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

AN ARTIFICIAL FISH SWARM ALGORITHM TO SOLVE THE SET COVERING PROBLEM

SEBASTIÁN PATRICIO MANSILLA VILLABLANCA

MASTER'S THESIS DEGREE
MAGÍSTER EN INGENIERÍA INFORMÁTICA

AUGUST, 2016

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

AN ARTIFICIAL FISH SWARM ALGORITHM TO SOLVE THE SET COVERING PROBLEM

SEBASTIÁN PATRICIO MANSILLA VILLABLANCA

Dr. Broderick Crawford Labrín
Master's Thesis Advisor

Study Program: **Magíster en Ingeniería Informática**

AUGUST, 2016

Abstract

Metaheuristics provide "acceptable" solutions in a reasonable time for solving hard and complex problems when it is expensive to find an optimal solution, especially with a limited computing capacity. Particularly, metaheuristics can solve The Set Covering Problem (SCP) which consists in finding a set of solutions to cover a set of needs at the lowest possible cost. The main goal of this thesis is to show the performance of two versions of Artificial Fish Swarm Algorithm (AFSA) applied to SCP. This algorithm simulates the behavior of a fish in water, which belongs to a shoal, and it uses a population of points in space to represent the position of fish in the shoal. This method was tested on 70 files obtained from OR-Library website. It is expected that AFSA provides another alternative based population metaheuristic to solve minimization problems in different contexts, not only in the academic scope, also in industry.

Keywords: Set Covering Problem, Artificial Fish Swarm Algorithm, Metaheuristics, Combinatorial Optimization Problem.

Resumen

Las metaheurísticas proveen soluciones "aceptables", en un tiempo razonable, para resolver problemas difíciles y complejos cuando es costoso encontrar una solución óptima, especialmente con una capacidad de procesamiento limitada. En particular, las metaheurísticas pueden resolver el problema de cobertura de conjuntos, en inglés "Set Covering Problem (SCP)", el cual consiste en encontrar un conjunto de soluciones, que permiten cubrir un conjunto de necesidades, con el menor costo posible. El principal propósito de esta tesis, es mostrar el desempeño de dos versiones del algoritmo artificial de cardumen de peces, en inglés "Artificial Fish Swarm Algorithm (AFSA)" aplicado al SCP. Este algoritmo, simula el comportamiento de un pez dentro del agua, el cual pertenece a un cardumen y utiliza una población de puntos en el espacio, para representar la posición del pez dentro del cardumen. Este método, ha sido probado sobre 70 archivos obtenidos del sitio web de OR-Library. Se espera que AFSA, pueda proveer otra alternativa de metaheurística basada en población, que sea capaz de resolver problemas de minimización y en diferentes contextos, no solo en el ámbito académico, también en la industria.

Palabras Clave: Problema de Cobertura de Conjuntos, Optimización de Cardumen de Peces Artificial, Metaheurísticas, Optimización Combinatoria.

General Index

List of Figures

List of Tables

1 Chapter 1: Introduction

1.1 Introduction

Applications of optimization are numerous. There is no company that is not involved in solving optimization problems and almost every process has a potential to be optimized. Also, many challenging works in science and industry can be formulated as optimization problems. There are different kinds of optimization such as minimization of time, minimization of cost, minimization of risk, maximization of profit, maximization of quality or maximization of efficiency. For example, there are many ways of designing a network to optimize the cost, the quality of service or scheduling a production to optimize the time, and so on. In real life, optimization problems are very common such as in science, engineering, economics and business. They are complex, difficult to solve and they cannot be solved with exact methods within a reasonable amount of time. Therefore, approximate algorithms are the main alternative to solve this kind of problems, like metaheuristics, which are algorithms that can be used to solve any optimization problem. Moreover, they solve instances of problems that are believed to be hard in general. These algorithms can reach this objective by reducing the size of the problem and/or by exploring the search space efficiently. In general, they have six main purposes; solving problems faster, solving large problems, obtaining robust algorithms, they are simple to design and implement, and they are very flexible [1].

Metaheuristics are a part of optimization in computer science and applied mathematics that are related to algorithms and computational complexity theory. They represent a family of approximate optimization techniques which have gained popularity in the past two decades. Until year 2009, they are among the most promising and successful techniques. Many metaheuristics have been developed in various academic scopes such as artificial intelligence, computational intelligence, soft computing, mathematical programming and operations research. Many metaheuristics imitate natural behaviors to solve complex optimization problems such as evolution of species, particle swarms or bee colonies. So, in practice, metaheuristics are increasingly interesting in diverse technologies, industries, and services since they have proved to be efficient algorithms in solving a wide range of complex real life optimization problems in different domains such as logistics, bioinformatics, engineering design, networking, transportation, data mining, finance, business and among others [1].

One of the classical problems that Metaheuristics try to solve is the Set Covering Problem (SCP) which represents an important class of NP-hard (non-deterministic polynomial-time hard) combinatorial optimization problems with many applications in different domains such as optimal selection of ingot sizes [2] or assign fire companies to fire houses [3], and so on. In recent years, SCP has been solved with good results, in academic scope, with different algorithms such as cultural algorithm [4], fruit fly optimization algorithm [5], teaching-learning-based optimization algorithm [6], a beam-search approach [7], nature inspired intelligent water drop algorithm [8], binary invasive weed optimization algorithm [9], firework explosions [10, 11], biogeography-based optimization algorithm [12, 13], cat swarm optimization [14, 15, 16, 17], binary cuckoo search algorithm [18], swarm intelligence algorithms [19], binary harmony search algorithm [20], the soccer league competition algorithm [21] or binary black hole algorithm [22]. SCP consists of choosing a subset of columns at a minimal cost in a way to cover all the rows (at least one on each line) [1]. In other words, it consists of finding a set of solutions at the lowest possible cost, with the constraints of a matrix that has zeros and ones, where each row must be covered of at least one column.

The main goal of this thesis is to show the performance of Artificial Fish Swarm Algorithm (AFSA) applied to SCP, another alternative based population metaheuristic. Previously, it was tested on the knapsack problem [23, 24, 25], a maximization problem. Therefore, the objective is to transform this algorithm to be able to solve a minimization problem. In recent years, this algorithm has been developed in continuous global optimization and it has different version. So, in the present work the simplified and the improved version of this algorithm will be applied in a binary search space. This algorithm, simulates the behavior of a fish in water, which belongs to a shoal, and it uses a population of points in space to represent the position of fish in the shoal. AFSA is based on five main behaviors such as random, chasing, swarming, searching and leaping and it has the following steps, initialization of the population, generation of trial points, dealing with SCP constraints, selection of a new population, population reinitializing, local search and termination conditions.

This method has been tested by programming this algorithm in Java Language, on 70 files, all of them obtained from OR-Library website [26]. These 70 files are formatted as: number of rows m, number of columns n, the cost of each column $c_j, \forall\, j \in \{1,\ldots,n\}$, and for each row $i, \forall\, i \in \{1,...,m\}$ the number of columns which cover row i followed by a list of the columns which cover rows i. Each one represents a different kind of SCP costs. Also, these 70 files were chosen in order to solve SCP academically and theoretically, with the results of the experiments obtained, AFSA is expected to provide another metaheuristic alternative based on population that is able to solve the SCP in different contexts, not only in academic scope, but also in the industry or engineering in order to apply this algorithm to solve minimization problems.

1.2 General Objective

Solve the Set Covering Problem using an Artificial Fish Swarm Algorithm.

1.3 Specific Objectives

- Study and understand of SCP.
- Research and understand of AFSA metaheuristic.
- Modify and adjust AFSA in order to solve SCP.
- Solve SCP using the two selected versions of AFSA.
- Analyze and compare the results obtained with the two versions of AFSA.

2 Chapter 2: Conceptual Framework

2.1 Metaheuristics

2.1.1 Definition

The meaning of the word metaheuristic comes from an old greek word; heuriskein, which means the art of discovering new strategies or rules to solve problems. The suffix meta is also a greek word, which means "upper level methodology". Metaheuristics represent a family of approximate optimization and they may provide "acceptable" or "good" solutions in a reasonable time, seeking solutions in the search space in order to solve hard and complex problems in science and engineering. Unlike exact optimization algorithms, metaheuristics do not guarantee to obtain the optimal solutions, although, sometimes it is possible to find a local optimum which can be very close to the optimal solution or even obtain the global optimum in some cases. In metaheuristics, there are two focus that must be taken into account; exploration of the search space (diversification) and exploitation of the best solutions found (intensification) [1].

This thesis is based on a population metaheuristic which simulates the behavior of species. In this case, AFSA. For these purposes, the concept is defined as follows:

Population-based metaheuristics: They start from an initial population of solutions. Next, they iteratively apply a generation phase where a new population of solutions is created and the replacement phase where the population is selected from the current and the new population. This process iterates until a given stopping criteria. The generation and the replacement phases may be memoryless. In this case, the two procedures are based only on the current population. Otherwise, some history of the search stored in a memory can be used in the generation of the new population and the replacement of the old population. Most of these metaheuristics are nature-inspired algorithms [1].

2.1.2 Main Definitions

- **Objective Function:** It is defined by the objective to be reached. It is an equation that needs to be optimized by fulfilling the constraints given with variables that need to be maximized or minimized. The value of its result defines the solution quality.
- **Variable:** It is an element that can take any value within the domain.
- **Solution Representation:** It is represented by an array of n components which represent the n variables that are included the solutions.
- **Fitness:** It is the value of the evaluation of the objective function.
- **Domain:** It is the set of values that may take the variables.
- **Constraint:** It is a condition that must be met in order to satisfy the problem.
- **Search Space:** It is the set of all possible solutions.
- **Parameter:** It is a numeric value that is utilized in order to configure the metaheuristic.
- **Global Optimum:** It is a solution that has the best objective function of all search space solutions.

- **Local Optimum:** It is a solution that has a better objective function compare to its neighbors.
- **Initial Population:** It comprises of all members which start the metaheuristic in order to find the problem solution.
- **Exploration/Diversification:** In diversification, non-explored regions must be visited to be sure that all regions of the search space are evenly explored and that the search is not confined to only a reduced number of regions.
- **Exploitation/Intensification:** In intensification, the promising regions are explored more thoroughly in the hope of finding better solutions.

2.1.3 When Utilize Metaheuristics?

First of all, the complexity of a problem gives an indication of the hardness of the problem. It is important to know the size of input instances that the algorithm is supposed to solve. Even if a problem is NP-hard, small instances may be solved by an exact approach. Moreover, the structure of the instances plays an important role. Some medium or even large size problems with a specific structure may be solved by exact algorithms. Finally, the required search time to solve a given problem is an important issue in the selection of an optimization algorithm [1].

It is unwise to use metaheuristics to solve problems where efficient exact algorithms are available. In the case where those exact algorithms give "acceptable" search time to solve the target instances, metaheuristics are useless. For instance, one should not use a metaheuristic to find a minimum spanning tree or a shortest path in a graph. Known polynomial-time exact algorithms exist for those problems [1].

Hence for easy optimization problems, metaheuristics are seldom used. Unfortunately, one can see many engineers and even researchers solving polynomial optimization problems with metaheuristics. So the first guideline in solving a problem is to analyze its complexity. If the problem can be reduced to a classical or an already solved problem in the literature, it is suggested to take a look at the state of the art in order to find the best optimization algorithms that can solve the problem. Otherwise, if there are related problems, the same methodology must be applied [1].

3 Chapter 3: Theoretical Framework

3.1 Set Covering Problem

3.1.1 SCP Definition

The SCP is a popular $\mathscr{NP}$ - *hard* problem [27] and it is a classical problem of combinatorial optimization which consists of finding a set of solutions that allow to cover a set of necessities with a lowest possible cost [28].

The SCP formulation is the following: given a binary matrix $A = (a_{ij})$ of zeros and ones with dimension of m-rows and n-columns, and let a nonnegative n-dimensional integer costs vector C. Also, $I = \{1,2,\ldots,m\}$ and $J = \{1,2,\ldots,n\}$ are set of rows and columns, respectively. So, $c_j > 0$ for ($j \in J$) the cost of selecting the column j of matrix A which covers a row i if $a_{ij} = 1$, $a_{ij} = 0$ otherwise. Thereby, the objective of SCP is to minimize the subset of columns or variables $S \subseteq J$, where each row $i \in I$ should be covered at least by one column $j \in S$. Therefore, the SCP can be formulated mathematically as follows [28, 27] in equation 1:

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

Subject to:

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

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

Where x_j is a variable, so $x_j = 1$ if the column j is within solution and 0 otherwise. The constraints of SCP, ensure that each row i should be covered for at least one column j.

Shown above in equation 1 is a typical representation of SCP which is called column-based representation and it is a string of 0 and 1 with size n, where n is the amount of columns of SCP [29]. The objective of representation that was shown before is to find a minimum cost $S \subseteq J$, such that each row $i \in I$ is covered by at least one column $j \in S$.

3.1.2 State of the Art

There are two kinds of methods that can solve SCP. In the first place, exact methods that can be the following algorithms: dynamic programming, branch and bound, branch and cut, branch and price, constraint programming, or iterative deepening algorithms. These enumerative methods may be viewed as tree search algorithms. The search is done over the whole search space of interest, and the problem is solved by subdividing it into simpler problems [1]. An example of solving SCP with exact method is branch and bound or branch and cut [30].

On the other hand and unlike exact methods, metaheuristics allow to tackle large size problem instances by delivering satisfactory solutions in a reasonable time. There is no guarantee of finding global optimal solutions. Their use in many applications shows their efficiency and effectiveness to solve large and complex problems [1]. In the past, SCP was successfully solved with classical metaheuristics such as genetic algorithm [31, 32, 29], taboo search [33] or

simulated annealing [34, 30]. In other cases, metaheuristics inspired in nature have solved with good results the SCP, for instance, artificial bee colony [35, 36], particle swarm optimization [37, 38], ant colony optimization [39, 40, 41], firefly algorithm [42, 43], shuffled frog leaping algorithm [44, 45], fruit fly optimization algorithm [46, 47, 48], cuckoo search algorithm [49] or cat swarm optimization [50, 51, 52]. Additionally, there are other metaheuristics that have solved the SCP and they are non nature inspired, such as cultural algorithm [4], teaching-learning-based optimization algorithm [53], fireworks [54] or biogeography-based optimization [55].

In others works there are comparisons among different kind of metaheuristics [56, 57] that solve the SCP with good results.

Finally, the SCP has been solved previously with the metaheuristics mentioned above in academic scope. However, these metaheuristics have been applied to real problems such as genetic algorithm in order to solve a trainer scheduling problem [58], firefly algorithm to solve a project scheduling problem [59], firefly algorithm to solve the manufacturing cell design problem [60], hybrid ant algorithm for the airline crew pairing problem [61], and so on.

Therefore, metaheuristics not only can solve theoretical problems, also they can solve problems of the real world.

3.1.3 Applications

There are many applications of metaheuristics in real life and they fall into a large number of areas; some them can be [1]:

1. Engineering design, topology optimization and structural optimization in electronics and VLSI, aerodynamics, fluid dynamics, telecommunications, automotive and robotics.
2. Machine learning and data mining in bioinformatics and computational biology, and finance.
3. System modeling, simulation and identification in chemistry, physics, and biology; control, signal, and image processing.
4. Planning in routing problems, robot planning, scheduling and production problems, logistics and transportation, supply chain management, environment and so on.

The SCP has different applications in real life such as location of emergency services [62], assignment of flight crews [63]. Moreover, assignment of bus crews [64, 65], logical analysis of numeric data [66], steel production [67], vehicle routes [68], the efficient scheduling of a fleet of ships engaged in pickup and delivery of bulk cargoes [69], in other cases may be petroleum, or metals. Another example [70] is the of Bethlehem Steel Corporation, where this company applied set covering in order to select the optimal ingot sizes and internal ingot mold sizes from among feasible sizes.

3.1.4 Example of SCP

A classical example is the installation of firefighter stations. This example enables explaining SCP in a practical way, given a station with capability of covering all emergencies in its area or in its closest areas. For instance, if a station is built in area one (see Figure 1) it can cover the emergencies of its area and emergencies of its neighborhoods, which means, in *area* 1, *area* 2, *area* 3 and *area* 5.

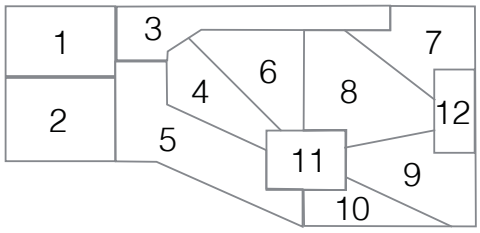

Figure 1: Example of SCP's areas.

It is necessary to build the number of stations that allow to cover all areas when emergencies occur. There is a constraint which is that all areas should be cover for at least one station and the objective is minimize the number of built stations. In this example, the cost of building an station is equal in all areas. The variable x_j is the area in the map and it could be 1 if the station is going to be built and 0 if not. Therefore, SCP applied to this example is formulated as follows in equation 4:

$$\text{Minimize} \quad \sum_{j=1}^{n} x_j \tag{4}$$

With a cost $c_j = 1$, which means a unicost SCP.

Subject to:

$$x_1 + x_2 + x_3 + x_5 \geq 1 \tag{5}$$

$$x_2 + x_1 + x_5 \geq 1 \tag{6}$$

$$x_3 + x_1 + x_4 + x_5 + x_6 + x_7 + x_8 \geq 1 \tag{7}$$

$$x_4 + x_3 + x_5 + x_6 + x_{11} \geq 1 \tag{8}$$

$$x_5 + x_1 + x_2 + x_3 + x_4 + x_{10} + x_{11} \geq 1 \tag{9}$$

$$x_6 + x_3 + x_4 + x_8 + x_{11} \geq 1 \tag{10}$$

$$x_7 + x_3 + x_8 + x_{12} \geq 1 \tag{11}$$

$$x_8 + x_3 + x_6 + x_7 + x_9 + x_{11} + x_{12} \geq 1 \tag{12}$$

$$x_9 + x_8 + x_{10} + x_{11} + x_{12} \geq 1 \tag{13}$$

$$x_{10} + x_5 + x_9 + x_{11} \geq 1 \tag{14}$$

$$x_{11} + x_4 + x_5 + x_6 + x_8 + x_9 + x_{10} \geq 1 \tag{15}$$

$$x_{12} + x_7 + x_8 + x_9 \geq 1 \tag{16}$$

$$x_j \in \{0,1\},\ \forall j \in \{1,...,12\} \tag{17}$$

The first constraint indicates that for covering *area* 1 it is necessary to build one station in the same area or in its neighborhood. The second one is for *area* 2, and so on. One possible optimal solution is to build firemen stations in area 5 and 8, which means that $x_5 = x_8 = 1$ and $x_1 = x_2 = x_3 = x_4 = x_6 = x_7 = x_9 = x_{10} = x_{11} = x_{12} = 0$. This could be as in Figure 2, where green areas are the areas covered by *station* 5, red areas are the areas covered by *station* 8 and blue areas are the areas covered by 5 and 8 at the same time.

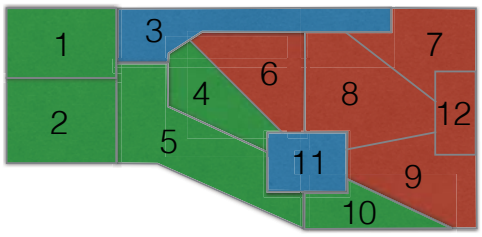

Figure 2: Solution Example of SCP's areas.

3.2 Artificial Fish Swarm Algorithm

3.2.1 AFSA Definition

Bio-inspired metaheuristics are a type of metaheuristics and are methods that simulate the behavior of a swarm or group of animals. Their purpose is also, they try to solve problems of optimization such as maximization or minimization. In this case, AFSA was created with the intention of solving the knapsack problem which is a kind of maximization problem. Hence, the objective of this work is to transform the application of a maximization problem into a minimization problem and observe its results because many others nature inspired metaheuristics have had good results on SCP, as mentioned above.

As many other nature inspired metaheuristics which imitate the behavior of animal or insect population, AFSA is not the exception. According to [23], this algorithm was proposed and applied in order to solve optimization problems and it simulates the behavior of a fish swarm inside the water where a fish represents a point or a fictitious entity of a true fish in a population and the swarm movements are randomly.

The main fish swarm behaviors are summarized as follows [23]:

1. **Random Behavior:** In order to find companion and food, a fish swims randomly in water.
2. **Chasing Behavior:** If food is discovered by a fish, the others in the neighborhood go after it quickly.
3. **Swarming Behavior:** In order to guarantee the survival of the swarm and avoid dangers from predators, fish move together in groups.
4. **Searching Behavior:** Fish go directly and quickly to a region, when it is discovered that it has more food. This can be by instinct or vision.
5. **Leaping Behavior:** Fish leap to look for food in other regions, when it stagnates in a region.

Artificial fish swarm try to search good results due to these five behaviors. Also, AFSA works with feasible solutions.

In addition to the explanation shown above, there is another description of AFSA which is proposed with more details in [24]. The concept of "visual scope" is the main concept utilized in that version of AFSA, and it represents how close the neighborhood is to a fish or point.

Depending on the position of a point related to the population, three situations may occur [24]:

a) The "visual scope" is empty, and the current point with no other points in its neighborhood, moves randomly looking for a better region.

b) When the "visual scope" is not crowded, the current point can move towards the best point within the "visual scope", or, if this best point does not improve the objective function value it moves towards the central point of the "visual scope".

c) When the "visual scope" is crowded, the current point has some difficulty in following any particular point, and searches for a better region by randomly choosing another point (from the "visual scope") and moving towards it.

The condition that decides when the "visual scope" of the current point is not crowded and the central point inside the "visual scope" are explained in [24].

3.2.2 Proposed Algorithm of AFSA Binary Version

Before of applying AFSA on SCP, it is necessary to show the binary version of this algorithm which was proposed by [23], but it was applied on the knapsack problem. That algorithm is shown in Algorithm 1 and it is the following:

Algorithm 1 Binary Version of AFSA

Set parameter values
Set $t = 1$ and randomly initialize $\boldsymbol{x}_i$, $i = 1, 2, ..., N$
Perform decoding in order to deal with constraints and evaluate z. Identify x_{max} and $z(x_{max})$
if Termination conditions are met **then**
    Stop
end if
for all $\boldsymbol{x}_i$ **do**
    Calculate "visual scope" and "crowding factor"
    Perform fish behaviors to create trial point $\boldsymbol{y}_i$
    Perform decoding to make the trial point feasible
end for
Perform selection to create new current points
Evaluate z and identify x_{max} and $z(x_{max})$
if $t\%L = 0$ **then**
    Perform leaping behavior
end if
Set $t = t + 1$ and go to *step* 4

3.2.3 How has AFSA been applied before?

With regard to specific AFSA, Improved Binary Artificial Fish Swarm Algorithm (IbAFSA) is an algorithm that was used for solving the $0-1$ multidimensional knapsack problems (MKP) in 2014 [24]. Also, this algorithm has other variations and recently, an artificial fish swarm algorithm has been developed in continuous global optimization. The algorithm uses a population of points in space to represent the position of fish in the school [24], for instance, in 2012 the same MKP was solved with Binary artificial fish swarm algorithm (bAFSA) [23] and last year, in 2015, a new Simplified binary artificial fish swarm algorithm (SbAFSA) was utilized for solving large $0-1$ MKP [25]. Therefore,

it is necessary to make some changes to this algorithm for applying it to SCP.

On the other hand, AFSA has been utilized in non binary problems such as forecasting stock indices using radial basis function neural networks optimized by AFSA in 2011 [71], an improved artificial fish swarm algorithm (IAFSA) for multi robot task scheduling was used in 2009 [72], IAFSA and its application in feed-forward neural networks [73], studies on AFSA based on decomposition and coordination techniques [74], wavelet threshold optimization with AFSA [75] or efficient job scheduling in grid computing with modified AFSA [76].

4 Chapter 4: Methodological Framework

4.1 Artificial Fish Swarm Algorithm and its Different Versions

In this section, the main steps of AFSA will be shown with its two versions, improved binary version and simplified binary version in order to solve SCP.

4.2 Improved Binary Version of AFSA Applied to SCP

The proposed improved binary version of AFSA will be presented to solve the 0-1 multidimensional problems [24]. The main steps of the algorithm is described in the following paragraphs in order to solve SCP specifically.

4.2.1 Initialization of Population

The best representation scheme to solve binary problems in improved version of AFSA is N points or solutions, $\boldsymbol{x}_i, i = 1,2,...,N$ which are randomly initialized and each one is represented by a binary 0/1 string of length n.

4.2.2 Generating Trial Points in Improved AFSA

In improved version of AFSA the Hamming distance, H_d, is used to identify the points inside the "visual scope" of current point $\boldsymbol{x}_i$ [24]. The Hamming distance between two bit sequences of equal length is the number of positions at which the corresponding bits are different. After calculating the Hamming distance between all pair of points from the population, the np_i points inside the "visual scope" of current point $\boldsymbol{x}_i$ are identified as the points that satisfy the condition $H_d(\boldsymbol{x}_i,\boldsymbol{x}_j) \leq \nu$, for $j \in \{1,...,N\}, j \notin i$ [24] , where:

$$\nu = \delta\ n, \tag{18}$$

And

$$C_f \equiv \frac{np_i}{N} \leq \theta, \tag{19}$$

Where C_f is the crowding factor and $\theta \in (0,1)$ is the crowd parameter.

In equation 18, $\delta \in (0,1)$ is the visual parameter and n represents the maximum Hamming distance between two binary points. After computing np_i, the crowding factor C_f of $\boldsymbol{x}_i$ is calculated using the equation 19. Depending on the value of C_f , the "visual scope" can be empty, not crowded or crowded.

Chasing behavior: If the "visual scope" of $\boldsymbol{x}_i$ is not crowded and the point that has the best objective function value inside the "visual scope", denoted by $\boldsymbol{x}_{min}$ and satisfies $z(\boldsymbol{x}_{min}) \leq z(\boldsymbol{x}_i)$. In chasing, each bit of the trial point, $\boldsymbol{y}_i$, is generated by copying the corresponding bit from $\boldsymbol{x}_i$ or from $\boldsymbol{x}_{min}$ with equal probability. This operation is similar to the

uniform crossover present in genetic/evolutionary algorithms [24]. In this work is according to the following criterion that appears in equation 20:

$$\boldsymbol{y}_i^d(t) = \begin{cases} \boldsymbol{x}_i^d(t), \ if \ rand(0,1) \leq 0.5 \\ \\ \boldsymbol{x}_{min}^d(t), \ \text{otherwise} \end{cases} \tag{20}$$

Swarming behavior: When the "visual scope" is not crowded and $z(\boldsymbol{x}_{min}) > z(\boldsymbol{x}_i)$ (chasing is not possible), then if $z(\bar{\boldsymbol{x}}) \leq z(\boldsymbol{x}_i)$, where $\bar{\boldsymbol{x}}$ is the central point inside the "visual scope" of the point $\boldsymbol{x}_i$. The central $\bar{\boldsymbol{x}}$, is the point closest to all the other points in the "visual scope", in the sense that the average Hamming distance to all other points in the "visual scope" is minimal. The pseudocode to compute the central point is shown in Algorithm 2 and there are more details in [24].

Algorithm 2 Central Point

Require: Set $\mathbf{I}^i$ and the np_i points inside the "visual scope" of $\boldsymbol{x}_i$

for j = 1 to n **do**
  Compute $\bar{x}_j = \frac{\sum_{l \in \mathbf{I}^i} \boldsymbol{x}_l}{np^i}$
  if $\bar{x}_j = 0.5$ **then**
    Set $\bar{x}_j$=Random Integer {0,1}
  else
    Set $\bar{x}_j$=Round($\bar{x}_j$)
  end if
end for
return Central Point $\bar{\boldsymbol{x}}$

In swarming, each bit of the trial $\boldsymbol{y}_i$ is created by copying the corresponding bit from $\boldsymbol{x}_i$ or from $\bar{\boldsymbol{x}}$ with equal probability [24]. In this work is according to the following criterion that appears in equation 21:

$$\boldsymbol{y}_i^d(t) = \begin{cases} \boldsymbol{x}_i^d(t), \ if \ rand(0,1) \leq 0.5 \\ \\ \bar{\boldsymbol{x}}^d(t), \ \text{otherwise} \end{cases} \tag{21}$$

Searching behavior: The searching behavior can be implemented in the following situations [24]:

a) When the "visual scope" is not crowded and neither $\boldsymbol{x}_{min}$ nor $\bar{\boldsymbol{x}}$ improves the objective function value.

b) When the "visual scope" is crowded.

Here, a point $\boldsymbol{x}_{rand}$ (rand $\in \mathbf{I}_i$) inside the "visual scope" of $\boldsymbol{x}_i$ is randomly selected and the searching behavior is to be implemented if $z(\boldsymbol{x}_{rand}) \leq z(\boldsymbol{x}_i)$. Otherwise, a random behavior is implemented [24]. In searching, each bit of $\boldsymbol{y}_i$ is created by copying the corresponding bit from $\boldsymbol{x}_i$ or $\boldsymbol{x}_{rand}$ with equal probability [24]. In this work is according to

criterion that appears in equation 22:

$$y_i^d(t) = \begin{cases} x_i^d(t),\ if\ rand(0,1) \leq 0.5 \\ \\ x_{rand}^d(t),\ \text{otherwise} \end{cases} \tag{22}$$

Random behavior: When the "visual scope" of x_i is empty or the other behavior were not performed, the point x_i performs the random behavior [24]. In this case, the trial point y_i is created by randomly setting a binary string of $0/1$ bits of length n [24].

4.2.3 Constraints Handling

In [24] AFSA was used to solve 0-1 multidimensional knapsack problems and it is shown an algorithm for those constraints handling. Therefore, a different method for constraints handling will be shown in order to solve SCP in the following algorithms.

The pseudocode to transform all members of the population into feasible solutions is shown in Algorithm 3.

Algorithm 3 Feasible Population in Improved Binary Version of AFSA

```
for all x_i, i = 1,2,...,N do
    for j = 1 to Number of Rows do
        for d = 1 to n do
            if a_{j,d} = 1 and x_i^d(t) = 1 then
                Break
            end if
            if a_{j,d} = 1 and x_i^d(t) = 0 then
                Set x_i^d(t) = 1
                Break
            end if
        end for
    end for
end for
```

The pseudocode to eliminate the redundancy in all members of the population is shown in Algorithm 4.

Algorithm 4 Elimination of the Redundancy of Population in Improved Binary Version of AFSA

for all $\boldsymbol{x}_i$, $i = 1,2,...,N$ **do**
  Set $flag = 0$
  Set $count = 0$
  Set ArrayList *rows*
  for $d = 1$ to n **do**
    for $j = 1$ to $Number\ of\ Rows$ **do**
      if $flag = 1$ *and* $\boldsymbol{x}_i^d(t) = 1$ **then**
        Set $\boldsymbol{x}_i^d(t) = 0$
        Break
      end if
      if $a_{j,d} = 1$ and $flag = 0$ and $\boldsymbol{x}_i^d(t) = 1$ and *row* is not contained in ArrayList **then**
        Set $count = count + 1$
        Add row to ArrayList
      end if
      if $count = Number\ of\ Rows$ **then**
        Set $flag = 1$
      end if
    end for
  end for
end for

Where $a_{j,d}$ corresponds to the matrix of constraints of the set covering problem of each instance.

4.2.4 Selection of New Population

Among the trial points $\boldsymbol{y}_i$ and the current points $\boldsymbol{x}_i, i = 1,2,...,N$, at iteration t, in order to decide whether or not they should become members of the population in the next iteration, $t+1$, the trial point is compared to the current point using the following criterion [24] in equation 23:

$$\boldsymbol{x}_i^d(t+1) = \begin{cases} \boldsymbol{y}_i^d(t)\ if\ z(\boldsymbol{y}_i^d(t)) \leq z(\boldsymbol{x}_i^d(t)) \\ \\ \boldsymbol{x}_i^d(t)\ \text{otherwise} \end{cases} \tag{23}$$

4.2.5 Leaping Behavior

According to [24], when the best objective function value in the population does not change for a certain number of iterations, the algorithm may have stagnated and the other points of the population will eventually converge to that objective function value. To be able to escape from that region and to try to converge to the optimal solution, the algorithm performs the leaping behavior, at every L iterations. In the leaping behavior, a point $\boldsymbol{x}_{rand}$ (rand $\in$

$\{1,2,...,N\}$) distinct from $\boldsymbol{x}_{min}$ is randomly selected from the current population and some randomly selected bits of the point are changed from 0 to 1 or vice versa with probability p_m. The value $p_m = 0.01$ is widely used in binary representation methods. However, in this work, a probability $p_m = 0.1$ will be utilized. The described operation is similar to a mutation with probability p_m of genetic/evolutionary algorithms. Then, this new point replaces the point $\boldsymbol{x}_{rand}$ [24].

4.2.6 Termination Conditions

Let T_{max} be the maximum number of iterations. Let z_{min} be the best objective function value attained at iteration t and z_{opt} be the known optimal value available in the literature. Then, the proposed conditions of stopping the algorithm are one of these conditions that appear in equation 24:

$$t > T_{max} \; or \; z_{min} \leq z_{opt} \tag{24}$$

This condition enables the algorithm to terminate when the minimum solution is equal or minor to the best known solution; otherwise, it continues execution until T_{max} is reached.

4.2.7 Reinitialization of the Population

Past experiments with binary AFSA [23] have shown that, at certain quantity of iterations, all the individual points in a population converge to a non-optimal solution, even after the leaping behavior has been performed. To diversify the search [24], it is proposed to reinitialize the population randomly at every R iterations, keeping the best solution found so far. In practical terms, this technique has greatly improved the quality of the solutions and increased the consistency of the IbAFSA [24].

4.3 Simplified Binary Version of AFSA Applied to SCP

In this section, the main steps of AFSA will be shown and its simplified binary version in order to solve SCP. According to the authors [25] AFSA converges to a non-optimal solution in previous versions like [24]. Therefore, some modifications of AFSA were proposed and they were slightly modified in order to solve SCP.

4.3.1 Features that were Modified in AFSA

In [25] the main modifications were: the "visual scope" concept was rejected; The behavior depends on two probability values; Swarming behavior is never utilized; An effect-based crossover is used instead of an uniform crossover in different behaviors to create trial points; A local search with two steps was implemented; Among other modifications that are explained with more details in [25]. Also, a repair function was introduced for handling the SCP constraints.

Next, the steps of AFSA will be explained in order to obtain SCP results.

4.3.2 Initialization of Population

As many other metaheuristics, it is necessary to initialize the population with objective of finding good solutions. Therefore, the best representation of a population is N current points, $\boldsymbol{x}_i$, where $i \in \{1,2,...,N\}$ each one represented by a binary $0/1$ string of length n and they are randomly generated.

4.3.3 Generation of Trial Population in Simplified AFSA

This metaheuristic works with a trial population at each iteration. So, in order to create trial points in consecutive iterations based on behaviors of random, chasing, and searching is necessary utilize crossover and mutation after the initialization of population. In [25] probabilities of $0 \leq \tau_1 \leq \tau_2 \leq 1$ were introduced and they are the responsible to reach this objective.

Random behavior: If a fish does not have companion in its neighborhood, then it moves randomly looking for it in another region [25]. This happens when a random number $rand(0,1)$ is less than or equal to τ_1. The trial point $\boldsymbol{y}_i$ is created randomly setting $0/1$ bits of length n [25].

Chasing behavior: When a fish, or a group of fish in the swarm, discover food, and the others go quickly after it [25]. This happens when $rand(0,1) \geq \tau_2$ and it is related to the movement towards the best point found so far in the population, $\boldsymbol{x}_{min}$. The trial point $\boldsymbol{y}_i$ is created using an effect-based crossover (see Algorithm 5) between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{min}$ [25].

Searching behavior: When fish discovers a region with more food, by vision or instinct, it goes directly and quickly to that region [25]. This behavior is related to the movement towards a point $\boldsymbol{x}_{rand}$ where "rand" is an index randomly chosen from the set $\{i = 1,2,...,N\}$. When $\tau_1 < rand(0,1) < \tau_2$ it is implemented. An effect-based crossover (see Algorithm 5) between $\boldsymbol{x}_{rand}$ and $\boldsymbol{x}_i$ is utilized to create the trial point $\boldsymbol{y}_i$ [25].

Trial point corresponding to the best point: In [25], the 3 behaviors explained above are implemented to create $N-1$ trial points; the best point $\boldsymbol{x}_{min}$ uses a 4 flip-bit mutation. It is performed on the point $\boldsymbol{x}_{min}$ to create the corresponding trial point $\boldsymbol{y}_i$. In this operation 4 positions are randomly selected, and the bits of the corresponding positions are changed from 0 to 1 or vice versa [25].

4.3.4 The Effect-based Crossover in Simplified Binary Version of AFSA

In order to obtain the trial point in chasing and searching behavior, it necessary to calculate the $effect\ ratio\ ER_{\boldsymbol{u},x_i}$ of u on the current point $\boldsymbol{x}_i$, according to [25] which appears in equation 25 and 26. It can obtain with the following two formulas:

$$ER_{u,x_i} = \frac{q(u)}{q(u)+q(x_i)} \tag{25}$$

$$q(x_i) = \exp[\frac{-(z(x_{min}) - z(x_i))}{(z(x_{min}) - z(x_{max}))}] \tag{26}$$

$u = \boldsymbol{x}_{min}$ is used with chasing behavior, $u = \boldsymbol{x}_{rand}$ is used with searching behavior and $\boldsymbol{x}_{max}$ is the worst point of the population. The effect-based crossover to obtain the trial point $\boldsymbol{y}_i$ is shown in Algorithm 5 according to [25].

Algorithm 5 Effect-based Crossover

Require: current point $\boldsymbol{x}_i$, $\boldsymbol{u}$ and $ER_{\boldsymbol{u},x_i}$

1: **for** $d = 1$ to n **do**
2: **if** $rand(0,1) < ER_{\boldsymbol{u},x_i}$ **then**
3: $y_i^d(t) = u_i^d(t)$
4: **else**
5: $y_i^d(t) = x_i^d(t)$
6: **end if**
7: **end for**
8: **return** trial point y_i

4.3.5 Deal with Constraints of SCP

In order to obtain good results, it is necessary to introduce an appropriate method that helps with SCP constraints. Therefore, a repair function will be shown for handling the SCP constraints in the Algorithm 6. According to [77], Algorithm 6 shows a repair method where all rows not covered are identified and the columns required are added. Hence, in this way all the constraints will be covered. The search of these columns is based on the relationship shown in the equation number 27.

$$\frac{cost\ of\ one\ column}{amount\ of\ columns\ not\ covered} \tag{27}$$

Once the columns are added and the solution is feasible, a method is applied to remove redundant columns of the solution. The redundant columns are those that are removed, the solution remains a feasible solution. The algorithm of this repair method is detailed in the Algorithm 6. Where:

a) I is the set of all rows

b) J is the set of all columns

c) J_i is the set of columns that cover the row $i, i \in I$

d) I_j is the set of rows covered by the column $j, j \in J$

e) S is the set of columns of the solution

f) U is the set of columns not covered

g) w_i is the number of columns that cover the row $i, \forall i \in I$ in S

Algorithm 6 Repair Operator for Dealing with SCP Constraints

$w_i \leftarrow \mid S \cap J_i \mid \forall i \in I$;
$U \leftarrow \{i \mid w_i = 0\}, \forall i \in I$;
for $i \in U$ **do**
    find the first column j in J_i that minimize $\frac{c_j}{|U \cap I_j|} S \leftarrow S \cap j$;
    $w_i \leftarrow w_i + 1, \forall i \in I_j$;
    $U \leftarrow U - I_j$;
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

4.3.6 Selection of New Population

The new population is selected between trial population and current population. Each trial point contends against the current point, therefore, if $z(\boldsymbol{y}_i) \leq z(\boldsymbol{x}_i)$, then the trial point becomes a member of the new population to the next iteration; otherwise, the current point is maintained to the next iteration, according to [25] the criterion that appears in equation 28.

$$\boldsymbol{x}_i^d(t+1) = \begin{cases} \boldsymbol{y}_i^d(t) \; if \; z(\boldsymbol{y}_i^d(t)) \leq z(\boldsymbol{x}_i^d(t)) \\ \\ \boldsymbol{x}_i^d(t) \text{ otherwise} \end{cases} \tag{28}$$

4.3.7 Reinitialization of Current Population

In [25], every certain iterations, this metaheuristic replaces the population of the last iteration with a new population to the next iteration. Therefore, utilizing the same values, it will be done a randomly reinitialization of 50% of the population at every R iterations, where R is a positive integer parameter.

4.3.8 Exploitation or Local Search

Exploitation is related to leaping behavior of AFSA. So, according to the authors [25], in oder to obtain better solutions and improve old versions of AFSA, the concept of exploitation/local search was utilized and its purpose is to find better solutions when the method obtains the same solution as the iterations pass. This is based on a flip-bit mutation which N_{loc} points are selected randomly from the population, where $N_{loc} = \tau_3 N$ with $\tau_3 \in (0,1)$. This mutation changes the bit values of those points from 0 to 1 and vice versa, according to p_m probability. After that, those new points are made feasible by using the repair function of SCP explained in 4.5 *Section*. Then they become members of the population.

Afterwards, the best point of the population is identified and another mutation is operated on N_{ref} positions, with $N_{ref} = \tau_3 n$, those positions are randomly selected from the point [25].Then it becomes a member of the population, if it improves z_{min} at that moment. This mutation is used L times, where L is s a positive integer parameter.

4.3.9 Conditions to Finish the Algorithm

As many other metaheuristics, it is necessary to stop this metaheuristic when it is almost impossible to find a better solution and it is unnecessary to continue wasting the computational resource. Therefore, in accordance with [25], AFSA terminates when the known optimal solution is reached or a maximum number of iterations, T_{max}, is exceeded. The criteria of stopping the algorithm appear in equation 29:

$$t > T_{max} \; or \; z_{min} \leq z_{opt} \tag{29}$$

Where z_{min} is the best objective function value attained at iteration t and z_{opt} is the known optimal value available in the literature.

4.4 Proposed Algorithms

In this section, it will be shown the pseudo-codes for the proposed algorithms and then their flow charts.

4.4.1 Improved Binary Version of AFSA

The pseudocode of the improved version of AFSA for solving the SCP is shown in Algorithm 7.

Algorithm 7 Improved Binary Version of AFSA applied to SCP

Require: T_{max} and z_{opt} and other values of parameters

Set $t = 1$ Initialize population $\boldsymbol{x}_i$, $i = 1,2,...,N$
Execute algorithm 3 and 4, evaluate population and identify x_{min} and z_{min}
while 'termination conditions are not met' **do**
    if $t\%R = 0$ **then**
        Reinitialize population $\boldsymbol{x}_i$, $i = 1,2,...,N$, keeping x_{min} and z_{min}
        Execute algorithm 3 and 4, evaluate population and identify x_{min} and z_{min}
    end if
    for all $\boldsymbol{x}_i$ **do**
        Compute "visual scope" and "crowding factor"
        if "visual scope" is empty **then**
            Execute random behavior to create trial point $\boldsymbol{y}_i$
        else if "visual scope" is not crowded **then**
            if $z(\boldsymbol{x}_{min}) \leq z(\boldsymbol{x}_i)$ **then**
                Execute chasing behavior to create trial point $\boldsymbol{y}_i$
            else if $z(\bar{\boldsymbol{x}}) \leq z(\boldsymbol{x}_i)$ **then**
                Execute swarming behavior to create trial point $\boldsymbol{y}_i$
            else if $z(\boldsymbol{x}_{rand}) \leq z(\boldsymbol{x}_i)$ **then**
                Execute searching behavior to create trial point $\boldsymbol{y}_i$
            else
                Execute random behavior to create trial point $\boldsymbol{y}_i$
            end if
        else if "visual scope" is crowded **then**
            if $z(\boldsymbol{x}_{rand}) \leq z(\boldsymbol{x}_i)$ **then**
                Execute searching behavior to create trial point $\boldsymbol{y}_i$
            else
                Execute random behavior to create trial point $\boldsymbol{y}_i$
            end if
        end if
    end for
    Execute algorithm 3 and 4 to get $\boldsymbol{y}_i$, $i = 1,2,...,N$ and evaluate them
    Select new population to the next iteration, $\boldsymbol{x}_i^d(t+1)$, $i = 1,2,...,N$
    if $t\%L = 0$ **then**
        while $x_{min} = x_{rand}$ **do**
            Set $x_{rand} = \boldsymbol{x}_i$, $i = 1,2,...,N$
            if x_{min} != x_{rand} **then**
                Break
            end if
        end while
        Execute leaping behavior, algorithm 3 and 4, and evaluate it
    end if
    Identify x_{min} and z_{min}
    Set $t = t + 1$
end while
return x_{min} and z_{min}

In the Figure 3, it is possible to see the flow chart for improved version of AFSA:

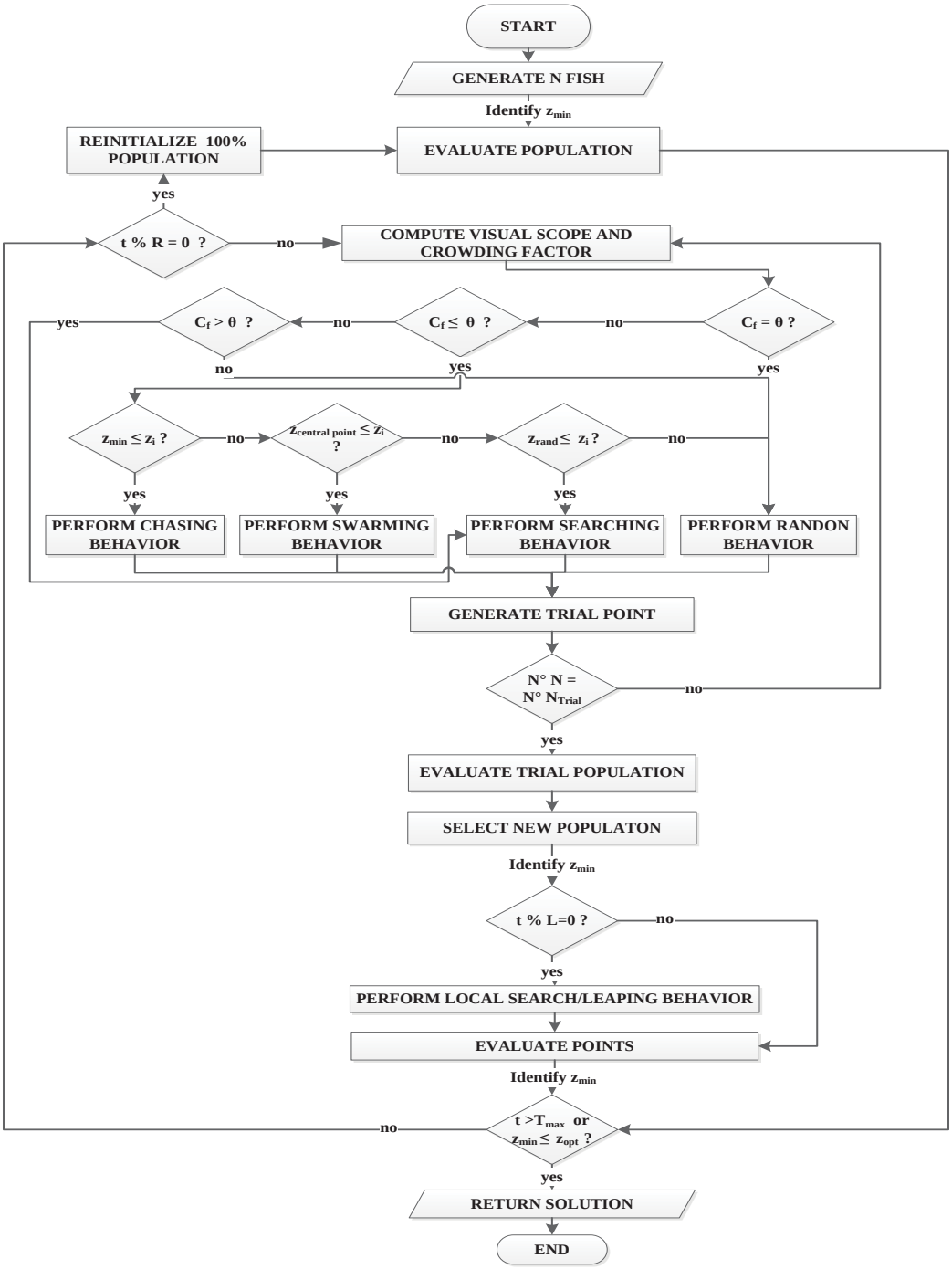

Figure 3: Flow Chart of Improved Binary Version of AFSA in Order to Solve SCP

4.4.2 Simplified Binary Version of AFSA

The pseudocode of the simplified version of AFSA for solving the SCP is shown in Algorithm 8.

Algorithm 8 Simplified Binary Version of AFSA Applied to SCP

Require: T_{max} and z_{opt} and other values of parameters

Set $t = 1$ Initialize population $\boldsymbol{x}_i$, $i = 1,2,...,N$
Execute SCP repair function in order to evaluate the population, identify x_{min} and z_{min}
while $t \leq T_{max}$ *or* $z_{min} \geq z_{opt}$ **do**
    if $t\%R = 0$ **then**
        Reinitialize 50% of the population, keeping x_{min} and z_{min}
        Execute SCP repair function in order to evaluate population, identify x_{min} and z_{min}
    end if
    for $i = 1$ to N **do**
        if $i = x_{min}$ **then**
            Execute 4 flip-bit mutation to create trial point $\boldsymbol{y}_i$
        else
            if $rand(0,1) \leq \tau_1$ **then**
                Execute random behavior to create trial point $\boldsymbol{y}_i$
            else if $rand(0,1) \geq \tau_2$ **then**
                Execute chasing behavior to create trial point $\boldsymbol{y}_i$
            else
                Execute searching behavior to create trial point $\boldsymbol{y}_i$
            end if
        end if
    end for
    Execute SCP repair function in order to evaluate and get $\boldsymbol{y}_i$, $i = 1,2,...,N$
    Select new population to the next iteration, $\boldsymbol{x}_i^d(t+1)$, $i = 1,2,...,N$
    if $t\%L = 0$ **then**
        Execute exploitation/local search - leaping behavior
        Identify x_{min} and z_{min}
    end if
    Set $t = t + 1$
end while
return x^{min} and z_{min}

In the Figure 4, it is possible to see the flow chart for simplified version of AFSA:

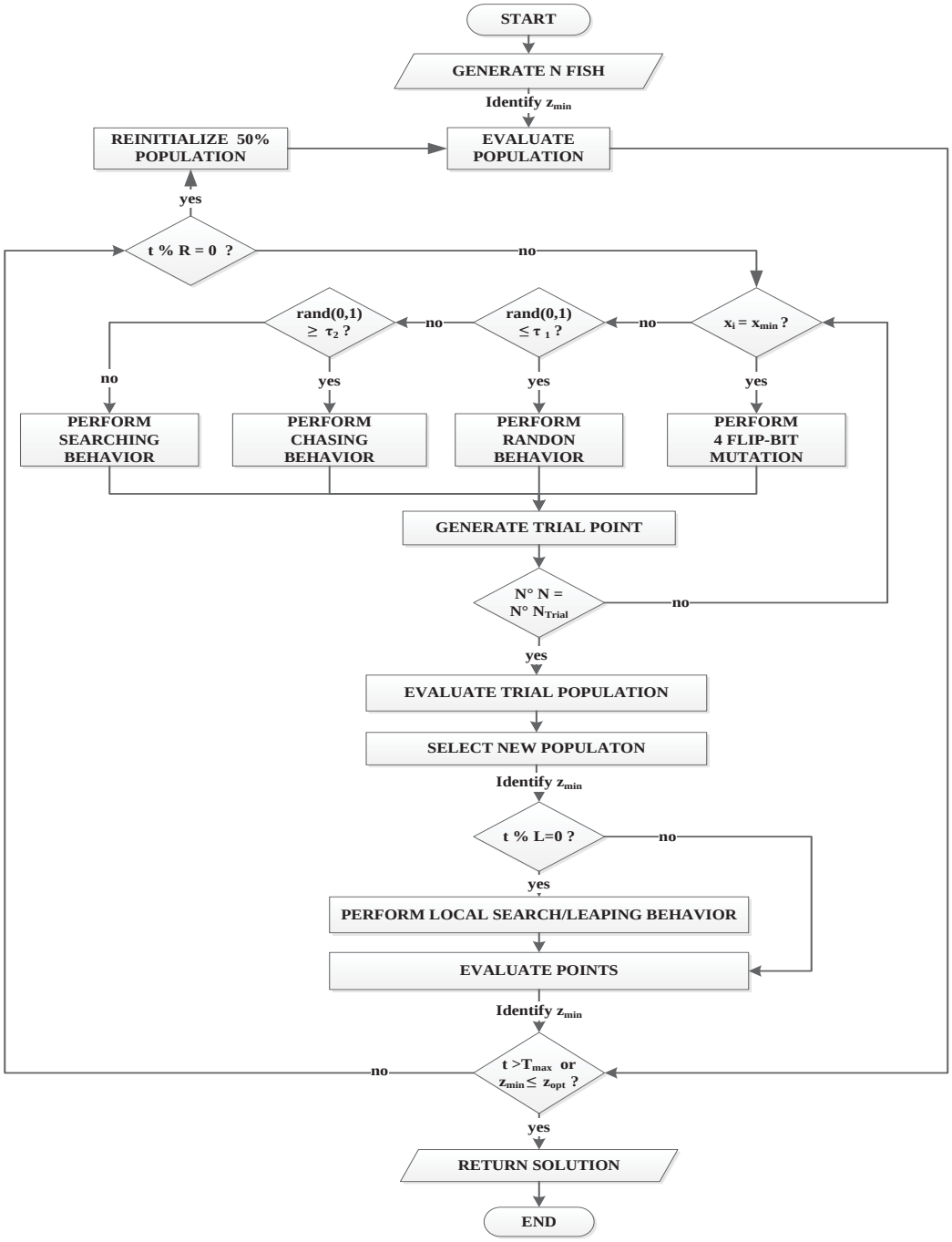

Figure 4: Flow Chart of Simplified Binary Version of AFSA in Order to Solve SCP

4.4.3 Symbology of AFSA Algorithms

Here, it will be explained the symbology of each version of AFSA shown above. In the first place, improved version of AFSA and then simplified version of AFSA.

Improved Binary Version of AFSA:

- N: Population of fish.
- z_{min}: Minimum solution found at that moment.
- t: Iteration.
- R: Reinitialization of the population at iteration t.
- C_f: Crowding factor.
- θ: Crowd parameter.
- z_i: Fitness values of point or fish i at that moment.
- $z_{central point}$: Fitness values of central point at that moment.
- z_{rand}: Fitness values of point or fish i, randomly selected of the population N, at that moment.
- $N°\ N$: Number of members of the population.
- $N°\ N_{Trial}$: Number of members of the trial population.
- L: Leaping behavior at iteration t.
- T_{max}: Maximum number of iterations.
- z_{opt}: Best known solution.

Simplified Binary Version of AFSA:

- N: Population of fish.
- z_{min}: Minimum solution found at that moment.
- t: Iteration.
- R: Reinitialization of the population at iteration t.
- $rand(0,1)$: Random number between 0 and 1.
- τ_1: Probability τ_1.
- τ_2: Probability τ_2.
- x_i: Point or fish i.
- x_{min}: Point or fish of the minimum solution found at that moment.
- $N°\ N$: Number of members of the population.
- $N°\ N_{Trial}$: Number of members of the trial population.

- L: Local search at iteration t.
- T_{max}: Maximum number of iterations.
- z_{opt}: Best known solution.

5 Chapter 5: Results

5.1 Experimental Results

After many experiments, the obtained results after performing AFSA will be shown to solve SCP. At the middle of this section it is possible to find the Table 4 and Table 5, which show the results of SCP with more details.

5.1.1 Experiments

In related works, algorithms have been run 30 times for each instance, also it is an accepted number in the literature. Therefore, this proposed algorithm was run with that number of times. Moreover, this algorithm needs almost 20 hours to analyze the last 10 files, sets from *NRG* to *NRH*, because each one has a large matrix with a great deal of rows and columns. This should not be considered a problem in academic area but it could be taken into account as a problem if it is applied in a real problem due to the large number of hours.

These algorithms tested the 70 data files from the OR-Library [26], 25 of them are the instance sets 4,5,6 was originally from Balas and Ho [78], the others 25, the sets A, B, C, D, E from Beasley [77] and 20 of these data files are the test problem sets E, F, G, H from Beasley [79]. These 70 files are formatted as: number of rows n, number of columns m, the cost of each column $c_j, j \in \{1,\ldots,n\}$, and for each row $i, i \in \{1,...,m\}$ the number of columns which cover row i followed by a list of the columns which cover rows i. The characteristics of the instances are summarized in Tables 1.

Table 1 contains the data set, the number of files of each data set, the number of rows and columns of each file of the set, and finally the density of the each set.

Table 1: Characteristics of the Data Sets

Data Set	*Number of Files*	*Number of Rows*	*Number of Columns*	*Density (%)*
4	10	200	1000	2
5	10	200	2000	2
6	5	200	1000	5
A	5	300	3000	2
B	5	300	3000	5
C	5	400	4000	2
D	5	400	4000	5
E	5	50	500	20
NRE	5	500	5000	10
NRF	5	500	5000	20
NRG	5	1000	10000	2
NRH	5	1000	10000	5

AFSA algorithms were implemented in Java programming language, using Eclipse IDE, with the following hardware: Intel core i5 dual core 2.60 GHz processor, 8 GB RAM and it was run under OSX Yosemite. Furthermore, parameters that will be shown in the following paragraphs were obtained experimentally. First, they were probed with parameters that were proposed by authors and then the value of each parameter was modified until obtain the final configuration

through different experiments with the algorithms.

The first experiment of improved version of AFSA was utilized with Algorithm 3 and Algorithm 4 in order to deal with SCP constraints. Additionally, it was run 30 times for most instances and 20, 10 and 5 times for the instances which have a great deal of rows and columns, especially for the last 20 sets, because of the number of hours that this algorithm needs to evaluate each instance. Then the average minimum values were obtained for these 30, 20, 10 and 5 times. Also, this program was executed only with feasible solutions and with the following configuration of parameters:

- A population of $N = 20$ fish.
- Visual parameter $\delta = 0.98$.
- Crowd parameter $\theta = 0.98$.
- Probability $p_m = 0.1$.
- Leaping $L = 5$.
- Reinitialization of population $R = 10$.
- Each trial was run 700 iterations.

The overview of this experiment is shown in Table 2 and with more detail in Table 4.

Similar to the first algorithm, the first experiment of simplified version of AFSA was utilized with Algorithm 6 as repair function. Additionally, it was run 30 times for most instances and 20 times for other instances. After all results, the average minimum values were obtained from these 30 and 20 times for each one of the files. Moreover, the program was executed only with feasible solutions and with the following configuration of parameters:

- A population of $N = 20$ fish.
- Probability $\tau_1 = 0.1$.
- Probability $\tau_2 = 0.9$.
- Probability $\tau_3 = 0.1$.
- Probability $p_m = 0.1$.
- Leaping $L = 50$.
- Reinitialization of population $R = 10$.
- Each trial was run 1000 iterations.

The summary of this experiment is shown in Table 2 and with more details in Table 4.

The second experiment of improved version of AFSA was utilized with Algorithm 6 as repair function. Additionally, it was run 30 times for all instances. Then the average minimum values were obtained from these 30 times. Also, this program was executed only with feasible solutions and with the following configuration of parameters:

- A population of $N = 20$ fish.
- Visual parameter $\delta = 0.98$.
- Crowd parameter $\theta = 0.98$.
- Probability $p_m = 0.1$.
- Leaping $L = 5$.
- Reinitialization of population $R = 10$.
- Each trial was run 2000 iterations.

The overview of this experiment is shown in Table 3 and with more details in Table 5.

Likewise, the second experiment of simplified version of AFSA was utilized with Algorithm 6 as repair function. Additionally, it was run 30 times for all instances. Then the average minimum values were obtained from these 30 times. Moreover, the program was executed only with feasible solutions and with the following configuration of parameters:

- A population of $N = 45$ fish.
- Probability $\tau_1 = 0.1$.
- Probability $\tau_2 = 0.9$.
- Probability $\tau_3 = 0.1$.
- Probability $p_m = 0.1$.
- Leaping $L = 30$.
- Reinitialization of population $R = 5$.
- Each trial was run 1500 iterations.

The summary of this experiment is shown in Table 3 and with more details in Table 5.

Table 2 (Experiment 1) and Table 3 (Experiment 2) contain the data set, the number of files of each data set and finally the average *RPD* of improved version of AFSA and simplified version of AFSA of each set receptively. *RPD* is shown in equation 30.

Table 4 and Table 5 show the results of each set analyzed. Each table shows, in the first column, the number of experiment of each instance. Then, *Instance*, indicates each benchmark evaluated, and Z_{opt} shows the best known solution value of each instance. The second and third column, $Improved\ AFSA$ and $Simplified\ AFSA$ respectively show Z_{min}, Z_{max}, Z_{avg} which represent the minimum, maximum among minimums, and average of minimums solutions obtained. The last value reports the relative percentage deviation *RPD* which represents the deviation of the objective value (best known solution) f_{opt} from f_{min} which is the minimum value obtained for each instance. *RPD* was calculated as follows in equation 30:

$$\text{RPD} = \frac{100(f_{min} - f_{opt})}{f_{opt}} \tag{30}$$

Table 2: Average RPD of Each Data Set - Experiment 1

Data Set	*Number of Files*	*Average RPD Improved AFSA*	*Average RPD Simplified AFSA*
4	10	28,23	0,84
5	10	34,13	0,83
6	5	36,53	0,78
A	5	38,91	1,83
B	5	43,96	2,92
C	5	40,37	2,29
D	5	44,18	3,65
E	5	64	0,0
NRE	5	39,49	4,93
NRF	5	50,48	7,16
NRG	5	48,33	5,29
NRH	5	51,56	7,09

Table 3: Average RPD of Each Data Set - Experiment 2

Data Set	*Number of Files*	*Average RPD Improved AFSA*	*Average RPD Simplified AFSA*
4	10	0,25	0,15
5	10	0,59	0,13
6	5	0,14	0,14
A	5	0,75	0,75
B	5	1,32	0,0
C	5	1,30	1,03
D	5	2,07	1,46
E	5	0,0	0,0
NRE	5	3,58	2,83
NRF	5	7,25	10,12
NRG	5	5,03	3,04
NRH	5	9,09	5,34

Table 4: Results of Non-Unicost SCP benchmarks (4, 5, 6, A, B, C, D, E, NRE, NRF, NRG and NRH sets) - Experiment 1

			Improved AFSA				*Simplified AFSA*			
Number	*Instance*	Z_{opt}	Z_{min}	Z_{max}	Z_{avg}	RPD	Z_{min}	Z_{max}	Z_{avg}	RPD
1	4.1	429	487	497	488,83	13,52	430	445	437,4	0,23
2	4.2	512	679	703	681,7	32,62	515	546	530,83	0,59
3	4.3	516	638	681	652,64	23,64	519	543	528,27	0,58
4	4.4	494	637	687	649,95	28,95	495	532	514,83	0,20
5	4.5	512	687	698	687,7	34,18	514	536	521,73	0,39
6	4.6	560	732	739	732,23	30,71	565	597	580,9	0,89
7	4.7	430	570	570	570	32,56	432	447	437,37	0,47
8	4.8	492	613	613	613	24,59	492	514	501,73	0,0
9	4.9	641	876	915	883,7	36,66	658	688	669,8	2,65
10	4.10	514	642	642	642	24,9	525	559	539,6	2,14
11	5.1	253	320	320	320	26,48	254	271	263,03	0,40
12	5.2	302	443	443	443	46,69	310	318	314,27	2,65
13	5.3	226	278	278	278	23,01	228	244	232,77	0,88
14	5.4	242	306	306	306	26,45	242	247	244,77	0,0
15	5.5	211	280	280	280	32,7	212	215	212,6	0,47
16	5.6	213	309	309	309	45,07	214	242	227,77	0,47
17	5.7	293	406	406	406	38,57	299	315	307,9	2,05
18	5.8	288	397	448	415,9	37,85	291	313	298,97	1,04
19	5.9	279	383	383	383	37,28	279	296	285,73	0,0
20	5.10	265	337	337	337	27,17	266	276	272,07	0,38
21	6.1	138	190	190	190	37,68	138	153	146,37	0,0
22	6.2	146	223	223	223	52,74	149	156	151,97	2,05
23	6.3	145	195	195	195	34,48	145	161	149,63	0,0
24	6.4	131	157	157	157	19,85	131	137	134,17	0,0
25	6.5	161	222	222	222	37,88	164	181	172,67	1,86
26	A.1	253	338	338	338	33,6	256	270	259,6	1,19
27	A.2	252	365	372	365,7	44,84	258	276	264,4	2,38
28	A.3	232	308	3011	308,6	32,76	235	255	246,2	1,29
29	A.4	234	232	232	232	38,03	243	266	252,25	3,85
30	A.5	236	343	343	343	45,34	237	259	244,9	0,42
31	B.1	69	97	97	97	40,58	72	88	78,3	4,35
32	B.2	76	112	112	112	47,37	79	94	84,67	3,95
33	B.3	80	108	108	108	35	82	89	85,6	2,5
34	B.4	79	116	116	116	46,84	82	96	86,45	3,80
35	B.5	72	108	108	108	50	72	89	79,5	0,0
36	C.1	227	295	300	295,4	26,96	231	252	238,85	1,76
37	C.2	219	301	303	301,07	37,44	227	254	236,5	3,65
38	C.3	243	353	401	370,97	45,27	251	274	263,15	3,29
39	C.4	219	316	316	316	44,29	223	253	240,1	1,,83
40	C.5	215	318	318	318	47,91	217	250	228,3	0,93
41	D.1	60	89	89	89	48,33	60	81	66,6	0,0
42	D.2	66	93	93	93	40,91	69	83	73,35	4,54
43	D.3	72	107	107	107	48,61	76	87	82,4	5,56
44	D.4	62	83	83	83	33,87	64	76	69,05	3,23
45	D.5	61	91	91	91	49,18	64	78	68,9	4,92
46	E.1	5	9	9	9	80	5	6	5,87	0,0
47	E.2	5	10	10	10	100	5	6	5,5	0,0
48	E.3	5	7	7	7	40	5	6	5,2	0,0
49	E.4	5	6	6	6	20	5	6	5,7	0,0
50	E.5	5	9	9	9	80	5	6	5,57	0,0
51	NRE.1	29	37	37	37	27,59	29	39	32,1	0,0
52	NRE.2	30	45	45	45	50	32	40	32,25	6,67
52	NRE.3	27	42	42	42	55,56	28	35	32,1	3,70
54	NRE.4	28	37	37	37	32,14	30	38	33,05	7,14
55	NRE.5	28	37	37	37	32,14	30	35	31,95	7,14
56	NRF.1	14	20	20	20	42,86	15	18	16,75	7,14
57	NRF.2	15	18	18	18	20	16	18	17,05	6,67
58	NRF.3	14	25	25	25	78,57	15	20	17,25	7,14
59	NRF.4	14	22	22	22	57,14	15	19	16,45	7,14
60	NRF.5	13	20	20	20	53,85	14	18	15,75	7,69
61	NRG.1	176	261	264	261,43	48,3	184	249	194,05	4,54
62	NRG.2	154	243	243	243	57,79	162	170	166,5	5,19
63	NRG.3	166	228	231	228,35	37,34	174	268	184,7	4,82
64	NRG.4	168	252	252	252	50	178	284	190,55	5,95
65	NRG.5	168	249	249	249	48,21	178	344	193,6	5,95
66	NRH.1	63	99	99	99	57,14	66	100	72,15	4,76
67	NRH.2	63	98	98	98	55,56	66	129	72,0	4,76
68	NRH.3	59	93	93	93	57,63	66	79	68,7	11,86
69	NRH.4	58	84	84	84	44,83	63	123	70,5	8,62
70	NRH.5	55	79	79	79	42,64	58	71	60,7	5,45

Table 5: Results of Non-Unicost SCP benchmarks (4, 5, 6, A, B, C, D, E, NRE, NRF, NRG and NRH sets) - Experiment 2

			Improved AFSA				*Simplified AFSA*			
Number	*Instance*	Z_{opt}	Z_{min}	Z_{max}	Z_{avg}	RPD	Z_{min}	Z_{max}	Z_{avg}	RPD
1	4.1	429	430	443	435,1	0,23	430	443	435,97	0,23
2	4.2	512	512	540	521,73	0,0	512	552	525,77	0,0
3	4.3	516	516	532	519,2	0,0	516	528	520,27	0,0
4	4.4	494	495	510	504,23	0,20	495	521	508,63	0,20
5	4.5	512	514	526	517,7	0,39	512	526	517,93	0,0
6	4.6	560	560	590	568,53	0,0	560	592	571,6	0,0
7	4.7	430	431	441	434,6	0,23	432	441	434,83	0,47
8	4.8	492	493	505	498,17	0,20	492	506	498,53	0,0
9	4.9	641	649	677	659,93	1,25	645	684	661,83	0,62
10	4.10	514	514	541	522,73	0,0	514	545	520,77	0,0
11	5.1	253	255	276	263,23	0,79	253	272	262,03	0,0
12	5.2*	302	311	317	312,4	2,98	306	318	311,93	1,32
13	5.3	226	228	241	230,43	0,88	226	243	230,87	0,0
14	5.4	242	242	248	243,83	0,0	242	249	243,83	0,0
15	5.5	211	211	216	211,93	0,0	211	217	212,43	0,0
16	5.6	213	214	239	220,83	0,47	213	235	221,9	0,0
17	5.7	293	293	314	304,23	0,0	293	313	304,33	0,0
18	5.8	288	288	310	297,53	0,0	288	305	296,57	0,0
19	5.9	279	279	295	284,7	0,0	279	292	283,37	0,0
20	5.10	265	267	275	271,2	0,75	265	274	270,93	0,0
21	6.1	138	138	151	143,73	0,0	138	149	144,13	0,0
22	6.2*	146	147	154	150,73	0,68	147	155	150,77	0,68
23	6.3	145	145	157	150,17	0,0	145	156	150,0	0,0
24	6.4	131	131	139	133,0	0,0	131	137	132,6	0,0
25	6.5	161	161	180	167,67	0,0	161	180	170,47	0,0
26	A.1*	253	253	265	258,43	0,0	254	264	257,73	0,40
27	A.2	252	255	276	263,33	1,19	254	276	263,57	0,79
28	A.3	232	235	248	240,23	1,29	235	249	240,4	1,29
29	A.4	234	235	252	242,03	0,43	236	249	242,73	0,85
30	A.5	236	238	245	240,03	0,85	237	244	239,37	0,42
31	B.1	69	70	83	75,3	1,45	69	81	74,77	0,0
32	B.2	76	78	89	83,0	2,63	76	90	83,47	0,0
33	B.3	80	80	88	83,23	0,0	80	88	83,37	0,0
34	B.4	79	81	88	84,03	2,53	79	88	83,9	0,0
35	B.5	72	72	80	74,17	0,0	72	81	75,17	0,0
36	C.1	227	229	237	233,57	0,88	228	238	232,83	0,44
37	C.2	219	220	233	225,63	0,46	220	236	225,67	0,46
38	C.3	243	250	275	256,83	2,88	250	274	256,47	2,88
39	C.4	219	222	240	230,57	1,37	222	239	231,47	1,37
40	C.5	215	217	227	219,57	0,93	215	225	218,8	0,0
41	D.1	60	60	70	63,43	0,0	60	64	61,6	0,0
42	D.2	66	67	79	71,5	1,52	67	73	70,13	1,52
43	D.3	72	76	86	78,3	5,56	75	80	77,63	4,17
44	D.4	62	63	74	65,93	1,61	63	67	65,03	1,61
45	D.5	61	62	73	64,93	1,64	61	66	63,73	0,0
46	E.1	5	5	6	5,87	0,0	5	6	5,93	0,0
47	E.2	5	5	6	5,2	0,0	5	6	5,17	0,0
48	E.3	5	5	6	5,13	0,0	5	6	5,03	0,0
49	E.4	5	5	6	5,53	0,0	5	6	5,37	0,0
50	E.5	5	5	6	5,73	0,0	5	6	5,2	0,0
51	NRE.1	29	29	35	31,2	0,0	29	30	29,47	0,0
52	NRE.2	30	31	37	33,57	3,33	31	34	32,53	3,33
52	NRE.3	27	29	35	30,83	7,41	28	32	30,07	3,70
54	NRE.4	28	30	36	32,37	7,14	29	33	30,7	3,57
55	NRE.5	28	28	40	31,07	0,0	29	32	29,6	3,57
56	NRF.1	14	15	17	15,8	7,14	15	17	15,83	7,14
57	NRF.2	15	15	19	16,67	0,0	16	18	16,4	6,67
58	NRF.3*	14	16	21	17,33	14,29	16	18	16,47	14,29
59	NRF.4	14	15	17	16,0	7,14	15	17	15,83	7,14
60	NRF.5*	13	14	17	15,53	7,69	15	16	15,63	15,38
61	NRG.1	176	187	108	196,37	6,25	180	195	187,4	2,27
62	NRG.2	154	161	178	168,23	4,55	161	168	164,13	4,55
63	NRG.3	166	174	186	179,83	4,82	171	180	176,1	3,01
64	NRG.4	168	175	195	182,73	4,17	171	183	177,87	1,79
65	NRG.5	168	177	195	184,47	5,36	174	185	179,73	3,57
66	NRH.1*	63	68	82	74,6	7,94	67	70	68,53	6,35
67	NRH.2	63	68	85	74,23	7,94	66	74	68,67	4,76
68	NRH.3	59	66	79	71,97	11,86	63	69	66,17	6,78
69	NRH.4	58	63	76	68,33	8,62	61	66	63,6	5,17
70	NRH.5	55	60	71	65,63	9,09	57	64	59,8	3,64

The symbol "$*$" means that it was possible to find a better solution in experiment 2 of simplified version of AFSA for those instances with a specific configuration of parameters. The results are shown in Table 6 and the configuration of parameters were the following:

The configuration of parameters for SCP 5.2 was the following:

- A population of $N = 45$ fish.
- Probability $\tau_1 = 0.3$.
- Probability $\tau_2 = 0.7$.
- Probability $\tau_3 = 0.2$.
- Probability $p_m = 0.1$.
- Leaping $L = 30$.
- Reinitialization of population $R = 5$.
- The trial was run 2100 iterations, 30 times.

The configuration of parameters for SCP 6.2 was the following:

- A population of $N = 45$ fish.
- Probability $\tau_1 = 0.3$.
- Probability $\tau_2 = 0.7$.
- Probability $\tau_3 = 0.1$.
- Probability $p_m = 0.1$.
- Leaping $L = 30$.
- Reinitialization of population $R = 5$.
- Each trial was run 1800 iterations, 30 times.

The configuration of parameters for SCP $A.1$ was the following:

- A population of $N = 45$ fish.
- Probability $\tau_1 = 0.2$.
- Probability $\tau_2 = 0.8$.
- Probability $\tau_3 = 0.1$.
- Probability $p_m = 0.1$.
- Leaping $L = 50$.
- Reinitialization of population $R = 30$.
- Each trial was run 1500 iterations, 30 times.

The configuration of parameters for SCP *NRF*.3, *NRF*.5 and *NRH*.1 was the following:

- A population of $N = 20$ fish.
- Probability $\tau_1 = 0.1$.
- Probability $\tau_2 = 0.9$.
- Probability $\tau_3 = 0.1$.
- Probability $p_m = 0.1$.
- Leaping $L = 50$.
- Reinitialization of population $R = 10$.
- Each trial was run 1000 iterations, 30 times.

Table 6: Results of Non-Unicost SCP benchmarks 5.2, 6.2, *A*.1, *NRF*.3, *NRF*.5 and *NRH*.1- Experiment 3

			Simplified AFSA			
Number	*Instance*	Z_{opt}	Z_{min}	Z_{max}	Z_{avg}	RPD
1	5.2	302	305	321	311,8	0,99
2	6.2	146	146	160	150,9	0,0
3	A.1	253	253	263	258,63	0,0
4	NRF.3	14	15	20	17,25	7,14
5	NRF.5	13	14	18	15,75	7,69
6	NRH.1	63	66	100	72,15	4,76

With the second experiment of improved and simplified version of AFSA, the time of processing was calculated of each SCP instance. Nevertheless, each time, it was calculated with same configuration of parameters which were shown in Table 5. The overview of those times can see in Table 7 and with more details in Table 8.

Note: Each "processing time" was extrapolated. For that reason each "time" is an approximation.

Table 7: Average Processing Time of Each Data Set - Experiment 2

Data Set	*Number of Files*	*Average Time Improved AFSA (s)*	*Average Time Simplified AFSA (s)*	*Average Time Improved AFSA (min)*	*Average Time Simplified AFSA (min)*
4	10	46,4	56,8	0,77	0,95
5	10	120,3	98,2	2,01	1,64
6	5	45,6	45,2	0,76	0,77
A	5	213,6	210	3,56	3,50
B	5	201,8	185,2	3,36	3,09
C	5	328,2	366,4	5,47	6,11
D	5	309,4	324,4	5,16	5,41
E	5	11,2	9,2	0,19	0,15
NRE	5	453,2	527,4	7,55	8,79
NRF	5	480	589,6	8,00	9,83
NRG	5	1386,6	2415,4	22,81	40,22
NRH	5	1394,6	2394,4	23,24	39,91

Table 8: Results of Processing Time for Non-Unicost SCP benchmarks (4, 5, 6, A, B, C, D, E, NRE, NRF, NRG and NRH sets) - Experiment 2

Number	*Instance*	*Average Time Improved AFSA (s)*	*Average Time Simplified AFSA (s)*	*Average Time Improved AFSA (min)*	*Average Time Simplified AFSA (min)*
1	4.1	43	57	0,72	0,95
2	4.2	43	60	0,72	1,00
3	4.3	59	58	0,98	0,97
4	4.4	43	47	0,72	0,78
5	4.5	43	61	0,72	1,02
6	4.6	43	59	0,72	0,98
7	4.7	43	58	0,72	0,97
8	4.8	43	46	0,72	0,77
9	4.9	61	65	1,02	1,08
10	4.10	43	57	0,72	0,95
11	5.1	118	97	1,97	1,62
12	5.2	118	99	1,97	1,65
13	5.3	116	96	1,93	1,60
14	5.4	117	97	1,95	1,62
15	5.5	148	97	2,47	1,62
16	5.6	118	99	1,97	1,65
17	5.7	114	99	1,90	1,65
18	5.8	119	100	1,98	1,67
19	5.9	116	100	1,93	1,67
20	5.10	119	98	1,98	1,63
21	6.1	43	47	0,72	0,78
22	6.2	43	46	0,72	0,77
23	6.3	41	45	0,68	0,75
24	6.4	42	46	0,70	0,77
25	6.5	59	47	0,98	0,78
26	A.1	216	214	3,60	3,57
27	A.2	212	214	3,53	3,57
28	A.3	211	205	3,52	3,42
29	A.4	212	205	3,53	3,42
30	A.5	217	212	3,62	3,53
31	B.1	202	181	3,37	3,02
32	B.2	198	190	3,30	3,17
33	B.3	198	184	3,30	3,07
34	B.4	204	183	3,40	3,05
35	B.5	207	188	3,45	313
36	C.1	333	360	5,55	6,00
37	C.2	336	357	5,60	5,95
38	C.3	324	384	5,40	6,40
39	C.4	320	366	5,33	6,10
40	C.5	328	365	5,47	6,08
41	D.1	303	324	5,05	5,40
42	D.2	305	322	5,08	5,37
43	D.3	306	317	5,10	5,28
44	D.4	318	327	5,30	5,45
45	D.5	315	332	5,25	5,53
46	E.1	13	8	0,22	0,13
47	E.2	6	8	0,10	0,13
48	E.3	9	14	0,15	0,23
49	E.4	15	8	0,25	0,13
50	E.5	13	8	0,22	0,13
51	NRE.1	444	535	7,40	8,92
52	NRE.2	462	542	7,70	9,03
52	NRE.3	445	524	7,42	8,73
54	NRE.4	450	518	7,50	8,63
55	NRE.5	465	518	7,75	8,63
56	NRF.1	492	587	8,20	9,78
57	NRF.2	480	599	8,00	9,98
58	NRF.3	475	599	7,92	9,98
59	NRF.4	477	569	7,95	9,48
60	NRF.5	476	594	9,93	9,90
61	NRG.1	1415	2427	23,58	40,45
62	NRG.2	1227	2404	20,45	40,07
63	NRG.3	1422	2386	23,70	39,77
64	NRG.4	1405	2430	23,42	40,50
65	NRG.5	1374	2430	22,90	40,33
66	NRH.1	1395	2394	23,25	39,90
67	NRH.2	1460	2364	24,33	39,40
68	NRH.3	1410	2419	23,50	40,32
69	NRH.4	1340	2337	22,33	38,95
70	NRH.5	1368	2458	22,80	40,97

In spite of the fact that these two AFSA algorithms were utilized in order to analyze 70 files each one. The criteria to create the Table 10, which shows the best results obtained, was the following and in this order; First, the best RPD value between them, in the second place, the lowest value between the average of minimums and finally the lowest value between the maximums obtained. Thereby, with these two versions of the algorithm AFSA, it was obtained 140 results, among which it was chosen the best 70 results. Table 9 contains the summary of the Table 10 and it is the following:

Table 9: Summary of Average RPD of Each Data Set

Data Set	*Number of Files*	*Average RPD AFSA*	*Average Execution Time* (s)
4	10	0,128	50.3
5	10	0,099	104,8
6	5	0,0	47,2
A	5	0,568	213
B	5	0,0	191,8
C	5	1,03	353
D	5	1,46	324,4
E	5	0,0	10,2
NRE	5	2,12	516,8
NRF	5	5,822	523,2
NRG	5	3,038	2415,4
NRH	5	5,022	2394,4

Table 10: Best Results of Non-Unicost SCP benchmarks (4, 5, 6, A, B, C, D, E, NRE, NRF, NRG and NRH sets)

Number	*Instance*	Z_{opt}	Z_{min}	Z_{max}	Z_{avg}	RPD	*Execution Time (s)*	*AFSA Version*
1	4.1	429	430	443	435,1	0,23	43	Improved
2	4.2	512	512	540	521,73	0,0	43	Improved
3	4.3	516	516	532	519,2	0,0	59	Improved
4	4.4	494	495	510	504,23	0,20	43	Improved
5	4.5	512	512	526	517,93	0,0	61	Simplified
6	4.6	560	560	590	568,53	0,0	43	Improved
7	4.7	430	431	441	434,6	0,23	43	Improved
8	4.8	492	492	506	498,53	0,0	46	Simplified
9	4.9	641	645	684	661,83	0,62	65	Simplified
10	4.10	514	514	545	520,77	0,0	57	Simplified
11	5.1	253	253	272	262,03	0,0	97	Simplified
12	5.2	302	305	321	311,8	0,99	99	Simplified
13	5.3	226	226	243	230,87	0,0	96	Simplified
14	5.4	242	242	248	243,83	0,0	97	Simplified
15	5.5	211	211	216	211,93	0,0	148	Improved
16	5.6	213	213	235	221,9	0,0	99	Simplified
17	5.7	293	293	314	304,23	0,0	114	Improved
18	5.8	288	288	305	296,57	0,0	100	Simplified
19	5.9	279	279	292	283,37	0,0	100	Simplified
20	5.10	265	265	274	270,93	0,0	98	Simplified
21	6.1	138	138	151	143,73	0,0	43	Improved
22	6.2	146	146	160	150,9	0,0	43	Improved
23	6.3	145	145	156	150,0	0,0	45	Simplified
24	6.4	131	131	137	132,6	0,0	46	Simplified
25	6.5	161	161	180	167,67	0,0	59	Improved
26	A.1	253	253	265	258,43	0,0	216	Improved
27	A.2	252	254	276	263,57	0,79	214	Simplified
28	A.3	232	235	248	240,23	1,29	211	Improved
29	A.4	234	235	252	242,03	0,43	212	Improved
30	A.5	236	237	244	239,37	0,42	212	Simplified
31	B.1	69	69	81	74,77	0,0	181	Simplified
32	B.2	76	76	90	83,47	0,0	190	Simplified
33	B.3	80	80	88	83,23	0,0	198	Improved
34	B.4	79	79	88	83,9	0,0	183	Simplified
35	B.5	72	72	80	74,17	0,0	207	Improved
36	C.1	227	228	238	232,83	0,44	360	Simplified
37	C.2	219	220	233	225,63	0,46	336	Improved
38	C.3	243	250	274	256,47	2,88	384	Simplified
39	C.4	219	222	240	230,57	1,37	320	Improved
40	C.5	215	215	225	218,8	0,0	365	Simplified
41	D.1	60	60	64	61,6	0,0	324	Simplified
42	D.2	66	67	73	70,13	1,52	322	Simplified
43	D.3	72	75	80	77,63	4,17	317	Simplified
44	D.4	62	63	67	65,03	1,61	327	Simplified
45	D.5	61	61	66	63,73	0,0	332	Simplified
46	E.1	5	5	6	5,87	0,0	13	Improved
47	E.2	5	5	6	5,17	0,0	8	Simplified
48	E.3	5	5	6	5,03	0,0	14	Simplified
49	E.4	5	5	6	5,37	0,0	8	Simplified
50	E.5	5	5	6	5,2	0,0	8	Simplified
51	NRE.1	29	29	30	29,47	0,0	535	Simplified
52	NRE.2	30	31	34	32,53	3,33	542	Simplified
52	NRE.3	27	28	32	30,07	3,70	524	Simplified
54	NRE.4	28	29	33	30,7	3,57	518	Simplified
55	NRE.5	28	28	40	31,07	0,0	465	Improved
56	NRF.1	14	15	17	15,8	7,14	492	Improved
57	NRF.2	15	15	19	16,67	0,0	480	Improved
58	NRF.3	14	15	20	17,25	7,14	599	Simplified
59	NRF.4	14	15	17	15,83	7,14	569	Simplified
60	NRF.5	13	14	18	15,75	7,69	476	Improved
61	NRG.1	176	180	195	187,4	2,27	2427	Simplified
62	NRG.2	154	161	168	164,13	4,55	2404	Simplified
63	NRG.3	166	171	180	176,1	3,01	2386	Simplified
64	NRG.4	168	171	183	177,87	1,79	2430	Simplified
65	NRG.5	168	174	185	179,73	3,57	2430	Simplified
66	NRH.1	63	66	100	72,15	4,76	2394	Simplified
67	NRH.2	63	66	74	68,67	4,76	2364	Simplified
68	NRH.3	59	63	69	66,17	6,78	2419	Simplified
69	NRH.4	58	61	66	63,6	5,17	2337	Simplified
70	NRH.5	55 9,09	57	64	59,8	3,64	2458	Simplified

5.1.2 Statistical Analysis

In the last part of this work, an analysis was performed in order to obtain which one of these two versions of AFSA has obtained better results. This kind of analysis appears in [80] and the authors compare different metaheuristics with statistical analysis. Therefore, in the following paragraphs there are a statistical analyses which compare improved and simplified version of AFSA.

First of all, it is necessary to perform an analysis of outliers with the 30 independent executions for each instance and for both versions of AFSA. After that, the outliers obtained were erased of the samples, because they are data that do not follow the normal behavior of the algorithms. So, the outliers were computed with Microsoft Excel and according to these steps:

1. The median of data.
2. The first and the third quartile of the data (Q_1 and Q_3).
3. The interquartile range (IQR).
4. Then the extreme and mild outliers were calculated according to these two equations:

$$Mild\ Outlier : value < Q_1 - 1,5\ IQR\ or\ > Q_3 + 1,5\ IQR, \tag{31}$$

$$Extreme\ Outlier : value < Q_1 - 3\ IQR\ or\ > Q_3 + 3\ IQR, \tag{32}$$

5. Finally, when an outlier was found in the data, it was erased from the sample if it was a mild or extreme outlier in order to obtain a better sample for the next analysis.

Note: The tables associated to the analysis of outliers are shown in the appendix number 3 of this thesis.

Then, according to Figure 5, it was studied each data whether came from a normal distribution through different tests such as Kolmogorov-Smirnov, Shapiro-Wilk, Anderson-Darling and Kolmogorov-Smirnov-Lilliefors. These tests were computed with Software R and those results are shown in Table 11.

In order to determine if data follow a normal distribution, the following two hypothesis were taken into account:

- H_0: Data follow a normal distribution; If p-value is greater than 0.05 ($p - value\ >\ 0,05$).
- H_1: Otherwise; If p-value is lower than 0.05 ($p - value\ <\ 0,05$). So, it cannot assume H_0.

Next and according to Figure 5 and Table 11, as data are independent and at least one of them does not follow a normal distribution, it is possible to assume that the use of a non-parametric test and compare them statistically through the Wilcoxon-Mann-Whitney's test. However, there were some exceptions like *SCP* 4.9, 5.1, 5.7, *B*.1, *C*.4, *NRG*.2, *NRG*.3, *NRG*.5 where all them follow a normal distribution. To statistically compare all of them, T-test was used. These two test were calculated with Software R and those results are shown in Tables 12, 13, 14, 15 and 16.

To proceed, again, the following two hypothesis in two sides were taken into consideration, first:

- H_0: $\tilde{X}_{Improved\ AFSA} \leq \tilde{X}_{Simplified\ AFSA}$; If p-value is greater than 0.05 $(p-value\ >\ 0,05)$.
- H_1: $\tilde{X}_{Improved\ AFSA} > \tilde{X}_{Simplified\ AFSA}$; If p-value is lower than 0.05 $(p-value\ <\ 0,05)$. So, it cannot assume H_0.

Then, on the other hand, the following two hypothesis:

- H_0: $\tilde{X}_{Simplified\ AFSA} \leq \tilde{X}_{Improved\ AFSA}$; If p-value is greater than 0.05 $(p-value\ >\ 0,05)$.
- H_1: $\tilde{X}_{Simplified\ AFSA} > \tilde{X}_{Improved\ AFSA}$; If p-value is lower than 0.05 $(p-value\ <\ 0,05)$. So, it cannot assume H_0.

Where $\tilde{X}$ corresponds to the arithmetic median of fitness values obtained.

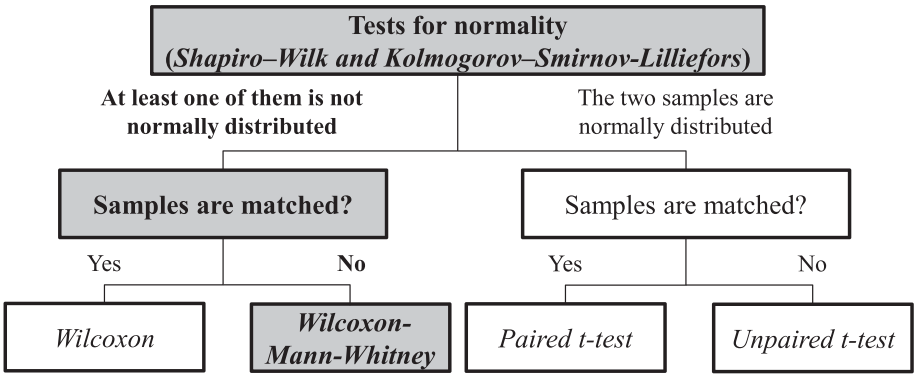

Figure 5: Statistical Methodology Chart

After of executing the tests if $p-value$ in lower than 0.05 then H_0 is rejected and H_1 is accepted. Thereby, it is assumed that the improved version of AFSA provides results than the simplified version of AFSA and vice versa.

Tables 12 to 16 show the results obtained from statistical analysis explained above. According to these results, it is clear that improved version of AFSA has better fitness values in the last 20 files, the instances that have more rows and columns. In the rest of the files, they have results very similar. Nevertheless, and summarizing improved version of AFSA has better performance in 19 of 70 files (27,14%), *SCP D*.1, *D*.2, *D*.5, *E*.5, *NRE*.1, *NRE*.2, *NRE*.4, *NRE*.5, *NRF*.3, *NRG*.1, *NRG*.2, *NRG*.3, *NRG*.4, *NRG*.5, *NRH*.1, *NRH*.2, *NRH*.3, *NRH*.4 and *NRH*.5. On the other hand, simplified version of AFSA has better performance in 5 of 70 files (7,14%), *SCP* 4.3, 4.4, 6.1, 6.5 and *B*.5. Therefore, it is possible to say that the improved version of AFSA performs better than the simplified version of AFSA in some cases, specially in files that have great deal of rows and columns. The details of these results are shown in Table 17, where symbol "$\sim$" means that these two versions of AFSA are similar and symbol "$\star$ (star)" means that one of them is better than other.

Table 11: Normality Tests Results of Improved and Simplified Version AFSA - Experiment 2

		Kolmogorov – Smirnov Test		*Shapiro – Wilk Test*		*Anderson – Darling Test*		*Kolmogorov – Smirnov – Lilliefors Test*	
Number	*Instance*	*Normal Dist. Improved AFSA ?*	*Normal Dist. Simplified AFSA ?*	*Normal Dist. Improved AFSA ?*	*Normal Dist. Simplified AFSA ?*	*Normal Dist. Improved AFSA ?*	*Normal Dist. Simplified AFSA ?*	*Normal Dist. Improved AFSA ?*	*Normal Dist. Simplified AFSA ?*
1	4.1	NO	YES	NO	NO	NO	NO	NO	NO
2	4.2	YES	YES	NO	NO	NO	NO	YES	YES
3	4.3	NO	YES	NO	NO	NO	NO	NO	NO
4	4.4	NO	YES	NO	NO	NO	NO	NO	NO
5	4.5	YES	NO	NO	NO	NO	NO	NO	NO
6	4.6	YES	YES	NO	NO	NO	YES	NO	YES
7	4.7	YES	YES	YES	NO	NO	NO	NO	YES
8	4.8	NO	NO	NO	NO	NO	NO	NO	NO
9	4.9*	YES	YES	YES	YES	YES	YES	YES	YES
10	4.10	YES	YES	NO	YES	NO	YES	YES	YES
11	5.1*	YES	YES	YES	YES	YES	YES	YES	YES
12	5.2	NO	YES	NO	YES	NO	YES	NO	YES
13	5.3	NO	YES	NO	YES	NO	YES	NO	NO
14	5.4	NO	YES	NO	NO	NO	NO	NO	NO
15	5.5	NO	NO	-	-	-	-	-	-
16	5.6	NO	YES	NO	NO	NO	NO	NO	NO
17	5.7*	YES	YES	YES	YES	YES	YES	YES	YES
18	5.8	YES	YES	NO	YES	NO	YES	YES	NO
19	5.9	NO	NO	NO	NO	NO	NO	NO	NO
20	5.10	NO	YES	NO	NO	NO	NO	NO	NO
21	6.1	YES	YES	YES	NO	YES	NO	YES	NO
22	6.2	YES	YES	NO	NO	NO	YES	NO	YES
23	6.3	NO	YES	NO	YES	NO	NO	NO	NO
24	6.4	YES	NO	NO	NO	NO	NO	NO	NO
25	6.5	YES	YES	YES	YES	YES	YES	NO	YES
21	A.1	YES	YES	YES	YES	YES	NO	NO	NO
22	A.2	YES	YES	YES	YES	NO	YES	NO	YES
23	A.3	YES	YES	YES	NO	YES	NO	YES	NO
24	A.4	YES	YES	NO	YES	NO	YES	NO	YES
25	A.5	YES	YES	NO	NO	NO	NO	NO	NO
31	B.1*	YES	YES	YES	YES	YES	YES	YES	YES
32	B.2	YES	YES	YES	NO	YES	NO	YES	NO
33	B.3	YES	YES	YES	YES	YES	YES	NO	YES
34	B.4	YES	YES	YES	YES	NO	NO	NO	NO
35	B.5	YES	YES	NO	NO	NO	NO	NO	NO
36	C.1	YES	YES	YES	YES	YES	NO	YES	NO
37	C.2	YES	YES	YES	YES	YES	YES	NO	YES
38	C.3	YES	YES	YES	YES	NO	YES	NO	YES
39	C.4*	YES	YES	YES	YES	YES	YES	YES	YES
40	C.5	NO	YES	NO	NO	NO	NO	NO	NO
41	D.1	YES	YES	NO	NO	NO	NO	NO	NO
42	D.2	YES	YES	YES	YES	YES	NO	YES	NO
43	D.3	YES	YES	NO	NO	NO	NO	NO	NO
44	D.4	YES	YES	NO	NO	NO	NO	NO	NO
45	D.5	NO	NO	NO	NO	NO	NO	NO	NO
46	E.1	NO	NO	-	-	-	-	-	-
47	E.2	NO	NO	-	-	-	-	-	-
48	E.3	NO	NO	-	-	-	-	-	-
49	E.4	NO	NO	NO	NO	NO	NO	NO	NO
50	E.5	NO	NO	NO	-	NO	-	NO	-
51	NRE.1	NO	NO	NO	NO	NO	NO	NO	NO
52	NRE.2	YES	NO	NO	NO	NO	NO	NO	NO
52	NRE.3	NO	YES	NO	NO	NO	NO	NO	NO
54	NRE.4	YES	YES	NO	NO	NO	NO	NO	NO
55	NRE.5	YES	NO	NO	NO	NO	NO	NO	NO
56	NRF.1	NO	NO	NO	-	NO	-	NO	-
57	NRF.2	NO	NO	NO	NO	NO	NO	NO	NO
58	NRF.3	NO	NO	NO	NO	NO	NO	NO	NO
59	NRF.4	NO	NO	-	-	-	-	-	-
60	NRF.5	NO	NO	NO	NO	NO	NO	NO	NO
61	NRG.1	YES	YES	YES	YES	NO	YES	NO	YES
62	NRG.2*	YES	YES	YES	YES	YES	YES	YES	YES
63	NRG.3*	YES	YES	YES	YES	YES	YES	YES	YES
64	NRG.4	YES	YES	YES	YES	YES	YES	YES	NO
65	NRG.5*	YES	YES	YES	YES	YES	YES	YES	YES
66	NRH.1	YES	YES	YES	NO	YES	NO	YES	NO
67	NRH.2	YES	YES	YES	NO	YES	NO	YES	NO
68	NRH.3	YES	YES	YES	NO	YES	NO	YES	NO
69	NRH.4	YES	YES	YES	NO	YES	NO	YES	NO
70	NRH.5	YES	NO	YES	NO	YES	NO	YES	NO

•Symbol "*" means that obtained results follow a normal distribution in all performed tests.
•Symbol "-" means that obtained results have the same minimum.

Table 12: P-values of Improved and Simplified Version AFSA for Non-Unicost SCP benchmarks (4 and 5 sets) - Experiment 2

Data Set 4				*Data Set* 5			
H_0		*Improved AFSA*	*Simplified AFSA*	H_0		*Improved AFSA*	*Simplified AFSA*
4.1	*Improved AFSA*	-	$> 0,05$	5.1*	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
4.2	*Improved AFSA*	-	$> 0,05$	5.2	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
4.3	*Improved AFSA*	-	$> 0,05$	5.3	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$0,03972$	-		*Simplified AFSA*	$> 0,05$	-
4.4	*Improved AFSA*	-	$> 0,05$	5.4	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$7,263e^{-05}$	-		*Simplified AFSA*	$> 0,05$	-
4.5	*Improved AFSA*	-	$> 0,05$	5.5	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
4.6	*Improved AFSA*	-	$> 0,05$	5.6	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
4.7	*Improved AFSA*	-	$> 0,05$	5.7*	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
4.8	*Improved AFSA*	-	$> 0,05$	5.8	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
4.9*	*Improved AFSA*	-	$> 0,05$	5.9	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
4.10	*Improved AFSA*	-	$> 0,05$	5.10	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-

•Symbol "*" means that obtained results follow a normal distribution.

Table 13: P-values of Improved and Simplified Version AFSA for Non-Unicost SCP benchmarks (6, A and B sets) - Experiment 2

Data Set 6				*Data Set A*				*Data Set B*			
H_0		*Improved AFSA*	*Simplified AFSA*	H_0		*Improved AFSA*	*Simplified AFSA*	H_0		*Improved AFSA*	*Simplified AFSA*
6.1	*Improved AFSA*	-	$> 0,05$	A.1	*Improved AFSA*	-	$> 0,05$	B.1*	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$0,03833$	-		*Simplified AFSA*	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
6.2	*Improved AFSA*	-	$> 0,05$	A.2	*Improved AFSA*	-	$> 0,05$	B.2	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
6.3	*Improved AFSA*	-	$> 0,05$	A.3	*Improved AFSA*	-	$> 0,05$	B.3	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
6.4	*Improved AFSA*	-	$> 0,05$	A.4	*Improved AFSA*	-	$> 0,05$	B.4	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
6.5	*Improved AFSA*	-	$> 0,05$	A.5	*Improved AFSA*	-	$> 0,05$	B.5	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$0,003887$	-		*Simplified AFSA*	$> 0,05$	-		*Simplified AFSA*	$0,03282$	-

•Symbol "*" means that obtained results follow a normal distribution.

Table 14: P-values of Improved and Simplified Version AFSA for Non-Unicost SCP benchmarks (C, D and E sets) - Experiment 2

Data Set C				*Data Set D*				*Data Set E*			
H_0		*Improved AFSA*	*Simplified AFSA*	H_0		*Improved AFSA*	*Simplified AFSA*	H_0		*Improved AFSA*	*Simplified AFSA*
C.1	*Improved AFSA*	-	$> 0,05$	D.1	*Improved AFSA*	-	$0,001317$	E.1	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
C.2	*Improved AFSA*	-	$> 0,05$	D.2	*Improved AFSA*	-	$0,01168$	E.2	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
C.3	*Improved AFSA*	-	$> 0,05$	D.3	*Improved AFSA*	-	$> 0,05$	E.3	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
C.4*	*Improved AFSA*	-	$> 0,05$	D.4	*Improved AFSA*	-	$> 0,05$	E.4	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
C.5	*Improved AFSA*	-	$> 0,05$	D.5	*Improved AFSA*	-	$0,003945$	E.5	*Improved AFSA*	-	$3.546e^{-08}$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-

•Symbol "*" means that obtained results follow a normal distribution.

Table 15: P-values of Improved and Simplified Version AFSA for Non-Unicost SCP benchmarks (NRE and NRF sets) - Experiment 2

Data Set NRE				*Data Set NRF*			
H_0		*Improved AFSA*	*Simplified AFSA*	H_0		*Improved AFSA*	*Simplified AFSA*
NRE.1	*Improved AFSA*	-	$2,101e^{-06}$	NRF.1	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
NRE.2	*Improved AFSA*	-	$0,0004567$	NRF.2	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
NRE.3	*Improved AFSA*	-	$> 0,05$	NRF.3	*Improved AFSA*	-	$0,000119$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
NRE.4	*Improved AFSA*	-	$0,0001198$	NRF.4	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
NRE.5	*Improved AFSA*	-	$0,0001129$	NRF.5	*Improved AFSA*	-	$> 0,05$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-

Table 16: P-values of Improved and Simplified Version AFSA for Non-Unicost SCP benchmarks (NRG and NRH sets) - Experiment 2

Data Set NRG				*Data Set NRH*			
H_0		*Improved AFSA*	*Simplified AFSA*	H_0		*Improved AFSA*	*Simplified AFSA*
NRG.1	*Improved AFSA*	-	$8,473e^{-10}$	NRH.1	*Improved AFSA*	-	$1,141e^{-10}$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
NRG.2*	*Improved AFSA*	-	$7,153e^{-06}$	NRH.2	*Improved AFSA*	-	$1,697e^{-08}$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
NRG.3*	*Improved AFSA*	-	$1,439e^{-07}$	NRH.3	*Improved AFSA*	-	$8,81e^{-10}$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
NRG.4	*Improved AFSA*	-	$1,285e^{-05}$	NRH.4	*Improved AFSA*	-	$1,286e^{-09}$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-
NRG.5*	*Improved AFSA*	-	$4,415e^{-06}$	NRH.5	*Improved AFSA*	-	$7,789e^{-11}$
	Simplified AFSA	$> 0,05$	-		*Simplified AFSA*	$> 0,05$	-

•Symbol "*" means that obtained results follow a normal distribution.

Table 17: Summary of P-values of Improved and Simplified Version AFSA for Non-Unicost SCP benchmarks (4, 5, 6, A, B, C, D, E, NRE, NRF, NRG and NRH sets) - Experiment 2

Number	*Instance*	*Improved AFSA*	*Simplified AFSA*
1	4.1	$\sim$	$\sim$
2	4.2	$\sim$	$\sim$
3	4.3	-	$\star$
4	4.4	-	$\star$
5	4.5	$\sim$	$\sim$
6	4.6	$\sim$	$\sim$
7	4.7	$\sim$	$\sim$
8	4.8	$\sim$	$\sim$
9	4.9	$\sim$	$\sim$
10	4.10	$\sim$	$\sim$
11	5.1	$\sim$	$\sim$
12	5.2	$\sim$	$\sim$
13	5.3	$\sim$	$\sim$
14	5.4	$\sim$	$\sim$
15	5.5	$\sim$	$\sim$
16	5.6	$\sim$	$\sim$
17	5.7	$\sim$	$\sim$
18	5.8	$\sim$	$\sim$
19	5.9	$\sim$	$\sim$
20	5.10	$\sim$	$\sim$
21	6.1	-	$\star$
22	6.2	$\sim$	$\sim$
23	6.3	$\sim$	$\sim$
24	6.4	$\sim$	$\sim$
25	6.5	-	$\star$
26	A.1	$\sim$	$\sim$
27	A.2	$\sim$	$\sim$
28	A.3	$\sim$	$\sim$
29	A.4	$\sim$	$\sim$
30	A.5	$\sim$	$\sim$
31	B.1	$\sim$	$\sim$
32	B.2	$\sim$	$\sim$
33	B.3	$\sim$	$\sim$
34	B.4	$\sim$	$\sim$
35	B.5	-	$\star$
36	C.1	$\sim$	$\sim$
37	C.2	$\sim$	$\sim$
38	C.3	$\sim$	$\sim$
39	C.4	$\sim$	$\sim$
40	C.5	$\sim$	$\sim$
41	D.1	$\star$	-
42	D.2	$\star$	-
43	D.3	$\sim$	$\sim$
44	D.4	$\sim$	$\sim$
45	D.5	$\star$	-
46	E.1	$\sim$	$\sim$
47	E.2	$\sim$	$\sim$
48	E.3	$\sim$	$\sim$
49	E.4	$\sim$	$\sim$
50	E.5	$\star$	-
51	NRE.1	$\star$	-
52	NRE.2	$\star$	-
52	NRE.3	$\sim$	$\sim$
54	NRE.4	$\star$	-
55	NRE.5	$\star$	-
56	NRF.1	$\sim$	$\sim$
57	NRF.2	$\sim$	$\sim$
58	NRF.3	$\star$	-
59	NRF.4	$\sim$	$\sim$
60	NRF.5	$\sim$	$\sim$
61	NRG.1	$\star$	-
62	NRG.2	$\star$	-
63	NRG.3	$\star$	-
64	NRG.4	$\star$	-
65	NRG.5	$\star$	-
66	NRH.1	$\star$	-
67	NRH.2	$\star$	-
68	NRH.3	$\star$	-
69	NRH.4	$\star$	-
70	NRH.5	$\star$	-

•Symbol "$\sim$" means that these two versions of AFSA are similar and Symbol "$\star$ (star)" means that one of them is better than other.

6 Chapter 6: Discussion and Conclusion

6.1 Discussions

This thesis was a comparative study between two versions of AFSA, improved and simplified, which imitates a fish within a shoal inside the water, in order to solve SCP. These two mentioned algorithms were tried out in two ways. The first experiment of improved version was used with Algorithm 3 and Algorithm 4, the first one which transform the members of population into feasible solutions and the second one as that can eliminate redundancy of the solutions. Then, the first experiment of simplified version which was utilized with Algorithm 6 as repair function, this algorithm can accomplish two tasks, it can convert the solutions into feasible and also it can deal with the SCP constraints. On the other hand, the second experiment of improved and simplified were utilized with Algorithm 6 as repair function.

It has been seen in Table 4 that the first algorithm had worse results than the second one, although it had minor variability in its results because after several iterations the first algorithm converges to a non-optimal solution and it stagnates in the same result. However, the second algorithm converges quickly to better solutions, in some cases optimal solutions, but it requires more iterations and it has more variability in comparison to the first one.

Table 5 shows very similar results between these two versions of AFSA. It means that the repair function, which is shown in Algorithm 6, helps a lot in order to obtain better results. Hence, these results indicate that the improved version of AFSA and simplified version of AFSA are algorithms that can obtain global optimums in many cases.

Regarding experiment 2, these two versions of AFSA optimization have a good performance with almost all instances. It has observed that sets $4,5,6$ and A,B,CD,E and NRE obtained each one an Average RPD minor to 4%, and sets NRF,NRG,NRH obtained an Average RPD minor to 11%. However, in the results of the last 20 instances, the Average RPD starts to decrease.

In comparison to other other metaheuristics such as Artificial Bee Colony Algorithm (ABCA) [81], Binary Firefly Algorithm (BFA) [82] or Genetic Algorithm (GA) [29], Metaheuristic Algorithm Based on Gravity (ABG) [83], Binary Cat Swarm Optimization Algorithm (BCSOA) [52] and Teaching-Learning-Based Optimization Algorithm (TLBOA) [6]. The results of these two version of AFSA are very similar compared with the algorithms mentioned before and they are even better in other cases. Table 18 (where symbol "-" means that set E was not performed with those algorithms) shows comparative results among these algorithms. For this reason, this technique requires more study because of the results obtained and it is believed that it is possible to obtain the global optimums in all instances or decreasing the RPD value even more for those instances that do not obtain the optimal solution. Specially, after obtaining additional results in experiment 3, where instance 5.2, 6.2, $A.1$, $NRF.3$, $NRF.5$ and $NRH.1$ have obtained better results with simplified version of AFSA and with specific a configuration of parameters for those benchmarks. So, there is an additional space for further investigation.

Additionally, it was observed that in the first 300 iterations, these two algorithms converge quickly to very good solutions or optimal solutions in some cases. After the 300 iterations, sometimes, these algorithms can obtain a slightly better results. Nevertheless, in most cases these version of AFSA maintain the same result reached at the first 300

iterations. Therefore, It could be necessary to find a better configuration of parameters to obtain the optimal results or closest to the optimal in all instances and that could reduce the variability in their results.

Another characteristic that was observed is that these two versions of AFSA requires a lot of processing time with some benchmarks, especially the last 10 files, sets from *NRG* to *NRH*. Improved version needs almost 25 minutes on average for those sets and simplified version needs almost 41 minutes. However, these two processing times may be due to the number of iterations and the number of members in their population. Hence, it could be possible that it is necessary to introduce the technique of the dimension reduction of the problem in order to reduce the processing time, obtain two better algorithms and strong techniques. The mentioned method of the dimension reduction of the problem is shown in [77].

According to outlier analysis results, it is clear that the improved version of AFSA has better fitness values in the last 20 files, the instances that have more rows and columns. In the rest of the files, they have results very similar. Therefore and summarizing improved version of AFSA has better performance in 19 of 70 files (27,14%). On the other hand, the simplified version of AFSA has better performance in 5 of 70 files (7,14%). Thereby, it is possible to say that improved version of AFSA performs better than simplified version of AFSA in some cases, specially in files that have a great deal of rows and columns.

6.2 Conclusions

According to the results shown in this thesis, it is considered that it has fulfilled the general and the specific objectives proposed at the beginning of this work. It has solved Solve the Set Covering Problem using the Artificial Fish Swarm Algorithm. Moreover, it has studied and understood the SCP, it has researched and comprehended how works the metaheuristic of AFSA, it has modified and adjusted AFSA in order to solve SCP, it has solved SCP using the two selected versions of AFSA, improved and simplified, and finally it has analyzed and compared the results obtained with the these two AFSA versions.

Improved and Simplified version of AFSA have very good results with Algorithm 6 utilized as repair function. Moreover, they converge quickly in few iterations and they have an acceptable time of processing according to the duration obtained for each version of this algorithm.

In comparison to other other metaheuristics based on population and taking into consideration the results obtained, AFSA provides another alternative based population metaheuristic that is able to solve minimization problems in different contexts, not only in academic scope, also in the industry.

The final conclusion is that both of them have very similar results according to the statistical analysis done on this thesis. However, it is believed that improved version of AFSA has a slightly better performance than simplified version of AFSA. Specially in files that are bigger in their dimension.

This thesis shown that these two algorithms are a good way to solve SCP with feasible solutions, in a binary domain.

An interesting future work would be to apply other versions of AFSA on SCP, making a comparison between those different versions of AFSA, utilize AFSA in order to solve SCP uni-cost, or search for a particular configuration of parameters for each instance analyzed in this thesis. Also, introducing the dimension reduction of the problem which was mentioned above.

The following section, *Appendix*, shows more charts with the results that were explained in *Results Chapter*.

Table 18: Comparative Results Among Different Algorithms of Non-Unicost SCP benchmarks (4, 5, 6, A, B, C, D, E, NRE, NRF, NRG and NRH sets) - Experiment 2

			ABCA	*BFA*	*GA*	*ABG*	*BCSOA*	*TLBOA*	*Improved AFSA*	*Simplified AFSA*
Number	*Instance*	Z_{opt}	Z_{min}	Z_{min}	Z_{min}	Z_{min}	Z_{min}	Z_{min}	Z_{min}	Z_{min}
1	4.1	429	430	481	429	429	459	430	430	430
2	4.2	512	512	580	512	512	570	524	512	512
3	4.3	516	516	619	516	516	590	526	516	516
4	4.4	494	494	537	494	494	547	501	495	495
5	4.5	512	512	609	512	512	545	518	514	512
6	4.6	560	561	653	560	560	637	566	560	560
7	4.7	430	430	491	430	430	462	433	431	432
8	4.8	492	493	565	492	492	546	507	493	492
9	4.9	641	643	749	641	641	711	660	649	645
10	4.10	514	514	550	514	514	537	524	514	514
11	5.1	253	254	296	253	253	279	257	255	253
12	5.2	302	309	372	302	302	339	311	311	306
13	5.3	226	228	250	228	226	247	228	228	226
14	5.4	242	242	277	242	242	251	244	242	242
15	5.5	211	211	253	211	211	230	215	211	211
16	5.6	213	213	264	213	213	232	217	214	213
17	5.7	293	296	337	293	293	332	293	293	293
18	5.8	288	288	326	288	288	320	294	288	288
19	5.9	279	280	350	279	279	295	281	279	279
20	5.10	265	266	321	265	265	285	268	267	265
21	6.1	138	140	173	138	138	151	143	138	138
22	6.2	146	146	180	146	146	152	148	147	147
23	6.3	145	145	160	145	145	160	148	145	145
24	6.4	131	131	161	131	131	138	131	131	131
25	6.5	161	161	186	161	161	169	167	161	161
26	A.1	253	254	285	253	253	286	257	253	254
27	A.2	252	254	285	252	252	274	263	255	254
28	A.3	232	234	272	232	232	257	242	235	235
29	A.4	234	234	297	234	234	248	237	235	236
30	A.5	236	237	262	236	236	244	239	238	237
31	B.1	69	69	80	69	69	79	72	70	69
32	B.2	76	76	92	76	76	86	82	78	76
33	B.3	80	80	93	80	80	85	80	80	80
34	B.4	79	79	98	79	79	89	82	81	79
35	B.5	72	72	87	72	72	73	72	72	72
36	C.1	227	227	279	227	227	242	235	229	228
37	C.2	219	219	272	219	219	240	226	220	220
38	C.3	243	243	288	243	243	277	263	250	250
39	C.4	219	219	262	219	219	250	238	222	222
40	C.5	215	215	262	215	215	243	220	217	215
41	D.1	60	60	71	60	60	65	62	60	60
42	D.2	66	66	75	66	66	70	70	67	67
43	D.3	72	72	88	72	72	79	77	76	75
44	D.4	62	62	71	62	62	64	65	63	63
45	D.5	61	61	71	61	61	65	64	62	61
46	E.1	5	-	-	-	-	-	-	5	5
47	E.2	5	-	-	-	-	-	-	5	5
48	E.3	5	-	-	-	-	-	-	5	5
49	E.4	5	-	-	-	-	-	-	5	5
50	E.5	5	-	-	-	-	-	-	5	5
51	NRE.1	29	29	32	29	29	29	30	29	29
52	NRE.2	30	30	36	30	30	34	34	31	31
52	NRE.3	27	27	35	27	27	31	29	29	28
54	NRE.4	28	28	34	28	28	32	32	30	29
55	NRE.5	28	28	34	28	28	30	30	28	29
56	NRF.1	14	14	17	14	14	17	17	15	15
57	NRF.2	15	15	17	15	15	18	17	15	16
58	NRF.3	14	14	21	14	14	17	17	16	16
59	NRF.4	14	14	19	14	14	17	16	15	15
60	NRF.5	13	13	16	13	13	15	15	14	15
61	NRG.1	176	176	230	176	176	190	193	187	180
62	NRG.2	154	154	191	155	154	165	164	161	161
63	NRG.3	166	166	198	166	166	187	178	174	171
64	NRG.4	168	168	214	168	168	179	180	175	171
65	NRG.5	168	168	223	168	168	181	183	177	174
66	NRH.1	63	63	85	64	63	70	71	68	67
67	NRH.2	63	63	81	64	63	67	67	68	66
68	NRH.3	59	59	76	59	59	74	68	66	63
69	NRH.4	58	58	75	58	58	68	66	63	61
70	NRH.5	55	55	68	56	55	66	60	60	57

7 Chapter 7: Appendix

7.1 Appendixes

7.1.1 Appendix 1 - Convergence Charts Experiment 1

This section shows some convergence charts that explain the performance of some instances for experiment 1. In this case *SCP* 4.1 represented by *Figure 6*, *SCP* 4.8 represented by *Figure 7*, *SCP* 5.4 represented by *Figure 8*, *SCP* 5.9 represented by *Figure 9*, *SCP* 6.1 represented by *Figure 10*, *SCP A*.5 represented by *Figure 11*, *SCP B*.5 represented by *Figure 12*, *SCP C*.5 represented by *Figure 13*, *SCP D*.1 represented by *Figure 14*, *SCP E*.5 represented by *Figure 15*, *SCP NRE*.1 represented by *Figure 16*, *SCP NRF*.3 represented by *Figure 17*, *SCP NRG*.1 represented by *Figure 18* and *SCP NRH*.2 represented by *Figure 19*.

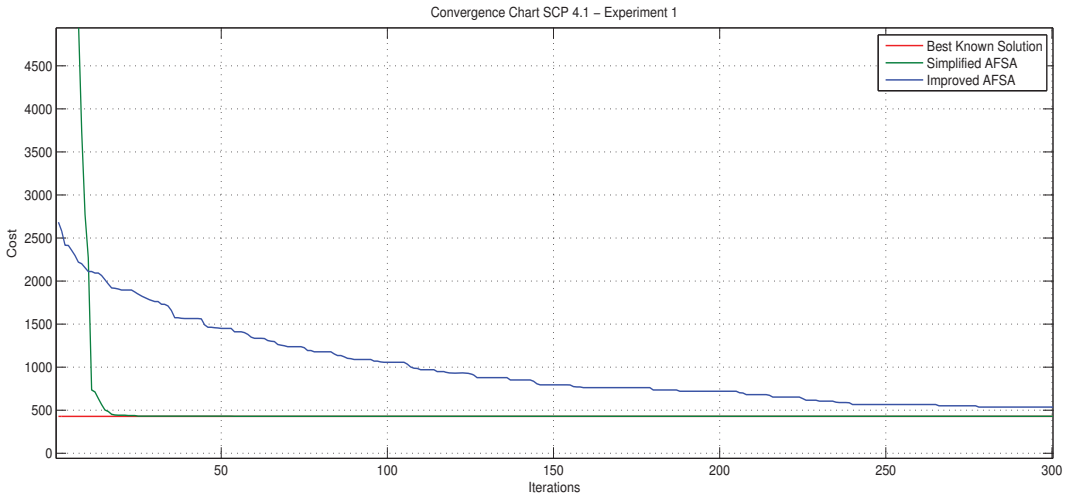

Figure 6: AFSA-SCP 4.1 Convergence Chart - Exp.1

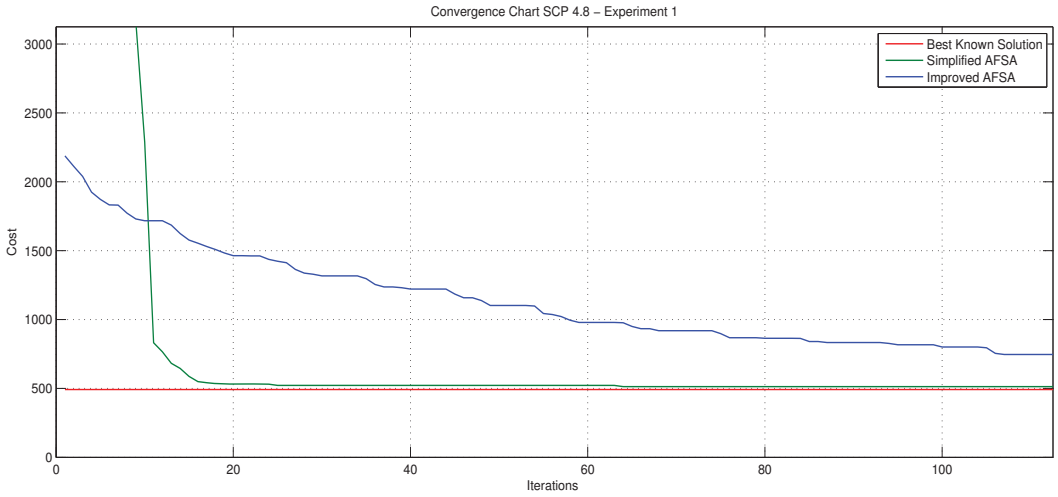

Figure 7: AFSA-SCP 4.8 Convergence Chart - Exp.1

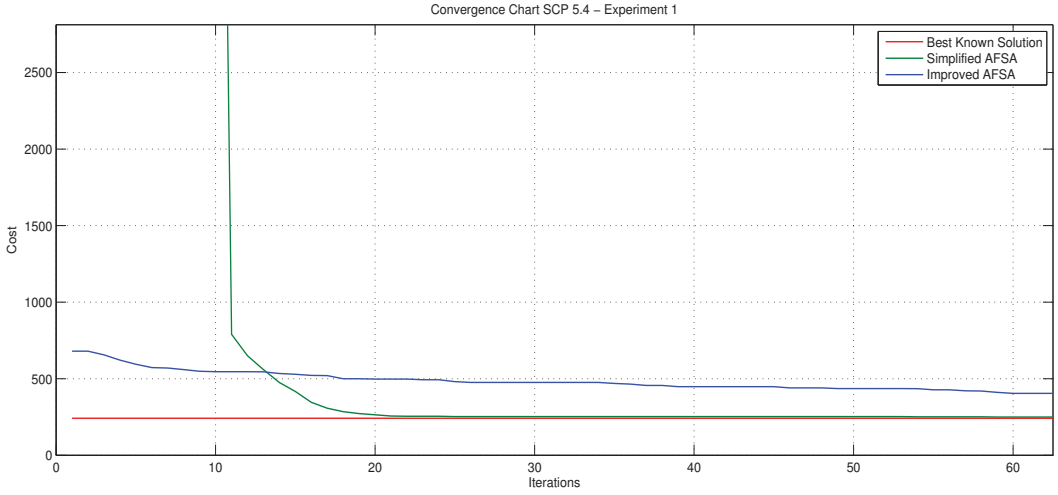

Figure 8: AFSA-SCP 5.4 Convergence Chart - Exp.1

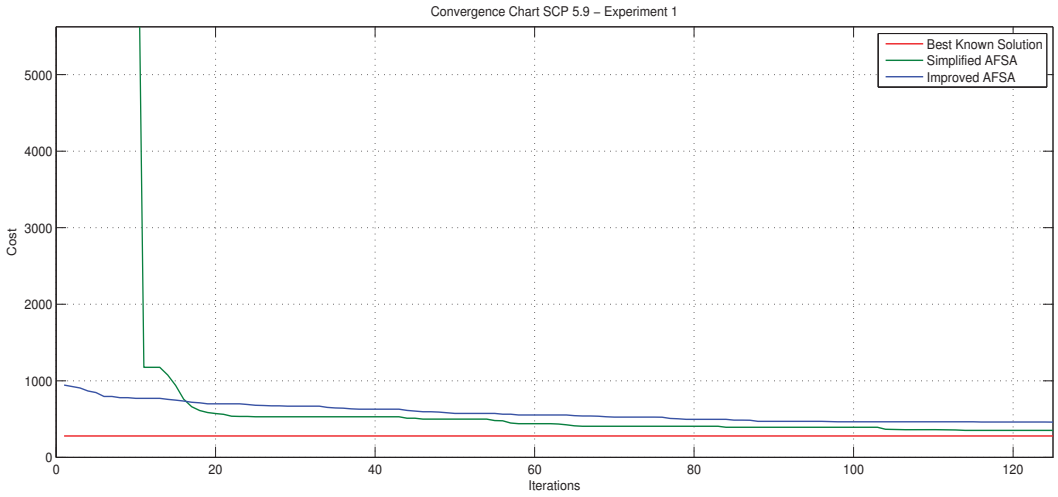

Figure 9: AFSA-SCP 5.9 Convergence Chart - Exp.1

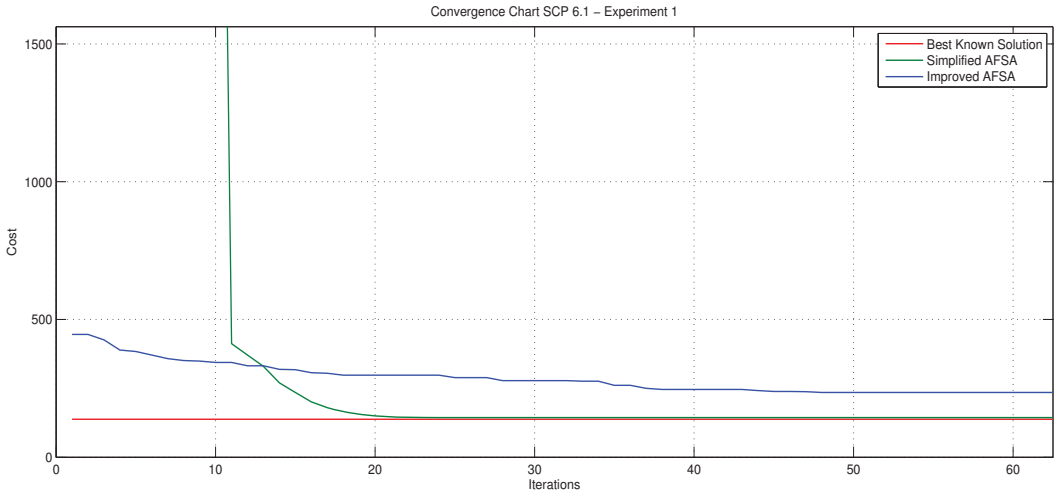

Figure 10: AFSA-SCP 6.1 Convergence Chart - Exp.1

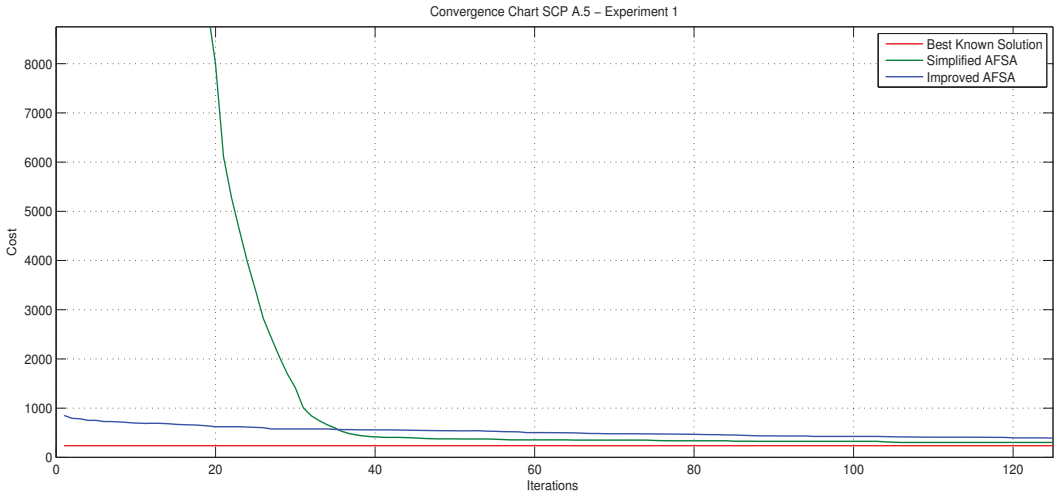

Figure 11: AFSA-SCP A.5 Convergence Chart - Exp.1

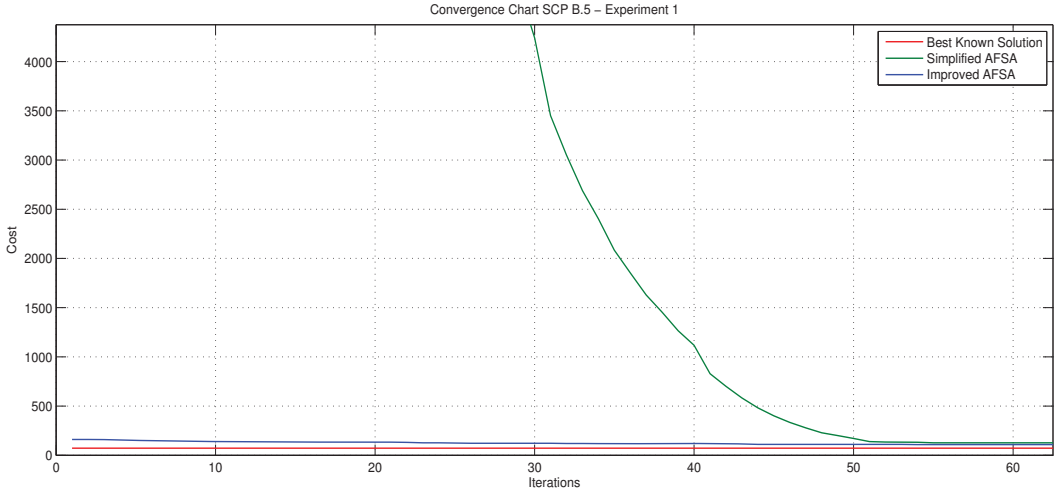

Figure 12: AFSA-SCP B.5 Convergence Chart - Exp.1

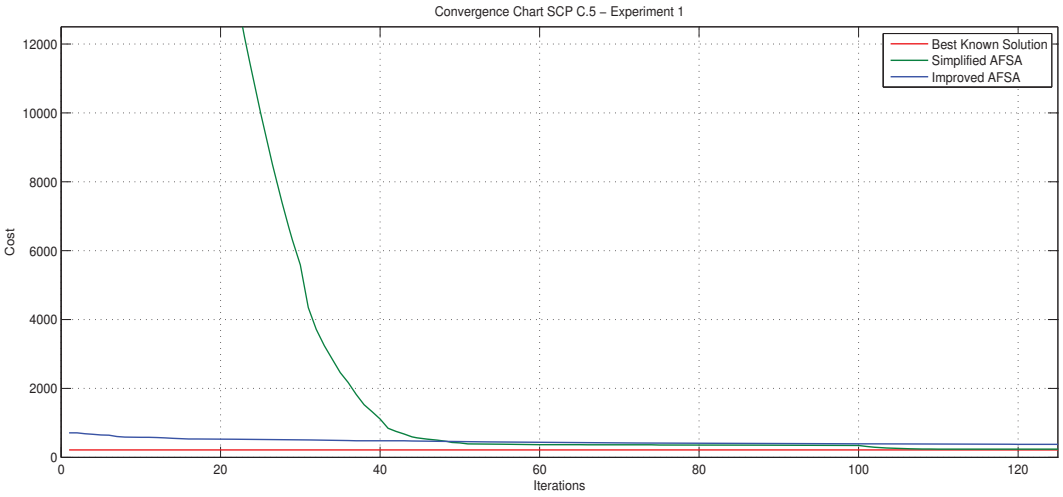

Figure 13: AFSA-SCP C.5 Convergence Chart - Exp.1

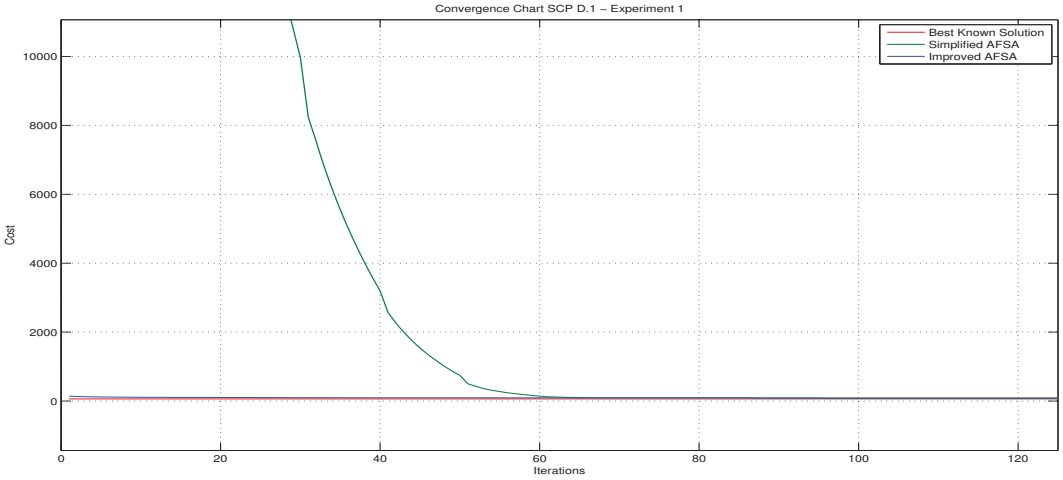

Figure 14: AFSA-SCP D.1 Convergence Chart - Exp.1

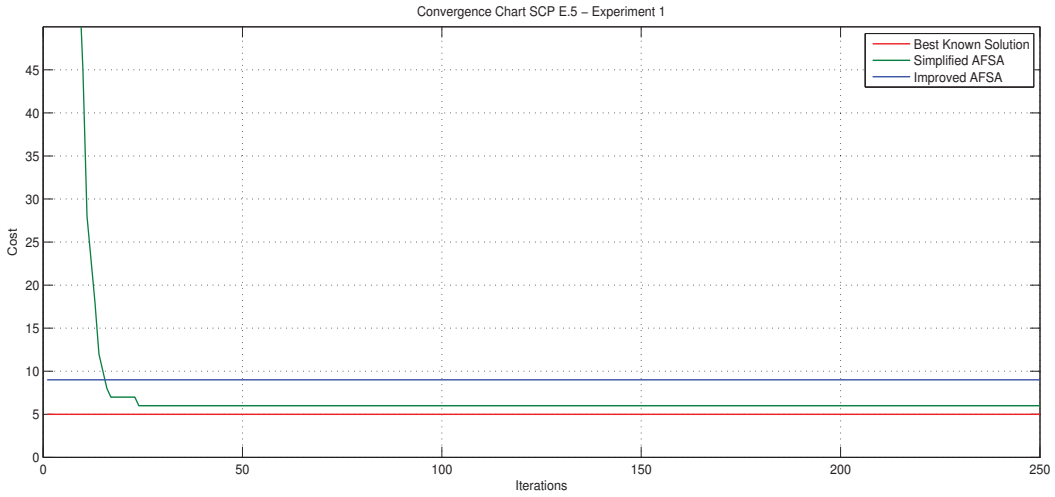

Figure 15: AFSA-SCP E.5 Convergence Chart - Exp.1

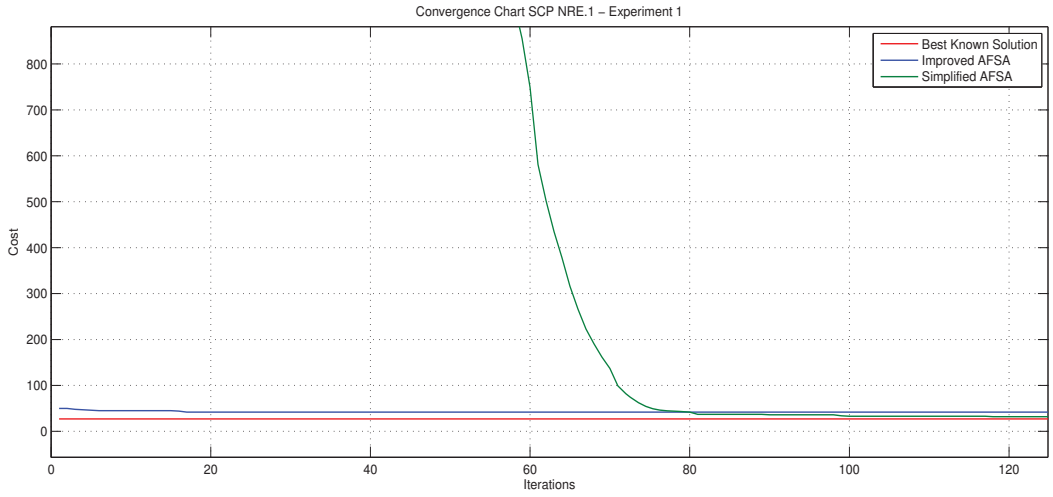

Figure 16: AFSA-SCP NRE.1 Convergence Chart- Exp.1

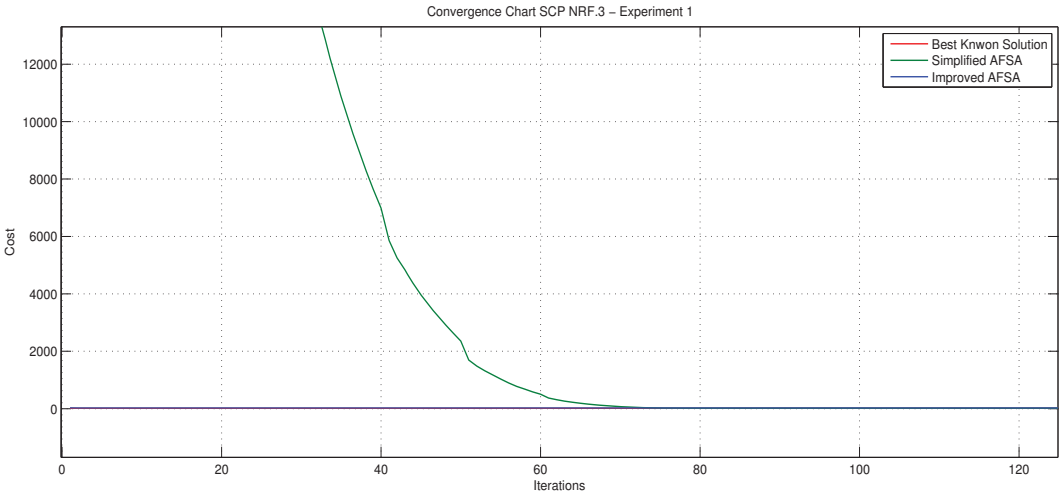

Figure 17: AFSA-SCP NRF.3 Convergence Chart - Exp.1

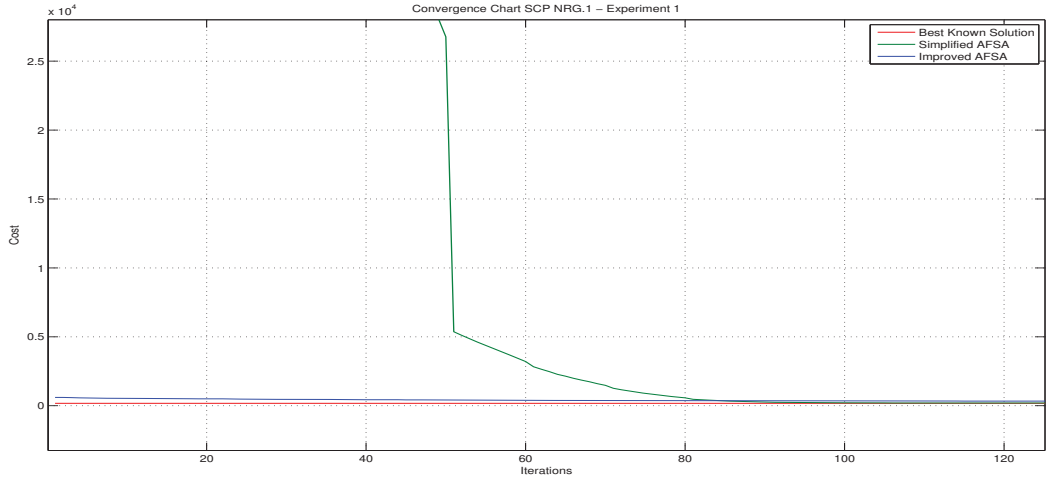

Figure 18: AFSA-SCP NRG.1 Convergence Chart - Exp.1

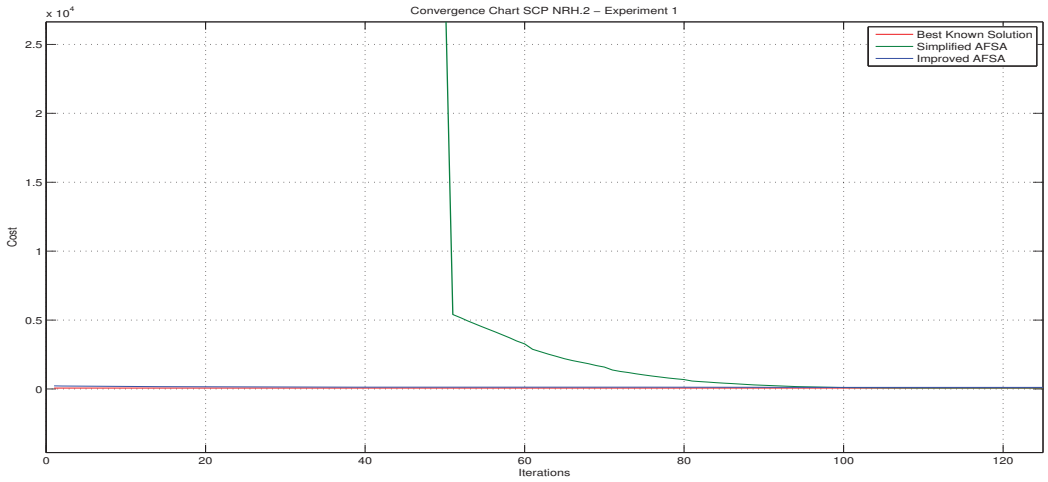

Figure 19: AFSA-SCP NRH.2 Convergence Chart - Exp.1

7.1.2 Appendix 2 - Convergence Charts Experiment 2

Here, some convergence charts are shown, which explain the performance of some instances for experiment 2. In this case *SCP* 4.7 represented by *Figure 20*, *SCP* 4.9 represented by *Figure 21*. *SCP* 5.2 represented by *Figure 22*, *SCP* 5.3 represented by *Figure 23*, *SCP* 6.2 represented by *Figure 24*, *SCP A*.2 represented by *Figure 25*, *SCP B*.2 represented by *Figure 26*, *SCP C*.3 represented by *Figure 27*, *SCP D*.3 represented by *Figure 28*, *SCP E*.3 represented by *Figure 29*, *SCP NRE*.3 represented by *Figure 30*, *SCP NRF*.2 represented by *Figure 31*, *SCP NRG*.4 represented by *Figure 32* and *SCP NRH*.3 represented by *Figure 33*.

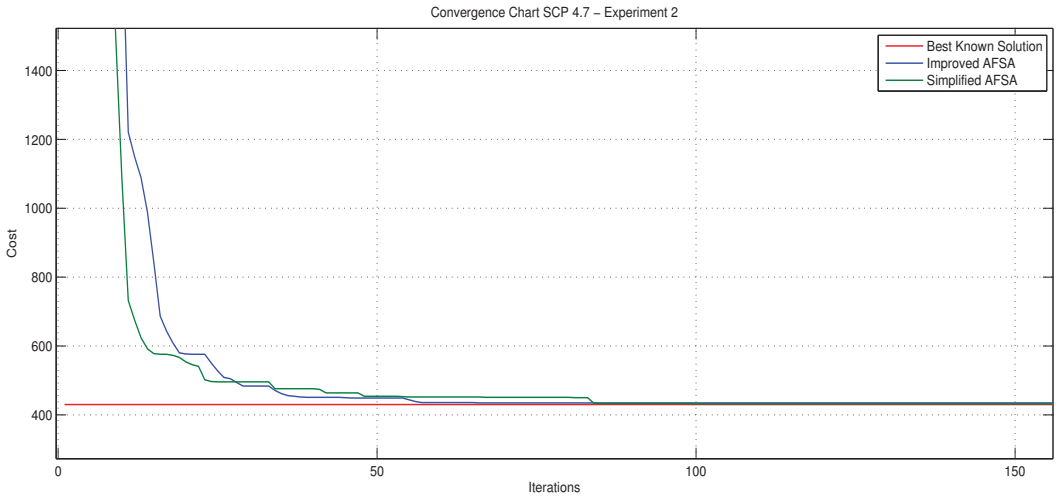

Figure 20: AFSA-SCP 4.7 Convergence Chart - Exp.2

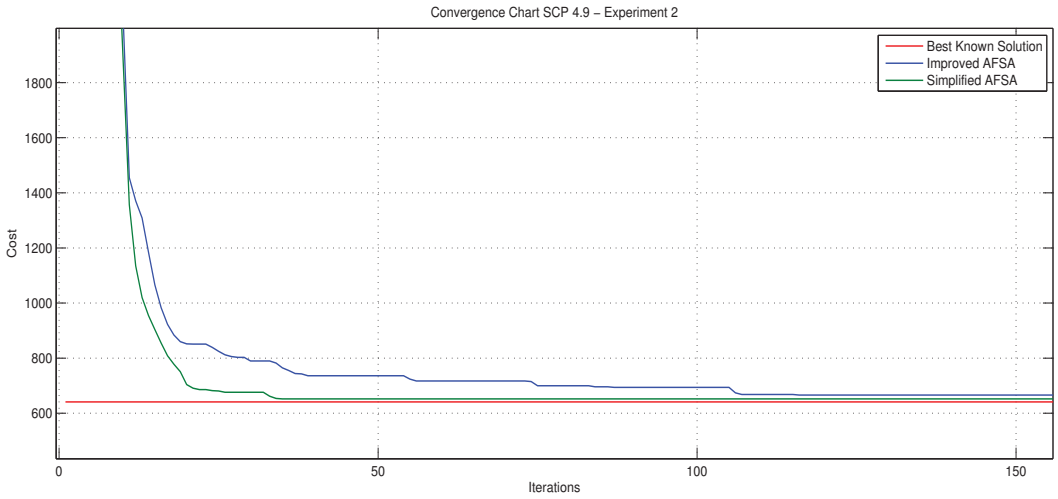

Figure 21: AFSA-SCP 4.9 Convergence Chart - Exp.2

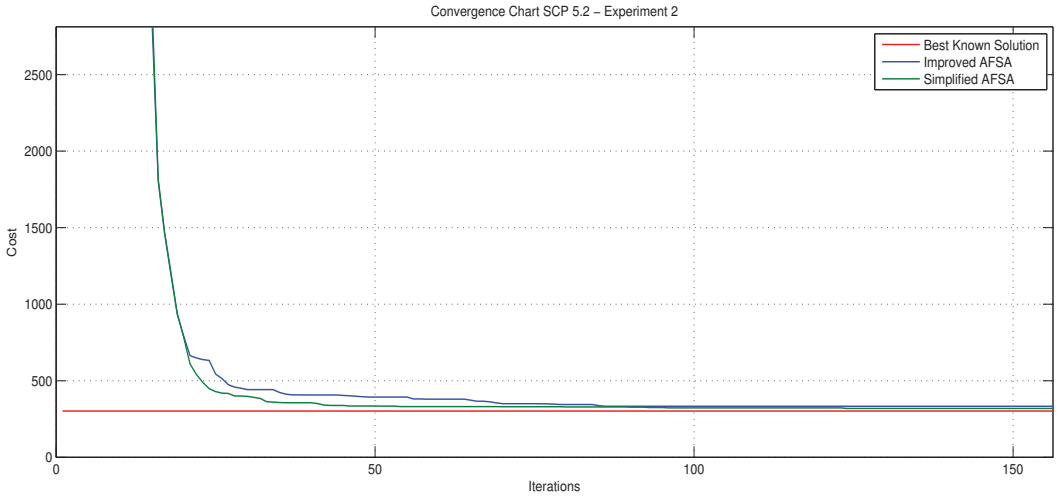

Figure 22: AFSA-SCP 5.2 Convergence Chart - Exp.2

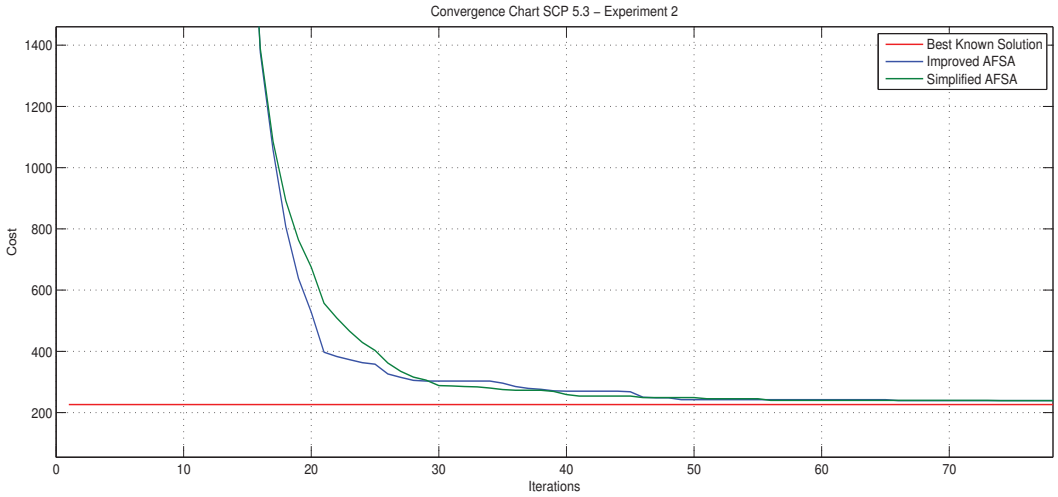

Figure 23: AFSA-SCP 5.3 Convergence Chart - Exp.2

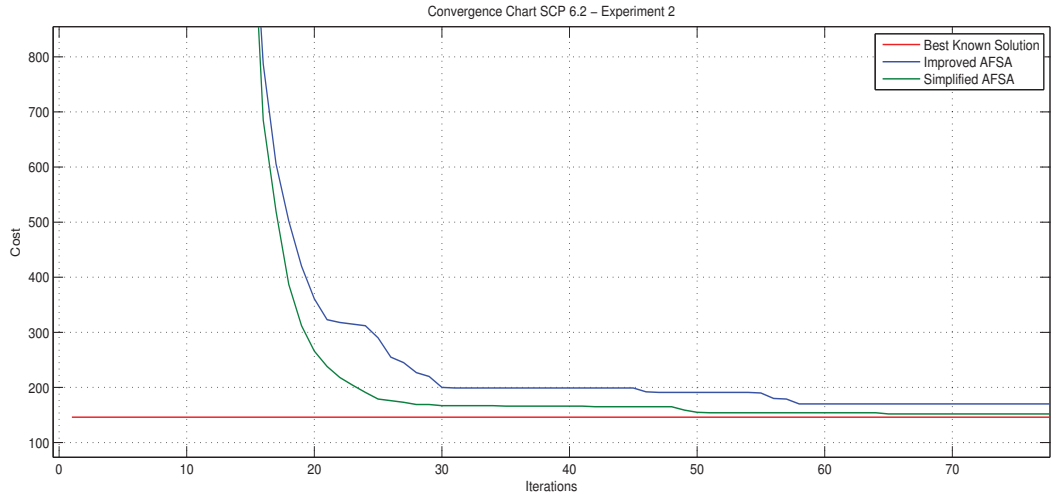

Figure 24: AFSA-SCP 6.2 Convergence Chart - Exp.2

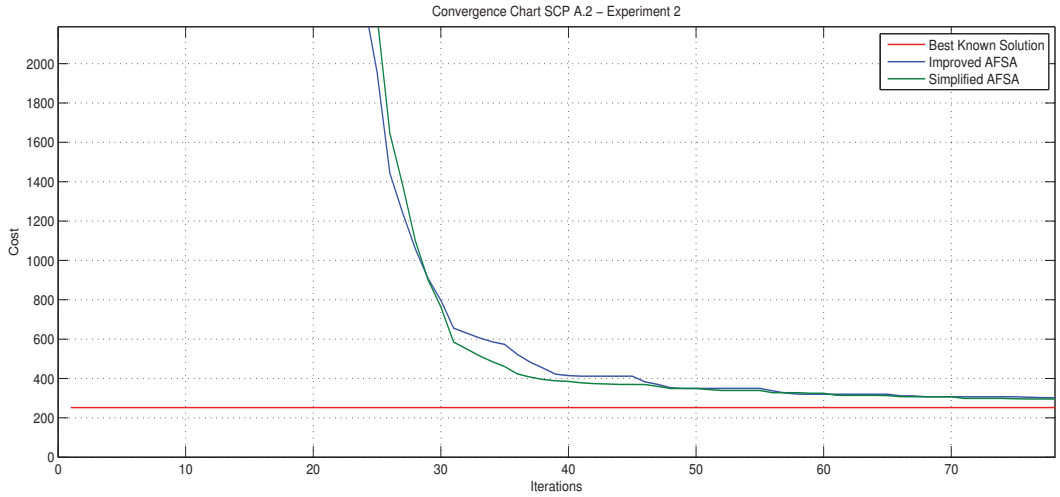

Figure 25: AFSA-SCP A.2 Convergence Chart - Exp.2

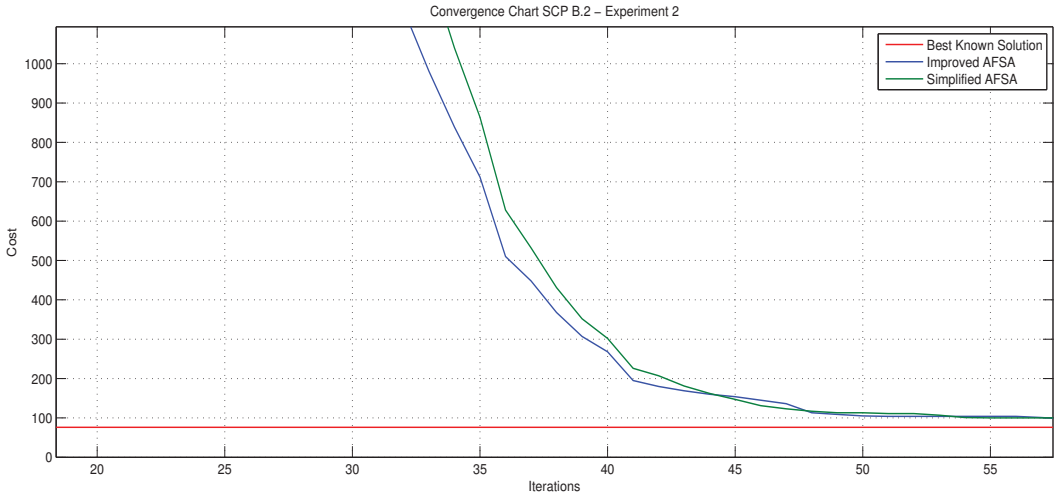

Figure 26: AFSA-SCP B.2 Convergence Chart - Exp.2

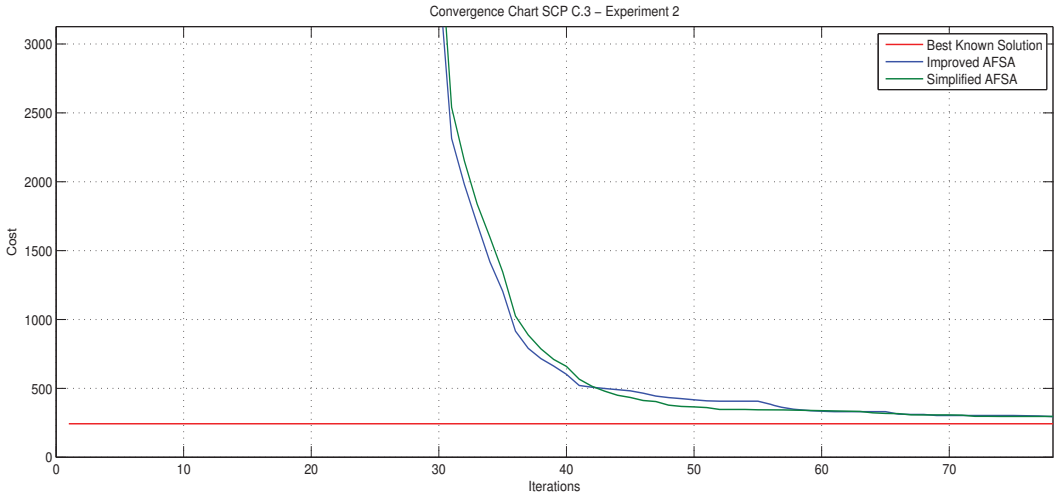

Figure 27: AFSA-SCP C.3 Convergence Chart - Exp.2

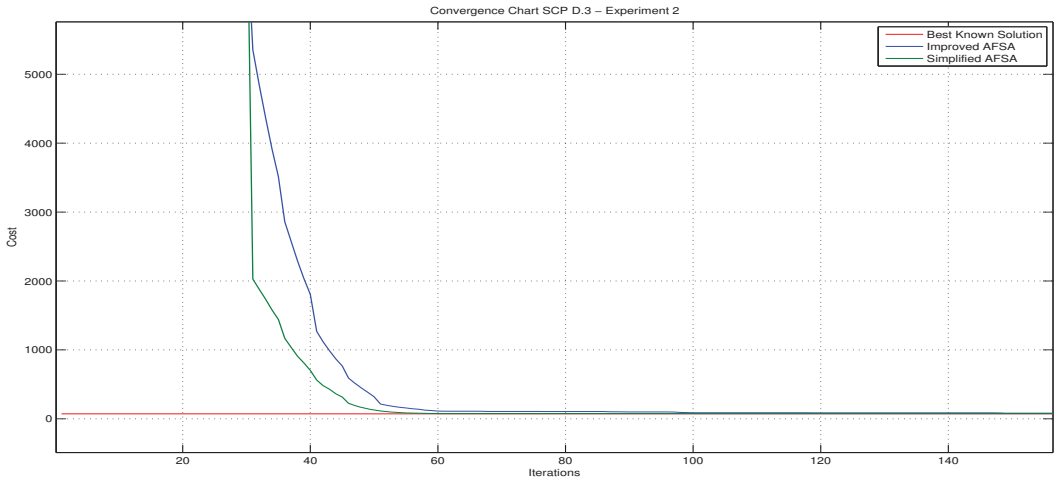

Figure 28: AFSA-SCP D.3 Convergence Chart - Exp.2

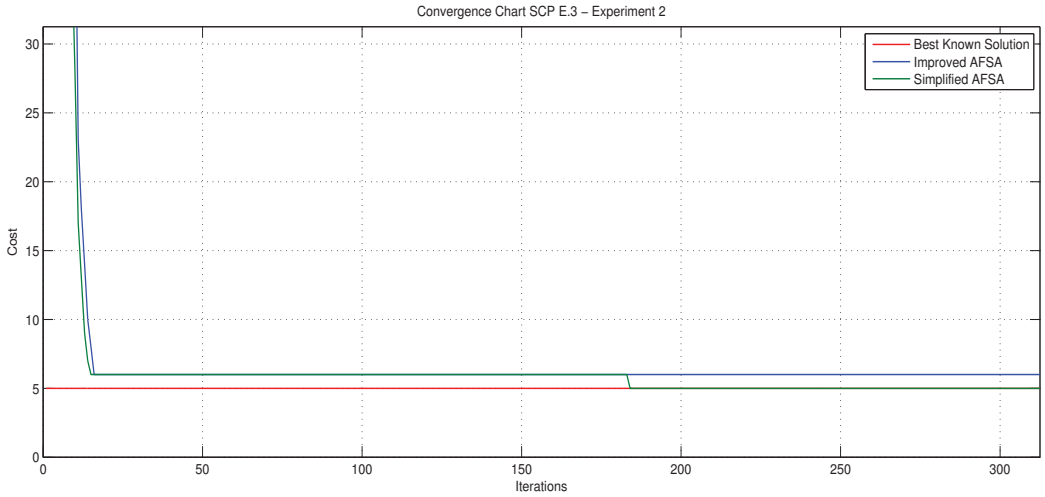

Figure 29: AFSA-SCP E.3 Convergence Chart - Exp.2

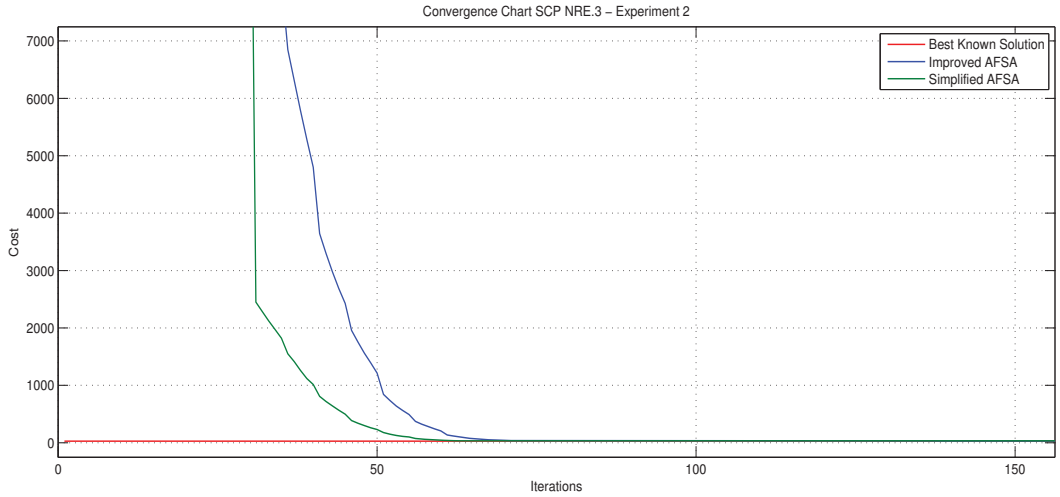

Figure 30: AFSA-SCP NRE.3 Convergence Chart- Exp.2

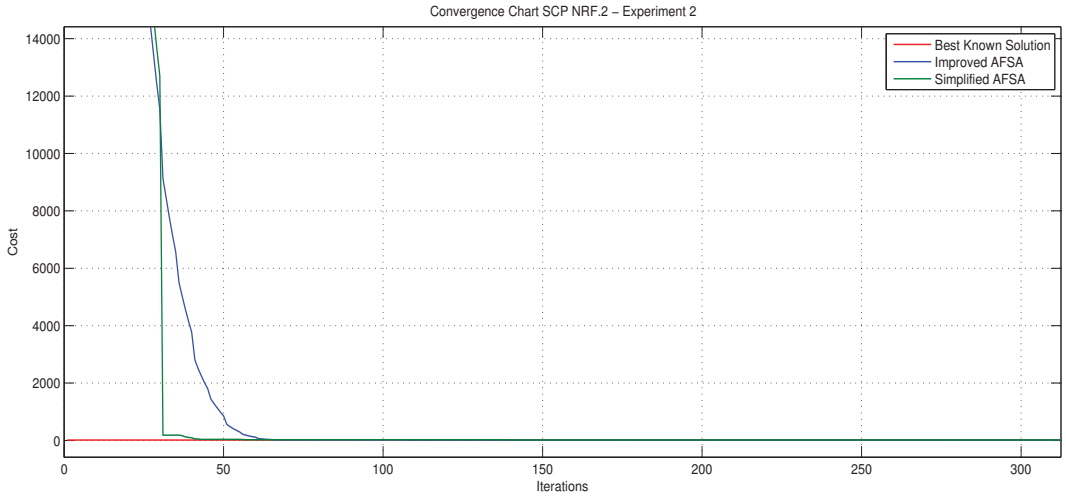

Figure 31: AFSA-SCP NRF.2 Convergence Chart - Exp.2

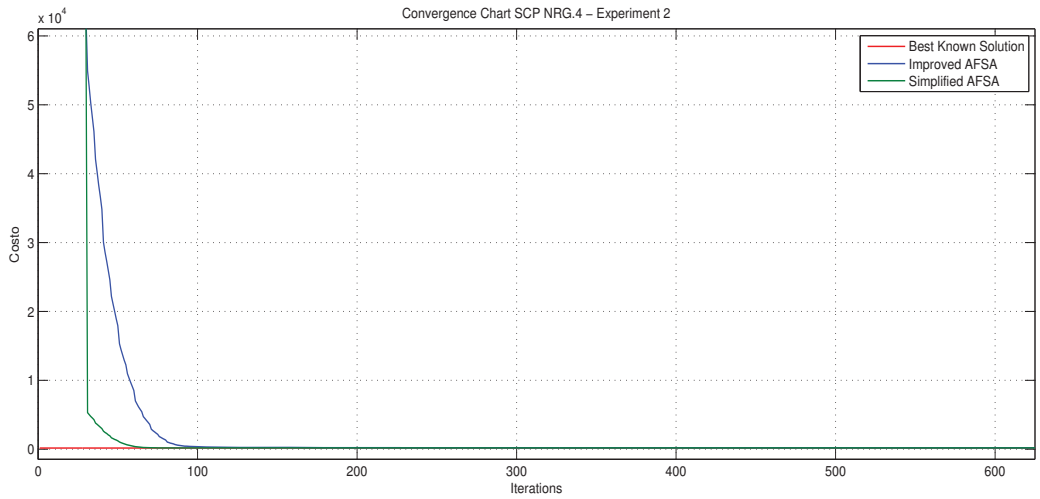

Figure 32: AFSA-SCP NRG.4 Convergence Chart - Exp.2

S

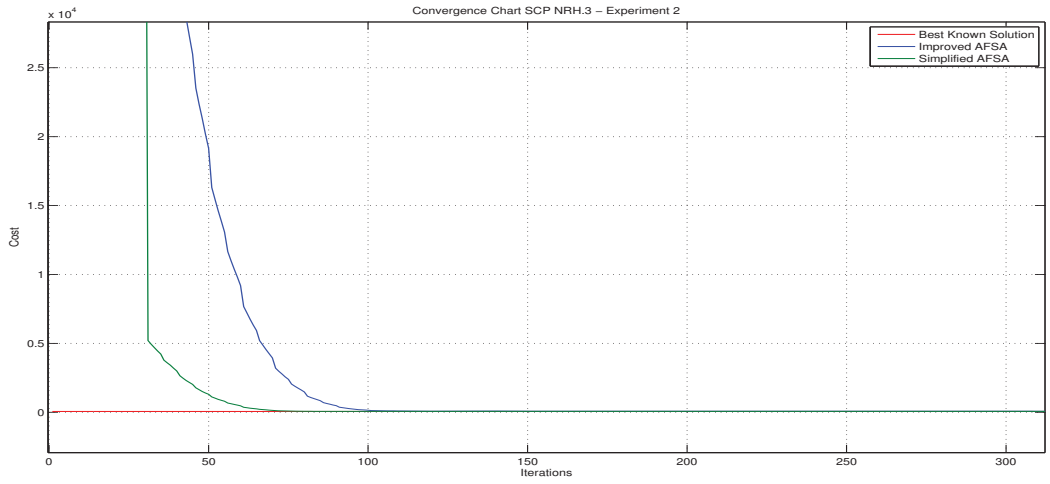

Figure 33: AFSA-SCP NRH.3 Convergence Chart - Exp.2

7.1.3 Appendix 3 - Tables of Outlier Analysis

Finally, in the following tables, it is possible to see the 30 independent executions for each instance. Each table has the number of iterations, the fitness value obtained by improved AFSA, the outlier analysis (whether it is mild or extreme outlier), the fitness value obtained by simplified AFSA and the corresponding outlier analysis for that version.

Table 19: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 4.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	431	NO	NO	443	NO	NO
2	437	NO	NO	437	NO	NO
3	437	NO	NO	437	NO	NO
4	440	NO	NO	435	NO	NO
5	437	NO	NO	433	NO	NO
6	437	NO	NO	433	NO	NO
7	437	NO	NO	433	NO	NO
8	430	NO	NO	432	NO	NO
9	433	NO	NO	437	NO	NO
10	432	NO	NO	437	NO	NO
11	437	NO	NO	437	NO	NO
12	437	NO	NO	433	NO	NO
13	438	NO	NO	432	NO	NO
14	437	NO	NO	437	NO	NO
15	433	NO	NO	433	NO	NO
16	437	NO	NO	443	NO	NO
17	433	NO	NO	437	NO	NO
18	437	NO	NO	433	NO	NO
19	443	NO	NO	437	NO	NO
20	437	NO	NO	431	NO	NO
21	432	NO	NO	433	NO	NO
22	433	NO	NO	433	NO	NO
23	437	NO	NO	433	NO	NO
24	432	NO	NO	443	NO	NO
25	432	NO	NO	437	NO	NO
26	437	NO	NO	437	NO	NO
27	433	NO	NO	443	NO	NO
28	433	NO	NO	437	NO	NO
29	432	NO	NO	430	NO	NO
30	432	NO	NO	443	NO	NO

Table 20: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 4.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	518	NO	NO	520	NO	NO
2	512	NO	NO	532	NO	NO
3	521	NO	NO	534	NO	NO
4	530	NO	NO	523	NO	NO
5	521	NO	NO	523	NO	NO
6	512	NO	NO	533	NO	NO
7	512	NO	NO	512	NO	NO
8	527	NO	NO	512	NO	NO
9	523	NO	NO	530	NO	NO
10	521	NO	NO	521	NO	NO
11	530	NO	NO	532	NO	NO
12	513	NO	NO	520	NO	NO
13	512	NO	NO	514	NO	NO
14	512	NO	NO	541	NO	NO
15	534	NO	NO	552	NO	NO
16	530	NO	NO	514	NO	NO
17	512	NO	NO	530	NO	NO
18	521	NO	NO	521	NO	NO
19	521	NO	NO	549	NO	NO
20	520	NO	NO	512	NO	NO
21	528	NO	NO	513	NO	NO
22	512	NO	NO	512	NO	NO
23	515	NO	NO	521	NO	NO
24	534	NO	NO	538	NO	NO
25	531	NO	NO	550	NO	NO
26	514	NO	NO	512	NO	NO
27	525	NO	NO	525	NO	NO
28	521	NO	NO	534	NO	NO
29	540	NO	NO	531	NO	NO
30	530	NO	NO	512	NO	NO

Table 21: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 4.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	516	NO	NO	520	NO	NO
2	527	NO	NO	517	NO	NO
3	521	NO	NO	516	NO	NO
4	516	NO	NO	521	NO	NO
5	520	NO	NO	516	NO	NO
6	516	NO	NO	522	NO	NO
7	516	NO	NO	517	NO	NO
8	521	NO	NO	521	NO	NO
9	516	NO	NO	520	NO	NO
10	519	NO	NO	516	NO	NO
11	520	NO	NO	525	NO	NO
12	526	NO	NO	516	NO	NO
13	521	NO	NO	521	NO	NO
14	520	NO	NO	528	NO	NO
15	516	NO	NO	527	NO	NO
16	516	NO	NO	521	NO	NO
17	521	NO	NO	520	NO	NO
18	516	NO	NO	520	NO	NO
19	532	YES	NO	525	NO	NO
20	516	NO	NO	516	NO	NO
21	520	NO	NO	517	NO	NO
22	516	NO	NO	520	NO	NO
23	516	NO	NO	521	NO	NO
24	518	NO	NO	516	NO	NO
25	520	NO	NO	527	NO	NO
26	520	NO	NO	516	NO	NO
27	521	NO	NO	527	NO	NO
28	521	NO	NO	517	NO	NO
29	516	NO	NO	522	NO	NO
30	516	NO	NO	520	NO	NO

Table 22: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 4.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	503	NO	NO	505	NO	NO
2	508	NO	NO	508	NO	NO
3	500	NO	NO	510	NO	NO
4	509	NO	NO	510	NO	NO
5	500	NO	NO	512	NO	NO
6	495	NO	NO	512	NO	NO
7	502	NO	NO	502	YES	NO
8	510	NO	NO	508	NO	NO
9	509	NO	NO	510	NO	NO
10	495	NO	NO	510	NO	NO
11	495	NO	NO	510	NO	NO
12	510	NO	NO	508	NO	NO
13	500	NO	NO	510	NO	NO
14	500	NO	NO	498	YES	YES
15	510	NO	NO	508	NO	NO
16	495	NO	NO	510	NO	NO
17	508	NO	NO	510	NO	NO
18	510	NO	NO	512	NO	NO
19	508	NO	NO	495	YES	YES
20	508	NO	NO	512	NO	NO
21	499	NO	NO	510	NO	NO
22	508	NO	NO	509	NO	NO
23	500	NO	NO	521	YES	YES
24	505	NO	NO	513	NO	NO
25	503	NO	NO	508	NO	NO
26	510	NO	NO	506	NO	NO
27	508	NO	NO	508	NO	NO
28	502	NO	NO	508	NO	NO
29	508	NO	NO	508	NO	NO
30	509	NO	NO	508	NO	NO

Table 23: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 4.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	514	YES	NO	518	NO	NO
2	517	NO	NO	518	NO	NO
3	516	NO	NO	518	NO	NO
4	518	NO	NO	517	NO	NO
5	517	NO	NO	516	NO	NO
6	514	YES	NO	517	NO	NO
7	526	YES	YES	512	YES	YES
8	517	NO	NO	518	NO	NO
9	517	NO	NO	517	NO	NO
10	517	NO	NO	526	YES	YES
11	517	NO	NO	518	NO	NO
12	517	NO	NO	518	NO	NO
13	517	NO	NO	517	NO	NO
14	518	NO	NO	518	NO	NO
15	518	NO	NO	518	NO	NO
16	518	NO	NO	526	YES	YES
17	526	YES	YES	517	NO	NO
18	517	NO	NO	514	YES	NO
19	514	YES	NO	514	YES	NO
20	518	NO	NO	518	NO	NO
21	518	NO	NO	518	NO	NO
22	516	NO	NO	517	NO	NO
23	514	YES	NO	518	NO	NO
24	526	YES	YES	517	NO	NO
25	516	NO	NO	518	NO	NO
26	517	NO	NO	526	YES	YES
27	517	NO	NO	518	NO	NO
28	518	NO	NO	517	NO	NO
29	518	NO	NO	517	NO	NO
30	518	NO	NO	517	NO	NO

Table 24: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 4.6 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	590	NO	NO	589	NO	NO
2	575	NO	NO	562	NO	NO
3	572	NO	NO	571	NO	NO
4	562	NO	NO	571	NO	NO
5	562	NO	NO	562	NO	NO
6	562	NO	NO	560	NO	NO
7	565	NO	NO	574	NO	NO
8	561	NO	NO	577	NO	NO
9	571	NO	NO	572	NO	NO
10	585	NO	NO	562	NO	NO
11	562	NO	NO	566	NO	NO
12	561	NO	NO	562	NO	NO
13	561	NO	NO	577	NO	NO
14	562	NO	NO	574	NO	NO
15	562	NO	NO	560	NO	NO
16	576	NO	NO	562	NO	NO
17	574	NO	NO	571	NO	NO
18	578	NO	NO	570	NO	NO
19	562	NO	NO	579	NO	NO
20	575	NO	NO	576	NO	NO
21	562	NO	NO	572	NO	NO
22	564	NO	NO	569	NO	NO
23	575	NO	NO	574	NO	NO
24	575	NO	NO	577	NO	NO
25	562	NO	NO	592	NO	NO
26	563	NO	NO	569	NO	NO
27	573	NO	NO	588	NO	NO
28	560	NO	NO	561	NO	NO
29	572	NO	NO	565	NO	NO
30	572	NO	NO	584	NO	NO

Table 25: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 4.7 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	433	NO	NO	439	NO	NO
2	436	NO	NO	432	NO	NO
3	433	NO	NO	432	NO	NO
4	437	NO	NO	433	NO	NO
5	434	NO	NO	436	NO	NO
6	440	NO	NO	433	NO	NO
7	432	NO	NO	436	NO	NO
8	434	NO	NO	436	NO	NO
9	431	NO	NO	433	NO	NO
10	432	NO	NO	436	NO	NO
11	432	NO	NO	434	NO	NO
12	436	NO	NO	435	NO	NO
13	434	NO	NO	433	NO	NO
14	433	NO	NO	432	NO	NO
15	432	NO	NO	434	NO	NO
16	433	NO	NO	433	NO	NO
17	433	NO	NO	436	NO	NO
18	435	NO	NO	436	NO	NO
19	433	NO	NO	434	NO	NO
20	436	NO	NO	437	NO	NO
21	437	NO	NO	436	NO	NO
22	434	NO	NO	432	NO	NO
23	436	NO	NO	440	NO	NO
24	436	NO	NO	434	NO	NO
25	441	YES	NO	434	NO	NO
26	433	NO	NO	435	NO	NO
27	434	NO	NO	435	NO	NO
28	437	NO	NO	433	NO	NO
29	436	NO	NO	441	YES	NO
30	435	NO	NO	435	NO	NO

Table 26: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 4.8 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	495	NO	NO	499	NO	NO
2	499	NO	NO	497	NO	NO
3	497	NO	NO	505	YES	NO
4	497	NO	NO	495	NO	NO
5	499	NO	NO	499	NO	NO
6	493	YES	NO	499	NO	NO
7	505	YES	NO	497	NO	NO
8	493	YES	NO	497	NO	NO
9	499	NO	NO	499	NO	NO
10	501	NO	NO	499	NO	NO
11	499	NO	NO	499	NO	NO
12	499	NO	NO	492	YES	NO
13	499	NO	NO	499	NO	NO
14	499	NO	NO	499	NO	NO
15	494	NO	NO	499	NO	NO
16	499	NO	NO	499	NO	NO
17	499	NO	NO	499	NO	NO
18	499	NO	NO	494	NO	NO
19	499	NO	NO	505	YES	NO
20	499	NO	NO	506	YES	YES
21	497	NO	NO	499	NO	NO
22	497	NO	NO	505	YES	NO
23	497	NO	NO	499	NO	NO
24	499	NO	NO	497	NO	NO
25	499	NO	NO	496	NO	NO
26	497	NO	NO	497	NO	NO
27	499	NO	NO	493	YES	NO
28	499	NO	NO	495	NO	NO
29	499	NO	NO	499	NO	NO
30	499	NO	NO	499	NO	NO

Table 27: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 4.9 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	665	NO	NO	660	NO	NO
2	677	NO	NO	663	NO	NO
3	649	NO	NO	665	NO	NO
4	656	NO	NO	662	NO	NO
5	665	NO	NO	658	NO	NO
6	665	NO	NO	665	NO	NO
7	660	NO	NO	645	YES	NO
8	661	NO	NO	666	NO	NO
9	661	NO	NO	646	NO	NO
10	655	NO	NO	659	NO	NO
11	654	NO	NO	670	NO	NO
12	655	NO	NO	658	NO	NO
13	665	NO	NO	655	NO	NO
14	663	NO	NO	663	NO	NO
15	660	NO	NO	684	YES	NO
16	657	NO	NO	666	NO	NO
17	667	NO	NO	659	NO	NO
18	649	NO	NO	649	NO	NO
19	659	NO	NO	664	NO	NO
20	663	NO	NO	655	NO	NO
21	662	NO	NO	666	NO	NO
22	660	NO	NO	658	NO	NO
23	655	NO	NO	655	NO	NO
24	664	NO	NO	655	NO	NO
25	660	NO	NO	684	YES	NO
26	657	NO	NO	664	NO	NO
27	655	NO	NO	669	NO	NO
28	666	NO	NO	664	NO	NO
29	656	NO	NO	664	NO	NO
30	657	NO	NO	664	NO	NO

Table 28: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 4.10 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	521	NO	NO	545	YES	YES
2	514	NO	NO	516	NO	NO
3	523	NO	NO	518	NO	NO
4	525	NO	NO	519	NO	NO
5	518	NO	NO	515	NO	NO
6	514	NO	NO	520	NO	NO
7	534	NO	NO	518	NO	NO
8	529	NO	NO	520	NO	NO
9	514	NO	NO	532	YES	NO
10	520	NO	NO	521	NO	NO
11	516	NO	NO	519	NO	NO
12	529	NO	NO	521	NO	NO
13	523	NO	NO	521	NO	NO
14	520	NO	NO	514	NO	NO
15	532	NO	NO	519	NO	NO
16	532	NO	NO	519	NO	NO
17	515	NO	NO	516	NO	NO
18	517	NO	NO	519	NO	NO
19	525	NO	NO	521	NO	NO
20	520	NO	NO	521	NO	NO
21	516	NO	NO	526	NO	NO
22	532	NO	NO	523	NO	NO
23	529	NO	NO	518	NO	NO
24	518	NO	NO	517	NO	NO
25	541	NO	NO	519	NO	NO
26	518	NO	NO	517	NO	NO
27	516	NO	NO	523	NO	NO
28	523	NO	NO	520	NO	NO
29	519	NO	NO	523	NO	NO
30	529	NO	NO	523	NO	NO

Table 29: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 5.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	261	NO	NO	267	NO	NO
2	265	NO	NO	260	NO	NO
3	267	NO	NO	261	NO	NO
4	265	NO	NO	266	NO	NO
5	256	NO	NO	268	NO	NO
6	261	NO	NO	253	YES	NO
7	259	NO	NO	263	NO	NO
8	265	NO	NO	254	NO	NO
9	276	NO	NO	262	NO	NO
10	261	NO	NO	265	NO	NO
11	267	NO	NO	268	NO	NO
12	261	NO	NO	256	NO	NO
13	266	NO	NO	264	NO	NO
14	267	NO	NO	260	NO	NO
15	268	NO	NO	264	NO	NO
16	259	NO	NO	264	NO	NO
17	262	NO	NO	261	NO	NO
18	266	NO	NO	260	NO	NO
19	267	NO	NO	257	NO	NO
20	262	NO	NO	253	YES	NO
21	262	NO	NO	272	YES	NO
22	258	NO	NO	263	NO	NO
23	267	NO	NO	261	NO	NO
24	265	NO	NO	264	NO	NO
25	261	NO	NO	261	NO	NO
26	268	NO	NO	264	NO	NO
27	259	NO	NO	262	NO	NO
28	260	NO	NO	262	NO	NO
29	261	NO	NO	260	NO	NO
30	255	NO	NO	266	NO	NO

Table 30: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 5.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	312	NO	NO	313	NO	NO
2	317	YES	NO	310	NO	NO
3	311	NO	NO	310	NO	NO
4	313	NO	NO	313	NO	NO
5	313	NO	NO	312	NO	NO
6	313	NO	NO	314	NO	NO
7	311	NO	NO	311	NO	NO
8	313	NO	NO	312	NO	NO
9	315	NO	NO	313	NO	NO
10	311	NO	NO	314	NO	NO
11	314	NO	NO	311	NO	NO
12	314	NO	NO	313	NO	NO
13	317	YES	NO	309	NO	NO
14	311	NO	NO	307	NO	NO
15	313	NO	NO	306	NO	NO
16	311	NO	NO	311	NO	NO
17	314	NO	NO	318	NO	NO
18	314	NO	NO	314	NO	NO
19	311	NO	NO	317	NO	NO
20	313	NO	NO	306	NO	NO
21	311	NO	NO	313	NO	NO
22	311	NO	NO	316	NO	NO
23	311	NO	NO	316	NO	NO
24	311	NO	NO	308	NO	NO
25	311	NO	NO	307	NO	NO
26	311	NO	NO	313	NO	NO
27	311	NO	NO	312	NO	NO
28	311	NO	NO	317	NO	NO
29	311	NO	NO	311	NO	NO
30	312	NO	NO	311	NO	NO

Table 31: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 5.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	230	NO	NO	230	NO	NO
2	230	NO	NO	230	NO	NO
3	232	NO	NO	240	YES	NO
4	230	NO	NO	232	NO	NO
5	229	NO	NO	231	NO	NO
6	230	NO	NO	231	NO	NO
7	231	NO	NO	228	NO	NO
8	229	NO	NO	230	NO	NO
9	231	NO	NO	228	NO	NO
10	230	NO	NO	230	NO	NO
11	231	NO	NO	233	NO	NO
12	228	NO	NO	229	NO	NO
13	231	NO	NO	233	NO	NO
14	230	NO	NO	228	NO	NO
15	230	NO	NO	230	NO	NO
16	234	YES	NO	234	NO	NO
17	230	NO	NO	228	NO	NO
18	230	NO	NO	228	NO	NO
19	230	NO	NO	230	NO	NO
20	231	NO	NO	233	NO	NO
21	228	NO	NO	230	NO	NO
22	228	NO	NO	231	NO	NO
23	230	NO	NO	229	NO	NO
24	231	NO	NO	243	YES	YES
25	228	NO	NO	231	NO	NO
26	230	NO	NO	230	NO	NO
27	229	NO	NO	229	NO	NO
28	230	NO	NO	229	NO	NO
29	231	NO	NO	226	NO	NO
30	241	YES	YES	232	NO	NO

Table 32: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 5.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	242	NO	NO	245	NO	NO
2	245	NO	NO	245	NO	NO
3	242	NO	NO	245	NO	NO
4	242	NO	NO	245	NO	NO
5	243	NO	NO	243	NO	NO
6	248	NO	NO	243	NO	NO
7	245	NO	NO	243	NO	NO
8	243	NO	NO	242	NO	NO
9	245	NO	NO	242	NO	NO
10	245	NO	NO	243	NO	NO
11	242	NO	NO	245	NO	NO
12	245	NO	NO	242	NO	NO
13	247	NO	NO	245	NO	NO
14	243	NO	NO	242	NO	NO
15	242	NO	NO	242	NO	NO
16	245	NO	NO	242	NO	NO
17	243	NO	NO	244	NO	NO
18	245	NO	NO	242	NO	NO
19	245	NO	NO	245	NO	NO
20	242	NO	NO	242	NO	NO
21	242	NO	NO	245	NO	NO
22	243	NO	NO	245	NO	NO
23	245	NO	NO	245	NO	NO
24	242	NO	NO	243	NO	NO
25	246	NO	NO	249	NO	NO
26	243	NO	NO	246	NO	NO
27	242	NO	NO	245	NO	NO
28	245	NO	NO	245	NO	NO
29	245	NO	NO	243	NO	NO
30	243	NO	NO	242	NO	NO

Table 33: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 5.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	212	NO	NO	212	NO	NO
2	211	YES	YES	212	NO	NO
3	212	NO	NO	212	NO	NO
4	212	NO	NO	215	YES	YES
5	212	NO	NO	212	NO	NO
6	211	YES	YES	212	NO	NO
7	212	NO	NO	212	NO	NO
8	216	YES	YES	212	NO	NO
9	212	NO	NO	215	YES	YES
10	212	NO	NO	212	NO	NO
11	212	NO	NO	212	NO	NO
12	211	YES	YES	211	YES	YES
13	212	NO	NO	211	YES	YES
14	212	NO	NO	212	NO	NO
15	212	NO	NO	212	NO	NO
16	212	NO	NO	214	YES	YES
17	212	NO	NO	211	YES	YES
18	211	YES	YES	212	NO	NO
19	212	NO	NO	212	NO	NO
20	212	NO	NO	212	NO	NO
21	212	NO	NO	217	YES	YES
22	211	YES	YES	212	NO	NO
23	212	NO	NO	211	YES	YES
24	212	NO	NO	212	NO	NO
25	211	YES	YES	212	NO	NO
26	212	NO	NO	212	NO	NO
27	212	NO	NO	211	YES	YES
28	212	NO	NO	214	YES	YES
29	212	NO	NO	215	YES	YES
30	212	NO	NO	212	NO	NO

Table 34: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 5.6 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	233	NO	NO	213	NO	NO
2	216	NO	NO	225	NO	NO
3	228	NO	NO	217	NO	NO
4	214	NO	NO	226	NO	NO
5	218	NO	NO	226	NO	NO
6	216	NO	NO	214	NO	NO
7	226	NO	NO	232	NO	NO
8	230	NO	NO	217	NO	NO
9	216	NO	NO	226	NO	NO
10	239	NO	NO	217	NO	NO
11	237	NO	NO	218	NO	NO
12	214	NO	NO	226	NO	NO
13	226	NO	NO	217	NO	NO
14	217	NO	NO	214	NO	NO
15	216	NO	NO	226	NO	NO
16	216	NO	NO	229	NO	NO
17	217	NO	NO	216	NO	NO
18	217	NO	NO	223	NO	NO
19	218	NO	NO	217	NO	NO
20	222	NO	NO	235	NO	NO
21	214	NO	NO	218	NO	NO
22	217	NO	NO	232	NO	NO
23	223	NO	NO	219	NO	NO
24	214	NO	NO	216	NO	NO
25	216	NO	NO	232	NO	NO
26	223	NO	NO	214	NO	NO
27	218	NO	NO	221	NO	NO
28	214	NO	NO	229	NO	NO
29	233	NO	NO	226	NO	NO
30	217	NO	NO	216	NO	NO

Table 35: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 5.7 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	304	NO	NO	311	NO	NO
2	301	NO	NO	313	NO	NO
3	304	NO	NO	307	NO	NO
4	301	NO	NO	303	NO	NO
5	302	NO	NO	303	NO	NO
6	304	NO	NO	313	NO	NO
7	306	NO	NO	307	NO	NO
8	305	NO	NO	293	NO	NO
9	304	NO	NO	298	NO	NO
10	314	NO	NO	312	NO	NO
11	308	NO	NO	297	NO	NO
12	307	NO	NO	307	NO	NO
13	298	NO	NO	309	NO	NO
14	299	NO	NO	293	NO	NO
15	301	NO	NO	312	NO	NO
16	308	NO	NO	309	NO	NO
17	301	NO	NO	301	NO	NO
18	293	NO	NO	304	NO	NO
19	310	NO	NO	308	NO	NO
20	304	NO	NO	301	NO	NO
21	307	NO	NO	299	NO	NO
22	305	NO	NO	299	NO	NO
23	301	NO	NO	300	NO	NO
24	300	NO	NO	293	NO	NO
25	303	NO	NO	311	NO	NO
26	307	NO	NO	308	NO	NO
27	308	NO	NO	306	NO	NO
28	302	NO	NO	301	NO	NO
29	308	NO	NO	307	NO	NO
30	312	NO	NO	305	NO	NO

Table 36: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 5.8 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	303	NO	NO	300	NO	NO
2	294	NO	NO	302	NO	NO
3	301	NO	NO	288	NO	NO
4	290	NO	NO	300	NO	NO
5	299	NO	NO	291	NO	NO
6	288	NO	NO	304	NO	NO
7	304	NO	NO	305	NO	NO
8	299	NO	NO	298	NO	NO
9	291	NO	NO	293	NO	NO
10	289	NO	NO	298	NO	NO
11	306	NO	NO	296	NO	NO
12	289	NO	NO	299	NO	NO
13	305	NO	NO	293	NO	NO
14	295	NO	NO	293	NO	NO
15	294	NO	NO	291	NO	NO
16	290	NO	NO	295	NO	NO
17	301	NO	NO	302	NO	NO
18	299	NO	NO	292	NO	NO
19	306	NO	NO	297	NO	NO
20	305	NO	NO	293	NO	NO
21	300	NO	NO	292	NO	NO
22	302	NO	NO	303	NO	NO
23	295	NO	NO	291	NO	NO
24	289	NO	NO	289	NO	NO
25	291	NO	NO	304	NO	NO
26	295	NO	NO	289	NO	NO
27	304	NO	NO	301	NO	NO
28	289	NO	NO	297	NO	NO
29	310	NO	NO	301	NO	NO
30	303	NO	NO	300	NO	NO

Table 37: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 5.9 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	280	NO	NO	281	NO	NO
2	280	NO	NO	280	NO	NO
3	280	NO	NO	288	NO	NO
4	288	NO	NO	279	NO	NO
5	280	NO	NO	288	NO	NO
6	280	NO	NO	288	NO	NO
7	289	NO	NO	280	NO	NO
8	281	NO	NO	289	NO	NO
9	279	NO	NO	280	NO	NO
10	288	NO	NO	279	NO	NO
11	288	NO	NO	280	NO	NO
12	287	NO	NO	280	NO	NO
13	287	NO	NO	288	NO	NO
14	279	NO	NO	279	NO	NO
15	291	NO	NO	291	NO	NO
16	295	NO	NO	280	NO	NO
17	279	NO	NO	289	NO	NO
18	279	NO	NO	279	NO	NO
19	288	NO	NO	291	NO	NO
20	286	NO	NO	289	NO	NO
21	289	NO	NO	279	NO	NO
22	280	NO	NO	280	NO	NO
23	286	NO	NO	280	NO	NO
24	294	NO	NO	292	NO	NO
25	279	NO	NO	280	NO	NO
26	280	NO	NO	280	NO	NO
27	280	NO	NO	281	NO	NO
28	290	NO	NO	280	NO	NO
29	289	NO	NO	291	NO	NO
30	290	NO	NO	280	NO	NO

Table 38: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 5.10 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	272	NO	NO	268	NO	NO
2	273	NO	NO	271	NO	NO
3	271	NO	NO	273	NO	NO
4	272	NO	NO	274	NO	NO
5	271	NO	NO	274	NO	NO
6	272	NO	NO	270	NO	NO
7	270	NO	NO	270	NO	NO
8	271	NO	NO	271	NO	NO
9	269	NO	NO	272	NO	NO
10	275	YES	NO	270	NO	NO
11	267	YES	NO	272	NO	NO
12	268	YES	NO	269	NO	NO
13	272	NO	NO	265	YES	NO
14	269	NO	NO	272	NO	NO
15	271	NO	NO	268	NO	NO
16	272	NO	NO	272	NO	NO
17	272	NO	NO	271	NO	NO
18	271	NO	NO	271	NO	NO
19	273	NO	NO	269	NO	NO
20	272	NO	NO	273	NO	NO
21	268	YES	NO	272	NO	NO
22	272	NO	NO	272	NO	NO
23	273	NO	NO	272	NO	NO
24	272	NO	NO	269	NO	NO
25	270	NO	NO	273	NO	NO
26	272	NO	NO	272	NO	NO
27	272	NO	NO	272	NO	NO
28	271	NO	NO	267	NO	NO
29	272	NO	NO	272	NO	NO
30	271	NO	NO	272	NO	NO

Table 39: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 6.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	146	NO	NO	144	NO	NO
2	145	NO	NO	138	YES	NO
3	143	NO	NO	143	NO	NO
4	146	NO	NO	145	NO	NO
5	145	NO	NO	144	NO	NO
6	141	NO	NO	143	NO	NO
7	138	NO	NO	145	NO	NO
8	145	NO	NO	146	NO	NO
9	147	NO	NO	138	YES	NO
10	141	NO	NO	144	NO	NO
11	144	NO	NO	144	NO	NO
12	149	NO	NO	149	NO	NO
13	143	NO	NO	143	NO	NO
14	143	NO	NO	142	NO	NO
15	140	NO	NO	148	NO	NO
16	143	NO	NO	145	NO	NO
17	142	NO	NO	144	NO	NO
18	144	NO	NO	138	YES	NO
19	145	NO	NO	146	NO	NO
20	143	NO	NO	149	NO	NO
21	143	NO	NO	148	NO	NO
22	143	NO	NO	143	NO	NO
23	142	NO	NO	145	NO	NO
24	145	NO	NO	143	NO	NO
25	145	NO	NO	147	NO	NO
26	141	NO	NO	143	NO	NO
27	146	NO	NO	144	NO	NO
28	151	YES	NO	142	NO	NO
29	145	NO	NO	145	NO	NO
30	138	NO	NO	146	NO	NO

Table 40: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 6.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	154	NO	NO	148	NO	NO
2	151	NO	NO	152	NO	NO
3	151	NO	NO	155	NO	NO
4	154	NO	NO	150	NO	NO
5	153	NO	NO	147	NO	NO
6	154	NO	NO	153	NO	NO
7	151	NO	NO	151	NO	NO
8	151	NO	NO	148	NO	NO
9	152	NO	NO	153	NO	NO
10	150	NO	NO	153	NO	NO
11	150	NO	NO	151	NO	NO
12	147	NO	NO	155	NO	NO
13	151	NO	NO	147	NO	NO
14	150	NO	NO	155	NO	NO
15	148	NO	NO	154	NO	NO
16	150	NO	NO	150	NO	NO
17	151	NO	NO	150	NO	NO
18	149	NO	NO	147	NO	NO
19	151	NO	NO	154	NO	NO
20	153	NO	NO	151	NO	NO
21	151	NO	NO	147	NO	NO
22	147	NO	NO	154	NO	NO
23	147	NO	NO	151	NO	NO
24	151	NO	NO	149	NO	NO
25	148	NO	NO	151	NO	NO
26	152	NO	NO	150	NO	NO
27	153	NO	NO	147	NO	NO
28	153	NO	NO	153	NO	NO
29	151	NO	NO	147	NO	NO
30	148	NO	NO	150	NO	NO

Table 41: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 6.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	156	NO	NO	150	NO	NO
2	155	NO	NO	156	YES	NO
3	148	NO	NO	154	NO	NO
4	155	NO	NO	150	NO	NO
5	150	NO	NO	150	NO	NO
6	150	NO	NO	150	NO	NO
7	148	NO	NO	151	NO	NO
8	145	NO	NO	148	NO	NO
9	148	NO	NO	149	NO	NO
10	148	NO	NO	148	NO	NO
11	150	NO	NO	151	NO	NO
12	150	NO	NO	148	NO	NO
13	145	NO	NO	151	NO	NO
14	149	NO	NO	151	NO	NO
15	148	NO	NO	148	NO	NO
16	148	NO	NO	151	NO	NO
17	157	NO	NO	152	NO	NO
18	156	NO	NO	148	NO	NO
19	157	NO	NO	156	YES	NO
20	148	NO	NO	148	NO	NO
21	150	NO	NO	148	NO	NO
22	150	NO	NO	150	NO	NO
23	150	NO	NO	148	NO	NO
24	152	NO	NO	145	NO	NO
25	148	NO	NO	148	NO	NO
26	149	NO	NO	148	NO	NO
27	145	NO	NO	152	NO	NO
28	153	NO	NO	156	YES	NO
29	149	NO	NO	150	NO	NO
30	148	NO	NO	145	NO	NO

Table 42: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 6.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	134	NO	NO	133	NO	NO
2	133	NO	NO	134	NO	NO
3	132	NO	NO	131	NO	NO
4	134	NO	NO	134	NO	NO
5	134	NO	NO	131	NO	NO
6	137	NO	NO	134	NO	NO
7	134	NO	NO	131	NO	NO
8	131	NO	NO	133	NO	NO
9	131	NO	NO	133	NO	NO
10	133	NO	NO	133	NO	NO
11	134	NO	NO	131	NO	NO
12	131	NO	NO	131	NO	NO
13	133	NO	NO	137	NO	NO
14	131	NO	NO	133	NO	NO
15	133	NO	NO	131	NO	NO
16	134	NO	NO	131	NO	NO
17	131	NO	NO	131	NO	NO
18	131	NO	NO	132	NO	NO
19	134	NO	NO	134	NO	NO
20	131	NO	NO	133	NO	NO
21	131	NO	NO	131	NO	NO
22	131	NO	NO	134	NO	NO
23	134	NO	NO	131	NO	NO
24	134	NO	NO	136	NO	NO
25	131	NO	NO	133	NO	NO
26	139	YES	NO	131	NO	NO
27	135	NO	NO	135	NO	NO
28	134	NO	NO	131	NO	NO
29	134	NO	NO	131	NO	NO
30	131	NO	NO	134	NO	NO

Table 43: Outlier Analysis of Improved and Simplified version AFSA, *SCP* 6.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	172	NO	NO	164	NO	NO
2	167	NO	NO	164	NO	NO
3	169	NO	NO	172	NO	NO
4	165	NO	NO	177	NO	NO
5	163	NO	NO	175	NO	NO
6	164	NO	NO	168	NO	NO
7	164	NO	NO	169	NO	NO
8	165	NO	NO	169	NO	NO
9	169	NO	NO	167	NO	NO
10	173	NO	NO	166	NO	NO
11	166	NO	NO	163	NO	NO
12	164	NO	NO	168	NO	NO
13	165	NO	NO	171	NO	NO
14	161	NO	NO	161	NO	NO
15	168	NO	NO	170	NO	NO
16	171	NO	NO	176	NO	NO
17	180	YES	NO	179	NO	NO
18	176	NO	NO	173	NO	NO
19	162	NO	NO	171	NO	NO
20	165	NO	NO	175	NO	NO
21	165	NO	NO	170	NO	NO
22	175	NO	NO	175	NO	NO
23	172	NO	NO	169	NO	NO
24	163	NO	NO	170	NO	NO
25	167	NO	NO	166	NO	NO
26	165	NO	NO	166	NO	NO
27	168	NO	NO	169	NO	NO
28	169	NO	NO	176	NO	NO
29	168	NO	NO	180	NO	NO
30	169	NO	NO	175	NO	NO

Table 44: Outlier Analysis of Improved and Simplified version AFSA, *SCP A*.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	265	NO	NO	260	NO	NO
2	256	NO	NO	260	NO	NO
3	253	NO	NO	255	NO	NO
4	258	NO	NO	254	NO	NO
5	256	NO	NO	258	NO	NO
6	261	NO	NO	258	NO	NO
7	255	NO	NO	255	NO	NO
8	258	NO	NO	255	NO	NO
9	258	NO	NO	260	NO	NO
10	258	NO	NO	256	NO	NO
11	261	NO	NO	256	NO	NO
12	256	NO	NO	257	NO	NO
13	259	NO	NO	260	NO	NO
14	260	NO	NO	256	NO	NO
15	257	NO	NO	255	NO	NO
16	257	NO	NO	258	NO	NO
17	261	NO	NO	256	NO	NO
18	262	NO	NO	259	NO	NO
19	261	NO	NO	258	NO	NO
20	257	NO	NO	262	NO	NO
21	256	NO	NO	258	NO	NO
22	255	NO	NO	257	NO	NO
23	262	NO	NO	258	NO	NO
24	256	NO	NO	264	NO	NO
25	260	NO	NO	258	NO	NO
26	257	NO	NO	261	NO	NO
27	262	NO	NO	257	NO	NO
28	258	NO	NO	255	NO	NO
29	257	NO	NO	258	NO	NO
30	261	NO	NO	258	NO	NO

Table 45: Outlier Analysis of Improved and Simplified version AFSA, *SCP A*.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	276	YES	NO	263	NO	NO
2	262	NO	NO	270	NO	NO
3	265	NO	NO	264	NO	NO
4	261	NO	NO	276	YES	NO
5	265	NO	NO	264	NO	NO
6	263	NO	NO	266	NO	NO
7	255	NO	NO	260	NO	NO
8	268	NO	NO	274	YES	NO
9	260	NO	NO	264	NO	NO
10	261	NO	NO	254	NO	NO
11	266	NO	NO	262	NO	NO
12	260	NO	NO	263	NO	NO
13	263	NO	NO	266	NO	NO
14	261	NO	NO	263	NO	NO
15	261	NO	NO	266	NO	NO
16	263	NO	NO	264	NO	NO
17	261	NO	NO	264	NO	NO
18	270	NO	NO	259	NO	NO
19	259	NO	NO	269	NO	NO
20	261	NO	NO	258	NO	NO
21	269	NO	NO	264	NO	NO
22	261	NO	NO	262	NO	NO
23	268	NO	NO	261	NO	NO
24	261	NO	NO	264	NO	NO
25	266	NO	NO	261	NO	NO
26	264	NO	NO	261	NO	NO
27	261	NO	NO	269	NO	NO
28	265	NO	NO	258	NO	NO
29	263	NO	NO	259	NO	NO
30	261	NO	NO	259	NO	NO

Table 46: Outlier Analysis of Improved and Simplified version AFSA, *SCP A*.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	242	NO	NO	249	NO	NO
2	241	NO	NO	238	NO	NO
3	236	NO	NO	238	NO	NO
4	240	NO	NO	236	NO	NO
5	238	NO	NO	239	NO	NO
6	235	NO	NO	235	NO	NO
7	243	NO	NO	244	NO	NO
8	240	NO	NO	246	NO	NO
9	237	NO	NO	239	NO	NO
10	242	NO	NO	241	NO	NO
11	246	NO	NO	240	NO	NO
12	236	NO	NO	239	NO	NO
13	239	NO	NO	246	NO	NO
14	244	NO	NO	242	NO	NO
15	243	NO	NO	240	NO	NO
16	243	NO	NO	246	NO	NO
17	235	NO	NO	242	NO	NO
18	242	NO	NO	235	NO	NO
19	242	NO	NO	236	NO	NO
20	240	NO	NO	240	NO	NO
21	239	NO	NO	237	NO	NO
22	237	NO	NO	238	NO	NO
23	242	NO	NO	238	NO	NO
24	241	NO	NO	235	NO	NO
25	248	NO	NO	247	NO	NO
26	238	NO	NO	238	NO	NO
27	242	NO	NO	238	NO	NO
28	237	NO	NO	245	NO	NO
29	244	NO	NO	248	NO	NO
30	235	NO	NO	237	NO	NO

Table 47: Outlier Analysis of Improved and Simplified version AFSA, *SCP A*.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	239	NO	NO	244	NO	NO
2	239	NO	NO	237	NO	NO
3	245	NO	NO	240	NO	NO
4	239	NO	NO	247	NO	NO
5	239	NO	NO	246	NO	NO
6	238	NO	NO	245	NO	NO
7	241	NO	NO	239	NO	NO
8	243	NO	NO	246	NO	NO
9	238	NO	NO	243	NO	NO
10	240	NO	NO	249	NO	NO
11	237	NO	NO	244	NO	NO
12	235	NO	NO	244	NO	NO
13	240	NO	NO	246	NO	NO
14	244	NO	NO	246	NO	NO
15	238	NO	NO	247	NO	NO
16	240	NO	NO	237	NO	NO
17	237	NO	NO	237	NO	NO
18	251	NO	NO	239	NO	NO
19	240	NO	NO	242	NO	NO
20	251	NO	NO	240	NO	NO
21	240	NO	NO	246	NO	NO
22	240	NO	NO	249	NO	NO
23	249	NO	NO	237	NO	NO
24	244	NO	NO	236	NO	NO
25	247	NO	NO	243	NO	NO
26	244	NO	NO	247	NO	NO
27	247	NO	NO	243	NO	NO
28	245	NO	NO	240	NO	NO
29	239	NO	NO	242	NO	NO
30	252	NO	NO	241	NO	NO

Table 48: Outlier Analysis of Improved and Simplified version AFSA, *SCP A*.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	240	NO	NO	238	NO	NO
2	240	NO	NO	243	NO	NO
3	238	NO	NO	240	NO	NO
4	240	NO	NO	240	NO	NO
5	238	NO	NO	239	NO	NO
6	240	NO	NO	239	NO	NO
7	240	NO	NO	238	NO	NO
8	243	YES	NO	239	NO	NO
9	239	NO	NO	239	NO	NO
10	239	NO	NO	238	NO	NO
11	242	NO	NO	239	NO	NO
12	239	NO	NO	239	NO	NO
13	240	NO	NO	239	NO	NO
14	240	NO	NO	240	NO	NO
15	239	NO	NO	238	NO	NO
16	238	NO	NO	240	NO	NO
17	239	NO	NO	237	NO	NO
18	239	NO	NO	239	NO	NO
19	245	YES	YES	239	NO	NO
20	238	NO	NO	240	NO	NO
21	242	NO	NO	239	NO	NO
22	239	NO	NO	244	YES	NO
23	243	YES	NO	238	NO	NO
24	240	NO	NO	240	NO	NO
25	239	NO	NO	238	NO	NO
26	240	NO	NO	241	NO	NO
27	241	NO	NO	240	NO	NO
28	240	NO	NO	238	NO	NO
29	241	NO	NO	241	NO	NO
30	240	NO	NO	239	NO	NO

Table 49: Outlier Analysis of Improved and Simplified version AFSA, *SCP B*.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	81	NO	NO	73	NO	NO
2	74	NO	NO	76	NO	NO
3	78	NO	NO	81	NO	NO
4	72	NO	NO	73	NO	NO
5	73	NO	NO	77	NO	NO
6	76	NO	NO	77	NO	NO
7	83	NO	NO	77	NO	NO
8	74	NO	NO	76	NO	NO
9	73	NO	NO	74	NO	NO
10	71	NO	NO	78	NO	NO
11	77	NO	NO	75	NO	NO
12	78	NO	NO	77	NO	NO
13	77	NO	NO	75	NO	NO
14	70	NO	NO	75	NO	NO
15	79	NO	NO	78	NO	NO
16	77	NO	NO	69	NO	NO
17	77	NO	NO	70	NO	NO
18	78	NO	NO	69	NO	NO
19	72	NO	NO	73	NO	NO
20	72	NO	NO	74	NO	NO
21	77	NO	NO	74	NO	NO
22	74	NO	NO	73	NO	NO
23	72	NO	NO	75	NO	NO
24	80	NO	NO	73	NO	NO
25	79	NO	NO	77	NO	NO
26	74	NO	NO	76	NO	NO
27	72	NO	NO	79	NO	NO
28	72	NO	NO	74	NO	NO
29	71	NO	NO	76	NO	NO
30	76	NO	NO	69	NO	NO

Table 50: Outlier Analysis of Improved and Simplified version AFSA, *SCP B*.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	89	NO	NO	82	NO	NO
2	84	NO	NO	82	NO	NO
3	82	NO	NO	83	NO	NO
4	83	NO	NO	82	NO	NO
5	82	NO	NO	83	NO	NO
6	81	NO	NO	84	NO	NO
7	82	NO	NO	83	NO	NO
8	84	NO	NO	84	NO	NO
9	81	NO	NO	76	YES	NO
10	80	NO	NO	86	NO	NO
11	78	NO	NO	87	NO	NO
12	81	NO	NO	81	NO	NO
13	80	NO	NO	83	NO	NO
14	84	NO	NO	87	NO	NO
15	83	NO	NO	85	NO	NO
16	87	NO	NO	82	NO	NO
17	78	NO	NO	86	NO	NO
18	87	NO	NO	85	NO	NO
19	89	NO	NO	81	NO	NO
20	79	NO	NO	76	YES	NO
21	86	NO	NO	82	NO	NO
22	85	NO	NO	86	NO	NO
23	84	NO	NO	90	NO	NO
24	82	NO	NO	82	NO	NO
25	83	NO	NO	82	NO	NO
26	82	NO	NO	85	NO	NO
27	88	NO	NO	85	NO	NO
28	80	NO	NO	85	NO	NO
29	82	NO	NO	87	NO	NO
30	84	NO	NO	82	NO	NO

Table 51: Outlier Analysis of Improved and Simplified version AFSA, *SCP B*.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	81	NO	NO	86	NO	NO
2	85	NO	NO	83	NO	NO
3	80	NO	NO	80	NO	NO
4	84	NO	NO	84	NO	NO
5	80	NO	NO	81	NO	NO
6	85	NO	NO	84	NO	NO
7	86	NO	NO	84	NO	NO
8	84	NO	NO	82	NO	NO
9	82	NO	NO	82	NO	NO
10	82	NO	NO	85	NO	NO
11	88	NO	NO	82	NO	NO
12	82	NO	NO	86	NO	NO
13	85	NO	NO	82	NO	NO
14	84	NO	NO	85	NO	NO
15	81	NO	NO	80	NO	NO
16	83	NO	NO	88	NO	NO
17	84	NO	NO	83	NO	NO
18	84	NO	NO	85	NO	NO
19	82	NO	NO	82	NO	NO
20	84	NO	NO	82	NO	NO
21	82	NO	NO	83	NO	NO
22	86	NO	NO	84	NO	NO
23	82	NO	NO	84	NO	NO
24	85	NO	NO	83	NO	NO
25	81	NO	NO	85	NO	NO
26	83	NO	NO	83	NO	NO
27	85	NO	NO	85	NO	NO
28	82	NO	NO	82	NO	NO
29	80	NO	NO	85	NO	NO
30	85	NO	NO	81	NO	NO

Table 52: Outlier Analysis of Improved and Simplified version AFSA, *SCP B*.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	83	NO	NO	84	NO	NO
2	82	NO	NO	83	NO	NO
3	86	NO	NO	87	NO	NO
4	83	NO	NO	88	NO	NO
5	86	NO	NO	84	NO	NO
6	85	NO	NO	85	NO	NO
7	85	NO	NO	82	NO	NO
8	83	NO	NO	86	NO	NO
9	81	NO	NO	84	NO	NO
10	85	NO	NO	79	YES	NO
11	84	NO	NO	83	NO	NO
12	86	NO	NO	84	NO	NO
13	86	NO	NO	84	NO	NO
14	84	NO	NO	83	NO	NO
15	84	NO	NO	82	NO	NO
16	84	NO	NO	88	NO	NO
17	88	NO	NO	83	NO	NO
18	84	NO	NO	84	NO	NO
19	84	NO	NO	82	NO	NO
20	85	NO	NO	80	NO	NO
21	83	NO	NO	83	NO	NO
22	85	NO	NO	86	NO	NO
23	81	NO	NO	83	NO	NO
24	81	NO	NO	88	NO	NO
25	85	NO	NO	86	NO	NO
26	84	NO	NO	85	NO	NO
27	82	NO	NO	84	NO	NO
28	84	NO	NO	80	NO	NO
29	84	NO	NO	84	NO	NO
30	84	NO	NO	83	NO	NO

Table 53: Outlier Analysis of Improved and Simplified version AFSA, *SCP B*.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	72	NO	NO	72	NO	NO
2	75	NO	NO	75	NO	NO
3	73	NO	NO	81	NO	NO
4	72	NO	NO	77	NO	NO
5	72	NO	NO	73	NO	NO
6	78	NO	NO	72	NO	NO
7	75	NO	NO	72	NO	NO
8	75	NO	NO	72	NO	NO
9	80	YES	NO	75	NO	NO
10	75	NO	NO	74	NO	NO
11	79	NO	NO	78	NO	NO
12	74	NO	NO	75	NO	NO
13	78	NO	NO	73	NO	NO
14	73	NO	NO	80	NO	NO
15	72	NO	NO	79	NO	NO
16	73	NO	NO	73	NO	NO
17	72	NO	NO	75	NO	NO
18	72	NO	NO	75	NO	NO
19	73	NO	NO	73	NO	NO
20	75	NO	NO	77	NO	NO
21	73	NO	NO	81	NO	NO
22	79	NO	NO	74	NO	NO
23	74	NO	NO	74	NO	NO
24	72	NO	NO	75	NO	NO
25	72	NO	NO	75	NO	NO
26	72	NO	NO	80	NO	NO
27	72	NO	NO	72	NO	NO
28	72	NO	NO	75	NO	NO
29	77	NO	NO	72	NO	NO
30	74	NO	NO	76	NO	NO

Table 54: Outlier Analysis of Improved and Simplified version AFSA, *SCP C*.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	237	NO	NO	228	YES	NO
2	234	NO	NO	233	NO	NO
3	234	NO	NO	237	NO	NO
4	235	NO	NO	238	YES	NO
5	233	NO	NO	232	NO	NO
6	233	NO	NO	234	NO	NO
7	236	NO	NO	236	NO	NO
8	231	NO	NO	232	NO	NO
9	234	NO	NO	233	NO	NO
10	236	NO	NO	234	NO	NO
11	236	NO	NO	233	NO	NO
12	231	NO	NO	234	NO	NO
13	233	NO	NO	229	NO	NO
14	233	NO	NO	235	NO	NO
15	234	NO	NO	229	NO	NO
16	229	NO	NO	233	NO	NO
17	232	NO	NO	236	NO	NO
18	233	NO	NO	232	NO	NO
19	233	NO	NO	233	NO	NO
20	236	NO	NO	236	NO	NO
21	236	NO	NO	232	NO	NO
22	234	NO	NO	233	NO	NO
23	233	NO	NO	230	NO	NO
24	232	NO	NO	233	NO	NO
25	234	NO	NO	232	NO	NO
26	230	NO	NO	229	NO	NO
27	234	NO	NO	234	NO	NO
28	232	NO	NO	230	NO	NO
29	237	NO	NO	232	NO	NO
30	232	NO	NO	233	NO	NO

Table 55: Outlier Analysis of Improved and Simplified version AFSA, *SCP C*.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	226	NO	NO	229	NO	NO
2	225	NO	NO	224	NO	NO
3	221	NO	NO	226	NO	NO
4	229	NO	NO	222	NO	NO
5	223	NO	NO	225	NO	NO
6	224	NO	NO	222	NO	NO
7	225	NO	NO	228	NO	NO
8	225	NO	NO	224	NO	NO
9	225	NO	NO	221	NO	NO
10	231	NO	NO	227	NO	NO
11	225	NO	NO	227	NO	NO
12	226	NO	NO	225	NO	NO
13	222	NO	NO	224	NO	NO
14	229	NO	NO	226	NO	NO
15	224	NO	NO	222	NO	NO
16	227	NO	NO	228	NO	NO
17	224	NO	NO	227	NO	NO
18	224	NO	NO	227	NO	NO
19	230	NO	NO	232	NO	NO
20	226	NO	NO	221	NO	NO
21	224	NO	NO	227	NO	NO
22	224	NO	NO	225	NO	NO
23	232	NO	NO	230	NO	NO
24	221	NO	NO	225	NO	NO
25	220	NO	NO	229	NO	NO
26	223	NO	NO	236	YES	NO
27	228	NO	NO	221	NO	NO
28	222	NO	NO	224	NO	NO
29	233	NO	NO	226	NO	NO
30	231	NO	NO	220	NO	NO

Table 56: Outlier Analysis of Improved and Simplified version AFSA, *SCP C*.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	260	NO	NO	250	NO	NO
2	263	NO	NO	257	NO	NO
3	260	NO	NO	274	YES	NO
4	256	NO	NO	250	NO	NO
5	253	NO	NO	260	NO	NO
6	263	NO	NO	250	NO	NO
7	254	NO	NO	264	NO	NO
8	254	NO	NO	256	NO	NO
9	257	NO	NO	251	NO	NO
10	250	NO	NO	255	NO	NO
11	251	NO	NO	257	NO	NO
12	256	NO	NO	259	NO	NO
13	254	NO	NO	257	NO	NO
14	275	YES	NO	251	NO	NO
15	256	NO	NO	252	NO	NO
16	262	NO	NO	262	NO	NO
17	253	NO	NO	253	NO	NO
18	255	NO	NO	260	NO	NO
19	260	NO	NO	257	NO	NO
20	256	NO	NO	250	NO	NO
21	258	NO	NO	256	NO	NO
22	266	NO	NO	256	NO	NO
23	252	NO	NO	261	NO	NO
24	256	NO	NO	267	NO	NO
25	253	NO	NO	254	NO	NO
26	256	NO	NO	257	NO	NO
27	253	NO	NO	255	NO	NO
28	254	NO	NO	252	NO	NO
29	253	NO	NO	252	NO	NO
30	256	NO	NO	259	NO	NO

Table 57: Outlier Analysis of Improved and Simplified version AFSA, *SCP C*.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	232	NO	NO	235	NO	NO
2	222	NO	NO	222	NO	NO
3	226	NO	NO	227	NO	NO
4	232	NO	NO	233	NO	NO
5	236	NO	NO	226	NO	NO
6	222	NO	NO	226	NO	NO
7	229	NO	NO	230	NO	NO
8	224	NO	NO	228	NO	NO
9	230	NO	NO	230	NO	NO
10	231	NO	NO	232	NO	NO
11	231	NO	NO	231	NO	NO
12	231	NO	NO	236	NO	NO
13	233	NO	NO	238	NO	NO
14	234	NO	NO	235	NO	NO
15	240	YES	NO	236	NO	NO
16	226	NO	NO	224	NO	NO
17	238	NO	NO	232	NO	NO
18	233	NO	NO	235	NO	NO
19	229	NO	NO	227	NO	NO
20	238	NO	NO	228	NO	NO
21	231	NO	NO	239	NO	NO
22	230	NO	NO	230	NO	NO
23	227	NO	NO	231	NO	NO
24	227	NO	NO	230	NO	NO
25	229	NO	NO	235	NO	NO
26	234	NO	NO	237	NO	NO
27	231	NO	NO	238	NO	NO
28	231	NO	NO	229	NO	NO
29	229	NO	NO	231	NO	NO
30	231	NO	NO	233	NO	NO

Table 58: Outlier Analysis of Improved and Simplified version AFSA, *SCP C*.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	218	NO	NO	219	NO	NO
2	222	NO	NO	219	NO	NO
3	222	NO	NO	221	NO	NO
4	218	NO	NO	218	NO	NO
5	222	NO	NO	217	NO	NO
6	217	NO	NO	221	NO	NO
7	225	NO	NO	218	NO	NO
8	224	NO	NO	218	NO	NO
9	217	NO	NO	221	NO	NO
10	221	NO	NO	225	NO	NO
11	222	NO	NO	216	NO	NO
12	218	NO	NO	217	NO	NO
13	218	NO	NO	217	NO	NO
14	221	NO	NO	218	NO	NO
15	217	NO	NO	223	NO	NO
16	217	NO	NO	215	NO	NO
17	218	NO	NO	221	NO	NO
18	219	NO	NO	217	NO	NO
19	217	NO	NO	220	NO	NO
20	221	NO	NO	218	NO	NO
21	227	NO	NO	219	NO	NO
22	219	NO	NO	218	NO	NO
23	218	NO	NO	221	NO	NO
24	218	NO	NO	217	NO	NO
25	222	NO	NO	217	NO	NO
26	218	NO	NO	220	NO	NO
27	217	NO	NO	219	NO	NO
28	217	NO	NO	218	NO	NO
29	218	NO	NO	218	NO	NO
30	219	NO	NO	218	NO	NO

Table 59: Outlier Analysis of Improved and Simplified version AFSA, *SCP D*.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	61	NO	NO	63	NO	NO
2	64	NO	NO	60	NO	NO
3	66	NO	NO	63	NO	NO
4	61	NO	NO	62	NO	NO
5	65	NO	NO	62	NO	NO
6	62	NO	NO	61	NO	NO
7	62	NO	NO	61	NO	NO
8	62	NO	NO	60	NO	NO
9	66	NO	NO	62	NO	NO
10	63	NO	NO	60	NO	NO
11	60	NO	NO	61	NO	NO
12	62	NO	NO	61	NO	NO
13	66	NO	NO	61	NO	NO
14	65	NO	NO	60	NO	NO
15	62	NO	NO	62	NO	NO
16	62	NO	NO	62	NO	NO
17	70	NO	NO	64	NO	NO
18	63	NO	NO	63	NO	NO
19	66	NO	NO	62	NO	NO
20	67	NO	NO	60	NO	NO
21	62	NO	NO	63	NO	NO
22	61	NO	NO	62	NO	NO
23	61	NO	NO	63	NO	NO
24	61	NO	NO	61	NO	NO
25	65	NO	NO	62	NO	NO
26	65	NO	NO	60	NO	NO
27	67	NO	NO	64	NO	NO
28	62	NO	NO	62	NO	NO
29	61	NO	NO	60	NO	NO
30	63	NO	NO	61	NO	NO

Table 60: Outlier Analysis of Improved and Simplified version AFSA, *SCP D*.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	73	NO	NO	69	NO	NO
2	73	NO	NO	71	NO	NO
3	69	NO	NO	72	NO	NO
4	71	NO	NO	72	NO	NO
5	68	NO	NO	72	NO	NO
6	69	NO	NO	68	NO	NO
7	72	NO	NO	69	NO	NO
8	74	NO	NO	69	NO	NO
9	71	NO	NO	70	NO	NO
10	69	NO	NO	72	NO	NO
11	72	NO	NO	71	NO	NO
12	71	NO	NO	71	NO	NO
13	71	NO	NO	71	NO	NO
14	71	NO	NO	71	NO	NO
15	73	NO	NO	73	NO	NO
16	69	NO	NO	67	NO	NO
17	69	NO	NO	70	NO	NO
18	73	NO	NO	70	NO	NO
19	71	NO	NO	69	NO	NO
20	71	NO	NO	70	NO	NO
21	72	NO	NO	69	NO	NO
22	67	NO	NO	70	NO	NO
23	72	NO	NO	72	NO	NO
24	69	NO	NO	69	NO	NO
25	79	NO	NO	70	NO	NO
26	75	NO	NO	69	NO	NO
27	70	NO	NO	69	NO	NO
28	76	NO	NO	69	NO	NO
29	71	NO	NO	71	NO	NO
30	74	NO	NO	69	NO	NO

Table 61: Outlier Analysis of Improved and Simplified version AFSA, *SCP D*.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	76	NO	NO	78	NO	NO
2	78	NO	NO	77	NO	NO
3	78	NO	NO	77	NO	NO
4	76	NO	NO	76	NO	NO
5	79	NO	NO	79	NO	NO
6	79	NO	NO	80	NO	NO
7	76	NO	NO	76	NO	NO
8	76	NO	NO	78	NO	NO
9	76	NO	NO	79	NO	NO
10	86	NO	NO	79	NO	NO
11	80	NO	NO	80	NO	NO
12	82	NO	NO	77	NO	NO
13	77	NO	NO	77	NO	NO
14	76	NO	NO	76	NO	NO
15	79	NO	NO	79	NO	NO
16	76	NO	NO	78	NO	NO
17	78	NO	NO	76	NO	NO
18	76	NO	NO	79	NO	NO
19	80	NO	NO	76	NO	NO
20	80	NO	NO	75	NO	NO
21	76	NO	NO	76	NO	NO
22	83	NO	NO	77	NO	NO
23	78	NO	NO	76	NO	NO
24	77	NO	NO	79	NO	NO
25	77	NO	NO	79	NO	NO
26	77	NO	NO	79	NO	NO
27	78	NO	NO	77	NO	NO
28	77	NO	NO	77	NO	NO
29	81	NO	NO	79	NO	NO
30	81	NO	NO	78	NO	NO

Table 62: Outlier Analysis of Improved and Simplified version AFSA, *SCP D*.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	63	NO	NO	63	NO	NO
2	68	NO	NO	66	NO	NO
3	64	NO	NO	64	NO	NO
4	65	NO	NO	67	NO	NO
5	65	NO	NO	66	NO	NO
6	65	NO	NO	65	NO	NO
7	65	NO	NO	65	NO	NO
8	67	NO	NO	66	NO	NO
9	63	NO	NO	65	NO	NO
10	64	NO	NO	65	NO	NO
11	65	NO	NO	66	NO	NO
12	64	NO	NO	65	NO	NO
13	65	NO	NO	65	NO	NO
14	69	NO	NO	66	NO	NO
15	66	NO	NO	64	NO	NO
16	63	NO	NO	64	NO	NO
17	69	NO	NO	65	NO	NO
18	74	YES	YES	66	NO	NO
19	67	NO	NO	66	NO	NO
20	70	NO	NO	64	NO	NO
21	65	NO	NO	65	NO	NO
22	64	NO	NO	66	NO	NO
23	66	NO	NO	64	NO	NO
24	65	NO	NO	63	NO	NO
25	65	NO	NO	64	NO	NO
26	65	NO	NO	66	NO	NO
27	67	NO	NO	67	NO	NO
28	67	NO	NO	63	NO	NO
29	66	NO	NO	65	NO	NO
30	67	NO	NO	65	NO	NO

Table 63: Outlier Analysis of Improved and Simplified version AFSA, *SCP D*.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	62	YES	NO	65	NO	NO
2	66	NO	NO	63	NO	NO
3	62	YES	NO	64	NO	NO
4	64	NO	NO	63	NO	NO
5	64	NO	NO	63	NO	NO
6	63	NO	NO	63	NO	NO
7	64	NO	NO	63	NO	NO
8	71	YES	YES	66	YES	NO
9	73	YES	YES	64	NO	NO
10	64	NO	NO	62	NO	NO
11	64	NO	NO	61	YES	NO
12	67	NO	NO	63	NO	NO
13	65	NO	NO	66	YES	NO
14	63	NO	NO	62	NO	NO
15	66	NO	NO	63	NO	NO
16	64	NO	NO	64	NO	NO
17	65	NO	NO	65	NO	NO
18	65	NO	NO	64	NO	NO
19	65	NO	NO	64	NO	NO
20	64	NO	NO	64	NO	NO
21	63	NO	NO	63	NO	NO
22	63	NO	NO	64	NO	NO
23	65	NO	NO	63	NO	NO
24	64	NO	NO	64	NO	NO
25	68	YES	NO	64	NO	NO
26	64	NO	NO	66	YES	NO
27	68	YES	NO	64	NO	NO
28	64	NO	NO	64	NO	NO
29	64	NO	NO	64	NO	NO
30	64	NO	NO	64	NO	NO

Table 64: Outlier Analysis of Improved and Simplified version AFSA, *SCP E*.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	6	NO	NO	6	NO	NO
2	5	YES	YES	6	NO	NO
3	6	NO	NO	5	YES	YES
4	6	NO	NO	6	NO	NO
5	6	NO	NO	6	NO	NO
6	6	NO	NO	6	NO	NO
7	6	NO	NO	6	NO	NO
8	6	NO	NO	6	NO	NO
9	6	NO	NO	6	NO	NO
10	6	NO	NO	6	NO	NO
11	6	NO	NO	6	NO	NO
12	6	NO	NO	6	NO	NO
13	6	NO	NO	6	NO	NO
14	6	NO	NO	6	NO	NO
15	5	YES	YES	6	NO	NO
16	6	NO	NO	6	NO	NO
17	6	NO	NO	6	NO	NO
18	6	NO	NO	6	NO	NO
19	6	NO	NO	6	NO	NO
20	6	NO	NO	6	NO	NO
21	6	NO	NO	5	YES	YES
22	6	NO	NO	6	NO	NO
23	6	NO	NO	6	NO	NO
24	6	NO	NO	6	NO	NO
25	6	NO	NO	6	NO	NO
26	6	NO	NO	6	NO	NO
27	6	NO	NO	6	NO	NO
28	5	YES	YES	6	NO	NO
29	5	YES	YES	6	NO	NO
30	6	NO	NO	6	NO	NO

Table 65: Outlier Analysis of Improved and Simplified version AFSA, *SCP E*.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	5	NO	NO	5	NO	NO
2	5	NO	NO	5	NO	NO
3	5	NO	NO	5	NO	NO
4	5	NO	NO	5	NO	NO
5	5	NO	NO	5	NO	NO
6	5	NO	NO	5	NO	NO
7	5	NO	NO	5	NO	NO
8	5	NO	NO	5	NO	NO
9	6	YES	YES	5	NO	NO
10	5	NO	NO	5	NO	NO
11	6	YES	YES	6	YES	YES
12	6	YES	YES	5	NO	NO
13	5	NO	NO	6	YES	YES
14	5	NO	NO	6	YES	YES
15	5	NO	NO	5	NO	NO
16	5	NO	NO	5	NO	NO
17	5	NO	NO	5	NO	NO
18	6	YES	YES	5	NO	NO
19	5	NO	NO	5	NO	NO
20	5	NO	NO	5	NO	NO
21	6	YES	YES	5	NO	NO
22	5	NO	NO	5	NO	NO
23	5	NO	NO	5	NO	NO
24	5	NO	NO	6	YES	YES
25	5	NO	NO	5	NO	NO
26	5	NO	NO	6	YES	YES
27	5	NO	NO	5	NO	NO
28	6	YES	YES	5	NO	NO
29	5	NO	NO	5	NO	NO
30	5	NO	NO	5	NO	NO

Table 66: Outlier Analysis of Improved and Simplified version AFSA, *SCP E*.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	5	NO	NO	5	NO	NO
2	5	NO	NO	5	NO	NO
3	6	YES	YES	5	NO	NO
4	5	NO	NO	5	NO	NO
5	5	NO	NO	5	NO	NO
6	5	NO	NO	5	NO	NO
7	5	NO	NO	5	NO	NO
8	6	YES	YES	5	NO	NO
9	5	NO	NO	6	YES	YES
10	5	NO	NO	5	NO	NO
11	5	NO	NO	5	NO	NO
12	5	NO	NO	5	NO	NO
13	5	NO	NO	5	NO	NO
14	6	YES	YES	5	NO	NO
15	5	NO	NO	5	NO	NO
16	6	YES	YES	5	NO	NO
17	5	NO	NO	5	NO	NO
18	5	NO	NO	5	NO	NO
19	5	NO	NO	5	NO	NO
20	5	NO	NO	5	NO	NO
21	5	NO	NO	5	NO	NO
22	5	NO	NO	5	NO	NO
23	5	NO	NO	5	NO	NO
24	5	NO	NO	5	NO	NO
25	5	NO	NO	5	NO	NO
26	5	NO	NO	5	NO	NO
27	5	NO	NO	5	NO	NO
28	5	NO	NO	5	NO	NO
29	5	NO	NO	5	NO	NO
30	5	NO	NO	5	NO	NO

Table 67: Outlier Analysis of Improved and Simplified version AFSA, *SCP E*.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	5	NO	NO	5	NO	NO
2	5	NO	NO	6	NO	NO
3	5	NO	NO	5	NO	NO
4	5	NO	NO	5	NO	NO
5	5	NO	NO	5	NO	NO
6	5	NO	NO	5	NO	NO
7	6	NO	NO	5	NO	NO
8	6	NO	NO	6	NO	NO
9	5	NO	NO	5	NO	NO
10	6	NO	NO	5	NO	NO
11	5	NO	NO	5	NO	NO
12	5	NO	NO	5	NO	NO
13	6	NO	NO	6	NO	NO
14	6	NO	NO	6	NO	NO
15	5	NO	NO	5	NO	NO
16	5	NO	NO	6	NO	NO
17	6	NO	NO	5	NO	NO
18	5	NO	NO	5	NO	NO
19	6	NO	NO	5	NO	NO
20	6	NO	NO	5	NO	NO
21	6	NO	NO	6	NO	NO
22	6	NO	NO	5	NO	NO
23	6	NO	NO	5	NO	NO
24	6	NO	NO	5	NO	NO
25	6	NO	NO	6	NO	NO
26	6	NO	NO	5	NO	NO
27	5	NO	NO	6	NO	NO
28	6	NO	NO	6	NO	NO
29	5	NO	NO	6	NO	NO
30	6	NO	NO	6	NO	NO

Table 68: Outlier Analysis of Improved and Simplified version AFSA, *SCP E*.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	6	NO	NO	5	NO	NO
2	5	NO	NO	6	YES	YES
3	6	NO	NO	5	NO	NO
4	6	NO	NO	5	NO	NO
5	6	NO	NO	5	NO	NO
6	6	NO	NO	6	YES	YES
7	6	NO	NO	5	NO	NO
8	5	NO	NO	5	NO	NO
9	5	NO	NO	5	NO	NO
10	6	NO	NO	5	NO	NO
11	6	NO	NO	5	NO	NO
12	6	NO	NO	5	NO	NO
13	6	NO	NO	5	NO	NO
14	6	NO	NO	5	NO	NO
15	6	NO	NO	5	NO	NO
16	6	NO	NO	5	NO	NO
17	5	NO	NO	5	NO	NO
18	5	NO	NO	5	NO	NO
19	6	NO	NO	5	NO	NO
20	5	NO	NO	5	NO	NO
21	6	NO	NO	5	NO	NO
22	6	NO	NO	6	YES	YES
23	6	NO	NO	5	NO	NO
24	5	NO	NO	6	YES	YES
25	6	NO	NO	6	YES	YES
26	6	NO	NO	5	NO	NO
27	6	NO	NO	6	YES	YES
28	6	NO	NO	5	NO	NO
29	5	NO	NO	5	NO	NO
30	6	NO	NO	5	NO	NO

Table 69: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRE*.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	30	NO	NO	30	NO	NO
2	30	NO	NO	30	NO	NO
3	31	NO	NO	29	NO	NO
4	29	NO	NO	30	NO	NO
5	31	NO	NO	29	NO	NO
6	34	NO	NO	30	NO	NO
7	31	NO	NO	29	NO	NO
8	30	NO	NO	30	NO	NO
9	35	NO	NO	30	NO	NO
10	31	NO	NO	30	NO	NO
11	30	NO	NO	29	NO	NO
12	29	NO	NO	29	NO	NO
13	33	NO	NO	30	NO	NO
14	35	NO	NO	29	NO	NO
15	31	NO	NO	29	NO	NO
16	31	NO	NO	29	NO	NO
17	33	NO	NO	30	NO	NO
18	29	NO	NO	29	NO	NO
19	31	NO	NO	29	NO	NO
20	30	NO	NO	30	NO	NO
21	33	NO	NO	29	NO	NO
22	31	NO	NO	29	NO	NO
23	31	NO	NO	30	NO	NO
24	31	NO	NO	29	NO	NO
25	29	NO	NO	29	NO	NO
26	30	NO	NO	30	NO	NO
27	31	NO	NO	30	NO	NO
28	29	NO	NO	29	NO	NO
29	33	NO	NO	30	NO	NO
30	34	NO	NO	29	NO	NO

Table 70: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRE*.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	36	YES	NO	33	NO	NO
2	32	NO	NO	32	NO	NO
3	33	NO	NO	33	NO	NO
4	33	NO	NO	34	NO	NO
5	33	NO	NO	32	NO	NO
6	33	NO	NO	32	NO	NO
7	34	NO	NO	33	NO	NO
8	37	YES	NO	32	NO	NO
9	35	NO	NO	34	NO	NO
10	34	NO	NO	32	NO	NO
11	34	NO	NO	32	NO	NO
12	34	NO	NO	32	NO	NO
13	34	NO	NO	33	NO	NO
14	33	NO	NO	34	NO	NO
15	35	NO	NO	32	NO	NO
16	35	NO	NO	32	NO	NO
17	34	NO	NO	32	NO	NO
18	31	YES	NO	32	NO	NO
19	32	NO	NO	33	NO	NO
20	34	NO	NO	34	NO	NO
21	34	NO	NO	32	NO	NO
22	34	NO	NO	32	NO	NO
23	33	NO	NO	33	NO	NO
24	33	NO	NO	33	NO	NO
25	32	NO	NO	31	NO	NO
26	33	NO	NO	34	NO	NO
27	33	NO	NO	31	NO	NO
28	35	NO	NO	32	NO	NO
29	32	NO	NO	32	NO	NO
30	32	NO	NO	33	NO	NO

Table 71: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRE*.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	29	NO	NO	30	NO	NO
2	32	NO	NO	32	NO	NO
3	30	NO	NO	28	NO	NO
4	30	NO	NO	29	NO	NO
5	33	NO	NO	29	NO	NO
6	30	NO	NO	31	NO	NO
7	29	NO	NO	29	NO	NO
8	30	NO	NO	31	NO	NO
9	32	NO	NO	31	NO	NO
10	30	NO	NO	29	NO	NO
11	31	NO	NO	31	NO	NO
12	31	NO	NO	29	NO	NO
13	29	NO	NO	31	NO	NO
14	30	NO	NO	30	NO	NO
15	30	NO	NO	31	NO	NO
16	29	NO	NO	31	NO	NO
17	33	NO	NO	30	NO	NO
18	31	NO	NO	30	NO	NO
19	33	NO	NO	28	NO	NO
20	35	NO	NO	31	NO	NO
21	32	NO	NO	30	NO	NO
22	30	NO	NO	30	NO	NO
23	30	NO	NO	31	NO	NO
24	30	NO	NO	29	NO	NO
25	29	NO	NO	30	NO	NO
26	32	NO	NO	30	NO	NO
27	32	NO	NO	30	NO	NO
28	29	NO	NO	30	NO	NO
29	34	NO	NO	30	NO	NO
30	30	NO	NO	31	NO	NO

Table 72: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRE*.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	33	NO	NO	29	NO	NO
2	31	NO	NO	29	NO	NO
3	31	NO	NO	29	NO	NO
4	31	NO	NO	33	NO	NO
5	34	NO	NO	29	NO	NO
6	30	NO	NO	30	NO	NO
7	34	NO	NO	30	NO	NO
8	33	NO	NO	32	NO	NO
9	36	NO	NO	29	NO	NO
10	30	NO	NO	31	NO	NO
11	32	NO	NO	31	NO	NO
12	33	NO	NO	30	NO	NO
13	32	NO	NO	32	NO	NO
14	33	NO	NO	32	NO	NO
15	34	NO	NO	30	NO	NO
16	32	NO	NO	31	NO	NO
17	30	NO	NO	31	NO	NO
18	33	NO	NO	32	NO	NO
19	32	NO	NO	33	NO	NO
20	32	NO	NO	32	NO	NO
21	35	NO	NO	29	NO	NO
22	36	NO	NO	32	NO	NO
23	31	NO	NO	33	NO	NO
24	31	NO	NO	30	NO	NO
25	31	NO	NO	32	NO	NO
26	31	NO	NO	30	NO	NO
27	32	NO	NO	29	NO	NO
28	34	NO	NO	30	NO	NO
29	31	NO	NO	31	NO	NO
30	33	NO	NO	30	NO	NO

Table 73: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRE*.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	32	NO	NO	29	NO	NO
2	32	NO	NO	29	NO	NO
3	30	NO	NO	29	NO	NO
4	31	NO	NO	30	NO	NO
5	31	NO	NO	29	NO	NO
6	31	NO	NO	30	NO	NO
7	35	NO	NO	29	NO	NO
8	40	YES	YES	31	NO	NO
9	32	NO	NO	32	YES	NO
10	31	NO	NO	30	NO	NO
11	29	NO	NO	29	NO	NO
12	29	NO	NO	30	NO	NO
13	28	NO	NO	31	NO	NO
14	30	NO	NO	29	NO	NO
15	29	NO	NO	29	NO	NO
16	31	NO	NO	30	NO	NO
17	30	NO	NO	29	NO	NO
18	30	NO	NO	30	NO	NO
19	32	NO	NO	29	NO	NO
20	30	NO	NO	29	NO	NO
21	35	NO	NO	29	NO	NO
22	30	NO	NO	29	NO	NO
23	30	NO	NO	29	NO	NO
24	30	NO	NO	30	NO	NO
25	31	NO	NO	30	NO	NO
26	29	NO	NO	30	NO	NO
27	30	NO	NO	30	NO	NO
28	30	NO	NO	29	NO	NO
29	32	NO	NO	29	NO	NO
30	32	NO	NO	30	NO	NO

Table 74: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRF*.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	17	NO	NO	16	NO	NO
2	15	NO	NO	16	NO	NO
3	15	NO	NO	15	YES	NO
4	16	NO	NO	15	YES	NO
5	16	NO	NO	16	NO	NO
6	15	NO	NO	16	NO	NO
7	16	NO	NO	15	YES	NO
8	16	NO	NO	16	NO	NO
9	16	NO	NO	16	NO	NO
10	16	NO	NO	16	NO	NO
11	15	NO	NO	17	YES	YES
12	16	NO	NO	16	NO	NO
13	15	NO	NO	16	NO	NO
14	16	NO	NO	16	NO	NO
15	17	NO	NO	16	NO	NO
16	17	NO	NO	16	NO	NO
17	16	NO	NO	15	YES	NO
18	16	NO	NO	16	NO	NO
19	16	NO	NO	15	YES	NO
20	15	NO	NO	16	NO	NO
21	15	NO	NO	16	NO	NO
22	15	NO	NO	15	YES	NO
23	15	NO	NO	17	YES	YES
24	16	NO	NO	16	NO	NO
25	16	NO	NO	16	NO	NO
26	16	NO	NO	16	NO	NO
27	16	NO	NO	15	YES	NO
28	16	NO	NO	16	NO	NO
29	16	NO	NO	16	NO	NO
30	16	NO	NO	16	NO	NO

Table 75: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRF*.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	16	NO	NO	16	NO	NO
2	16	NO	NO	16	NO	NO
3	17	NO	NO	16	NO	NO
4	15	NO	NO	18	NO	NO
5	17	NO	NO	16	NO	NO
6	19	YES	NO	16	NO	NO
7	16	NO	NO	16	NO	NO
8	18	NO	NO	16	NO	NO
9	18	NO	NO	16	NO	NO
10	16	NO	NO	16	NO	NO
11	17	NO	NO	16	NO	NO
12	16	NO	NO	16	NO	NO
13	17	NO	NO	16	NO	NO
14	16	NO	NO	17	NO	NO
15	18	NO	NO	17	NO	NO
16	17	NO	NO	17	NO	NO
17	17	NO	NO	17	NO	NO
18	16	NO	NO	16	NO	NO
19	16	NO	NO	17	NO	NO
20	16	NO	NO	16	NO	NO
21	16	NO	NO	16	NO	NO
22	17	NO	NO	17	NO	NO
23	16	NO	NO	16	NO	NO
24	16	NO	NO	17	NO	NO
25	17	NO	NO	17	NO	NO
26	17	NO	NO	16	NO	NO
27	17	NO	NO	17	NO	NO
28	17	NO	NO	16	NO	NO
29	16	NO	NO	16	NO	NO
30	17	NO	NO	17	NO	NO

Table 76: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRF*.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	16	NO	NO	17	NO	NO
2	16	NO	NO	17	NO	NO
3	17	NO	NO	16	NO	NO
4	18	NO	NO	17	NO	NO
5	19	NO	NO	16	NO	NO
6	18	NO	NO	16	NO	NO
7	17	NO	NO	16	NO	NO
8	17	NO	NO	16	NO	NO
9	17	NO	NO	16	NO	NO
10	17	NO	NO	16	NO	NO
11	17	NO	NO	16	NO	NO
12	17	NO	NO	16	NO	NO
13	17	NO	NO	17	NO	NO
14	17	NO	NO	16	NO	NO
15	16	NO	NO	17	NO	NO
16	18	NO	NO	17	NO	NO
17	18	NO	NO	17	NO	NO
18	18	NO	NO	17	NO	NO
19	16	NO	NO	16	NO	NO
20	17	NO	NO	16	NO	NO
21	18	NO	NO	17	NO	NO
22	16	NO	NO	17	NO	NO
23	21	YES	NO	16	NO	NO
24	18	NO	NO	16	NO	NO
25	18	NO	NO	16	NO	NO
26	18	NO	NO	16	NO	NO
27	17	NO	NO	17	NO	NO
28	17	NO	NO	16	NO	NO
29	17	NO	NO	17	NO	NO
30	17	NO	NO	18	NO	NO

Table 77: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRF*.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	16	NO	NO	16	NO	NO
2	17	YES	YES	16	NO	NO
3	15	YES	YES	16	NO	NO
4	17	YES	YES	16	NO	NO
5	15	YES	YES	16	NO	NO
6	16	NO	NO	15	YES	NO
7	15	YES	YES	15	YES	NO
8	16	NO	NO	16	NO	NO
9	17	YES	YES	16	NO	NO
10	17	YES	YES	16	NO	NO
11	15	YES	YES	16	NO	NO
12	16	NO	NO	15	YES	NO
13	16	NO	NO	16	NO	NO
14	16	NO	NO	17	YES	YES
15	16	NO	NO	15	YES	NO
16	16	NO	NO	16	NO	NO
17	16	NO	NO	16	NO	NO
18	15	YES	YES	15	YES	NO
19	16	NO	NO	16	NO	NO
20	16	NO	NO	16	NO	NO
21	17	YES	YES	15	YES	NO
22	15	YES	YES	16	NO	NO
23	17	YES	YES	16	NO	NO
24	16	NO	NO	16	NO	NO
25	16	NO	NO	16	NO	NO
26	16	NO	NO	17	YES	YES
27	16	NO	NO	15	YES	NO
28	16	NO	NO	16	NO	NO
29	16	NO	NO	16	NO	NO
30	16	NO	NO	16	NO	NO

Table 78: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRF*.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	14	NO	NO	15	NO	NO
2	16	NO	NO	16	NO	NO
3	14	NO	NO	16	NO	NO
4	16	NO	NO	16	NO	NO
5	16	NO	NO	16	NO	NO
6	15	NO	NO	15	NO	NO
7	14	NO	NO	15	NO	NO
8	15	NO	NO	15	NO	NO
9	15	NO	NO	16	NO	NO
10	16	NO	NO	16	NO	NO
11	15	NO	NO	15	NO	NO
12	15	NO	NO	15	NO	NO
13	16	NO	NO	16	NO	NO
14	16	NO	NO	16	NO	NO
15	16	NO	NO	15	NO	NO
16	16	NO	NO	16	NO	NO
17	16	NO	NO	16	NO	NO
18	17	NO	NO	16	NO	NO
19	16	NO	NO	16	NO	NO
20	16	NO	NO	15	NO	NO
21	17	NO	NO	16	NO	NO
22	16	NO	NO	16	NO	NO
23	15	NO	NO	15	NO	NO
24	15	NO	NO	16	NO	NO
25	15	NO	NO	16	NO	NO
26	15	NO	NO	15	NO	NO
27	16	NO	NO	16	NO	NO
28	16	NO	NO	15	NO	NO
29	15	NO	NO	16	NO	NO
30	16	NO	NO	16	NO	NO

Table 79: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRG*.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	203	NO	NO	187	NO	NO
2	193	NO	NO	186	NO	NO
3	195	NO	NO	184	NO	NO
4	187	NO	NO	190	NO	NO
5	194	NO	NO	190	NO	NO
6	191	NO	NO	188	NO	NO
7	195	NO	NO	190	NO	NO
8	198	NO	NO	185	NO	NO
9	194	NO	NO	185	NO	NO
10	192	NO	NO	189	NO	NO
11	197	NO	NO	188	NO	NO
12	192	NO	NO	188	NO	NO
13	208	YES	NO	180	NO	NO
14	193	NO	NO	184	NO	NO
15	191	NO	NO	186	NO	NO
16	194	NO	NO	185	NO	NO
17	200	NO	NO	188	NO	NO
18	194	NO	NO	186	NO	NO
19	206	NO	NO	187	NO	NO
20	195	NO	NO	185	NO	NO
21	203	NO	NO	187	NO	NO
22	197	NO	NO	187	NO	NO
23	194	NO	NO	193	NO	NO
24	192	NO	NO	195	NO	NO
25	206	NO	NO	189	NO	NO
26	198	NO	NO	187	NO	NO
27	198	NO	NO	189	NO	NO
28	199	NO	NO	189	NO	NO
29	198	NO	NO	191	NO	NO
30	194	NO	NO	184	NO	NO

Table 80: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRG*.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	166	NO	NO	166	NO	NO
2	176	NO	NO	166	NO	NO
3	163	NO	NO	165	NO	NO
4	174	NO	NO	164	NO	NO
5	169	NO	NO	165	NO	NO
6	170	NO	NO	161	NO	NO
7	164	NO	NO	163	NO	NO
8	165	NO	NO	168	NO	NO
9	163	NO	NO	165	NO	NO
10	175	NO	NO	165	NO	NO
11	167	NO	NO	161	NO	NO
12	168	NO	NO	163	NO	NO
13	171	NO	NO	164	NO	NO
14	166	NO	NO	163	NO	NO
15	166	NO	NO	164	NO	NO
16	171	NO	NO	164	NO	NO
17	170	NO	NO	164	NO	NO
18	166	NO	NO	166	NO	NO
19	171	NO	NO	163	NO	NO
20	170	NO	NO	163	NO	NO
21	162	NO	NO	163	NO	NO
22	167	NO	NO	165	NO	NO
23	167	NO	NO	165	NO	NO
24	172	NO	NO	165	NO	NO
25	178	NO	NO	166	NO	NO
26	166	NO	NO	163	NO	NO
27	161	NO	NO	161	NO	NO
28	170	NO	NO	164	NO	NO
29	169	NO	NO	167	NO	NO
30	164	NO	NO	162	NO	NO

Table 81: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRG*.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	176	NO	NO	177	NO	NO
2	181	NO	NO	180	NO	NO
3	181	NO	NO	177	NO	NO
4	180	NO	NO	180	NO	NO
5	178	NO	NO	175	NO	NO
6	182	NO	NO	178	NO	NO
7	179	NO	NO	176	NO	NO
8	186	NO	NO	175	NO	NO
9	182	NO	NO	177	NO	NO
10	175	NO	NO	178	NO	NO
11	179	NO	NO	173	NO	NO
12	182	NO	NO	176	NO	NO
13	182	NO	NO	177	NO	NO
14	181	NO	NO	175	NO	NO
15	174	NO	NO	176	NO	NO
16	179	NO	NO	173	NO	NO
17	181	NO	NO	178	NO	NO
18	178	NO	NO	173	NO	NO
19	178	NO	NO	175	NO	NO
20	180	NO	NO	179	NO	NO
21	176	NO	NO	175	NO	NO
22	184	NO	NO	174	NO	NO
23	176	NO	NO	171	NO	NO
24	181	NO	NO	177	NO	NO
25	183	NO	NO	178	NO	NO
26	181	NO	NO	174	NO	NO
27	180	NO	NO	174	NO	NO
28	182	NO	NO	178	NO	NO
29	179	NO	NO	179	NO	NO
30	179	NO	NO	175	NO	NO

Table 82: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRG*.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	185	NO	NO	177	NO	NO
2	178	NO	NO	178	NO	NO
3	183	NO	NO	177	NO	NO
4	179	NO	NO	174	NO	NO
5	180	NO	NO	177	NO	NO
6	180	NO	NO	175	NO	NO
7	183	NO	NO	171	NO	NO
8	184	NO	NO	179	NO	NO
9	177	NO	NO	177	NO	NO
10	193	NO	NO	179	NO	NO
11	180	NO	NO	177	NO	NO
12	195	YES	NO	177	NO	NO
13	188	NO	NO	183	NO	NO
14	186	NO	NO	177	NO	NO
15	175	NO	NO	175	NO	NO
16	184	NO	NO	181	NO	NO
17	183	NO	NO	178	NO	NO
18	183	NO	NO	182	NO	NO
19	190	NO	NO	180	NO	NO
20	185	NO	NO	175	NO	NO
21	185	NO	NO	179	NO	NO
22	183	NO	NO	182	NO	NO
23	180	NO	NO	176	NO	NO
24	179	NO	NO	176	NO	NO
25	178	NO	NO	177	NO	NO
26	183	NO	NO	175	NO	NO
27	177	NO	NO	181	NO	NO
28	178	NO	NO	182	NO	NO
29	189	NO	NO	177	NO	NO
30	179	NO	NO	182	NO	NO

Table 83: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRG*.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	179	NO	NO	180	NO	NO
2	178	NO	NO	180	NO	NO
3	177	NO	NO	175	NO	NO
4	192	NO	NO	179	NO	NO
5	185	NO	NO	178	NO	NO
6	180	NO	NO	184	NO	NO
7	188	NO	NO	181	NO	NO
8	189	NO	NO	175	NO	NO
9	182	NO	NO	184	NO	NO
10	183	NO	NO	184	NO	NO
11	195	NO	NO	181	NO	NO
12	187	NO	NO	177	NO	NO
13	183	NO	NO	178	NO	NO
14	182	NO	NO	179	NO	NO
15	188	NO	NO	182	NO	NO
16	183	NO	NO	179	NO	NO
17	192	NO	NO	177	NO	NO
18	184	NO	NO	181	NO	NO
19	179	NO	NO	179	NO	NO
20	179	NO	NO	181	NO	NO
21	183	NO	NO	182	NO	NO
22	184	NO	NO	174	NO	NO
23	185	NO	NO	179	NO	NO
24	187	NO	NO	179	NO	NO
25	181	NO	NO	185	NO	NO
26	184	NO	NO	182	NO	NO
27	187	NO	NO	179	NO	NO
28	183	NO	NO	179	NO	NO
29	184	NO	NO	181	NO	NO
30	191	NO	NO	178	NO	NO

Table 84: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRH*.1 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	76	NO	NO	70	NO	NO
2	72	NO	NO	69	NO	NO
3	74	NO	NO	69	NO	NO
4	82	NO	NO	67	NO	NO
5	72	NO	NO	69	NO	NO
6	74	NO	NO	68	NO	NO
7	79	NO	NO	68	NO	NO
8	75	NO	NO	67	NO	NO
9	73	NO	NO	70	NO	NO
10	77	NO	NO	68	NO	NO
11	75	NO	NO	70	NO	NO
12	78	NO	NO	68	NO	NO
13	70	NO	NO	70	NO	NO
14	77	NO	NO	69	NO	NO
15	76	NO	NO	67	NO	NO
16	75	NO	NO	69	NO	NO
17	81	NO	NO	70	NO	NO
18	72	NO	NO	70	NO	NO
19	77	NO	NO	68	NO	NO
20	68	NO	NO	68	NO	NO
21	74	NO	NO	69	NO	NO
22	71	NO	NO	68	NO	NO
23	71	NO	NO	70	NO	NO
24	74	NO	NO	67	NO	NO
25	73	NO	NO	70	NO	NO
26	72	NO	NO	67	NO	NO
27	80	NO	NO	67	NO	NO
28	72	NO	NO	69	NO	NO
29	77	NO	NO	68	NO	NO
30	71	NO	NO	67	NO	NO

Table 85: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRH*.2 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	74	NO	NO	69	NO	NO
2	80	NO	NO	70	NO	NO
3	72	NO	NO	66	NO	NO
4	71	NO	NO	69	NO	NO
5	71	NO	NO	67	NO	NO
6	77	NO	NO	66	NO	NO
7	76	NO	NO	69	NO	NO
8	73	NO	NO	70	NO	NO
9	74	NO	NO	71	NO	NO
10	79	NO	NO	69	NO	NO
11	85	NO	NO	69	NO	NO
12	76	NO	NO	67	NO	NO
13	74	NO	NO	70	NO	NO
14	76	NO	NO	69	NO	NO
15	77	NO	NO	67	NO	NO
16	73	NO	NO	68	NO	NO
17	77	NO	NO	72	NO	NO
18	70	NO	NO	66	NO	NO
19	70	NO	NO	69	NO	NO
20	69	NO	NO	74	NO	NO
21	75	NO	NO	67	NO	NO
22	68	NO	NO	69	NO	NO
23	79	NO	NO	67	NO	NO
24	71	NO	NO	66	NO	NO
25	70	NO	NO	70	NO	NO
26	77	NO	NO	69	NO	NO
27	77	NO	NO	71	NO	NO
28	75	NO	NO	69	NO	NO
29	72	NO	NO	68	NO	NO
30	69	NO	NO	67	NO	NO

Table 86: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRH*.3 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	77	NO	NO	66	NO	NO
2	70	NO	NO	63	NO	NO
3	69	NO	NO	65	NO	NO
4	70	NO	NO	68	NO	NO
5	70	NO	NO	67	NO	NO
6	72	NO	NO	63	NO	NO
7	72	NO	NO	64	NO	NO
8	73	NO	NO	65	NO	NO
9	77	NO	NO	67	NO	NO
10	66	NO	NO	68	NO	NO
11	72	NO	NO	68	NO	NO
12	70	NO	NO	67	NO	NO
13	72	NO	NO	67	NO	NO
14	70	NO	NO	68	NO	NO
15	75	NO	NO	66	NO	NO
16	75	NO	NO	66	NO	NO
17	71	NO	NO	64	NO	NO
18	71	NO	NO	66	NO	NO
19	76	NO	NO	67	NO	NO
20	77	NO	NO	67	NO	NO
21	72	NO	NO	69	NO	NO
22	70	NO	NO	67	NO	NO
23	73	NO	NO	66	NO	NO
24	72	NO	NO	66	NO	NO
25	69	NO	NO	64	NO	NO
26	68	NO	NO	65	NO	NO
27	66	NO	NO	66	NO	NO
28	73	NO	NO	67	NO	NO
29	72	NO	NO	67	NO	NO
30	79	YES	NO	66	NO	NO

Table 87: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRH*.4 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	65	NO	NO	62	NO	NO
2	69	NO	NO	64	NO	NO
3	68	NO	NO	63	NO	NO
4	71	NO	NO	63	NO	NO
5	69	NO	NO	63	NO	NO
6	66	NO	NO	62	NO	NO
7	68	NO	NO	64	NO	NO
8	76	NO	NO	66	NO	NO
9	68	NO	NO	64	NO	NO
10	68	NO	NO	65	NO	NO
11	66	NO	NO	63	NO	NO
12	67	NO	NO	63	NO	NO
13	64	NO	NO	63	NO	NO
14	69	NO	NO	64	NO	NO
15	70	NO	NO	63	NO	NO
16	72	NO	NO	64	NO	NO
17	63	NO	NO	65	NO	NO
18	70	NO	NO	64	NO	NO
19	67	NO	NO	65	NO	NO
20	66	NO	NO	63	NO	NO
21	71	NO	NO	65	NO	NO
22	67	NO	NO	65	NO	NO
23	65	NO	NO	65	NO	NO
24	72	NO	NO	63	NO	NO
25	70	NO	NO	61	YES	NO
26	65	NO	NO	62	NO	NO
27	66	NO	NO	63	NO	NO
28	75	NO	NO	64	NO	NO
29	68	NO	NO	64	NO	NO
30	69	NO	NO	63	NO	NO

Table 88: Outlier Analysis of Improved and Simplified version AFSA, *SCP NRH*.5 - Experiment 2

Iteration Number	*Improved AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*	*Simplified AFSA*	*Is it Mild Outlier ?*	*Is it Extreme Outlier ?*
1	69	NO	NO	60	NO	NO
2	65	NO	NO	60	NO	NO
3	69	NO	NO	61	NO	NO
4	62	NO	NO	60	NO	NO
5	69	NO	NO	60	NO	NO
6	63	NO	NO	59	NO	NO
7	61	NO	NO	60	NO	NO
8	68	NO	NO	60	NO	NO
9	63	NO	NO	60	NO	NO
10	64	NO	NO	57	YES	NO
11	66	NO	NO	60	NO	NO
12	66	NO	NO	59	NO	NO
13	63	NO	NO	60	NO	NO
14	67	NO	NO	59	NO	NO
15	69	NO	NO	60	NO	NO
16	68	NO	NO	60	NO	NO
17	65	NO	NO	58	NO	NO
18	64	NO	NO	60	NO	NO
19	67	NO	NO	59	NO	NO
20	65	NO	NO	64	YES	YES
21	68	NO	NO	61	NO	NO
22	66	NO	NO	61	NO	NO
23	71	NO	NO	61	NO	NO
24	62	NO	NO	60	NO	NO
25	63	NO	NO	60	NO	NO
26	69	NO	NO	60	NO	NO
27	63	NO	NO	59	NO	NO
28	71	NO	NO	59	NO	NO
29	63	NO	NO	58	NO	NO
30	60	NO	NO	59	NO	NO

References

[1] El-Ghazali Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009.

[2] Francis J. Vasko, Floyd E. Wolf, and Kenneth L. Stott. Optimal selection of ingot sizes via set covering. *Operations Research*, 35(3):346–353, 1987.

[3] Warren Walker. Using the set-covering problem to assign fire companies to fire houses. *Operations Research*, 22(2):275–277, 1974.

[4] Broderick Crawford, Ricardo Soto, and Eric Monfroy. Cultural algorithms for the set covering problem. In *Advances in Swarm Intelligence*, pages 27–34. Springer, 2013.

[5] Broderick Crawford, Ricardo Soto, Claudio Torres-Rojas, Cristian Peña, Marco Riquelme-Leiva, Sanjay Misra, Franklin Johnson, and Fernando Paredes. A binary fruit fly optimization algorithm to solve the set covering problem. In *Computational Science and Its Applications–ICCSA 2015*, pages 411–420. Springer, 2015.

[6] Broderick Crawford, Ricardo Soto, Felipe Aballay, Sanjay Misra, Franklin Johnson, and Fernando Paredes. A teaching-learning-based optimization algorithm for solving set covering problems. In *Computational Science and Its Applications–ICCSA 2015*, pages 421–430. Springer, 2015.

[7] Victor Reyes, Ignacio Araya, Broderick Crawford, Ricardo Soto, and Eduardo Olguín. *A Beam-Search Approach to the Set Covering Problem*, pages 395–402. Springer International Publishing, Cham, 2016.

[8] Broderick Crawford, Ricardo Soto, Jorge Córdova, and Eduardo Olguín. *A Nature Inspired Intelligent Water Drop Algorithm and Its Application for Solving The Set Covering Problem*, pages 437–447. Springer International Publishing, Cham, 2016.

[9] Broderick Crawford, Ricardo Soto, Ismael Fuenzalida Legüe, and Eduardo Olguín. *A Binary Invasive Weed Optimization Algorithm for the Set Covering Problem*, pages 459–468. Springer International Publishing, Cham, 2016.

[10] Broderick Crawford, Ricardo Soto, Gonzalo Astudillo, and Eduardo Olguín. *Fireworks Explosion Can Solve the Set Covering Problem*, pages 477–490. Springer International Publishing, Cham, 2016.

[11] Broderick Crawford, Ricardo Soto, Gonzalo Astudillo, Eduardo Olguín, and Sanjay Misra. *Solving Set Covering Problem with Fireworks Explosion*, pages 273–283. Springer International Publishing, Cham, 2016.

[12] Broderick Crawford, Ricardo Soto, Luis Riquelme, and Eduardo Olguín. *Biogeography-Based Optimization Algorithm for Solving the Set Covering Problem*, pages 273–283. Springer International Publishing, Cham, 2016.

[13] Broderick Crawford, Ricardo Soto, Luis Riquelme, Eduardo Olguín, and Sanjay Misra. *Set Covering Problem Resolution by Biogeography-Based Optimization Algorithm*, pages 153–165. Springer International Publishing, Cham, 2016.

[14] Broderick Crawford, Ricardo Soto, Hugo Caballero, and Eduardo Olguín. *A Bi-Objetive Cat Swarm Optimization Algorithm for Set Covering Problem*, pages 491–500. Springer International Publishing, Cham, 2016.

[15] Broderick Crawford, Ricardo Soto, Hugo Caballero, Eduardo Olguín, and Sanjay Misra. *Solving Biobjective Set Covering Problem Using Binary Cat Swarm Optimization Algorithm*, pages 220–231. Springer International Publishing, Cham, 2016.

[16] Broderick Crawford, Ricardo Soto, Natalia Berrios, Eduardo Olguín, and Sanjay Misra. *Cat Swarm Optimization with Different Transfer Functions for Solving Set Covering Problems*, pages 228–240. Springer International Publishing, Cham, 2016.

[17] Broderick Crawford, Ricardo Soto, Natalia Berríos, and Eduardo Olguín. Cat swarm optimization with different binarization methods for solving set covering problems. In *Artificial Intelligence Perspectives in Intelligent Systems - Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Vol 1*, pages 511–524, 2016.

[18] Ricardo Soto, Broderick Crawford, Rodrigo Olivares, Jorge Barraza, Franklin Johnson, and Fernando Paredes. *A Binary Cuckoo Search Algorithm for Solving the Set Covering Problem*, pages 88–97. Springer International Publishing, Cham, 2015.

[19] Broderick Crawford, Ricardo Soto, Rodrigo Cuesta, Miguel Olivares-Suárez, Franklin Johnson, and Eduardo Olguín. Two swarm intelligence algorithms for the set covering problem. In *Proceedings of the 9th International Conference on Software Engineering and Applications (ICSOFT 2014)*, pages 60–69, 2014.

[20] Juan Salas, Broderick Crawford, Ricardo Soto, Álvaro Gómez Rubio, Adrián Jaramillo, Sebastián Mansilla Villablanca, and Eduardo Olguín. *Binary Harmony Search Algorithm for Solving Set-Covering Problem*, pages 917–930. Springer International Publishing, Cham, 2016.

[21] Adrián Jaramillo, Broderick Crawford, Ricardo Soto, Sebastián Mansilla Villablanca, Álvaro Gómez Rubio, Juan Salas, and Eduardo Olguín. *Solving the Set Covering Problem with the Soccer League Competition Algorithm*, pages 884–891. Springer International Publishing, Cham, 2016.

[22] Álvaro Gómez Rubio, Broderick Crawford, Ricardo Soto, Adrián Jaramillo, Sebastián Mansilla Villablanca, Juan Salas, and Eduardo Olguín. *An Binary Black Hole Algorithm to Solve Set Covering Problem*, pages 873–883. Springer International Publishing, Cham, 2016.

[23] Md. Abul Kalam Azad, Ana Maria A.C. Rocha, and Edite M.G.P. Fernandes. Solving multidimensional 0–1 knapsack problem with an artificial fish swarm algorithm. In *Computational Science and Its Applications–ICCSA 2012*, pages 72–86. Springer, 2012.

[24] Md. Abul Kalam Azad, Ana Maria A.C. Rocha, and Edite M.G.P. Fernandes. Improved binary artificial fish swarm algorithm for the 0–1 multidimensional knapsack problems. *Swarm and Evolutionary Computation*, 14:66–75, 2014.

[25] Md. Abul Kalam Azad, Ana Maria A.C. Rocha, and Edite M.G.P. Fernandes. Solving large 0–1 multidimensional knapsack problems by a new simplified binary artificial fish swarm algorithm. *Journal of Mathematical Modelling and Algorithms in Operations Research*, pages 1–18, 2015.

[26] John E. Beasley. *OR-Library*, June 1990; Last update: June 2016 (accessed July 15, 2016). `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`.

[27] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[28] Broderick Crawford, Ricardo Soto, Cristian Peña, Wenceslao Palma, Franklin Johnson, and Fernando Paredes. Solving the set covering problem with a shuffled frog leaping algorithm. In *Intelligent Information and Database Systems - 7th Asian Conference, ACIIDS 2015, Bali, Indonesia, March 23-25, 2015, Proceedings, Part II*, pages 41–50, 2015.

[29] John E. Beasley and Paul C. Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94(2):392–404, 1996.

[30] Rosa González Silos. Algorítmos heurísticos para el problema del set covering: Mejora mediante aleatorización. 2013.

[31] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs (3rd Ed.)*. Springer-Verlag, London, UK, UK, 1996.

[32] Zbigniew Michalewicz. *Genetic algorithms+ data structures= evolution programs*. Springer Science & Business Media, 2013.

[33] Marco Caserta. Tabu search-based metaheuristic algorithm for large-scale set covering problems. In KarlF. Doerner, Michel Gendreau, Peter Greistorfer, Walter Gutjahr, RichardF. Hartl, and Marc Reimann, editors, *Metaheuristics*, volume 39 of *Operations Research/Computer Science Interfaces Series*, pages 43–63. Springer US, 2007.

[34] Michael J. Brusco, Larry W. Jacobs, and Gary M. Thompson. A morphing procedure to supplement a simulated annealing heuristic for cost for cost-and coverage-correlated set-covering problems. *Annals of Operations Research*, 86(0):611–627, 1999.

[35] Rodrigo Cuesta, Broderick Crawford, Ricardo Soto, and Fernando Paredes. *Modern Trends and Techniques in Computer Science: 3rd Computer Science On-line Conference 2014 (CSOC 2014)*, chapter An Artificial Bee Colony Algorithm for the Set Covering Problem, pages 53–63. Springer International Publishing, Cham, 2014.

[36] Broderick Crawford, Ricardo Soto, Rodrigo Cuesta, and Fernando Paredes. Application of the artificial bee colony algorithm for solving the set covering problem. *The Scientific World Journal*, 2014, 2014.

[37] Broderick Crawford, Ricardo Soto, Eric Monfroy, Wenceslao Palma, Carlos Castro, and Fernando Paredes. Parameter tuning of a choice-function based hyperheuristic using particle swarm optimization. *Expert Systems with Applications*, 40(5):1690–1695, 2013.

[38] Yuhui Shi and Russell C. Eberhart. Empirical study of particle swarm optimization. *Proc. of the Congress on Evolutionary Computation*, pages 1945–1950, 1999.

[39] Broderick Crawford, Ricardo Soto, Eric Monfroy, Fernando Paredes, and Wenceslao Palma. A hybrid ant algorithm for the set covering problem. 2014.

[40] Lucas Lessing, Irina Dumitrescu, and Thomas Stützle. A comparison between aco algorithms for the set covering problem. In *Proceedings of ANTS 2004, Lecture Notes in Computer Science*, volume 3172, pages 1–12, 2004.

[41] Zhigang Ren, Zuren Feng, Liangjun Ke, and Zhaojun Zhang. New ideas for applying ant colony optimization to the set covering problem. *Computers & Industrial Engineering*, 58(4):774 – 784, 2010.

[42] Broderick Crawford, Ricardo Soto, Miguel Olivares-Suárez, and Fernando Paredes. A binary firefly algorithm for the set covering problem. In *Modern Trends and Techniques in Computer Science*, pages 65–73. Springer, 2014.

[43] Broderick Crawford, Ricardo Soto, Marco Riquelme-Leiva, Cristian Peña, Claudio Torres-Rojas, Franklin Johnson, and Fernando Paredes. Modified binary firefly algorithms with different transfer functions for solving set covering problems. In *Software Engineering in Intelligent Systems*, pages 307–315. Springer, 2015.

[44] Broderick Crawford, Ricardo Soto, Cristian Peña, Wenceslao Palma, Franklin Johnson, and Fernando Paredes. *Intelligent Information and Database Systems: 7th Asian Conference, ACIIDS 2015, Bali, Indonesia, March 23-25, 2015, Proceedings, Part II*, chapter Solving the Set Covering Problem with a Shuffled Frog Leaping Algorithm, pages 41–50. Springer International Publishing, Cham, 2015.

[45] Broderick Crawford, Ricardo Soto, Cristian Peña, Marco Riquelme-Leiva, Claudio Torres-Rojas, Franklin Johnson, and Fernando Paredes. *Software Engineering in Intelligent Systems: Proceedings of the 4th Computer Science On-line Conference 2015 (CSOC2015), Vol 3: Software Engineering in Intelligent Systems*, chapter Binarization Methods for Shuffled Frog Leaping Algorithms That Solve Set Covering Problems, pages 317–326. Springer International Publishing, Cham, 2015.

[46] Broderick Crawford, Ricardo Soto, Claudio Torres-Rojas, Cristian Peña, Marco Riquelme-Leiva, Sanjay Misra, Franklin Johnson, and Fernando Paredes. *Computational Science and Its Applications – ICCSA 2015: 15th International Conference, Banff, AB, Canada, June 22-25, 2015, Proceedings, Part IV*, chapter A Binary Fruit Fly Optimization Algorithm to Solve the Set Covering Problem, pages 411–420. Springer International Publishing, Cham, 2015.

[47] Fereidoun Amini and Pedram Ghaderi. Hybridization of harmony search and ant colony optimization for optimal locating of structural dampers. *Appl. Soft Comput.*, 13(5):2272–2280, 2013.

[48] Zhigang Ren, Zuren Feng, Liangjun Ke, and Zhaojun Zhang. New ideas for applying ant colony optimization to the set covering problem. *Computers & Industrial Engineering*, pages 774 – 784, 2010.

[49] Ricardo Soto, Broderick Crawford, Rodrigo Olivares, Jorge Barraza, Franklin Johnson, and Fernando Paredes. *Bioinspired Computation in Artificial Systems: International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2015, Elche, Spain, June 1-5, 2015, Proceedings, Part II*, chapter A Binary Cuckoo Search Algorithm for Solving the Set Covering Problem, pages 88–97. Springer International Publishing, Cham, 2015.

[50] Broderick Crawford, Ricardo Soto, Natalia Berríos, Franklin Johnson, and Fernando Paredes. *Advances in Swarm and Computational Intelligence: 6th International Conference, ICSI 2015, held in conjunction with the Second BRICS Congress, CCI 2015, Beijing, China, June 25-28, 2015, Proceedings, Part I*, chapter Solving the Set Covering Problem with Binary Cat Swarm Optimization, pages 41–48. Springer International Publishing, Cham, 2015.

[51] Broderick Crawford, Ricardo Soto, Natalia Berrios, and Eduardo Olguín. *Artificial Intelligence Perspectives in Intelligent Systems: Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Vol 1*, chapter Cat Swarm Optimization with Different Binarization Methods for Solving Set Covering Problems, pages 511–524. Springer International Publishing, Cham, 2016.

[52] Broderick Crawford, Ricardo Soto, Natalia Berríos, Franklin Johnson, Fernando Paredes, Carlos Castro, and Enrique Norero. A binary cat swarm optimization algorithm for the non-unicost set covering problem. *Mathematical Problems in Engineering*, 2015, 2015.

[53] Broderick Crawford, Ricardo Soto, Felipe Aballay, Sanjay Misra, Franklin Johnson, and Fernando Paredes. *Computational Science and Its Applications – ICCSA 2015: 15th International Conference, Banff, AB, Canada, June 22-25, 2015, Proceedings, Part IV*, chapter A Teaching-Learning-Based Optimization Algorithm for Solving Set Covering Problems, pages 421–430. Springer International Publishing, Cham, 2015.

[54] Broderick Crawford, Ricardo Soto, Gonzalo Astudillo, and Eduardo Olguín. *Artificial Intelligence Perspectives in Intelligent Systems: Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Vol 1*, chapter Fireworks Explosion Can Solve the Set Covering Problem, pages 477–490. Springer International Publishing, Cham, 2016.

[55] Broderick Crawford, Ricardo Soto, Luis Riquelme, and Eduardo Olguín. *Artificial Intelligence Perspectives in Intelligent Systems: Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Vol 1*, chapter Biogeography-Based Optimization Algorithm for Solving the Set Covering Problem, pages 273–283. Springer International Publishing, Cham, 2016.

[56] Ricardo Soto, Broderick Crawford, Cristian Galleguillos, Jorge Barraza, Sebastián Lizama, Alexis Muñoz, José Vilches, Sanjay Misra, and Fernando Paredes. *Computational Science and Its Applications – ICCSA 2015: 15th International Conference, Banff, AB, Canada, June 22-25, 2015, Proceedings, Part I*, chapter Comparing Cuckoo Search, Bee Colony, Firefly Optimization, and Electromagnetism-Like Algorithms for Solving the Set Covering Problem, pages 187–202. Springer International Publishing, Cham, 2015.

[57] Broderick Crawford, Ricardo Soto, Cristian Peña, Marco Riquelme-Leiva, Claudio Torres-Rojas, Sanjay Misra, Franklin Johnson, and Fernando Paredes. *Computational Science and Its Applications – ICCSA 2015: 15th International Conference, Banff, AB, Canada, June 22-25, 2015, Proceedings, Part IV*, chapter A Comparison of Three Recent Nature-Inspired Metaheuristics for the Set Covering Problem, pages 431–443. Springer International Publishing, Cham, 2015.

[58] Limin Han, Graham Kendall, and Peter Cowling. An adaptive length chromosome hyperheuristic genetic algorithm for a trainer scheduling problem. In *Proceedings of the fourth Asia-Pacific Conference on Simulated Evolution And Learning, (SEAL'02), Orchid Country Club, Singapore*, pages 267–271, 2002.

[59] Broderick Crawford, Ricardo Soto, Franklin Johnson, Carlos Valencia, and Fernando Paredes. *Artificial Intelligence Perspectives in Intelligent Systems: Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Vol 1*, chapter Firefly Algorithm to Solve a Project Scheduling Problem, pages 449–458. Springer International Publishing, Cham, 2016.

[60] Ricardo Soto, Broderick Crawford, Jacqueline Lama, and Fernando Paredes. *Artificial Intelligence Perspectives in Intelligent Systems: Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Vol 1*, chapter A Firefly Algorithm to Solve the Manufacturing Cell Design Problem, pages 103–114. Springer International Publishing, Cham, 2016.

[61] Broderick Crawford, Carlos Castro, and Eric Monfroy. *MICAI 2006: Advances in Artificial Intelligence: 5th Mexican International Conference on Artificial Intelligence, Apizaco, Mexico, November 13-17, 2006. Proceedings*, chapter A Hybrid Ant Algorithm for the Airline Crew Pairing Problem, pages 381–391. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[62] James W. Chrissis. Locating emergency service facilities in a developing area. *Fire Technology*, 16(1):63–69, 1980.

[63] Rafael Martí, Departament D estadística I Investigaci'ø Operativa, and Facultat De Matemátiques. Procedimientos metaheurísticos en optimización combinatoria.

[64] Alberto Caprara, Matteo Fischetti, and Paolo Toth. A heuristic method for the set covering problem. *Operations Research*, 47(5):730–743, 1999.

[65] Barbara M. Smith. Impacs - a bus crew scheduling system using integer programming. *Mathematical Programming*, 42(1):181–187, 1988.

[66] Endre Boros, Peter L. Hammer, Toshihide Ibaraki, and Alexander Kogan. Logical analysis of numerical data. *Math. Program.*, 79:163–190, 1997.

[67] Francis J. Vasko, Floyd E. Wolf, and Kenneth L. Stott. A set covering approach to metallurgical grade assignment. *European Journal of Operational Research*, 38(1):27–34, 1989.

[68] David M. Ryan and B. A. Foster. An integer programming approach to the vehicle scheduling problem. *Operational Research Quarterly*, 27:367–384, 1976.

[69] Marshall L. Fisher and Moshe B. Rosenwein. An interactive optimization system for bulk-cargo ship scheduling. *Naval Research Logistics (NRL)*, 36(1):27–42, 1989.

[70] Francis J. Vasko, Floyd E. Wolf, Kenneth L. Stott, and James W. Scheirer. Selecting optimal ingot sizes for bethlehem steel. *Interfaces*, 19(1):68–84, 1989.

[71] Wei Shen, Xiaopen Guo, Chao Wu, and Desheng Wu. Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm. *Knowledge-Based Systems*, 24(3):378–385, 2011.

[72] Wenjie Tian and Jicheng Liu. An improved artificial fish swarm algorithm for multi robot task scheduling. In *Natural Computation, 2009. ICNC'09. Fifth International Conference on*, volume 4, pages 127–130. IEEE, 2009.

[73] Cuiru Wang, Chunlei Zhou, and Jianwei Ma. An improved artificial fish-swarm algorithm and its application in feed-forward neural networks. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 5, pages 2890–2894. IEEE, 2005.

[74] Xiaolei Li and Jixin Qian. Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques. *Journal of Circuits and Systems*, 1:1–6, 2003.

[75] Mingyan Jiang and Dongfeng Yuan. Wavelet threshold optimization with artificial fish swarm algorithm. In *2005 International Conference on Neural Networks and Brain*, 2005.

[76] Saeed Farzi. Efficient job scheduling in grid computing with modified artificial fish swarm algorithm. *International Journal of computer theory and engineering*, 1(1):13, 2009.

[77] John E. Beasley. An algorithm for set covering problem. *European Journal of Operational Research*, 31(1):85–93, 1987.

[78] Egon Balas and Andrew Ho. *Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study*. Springer, 1980.

[79] John E. Beasley. A lagrangian heuristic for set-covering problems. *Naval Research Logistics (NRL)*, 37(1):151–164, 1990.

[80] Jose M. Lanza-Gutierrez and Juan A. Gomez-Pulido. Assuming multiobjective metaheuristics to solve a three-objective optimisation problem for relay node deployment in wireless sensor networks. *Applied Soft Computing*, 30:675–687, 2015.

[81] Rodrigo Cuesta, Broderick Crawford, Ricardo Soto, and Fernando Paredes. An artificial bee colony algorithm for the set covering problem. In *Modern Trends and Techniques in Computer Science*, pages 53–63. Springer, 2014.

[82] Broderick Crawford, Ricardo Soto, Miguel Olivares-Suárez, and Fernando Paredes. A binary firefly algorithm for the set covering problem. In Radek Silhavy, Roman Senkerik, Zuzana Kominkova Oplatkova, Petr Silhavy, and Zdenka Prokopova, editors, *Modern Trends and Techniques in Computer Science*, volume 285 of *Advances in Intelligent Systems and Computing*, pages 65–73. Springer International Publishing, 2014.

[83] Sivaramakrishnan R. Balachandar and K. Kannan. A meta-heuristic algorithm for set covering problem based on gravity. *International Journal of Computational and Mathematical Sciences*, 4(5):223–228, 2010.