CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter presents the research stage and the proposed design. This research adopts Design Science Research Methodology (Vom Brocke et al., 2020) to achieve three research objectives. This research is divided into three main stages, each delivering one of the objectives mentioned in Chapter 1. These stages are portrayed in Figure 3.1 and discussed in detail throughout Section 3.2. Next, the proposed design is discussed in detail. The formulations which enhanced solution that can select audio based on the trade-off between all characteristics on the cover audio selection and improves the performance of dynamic security characteristic of audio steganography are discussed in this part.

3.2 Research Stage

There are three stages conducted in this research. The research question, research objective, research process and research output for each stage are presented in Figure 3.1.

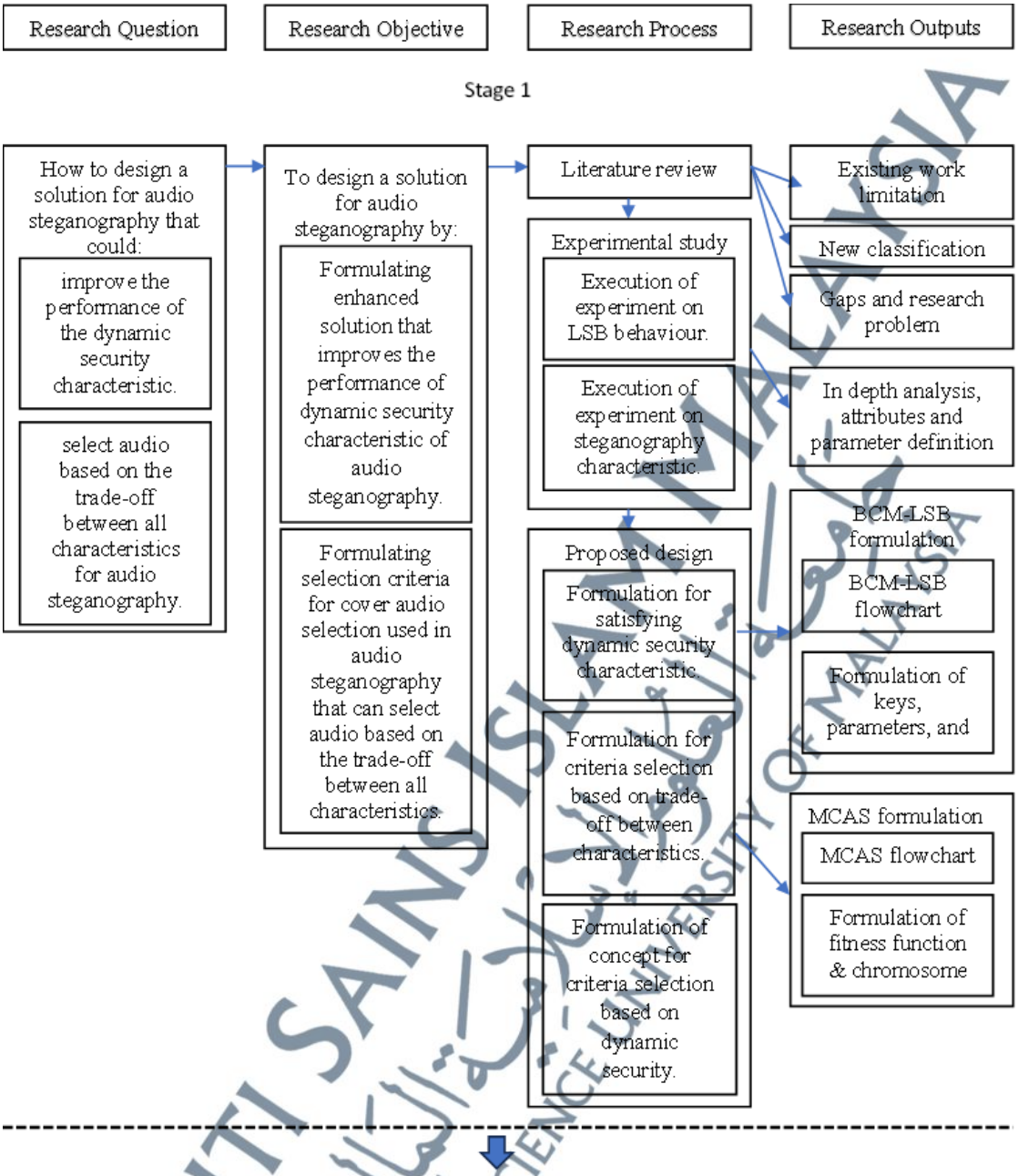

Figure 3.1: Research Stage

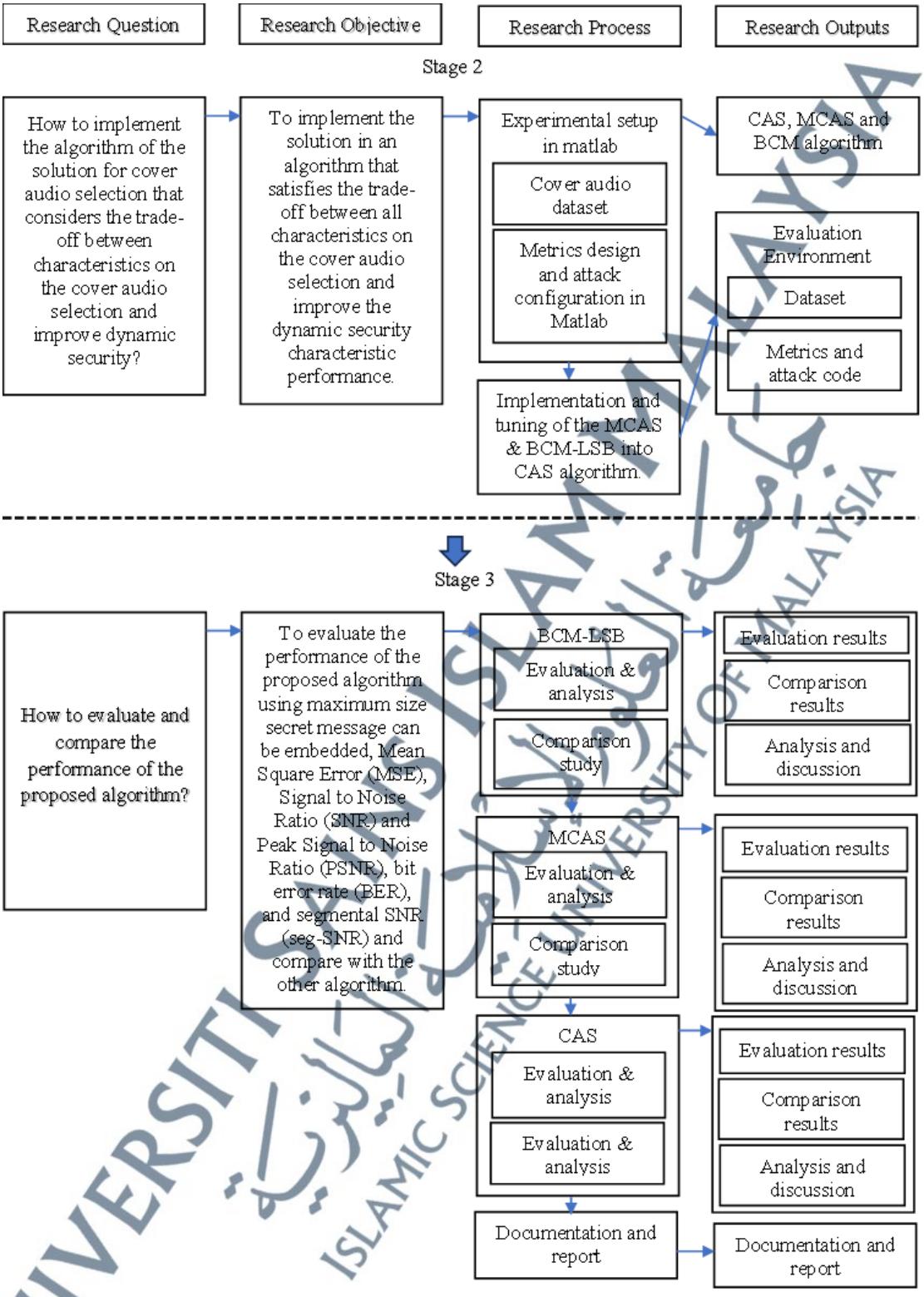

Figure 3.1: Research Stage, continued

3.2.1 First Stage: Design Proposed Solution

This first stage was covered in Chapter 2 and Chapter 3. The first part of the research process in this stage is literature review. The existing work, limitation, gap, and research problem are presented in Section 2.4 and Section 2.5. Next, the new classification for cover selection is presented in Section 2.6. Then, an experimental study is presented in Section 2.2.2 to produce an in-depth analysis, attributes and parameter definition focused more on dynamic security characteristics. The last research process under this stage is to design a solution by creating related formulations to reflect objective 1 which is discussed in Section 3.3. This section discussed Block-based Chaotic Multi-level Least Significant Bit (BCM-LSB) formulation and Multi Characteristic Audio Selection (MCAS) formulation including flowchart, formulation of keys, parameters, MODI and formulation of fitness function & chromosome. Several important components under BCM-LSB are 1) chaotic block which is based on chaotic map and variable bit embedded per sample, and 2) rounding mechanism which is based on fully utilising the audio samples. On the other hand, the important components of the MCAS are the 1) evaluation metrics and 2) selection rules.

3.2.2 Second Stage: Implementation of Proposed Solution

The second stage focuses mainly on the implementation of the proposed solution empirically using Matlab, which involves changing the parameter settings used in block-based chaotic multi-level LSB (BCM-LSB) and multi-characteristic-based cover audio selection (MCAS). These two proposed methods are then combined as Cover-Selection-Based Audio Steganography (CAS) which is discussed in Chapter 4.

3.2.3 Third Stage: Evaluation of Proposed Solution

The third stage of the research involved evaluating the proposed solution. Two main parts of evaluations are structured which are: 1) audio steganography comparison experiments and 2) cover selection comparison experiments. Section 3.2.3.1 discusses the audio dataset used for the evaluation. Section 3.2.3.2 discusses the evaluation environment used on steganography comparison experiments, while Section 3.2.3.3 presents the evaluation environment used on cover selection comparison experiments. The results of evaluation on algorithm's parameters and comparison studies are analysed and discussed in Chapter 5. Documenting entire work is the last stage in the research workflow. The documentation process includes both writing the thesis chapters and publishing papers.

3.2.3.1 Audio Dataset

Many audio parameters could give an impact to the experimental result. Therefore, several parameters such as audio format, sampling rate, audio sample size and audio duration range are fixed. The audio format used in this research is a mono wave file with a 44.1kHz sampling rate and 16-bit per sample, selected based on audio length from the combination of audio from speech and music. They ranged from one (1) to ten (10) seconds which were downloaded from a free online audio library, www.freesound.org. The dataset used in this research is added as media files (CDs) in Appendix 12.

3.2.3.2 Metric Used During Evaluation

Metrics used in the existing research are presented in Section 2.2.2 to measure algorithm performance. The selected metric(s) for evaluating this research are i)

imperceptibility, ii) capacity, iii) robustness, iv) dynamic security and v) quality and quantity of solution found. The evaluation metrics are presented in Table 3.1 and Table 3.2.

Table 3.1: Evaluation Metrics Used for Audio Steganography Characteristics

Characteristic	Metric Used	Formulation	Interpretation
Imperceptibility	SNR	Equation 2.2	Higher SNR indicates higher imperceptibility
	MSE	Equation 2.3	Lower MSE indicates higher imperceptibility
	PSNR	Equation 2.4	Higher PSNR indicates higher imperceptibility
Capacity	Maximum possible bits embedded	Explained in Section 2.2.2.1	Higher maximum possible bits embedded indicates higher capacity
Robustness	BER on AWGN attack	Equation 2.5	Lower BER indicates higher robustness
Dynamic Security	Seg-SNR Spike Visual Test	Explained in Section 2.2.2.5	Lower the spike produced in the seg-SNR graph, indicates higher dynamic security
	Difference Signal Visual Test	Explained in Section 2.2.2.5	Better distribution shown in the visual indicates higher dynamic security

Table 3.2: Evaluation Metrics Used for Solution Found by Cover Selection Method

Characteristic	Metric Used	Description	Interpretation
Quality and quantity of solution found	Number of Pareto solutions (NPS)	Equation 2.2	Higher NPS indicates better quantity of solution founds
	Best Solution, *Best_Sol*	highest quality in term highest normalized value between the several values calculated	Higher *Best_Sol* indicates better quality of solutions found
	number of dominated solutions, (NDS)	number of solutions dominated when solution in algorithm's Pareto front compared with the other solution in algorithm's Pareto front.	Lower NDS indicates better quality of solutions found
	$Worst_{Sol}$	Lowest value of *Best_Sol*	Higher $Worst_{Sol}$ indicates better quality of solutions found

3.2.3.3 Performance Evaluation Environment on Proposed Algorithm Parameters

The evaluation conducted on the proposed algorithm is divided into two subsection which are evaluation on the BCM-LSB parameters and MCAS parameters.

- BCM-LSB Parameters Evaluation Experiment

There are four evaluations conducted on two parameters, *bps* and set of steganographic key used (key *x*, key *n* and key *r*). The audio used are presented in the evaluation experiments presented in Table 3.3.

Table 3.3: Cover Audio Used With its Duration

Name	Duration (Second)
Music 1	1
Music 2	2.144
Music 3	3.154
Music 4	5.323
Music 5	9
Speech 1	1
Speech 2	2.067
Speech 3	3.19
Speech 4	5.486
Speech 5	9.366

The evaluation environments are listed below are presented in Table 3.4.

Table 3.4: Evaluation Conducted on BCM-LSB Parameters

Experiment	Objective	Setting
Impact of *bps* used on capacity	To investigate the *bps* against their maximum capacity.	• Ten random audio files from the combination of music-based and speech-based as in Table 3.3.
Impact of *bps* used on imperceptibility	To investigate the *bps* against their imperceptibility at maximum capacity.	• The same ten random audio files from the first set of capacity evaluations are used.

Experiment	Objective	Setting
Impact of *bps* used on robustness	To investigate the *bps* used against their robustness performance.	• Three random audio files are used. • Half maximal embedded capacity of each BCM-LSB variations algorithm is used
Impact of set of keys used on dynamic security	To investigate set of keys used against their dynamic security performance	• Seg-SNR Spike Visual Test are used. • One random audio is used. • maximum embedded capacity BCM-LSB algorithm are set at 25% and 35%.

- MCAS Parameters Evaluation Experiment

There are three evaluations conducted three parameters, number of generations used, size of population used, and implementation of removal duplicated solution. The evaluation environments are listed in Table 3.5.

Table 3.5: Evaluation Conducted on MCAS Parameters

Experiment	Objective	Setting
Impact of number of generations used on quality and quantity of the solution found	To investigate the number of generations used against the quality and quantity of the solution found	• 300 audio files were set in the database. • crossover probability value = 0.8 • mutation probability value = 0.05. • Embedding method parameters: ○ *bps* = 1-8 ○ key *x* = 92 ○ key *r* = 3.64 ○ key *n* = 1024
Impact of size of population used on quality and quantity of the solution found	To investigate the size of population used against the quality and quantity of the solution found	Similar setting as previous experiment.
Impact of implementation of removal duplicated solution on quality and quantity of the solution found	To investigate the implementation of removal duplicated solution against the quality and quantity of the solution found	Similar setting as previous experiment.

3.2.3.4 Performance Comparison Environment Against Existing Algorithms

The evaluation conducted on the proposed algorithm is divided into three subsection which are comparison on LSB embedding level, cover selection level and audio steganography model level. The evaluation environments are listed below.

- Comparison at LSB Embedding Level

  There are four comparisons on capacity, imperceptibility, robustness and dynamic security against the existing method. The selected existing methods are discussed further in Section 4.3.1. The comparisons are discussed in Table 3.6.

Table 3.6: Comparison Evaluation Conducted at LSB Embedding Level

Experiment	Objective	Setting
Comparison on the imperceptibility.	To compare the BCM-LSB against existing algorithm in terms of imperceptibility.	• Three audio files (Music Box, Speech-based and Music-based) with different natures and lengths are used. • Music Box is exactly 1 second in length. Purpose: reveal the performance of the methods within this specific second • Speech-based and Music-based with the duration around 3 seconds. Purpose: show a difference in performance when the nature of audio is changed. • The existing methods used for comparison are discussed in Section 4.3.1
Comparison on the capacity	To compare the BCM-LSB against existing algorithm in terms of capacity.	• Similar setting as previous experiment.
Comparison on the robustness	To compare the BCM-LSB against existing algorithm in terms of robustness.	• Similar setting as previous experiment.
Comparison on the dynamic security	To compare the BCM-LSB against existing algorithm in terms of dynamic security.	• Similar audio files are used as imperceptibility comparison. • uses the Seg-SNR Spike Visual Test and Difference Signal Visual Test. • embed 25% and 35% of the maximum capacity.

- Comparison at Cover Selection Level

There are two comparisons on the dynamic security formulation used and trade-off consideration during cover selection against the existing method. The selected existing methods are discussed further in Section 4.3.2. The comparisons are presented in Table 3.7.

Table 3.7: Comparison Evaluation Conducted at Cover Selection Level

Experiment	Objective	Setting

Comparison on trade-off consideration during cover selection	To compare the MCAS algorithm which consider the trade-off against existing cover selection algorithm which do not consider the trade-off in terms of quality and quantity solutions found.	• The existing methods used for comparison are discussed in Section 4.3.2.
		• 300 audio files were set in the database.
		• crossover probability value = 0.8
		• mutation probability value = 0.05.
		• Size of secret message = 4 KB and 8 KB.
		• Number of generations: 20.
		• Size of population:300.
		• Embedding method parameters:
		  ○ *bps* = 1-8
		  ○ key *x* = 92
		  ○ key *r* = 3.64
		  ○ key *n* = 1024

Comparison on the Dynamic Security Formulation Used	To compare the usage of newly proposed dynamic security metric used in MCAS against existing dynamic security evaluation metric in terms of quality and quantity solutions found.	• 300 audio files were set in the database.
		• crossover probability value = 0.8
		• mutation probability value = 0.05.
		• Size of secret message = 4 KB and 8 KB.
		• Number of generations: 20.
		• Size of population:300.
		• Embedding method parameters:
		  ○ *bps* = 1-8
		  ○ key *x* = 92
		  ○ key *r* = 3.64
		  ○ key *n* = 1024
		• Modified dynamic security metrics by Xin and Jiaojiao (2018). are used as comparison. Equation 3.1 represents the modified metrics:
		$dif_{sample} = \frac{1}{size_{sample} - size_{sampleused}}$ (3.1)

- Comparison at Audio Steganography Model Level

There is only one comparison at audio steganography model level. The selected existing model are discussed further in Section 4.3.2. The comparison conducted is discussed in Table 3.8.

Table 3.8: Comparison Evaluation Conducted at Cover Selection Level

Experiment	Objective	Setting
Comparison on the audio steganography model Used	To prove that a cover selection algorithm is needed to enhance the audio steganography without depending on the user's knowledge.	• size of secret message = 4KB • using setup for BCM. • cover audio database from the MCAS parameters evaluation experiments • crossover probability = 0.8 • mutation probability = 0.05. • number of generations = 20 • size of population = 300. • parameters related to the embedding method: ○ *bps* = 1-8 ○ key *x* = 1 - 99 ○ key *r* = 3.60 - 4.00 ○ key *n* = 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 819

3.3 CAS Design

This section presents the proposed solutions to the research objectives. As several issues were mentioned earlier in Chapter 2 which are the direct unauthorised detection and the existing trade-off between dynamic security and capacity. The proposed solutions to the above-mentioned issues are described as follows:

1. To avoid the direct message retrieval attack, a new embedding method must be designed to embed the secret message in a manner which does not provide an obvious pattern.
2. To satisfy the capacity and dynamic security trade-off, sequential and multiple embedding levels must be implemented as well as spreading the secret message throughout the cover audio.

In addition, the audio steganography needs the cover selection method to enhance the performance of its characteristics. The current cover selection domain focuses on the cover image steganography which left gap in the audio steganography domain to be filled. The cover selection method, in general, does not consider the trade-off between steganography characteristics. The unbalanced characteristics jeopardise the security of the created stego-file. Lastly, there are no specific dynamic security evaluation metrics for cover selection. However, there is a metric which evaluates the ratio between maximum capacity and the size of the secret message. As mentioned throughout Chapter 1 and Chapter 2, high dynamic security can be achieved by embedding maximum capacity inside the cover audio used which results in the message spreading throughout the cover audio. Following these three issues on the cover selection, the proposed solution to the above-mentioned issues is described as follows:

1. To select a cover for audio steganography, a selection rule or metric should be formulated to suitably evaluate the audio.
2. To consider and select the suitable cover, all characteristics must be evaluated simultaneously to ensure the selection process does not neglect any characteristic.
3. To evaluate dynamic security precisely, a new evaluation metric must be designed to calculate the dynamic security value accordingly.

Lastly, the additional formulation must be created to combine both solutions to make the new proposed algorithm sound. Therefore, the solution for the first part is presented in Section 3.3.1, the solution for the second part is presented in Section 3.3.2, and the solution combining both solutions 1 and 2 is presented in Section 3.3.3.

3.3.1 BCM-LSB: Formulation

The proposed solution to the above-mentioned issues related to LSB are described as follows:

3.3.1.1 Chaotic Block

The proposed method was inspired by the usage of the chaotic map, sequential embedding, and multiple embedding levels. However, given that the sequential embedding failed to have high dynamic security, a new mechanism named chaotic block was needed to improve the dynamic security and capacity simultaneously. The chaotic block was based on the idea of treating the whole cover audio as combination of smaller blocks of cover audio and the embedding process in each block is done in the chaotic manner. By implementing this idea, the message can be embedded chaotically throughout the cover audio while achieving high capacity.

As discussed previously, chaotic block allowed non-obvious embedding using the chaotic map. A chaotic map is commonly used to maintain the anonymity of information transmission (Ali et al., 2020). Previous research has utilised chaotic behaviour during the encoding procedure (Ali et al., 2018; Alwahbani & Elshoush, 2016; El-Khamy et al., 2019; Panda et al., 2015).

The logistic map is widely researched and is the simplest one-dimensional chaotic map. This research employed this map to identify the embedding location for each block, the size of the audio sample per block, and the secret message embedded size per block. Equation 3.2 is presented below (Alwahbani & Elshoush, 2016):

$$x_{n+1} = rx(1 - x_n) \quad (3.2)$$

where x_n is a real number within the range (0,1) for $n = 0, 1, 2, ...$ and r is the method parameter within the range (0,4). The value of r should exceed 3.57 to maintain the chaotic behaviour (Alwahbani & Elshoush, 2016).

The prerequisites of this mechanism included the 3 variables from the logistic map which are key x, r and n, the bit embedded per sample and the length of the secret message in binary. The process began by calculating the logistic map as in Equation 4.1. After that, the embedding location for each block can be calculated. p indicated the initial embedding location within the current block based on percentage. Using Equation 4.2, p was rounded to two decimal places to determine its final value. For example, if the size of block number, y is 100 and $p_y = 0.22$, then the first embedding initiates at sample number 22 of $block_y$. Equation 3.3 is presented below:

$$p = round(p = x, 2) \tag{3.3}$$

After the initial embedding point was determined using Equation 4.1 and 4.2, the secret message size embedded per block, s_{msg_block} can be computed. using Equation 3.4. Equation 3.4 is defined below:

$$s_{msg\_block} = floor\ (l_{bin\_msg}/n) \tag{3.4}$$

where l_{bin_msg} represents the secret message size in binary. Since there were some probabilities, there was a remainder from the calculation in Equation 3.4. Hence, the remainder was expected to be embedded in the last block. The remainder of the secret message size was calculated using Equation 3.5 which is defined below:

$$r_{bin\_msg} = l_{bin\_msg} - tm \tag{3.5}$$

where *tm* is defined using Equation 3.6.

$$tm = s_{info\_block} * n$$

(3.6)

By using r_{bin_msg} and s_{msg_block}, new secret message size embedded in the last block audio sample, $s_{last\_msg\_block}$ was computed using Equations 3.7 as below:

$$s_{last\_msg\_block} = s_{msg\_block} + r_{bin\_msg}$$

(3.7)

Next, the audio sample size per block, s_{sample_block} can be computed using Equation 3.8 as below:

$$s_{sample\_block} = floor\ (l_{audio\_sample}/n)$$

(3.8)

where l_{audio_sample} represents the audio sample size. In addition, there was a chance that a remainder was produced from the calculation Equation 3.8, which was a similar situation in Equation 3.4. Hence the remainder is expected to expand the last block audio sample. The remainder of the audio sample size can be calculated using Equation 3.9, which is defined below:

$$r_{audio\_sample} = l_{audio\_sample} - ts$$

(3.9)

where *ts* is defined using Equation 3.10.

$$ts = s_{sample\_block} * n$$

(3.10)

By using r_{audio_sample} and s_{sample_block}, new last audio sample block size, $s_{last\_sample\_block}$ was computed using Equations 3.11 as below:

$$s_{last\_sample\_block} = s_{sample\_block} + r_{audio\_sample}$$

(3.11)

The last block was extended to ensure that no audio samples were left unprocessed, as doing so would compromise the dynamic security of the stego-file. In addition, it also extended to ensure that the remainder of the secret message can be

embedded there. However, there was a case in which there were many unused audio samples in the other block while there was no audio sample left in the last block that could be used to embed the remainder of the secret message. Therefore, some modifications were needed to cater for this kind of situation. To calculate the available sample Equation 3.12 below can be used.

$$s_{sample\_available} = s_{last\_sample\_block} - s_{sample\_needed} \tag{3.12}$$

where s_{sample_needed} was calculated using Equation 3.13 below.

$$s_{sample_{needed}} = ceiling\left(\frac{s_{last_{msg_{block}}}}{bps}\right) \tag{3.13}$$

where *bps* is a bit embedded per sample. If the sample $s_{sample\_available}$ is negative value, the modification on the size of the sample block took place. The sizes of each sample block were recalculated. To determine the number of block samples which need to be readjusted their size, Equation 3.14 was used.

$$num_{block\_readjustment} = s_{sample\_available} * (-1) \tag{3.14}$$

After calculating $n_{block\_readjustment}$, Equation 3.15 was used to adjust the size of all the sample blocks.

$$\begin{aligned} s_{sample_{block}}(i=1) &= s_{sample\_block} - 1 \quad (1) \\ s_{sample_{block}}(i=2) &= s_{sample\_block} - 1 \quad (2) \\ s_{sample_{block}}(i=3) &= s_{sample\_block} - 1 \quad (3) \\ &\vdots \\ s_{sample_{block}}(i=j) &= s_{sample\_block} - 1 \quad (j) \end{aligned} \tag{3.15}$$

where j represents $num_{block\_readjustment}$. After completing the block size readjustment, $s_{last\_sample\_block}$ was recalculated back to update the current size of the

last block. $s_{last\_sample\_block}$ was calculated with the updated equation which was shown in Equation 3.16.

$$s_{last_{sample_{block}}} = l_{audio_{sample}} - \left(\sum_{i=1}^{n-1} s_{sample_{block_i}}\right) \quad (3.16)$$

After finalising the size of each audio sample block, the initial embedding point of each block can be calculated using Equation 3.17 as follows:

$$index_y = \sum_{i=1}^{y-1} s_{sample\_block_i} + initialembeddingpoint_y \quad (3.17)$$

where $initialembeddingpoint_y$ was calculated using Equation 3.18.

$$initialembeddingpoint_y = s_{sample\_block_y} * p \quad (3.18)$$

In summary, these series of equations were utilised to determine the embedding location for each block, the size of the audio sample per block, and the size of the secret message embedded per block, which was essential during the embedding process.

3.3.1.2 Parameters Setting in Chaotic Block

This section defines those factors that affect the BCM-LSB method. An additional factor, embedding behaviour, which was identified in the previous section, is defined, and implemented for BCM-LSB. This section also presents the algorithms integrated into BCM-LSB to improve overall performance.

- **BCM-LSB variables**

The performance of the BCM-LSB method is mainly determined by the bit embedded per sample and the number of blocks.

  - Bit embedded per sample

Bit embedded per sample, or *bps*, is the variable that has the most significant impact on the performance of BCM-LSB in terms of robustness, imperceptibility, and capacity. *bps* is used and manipulated to give a variety of characteristics' performances. A high value of *bps* can lead to high robustness and capacity, while a low value of *bps* can lead to high imperceptibility. As mentioned previously, as there is always a trade-off between these three (3) characteristics, both high and low value of *bps* is acceptable to be used as there is no pinpoint value that managed to improve these three (3) characteristics simultaneously. The highest *bps* used for BCM-LSB was eighth (8) while the lowest was one (1). These values were common and were reported in previous researches such as (Ahmed et al., 2010; Bender & Gruhl, 1996; Indrayani, 2020).

- Number of blocks

The number of blocks represented by key *n* is special to the chaotic block. It heavily impacted the dynamic security characteristic of BCM-LSB. The optimal number of blocks depends heavily on the size of the secret message and the size of the audio sample. As an example, the bigger size of the cover audio needed a higher number of sample audio blocks to keep its block size small for distributing the secret message without a big gap throughout the cover audio which resulted in low dynamic security. However, on the other hand, if the number of blocks is too high up until the size of the block is small enough, the secret message also cannot be distributed properly throughout the entire cover audio.

- Size of cover audio

The size of the cover audio is represented by the number of audio samples. It impacts the total number of audio samples per block as BCM-LSB divided the cover audio as equal as possible. The higher the size of the cover audio, the higher number of audio samples per block. Hence, more message is needed to satisfy the ratio of the secret message to the size of the audio samples.

- Size of secret message

  The size of the secret message is represented by the size of the secret message in binary. Similar to size of the cover audio, the secret message is divided as equally as possible and then embedded according to the block of the audio sample. Higher the size of the secret message, the higher the secret message embedded per block.

- **Embedding behaviour**

As explained previously, embedding behaviour can have a significant impact on performance. The sequential embedding behaviour is adopted in BCM-LSB because it has the highest capacity. However, sequential embedding has a dynamic security disadvantage and has been adapted using chaotic block embedding.

3.3.1.3 Embedding Keys and Security

Steganography's security is based on Kerckhoff's principle, which considers that the attacker has comprehensive knowledge of the embedding process but lacks the embedding key (Fridrich & Goljan, 2002). Therefore, embedding keys are required for every effective steganographic technique. The subsequent embedding keys are used for BCM-LSB:

1. Key *x*, *r* and *n*: These three key combinations are important to determine the initial embedding point. An attacker with prior knowledge of this set of keys can use his knowledge to find the difference between the clean and the stego parts of the signal.
2. *bps*: Even with the assumption that the attacker has comprehensive knowledge of the embedding, it is difficult to retrieve the secret message as *bps* is also impacted the number of samples used for embedding, which makes it vague to do the direct retrieval attack as there is also the unused sample of the sample block.
3. Message size: The number of samples used per block and the number of messages embedded per block can be retrieved easily hence compromising the chaotic block entirely if the message size is known to the attacker. Therefore, the message size is kept as a key.

Like many existing audio steganography methods, this proposed method assumes the usage of the shared key.

3.3.1.4 The Rounding Mechanism

Capacity is one of the three most important characteristics of audio steganography. Without high capacity, the sender has a problem sending a huge secret message in one cover. He needs to divide the secret message into several parts and embed it inside several cover audio. Although BCM-LSB has a high capacity in general as it implements sequential embedding behaviour, there are some cases that it cannot manifest its capacity full potential. There is always a loss in capacity in BCM-LSB due to the usage of chaotic blocks. In the chaotic block, the initial embedding for each block is always depends on the key *x*, *r* and *n*. In all cases except the value of *p* is 0.01 which

means that the embedding process starts at the first index sample of the block, there is loss capacity in whatever unused sample is at the beginning of the block. Therefore, the rounding mechanism is implemented if needed. The rounding mechanism is illustrated in Figure 3.2.

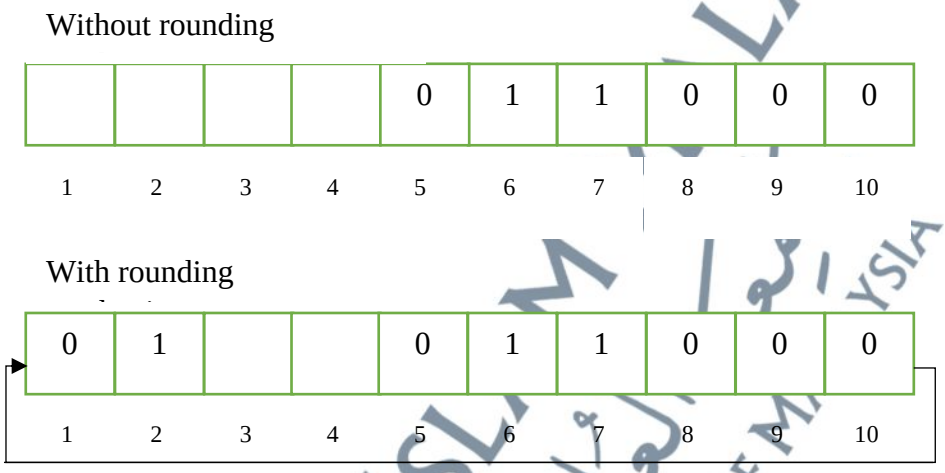

Figure 3.2: Visual representation of the rounding mechanism

Based on Figure 3.2, 1 byte ASCII character ‘a’ was encoded in the 10 audio samples. The ASCII character ‘a’ had the 01100001 binary data value. Assuming the p is 0.5, hence the embedding process starts at index 5. Since the embedding process needs eight (8) samples to complete the embedding process of the letter ‘a’, data cannot fully embed, which reduces the capacity despite having unused samples. However, by implementing a rounding mechanism, when the secret embedding process arrived at the last index, it went back to the beginning index to continue the embedding process. Therefore, by implementing a rounding mechanism, the capacity of BCM-LSB can achieve its full potential.

3.3.1.5 The General BCM-LSB Flowchart

This subsection presents an abstract flowchart of BCM-LSB, as illustrated in Figure 3.3. In this figure, the creation of a chaotic block starts after the sender inserts all the input needed. Then, the creation of a chaotic block took place. Lastly, the encoding process begins to produce a stego-file by using the variables produced during the creation of the chaotic block stage.

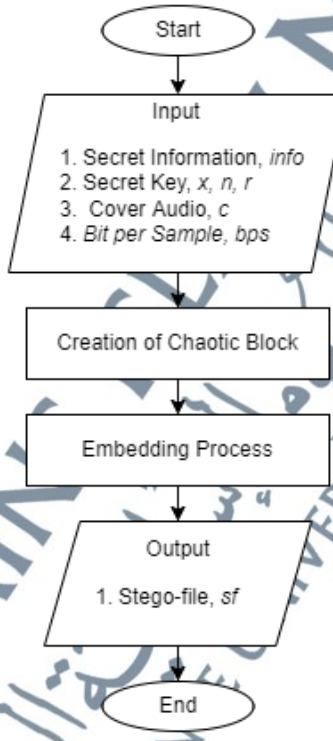

Figure 3.3: The general BCM-LSB Flowchart

3.3.2 MCAS: Formulation

This section presents the proposed solution to the research objective. Based on the author's knowledge after conducting the LR on the cover selection, there has been no audio cover selection proposed in the past. Most of the cover selections were focused on the selection of the image cover. They introduced the selection rules based on the metrics that can be used to evaluate the cover objectively. With the same logic applied, the proposed solution must be able to evaluate the cover audio objectively. As

mentioned in Chapter 2, the trade-offs between characteristics were not considered by the current cover selection methods. Hence, the newly proposed solution must consider the characteristic simultaneously. Lastly, the current dynamic security evaluation metric failed to consider other aspects related to dynamic security except for the ratio of the size of the secret message embedded to the size of cover audio. Therefore, the proposed solutions to the above-mentioned issues are described as follows:

1. To introduce cover audio selection, a metric which can evaluate the audio steganography characteristics must be formulated.
2. To consider the trade-off between the audio steganography characteristics, a new cover selection method must be designed to evaluate the characteristics simultaneously.
3. To introduce a new dynamic security evaluation metric, other aspect including the ratio of the secret message size embedded to the size of the cover audio, must be considered.

Therefore, the formulation of these solution is described and explain in the next subsection.

3.3.2.1 Selection Metric

As discussed previously, all the characteristics must be evaluated to ensure that the stego-file produced has a well-rounded performance and not just skewed heavily to one characteristic. Therefore, a set of evaluation metrics that considered capacity, imperceptibility and robustness which can be used in audio steganography need to be formulated. Basically, there are three objectives which need to be fulfilled which are:

1. Ensuring the capacity of cover audio is enough to embed all the secret message.
2. Maximising the imperceptibility of the selected cover audio.

3. Maximising the robustness characteristic of the embedding method.

The capacity of the cover audio depends on the audio size and the embedding method itself. However, since CAS used BCM-LSB for the embedding method which implemented sequential embedding, Capacity, *C*, can be modelled as Equation 3.19:

$$C = bps * sr * d \tag{3.19}$$

where *bps* is the bit embedded per sample, *d* refers to the duration of cover audio in second and *sr* represents the sample rate of the cover audio.

Next, imperceptibility can be represented by many metrics. However, SNR was used as the metric to quantify the imperceptibility characteristic. The SNR formulation can be modelled as Equation 2.2.

Lastly, robustness refers to the ability of the embedded message to withstand attacks. Since there are too many attacks which can be implemented to destroy the secret message, CAS only focused on the AWGN attack. AWGN attack can be done at various levels. However, a higher level of AWGN attack would destroy the secret message embedded by any embedding method. Therefore, a single point low level of AWGN attack was introduced to the stego-file to ensure there is a differentiation between low and high robust stego-file. The robustness of the stego-file can be tested using the Bit error rate (BER), which calculates the error from the secret message retrieval and evaluates the ratio of the number of embedded message bits that result in an error during the retrieval process to the overall size of the secret message. BER was calculated using Equation 2.5. Therefore, by implementing this formulation, which is suitable for measuring the audio steganography characteristics, the cover selection for the audio can be implemented.

3.3.2.2 Trade-off Consideration

Although existing cover selection methods proposed the robustness and imperceptibility metrics for the selection criteria, these methods provide a distinct list of cover depending on either robustness or imperceptibility. To select the cover which produced a well-rounded performance, all the characteristics measured must be considered simultaneously.

In several scientific domains, including engineering, economics, and logistics, where optimal decisions must be made in the presence of trade-offs between two or more competing objectives, multi-objective optimisation has been used. There is no one solution that concurrently maximises all audio steganography characteristics for the multi-objective optimisation issue that attempts to maximise all its characteristics. Therefore, MCAS selects a solution which has better characteristics. The solution selected was called a non-dominated solution. The concept of dominance was applied to this to multi-objective problems to compare two solution candidates, $solution_1$ and $solution_2$ and determine if one solution dominates the other one. In another word, dominance is a categorisation approach for solutions that assures the selection of the optimal solution. The $solution_1$ is considered dominates $solution_2$ if: 1) $solution_1$ is not worse than $solution_2$ for all the objectives, and 2) $solution_1$ is strictly better than $solution_2$ for at least one objective. By implementing this concept of dominance, all the characteristics can be measured simultaneously to determine each of the solutions' ranking. One of the methods that optimises the selection based on this multi-objective criteria is the Multi-Objective Evolutionary Algorithm (MOEA) technique. Hence, this research implemented NSGA-II, which is a method under MOEA. NSGA-II is implemented due to its low computational time, and it is effective on smaller multi-

objective optimisation (two or three objectives). The flowchart of NSGA-II is presented in Figure 3.4.

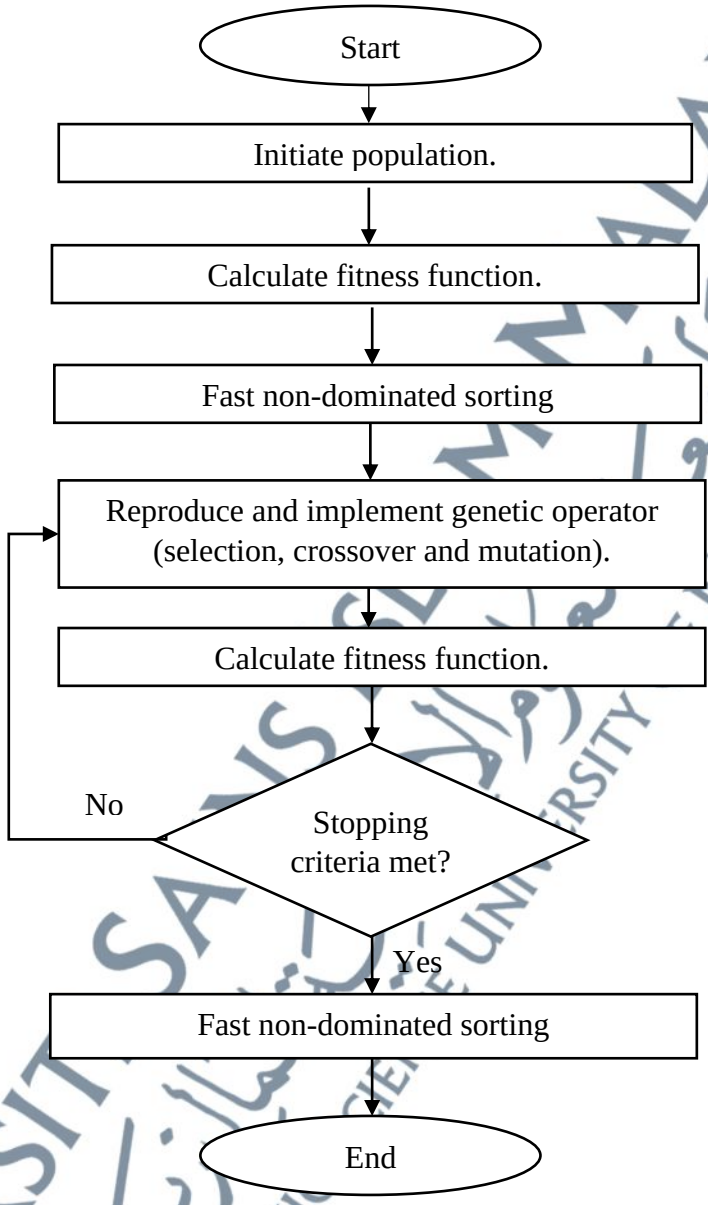

Figure 3.4: Flowchart of NSGA-II

Based on Figure 3.4, the NSGA-II started by initiating the population where all the individual inside the population represents the actual solution for the problem multi-objective optimisation problem. Next, the fitness functions for each objective were calculated. Then fast non-dominated sorting was conducted. After the desired solution was selected, all the solutions went through the crossover and mutation process. New

solutions produced are sorted alongside the old solution. These processes were repeated until the stopping criteria were fulfilled. This NSGA-II is personalised and tuned to become MCAS which is explained further in Chapter 4.

3.3.2.3 Dynamic Security Evaluation Metric

Dynamic security often is measured using two types of tests which are the Difference Signal Visual Test and the Seg-SNR Spike Test. Although both are commonly used visual results to determine the dynamic security characteristic level, Seg-SNR Spike Test, which represented through the graph, is easier to measure objectively. To get familiarised with the Seg-SNR Spike Test, another graph which represents the Seg-SNR spike Test is presented in Figure 3.5.

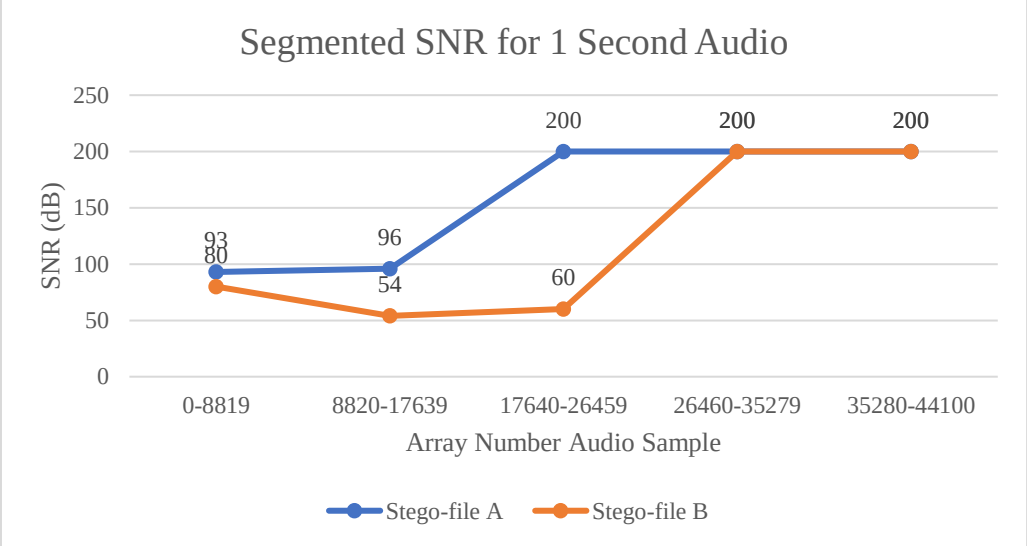

Figure 3.5: Seg-SNR Spike Test Example

A spike presented in Figure 3.5 indicates that there was a difference between the modified and clean audio samples (The seg-SNR value which has an infinity value was replaced by 200 dB for the sake of visual representation). The difference often resulted

from the size of the embedded secret message not matching the maximum capacity of cover audio.

There is an existing cover selection method which uses a ratio between the size of the secret message with the maximum capacity of the cover audio to select the cover audio. Although this idea is quite accurate, however, it does not represent the whole situation that happened. It is understandable that the spike is created due to the difference between the clean and modified audio sample however it also common that there are slight up and down trends throughout the Seg-SNR graph. Generally, to maintain high dynamic security level, the graph must be stable throughout each segment. Therefore, two situations were derived to determine the dynamic security level.

1. The increased number of segments that has value infinity indicates that the stego-file has lower quality.
2. If the number of segments that has value infinity are the same, high difference between the segment, indicates that the stego-file has lower quality.

Based on these two situations a new dynamic security evaluation metric was derived. Dynamic security, *DynSec* can be formulated and defined using Equation 3.20 until Equation 3.23.

$$DynSec = nis - sd \tag{3.20}$$

where

$$nis = (number\_segment - infinity\_region) * 10000 \tag{3.21}$$

$$sd = \sum_1^n segment_i - a \tag{3.22}$$

where

$$a = \sum_{1}^{n} (\frac{segment_i}{n})$$

(3.23)

The *nis* represents the non-infinity value segment while *sd* represents the total segment difference. *nis* is introduced to determine situation one (1) above while *sd* is used to determine situation two (2) above. 10000 is selected as the fixed value. Sometimes *sd* can be high enough hence a higher fixed value such as 10000 is appropriate to ensure *nis* value does not lose its significance to the *sd*. By implementing *DynSec,* the dynamic security can be calculated more effectively hence giving a better description of the dynamic security level during the selection process.

3.3.2.4 The General MCAS Flowchart

This subsection presents an abstract flowchart of MCAS as illustrated in Figure 3.6. Based on Figure 3.6, cover selection started retrieving the cover audio from the audio database. From there, the cover selection evaluated the audio repeatedly until the conditions were met. Then, it produced a set of solutions that satisfy the characteristic of audio steganography.

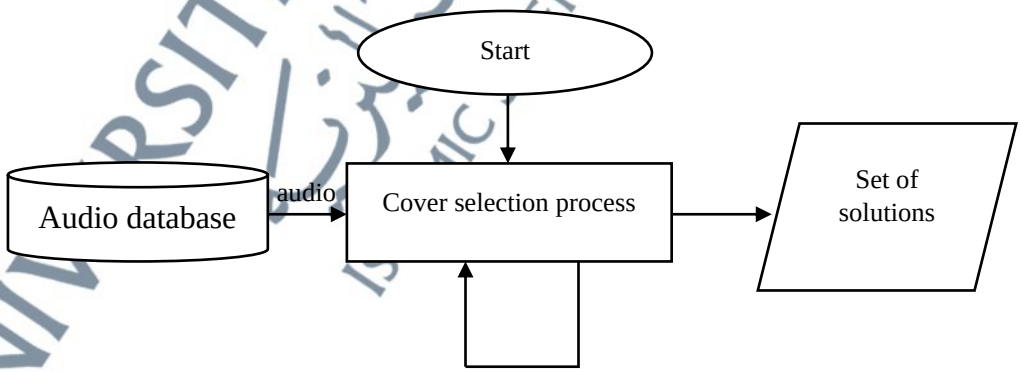

Figure 3.6: General flowchart of MCAS

3.3.3 CAS: Formulation

This section presents the proposed solution which combine BCM-LSB and the MCAS. Figure 3.7 presented the general flowchart of CAS.

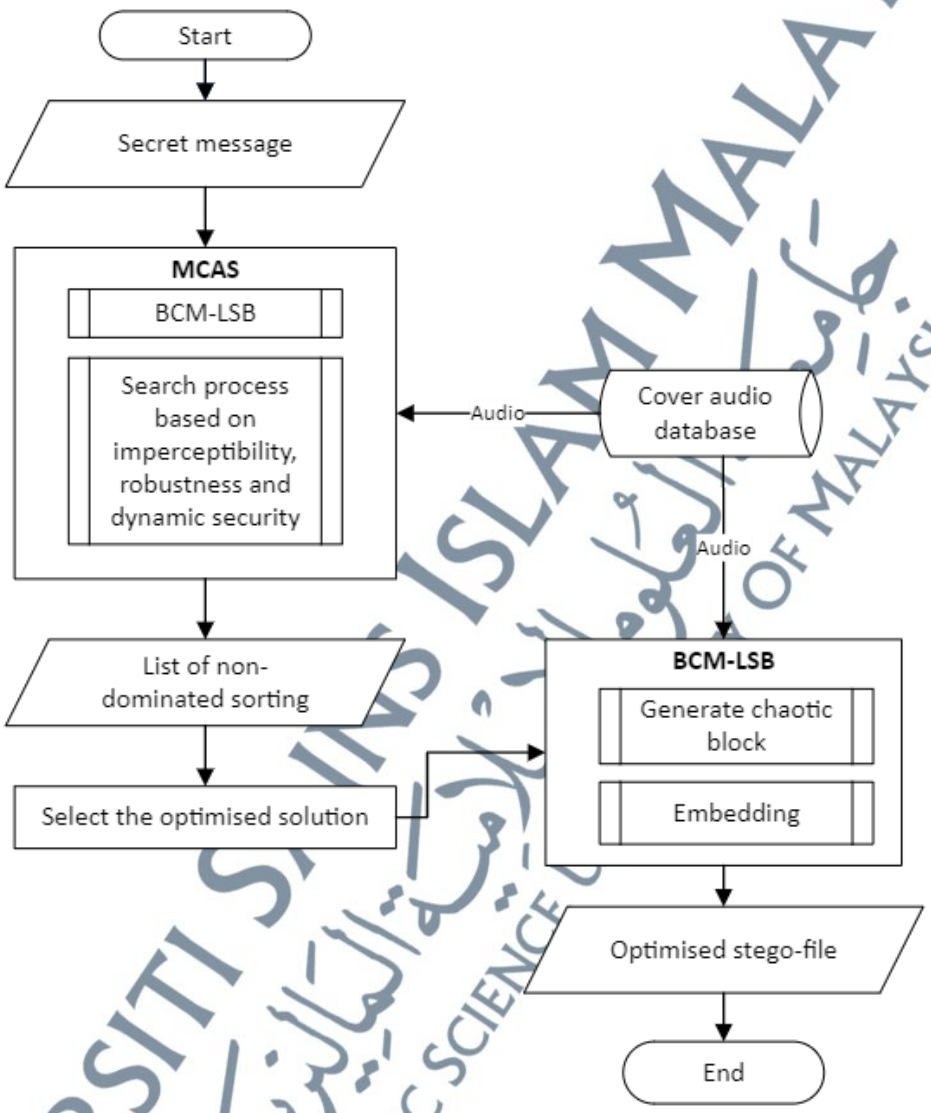

Figure 3.7: General Flowchart of CAS

Since MCAS was implemented to select the cover audio used by BCM-LSB, it was expected to use the parameters that affected the performance of BCM-LSB alongside the index cover audio in the database as a representative of the cover audio itself. Based on Figure 3.4, the NSGA-II was started by initialising the population. Therefore, to personalise the NSGA-II into MCAS, the individual inside the population

is customized. As a recap, BCM-LSB needs a set of keys x, r and n alongside the *bps* to initiate the embedding process. Therefore, these parameters were set as the chromosomes inside these individuals alongside the index cover audio inside the database. Therefore, the chromosome of the individual inside the population initiated by the MCAS is described in Figure 3.8.

Index audio	*bps*	*Key r*	*Key x*	*Key n*

Figure 3.8: Example of parameters used in chromosome forms

Next, the BCM-LSB only needed one cover audio to hide the secret message. However, the MCAS, which was inspired by the NSGA-II, usually came out with a set of solutions, all of which were not dominated by the others. Since the solutions were not dominated by others, a new formulation was needed for selecting just one solution to be used by the BCM-LSB. This can be achieved by normalising each value from three fitness functions which were values from the imperceptibility, robustness, and dynamic security characteristics. Then, they were summed together with a weight of 0.3, 0.34 and 0.33 respectively. The solution with the highest combined value of the normalized objective function values was selected as the best. The best solution can be calculated using Equation 3.6 until Equation 3.12. However, to avoid an overloading situation, this research automatically set all the imperceptibility, robustness, and dynamic security characteristics to the lowest value. This setup was made to indicate the maximum capacity is lower compared to the size of the secret message; hence this audio should be avoided. By setup the individual chromosome with the parameters used in BCM-LSB alongside with cover audio index and providing the formulation of the best solution, the combination of the BCM-LSB and MCAS can be done feasibly.

3.4 Chapter Summary

This chapter presents the research stage and CAS design. The research stage explains how a set of tasks fulfils each research objective presented in Chapter 1. The proposed design discusses and justifies the design of the proposed solution to achieve the research objectives.