

## A Machine Learning based Framework for Parameter based Multi-Objective Optimisation of Video CODECs

Maryam Al-Barwani\*, Eran A. Edirisinghe

Department of Computer Science, Loughborough University, Loughborough, UK

### ARTICLE INFO

Article history:

Received: 03 June, 2017

Accepted: 15 July, 2017

Online: 15 August, 2017

Keywords:

Video Codec

Encoder/Decoder

H.264/MPEG4-AVC

High Efficiency Video Coding

Multi-objective Optimisation

### ABSTRACT

All multimedia devices now incorporate video CODECs that comply with international video coding standards such as H.264 / MPEG4-AVC and the new High Efficiency Video Coding Standard (HEVC), otherwise known as H.265. Although the standard CODECs have been designed to include algorithms with optimal efficiency, a large number of coding parameters can be used to fine-tune their operation, within known constraints of for example, available computational power, bandwidth, energy consumption, etc. With the large number of such parameters involved, determining which parameters will play a significant role in providing optimal quality of service within given constraints is a further challenge that needs to be met. We propose a framework that uses machine learning algorithms to model the performance of a video CODEC based on the significant coding parameters. We define objective functions that can be used to model the video quality as Peak Signal-to-Noise Ratio (PSNR), CPU time utilization and Bit-Rate. We show that these objective functions can be practically utilised in video Encoder designs, in particular in their performance optimisation within given constraints. A Multi-objective Optimisation framework based on Genetic Algorithms is thus proposed to optimise the performance of a video codec. The framework is designed to jointly minimize the complexity, Bit-rate and to maximize the quality of the compressed video stream.

### 1. Introduction

This paper is an extension of work originally published in Future Technologies Conference FTC 2016 San Francisco, United States [1].

Applications that benefit from accurate video capture, efficient representation and coding, error-free transmission and subjectively optimised display, have been growing over the years due to the availability of higher network bandwidth, faster processor speed and advanced capture and display technologies. Recent studies have shown that coded video data is contributing a the major part of consumer internet traffic with a predicted share of 90% by 2019.

Some of the most extensively used applications include real-time video conferencing, video streaming over broadband networks and digital TV broadcasting. Most current mobile hand-held devices come equipped with a video camera that is able to capture and encode a video stream in a standard format. These devices also include video players, which can decode and play back video. All the above developments continuously demand

more efficient video coding algorithms that are able to reduce the bitrate without sacrificing video quality or to enable the increase of video resolution, without increasing the bitrate. High Efficiency Video Coding (HEVC) also known as H.265 is the most recent answer to this consumer demand, demand which supersedes the more widely used video coding standards such as MPEG-2 and H.264.

All advanced video CODECs have many parameters that can be used to control their operational characteristics, both at the encoder and decoder ends, enabling the possibility of fine tuning their operation for maximum efficiency within environments and application scenarios that are bound by various constraints. For example the available bandwidth will have an upper limit, the network will be subjected to delays and the decoder/display unit may have limitations in processing and display capabilities. Yet the encoder, transmission and decoder have many parameters that can be adjusted for them to be efficiently operational under the above mentioned constraints. Identifying the values of these parameters that results in the CODECs optimal performance under given constraints remains an open research problem of vital importance.

\*Corresponding author: Maryam Al-Barwani, Department of Computer Science, Loughborough University, UK | Email: m.al-barwani@lboro.ac.uk

The first step of parameter based optimisation of a video CODEC is the identification of the coding parameters that have a significant impact on its key properties, such as, bandwidth, image/video quality, and CPU cycles. Although an experienced user of a video CODEC can guess these parameters with some accuracy when the content of the video is known, a formal scientific approach is needed to accurately decide the parameter set, with minimum subjective error. Having obtained these parameters, it is then possible to model the key properties of the video CODEC described above based on the significant parameters. These models can then be used to optimise the performance of the video codec when operated under practical constraints, thus making the parameter based characterisation and modelling practically useful.

In this paper we propose a framework that is based on the fundamentals of machine learning that can be used to scientifically determine the significant coding parameters of a video CODEC. These parameters are then used to model the operational behaviour of the video CODEC for which machine learning algorithms are further utilised. We also show that this model can be used to establish the foundations of a multi-objective optimisation framework. Optimisations algorithms are widely used to solve many difficult optimisation problems in other research areas. Although the experiments conducted are limited to H.264 and H.265 standards, the proposed framework can be used in relation to any video coding standard.

For clarity of presentation, the remainder of this paper is structured as follows: Related work and the background of H.264 & H.265 video coding is introduced in section 2. Section 3 presents the proposed framework for performance modelling and the experiments conducted for establishing the framework. Section 4 presents a comprehensive analysis of the results of the performance modelling of H.264 along with analysis of Optimisation. Section 5 presents the results and analysis of H.265 video codec and Optimisation stages carried out using a Matlab based implementation. Finally, section 6 concludes the paper.

## 2. Related Work

A significant amount of research has been conducted in the past and presented in literature on the optimization of video coding/compression algorithms. Parameter-based optimization focuses on the selection of the optimal set of coding parameters that will influence the optimal overall performance of the video CODEC, given operational constraints such as bandwidth, distortion, and CPU.

In [2], a joint power-rate-distortion (P-R-D) framework, for the analysis, control and the optimization of the behaviour of rate-distortion (R-D) algorithms of a wireless video communication system, under constraints of energy consumption, was proposed.

In [3] it was shown that the approach presented in [4] cannot be easily extended to other video encoders. It presented a novel power-rate-distortion (P-R-D) optimization algorithm that can be used to minimize energy consumption of delay tolerant applications in portable video communication.

### 2.1. H.264 video codec.

A joint power-distortion model was presented in [5] and was analysed under two constraints, namely, power consumption and video quality. The approach jointly considered the power consumption and video quality and analysed the two problems

within a uniform optimization framework. The work presented in [6], proposed a power-rate-distortion (P-R-D) model of a video encoding system to maximize its operational lifetime. In [7] a novel equation for the prediction of distortion was proposed that was used for the optimization of quantization parameter (QP) selection. An improvement of image quality as compared to the standard rate control algorithm of the H.264 reference software JM15.01 was recorded.

A large number of coding parameters poses a practical operational problem of how to best select the set of parameters that will ensure the CODECs optimal performance, especially under multiple operational constraints. In [8], the challenge to determine H.264 parameter settings that have low complexity but still offer high video quality was investigated.

An improvement to the work presented in [8] was proposed by the same authors in [4], in which two further algorithms for finding additional parameter settings for the GBFOS-basic algorithm were presented. However, a significant constraint of the work of [7,3] is its limited use only within two constraints namely, complexity and video quality.

The author in [9] presented a detailed study of the importance of a multi-objective optimisation framework and the approach presented in [10] as a solution. The presented framework focused on the development of a joint complexity-rate distortion (C-M-R-D) optimization framework for a H.264 video CODEC, which could be extended to cover any number of constraints and to be used within any type of video CODEC.

### 2.2. H.265 video codec.

High Efficiency Video Coding (HEVC) is the next generation video coding standard being developed, the newest video coding standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group [11].

In [12] it was shown that HEVC provides significantly improved compression performance, i.e. an approximately 50% reduced bit rate as compared to the best existing video coding standards, under the same visual quality. The paper proposed a hardware-friendly method for RDO of HEVC intra coding. The results of the study showed that the proposed RD cost function provides 85.8% area reduction and 1260% throughput improvement in hardware design, with slight loss of bitrate and PSNR, which is suitable for real-time encoder applications.

A performance comparison of H.265, VP9 and H.264 encoders was presented in [13]. According to the experimental results, obtained for a whole test set of video sequences, by using similar encoding configurations for all three examined representative encoders, H.265/MPEG-HEVC was shown to provide significant average bit-rate savings of 43.3% and 39.3% relative to VP9 and H.264/MPEG-AVC, respectively.

In [14] it was shown that for resolutions of up to HD (1920x1080), code optimizations including heavy use of single instruction multiple-data (SIMD) instructions are sufficient to achieve HEVC real-time software decoding. It was further shown that when it came to decoding UHD video (3840x2160), single threaded execution with code optimization was not enough.

To improve the compression performance of current video coding standards by 50%, especially when it comes to transmitting high resolution video like 4K over the internet or in broadcast, the

50% bitrate reduction is essential. [15] shows that real-time decoding of 4K video with a frame-level parallel decoding approach using four desktop CPU cores is feasible.

Emerging video compression standard H.265/HEVC provides up to 2 times better compression efficiency compared to H.264/AVC standard. Iterative intra prediction search in [16] was proposed for the H.265/HEVC encoder to reduce the number of prediction modes for estimation: about 40% encoding time reduction for HM 10.1 intra-only coding with negligible bitrate increase and PSNR quality degradation. Additional speed-up techniques, including fast prediction error estimation, were offered.

### 3. Proposed Framework for Performance Modelling

The proposed framework for a MOO Multi-Objective Optimisation [2] is developed to determine the optimum coding parameters for a H.265 video CODEC, when working under multiple constraints as shown in Figure 1. The MOO framework is intended to minimize the complexity (CPU utilisation), bit-rate and to maximize the quality of the compressed video stream. The MOO framework proposed is accomplished by following the steps below.

- 1) Profiling experiments on the encoder and decoder were carried out to determine the coding parameters that have a significant impact on each of the objectives/constraints related to rate, distortion and CPU utilization. This was achieved by measuring the impact of each parameter (while being varied) on each of the above aspects.
- 2) Developing the objective function for each objective/constraint, based on the above significant parameters, by using a suitable regression procedure.
- 3) These objective functions can then be used within a genetic algorithm (GA) based multi-objective optimization framework to determine optimal parameter values.

In a practical multimedia application scenario a device captures a video, encodes it and transmits it via a network to another device that decodes and displays the content to a viewer. Assuming that the network has bandwidth constraints and the device in which the encoder is placed has compute power constraints and the potential viewers of content may demand at least meeting minimal quality levels, a situation occurs in which the proposed MOO framework can be used may arise.

The significant number of encoder parameters that control the encoder's bit rate, quality and computational power requirements can be selected, to ensure the encoder performance is optimal, under the given multiple constraints. However this requires the modelling of the encoder's bit-rate, quality and CPU utilisation, based on the large number of selectable encoder parameters. If mathematical objective functions can be derived for each of the above, a standard approach to optimisation can be used. Deriving objective functions, for example using mathematical regression, will need the determination of the significant coding parameters, the key focus of the research presented below. The same explanation can be applied to the selection of decoder parameters that results in optimal decoder performance. Within the research context of this paper, we assume that the data transmission network is assumed to be perfect, i.e. no delays, no bit losses, no errors etc. Therefore the bit stream generated by the encoder is transmitted

without any loss or alteration to the decoder, in real-time. The following section proposes the experimental process adopted to determine the significant coding parameters for both the encoder and decoder.

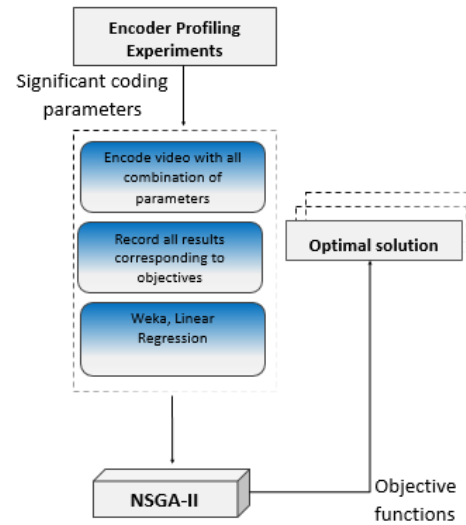


Figure 1: Proposed Multi-objective Optimisation framework.

### 4. Machine Learning based Framework for the analysis of significant coding parameters of H.264.

#### 4.1. Profiling Experiments/ Determining the Significant Coding Parameters of H.264.

This experiment was carried out using a configuration file of the encoder software named JM (Joint Model) Reference software version 18.6 [17]. Six video sequences were encoded and decoded in the H.264 encoder by using a configuration file to set parameters of the mentioned video sequences. In this analysis, six QCIF video sequences were chosen for the experiment with a QCIF Quarter Common Intermediate Format, a videoconferencing format that specifies data rates of 30 frames per second (fps), with each frame containing 144 lines and 176 pixels per line. This is one fourth the resolution of Full CIF.

The first 30 frames were encoded for each video. Starting with Claire, a video sequences have a simple motion foreground and a non-moving area in the background, with a news presenter is talking by moving her head and eyes and mouth slowly. Coastguard video sequences have fast movement; on both regions foreground and background simultaneously, a boat is moving fast, waving water and a second boat will come into the scene within the final frames. A football video sequence has complicated fast motion with many players moving very quickly at the same time. Foreman video has a slight movement in the background; the Foreman is talking and his head is moving quite rapidly.

Mobile video sequences has a fast background and foreground movement; simultaneously, with a calendar moving upward, a ball and train are moving towards the left side with background. Tennis video sequences have a slow motion foreground and gentle movement in the background: the player's hand is moving and bouncing the ball and slightly zooming out of motion in the last few frames. The above six videos have different properties and motions as shown in Table 1.

Table 2 also tabulates the sample values used in our experiments for each parameter from within their corresponding value ranges. The control variable Intra-Period, can take values: 0 (meaning that the first frame is coded as an I-frame and subsequent frames are coded as P-frames), 5 and 8. The search window size is assumed to take either of the two values 16 or 32. The control variable Quantization Parameter (QP) is assumed to take two possible values 17 or 49. The number of reference frames (NRF) can take values 2, 5 and 8.

Table 1: H.264 Selected Video Sequences.

|  |  |
|--|--|
|   |   |
| <b>Claire 494 frames</b>   | <b>Coastguard 300 frames</b>   |
|   |   |
| <b>Football 260 frames</b>   | <b>Foreman 300 frames</b>  |
|  |  |
| <b>Mobile 300 frames</b>   | <b>Tennis 150 frames</b>   |

Table 2: Significant parameters and value used

| Variables | Parameters               | Values Range | Variable Type |
|-----------|--------------------------|--------------|---------------|
| IP= x(1)  | Intra-Period             | (0,5,8)      | Numeric       |
| SR= x(2)  | Search-Range             | (16, 32)     | Numeric       |
| QP= x(3)  | Quantization Parameter   | (17,49)      | Numeric       |
| NRF= x(4) | Number-Reference- Frames | (2,5,8)      | Numeric       |

The computer chosen for the experiment has the following specification: HP Intel, Microsoft Windows 8.1 (64-bit), Intel® Core™ i5 CPU 4200Y @ 1.40 GHz, 4.00GB RAM.

The total encoding CPU time for each video sequence was recorded using Intel VTune Amplifier XE [18]. The following are the parameters used, with each video containing  $3*2*2*3 = 36$  Total Number of Instances. Table 3 presents 12 Instances out of the 36 Instances.

In each experiment one parameter will be changed while the rest of the parameters are fixed. This will help to observe the effect all parameters have on each Objective as shown in Table 3. The selected values of each coding parameter, the distortion (as PSNR is measured in decibels (dB)), bit-rate in kbps and the encoding time in seconds were recorded. Subsequently, the results were fed into the Linear Regression Analysis tool of WEKA [19] to generate the linear regression function for each objective.

Table 3: selected set of parameters for foreman sequence.

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | Bitrate | PSNR   | Encoding Total Time |
|-------|-------|-------|-------|---------|--------|---------------------|
| 0     | 16    | 17    | 2     | 44.606  | 547.62 | 61.627              |
| 0     | 16    | 17    | 5     | 44.635  | 473.74 | 75.058              |
| 0     | 16    | 17    | 8     | 44.636  | 471.7  | 84.624              |
| 0     | 32    | 49    | 2     | 22.871  | 9.98   | 41.24               |
| 0     | 32    | 49    | 5     | 23.449  | 9.21   | 57.3                |
| 0     | 32    | 49    | 8     | 23.405  | 9.38   | 71.303              |
| 5     | 16    | 17    | 2     | 45.751  | 819.97 | 55.727              |
| 5     | 16    | 17    | 5     | 45.819  | 712.66 | 69.253              |
| 5     | 16    | 17    | 8     | 45.822  | 714.2  | 75.729              |
| 8     | 16    | 17    | 2     | 44.818  | 618.74 | 57.825              |
| 8     | 16    | 17    | 5     | 44.906  | 544.33 | 69.269              |
| 8     | 16    | 17    | 8     | 44.899  | 540.87 | 77.812              |

Based on the output of the linear regression, only the functions of three objectives of the Foreman video sequences are given in equation (1), (2) and (3). Following are the obtained models for each video sequence, with  $f(1)$  representing PSNR,  $f(2)$  rate and  $f(3)$  encoding-time.  $x_5$ , representing PSNR as parameter and  $x_6$ , representing Bit-rate.

#### Foreman Linear Regression Model

$$f(1)_{PSNR} = -0.5926 * x(3) + 0.0798 * x(4) + 0.0046 x(6) + 52.0225 \quad (1)$$

$$f(2)_{Bitrate} = 6.8133 * x(1) - 18.5023 * x(3) + 890.0319 \quad (2)$$

$$f(3)_{Enc\_Time} = -0.5347 * x(1) + 0.1297 * x(2) - 0.5646 * x(3) + 4.0201 * x(4) + 60.362 \quad (3)$$

#### 4.2. H.264 Encoder Analysis

Experimental analysis was conducted separately for the encoder and decoder. Table 4 tabulates the correlation coefficients of the objective functions. They range between 0-1. In analysing the objective functions above, higher positive coefficients of coding parameters indicate higher positive dependency and higher negative coefficients represent higher negative dependency. If a certain parameter is not present in the objective function that means that the objective is independent of that parameter.

Table 4: Encoder Correlation Coefficient.

| Video      | PSNR   | Bit-rate | CPU    |
|------------|--------|----------|--------|
| Claire     | 1      | 0.9424   | 0.9460 |
| Coastguard | 0.9865 | 0.9865   | 0.8917 |
| Football   | 0.9997 | 0.9967   | 0.9849 |
| Foreman    | 0.9998 | 0.9678   | 0.9746 |
| Mobile     | 0.9999 | 0.9809   | 0.9588 |
| Tennis     | 0.9998 | 0.9757   | 0.9883 |

A careful analysis of the coding parameters that have non-zero weighting factors in the objective functions obtained and a comparison of relative magnitudes of the coefficients can lead to a direct correspondence with the properties of the video. For example, the analysis of the linear regression equations obtained for the Foreman video sequence identifies all four parameters to have significant impact on CPU utilisation (Encoding time) as in equation (3), namely:

- IntraPeriod
- Searchrange
- Quantization parameter
- NumberReferenceFrames

The most significant impact on CPU utilisation is the number of reference frames. This is expected due to the need to repeat the motion estimation process when NRF increases. The next significant impact is from the Quantization parameter. The impact from Search Range (SR) and Intra Period (IP) is relatively insignificant. For most videos with fast movement of objects (i.e. Football and Mobile) there is no impact from the Search Range. This is true given the fact that for videos with fast moving objects, best matches will not be found quickly, i.e. without having to scan the entire video.

For the same video, the following parameters were identified to have a significant impact on Bit-rate as in equation (2).

- IntraPeriod
- Quantization

In equation (1) the parameters that are identified to have a significant impact on PSNR are:

- Quantization parameter
- NumberReferenceFrames
- Bit-rate

The parameter that has the most significant impact on PSNR is bit-rate. It is noted that these two parameters are highly dependent

### 4.3. H.264 Decoder Analysis

An H.264 decoder takes a .264 file as input and outputs a raw YUV video stream. **Error! Reference source not found.** shows the output video artifact of frame 30 with quantization parameter (QP) of 49 that gives very low quality with PSNR of 24.189 db and 5.83 Bitrate compared to QP 17 that has 43.418 db and 979.02 Bitrate.

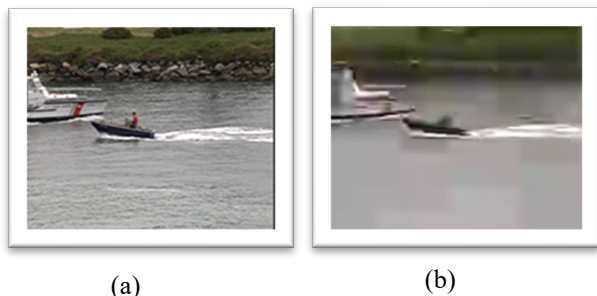


Figure 2: Sample image of frame at (a) QP= 17; (b) at QP= 49

The more QP is increased during the encoding of the video, the more the video lost information and the bitrate reduced.

Note that the Decoder parameters have no impact on Bit-rate and PSNR as these are determined by the encoder. In the proposed framework, the quality and the bit-rate received by the decoder are the same as the encoder output.

The computational complexity of the decoder is analysed using the same method used at the encoder end. For the six given video sequences, experiments were performed in order to find out those coding parameters that can significantly influence CPU utilisation. The objective functions thus obtained are listed within the equations below.

#### Claire Linear Regression Model

$$f(1)_{Dec\_Time} = 0.004 * x(1) + 0.0008 * x(2) - 0.0039 * x(3) + 0.373$$

#### Coastguard Linear Regression Model

$$f(1)_{Dec\_Time} = -0.0042 * x(2) - 0.0118 * x(3) + 0.8945$$

#### Football Linear Regression Model

$$f(1)_{Dec\_Time} = 0.0016 * x(1) - 0.022 * x(3) - 0.0033 * x(4) + 1.3455$$

#### Foreman Linear Regression Model

$$f(1)_{Dec\_Time} = -0.0146 * x(3) + 0.9353$$

#### Mobile Linear Regression Model

$$f(1)_{Dec\_Time} = 0.0071 * x(1) - 0.022 * x(3) - 0.0059 * x(4) + 1.3396$$

#### Tennis Linear Regression Model

$$f(1)_{Dec\_Time} = -0.0158 * x(3) - 0.0072 * x(4) + 1.0304 \quad (4)$$

Table 5 tabulates the correlation coefficients of the decoder objective functions. The Football video sequence has the highest correlation coefficient closely followed by mobile. From the analysis of the linear regression equations obtained to identify parameters that have significant impact on CPU utilisation, Equation 4 reveals that the quantization parameter has the most significant impact. QP has an impact in all the video sequences as evidenced by its presence in all objective functions and being the parameter having the highest magnitude coefficient.

Table 5: Decoder correlation coefficient

| Video Sequences | Decoder Time |
|-----------------|--------------|
| Claire          | 0.9593       |
| Coastguard      | 0.9217       |
| Football        | 0.9984       |
| Foreman         | 0.9786       |
| Mobile          | 0.9958       |
| Tennis          | 0.9873       |

The Encoder and Decoder analysis indicates that the objective functions obtained as a result of using the proposed framework is able to accurately define the significant coding parameters and further detail the level of significance of each parameter. They can also be related to the motion and content information of the videos.

4.4. Multiobjective Optimisation of H.264.

This section presents a framework for multi-objective optimisation of video CODECs. Specifically, an optimization scheme is proposed to determine the optimum coding parameters for a H.264 AVC video codec in a bandwidth constrained environment, which minimises codec complexity and video distortion. Solutions to the optimization problem are reached through a Non-dominated Sorting Genetic Algorithm (NSGA-II). NSGA-II is implemented in the genetic algorithm gamultobj, available in the MATLAB optimisation tool-box and the settings are fixed as shown in Figure 3.

4.4.1. Optimising the Encoder of h.264

The objective functions given in section 4.1 were used to optimise the encoder. These functions are then provided to the NSGA-II optimization tool along with the fitness function and number of variables. The fitness function corresponding to the objective function of the encoded videos is computed. Then populations are generated by applying the crossover and mutation operators described with the settings shown in Figure 3.

The NSGA-II provides all sets of optimal results that jointly minimize complexity, bit-rate and maximize quality. Since a single 3D graph is complex to visualize the optimality of the results, pairs of graphs were plotted.

Since optimization implemented in MATLAB minimizes the objective or fitness function, it solves problems of the form

$$\min_x f(x).$$

If you want to maximize  $f(x)$ ,  $-f(x)$  should be minimised, because the point at which the minimum of  $-f(x)$  occurs is the same as the point at which the maximum of  $f(x)$  occurs.

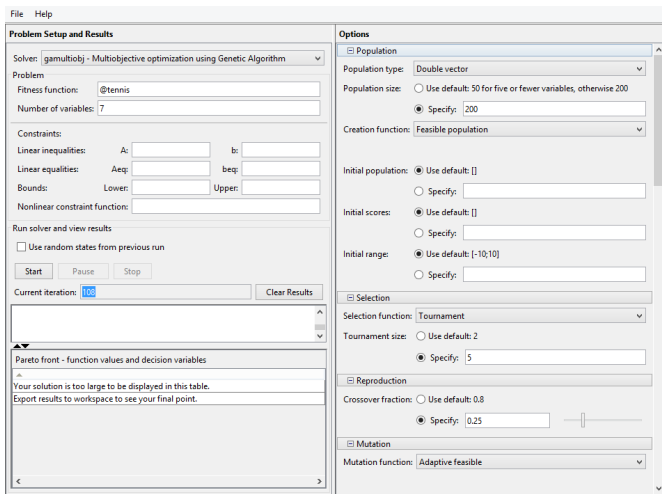


Figure 3: Set options for the problem

To obtain a Pareto front for two objective functions, the optimization is implemented using the above equations in section 4.1. A Pareto plots will appear as shown in Figure 4 – 6.

Figure 4 shows the Pareto front or set of non-dominated solutions for Bit-Rate and PSNR at the final stage of generations being used in the optimization process. One example optimised point is defined as:

IntraPeriod is -14.1153, SearchRange is 4.953125, QP is -9.39187, NRFrames is 21.66449, PSNR is 8.107256 and Bit-rate

is 8.700779. Whereas the optimal values for PSNR is -59.357, Bit-Rate is 967.6315 as shown in Figure 4. Similarly, the results showing the Pareto front of non-dominated solutions for PSNR vs. CPU in Figure 5 and Bit-rate Vs. CPU time are presented in Figure 6.

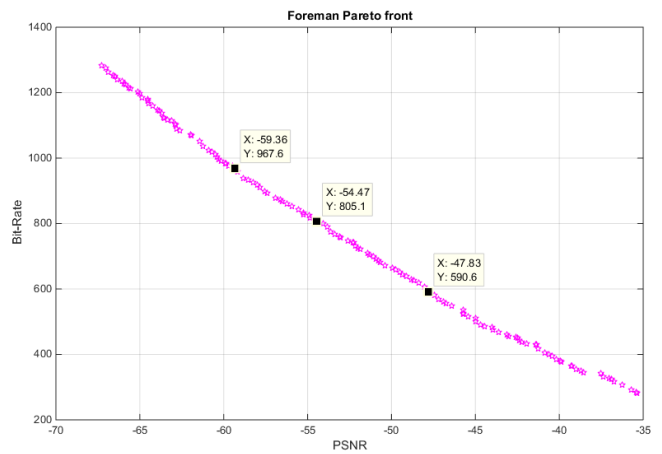


Figure 4: Pareto points for foreman PSNR vs. Bit-Rate.

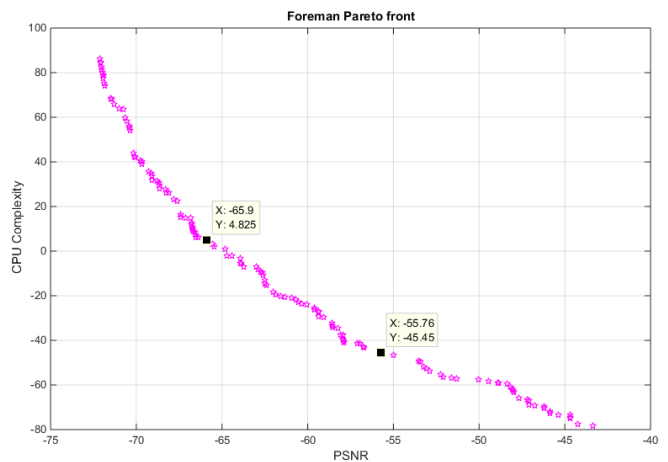


Figure 5: Pareto points for foreman PSNR vs. CPU.

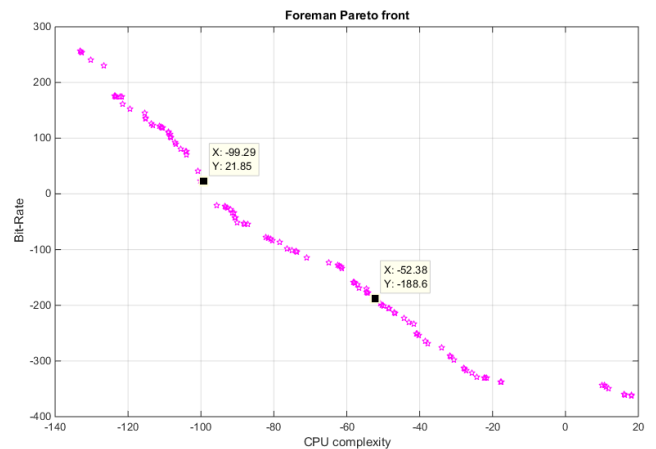


Figure 6: Pareto points for foreman CPU vs. Bit-Rate.

It is noted that the optimisation procedure described above results in a number of optimal solutions.

Table 6 Output data describing the results of MOO with GA for Figure 4 Figure 6.

| Problem           | Number of generations | Population Size | Pareto fraction | Size of non-dominated set | Function count | Average distance | Spread |
|-------------------|-----------------------|-----------------|-----------------|---------------------------|----------------|------------------|--------|
| PSNR vs. Bit-rate | 107                   | 200             | .7              | 140                       | 21601          | 0.0043           | 0.3793 |
| PSNR vs. CPU      | 127                   | 200             | .7              | 140                       | 25601          | 0.0196           | 0.3164 |
| CPU vs. Bit-rate  | 246                   | 200             | .7              | 137                       | 49401          | 0.0442           | 0.5231 |

According to Table 6, population size and Pareto fraction for the GA are set at 200 and 0.7, respectively, which are considered sufficient to generate a search for optimal solutions. The solver will try to limit the number of individuals in the current population that are on the Pareto front to 70 percent of the population size since the Pareto fraction is set to 0.7.

As the MOO algorithm is implemented in MATLAB, and stops at 107 generations and 21601 function counts, the GA selected 140 of the best individuals that are considered as non-dominated solutions out of 200 individuals in the population. Average distance between individuals is 0.0043, which indicates good convergence of the MOO solution, since it has a distance of less than 0.05 from the nearest point in the Pareto set.

4.4.2. Optimising the Decoder.

The analysis of the decoder is limited to decoder parameters that have a significant effect on only the decoder’s computational complexity. In the proposed framework, the quality and the bit-rate received by the decoder are the same as the encoder output. This means the decoder receives all data transmitted by the encoder at the same rate. In such cases, the decoder totally depends on encoder coding parameters.

Figure 7 illustrate graphs between Bit-Rate vs. CPU complexity and PSNR vs. CPU complexity on the way to final generation.

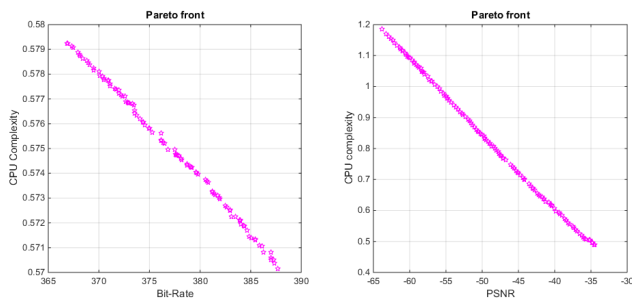


Figure 7: CPU a) Bit-Rate vs. CPU complexity and b) PSNR vs. CPU complexity

5. A Machine Learning based Framework for Parameter based Multi-Objective Optimisation of a H.265 Video CODEC.

5.1. H.265 Profiling Experiments.

This experiment was carried out using the Random Access (RA) configuration file of the Reference software for ITU-T H.265 high efficiency video coding named the HEVC test model (HM) version 16.8. Different resolutions can be used in each profiling experiment: 1080p which is representative for (Full HD) high definition systems with resolution of 1920x1080 pixels in a 16:9 aspect ratio, 2K Video a display resolution of 2560x1600 pixels with a 16:10 aspect ratio and 2160p (Ultra HD) which is representative for the next generation of high quality video. Each video sequence was encoded using a selected combination of possible parameter values of the initial set of encoder parameters.

In other words, each encoding instance corresponds to a combination of coding parameter values, selected from the possible exhaustive set that can be determined by varying each parameter within its entire range. For example, instead of using quantization parameter variations between 1-51 (that is the exhaustive set), only three sample values, 27, 37 and 45, were used (for further examples see Table 7). The table also tabulates the sample values used in our experiments for each parameter from within their corresponding value ranges.

Table 7: Settings for the Encoder in HM.







| Parameter                            | Meaning  | Values Range           |
|--------------------------------------|--|------------------------|
| SourceWidth<br>SourceHeight          | Specifies the width and height of the input video.   | 1920x1080<br>2560x1600 |
| FrameRate                            | Specifies the frame rate of the input video.   | Depends on video       |
| Internal Bit Depth                   | Specifies the bit depth used for coding. When 0, the setting defaults to the value of the MSBExtendedBitDepth.         | 8                      |
| Coding Unit Size/Depth               | Maximum coding unit width in pixel<br>Maximum coding unit height in pixel  | 64/4<br>64/4           |
| IntraPeriod                          | Period of I-frames. Specifies the intra frame period. A value of -1 implies an infinite period.                        | (16,32,48)             |
| GOPSize                              | Specifies the size of the cyclic GOP structure.  | 8                      |
| FastSearch                           | The use of a fast motion search.   | 1:TZ search            |
| SearchRange                          | Sets allowable search range for motion estimation.   | (64,128)               |
| Fast Encoding                        | Fast encoder decision  | (0 or 1)               |
| Quantization Parameter               | Specifies the base value of the quantization parameter. If it is non-integer, the QP is switched once during encoding. | (27,37,45)             |
| Asymmetric Motion Partitioning (AMP) | Enables or disables the use of asymmetric motion partitions.   | 1                      |
| Sample adaptive offset (SAO)         | Enables or disables the sample adaptive offset (SAO) filter.   | 1                      |
| Rate Control                         | Rate control: enables rate control or not.   | 0                      |

Table 8 shows selected sample frames of a set of six video sequences with different resolutions. Note that typical resolutions used in conjunction with H.265 video coding standard, i.e., 1080p and 2K resolution videos are used in all experiments, to carry out the analysis and make the relevant conclusions of this research. However, without any restrictions the proposed framework can be used in relation to a video sequence of any resolution, in particular

HD and full-HD 2K, 4k and beyond. The six selected video sequences have different properties of object motion, both in the foreground and background. Further differences exist in the scene content.

The experiments were initially conducted on a HP computer, running Microsoft Windows 8.1 (64-bit), having an Intel Core i5 CPU 4200Y @ 1.40 GHz and 4.00GB RAM. However it was found that coding HD resolution video is an intensive task that required for example, if encoded in the computer with the above specification, 10 hours to encode 50 frames of a 1920x1080 video at QP 37, intra period 48 and search range 64. Consequently, a decision was made to make use of a High Performance Computing (HPC) facility. Thus for all the experiments a HPC system using Redhat Enterprise Linux v6, with 20 cores of Intel Ivy Bridge Xeon E5-2670 containing 64GB RAM was used, significantly reducing the execution time per experiment.

Table 8: HEVC Tested Video Sequences.

|   |   |
|---|---|
|    |    |
| <b>YachtRide 1920x1080</b>  | <b>Traffic 2560x1600</b>  |
|   |   |
| <b>BasketballDrive 1920x1080</b>  | <b>Jockey 3840x2160</b>   |
|  |  |
| <b>Cactus 1920x1080</b>   | <b>YachtRide 3840x2160</b>  |

A sample of 36 data instances of the Cactus video sequence is presented in Table 9. These were used in the final stage of modelling the PSNR, Bit-rate and CPU utilisation. These are the inputs to the [19] linear regression based modelling process that result in the three objective functions that include the significant parameters, Intra Period as  $x_1$ , Search Range  $x_2$ , Quantization Parameter  $x_3$  and Fast Encoding  $x_4$ .

The resulting objective functions for Bit-rate, PSNR and CPU time are the final outcomes of the performance modelling of the CODEC. Separate experiments are performed for each of the sample test videos.

Based on the output of the linear regression algorithms applied as explained above, the objective functions for the three objectives (for the Cactus video) are found as presented in equation (5). These functions provide the means to discuss in detail the significance of each parameter and how they affect the PSNR, Bit-rate and CPU encoding time. The following section provides an analysis of the experimental results. In particular the analysis considers the test videos separately and discusses the impact of each coding parameter given the known properties of the contents of each video. [Note that for each video, a different model is generated based on the video's inherent properties.]

$$f(1)_{Bitrate} = -22.4664 * x(1) - 386.2482 * x(3) + 18066.616$$

$$f(2)_{PSNR} = -0.0039 * x(1) - 0.4404 * x(3) + 48.873$$

$$f(3)_{Enc\_Time} = 3.9537 * x(1) + 2.5501 * x(2) - 36.2174 * x(3) + 545.1239 * x(4) + 2768.025 \tag{5}$$

Table 9: Selected Set of Parameters for Cactus video Sequence.

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | Bitrate  | PSNR    | Encoding Total Time |
|-------|-------|-------|-------|----------|---------|---------------------|
| 16    | 64    | 27    | 1     | 8422.096 | 36.8076 | 2000.97             |
| 16    | 64    | 37    | 1     | 2195.592 | 32.7418 | 1612.08             |
| 16    | 64    | 45    | 1     | 736.12   | 28.9058 | 1474.57             |
| 16    | 128   | 27    | 1     | 8424.696 | 36.8054 | 2140.34             |
| 16    | 128   | 37    | 1     | 2197.152 | 32.7447 | 1722.19             |
| 16    | 128   | 45    | 1     | 735.304  | 28.9092 | 1556.66             |
| 16    | 64    | 27    | 0     | 8414.944 | 36.8149 | 2559.02             |
| 16    | 64    | 37    | 0     | 2196.192 | 32.7507 | 2106.38             |
| 16    | 64    | 45    | 0     | 735.864  | 28.9111 | 1925.5              |
| 16    | 128   | 27    | 0     | 8414.864 | 36.8149 | 2778.27             |
| 16    | 128   | 37    | 0     | 2195.184 | 32.7516 | 2287.14             |
| 16    | 128   | 45    | 0     | 736.728  | 28.9173 | 2067.34             |
| 32    | 64    | 27    | 1     | 6993.976 | 36.7337 | 2115.22             |
| 32    | 64    | 37    | 1     | 1726.648 | 32.6277 | 1698.1              |
| 32    | 64    | 45    | 1     | 561.512  | 28.7824 | 1553.63             |
| 32    | 128   | 27    | 1     | 6991.08  | 36.7323 | 2271.37             |
| 32    | 128   | 37    | 1     | 1725.24  | 32.6279 | 1819.83             |
| 32    | 128   | 45    | 1     | 561.896  | 28.7877 | 1634.66             |
| 32    | 64    | 27    | 0     | 6994.6   | 36.7419 | 2684.76             |
| 32    | 64    | 37    | 0     | 1725.12  | 32.6325 | 2198.14             |
| 32    | 64    | 45    | 0     | 563.04   | 28.7949 | 2002.86             |
| 32    | 128   | 27    | 0     | 6991.312 | 36.7421 | 2923.43             |
| 32    | 128   | 37    | 0     | 1725.536 | 32.6336 | 2411.49             |
| 32    | 128   | 45    | 0     | 561.952  | 28.7952 | 2164.01             |
| 48    | 64    | 27    | 1     | 6914.36  | 36.7438 | 2126.42             |
| 48    | 64    | 37    | 1     | 1722.568 | 32.6039 | 1719.68             |
| 48    | 64    | 45    | 1     | 560.656  | 28.7335 | 1556.47             |
| 48    | 128   | 27    | 1     | 6911.576 | 36.7428 | 2295.48             |
| 48    | 128   | 37    | 1     | 1720.96  | 32.5996 | 1867.74             |
| 48    | 128   | 45    | 1     | 559.032  | 28.7422 | 1656.8              |
| 48    | 64    | 27    | 0     | 6911.616 | 36.7488 | 2690.7              |
| 48    | 64    | 37    | 0     | 1720.944 | 32.6059 | 2216.83             |
| 48    | 64    | 45    | 0     | 562.424  | 28.742  | 2018.13             |
| 48    | 128   | 27    | 0     | 6912.472 | 36.7515 | 2962.86             |
| 48    | 128   | 37    | 0     | 1720.176 | 32.6066 | 2437.26             |
| 48    | 128   | 45    | 0     | 560.872  | 28.7534 | 2200.32             |



5.2. H.265 Encoder Analysis.

The Encoder objective functions obtained as a result of the experimental procedure presented in section 5.1 enables one to discuss the significance of each of the coding parameters. Following are the obtained models for each video sequence, with  $f(1)$  representing PSNR,  $f(2)$  rate and  $f(3)$  CPU encoding time.

Otabulates the correlation coefficients of the objective functions. They range between 0-1. A value closer to 1 represents the fact that the dependant variable (in this case Bit-Rate, PSNR or CPU utilisation) can be predicted very accurately from the coding parameters that play a role and have been included within the objective functions.

Table 10: Encoder Correlation Coefficient.

| Video     | PSNR   | Bit-rate | Enc Time |
|-----------|--------|----------|----------|
| Cactus    | 0.9989 | 0.9551   | 0.988    |
| YachtRide | 0.9981 | 0.9532   | 0.9837   |

In analysing the objective functions (5), higher positive coefficients of coding parameters indicate higher positive dependency and higher negative coefficients represent higher negative dependency. If a certain parameter is not present in the objective function, that means that the objective is independent of that parameter. A careful analysis of the coding parameters that have non-zero weighting factors in the objective functions obtained and a comparison of relative magnitudes of the coefficients can lead to a direct correspondence with the properties of the video; for example, the presence of motion in the foreground and background, the speed of movement of objects, sudden scene changes, camera pan/tilt/zoom effects and the general characteristics of the content of the video as well.

For example, the analysis of the linear regression equations obtained for the cactus video sequence identifies all four parameters to have significant impact on CPU utilisation, namely:

- IntraPeriod
- Searchrange
- Quantization parameter
- Fast Encoding

For the same video, the following parameters were identified to have a significant impact on Bit-rate.

- IntraPeriod
- Quantization parameter

The parameters that are identified to have a significant impact on PSNR are:

- IntraPeriod
- Quantization parameter

A more detailed and video sequence specific analysis can be presented as follows.

• **Analysis of the CPU Utilisation Experiment:**

The objective functions obtained for all tested video sequences for CPU encoding time indicates that the parameter that has the most

significant impact on CPU is Fast encoder decision. Further, in the selection of the Intra-Period, more I frames (smaller intra period) results in a higher processing time. The next significant impact is from the Quantization parameter. The impact from search range (SR) and Intra Period (IP) is relatively insignificant.

When search range increases, encoding time will slightly increase. These tests have no major impact on the quality of the video. Disabling FEN will also slightly increase encoding time. However it has no major impact on quality.

• **Analysis of the PSNR Experiment:**

The parameter that has the most significant impact on PSNR is QP. The PSNR results tabulated in 0 indicate that the two videos with the least amount of movement/changes, namely Cactus and YachtRide have the best correlation coefficients. This is expected due to the stability of the CODEC during the encoding of the individual frames of the coded sequence.

• **Analysis of the Bit-Rate Experiment:**

The parameter with the most significant impact is the QP. Lower quantisers result in higher bitrate and correspondingly higher visual quality as illustrated in Figure 8. (QP) has a very important impact on the compression rate of H.265.

In cactus, both PSNR and Bit-Rate have no impact from the Search Range. This is true given the fact that for videos with fast moving objects, best matches will not be found quickly, i.e. without having to scan the entire video. All objective functions include a similar constant term indicating that a fixed computational cost for encoding is present, which is independent of the selection of coding parameters. This is expected given the processes that exist, which are independent of the coding parameters.

5.3. H.265 Decoder Analysis

The computational complexity of the decoder is analysed using the same method used at the encoder end. Experiments were performed in order to find out those coding parameters that can significantly influence CPU utilisation. The objective functions thus obtained are listed within equation (6).

$$f_{Dec\_Time} = 0.0023 * x(2) - 0.0819 * x(3) + 0.1178 * x(4) + 8.2928 \tag{6}$$

Table 11 Decoder Correlation Coefficient

| Video     | Dec Time |
|-----------|----------|
| Cactus    | 0.9509   |
| YachtRide | 0.9263   |

Table 11 presents the correlation coefficients of the objective functions. The cactus video sequence has the highest correlation coefficient. The analysis of the linear regression equations is carried out to identify parameters that have significant impact on CPU utilisation. Equation (6) reveals that the Fast Encoding has the most significant impact, being the highest magnitude coefficient.

The Encoder and Decoder analyses indicate that the objective functions obtained as a result of using the proposed framework are able to accurately define the significant coding parameters and further detail the level of significance of each parameter. They can also be related to the motion and content information of the videos. More importantly, these objective functions model the behaviour/properties of the encoder and decoder thus allowing them to be used in multi-objective optimisation as described in the next section.

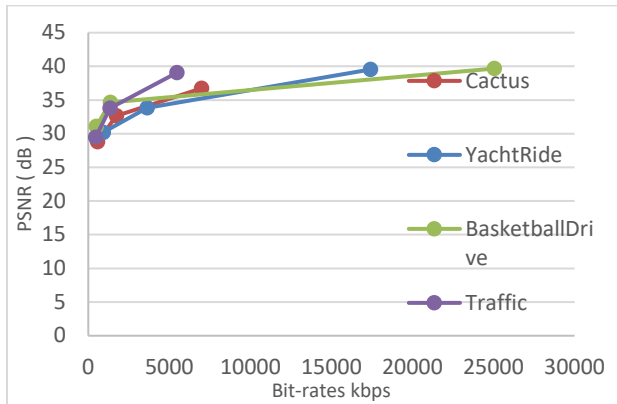


Figure 8: PSNR versus Bit-rate at QP 27, 37,45.

The HM encoder/ decoder configuration file takes a .265 file as input and outputs a raw YUV video stream as ReconFile. The output shown in Figure 9 indicates that the output video artifact of yacht video at frame 30 with quantization parameter (QP) of 45 gives very low quality with PSNR of 28.7877 dB compared to QP 27, which has 36.7323.

The more the QP is increased during the encoding of the video, the more the video loses information and bitrate reduces.

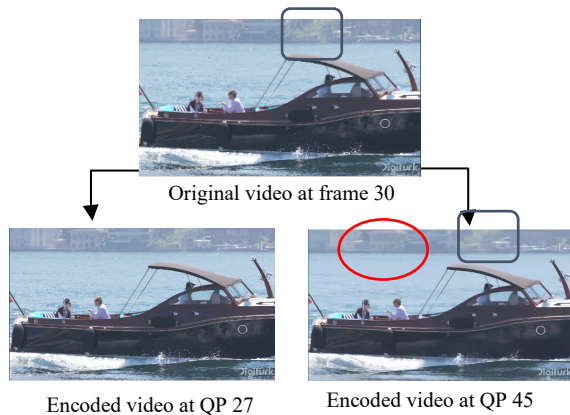


Figure 9: Shows the visual artifact with different QP

#### 5.4. Optimising the Encoder of H.265

In this section, the objective functions given in equation (5) were used to optimise the encoder performance under multiple constraints. The experiment was conducted, and the following setting of parameters for the GA were chosen (see Table 12)

Table 12: Settings for Multiobjective Genetic Algorithm Solver.

| gamultobj settings                |   |
|-----------------------------------|---|
| Fitness function:                 | @function                               |
| Number of variables:              | 4                                       |
| Bounds Constraints:               | lb = [16,64,27,0]<br>ub = [48,128,45,1] |
| Creation Function:                | Constraint dependent                    |
| Population Size:                  | 60                                      |
| Initial Population:               | Default                                 |
| Crossover Fraction:               | 0.8                                     |
| Mutation Function:                | Constraint dependent                    |
| Crossover Function:               | Intermediate                            |
| Crossover Ratio:                  | 0.8                                     |
| Pareto Front Population Fraction: | 0.35                                    |
| Maximum Generations:              | 300                                     |
| Plot functions:                   | Pareto front                            |

The results of Multi-Objective Optimisation (MOO) Pareto set analysis are presented in Figure 10 and Figure 11. Figure 10 shows the Pareto front or set of non-dominated solutions for Bit-Rate and PSNR. The Pareto front illustrated is within a limited range and hence shows that the points lie on a straight line. In practice, when the range of testing is increased the shape of the curve would represent a typical shape of a Pareto curve. The Pareto curve allows one to select optimum performance points and hence select the corresponding coding parameters that resulted in the objective function optimal values for coding the videos.

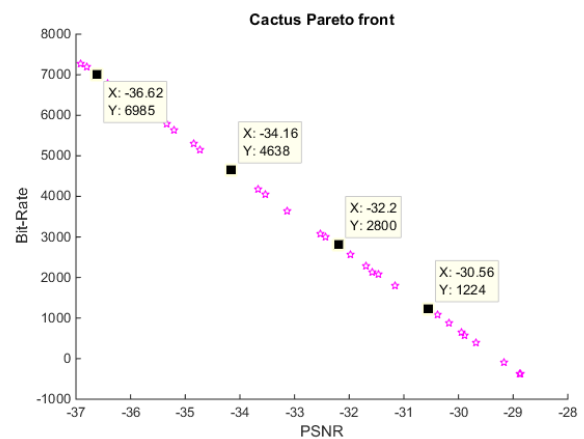


Figure 10: Pareto front for cactus video sequences PSNR vs. Bit-Rate.

For one optimal point of operations, the IntraPeriod is 27.49851, SearchRange is 83.34469, QP is 33.16743, Fast Encoding is 0.264558, PSNR. Whereas the optimal value for PSNR (X) is -34.1591, Bit-Rate (Y) is 4637.964 as shown in Figure 10. Similarly, the results showing the Pareto front of non-dominated solutions for Bit-rate Vs. CPU complexity are presented in Figure 11.

Table 13: Output data describing the results of MOO with GA for cactus.

| Problem          | Number of Generations | Size of population | Pareto fraction | Size of non-dominated set | Function count | Average distance | Spread |
|------------------|-----------------------|--------------------|-----------------|---------------------------|----------------|------------------|--------|
| PSNRvs.Bit-rate  | 259                   | 60                 | 0.35            | 35                        | 13001          | 0.0149           | 0.1487 |
| CPU vs. Bit-rate | 220                   | 60                 | 0.35            | 21                        | 13261          | 0.0308           | 0.1050 |

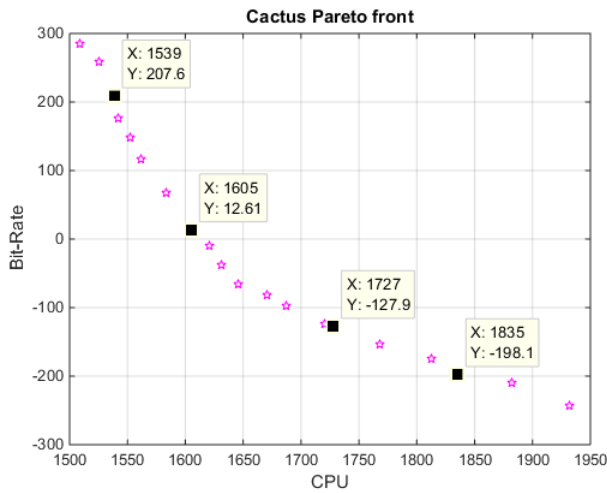


Figure 11: Pareto front for cactus video sequences CPU vs. Bit-Rate final Generation.

In Table 13, population size and Pareto fraction for the GA are set at 60 and 0.35, respectively, which are considered sufficient to generate search for optimal solutions.

At 259 generations and 13001 function counts, the GA selected 35 best individuals considered as non-dominated solutions out of 60 individuals in the population. Average distance between individuals is 0.0149, which indicates good convergence of the MOO solution.

**Conclusion**

In this paper we have proposed a machine learning based approach for the determination of significant coding parameters of a H265 video CODEC. In particular, we have used multivariate regression analysis in defining objective functions for CPU utilisation, PSNR and the bit-rate of a video CODEC when a given video is being Encoded/Decoded. We have been able to use known information about the content and the motion present in the test videos to justify the formation of the objective functions. We have shown that these regression equations provide the means for modelling the performance of a typical H.265 video CODEC. Finally, we have used these models to optimise the performance of a video CODEC under multiple constraints. For this purpose we demonstrated the effective use of a Genetic Algorithm based approach.

The proposed framework for the performance analysis, modelling and multi-objective optimisation of a H.265 video CODEC can be applied to any video CODEC and provides a useful contribution to the video coding community who are often

faced with the dilemma of selecting values for a large number of coding parameters with the intention of obtaining optimal performance of the CODEC under multiple performance constraints.

**6. Conflict of Interest**

The authors declare that there is no conflict of interest regarding the publication of this paper.

**Acknowledgment**

This research is supported by the Ministry of Man Power (MOMP) Muscat, Oman.

**References**

- [1] M. Al-Barwani and E. A. Edirisinghe, "A machine learning based framework for parameter based multi-objective optimisation of a H.265 video CODEC," *2016 Future Technologies Conference (FTC)*, pp. 553–559, 2016.
- [2] H. Zhihai, L. Yongfang, C. Lulin, I. Ahmad, and W. Dapeng, "Power-rate-distortion analysis for wireless video communication under energy constraints," *Circuits Syst. Video Technol. IEEE Trans.*, vol. 15, no. 5, pp. 645–658, 2005.
- [3] H. Zhihai, C. Wenye, and C. Xi, "Energy Minimization of Portable Video Communication Devices Based on Power-Rate-Distortion Optimization," *Circuits Syst. Video Technol. IEEE Trans.*, vol. 18, no. 5, pp. 596–608, 2008.
- [4] R. Vanam, E. A. Riskin, and R. E. Ladner, "H.264/MPEG-4 AVC Encoder Parameter Selection Algorithms for Complexity Distortion Tradeoff," in *Data Compression Conference, 2009. DCC '09.*, 2009, pp. 372–381.
- [5] P. Wei, L. Yan, and W. Feng, "Joint Power-Distortion Optimization on Devices with MPEG-4 AVC/H.264 Codec," in *Communications, 2006. ICC '06. IEEE International Conference on*, 2006, vol. 1, pp. 441–446.
- [6] K. Jaemoon, K. Jungsoo, K. Giwon, and K. Chong-Min, "Power-rate-distortion modeling for energy minimization of portable video encoding devices," in *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*, 2011, pp. 1–4.
- [7] C. Fu-Chuang and H. Yi-Pin, "Rate-distortion optimization of H.264/AVC rate control with novel distortion prediction equation," *Consum. Electron. IEEE Trans.*, vol. 57, no. 3, pp. 1264–1270, 2011.
- [8] R. Vanam, E. A. Riskin, S. S. Hemami, and R. E. Ladner, "Distortion-Complexity Optimization of the H.264/MPEG-4 AVC Encoder using the GBFOS Algorithm," in *Data Compression Conference, 2007. DCC '07*, 2007, pp. 303–312.
- [9] F. Al-Abri and D. Eran, "Multi-Objective Optimization Of Video Coding and Transcoding," Loughborough University, 2010.
- [10] F. Al-Abri, X. Li, E. A. Edirisinghe, and C. Grecos, "A Novel Framework for Multi-objective Optimization of Video CODECs," in *CyberWorlds, 2009. CW '09. International Conference on*, 2009, pp. 195–202.
- [11] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding," *Ieee Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [12] S. Weiwei, F. Yibo, H. Leilei, L. Jiali, and Z. Xiaoyang, "A hardware-friendly method for rate-distortion optimization of HEVC intra coding," in *VLSI Design, Automation and Test (VLSI-DAT), 2014 International Symposium on*, 2014, pp. 1–4.

- [13] D. Grois, D. Marpe, A. Mulyoff, B. Itzhaky, and O. Hadar, "Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders," *2013 Picture Coding Symposium (PCS)*. pp. 394–397, 2013.
- [14] B. Bross et al., "HEVC real-time decoding," in *SPIE Optical Engineering+ Applications*, 2013, p. 88561R–88561R.
- [15] B. Bross et al., "HEVC performance and complexity for 4K video," *Proc. 2013 IEEE 3rd Int. Conf. Consum. Electron. - Berlin, ICCE-Berlin 2013*, pp. 44–47, 2013.
- [16] M. P. Sharabayko and N. G. Markov, "Iterative intra prediction search for H.265/HEVC," *2013 Int. Sib. Conf. Control Commun. SIBCON 2013 - Proc.*, 2013.
- [17] F.-I. H. H. I. Dolby Laboratories Inc. Microsoft Corporation, "H.264/AVC Reference Software." Joint Video Team, Germany, 2009.
- [18] Intel, "Intel® VTune™ Amplifier XE," 2013. .
- [19] E. F. Mark Hall Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witte, "The WEKA Data Mining Software," 2009. [Online]. Available: <http://www.cs.waikato.ac.nz/ml/weka/>.