90
4.6	560	566	674	620.00	1.07	D.1	60	71	146	108.50	15.49
4.7	430	455	501	478.00	5.81	D.2	66	73	177	125.00	9.59
4.8	492	499	593	546.00	1.42	D.3	72	80	120	100.00	11.11
4.9	641	667	689	678.00	4.06	D.4	62	70	135	102.50	11.43
4.10	514	529	612	570.50	2.92	D.5	61	72	208	140.00	15.28
5.1	253	257	273	265.00	1.58	E.1	5	6	10	8.00	20.00
5.2	302	302	329	315.50	0.00	E.2	5	8	12	10.00	60.00
5.3	226	231	288	259.50	2.21	E.3	5	9	20	14.50	80.00
5.4	242	249	261	255.00	2.89	E.4	5	7	18	12.50	40.00
5.5	211	225	258	241.50	6.64	E.5	5	13	71	42.00	61.54
5.6	213	230	359	294.50	7.98	NRE1	29	72	93	82.50	148.27
5.7	293	314	372	343.00	7.17	NRE2	30	64	88	76.00	113.33
5.8	288	308	383	345.50	6.94	NRE3	27	59	77	68.00	118.51
5.9	279	296	391	343.50	6.09	NRE4	28	49	101	75.00	28.00
5.10	265	283	412	347.50	6.79	NRE5	28	33	65	49.00	17.85
6.1	138	146	201	173.50	5.80	NRF1	14	37	87	164.29	164.29
6.2	146	157	281	219.00	7.53	NRF2	15	18	54	36.00	20.00
6.3	145	153	195	175.50	7.59	NRF3	14	27	73	82.86	92.86
6.4	131	144	233	188.50	9.92	NRF4	14	29	41	35.00	107.14
6.5	161	177	258	217.50	9.94	NRF5	13	22	32	27.00	69.23
A.1	253	298	414	356.00	17.79	NRG1	176*	200	301	257.50	13.63
A.2	252	301	430	365.50	19.44	NRG2	151*	197	255	226.00	30.46
A.3	232	256	316	286.00	10.34	NRG3	166*	234	301	267.50	40.96
A.4	234	268	316	292.00	14.53	NRG4	168*	273	367	328.00	62.5
A.5	236	266	369	317.50	12.71	NRG5	168*	202	302	256.50	20.23
B.1	69	82	149	115.50	18.84	NRH1	63*	111	203	157.00	76.19
B.2	76	99	184	133.50	30.26	NRH2	63*	97	168	132.50	53.97
B.3	80	89	145	117.00	11.25	NRH3	59*	93	105	99.00	57.62
B4	79	88	104	96.00	11.39	NRH4	59*	96	176	136.00	65.52
B.5	72	88	119	99.50	22.22	NRH5	55*	89	182	135.50	61.82

* = Best results found in literature [19]

Where Z_{BKS} is the optimal for instance, Z_{min} is the minimum value found, Z_{max} is the maximum value found , Z_{avg} is the average value and RPD is the percentage of deviation from the optimum.

10 Analisys and conclusions.

In the following appendix is detailed the comparative analysis of all results, in order to have a global view of the performance for each algorithm settings.

10.1 Comparison of results.

For purposes of determining the effectiveness of the proposed improvements, the minimum obtained by each configuration, for each set of instances of the benchmark, were compared the average deviations of each ones respect to known optimals.

The following table presents the average RPD for each set of instances, in order to be able to determine easily configurations closer to known optimum, as well the instances that the algorithm processed better. The last line shows the average global RPD for all executions in order to assess the overall performance of each configuration relative to each other.

Table 6. RPD comparison

Instance	**S-Shape (5) average RPD**	**V-Shape (6) average RPD**	**Proposed (10) average RPD**	**Global average RPD**
SCP 4	7,66	26,27	2,28	12,07
SP 5	12,41	9,42	4,83	8,89
SP 6	13,80	10,70	7,74	10,75
A	18,94	14,94	14,96	16,28
B	24,89	22,07	18,79	21,92
C	12,78	12,86	11,07	12,23
D	22,78	15,72	14,20	17,56
E	100,00	72,00	72,00	81,33
NRE	122,13	137,66	94,60	118,13
NRF	110,34	44,44	90,70	81,83
NRG	49,76	49,35	33,05	44,06
NRH	61,22	91,86	63,02	72,04
TOTAL	**46,39**	**42,26**	**35,60**	**41,42**

Comparing the results of experiments with the best reported in the literature [7], we can see that results are acceptably closed to the best known fitness for 4 and 5 benchmarks, poor for 6, A, B, C and D ones and far away from them in the case of final ones. It is relevant to the case of series 4 and 5, which reached the optimum in a couple of instances. In the case of the first ones are deviations between 0% and 7,98%, while in the case of the last ones reach 118,51% of deviation.

Additionally, it can be noted that the RPD average are low for the first two sets, indicating a relatively consistent behavior in the search for the optimum. This occurs in the three configurations, so it can be inferred that the improvements in performance for improved algorithm is precisely because changes in operators and transfer function, since all the other aspects are equal in all executions.

10.2 Statistical Analysis.

Lanza and Gomez [27] proposed a method to compare different algorithms from point of view of its results regularity and consistency, we performed a statistical analysis to determine the quality of different versions of the algorithm.
In this case, the analysis was performed in two versions that showed the best results, ie, the second experiment (we'll call "v-shape") and the version with the proposed improvements (we'll call "proposed"). The previous thing to do is to determine the outliers may exist in the results of the different instances. They are determined by calculating the median for each 31 iterations, for each instance and for each version of the algorithm. Then, calculate the first and third quartiles (Q1 and Q3) and the interquartile range (IQR). This was done with a spreadsheet, so to determine the outliers according to the following conditions:

$$Mild\ outlier : value < Q_1 - (1,5\ IQR)\ or > Q_3 + (1,5\ IQR) \tag{10}$$

$$Extreme\ outlier : value < Q_1 - (3\ IQR)\ or > Q_3 + (3\ IQR) \tag{11}$$

No extreme outliers were found, but only a few milds, which were removed from the list prior to the next analysis.
The following diagram represents the method to be followed, depending on the nature of obtained data:

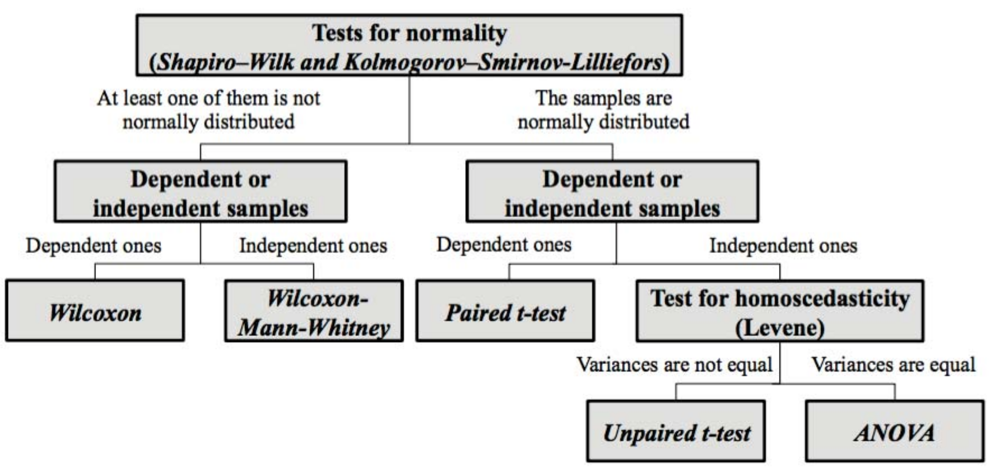

Fig. 10. Statistical Methodology Chart

Under the proposed method, the first thing is to determine the normality for each instance data. For this analysis the Shapiro-Wilk and Kolmogorov-Smirnov-Lilliefors tests were applied to the data sets, using R software.
In order to determine if data follow a normal distribution, it was taken into account the following two hypothesis:

$$H_0\text{: If p-value}>0.05\text{, then data follow a normal distribution.}$$

$$H_1\text{: Otherwise; if p-value}<0.05\text{, so it cannot assume } H_0.$$

The following table presents the results:

Table 7. Normality test results

Instance	V-Shape	Proposed	Test	Final Normal Distribution
4.1	NO	NO	Shapiro-Wilk	YES
4.2	NO	YES	Shapiro-Wilk	YES
4.3	NO	YES	Shapiro-Wilk	YES
4.4	YES	YES	ALL	NO
4.5	YES	YES	ALL	NO
4.6	YES	NO	Shapiro-Wilk	YES
4.7	NO	YES	Shapiro-Wilk	YES
4.8	YES	YES	ALL	NO
4.9	YES	YES	ALL	NO
4.10	NO	YES	Shapiro-Wilk	YES
5.1	YES	NO	Shapiro-Wilk	YES
5.2	YES	YES	Shapiro-Wilk	YES
5.3	YES	NO	Shapiro-Wilk	YES
5.4	NO	YES	Shapiro-Wilk	YES
5.5	NO	YES	Shapiro-Wilk	YES
5.6	YES	NO	Shapiro-Wilk	YES
5.7	YES	NO	Shapiro-Wilk	YES
5.8	NO	YES	Shapiro-Wilk	YES
5.9	NO	NO	Shapiro-Wilk	YES
5.10	YES	NO	Shapiro-Wilk	YES
6.1	NO	YES	Shapiro-Wilk	YES
6.2	YES	YES	ALL	NO
6.3	NO	NO	Shapiro-Wilk	YES
6.4	NO	NO	Lilliefors	YES
6.5	NO	NO	Shapiro-Wilk	YES
A.1	YES	NO	Shapiro-Wilk	YES
A.2	YES	YES	ALL	NO
A.3	YES	YES	ALL	NO
A.4	YES	NO	Shapiro-Wilk	YES
A.5	NO	NO	Shapiro-Wilk	YES
B.1	YES	YES	ALL	NO
B.2	NO	NO	Shapiro-Wilk	YES
B.3	YES	YES	ALL	NO
B.4	YES	YES	ALL	NO
B.5	YES	YES	ALL	NO
C.1	YES	NO	Shapiro-Wilk	YES
C.2	YES	YES	ALL	NO
C.3	NO	YES	Shapiro-Wilk	YES
C.4	NO	NO	Shapiro-Wilk	YES
C.5	YES	YES	ALL	NO
D.1	NO	NO	Shapiro-Wilk	YES
D.2	NO	YES	Shapiro-Wilk	YES
D.3	YES	NO	Shapiro-Wilk	YES
D.4	YES	YES	ALL	NO
D.5	NO	YES	Shapiro-Wilk	YES
E.1	YES	YES	ALL	NO
E.2	NO	YES	Shapiro-Wilk	YES
E.3	YES	YES	ALL	NO
E.4	YES	YES	ALL	NO
E.5	NO	YES	Shapiro-Wilk	YES
NRE.1	NO	YES	Shapiro-Wilk	YES
NRE.2	YES	NO	Shapiro-Wilk	YES
NRE.3	YES	YES	ALL	NO
NRE.4	YES	NO	Shapiro-Wilk	YES
NRE.5	YES	YES	ALL	NO
NRF.1	NO	NO	Shapiro-Wilk	YES
NRF.2	YES	NO	Shapiro-Wilk	YES
NRF.3	NO	YES	Shapiro-Wilk	YES
NRF.4	YES	YES	ALL	NO
NRF.5	NO	YES	Shapiro-Wilk	YES
NRG.1	NO	YES	Shapiro-Wilk	YES
NRG.2	NO	NO	Shapiro-Wilk	YES
NRG.3	NO	YES	Shapiro-Wilk	YES
NRG.4	NO	YES	Shapiro-Wilk	YES
NRG.5	YES	YES	Shapiro-Wilk	YES
NRH.1	YES	YES	ALL	NO
NRH.2	YES	YES	ALL	NO
NRH.3	YES	YES	ALL	NO
NRH.4	YES	YES	ALL	NO
NRH.5	YES	NO	Shapiro-Wilk	YES

The above table shows the result of normality test for each instance, for both algorithm versions, type of test used and the final result to decide on the way forward with the method. For methodological purposes, it is deemed to follow normal distribution when both versions of the algorithm presented that distribution. Otherwise, the entire sample does not.

Next according to the methodology, is validate if the data sets are matched. For this, the ”runs.test” test in sofware R was applied. The results obtained are as follows:

Table 8. Runs.test results for match test

Instance	V-Shape	Proposed	Data Set Is Matched?
4.1	NO	NO	NO
4.2	NO	YES	NO
4.3	NO	NO	NO
4.4	NO	NO	NO
4.5	NO	NO	NO
4.6	NO	NO	NO
4.7	YES	NO	NO
4.8	NO	NO	NO
4.9	YES	NO	NO
4.10	NO	NO	NO
5.1	NO	NO	NO
5.2	NO	NO	NO
5.3	NO	NO	NO
5.4	YES	NO	NO
5.5	NO	NO	NO
5.6	NO	NO	NO
5.7	NO	NO	NO
5.8	NO	NO	NO
5.9	NO	NO	NO
5.10	NO	NO	NO
6.1	NO	NO	NO
6.2	NO	NO	NO
6.3	YES	NO	NO
6.4	NO	YES	NO
6.5	NO	NO	NO
A.1	NO	NO	NO
A.2	NO	NO	NO
A.3	NO	NO	NO
A.4	NO	NO	NO
A.5	NO	NO	NO
B.1	NO	YES	NO
B.2	NO	NO	NO
B.3	NO	NO	NO
B.4	NO	NO	NO
B.5	NO	NO	NO
C.1	NO	NO	NO
C.2	NO	NO	NO
C.3	YES	NO	NO
C.4	NO	NO	NO
C.5	NO	NO	NO
D.1	NO	NO	NO
D.2	NO	NO	NO
D.3	NO	NO	NO
D.4	NO	NO	NO
D.5	NO	YES	NO
E.1	NO	NO	NO
E.2	NO	NO	NO
E.3	NO	NO	NO
E.4	NO	NO	NO
E.5	NO	NO	NO
NRE.1	NO	NO	NO
NRE.2	NO	NO	NO
NRE.3	NO	YES	NO
NRE.4	NO	NO	NO
NRE.5	NO	YES	NO
NRF.1	NO	NO	NO
NRF.2	NO	NO	NO
NRF.3	NO	NO	NO
NRF.4	NO	NO	NO
NRF.5	NO	NO	NO
NRG.1	NO	NO	NO
NRG.2	NO	NO	NO
NRG.3	NO	NO	NO
NRG.4	NO	NO	NO
NRG.5	NO	NO	NO
NRH.1	NO	NO	NO
NRH.2	NO	NO	NO
NRH.3	NO	NO	NO
NRH.4	NO	NO	NO
NRH.5	NO	NO	NO

From the previous table we can infer that the data are not matched. So, in the following step will apply the Wilcoxon-Mann-Whitney test or the Unpaired test, depending of kind of distribution, using R software. The following table shows which test will be apply in each instance, according the previous results.

Table 9. Summary of results and test to apply

Instance	Normal Distribution	Matched	Test to Apply	Instance	Normal Distribution	Matched	Test to Apply
4.1	NO	NO	Wilcoxon-Mann-Whitney	D.1	NO	NO	Wilcoxon-Mann-Whitney
4.2	NO	NO	Wilcoxon-Mann-Whitney	D.2	NO	NO	Wilcoxon-Mann-Whitney
4.3	NO	NO	Wilcoxon-Mann-Whitney	D.3	NO	NO	Wilcoxon-Mann-Whitney
4.4	YES	NO	Unpaired t-test	D.4	YES	NO	Unpairedt-test
4.5	YES	NO	UnpairedUnpaired t-test	D.5	NO	NO	Wilcoxon-Mann-Whitney
4.6	NO	NO	Wilcoxon-Mann-Whitney	E.1	YES	NO	Unpairedt-test
4.7	NO	NO	Wilcoxon-Mann-Whitney	E.2	NO	NO	Wilcoxon-Mann-Whitney
4.8	YES	NO	Unpairedt-test	E.3	YES	NO	Unpairedt-test
4.9	YES	NO	Unpaired t-test	E.4	YES	NO	Unpairedt-test
4.10	NO	NO	Wilcoxon-Mann-Whitney	E.5	NO	NO	Wilcoxon-Mann-Whitney
5.1	NO	NO	Wilcoxon-Mann-Whitney	NRE.1	NO	NO	Wilcoxon-Mann-Whitney
5.2	NO	NO	Wilcoxon-Mann-Whitney	NRE.2	NO	NO	Wilcoxon-Mann-Whitney
5.3	NO	NO	Wilcoxon-Mann-Whitney	NRE.3	YES	NO	Unpairedt-test
5.4	NO	NO	Wilcoxon-Mann-Whitney	NRE.4	NO	NO	Wilcoxon-Mann-Whitney
5.5	NO	NO	Wilcoxon-Mann-Whitney	NRE.5	YES	NO	Unpairedt-test
5.6	NO	NO	Wilcoxon-Mann-Whitney	NRF.1	NO	NO	Wilcoxon-Mann-Whitney
5.7	NO	NO	Wilcoxon-Mann-Whitney	NRF.2	NO	NO	Wilcoxon-Mann-Whitney
5.8	NO	NO	Wilcoxon-Mann-Whitney	NRF.3	NO	NO	Wilcoxon-Mann-Whitney
5.9	NO	NO	Wilcoxon-Mann-Whitney	NRF.4	YES	NO	Unpairedt-test
5.10	NO	NO	Wilcoxon-Mann-Whitney	NRF.5	NO	NO	Wilcoxon-Mann-Whitney
6.1	NO	NO	Wilcoxon-Mann-Whitney	NRG.1	NO	NO	Wilcoxon-Mann-Whitney
6.2	YES	NO	Unpairedt-test	NRG.2	YES	NO	Unpairedt-test
6.3	NO	NO	Wilcoxon-Mann-Whitney	NRG.3	YES	NO	Unpairedt-test
6.4	NO	NO	Wilcoxon-Mann-Whitney	NRG.4	YES	NO	Unpaired t-test
6.5	NO	NO	Wilcoxon-Mann-Whitney	NRG.5	YES	NO	Unpaired t-test
A.1	NO	NO	Wilcoxon-Mann-Whitney	NRH.1	NO	NO	Wilcoxon-Mann-Whitney
A.2	YES	NO	Unpaired t-test	NRH.2	NO	NO	Wilcoxon-Mann-Whitney
A.3	YES	NO	Unpaired t-test	NRH.3	NO	NO	Wilcoxon-Mann-Whitney
A.4	NO	NO	Wilcoxon-Mann-Whitney	NRH.4	NO	NO	Wilcoxon-Mann-Whitney
A.5	NO	NO	Wilcoxon-Mann-Whitney	NRH.5	YES	NO	Unpaired t-test
B.1	YES	NO	Unpaired t-test				
B.2	NO	NO	Wilcoxon-Mann-Whitney				
B.3	YES	NO	Unpaired t-test				
B.4	YES	NO	Unpaired t-test				
B.5	YES	NO	Unpaired t-test				
C.1	NO	NO	Wilcoxon-Mann-Whitney				
C.2	YES	NO	Unpaired t-test				
C.3	NO	NO	Wilcoxon-Mann-Whitney				
C.4	NO	NO	Wilcoxon-Mann-Whitney				
C.5	YES	NO	Unpaired t-test				

Next, to apply the final test we must consider two hypotheses:

$$H_0\text{: If p-value} > 0.05\text{, then } \bar{X} \text{ v-shape version} \leqslant \bar{X}\text{proposed version.}$$
$$H_1\text{: If p-value} < 0.05\text{, } \bar{X} \text{ v-shape version} > \bar{X}\text{proposed version. So, it cannot assume } H_0.$$

And complementarily:

$$H_0\text{: If p-value} > 0.05\text{, then } \bar{X} \text{ proposed version} \leqslant \bar{X}\text{v-shape version.}$$
$$H_1\text{: if p-value} < 0.05\text{, } \bar{X} \text{ proposed version} > \bar{X}\text{v-shape version. So, it cannot assume } H_0.$$

Where $\bar{X}$ is the arithmetic median of fitness values. If the p-value is less than 0.05 then the H_0 is accepted and it is assumed that the proposed version of the algorithm is better than the v-shape. Otherwise, it is assumed v-shape version delivers better results. In the appendices section you can consult the detailed results of the tests. The following table shows the version of algorithm selected as better for each instance:

Table 10. Selected versions

Instance	Selected	Instance	Selected	Instance	Selected
4.1	No diference	A.1	No diference	NRE.1	No diference
4.2	V-shape	A.2	No diference	NRE.2	Proposed
4.3	Proposed	A.3	No diference	NRE.3	Proposed
4.4	Proposed	A.4	No diference	NRE.4	Proposed
4.5	Proposed	A.5	V-shape	NRE.5	No diference
4.6	Proposed	B.1	Proposed	NRF.1	V-shape
4.7	Proposed	B.2	No diference	NRF.2	No diference
4.8	Proposed	B.3	V-shape	NRF.3	V-shape
4.9	No diference	B.4	No diference	NRF.4	No diference
4.10	Proposed	B.5	No diference	NRF.5	V-shape
5.1	V-shape	C.1	No diference	NRG.1	Proposed
5.2	Proposed	C.2	No diference	NRG.2	Proposed
5.3	Proposed	C.3	Proposed	NRG.3	Proposed
5.4	Proposed	C.4	No diference	NRG.4	Proposed
5.5	No diference	C.5	Proposed	NRG.5	V-shape
5.6	Proposed	D.1	No diference	NRH.1	Proposed
5.7	V-shape	D.2	No diference	NRH.2	No diference
5.8	No diference	D.3	No diference	NRH.3	No diference
5.9	No diference	D.4	No diference	NRH.4	No diference
5.10	Proposed	D.5	No diference	NRH.5	V-shape
6.1	No diference	E.1	No diference		
6.2	No diference	E.2	No diference		
6.3	Proposed	E.3	No diference		
6.4	Proposed	E.4	No diference		
6.5	No diference	E.5	V-shape		

The previous table shows the results for both algorithms, showing for most instances there are not difference between the two versions of the algorithms (48%) and finding a better performance for the proposed 25% of the proposed algorithm, as is showed in next chart:

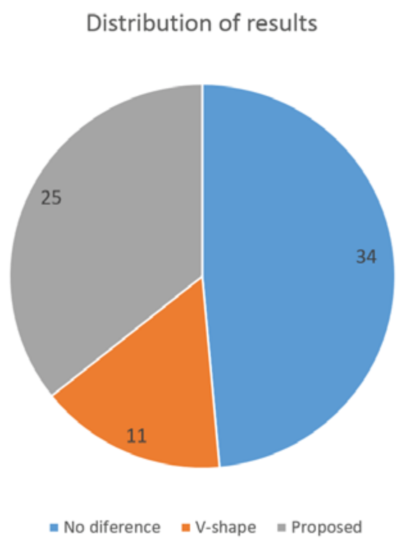

Fig. 11. Distribution of selected versions

In order to validate if there is any relationship between the results obtained by the algorithms and the algorithm selected by the statistical analysis, Pearson correlation index was calculated for the difference between the found minimums and selected algorithm. Thus, the difference was defined as:

$$Difference = v - shape\ version\ min - proposed\ version\ min \tag{12}$$

And the comparative table is:

Table 11. Comparative table

Instance	Selected	Difference	Instance	Selected	Difference	Instance	Selected	Difference	Instance	Selected	Difference
4.1	No diference	0	6.1	No diference	3	D.1	No diference	-4	NRG.1	Proposed	14
4.2	V-shape	35	6.2	No diference	-2	D.2	No diference	2	NRG.2	Proposed	18
4.3	Proposed	39	6.3	Proposed	7	D.3	No diference	1	NRG.3	Proposed	78
4.4	Proposed	11	6.4	Proposed	12	D.4	No diference	0	NRG.4	Proposed	16
4.5	Proposed	2	6.5	No diference	0	D.5	No diference	6	NRG.5	V-shape	9
4.6	Proposed	12	A.1	No diference	3	E.1	No diference	1	NRH.1	Proposed	-34
4.7	Proposed	17	A.2	No diference	0	E.2	No diference	4	NRH.2	No diference	18
4.8	Proposed	35	A.3	No diference	-2	E.3	No diference	-2	NRH.3	No diference	120
4.9	No diference	21	A.4	No diference	2	E.4	No diference	4	NRH.4	No diference	8
4.10	Proposed	26	A.5	V-shape	-3	E.5	V-shape	-7	NRH.5	V-shape	-11
5.1	V-shape	33	B.1	Proposed	11	NRE.1	No diference	2			
5.2	Proposed	28	B.2	No diference	-4	NRE.2	Proposed	33			
5.3	Proposed	10	B.3	V-shape	10	NRE.3	Proposed	15			
5.4	Proposed	12	B.4	No diference	-1	NRE.4	Proposed	6			
5.5	No diference	-3	B.5	No diference	-4	NRE.5	No diference	6			
5.6	Proposed	10	C.1	No diference	-1	NRF.1	V-shape	-16			
5.7	V-shape	19	C.2	No diference	3	NRF.2	No diference	0			
5.8	No diference	2	C.3	Proposed	7	NRF.3	V-shape	-10			
5.9	No diference	2	C.4	No diference	-4	NRF.4	No diference	-1			
5.10	Proposed	8	C.5	Proposed	15	NRF.5	V-shape	-5			

being the Pearson correlation coefficient 0.24, indicating there is no relevant correlation between the two columns.

10.3 Conclusions.

Although local optima found in last benchmarks are not good, they still show a better performance in the case of the improved algorithm, suggesting they can still achieve better results by additional debbuging and search for further improvements in operators and functions, as well as their execution with a higher number of iterations.

In all cases 31 times algorithm is executed, considering 20.000 iterations in each one. The rapid initial convergence is achieved reducing the value of the objective function significantly during early iterations, being much more gradual in subsequent ones and requiring the execution of those operators that stimulate exploration, such as collapse and Hawking radiation. This suggests that the algorithm has a tendency to fall in optimal locations, where cannot leave without the help of scanning components. In order to illustrate these trends, some graphs are presented showing the evolution of the algorithm during the iterations for different benchmarks, i.e., the successive fitness for ”black holes” in each of the 20,000 iterations:

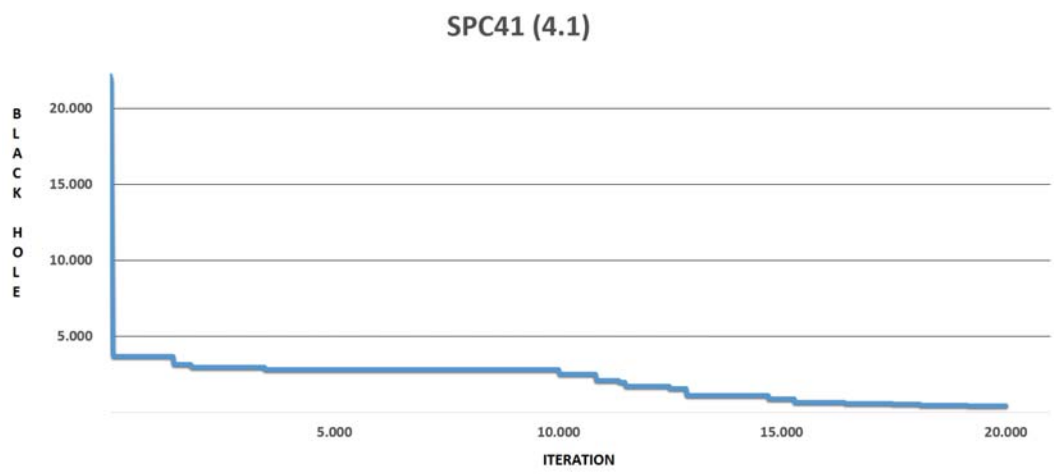

Fig. 12. SCP41 iterations

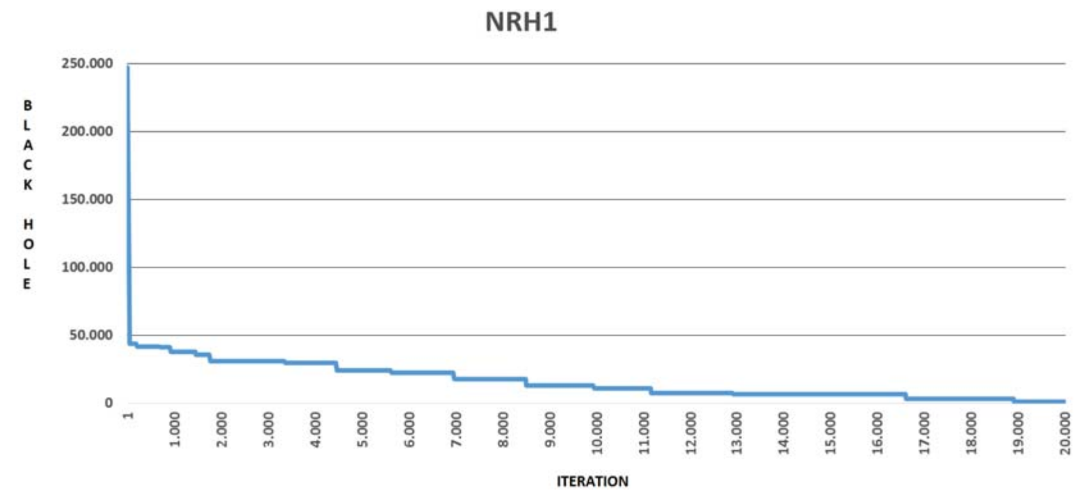

Fig. 13. NRH1 results

While the results for many of the benchmarks were poor due to have a high RPD, i.e., they are significant percentage deviation from known optimum, in absolute terms, considering the starting values in the search, the differences between the known global optimum and local optimum found are low. It is probably because these tests require a greater number of iterations than those made to improve performance, since the values clearly indicate a consistent downward trend, the number of variables is higher and the difference between the optimum and the start values is broader. An interesting analysis element is that the gap between the best and the worst outcome is small and relatively constant in practically all benchmarks, indicating the algorithm tends continuously towards an improvement of results, and the minimums are not just a product of suitable random values. The following chart explains this element:

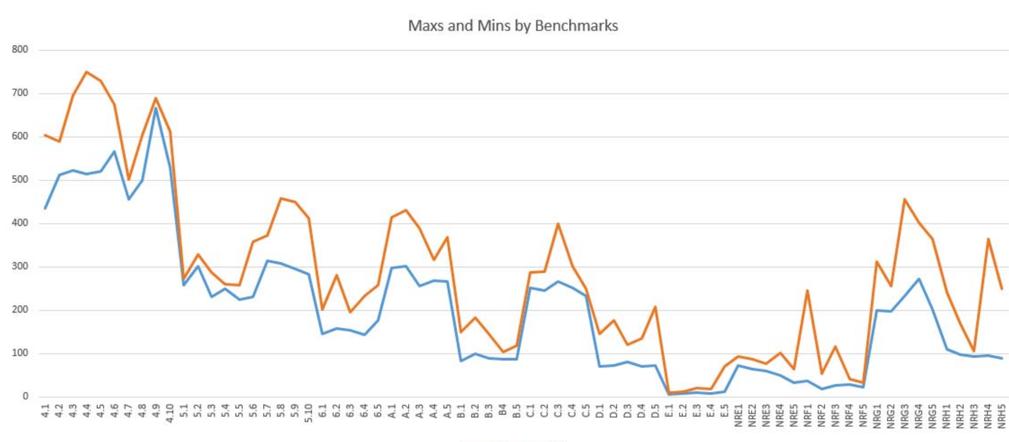

Fig. 14. Evolution of maxs and mins

On the other hand, the introduction of Hawking radiation increased the stochastic component of the algorithm stimulating exploration, which was highlighted in the best results. Additionally, the repair unfeasible vectors means a balance regarding the exploitation of the results, preventing the algorithm scroll through the results space without control.

Other notorious element are the large differences in results obtained with different methods of transfer and binarization, some ones simply conspired against acceptable results. Some future research could be related to better manage of distance and also the implementation of differentiated configuration parameters for different benchmark instances, ie, specific tunning according to process banchmark. Traditional methods work in the domain of real and are not directly applicable to a binary algorithm, forcing perform transformations that could degrade the performance of this. Techniques for binary distances, like Hamming [17] and others offer as an alternative, but can also distort the results because it have a logic unrelated to the context in which it is used. It would also be interesting the introduction of mutation operators to stimulate the elitist exploration or solutions.

Additionally, some authors [33] suggest implementing the rotation operator by adding elements such as mass and electric charge of the black hole, items not considered in this work.

About results of statistical analysis, we can see that improvements in the results obtained do not regularly match the improvements made in regularly, because instances in each ones are not always the same. Thus, we can conclude that the proposed improved algorithm version has a regularly higher than the original algorithm, but such improvements do not have a direct correlation with the closeness of the results obtained respect to known optimum.

Appendices

11 Comparative results.

The results obtained by the different configurations, on a comparative basis are presented in detail:

Table 12. comparison for SCP4 instances

Instances	Optimum	Proposed	S-Shape (5)	V-Shape (6)
4.1	429	435	439	435
4.2	512	512	553	547
4.3	516	522	566	561
4.4	494	514	527	525
4.5	512	520	589	522
4.6	560	566	580	578
4.7	430	455	481	472
4.8	492	499	502	534
4.9	641	667	689	688
4.10	514	529	565	555

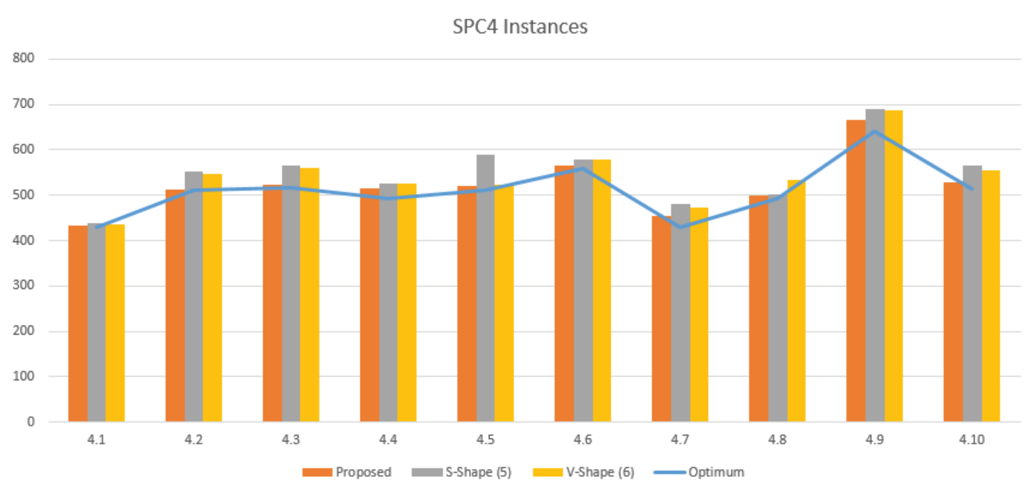

Fig. 15. SCP4 comparison

Table 13. comparison for SCP5 instances

Instances	Optimum	Proposed	S-Shape (5)	V-Shape (6)
5.1	253	257	284	290
5.2	302	302	342	330
5.3	226	231	251	241
5.4	242	249	273	261
5.5	211	225	232	222
5.6	213	230	249	240
5.7	293	314	339	333
5.8	288	308	313	310
5.9	279	296	311	298
5.10	265	283	297	291

Fig. 16. SCP5 comparison

Table 14. comparison for SCP6 instances

Instances	Optimum	Proposed	S-Shape (5)	V-Shape (6)
6.1	138	146	151	149
6.2	146	157	159	155
6.3	145	153	163	160
6.4	131	144	160	156
6.5	161	177	187	177

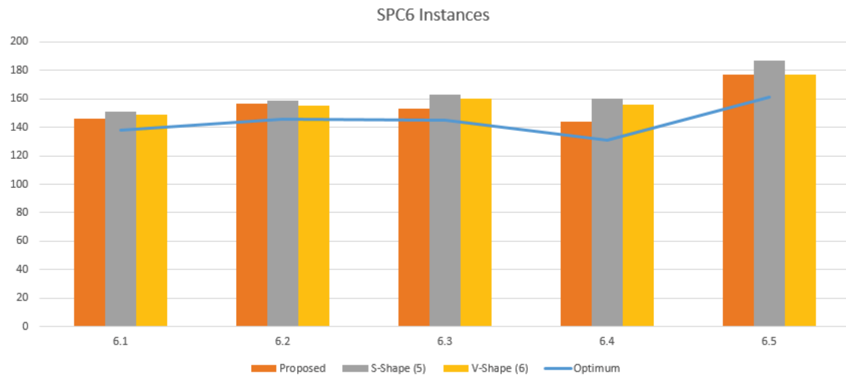

Fig. 17. SCP6 comparison

Table 15. comparison for A instances

Instances	Optimum	Proposed	S-Shape (5)	V-Shape (6)
A.1	253	298	315	301
A.2	252	301	303	301
A.3	232	256	259	254
A.4	234	268	289	270
A.5	236	266	271	263

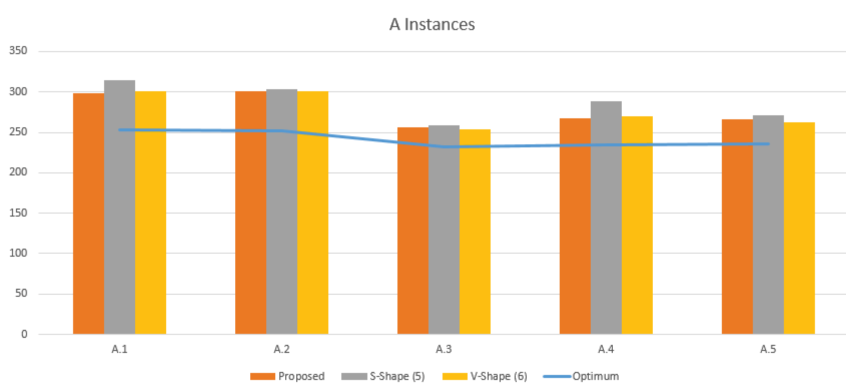

Fig. 18. SCP A comparison

Table 16. comparison for B instances

Instances	Optimum	Proposed	S-Shape (5)	V-Shape (6)
B.1	69	82	87	93
B.2	76	99	101	95
B.3	80	89	98	99
B4	79	88	91	87
B.5	72	88	92	84

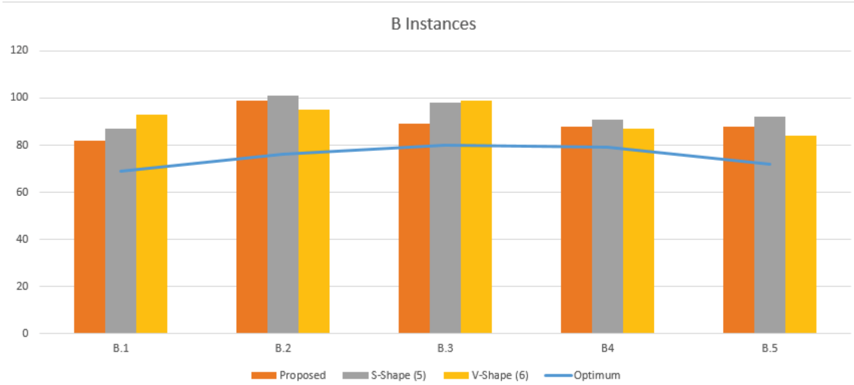

Fig. 19. SCP B comparison

Table 17. comparison for C instances

Instances	Optimum	Proposed	S-Shape (5)	V-Shape (6)
C.1	227	252	256	251
C.2	219	245	250	248
C.3	243	266	259	273
C.4	219	252	255	248
C.5	215	232	245	247

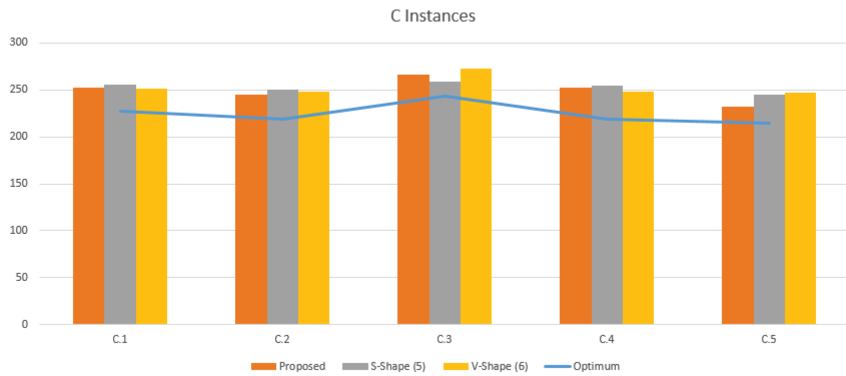

Fig. 20. SCP C comparison

Table 18. comparison for D instances

Instances	Optimum	Proposed	S-Shape (5)	V-Shape (6)
D.1	60	71	73	67
D.2	66	73	74	75
D.3	72	80	91	81
D.4	62	70	77	70
D.5	61	72	79	78

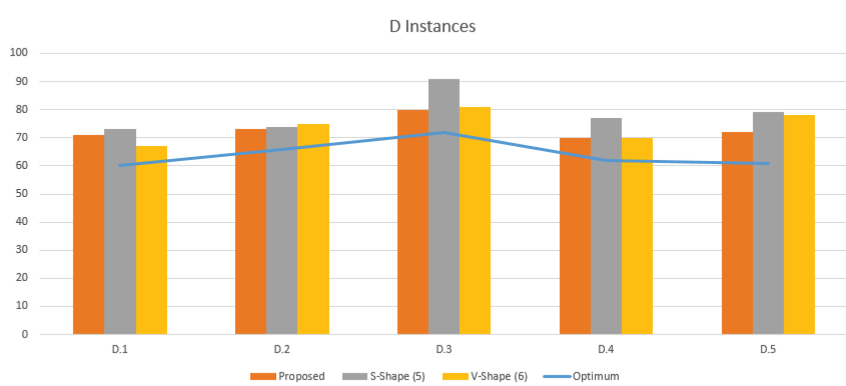

Fig. 21. SCP D comparison

Table 19. comparison for E instances

Instance	Optimum	Proposed	S-Shape (5)	V-Shape (6)
E.1	5	6	10	7
E.2	5	8	15	12
E.3	5	9	8	7
E.4	5	7	8	11
E.5	5	13	9	6

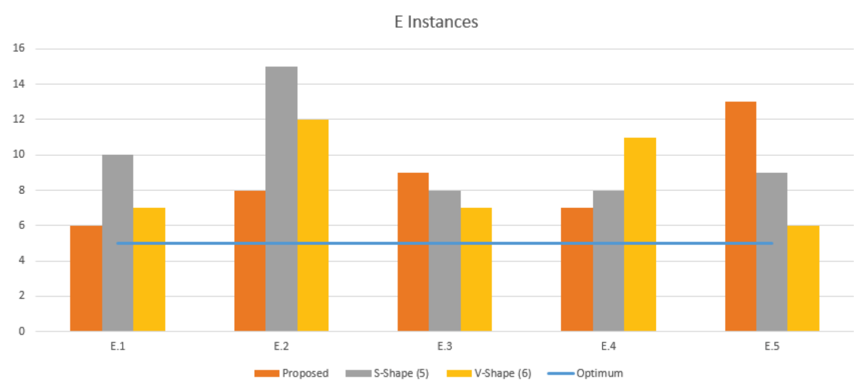

Fig. 22. SCP E comparison

Table 20. comparison for NRE instances

Instance	Optimum	Proposed	S-Shape (5)	V-Shape (6)
NRE1	29	72	77	74
NRE2	30	64	81	97
NRE3	27	59	82	74
NRE4	28	49	39	55
NRE5	28	33	37	39

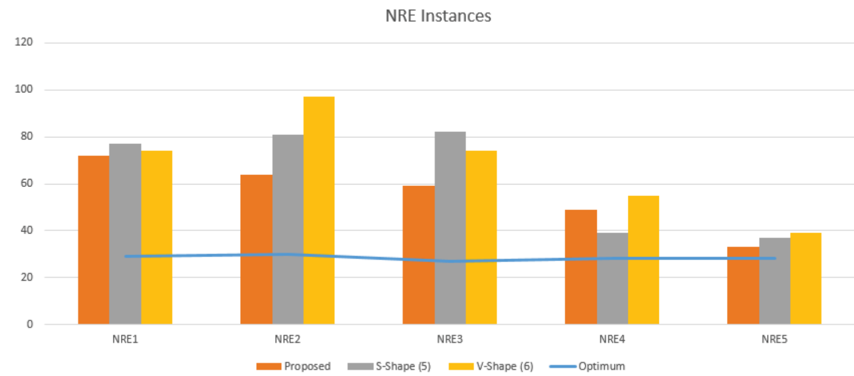

Fig. 23. SCP NRE comparison

Table 21. comparison for NRF instances

Instance	Optimum	Proposed	S-Shape (5)	V-Shape (6)
NRF1	14	37	34	21
NRF2	15	18	23	18
NRF3	14	27	28	17
NRF4	14	29	39	28
NRF5	13	22	23	17

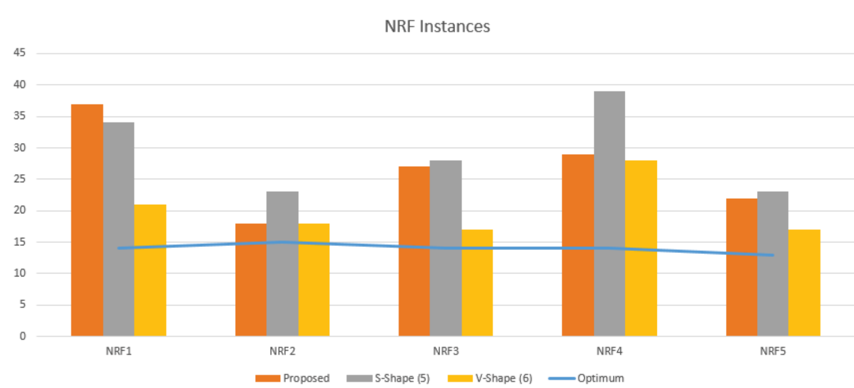

Fig. 24. SCP NRF comparison

Table 22. comparison for NRG instances

Instance	Optimum	Proposed	S-Shape (5)	V-Shape (6)
NRG1	176	200	220	214
NRG2	154	197	233	215
NRG3	166	234	318	312
NRG4	168	273	285	289
NRG5	168	202	187	211

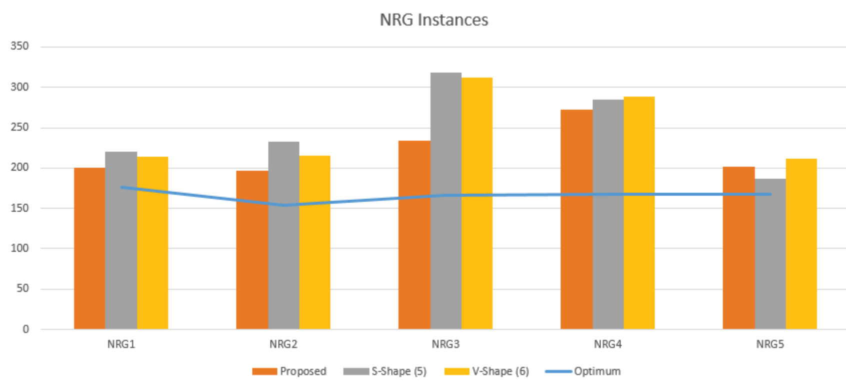

Fig. 25. SCP NRG comparison

Table 23. comparison for NRH instances

Instance	Optimum	Proposed	S-Shape (5)	V-Shape (6)
NRH1	63	111	98	77
NRH2	63	97	99	115
NRH3	59	93	96	213
NRH4	58	96	99	88
NRH5	55	89	88	78

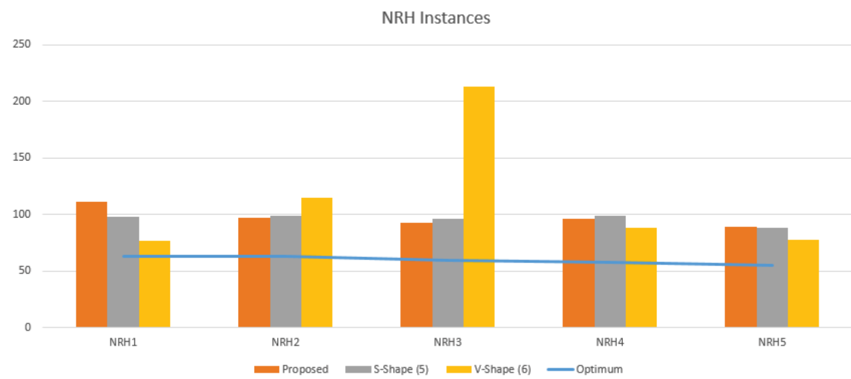

Fig. 26. SCP NRH comparison

Table 24. Wilcoxon-Mann-Whitney/Unpaired t-test part 1 results

Instance		V-shape	Proposed	Instance		V-shape	Proposed	Instance		V-shape	Proposed
4.1	V-shape	-	>0,05	6.1	V-Shape	-	>0,05	D.1	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
4.2	V-Shape	-	>0,05	6.2	V-Shape	-	>0,05	D.2	V-Shape	-	>0,05
	Proposed	6,80E-09	-		Proposed	>0,05	-		Proposed	>0,05	-
4.3	V-Shape	-	6,82E-09	6.3	V-Shape	-	0,02768	D.3	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
4.4	V-Shape	-	6,33E-03	6.4	V-Shape	-	0,01671	D.4	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
4.5	V-Shape	-	3,77E-02	6.5	V-Shape	-	>0,05	D.5	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
4.6	V-Shape	-	0,0001595	A.1	V-Shape	-	>0,05	E.1	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
4.7	V-Shape	-	6,14E-02	A.2	V-Shape	-	>0,05	E.2	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
4.8	V-Shape	-	2,20E-16	A.3	V-Shape	-	>0,05	E.3	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
4.9	V-Shape	-	>0,05	A.4	V-Shape	-	>0,05	E.4	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
4.10	V-Shape	-	3,20E-05	A.5	V-Shape	-	>0,05	E.5	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	0,0496	-		Proposed	1,67E-03	-
5.1	V-Shape	-	>0,05	B.1	V-Shape	-	9,73E-06	NRE.1	V-Shape	-	>0,05
	Proposed	7,18E-04	-		Proposed	>0,05	-		Proposed	>0,05	-
5.2	V-Shape	-	3,31E-08	B.2	V-Shape	-	>0,05	NRE.2	V-Shape	-	1,48E-08
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
5.3	V-Shape	-	0,0001507	B.3	V-Shape	-	>0,05	NRE.3	V-Shape	-	5,74E-05
	Proposed	>0,05	-		Proposed	0,0007408	-		Proposed	>0,05	-
5.4	V-Shape	-	2,70E-02	B.4	V-Shape	-	>0,05	NRE.4	V-Shape	-	0,009479
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
5.5	V-Shape	-	>0,05	B.5	V-Shape	-	>0,05	NRE.5	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
5.6	V-Shape	-	1,52E-02	C.1	V-Shape	-	>0,05	NRF.1	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	5,02E-06	-
5.7	V-Shape	-	>0,05	C.2	V-Shape	-	>0,05	NRF.2	V-Shape	-	>0,05
	Proposed	6,85E-09	-		Proposed	>0,05	-		Proposed	>0,05	-
5.8	V-Shape	-	>0,05	C.3	V-Shape	-	0,007696	NRF.3	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	3,69E-02	-
5.9	V-Shape	-	>0,05	C.4	V-Shape	-	>0,05	NRF.4	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	>0,05	-
5.10	V-Shape	-	0,00154	C.5	V-Shape	-	1,83E-04	NRF.5	V-Shape	-	>0,05
	Proposed	>0,05	-		Proposed	>0,05	-		Proposed	7,11E-03	-

Table 25. Wilcoxon-Mann-Whitney/Unpaired t-test part 2 results

Instance		V-shape	Proposed
NRG.1	V-Shape	-	1,58E-07
	Proposed	>0,05	-
NRG.2	V-Shape	-	6,79E-12
	Proposed	>0,05	-
NRG.3	V-Shape	-	6,76E-09
	Proposed	>0,05	-
NRG.4	V-Shape	-	1,04E-05
	Proposed	>0,05	-
NRG.5	V-Shape	-	>0,05
	Proposed	1,74E-05	-
NRH.1	V-Shape	-	2,20E-16
	Proposed	>0,05	-
NRH.2	V-Shape	-	>0,05
	Proposed	>0,05	-
NRH.3	V-Shape	-	>0,05
	Proposed	>0,05	-
NRH.4	V-Shape	-	>0,05
	Proposed	>0,05	-
NRH.5	V-Shape	-	>0,05
	Proposed	8,81E-05	-

References

1. L. A.Baykasoglu and P.Tapkan, *Artificial bee colony algorithm and its application to generalized assignment problem*, I-Tech Education and Publishing, Vienna, (2007).
2. B. C. Alvaro Gómez and R. Soto, *Solving the set covering problem with a binary black hole inspired algorithm*, Computational Science and Its Applications ICCSA 2016, 2 (2016), pp. 207–219.
3. R. S. Alvaro Gómez, Broderick Crawford, *A binary black hole algorithm to solve the set covering problem*, 2016 11th Iberian Conference on Information Systems and Technologies (CISTI), 2 (2016), pp. 1–5.
4. F. Amini and P. Ghaderi, *Hybridization of harmony search and ant colony optimization for optimal locating of structural dampers*, Applied Soft Computing, 13 (2013), pp. 2272–2280.
5. Artech, *Genexus*. url-http://www.genexus.com.
6. J. Beasley, *Or-library*. urlhttp://people.brunel.ac.uk/ mastjjb/jeb/orlib/scpinfo.html, 1990.
7. J. E. Beasley, *A lagrangian heuristic for set-covering problems*, Naval Research Logistics, 1 (1990), pp. 151–164.
8. M. Bellmore and H. D. Ratliff, *Optimal defense of multi-commodity networks*, Management Science, 18 (1971), pp. B–174.
9. H. Bouchekara, *Optimal power flow using black hole based optimization approach*, Applied Soft Computing, 24 (2014), pp. 879–888.
10. B. Crawford, R. Soto, F. Johnson, S. Misra, and F. Paredes, *The use of metaheuristics to software project scheduling problem*, in Computational Science and Its Applications - ICCSA 2014 - 14th International Conference, Guimarães, Portugal, June 30 - July 3, 2014, Proceedings, Part V, B. Murgante, S. Misra, A. M. A. C. Rocha, C. M. Torre, J. G. Rocha, M. I. Falcão, D. Taniar, B. O. Apduhan, and O. Gervasi, eds., vol. 8583 of Lecture Notes in Computer Science, Springer, 2014, pp. 215–226.
11. B. Crawford, R. Soto, F. Johnson, and F. Paredes, *A timetabling applied case solved with ant colony optimization*, in Artificial Intelligence Perspectives and Applications - Proceedings of the 4th Computer Science On-line Conference 2015 (CSOC2015), Vol 1: Artificial Intelligence Perspectives and Applications, R. Silhavy, R. Senkerik, Z. K. Oplatková, Z. Prokopova, and P. Silhavy, eds., vol. 347 of Advances in Intelligent Systems and Computing, Springer, 2015, pp. 267–276.
12. B. Crawford, R. Soto, C. Peña, M. Riquelme-Leiva, C. Torres-Rojas, S. Misra, F. Johnson, and F. Paredes, *A comparison of three recent nature-inspired metaheuristics for the set covering problem*, in Computational Science and Its Applications - ICCSA 2015 - 15th International Conference, Banff, AB, Canada, June 22-25, 2015, Proceedings, Part IV, O. Gervasi, B. Murgante, S. Misra, M. L. Gavrilova, A. M. A. C. Rocha, C. M. Torre, D. Taniar, and B. O. Apduhan, eds., vol. 9158 of Lecture Notes in Computer Science, Springer, 2015, pp. 431–443.
13. C. C.ReVelle and L.Falkson, *Applications of the location set-covering problem*, Geographical Analysis, 8 (1976), p. 65–76.

14. M. Desrochers and F. Soumis, *A column generation approach to the urban transit crew scheduling problem*, Transportation Science, 23 (1989), pp. 1–13.
15. F. E.Alba, G.Molina, *Optimal placement of antennae using metaheuristics*, NMA'06 Proceedings of the 6th international conference on Numerical methods and applications, (2006), pp. 214–222.
16. M. Farahmandian and A. Hatamlou, *Solving optimization problems using black hole algorithm*, Journal of Advanced Computer Science & Technology, 4 (2015), pp. 68–74.
17. D. Forney, *Generalized minimum distance decoding*, IEEE Transactions on Information Theory, 12 (1966), pp. 125–131.
18. R. S. Garfinkel and G. L. Nemhauser, *Optimal political districting by implicit enumeration techniques*, Management Science, 347 (2015), pp. 267–276.
19. O. Gervasi, B. Murgante, S. Misra, M. L. Gavrilova, A. M. A. C. Rocha, C. Torre, D. Taniar, and B. O. Apduhan, *Computational Science and Its Applications–ICCSA 2015: 15th International Conference, Banff, AB, Canada, June 22-25, 2015, Proceedings*, vol. 9157, Springer, 2015.
20. A. Gómez, *A binary black hole algorithm to solve the set covering problem*, 29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA AIE, 2 (2016), pp. 873–883.
21. D. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley Longman Publishing Co., Inc., (1989).
22. A. Hatamlou, *Black hole: A new heuristic optimization approach for data clustering*, Information sciences, 222 (2013), pp. 175–184.
23. S. Hawking, *Agujeros negros y pequeños universos*, Planeta, 1994.
24. S. Hawking and M. Jackson, *A brief history of time*, Dove Audio, 1993.
25. A. Heidari and R. Abbaspour, *Improved black hole algorithm for efficient low observable ucav path planning in constrained aerospace*, Advances in Computer Science: an International Journal, 3 (2014), pp. 87–92.
26. D. Hochba, *Approximation algorithms for np-hard problems*, ACM SIGACT News, (1997), pp. 40–52.
27. J. G. Jose Lanza, *Assuming multiobjective metaheuristics to solve a three-objective optimisation problem for relay node deployment in wireless sensor networks*, Applied Soft Computing, (2014), pp. 675–687.
28. Y. X. J.Zhang, K.Liu, *Random black hole particle swarm optimization and its application*, International Conference on Neural Networks and Signal Processing, 2008, pp. 359–365.
29. R. Karp, *Reducibility among combinatorial problems*. urlhttp://www.cs.berkeley.edu/ luca/cs172/karp.pdf, 1972.
30. F. Q. Kevin Curtin, Karen Hayslett-Mc Call, *Determining optimal police patrol areas with maximal covering and backup covering location models*, Netw Spat Econ, (2007).
31. N. B. Mostafa Nemati, Reza Salimi, *Black holes algorithm: A swarm algorithm inspired of blackholes for optimization problems*, (2013), pp. 143–150.
32. M. Nemati, H. Momeni, and N. Bazrkar, *Binary black holes algorithm*, Int. J. Comput. Appl, 79 (2013), pp. 36–42.
33. M. Nemati, R. Salimi, and N. Bazrkar, *Black holes algorithm: A swarm algorithm inspired of black holes for optimization problems*, IAES International Journal of Artificial Intelligence (IJ-AI), 2 (2013), pp. 143–150.
34. M. Nemati, R. Salimi, B. Moshref, and N. Bazrkar, *Fuzzy black holes algorithm*, IAES International Journal of Artificial Intelligence (IJ-AI), 3 (2014), pp. 49–55.
35. A. P. Piotrowski, J. J. Napiorkowski, and P. M. Rowinski, *How novel is the "novel" black hole optimization approach?*, Information Sciences, 267 (2014), pp. 191–200.
36. P.Itaim, *Resolución del problema de set-covering usando un algoritmo genético*, Universidad de Valparaíso, (2005).
37. J. K. Russ Eberhart, *A new optimizer using particle swarm theory*, Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on, (1995), pp. 39–43.
38. A. L. Seyedali Mirjalili, *S-shaped versus v-shaped transfer functions for binary particle swarm optimization*, 2 (2012), pp. 1–14.
39. R. Soto, B. Crawford, C. Galleguillos, J. Barraza, S. Lizama, A. Muñoz, J. Vilches, S. Misra, and F. Paredes, *Comparing cuckoo search, bee colony, firefly optimization, and electromagnetism-like algorithms for solving the set covering problem*, in Computational Science and Its Applications - ICCSA 2015 - 15th International Conference, Banff, AB, Canada, June 22-25, 2015, Proceedings, Part I, O. Gervasi, B. Murgante, S. Misra, M. L. Gavrilova, A. M. A. C. Rocha, C. M. Torre, D. Taniar, and B. O. Apduhan, eds., vol. 9155 of Lecture Notes in Computer Science, Springer, 2015, pp. 187–202.
40. F. Vasko, F. Wolf, and K. Stott, *Optimal selection of ingot sizes via set covering*, Operations Research, 35 (1987), pp. 346–353.
41. W.Marianov and C.Revelle, *The queuing probabilistic location set covering problem and some extensions*, Socio-Economic Planning Sciences, 28 (1994), pp. 167–178.
42. W.Walker, *Using the set-covering problem to assign fire companies to fire houses*, Operation Research, (1974), pp. 275–277.
43. F. R. W.Wang, S.Caro and B.Crawford, *Multi-objective robust optimization using a postoptimality sensitivity analysis technique: Application to a wind turbine design*, Journal of Mechanical Design, 137 (2015), p. 11.