Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 67 (2018) 362 – 367

www.elsevier.com/locate/procedia

11th CIRP Conference on Intelligent Computation in Manufacturing Engineering CIRP ICME '17

Optimizing the number of acoustic emission sensors using the bees algorithm for detecting surface fractures

Michael S. Packianather[a,*], Mark Eaton[a], Ioannis Papadopoulos[a], Theocharis Alexopoulos[a]

[a]*School of Engineering, Cardiff University, Queen's Buildings, The Parade, Cardiff CF24 3AA, UK*

* Tel.: +0044 29 20875911; fax: +0-000-000-0000 +44 29 20874597. *E-mail address:* PackianatherMS@cf.ac.uk

Abstract

Non-destructive testing methods have gained popularity as they become more widely available. Although there are several techniques that could be used for this purpose, this paper focuses on acoustic emission sensors for detecting surface fractures and the use of the Bees Algorithm, a swarm-based technique, for optimizing the number of sensors required to reliably detect surface fractures. The paper describes the approach that has been used in this study where the dimension of the surface is specified by the user. The results show that, in theory and through simulation, that the Bees Algorithm is capable of determining the minimum number of sensors needed to locate the surface fracture with an acceptable level of accuracy. The method described could be used for the purpose of optimization in other engineering as well as non-engineering applications.

Peer-review under responsibility of the scientific committee of the 11th CIRP Conference on Intelligent Computation in Manufacturing Engineering

Keywords: Acoustic Emission; Structural Health Monitoring; Bees Algorithm;

1. Introduction

The concept of Structural Health Monitoring (SHM) lies behind the need to ensure reliability and smooth operation of a product. In some cases the price for abnormalities in the operation of the product is just monetary. In other cases however, the price could be human lives, injuries, environmental damage, or other examples of socially unacceptable losses. In SHM, there are various methods for real-time damage detection and evaluation. Many forms of alarms are invented so that we, the operators/the users/the passengers/the by-standers can be alerted to the condition of an engineered system that we are dealing with. Usually SHM is achieved by placing various forms of sensors on the product of interest. These sensors work in one way or another in order to provide us with output information regarding the current real-time state of that product. These sensors work under the principle of necessity to achieve full coverage of the part/product, and to ensure reliable detection of any faults occurring in real-time. The engineer is the person who makes sure that these and other additional principles are satisfied in order to achieve reliable SHM. This is a general description of the aims of SHM. It fits however to the specific case with which this paper is dealing with, i.e., the technique of Acoustic Emission [1]. The optimization of parameters involved in acoustic emission and in SHM in general is a task that is performed on site by the human inspector. There is however a very big dependence on the inspector's skill, expertise and knowledge in order to achieve the best results. Therefore, it is attempted nowadays by many researchers and optimizers to create computational algorithms that employ various mathematical, heuristic as well as artificial intelligence techniques in order to bring standardization and total optimization to the processes like Acoustic Emission, and obtain thus the best results of SHM every time [2, 3, 4, 5]. This paper aims to thoroughly investigate the feasibility of attempting to tackle the optimization problem of acoustic emission sensor placement and sensor number minimization with the use of various standard algorithms as well as with the artificial intelligence and nature inspired Bees Algorithm. The

Peer-review under responsibility of the scientific committee of the 11th CIRP Conference on Intelligent Computation in Manufacturing Engineering
doi:10.1016/j.procir.2017.12.227

paper is organized as follows. The basic Acoustic Emission algorithm is described in section 2. The problem of finding the optimum position and number of acoustic emission sensors necessary to get full coverage for a given area is explained in section 3. Then finding the solution using the Random search approach and Bess Algorithm search approach are presented in sections 4 and 5. Finally the conclusion is given in section 6.

2. The initial acoustic emission Algorithm

A basic Acoustic Emission software that was created with MATLAB served as an initial platform which was later adapted and improved. Initially the program attempts to visualize an orthogonal rectangular metallic (Aluminum) plate by asking the user for certain parameters such as the plate length, width as well as thickness (the length of the z dimension). The metallic plate is then created in the form of a graph as shown in Figure 1.

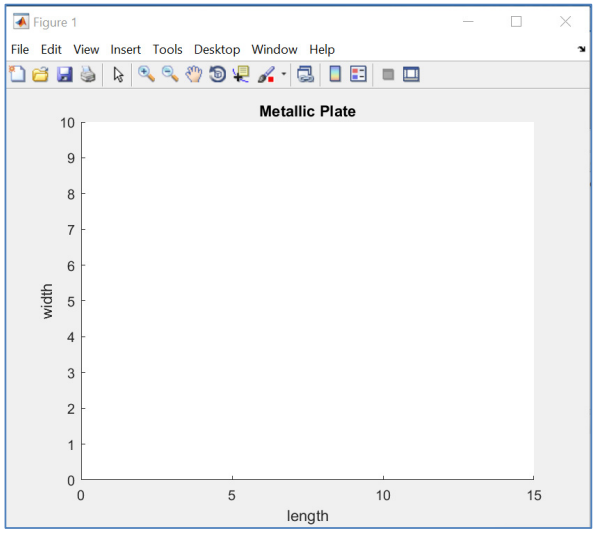

Fig. 1. The metallic plate is 'created' in the form of a graph.

The program then asks the user details about the acoustic emission event (the 'hit') that is going to occur on the plate. These details are the magnitude of the hit expressed in decibels (dB) and the minimum detection threshold bellow which any acoustic emission is disregarded as non-relevant with the testing procedure, or is considered as background noise. The attenuation curve for the acoustic emission signal is then created as shown in Figure 2.

The attenuation curve shown in red is calculated using the simple exponential decay function given in Equation 1.

$$A(x) = A_o * e^{-\alpha x} \quad (1)$$

Where:

- A_o is the initial 'source' amplitude of the hit in dB
- α is the attenuation coefficient which is set to 1.
- $A(x)$ is the value of the amplitude of the acoustic emission signal at a distance (x) away from its source.

The straight line shown in blue is plotted in order to show on the graph the point below which the acoustic emission signal is disregarded as non-relevant with the testing procedure, or is considered as background noise, i.e., the minimum detection threshold that was input earlier. It is immediately noticed that the intersection of the two curves is shown by a circle. At this distance the signal's amplitude drops below the threshold indicated by the user and it is therefore taken as the maximum distance up to which the signal travels. This is in fact not true because in reality the signal travels around the plate many times and bounces off the edges (edge effect) and depending on the thickness of the plate it does not attenuate easily. However, this distance is very useful because it can be considered to be the radius for the range of the sensors to be used later. This is because the sensors will act as an exponentially decaying source but only the other way around. They receive any signal starting from that distance away from them. In other words, they receive the lowest amplitude of the signal set by the user, provided that it is within the indicated distance/radius. The acoustic emission software will calculate and output this distance. For example, when the dimension of the plate is 15x10x1, the magnitude of the hit is 100dB and the threshold is 10dB then the distance/radius of the sensor was calculated to be 2.3232 units.

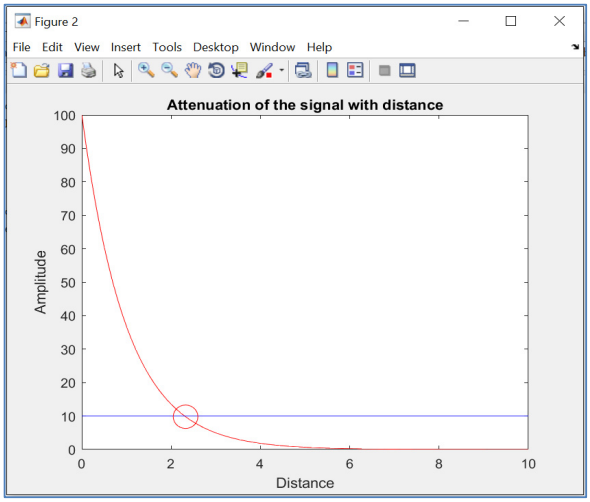

Fig. 2. The Attenuation of the signal with distance.

The algorithm in the software then begins the task of trying to position the sensors on the plate. It asks the user with a dialog box whether the sensor placement should take place manually or automatically. In manual sensor placement, the user inputs the exact locations where the sensors are to be mounted on the plate. In the case of the automatic sensor placement, the software calculates the number of sensors and positions needed in order to achieve full coverage of the plate area with the use of the sensor radius found earlier.

In the case of manual placement, the software takes the user inputs and does some calculations in order to show the Metallic Plate graph that was presented earlier with the sensors (dark blue) on the plate along with their radii (light blue). The source (red star) along with the radius (in red) indicated earlier by the exponential decay function (which is said to be the sensor radius) is illustrated in Figure 3.

If it is done automatically, so that full coverage of the whole plate is the goal and not just coverage of a specific area, the following sequence of events occurs. Initially, the software creates an invisible grid on the surface of the plate. The target is to create a sensor grid that provides full coverage to the whole area of the plate. The spacing between the lines both vertically as well as horizontally is equal to the sensor radius 'R'. At the intersections of the lines lie the sensor coordinates. The spacing between the sensors is obviously equal to 'R' as well. The spacing distance of 'R' was preferred over the '1.5R' due to the simple fact that a spacing of '1.5R' would provide insufficient coverage in some cases and would therefore leave uncovered areas on the plate. The thinking process behind this is schematically represented in Figure 4.

The software then asks the user to indicate the coordinates of the acoustic emission event (the damage source). The source coordinates are more or less irrelevant in terms of whether the source is covered or not because through automatic sensor placement the program would achieve full

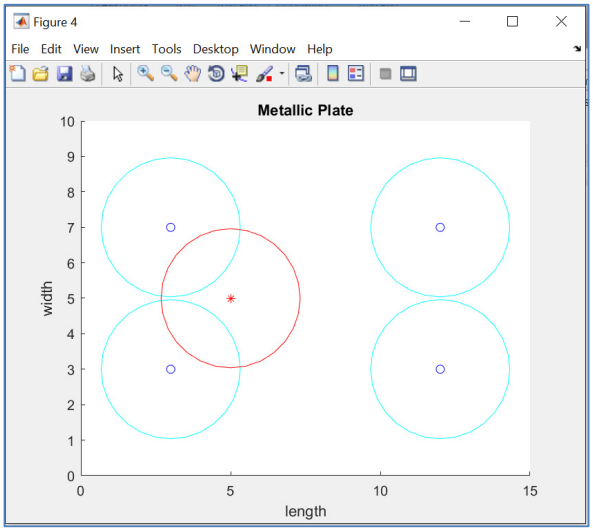

Fig. 3. The metallic plate along with the sensors which were manually placed by the user.

plate coverage as shown in Figure 5. Additionally, the program calculated the distance of the source from each sensor, as well as the time that was required for the acoustic emission signal to travel and arrive at each particular sensor. Although this gives full coverage, the focus of this paper is to investigate if the same could be achieved by placing a fewer number of sensors at some selected positions as discussed in the following section.

3. Optimizing the acoustic emission algorithm in terms of sensor positions and sensor numbers

The goal of full coverage was achieved with the basic Acoustic Emission software. However, lots of sensors were needed and they were located away from the plate's edges. Although the results computationally fulfilled the purpose of covering the plate, in practice, the signal detection that they offer will suffer from edge effects or low detectability due to the specific spatial configuration in actual conditions. The number of required sensors calculated by the algorithm in this example for a plate of 15x10x1 units was 35. This number was later compared with the number of sensors calculated by other optimization algorithms. Usually these sensors can be expensive and minimization of the number of sensors required to achieve full coverage will lead to great savings. It was assumed that with the use of artificial intelligence techniques such as the Bees Algorithm, and by placing the sensors at random positions on the plate while still aiming to achieve 100% coverage of the plate (by satisfying a certain cost function), the result would have an advantage over placing the sensors in a well-defined ordered manner. In particular, the assumption was that such an approach could achieve a reduction of the number of sensors used whilst still maintaining 100% coverage of the area of the plate.

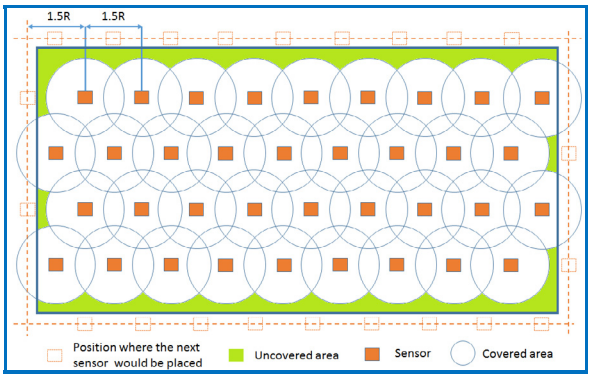

Fig. 4. Schematic illustration of why the sensor spacing should not be equal to any distance other than 'R'.

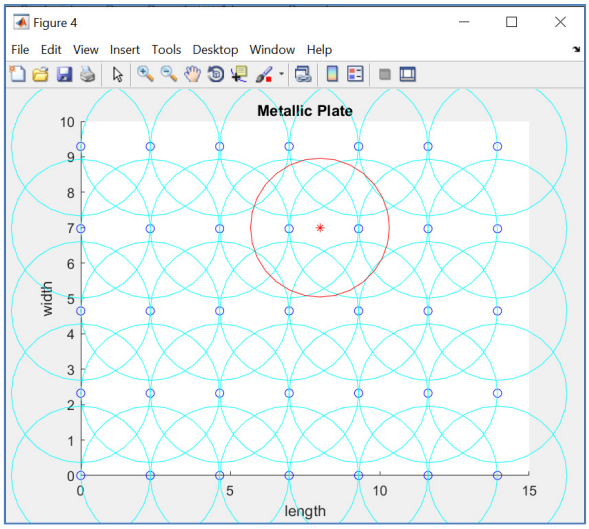

Fig. 5. The metallic plate along with the sensors which were automatically placed by the software.

4. The Random Search Approach

This search performed random placement of the sensors on the plate where the sensors were randomly mounted on various positions on the plate and then the coverage that they provide on the plate area was calculated. The final result is a percentage of the plate area that is covered by an adequate amount of sensors such that complete signal detection is achieved. The inputs required are plate dimension (length and width), the number of sensors available and their range.

After the required input by the user i.e. 15x10, 30 and 2.3232, the algorithm randomly places the number of sensors specified on the plate's area. It then calculates the percentage of the area that sensors cover w.r.t the plate area. It does that by creating a grid of points on the plate with a spacing of 0.5 units of sensor range (this depends on the resolution that is required) between them and looping through all of these points (thus looping throughout the whole plate) in order to count how many points are being covered by the sensors. A point is covered if it lies within the range of 3 sensors. The algorithm produces the graph shown in Figure 6 where the plate's boundaries are enclosed by the red lines. The blue shaded circles are the areas covered by each sensor. The algorithm counts the number of grid points that are within the range of three or more sensors (and are thus considered to be fully covered) and divides this number over the total grid points of the whole plate area. This is the percentage of the plate where full detection is achieved (in terms of coverage only). This method based on a random approach achieves higher coverage with higher number of sensors. Hence, what is needed is an intelligent algorithm which can mix heuristic and random search to find the optimum number of sensors and positions as described in the following section.

5. The Bees Algorithm Search Approach

The Bees Algorithm [6, 7] is an intelligent swarm based optimization algorithm which mimics the foraging behavior of honey bees found in nature. The algorithm maintains the search through the space by carrying out local search based on heuristics but at the same time allowing random search to take place with a small probability in order to avoid premature convergence. This approach presented in the form of a flowchart is Figure 7 can be used in order to search various search spaces and offer the best solutions from all the available options, depending on the problem at hand. The Bees Algorithm has been applied in areas like optimization of classifiers or clustering [8], manufacturing [9] and logistics [10, 11].

In this case, the Bees Algorithm was modified and sorted accordingly so that it could offer a solution for the problem of optimal sensor placement whilst reducing the sensor number to a minimum. The Bees Algorithm code consists of two parts. The search algorithm itself and an objective function.

The operation of the objective/fitness function inside the Bees Algorithm was substituted by another algorithm. The purpose of this algorithm is to accept some certain parameters as inputs and give back a specific sensor position/configuration as output that corresponds to the values given to these parameters. The initial idea was to randomly place a specified number of sensors with a specified radius on a metallic plate area of specified dimension. For example, the dimensions were set to 20x20, 5, and 2 units. Figure 8 shows the algorithm placing the 5 sensors randomly on the area of the plate.

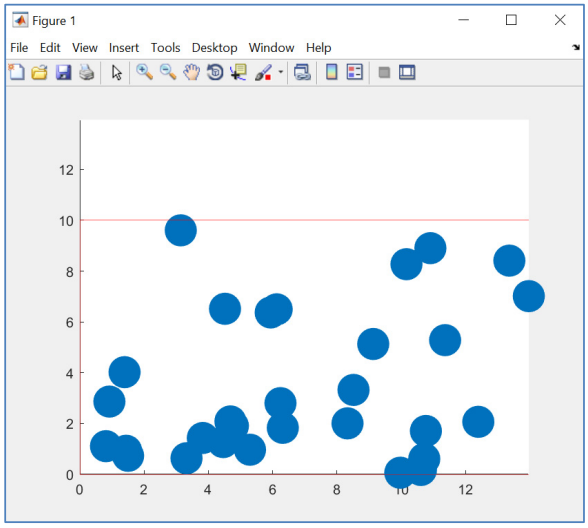

Fig. 6. Graph showing the plate and the areas on the plate covered by the randomly placed sensors.

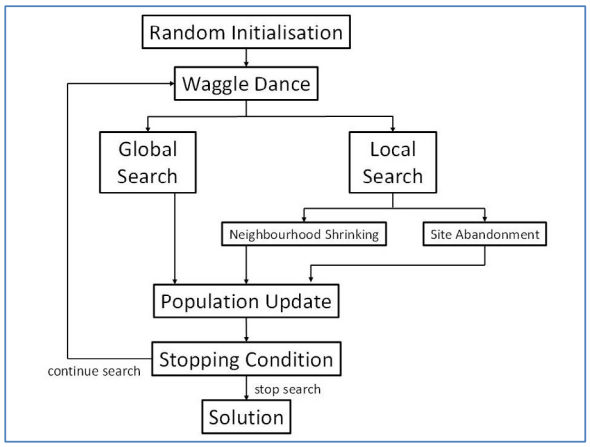

Fig. 7. Bees Algorithm Flowchart (The Bees Algorithm Webpage 2016).

The search begins by calling the function named “Fittest distances" by finding the distance of each sensor plotted in the plate-graph of Figure 8 with every other sensor on the plate as shown in Figure 9.

The order of the sensors is incremental. Starting from reference sensor 1 and presenting its distance from each consecutive sensor. The second set of distances shown are the distances of the reference sensor (sensor number 2 this time) from all the other sensors. It continues in this way for sensor number 3, 4 and 5 making them the reference sensors each time. The algorithm then chooses for each reference sensor the distance between it and another sensor that is closest to the optimum distance value of ‘R’ i.e. closest to the sensor radius specified.

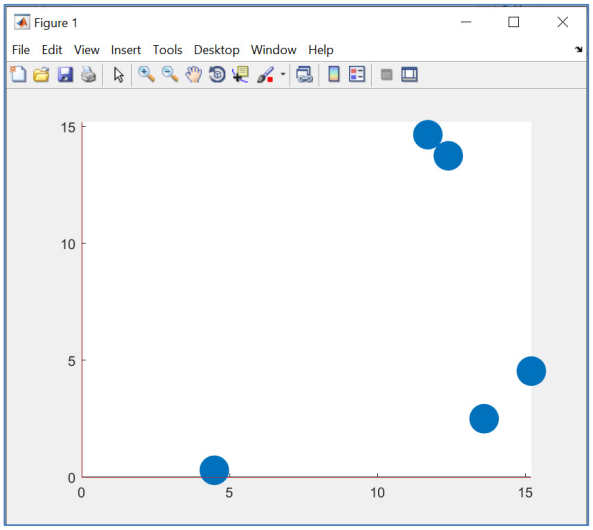

Fig. 8. The sensors are randomly placed on the metallic plate.

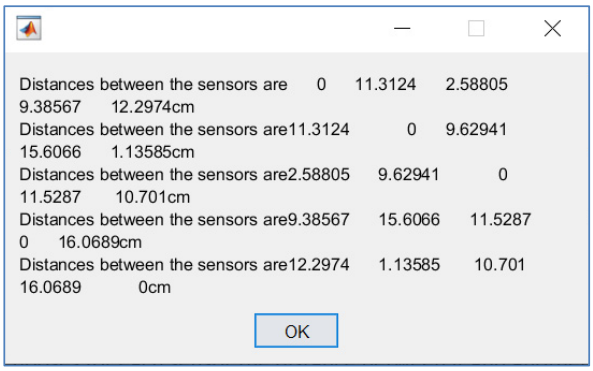

Fig. 9. Distances between sensor pairs.

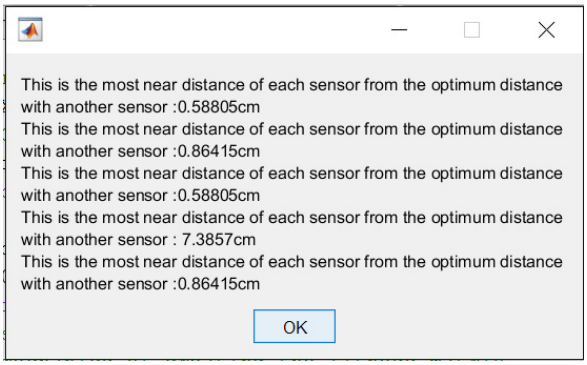

Fig. 10. Starting from sensor 1, the fittest distance of the reference sensor with its sensor pair.

For each sensor (starting from sensor 1 and continuing incrementally) the distance from a pair of sensors that is closest to the value ‘R’ (i.e. its fitness criterion) from all its other pairs of distances is recorded as shown in Figure 10.

It then outputs a message box that shows for each reference sensor starting from sensor 1 and finishing at sensor 5 the index of the other sensor that makes a pair with it whose distance is the closest to the value of R (i.e. the fittest). This is shown in Figure 11.

The sensor radius was chosen as the optimum distance between a pair of sensors. It is then displayed how close to the value ‘R’ is each distance value that occurs between the reference sensor and other sensors. This is the deviation of a pair’s distance from the optimum value i.e. a fitness criterion/value given in Figure 12.

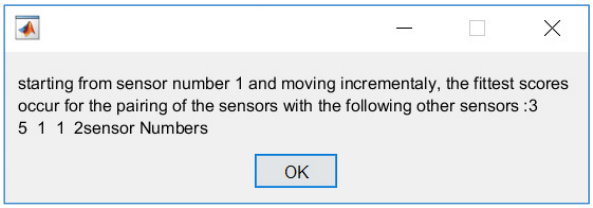

Fig. 11. For each reference sensor, the index of the matching sensor that allows the distance between them to be closest to the optimum distance R

```
b =

         0    9.3124    0.5881    7.3857   10.2974
    9.3124         0    7.6294   13.6066    0.8641
    0.5881    7.6294         0    9.5287    8.7010
    7.3857   13.6066    9.5287         0   14.0689
   10.2974    0.8641    8.7010   14.0689         0
```

Fig. 12. The proximity of each sensor distance from the reference sensor (denoted each time as a zero) to the optimum value of 'R'.

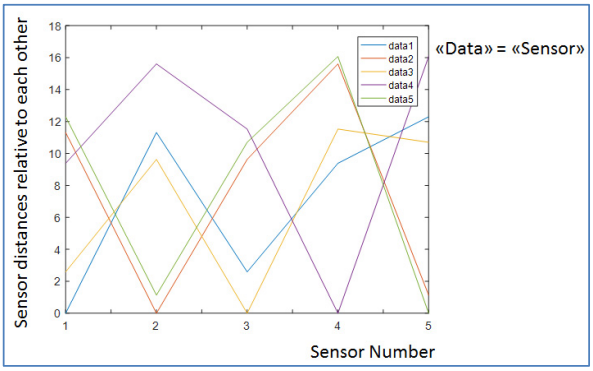

Fig. 13. The distances of the sensors on the plate relative to each other.

The algorithm outputs the graph shown in Figure 13 where the distance of each sensor from every other sensor is plotted against the sensor numbers (essentially the distances of the sensors relative to each other).

For each sensor its fittest pair is then chosen and presented in a matrix along with the fitness value that corresponds to that pair of sensors. In the end, a big 'field' of sensor pairs is produced along with their individual distances. Note that this whole procedure would happen for a vast amount of random sensor locations so that the resulting number of pairs would be big enough so that it qualifies for being used for Bees Algorithm. This field can then serve as the search space on which the Bees Algorithm search mechanisms can operate and find the global maxima (fittest distances). This was the philosophy behind the algorithm that served as the objective/fitness function for the Bees Algorithm search procedure.

6. Conclusion

In this paper a method of optimizing the position and number of Acoustic Emission sensors required to achieve full coverage in order to detect surface fractures have been presented. The method described is an intelligent optimization method based on swarm intelligence which is referred to as the Bees Algorithm. The study has shown that the proposed Bees Algorithm is capable of searching through a vast search space in an intelligent way in order to find the optimum solution by way of following both heuristic and random approach.

Acknowledgements

The authors would like to thank ASTUTE2020 for supporting this work.

References

[1] Carlos M. Acoustic emission heeding the warning sounds from materials. ASTM Standardization News, 2003; 31(10): 26-29.
[2] Yi T, Li H, Zhang X. A modified monkey algorithm for optimal sensor placement in structural health monitoring. Smart Mater. Struct. 2012; 21(10): 105033-105042.
[3] Rao ARM, Lakshmi K, Krishnakumar S. A generalized optimal sensor placement technique for structural health monitoring and system identification. Procedia Engineering, 2014; 86; 529-538.
[4] Yi TH, Li HN, Gu M. Optimal sensor placement for health monitoring of high-rise structure based on genetic algorithm. Mathematical Problems in Engineering, 2011; 2011: 395101-395113.
[5] Baxter MG, Pullin R, Holford KM, Evans SL. Delta T source location for acoustic emission. Mechanical systems and signal processing, 2007; 21(3): 1512-1520.
[6] Yuce B, Packianather MS, Mastrocinque E, Pham DT, Lambaise A. Honey bees inspired optimization method: the bees algorithm. Insects, 2013; 4(4): 646-662.
[7] Packianather MS, Landy M, Pham DT. Enhancing the speed of the Bees Algorithm using Pheromone-based Recruitment, IEEE INDIN, 2009; 789-794.
[8] Packianather MS, Kapoor B. A wrapper-based feature selection approach using Bees Algorithm for a wood defect classification system. IEEE System of Systems Engineering Conference (SoSE), 2015; 498-503.
[9] Yuce B, Pham DT, Packianather MS, Mastrocinque E. An enhancement to the Bees Algorithm with slope angle computation and Hill Climbing Algorithm and its applications on scheduling and continuous-type optimisation problem. Production & Manufacturing Research, 2015; 3(1): 3-19.
[10] Mastrocinque E, Yuce B, Lambiase A, Packianather MS. A multi-objective optimization for supply chain network using the bees algorithm. International Journal of Engineering Business Management, 2013; 5(38): 1-11.
[11] Yuce B, Mastrocinque E, Lambiase A, Packianather MS, Pham DT. A multi-objective supply chain optimisation using enhanced Bees Algorithm with adaptive neighbourhood search and site abandonment strategy. Swarm and Evolutionary Computation, 2014; 18: 71-82.