Differential Evolution Based Significant Data Region Identification on Large Storage Drives

Nitesh K. Bharadwaj and Upasna Singh

Abstract In today's scenario, almost every user involuntarily generates and utilizes several Gigabytes and Terabytes of data. It is due to the accessibility of diverse and inexpensive digital hard disk drives (HDDs) that have facilitated users with comparably large storage capacities. Almost every digital crime is directly or indirectly associated with storage devices. The ever increasing storage strength of HDD has elevated the forensic examination cost and complexities for the digital forensic investigator. The considerable amount of time is consumed during identification and analysis phase of Digital Forensic (DF) process which creates huge backlog of cases, as a result remarkable delay occurs for availing justice from judicial body. In this research, we propose a methodology to identify forensically significant data regions of suspected drive that can be helpful in accelerating overall digital investigation process. A proof-of-concept technique is developed that utilizes Differential Evolution (DE) for determining the significant data regions and data storage pattern of HDD. The proposed approach incorporates DE which internally utilizes the geometry information of the HDD, i.e. cylinder, track and sector values, for population generation and decision making. Throughout the paper DE samples are defined using the geometry information and entropy as fitness value. Storage devices with different storage capabilities were considered for the experiment and analysis. Detailed case study using the analysis on formatted suspected storage drives highlights the relevance of the proposed approach. The end result is series of output files, providing information about significant regions of the HDD, using which investigator can easily interpret and analyze the suspected drive. Finally, the proposed method is compared with the important functionalities of existing approaches.

Keywords Computational intelligence · Evolutionary computation · Differential evolution · Digital forensics · Storage drive

N.K. Bharadwaj (✉) · U. Singh
Department of Computer Science and Engineering, Defence Institute
of Advanced Technology, Pune 411025, India
e-mail: nitesh_pcse14@diat.ac.in; niteshb2k14@gmail.com

U. Singh
e-mail: upasnasingh@diat.ac.in; upasna.diat@gmail.com

A.K. Sangaiah et al. (eds.), *Intelligent Decision Support Systems for Sustainable Computing*, Studies in Computational Intelligence 705,
DOI 10.1007/978-3-319-53153-3_8

1 Introduction

Gordon Moore in 1965 visualized that of transistor count on an integrated circuit doubles approximately every two year, which has brought revolutionary development in speed, area and capacity of modern processors as well as storage-devices. As a result the technology has become cheaper and grown at exponential rate with reference to Moores law, which has finally converged everything towards digital world. Due to relentless scaling in device size and cost the use of digital devices, for example smart-phone, tablet, laptop, personal computers, camera etc., has been completely dissolved in our daily lives. Necessarily, every digital device cannot be utilized until equipped with memory devices, for example random access memory (RAM), secure digital card (SD), micro SD card, solid state drive (SSD), flash memory card, hard disk drive (HDD), universal serial bus (USB) etc. These storage devices act as a prerequisite to facilitate trending technological benefits to both personal and commercial users.

Every digital device has distinct architecture, configuration and functional capabilities, but the core element of device that enables user and system specific operation are storage devices. Along with the advanced facilities provided by modern digital devices, sometimes it also proves to be a major concern from the perspective of cyber-crimes and unethical activities of offenders. A cyber-crime is defined as a criminal activities carried out by means of computer or digital devices or the internet. The digital devices have created new criminal arena for cyberwarfare, cyber terrorism, fraud and financial crimes, cyberextortion etc. Concurrently, the technology also provides revolutionary platform to criminals and offensive group (cyber and non-cyber) to multiply their influence, for example unauthorized access (hacking), child pornography, electronic harassment, extortion, drug trafficking etc. Possibly every user activities data or information are stored in configured storage devices. Proliferation of digital devices in every domain has created new opportunities and investigative challenges for digital investigators across the world [1]. On the other hand, the examination of suspicious or criminal activities on the seized devices is carried out by digital forensic investigator. The responsibility of investigator is to prove or disapprove the existence of reported suspicious activity based on the forensically examined digital evidence and artifacts. The digital evidence is information (data having forensic value) found on wide range of electronic devices that is equivalent to digital fingerprint of the suspected system. However, the primary objective of digital forensic investigator is to convince the judicial body for justice by utilizing their available Digital Forensic (DF) technical and management strengths. Digital forensic is the process of collection, identification, preservation, examination, analysis and presentation of digital evidence with respect to reported criminal cases, that are legally and judicially acceptable.

1.1 Motivation and Focus

In the present scenario, where DF practitioners are continuing to face "coming digital forensic crisis" [2] possibly due to the storage device capabilities that are now comparably larger, very complex, can consist of huge amount of unstructured and structured data, can be easily intermixed with variety of devices, operating systems, file systems and, media types. Additionally, every year there is rapid increase in manufacturing of storage devices and advancement in their storage capacity beyond the capabilities of processors and existing forensic tools/software. Since, the investigation time increases with the increase data volume that need to be analysed, investigators have observed exponential growth in number of storage volumes registered for forensic analysis which is matter of concern over past few years [3]. During examination of large storage drive it is infeasible to process every byte of data which in turn consumes exhaustive time of the investigator. In the present era of huge volumes of data which will grow exponentially in future and hence, examination time will correspondingly become out of scope. This chapter focuses on alleviating the processing and examination time of suspected storage drive by utilizing computational intelligence (CI) techniques in order to optimize the overall traditional DF process by exploiting the important region of the drive. The idea is to initially determine the selected significant regions of the suspected drive instead of considering every bytes of suspected drive.

For the first time in this chapter we have utilized drive's structural information with differential evolution algorithm to determine significant regions of the drive for achieving fast forensic examination of storage drive. The objective of this work is to study and analyze the affect of evolutionary algorithm for overall advancement of traditional DF process. The assumption considered in this work is that although the suspected storage drive contains different types of information which are stored according to the availability of free spaces i.e. either fragmented or continuous locations. Among different types of available data the highest entropy data is the target requirement whereas the null data sectors are irrelevant to the investigator. The proposed technique focuses on identification of data sectors which have higher entropy values. Hence, this chapter focuses on fast and efficient evidence location identification by using differential evolution algorithm. The proposed methodology utilizes the differential evolution algorithm for determining and examining only the significant data regions of the disk irrespective of the specific target data. Finally, the investigator is provided with the sector locations of significant data for fast examination of suspected drive contents. The proposed approach can be utilized at digital forensic laboratory with no additional cost. The contributions of this chapter are as followed below:

- Trade-off rectangle is proposed to understand present scenario of DF with respect to the resource-utilization, evidence processing-time, accuracy and today's technological gap.

- Significant data regions identification on storage drive for fast examination with the help of popular evolutionary algorithm. The proposed approach best suits the following scenario:
  - Pattern analysis of the data stored in suspected storage drives
  - Determines sector locations of significant data from storage drives
  - Identifying significant data regions within large storage drives

The rest of the chapter is organized as follows: Sect. 2 provides insight into the digital forensic and its pressing issues along with the introduction to computational intelligence paradigm including the contribution of various evolutionary algorithm towards its wide acceptance in diverse application. Implementation details of DE based proposed methodology is covered in Sect. 3 while experimental setup and analysis results in addition with case study are discussed in Sect. 4. Brief discussion on findings and future scope of the proposed work is provided in Sect. 5. Finally, the chapter is concluded in Sect. 6.

2 Background and Related Work

Digital forensic is a practice of investigation as well as recovery of vital materials found in digital devices often associated with computer crimes. A full forensic examination requires processing of each and every byte of the suspected media to determine what it represents and how it is forensically significant to investigator. Hence, unpredictable time is consumed during full-forensic examination of large storage drives. In literature, for examination of HDD, different researchers and investigators presented their concern and contributed various methodologies, tools and techniques using [2–9]. The survey with respect to published and appreciated research as well as their preferred solution in the field of storage device forensic are provided in [3]. The literature covers the forensic solution in consideration with data mining, increased processing power, distributed processing and, artificial intelligence etc. The authors in [2] presented their concern towards the impact of increasing volume of data and the growing number of devices on DF. The paper [2] presents the method of DF data reduction by selective imaging (DRbSI) where, their proposed methodology presents procedure for collection of only information relevant files and databases. The available predetermined files and database are prime focus of the methodology in [2], but no consideration is made if the relevant files or databases are altered, deleted or formatted. The author in [3, 10] presented that the time required for collection and analysis of full disk imaging process increases with increasing volume of data. In contrast to this digital forensic triage is a recent term which engaged the mind of the researchers. *Digital forensic triage is a partial forensic examination conducted under (significant) time and resource constraints* [4]. Both pros and cons exists for triage, where on one side triage helps to reduces the risk of case backlogs in DF laboratories which in turn reduces the long wait of DF examination results, on the other hand the use of triage tools/software also possess a high risk of evidence being missed during investigation. The possibility when information gathering and analysis is not performed the risk of missed investigative opportunities comes into the picture, which is another drawback of triage. Ayers et al. in [11] discussed that due to increase in volume of data and examination complexity the existing forensic software and tools are becoming inadequate. The available forensic tools and software for examining large storage device are now appealing their scalability issues. In support to this the author in [12] presented the challenges of the next decade, and cited the difficulties in capturing, processing and reporting TBs of data. Moreover, most of the HDD available in market contains traces of previously used data that belongs to the former user. Hence, author in [12] shows fast identification of storage drive contents and to determine whether drive is properly wiped or not. In [13] N. Beebe et al. discusses the scope of data mining techniques, which once utilized in DF as an advanced tool it can provide fruitful results in terms of reduced processing time, less analysis cost, improving quality of information, and pattern extraction. The work in [14] demonstrated the Monte-Carlo filesystem search approach on various storage drives in order to search known files in minimum time. However, the scenario where information of known files are unavailable the search process becomes inefficient. The proposed approach identifies the significant regions of the storage drive for forensic examination irrespective of the target data and file system information.

Furthermore, efforts were made to reduce the evidence processing time with the help of *data reduction* approach. In DF investigation almost every reported case is associated with huge volume of data which is seized for forensic analysis provides great scope towards data reduction technique. The author in [15] stated that for a particular circumstance it is necessary to recognize what information needs to be accumulated such that accurate analysis is achieved. This contradicts the process of investigating everything with the practice of examining only required data which can help investigators to achieve accurate analysis. In this support Jonathan et al. in [6] presented *sifting collectors* approach efficiently images forensically relevant regions from Windows New Technology File System (NTFS) which contain important information about *e.g.* Windows OS files, registry, system metadata, temp files, history, logs, browser artefacts etc. However, the forensic imaging emphasis is more on the system and metadata files of Windows file system whereas large amount of actual data contents are not been considered. The work in [6] presents the acquisition of evidence from windows operating system and supported file system format i.e. NTFS. However, the reliability and feasibility issue in consideration with other operating systems and their supported file system types, for example Unix/Linux with FAT, ext2 or ext4 format, Mac-OS etc., were completely ignored. In this chapter, the proposed methodology follows the footprints of selective imaging and analysis methods by using the differential evolution algorithm. The identified significant regions help the investigator for deciding further course of investigative actions. The proposed approach performs the task irrespective of specified operating system, file system formats and particular files and databases. Moreover, if the suspected drive is deleted or formatted the proposed approach can easily spot the significant sector regions where highest entropy data exists. At the initial phase of investigation if the investigator examine the reported significant region of the suspected drive significant saving in examination time is achieved since the regions having null data will be completely omitted. The exclusion of null data and sectors significantly save the investigator from the analysis of null/zero data. The proposed approach to some extent overlaps with the methods defined in [4, 8]. In the following subsection we provide brief discussion on essential basic building blocks of the proposed approach.

2.1 Digital Forensics

In today scenario digital devices are positively evaluated as an vital evidential proof in judicial body. It is essential for an investigator to understand how digital devices are actively involved in criminal activities as well as the type, format and kind of evidence these devices contains. This requires contents analysis and examination of large storage drive. The conventional DF process takes hours to read entire storage drive while the process become exhaustive when fragmented and deleted files need to be considered. Hence, the extraction and collection of forensically sound digital information from large disk volume is burdensome and time hungry. Therefore, the traditional forensic investigation process cannot fulfil the modern demand due to the perpetually increasing capacity of hard drives. In this support we propose a technological trade-off by comparing accuracy, time complexity, resource utilization, development gap among traditional and ideal forensic process which as illustrated using Fig. 1. The ideal forensic process is primary preference of every forensic inves-

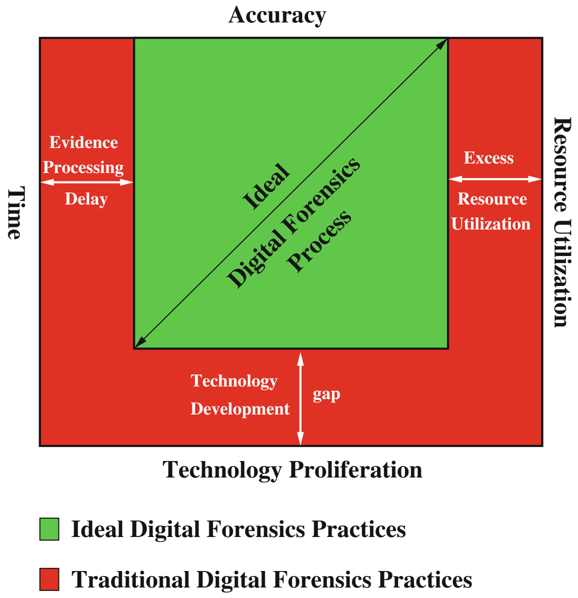

Accuracy
Time
Resource Utilization
Technology Proliferation
Evidence Processing Delay
Excess Resource Utilization
Ideal Digital Forensics Process
Technology Development gap
🟩 Ideal Digital Forensics Practices
🟥 Traditional Digital Forensics Practices

Fig. 1 Proposed trade-off using resource utilization, processing time, technology gap and, accuracy constraints of traditional and ideal DF process

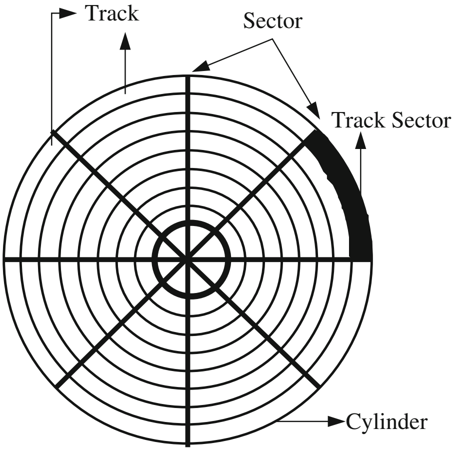


Fig. 2 Hard disk drive basic internal architecture

tigator where best forensic analysis result is procured in minimum time with limited resource utilization as represented in Fig. 1. The traditional forensic process proves to be time and resource inefficient that creates huge backlog of filed cases awaiting justice from judicial bodies. However, here we are not contesting to modify existing judicial process. Achieving best situation for forensic investigation requires introduction of new technological achievements and developments.

2.1.1 Storage Disk Drives

The storage disk drive in today's scenario is also termed as hard disk drive (HDD). The data in earlier storage devices like floppy disk is accessible using the geometry and storage structure of the device. The basic geometry of HDD is represented in Fig. 2 using which the data are allowed to write or read in particular sector or set of sectors. The disk comprises a stack of cylinder which spins and allows the drive heads to move over the tracks in order to access particular sectors. The information on a hard disk is stored in tracks, which are concentric circles placed on the surface of cylinder (also known as platter). The head is used to physically locate the location to access the data. The index number of tracks ranges from zero at the edges of cylinder which increases towards center of the track. Sections within each track are called as sectors, which is the basic physical storage unit on a HDD, having 512 bytes size. The cylinder-track-sector (CTS), is an classic method for giving addresses to each physical address of data on a HDD. Though geometrical metrics (platter, track and sectors) no longer have a direct physical relationship to the data stored on modern storage devices except floppy disk, virtual geometrical values (which can be translated by disk driver or software) are being used by many utility programs and file systems to access data by using sector locations. The reason is that less head movement is possible whenever geometry is used for data access. Although, each cylinder consists of two tracks (one on each side), the modern drives utilize only one track of the cylinder while other side of track is used for storing the control information that is unavailable to the operating users. Hence, the geometry metrics of HDD provides fast access to stored data.

The proposed methodology implicitly utilizes *hdparm* command line tool in order to extract the geometry information of storage drives. This command line utility is used to set and view hardware parameters of hard disk drives. Hdparm tool provides valuable geometry information (CTS parameters) about any storage device. For example let us try to derive disk drive capacity which is a function of cylinder (C), tracks (T), sectors (S) and sector size (B). Consider a particular disk has X_1 cylinder, X_2 track per cylinder, X_3 sectors per track and X_4 sectors size in bytes, the information are retrieved after execution of hdparm utility. The total storage capacity storage drive can be provided using the following Eq. 1.

$$\text{HDD\_Capacity} = \prod_{i=1}^{4} X_i \text{ (C, T, S, B) bytes} \tag{1}$$

In this chapter the CTS information extracted during run time is utilized throughout for identification of significant data regions from suspected storage drives. Therefore, the total capacity of the provided HDD geometry information can be easily derived and also each sector is independently accessed for analysis of data. This chapter utilizes the geometry metrics to access particular data of 512 bytes for searching significant data regions of the disk drive.

2.1.2 Major Issues in Storage Drive Forensics

The easy availability and accessibility of today's large storage drives has provided remarkable features to the criminal users and new challenges to the investigators. The open issues related to examination of storage drives from the perspective of investigator are discussed as followed: Every user now a day involuntarily generates and utilizes several Gigabytes (GB) and Terabytes (TB) of data where it is infeasible for investigator to analyze and process every byte of data within specified time bounds. The continuous development in storage technology i.e. increasing storage capacity in almost all consumer devices, the increasing number of variety of digital devices and cloud storage services has exponentially increased the number of devices seized per case. Digital investigation tools and techniques are lagging behind the pervasive advanced technologies due to which investigator exhaustive time is consumed in the earlier examination phase of traditional DF process. Seizure of number of devices as well as lack of well defined investigation tools creates huge backlogs of evidence awaiting analysis. Time constraint and processing delay also creates huge log of unprocessed cases which in turn develop huge delay in evidence processing as a result corresponding delay in judicial decision is obtained from legal bodies. In this chapter we present our concerns towards above mentioned investigation issues and propose a methodology for evidence locality identification to overall accelerate digital investigation process by utilizing popular evolutionary algorithm. In the next sub-section an insight into the computational intelligence paradigm is provided for better understanding of the proposed approach.

2.2 Computational Intelligence Paradigm

Before proceeding to problem objective it is necessary to understand what computational intelligence really is. Computational Intelligence (CI) is the branch of science and engineering where complex computational problems are solved by modeling problems according to the natural and biological intelligence, resulting in "*intelligent systems*". These intelligent systems encapsulate numbers of popular intelligent algorithms; artificial neural networks, evolutionary computation, artificial immune systems, fuzzy system and swarm intelligence. Hence, these intelligent algorithms belong to the field of *Artificial Intelligence* (AI). Alternatively, it is stated that computational intelligence is a sub-division of AI, where CI is analysis and study of adaptive mechanism to facilitate intelligent responses in complex and differing environment.

This section highlights the subcategories of computational intelligent paradigm namely evolutionary computation (EC) and differential evolution. Computational intelligence refers to the study and design of intelligent algorithms that has ability to learn specific task and behavior from the experimental data and result observations. CI techniques generally address the complex real-world problems for which conventional mathematical modeling is undefined or unpredictable. The final outcome of computational algorithms acts as a heuristic solution to solve particular objectives related to diverse area of research.

2.2.1 Evolutionary Computation

Evolutionary computation (EC) mimics the nature-inspired evaluation competence, where the major approach of survival of suitably best generation is supported while the weakest is left to die. In natural evolution process survival is achieved with the help of reproduction similarly the offspring is reproduced from two or more parents that contain the best characteristic of each parent. The produced individuals that inherit imperfect characteristics are always weak which lose combat to survive, for example in the bird species out of several one infant manages to get more food, becomes stronger, as a result the strong infant kicks down all weak siblings from the nest to die.

Individual in evolutionary algorithms (EA) is referred to as a chromosome and EA works on large population of individuals with maximum randomization. The characteristics of individuals in the population are defined by the chromosome where each characteristic is called as gene. Each individual of population compete for reproduction of suitable offspring. The offspring with desired values have long survival capabilities and good chances of reproduction for further population generation. Crossover and mutation are the two crucial processes for every evolutionary algorithm. Crossover is the process where offspring is generated by combining the sub-parts of two or more parents while mutation is the process where each offspring undergo alteration of some characteristics of the chromosome. The deciding factor for the survival of produced individual is fitness value that is derived by corresponding objective or fitness function. The constraints of problem and objective are modelled in the form of fitness function. In this way with the use of fitness function, population generation process, crossover process, mutation and selection process, a successful evolutionary computation can be utilized to solve the untouched problem of real world. However, in the literature several EA have been developed to solve several problems as listed below:

- Genetic algorithm: models the chromosome recombination and mutation process for searching heuristic solutions
- Differential evolution: Operates on vector differences hence, best for numerical optimization problems
- Cultural evolution: It is a process of dual inheritance where a set of behavioural traits form population individuals
- Genetic programming: Computer programs forms a set of genes and chromosomes for further reproduction and mutation operations
- Neuroevolution: The evolutionary algorithms are used to train artificial neural networks

In the literature over a period of time the real-world application have been successfully benefited from evolutionary computation, for example, operations research, robotics, combinatorial circuits optimization, process scheduling, fault tolerant systems, chemistry and physics etc. In this chapter the differential evolution algorithm is used to enhance overall DF process especially for examination of large storage drives from digital investigation perspective.

2.2.2 Differential Evolution

A popular evolutionary algorithm (EA) i.e. Differential evolution (DE) was proposed by Rainer Storn and Ken Price in 1997. DE has always provided acceptable results whenever used in a variety of problems from diverse fields [16]. Similar to other EAs, at each generation DE also utilizes mutation, crossover, and selection operations in order to achieve global solution by moving population toward the optimal solution. DE's performance is mainly dependent on two components:

1. Offspring vector generation: mutation and crossover operations
2. Control parameters: population size (NP), crossover control rate (CR), and scaling factor (F).

The details of each component in contrast to the prescribed objective are elaborated in the subsequent section of the chapter. Researchers are continuously utilizing DE for finding optimum solution to crack their research complications. For example, DE has been successfully applied to distinct areas of science and technology, such as signal processing [17], chemical engineering [18], machine intelligence & pattern recognition [19], and mechanical engineering design [20]. The experiments performed over several numerical benchmark problems [21] illustrate that DE performs better than the particle swarm optimization (PSO) [22] and genetic algorithm (GA) [23]. At the initial phase of DE algorithm generates random population by using uniform distribution followed by mutation, crossover and selection operations in order to further generate a new population. The crucial step in DE algorithm is generation of trial vectors. Mutation and crossover operations are the two basic steps for generating the trial vectors. The best trail vector is selected by the selection operator for next population generation. The discussion on implementation details of the proposed methodology based on DE algorithm along with the insight into important sub-operations of DE (crossover, mutation and selection) is provided in the following section.

3 Implementation of Differential Evolution Based Significant Data Region Identification

In this section discussion on DE based proposed significant region identification methodology is provided along with detailed into the internal characteristic of the algorithm. The proposed approach uses the HDD geometry to access every possible random sector data in the manner similar to that is shown in Eq. 2. The proposed technique generates random geometry information (C_{r1}, T_{r2}, S_{r3} values) to get access to random sector location of the HDD.

$$\begin{aligned} \text{Sector\_Number}_r &= C_{r1} \times \mathrm{T}_N \times \mathrm{S}_N + T_{r2} \times \mathrm{S}_N + S_{r3} \\ &\quad \text{where } C_{r1} \in C_{N1},\ T_{r2} \in T_{N2}, S_{r3} \in S_{N3} \\ &\quad f(r1, r2, r3) = rand(N1, N2,\ N3)\ and, \\ &\quad N1 = \#\text{Cylinders}, N2 = \#\text{Tracks}\ \&\ N3 = \#\text{Sector} \end{aligned} \tag{2}$$

where, C_{r1}, T_{r2}, S_{r3} and Sector_Number$_r$ are the randomly selected geometry parameters while, C_{N1}, T_{N2} and S_{N3} are the total cylinder, track and sector numbers of the disk drive. Every EA operates with maximum randomization of the parameters for better performance and results therefore, the proposed technique also rely on achieving more random behaviour after selecting random CTS values (C_{r1}, T_{r2} and S_{r3}), as represented in Eq. 2. The CTS values are the important numerical parameters that provide access to every possible data location of storage drive. The fitness function for the proposed approach is *Entropy* (E) of the sector data, the generalized equation for entropy is presented in Eq. 3, where $\forall x \in$ Sector data bytes. The entropy is calculated for every randomly selected sector which generates large amount of sector samples for further processing and analysis.

$$E(x) = -\sum_{\forall x} p(x)\log_2 p(x) \tag{3}$$

The higher value of entropy represents higher forensic relevance of the retrieved sector or the region. The idea is to traverse maximum number of data sectors for deciding the valid data region boundaries. In this direction the CTS numeric parameters of the disk are utilized in generation of the population or trial vectors that is required for DE algorithm. The initial population or trial vector is generated with the selection of random values of CTS and computation of corresponding entropy value for accessed sector, respectively. Here, the initial population size is fixed to 50 hence, it is not guaranteed that all the individuals from population will have entropy greater than 0. The presence of insignificant sectors (null sectors) varies the size of the next generation of population. However, as the number of iteration increases the size of population saturates to the desired number of individuals. In this way each individual and population consists of cylinder, track, sector numbers and corresponding fitness value for analysis of relevant data regions. Moreover, the completion of specified iterations and run of DE algorithm provides retrieved data sector locations having high entropy value. The retrieved data sectors can be manually examined in order to finalize the data relevant regions of the disk drive. The basic DE sub-operations with respect to the storage drive are briefly described as follows:

- Mutation: Each and every individual of the current population generates their corresponding trial vectors with the help of mutation operator. The target vector is mutated with weighted differential for generating the trial vector. Every offspring is the outcome of the crossover operation after using the recently generated trial vector. For example, let us consider N is the generation counter index, the mutation operator for trial vector generation $M_x(N)$ from the parent vector $P_x(N)$ for detailed discussion on the mutation operation as followed below:

1. The individuals from a population is a function of $M_x(N) \in f(C, T, S)$ and $P_x(N) \in f(C, T, S, E)$
2. A target vector $P_{x1}(N)$ is selected from the population, such that x ≠ x1 where, [x,x1] ∈ {C, T, S} one at a time
3. The two individuals P_{x2} and P_{x3} are also randomly selected from the population such that, x ≠ x1 ≠ x2 ≠ x3 where, [x, x1, x2, x3] ∈ {C, T, S} one at a time
4. The mutation operator proceeds the calculation of next trial vector once the target vector is mutated as described below:

$$M_x(N) = P_{x1}(N) + \underbrace{F \times \overbrace{(P_{x2}(N) - P_{x3}(N))}^{\text{Variation component}}}_{\text{Step size}} \tag{4}$$

where the mutation scale factor is F ∈ (0, 1) that controls the amplification of the differential variation [24].

- Crossover: Offspring $P'_x(N) \in f(C, T, S, E)$ is generated using the crossover of parent vector, $P_x(N)$ and the trial vector, $M_x(N)$ as follows:

$$P'_{xy}(N) = \begin{cases} M_{xy}(N), & \text{if } y \in Y \\ P_{xy}(N), & \text{Otherwise.} \end{cases} \tag{5}$$

The set of crossover points is represented by Y. Alternatively we can state that, Y is the points that will follow perturbation, $P_{xy}(N)$ which is the *yth* element of the vector $P_x(N)$.

- Selection: The selection operator consists of two functions as follows:

1. Selection of the individual for the mutation operation in order to generate the trial vector
2. Selection of the best among the parent and the offspring based on their corresponding fitness value for the next generation of population.

The fitness value is the deciding factor for the replacement of parent from the population. Whenever the offspring has better fitness as compared to the parent then the parent is replaced from the population otherwise, the parent remains in the population.

$$P_x(N+1) = \begin{cases} P'_x(N), & \text{if} f(P'_x(N) > P_x(N)) \\ P_x(N), & \text{Otherwise.} \end{cases} \tag{6}$$

This ensures that the average fitness of the population does not deteriorate. Algorithm 1 illustrates the pseudo code for DE algorithm based significant data region identification strategy, where scale factor is F, CR is the crossover rate, CTS are randomly selected geometric values of HDD and P is population vector.

Algorithm 1 Pseudo code for the DE based significant data region identification

```
Control parameters → number of iteration, CR and F are initialized;
Initial population $P(N) \in f(C, T, S)$ is generated;
while stopping criteria(s) $\neq$ true do
    for each individual $P_x(N) \in$ Population($N$) do
        Evaluate the fitness, $f(P_x(N))$ = Entropy(sector bytes);
        Trial vector is created by using the mutation operator $M_x(N)$;
        Offspring $P'_x(N)$ is created by applying the crossover operator;
        if then$f(P'_x(N))$ is better than $f(P_x(N))$
            Add $P'(N)$ to Population($N + 1$);
            Memorize the individual and related parameters for best fitness value
        else
            Add $P_x(N)$ to Population($N + 1$);#No fitness value is better than previous value
        end if
    end for
end while
Return all the individual with the best fitness value corresponding to each run as the solution;
```

The proposed approach creates output file corresponding to each run with several iterations of DE algorithm. The investigator can easily interpret the output files and examine only the reported regions of the drive for further analysis. However, it is necessary to take care of total DE evaluation in terms of number of run and iteration such that particular regions of HDD can be significantly analyzed. If the size of HDD is small the decision can be achieved using less number of DE evaluations whereas the large size of HDD requires more number of DE evaluations. A trade-off exists among size of storage drive, number of DE evaluation and accuracy. Along with the number of evaluation other parameters like CR and F also plays vital role in providing feasibility to solve real time problems. Similarly, in this chapter CR and F act as a critical parameter to decide overall efficiency of the analysis to identify data relevant region of the storage drive.

4 Experimental Setup

In order to evaluate the efficacy of the proposed approach several experiments are conducted using DE based significant data region identification approach that aims to overall accelerate evidence processing phase of DF process. The proposed approach is designed and developed under Python 2.7.11 environment installed on Kali Linux 2.0 operating system working over a multi-core desktop. The experimental environments are governed by the following parameter settings of DE based proposed methodology:

- Size of the population NP = 50
- Mutation rate F = 0.5 (default value),
- Cross-over rate CR = 0.5, (default value)
- Maximum number of DE evaluations for which the default value is set to 200000. Similarly, the runtime error or the stopping criterion for the proposed approach are defined in Table 1
- The implicit number of runs = 20 and iteration = 200, and graphs are plotted using the best fitness of each run.
- Data sector read size for fitness computation is 512 bytes

The stopping criteria for proposed approach are tabulated in Table 1, the program execution halts when any of the given criteria is unsatisfied. The analysis for identification of data relevant region on HDD is performed using storage drive of different capacity with different DE parameters as shown in Table 2.

Moreover, different data volumes are stored in the considered storage drives. For example, our experiment utilizes completely filled, completely wiped or partially filled storage drive. Since, every evidence seized under forensic investigation is classified unless directed from judicial body. The demonstration of the proposed approach is presented using a real time synthetic case study as discussed and demonstrated in the later section.

Table 1 Stopping criteria for execution of proposed technique

S. no.	Error	Details
1	Unable to locate the storage drive	The error occurs whenever the storage drive is undetected or the tool is provided with incorrect device ID
2	Unable to traverse output directory	Provided output directory is inaccessible for storing output files
3	Invalid number of runs	The default number of *run* is modified to $run \leq 0$
4	Cross-over rate (CR) is invalid	The default value of *CR* is altered to $CR \leq 0$ & $CR > 1$
5	Mutation rate (F) is invalid	The default value of *F* is altered to $F \leq 0$ & $F > 1$

Table 2 Experimental environment results inclusive of DE parameters

Drive size	Parameters	Identified sector regions
4 GB	CR = 0.5, F = 0.5, Runs = 20, Iterations per run = 200	864263, 1420474, 1693669, 1828392, 2320737, 2718215, 3595197, 3844604, 3860357, 4033162, 4148844, 4273448, 4292235, 5514173, 5744091, 6381149, 6782653, 7306662, 7306662, 7435147
8 GB	CR = 0.6, F = 0.4, Runs = 20, Iterations per run = 300	10426, 12661
16 GB	CR = 0.7, F = 0.4, Runs = 25, Iterations per run = 400	42427, 2330292, 3009887, 3885245, 6720428, 6745560, 7060698, 8930829, 9653354, 9653354, 9653354, 10029472, 12401876, 12401876, 12401876, 13660138, 15406446, 16282954, 16436649, 20819511, 20882496, 20882516, 20973998, 21026847, 27336892
1 TB	CR = 0.7, F = 0.3, Runs = 30, Iterations per run = 500	143497698, 284615592, 309868202, 320808730, 323561273, 326772585, 353198522, 429059156, 430042193, 501248142, 568475352, 595183261, 677497286, 759557978, 783639441, 890793828, 909184718, 913227618, 939949440, 941416087, 982055020, 1127464244, 1192365960, 1201495925, 1564215420, 1715947574, 1738762014, 1777650051, 1788991651, 1791885375

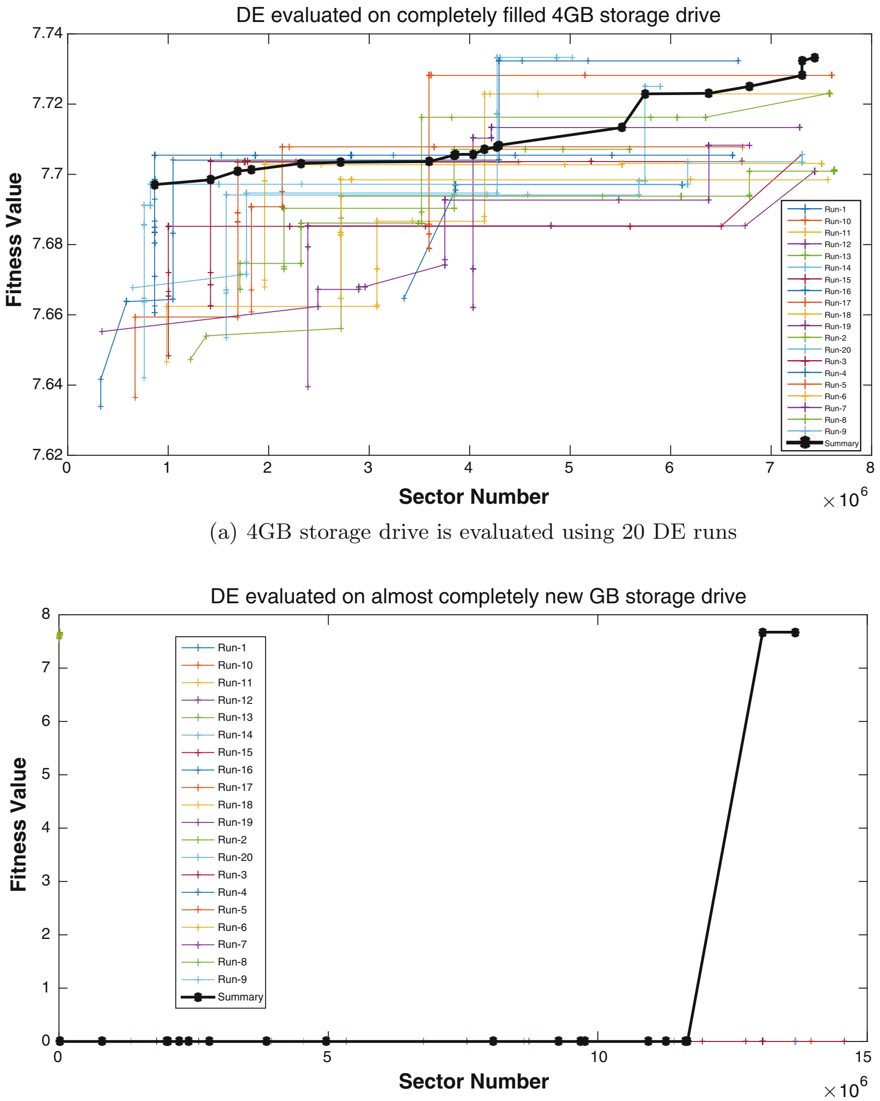
4.1 Experimental Results and Analysis

The storage drives with different capacities of 4 GB, 8 GB, 16 GB and 1 TB are used to validate the efficiency of the proposed significant data region identification approach. The proposed technique is evaluated against the considered storage drives using basic sector read size (512 bytes) and parameters as provided in Table 2. The result reveals that the proposed approach successfully reports the sector that consists of data with high entropy values among the processed sectors. Hence, there is high probability that the other sectors located around or close to the reported sectors also contains data having considerable entropy values. The consideration of other nearby sectors with respect to the extracted sectors, form a data region that can have forensic relevant artefacts which can be important for examination and analysis. The data sectors identified after execution of proposed approach are included in Table 2 which have highest entropy (fitness) value corresponding to particular run of DE.

Alternatively, investigator can also traverse the nearby sector locations corresponding to identified sectors for more advanced analysis. However, the drive regions which consist of null/irrelevant data are completely ignored until significant sector are identified. The experiments were performed on storage drives of 4 GB, 8 GB, 16 GB and 1 TB with varying DE parameters as shown in Table 2. Since, 4GB drive is completely filled with random data, more number of relevant sectors have been identified, which consist of almost all range of drive sectors, as illustrated in Fig. 3a. However, second experiment consist of 8GB disk drive which is almost empty or blank therefore, only couple of significant sectors have been identified while other ranges of disk sectors have been ignored due to presence of null data, as shown in Fig. 3b. Similarly, the experiments were performed on 16GB and 1TB disk drive for different parameters the result of which is illustrated in Table 2 and Fig. 4. The proposed technique determines the data sectors which have high entropy values irrespective of the amount of null data present in the suspected drive.

Furthermore, detailed insight into the proposed technique based on the experimental results is provided with the help of Figs. 3 and 4. The x-axis represents sector number while y-axis represents the fitness (entropy) value of corresponding sector data. Analysis on the referenced figures highlights the actual data storage pattern that exists within the suspected disk drive. The data pattern reflects the location and magnitude to data that actually exists within the storage drive. For example, the low entropy data exists at starting and mid of the storage drive whereas the high entropy data are located near the higher magnitude sector locations of the storage drives, as illustrated in Figs. 3 and 4.

The fitness value of initially generated population and final fitness value after completion of each run are also compared to analyze the diversity and convergence behavior of DE evaluation as illustrated in Figs. 5 and 6. The proposed approach identifies the significant data sectors even when a new storage drive is examined, as represented in Table 2 and Figs. 3b and 5b, where most of the fitness values converged to *zero* entropy and very few data relevant sectors have been identified. It might be possible that the reported sectors and its neighbor sector regions contains

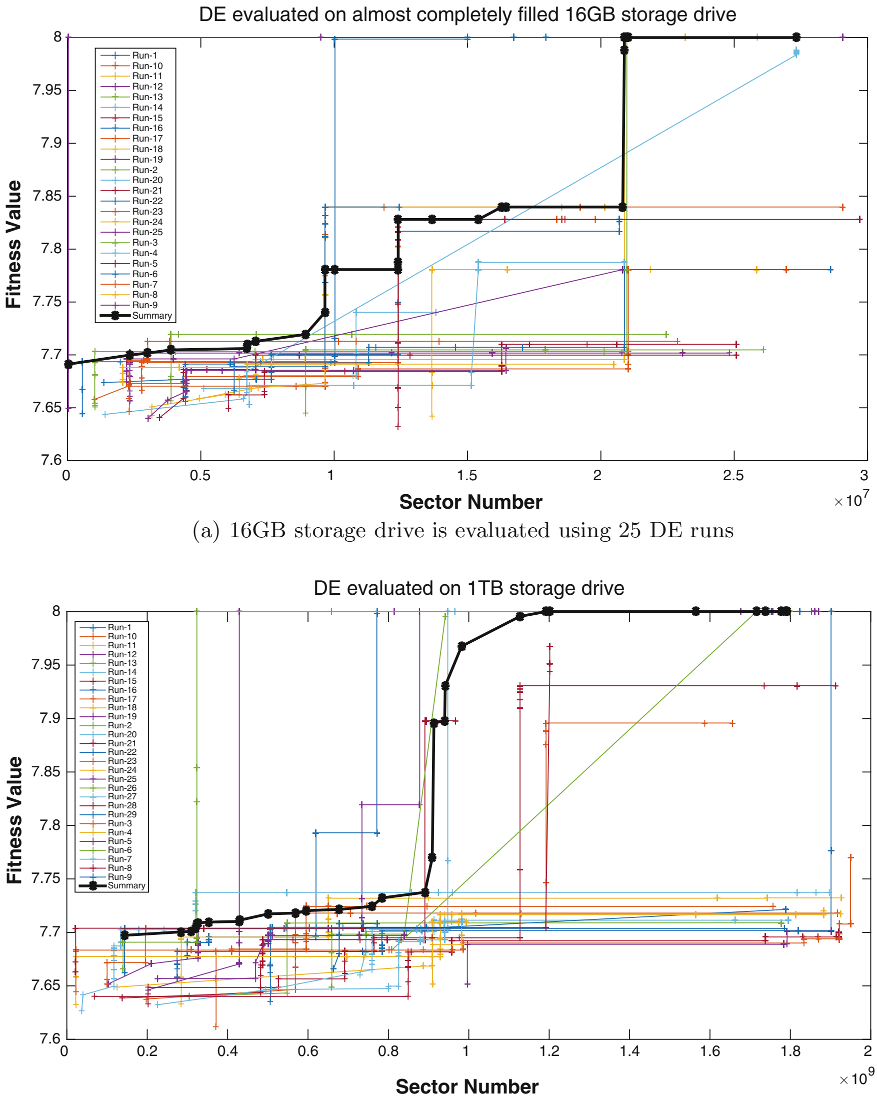


(a) 4GB storage drive is evaluated using 20 DE runs

(b) Twenty DE runs are evaluated on completely new 8GB drive

Fig. 3 Fitness values of each iteration is compared with best fitness values for every run of DE evaluation in sorted order. The analysis consists of disk drive with 4 GB and 8 GB storage capacity

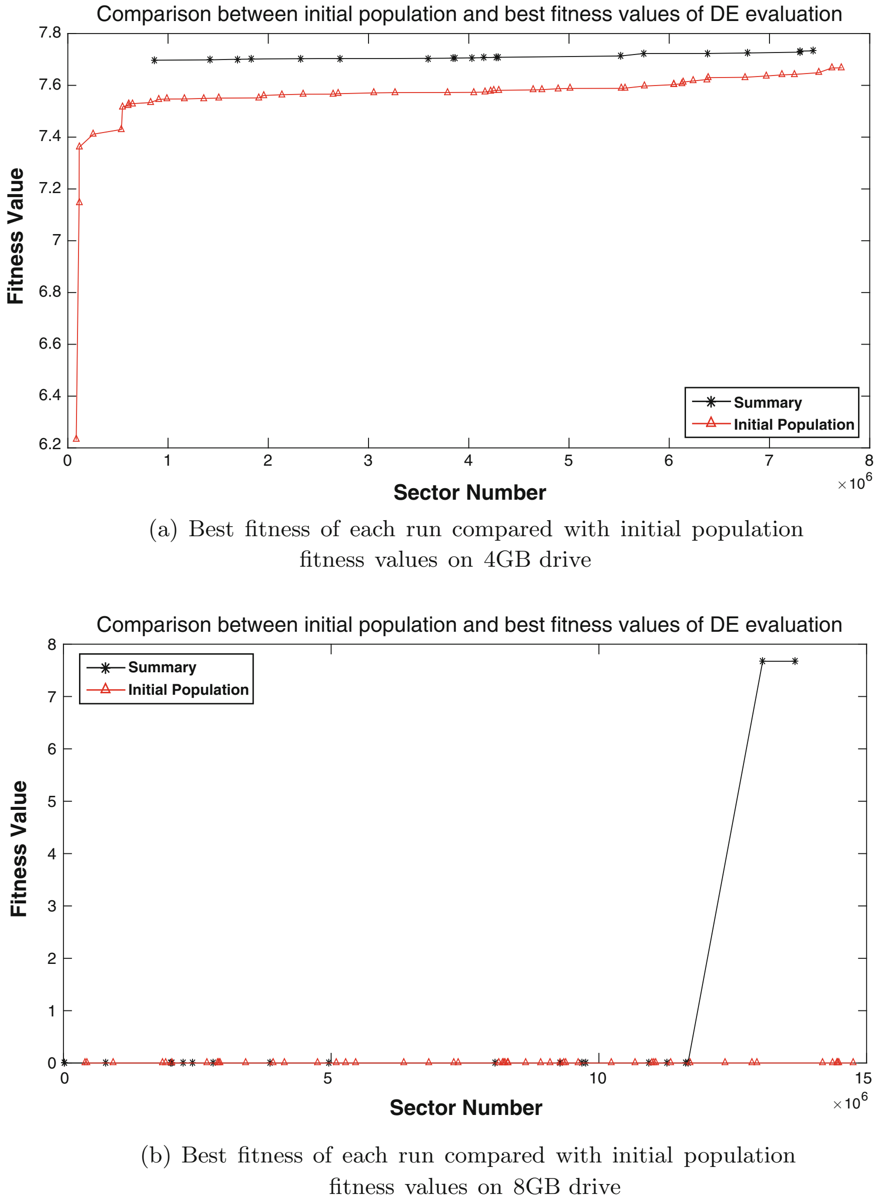
metadata or other vital information. Analysis of suspected HDD using the proposed methodology can comparably save examination time of digital investigator. Hence, the proposed approach can help in accelerating overall DF process and investigation by providing insight into the suspected HDDs.



(a) 16GB storage drive is evaluated using 25 DE runs

(b) 1TB storage drive evaluated using 30 DE runs

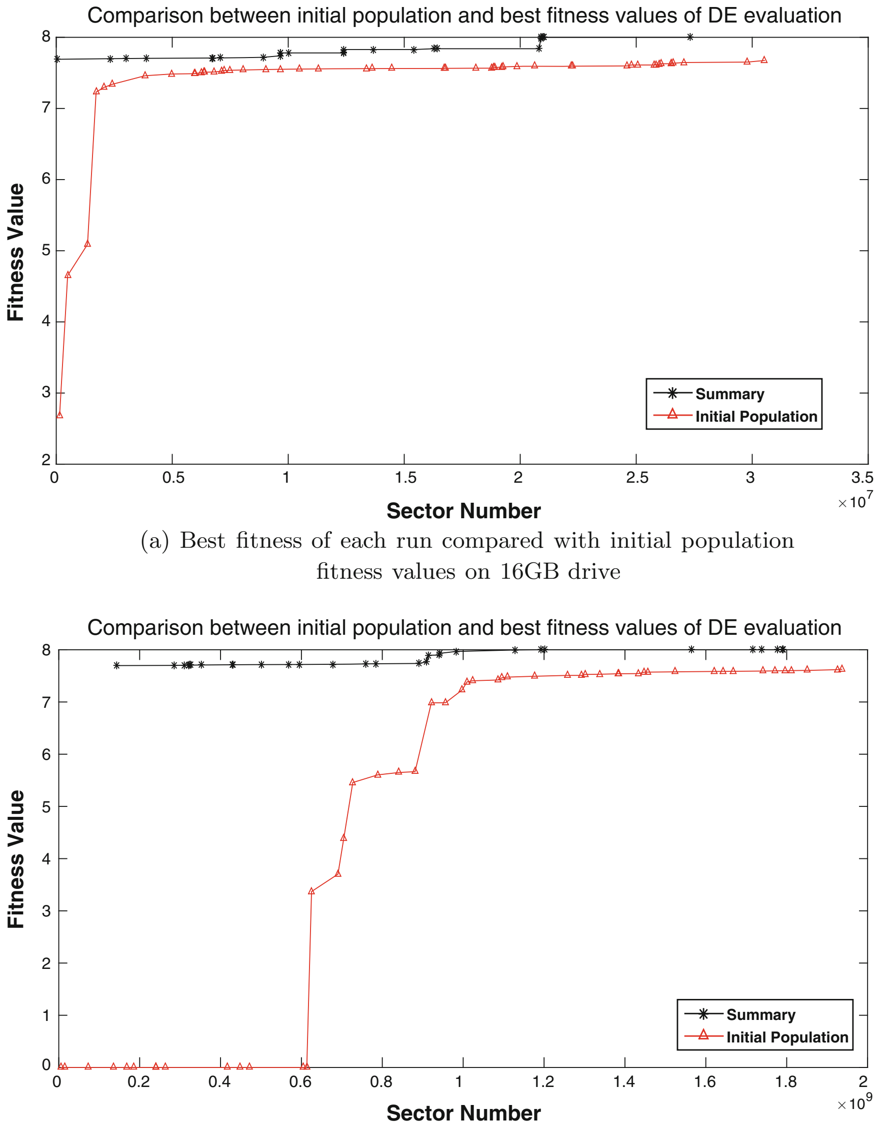
Fig. 4 For 16 GB and 1 TB storage capacity the fitness values of each iteration is compared with best fitness values for every run of DE evaluation in sorted order



(a) Best fitness of each run compared with initial population fitness values on 4GB drive

(b) Best fitness of each run compared with initial population fitness values on 8GB drive

Fig. 5 Comparison between fitness values of initially generated population and fitness values for each run of DE evaluation. The disk drive with different capacity have been analyzed **a** 4GB, **b** 8GB



(a) Best fitness of each run compared with initial population fitness values on 16GB drive

(b) Best fitness of each run compared with initial population fitness values on 1TB drive

Fig. 6 Comparison between fitness values of initially generated population and fitness values for each run of DE evaluation. The disk drive with different capacity have been analyzed **a** 16 GB and **b** 1TB

4.2 Case Study: Examination of Formatted Storage Drive

Two storage drives having storage capacity of 16 GB and 1 TB are suspected to contain obscene material (child pornography video). Digital investigator has seized the suspected drive for examination. Unfortunately, the seized drives (16 GB and 1 TB) were completely formatted before their seizure. The duty of the investigator is to identify the sensitive materials in order to approve or disapprove the allegation of suspected person.

During investigation the investigator is unaware of the actual contents of the storage drive. Additionally, it is unpredictable that the suspected drive is either properly wiped or completely or partially filled with data. The proposed technique which utilizes the storage drive structural information with other static parameters, such as CR = 0.4, F = 0.6, Runs = 20, NP = 100 and iteration = 300, as an input arguments to DE algorithm for identification of significant data region of the suspected drive. The proposed technique successfully identifies the existence of high entropy data even form the formatted suspected drive. The identified significant regions is provided in Table 3 where, the proposed technique successfully identifies high entropy data regions irrespective of analysis on the formatted drive. As an investigator instead of processing every byte of suspected drive, if identified sector and corresponding regions are examined, significant saving in investigative time and resources can be achieved. The region is extracted with the help of upper and lower bounds of identified sectors which can be represented with $[2^{\lceil log_2(\text{Sector Number})\rceil}$

Table 3 Case study analysis using proposed methodology

Drive size	Identified sectors	Significant regions
16 GB	17988238, 18027376, 18061904, 18074590, 18075466, 18085326, 18198933, 18296372, 18427476, 18478941, 18571921, 18912875, 18945782, 18961698, 19031569, 19036241, 19100409, 19100830, 19270940, 19338096, 19346907, 19362214, 19397889, 19398333, 19398433, 19469614, 19603038	16777217-to-33554431 sector ranges
1 TB	256747015, 346142300, 368445042, 554576085, 818309559, 867492316, 894690345, 900264512, 938450979	134217729-to-268435455, 268435455-to-536870911 and 536870912-to-1073741823 sector ranges

and $2^{\lfloor log_2(\text{Sector Number})\rfloor}$] respectively, in such a way that no important data can be missed. Hence, the proposed algorithm helps investigator to target the significant regions of the disk during the initial phase of investigation. The examination of identified sector region has revealed the existence of desired data instead of processing the complete disk drive for the investigative analysis. Thus, with the help of DE algorithm significant saving in evidence processing and analysis time can be achieved.

5 Discussion and Future Scope

The continuous advancement in the technology development from past few decades in coordination with increasing volume of data resulted in huge backlogs of digital investigation case and increase in evidence processing time. The significant data region identification method outlined in this work highlights the opportunity to elevate the processing speed of large storage drives forensic examination by utilizing storage drives structural information and differential evolution algorithm. The proposed approach can be applied to the forensic images and physical media to overall reduce the evidence processing time. The comparative summary of functionalities in existing approaches and proposed technique is provided in Table 4. The Table 4 does not compare the methods via experimentation, but variety of approaches with differing methods that were discussed in this chapter is highlighted. The general approaches of various methods were highlighted using the comparison. The future consideration and detailed comparison would be to undertake more insight into the highlighted literature of this chapter.

The proposed significant data region identification method using differential algorithm have demonstrated the potential to speed-up the evidence processing in comparison with full examination of large storage drives. However, the proposed approach is not meant to replace full evidence examination, and there are majority of circumstances where full evidence examination is mandatory. In this paper for the first time computational intelligence paradigm has been utilized to provide vision towards achieving fast digital forensic investigation results. The proposed trade-off rectangle lists out number of problems that need to be focused for advancement of existing digital forensic technology. The major development is required towards the development of novel DF methodology/tools/software in order to mitigate excess resource utilization, evidence acquisition and processing delay and so on. In this chapter we used differential evolution algorithm as a core decision maker, but it is also possible to utilize other evolutionary algorithms that are available under computational intelligence paradigm. The proposed technique outlined in this work locates the forensically significant region of the suspected drive. During investigation process, investigators are not provided with the internal details of suspected disk drives. Hence, the use of proposed technique prior to examination phase can provide data pattern as well as insight into the disk drive. Moreover, performance analysis of the proposed approach with alteration of standard DE parameters can be

Table 4 Comparative summary of existing work and proposed approach functionalities

Functions and methodology	Proposed approach	Full examination	Triage solutions	Garfinkel fast disk analysis	Sifting collector approach	Monte Carlo based search
Significant sectors and region identification	√	√	∗	X	X	X
Processing of forensically relevant data from deleted/formatted drive	√	√	−	√	X	X
Classify null and high entropy data region	√	X	X	√	X	X
Processing random sector samples	√	X	∗	√	X	√
Target/specific data/evidence processing	X	√	−	√	√	√
Ability to ignore null data and sectors	√	X	∗	∗	X	X
Human in the loop	∗	√	∗	−	√	√
Header based search	X	∗	∗	√	−	−
Reliable to physical media and mounted image	√	√	√	√	∗	√
Utilizes structural information of physical media	√	X	X	X	X	X
Not Specific to one file system	√	√	√	√	X	√
Speed-up acquisition, examination and analysis	√	X	√	√	√	√

(√) Addressed issue
(X) not addressed
(∗) may be applicable
(−) not necessarily applicable

next step of analysis. The performance and accuracy of the evolutionary algorithm varies with different parameters such as crossover, mutation, number of iterations etc. Therefore, the proposed technique is also provided with the flexibility to customize the execution characteristics and parameters of DE algorithm according to the user need. Hence, the declaration and definition of standard values i.e. CR, F, number of runs and iteration, and population size with respect to storage capacity of drive can be another dimension for future work. The consideration of various storage characteristics such as completely filled drive, completely wiped drive, formatted drive, deleted drives, contiguous file storage and fragmented file storage, can also be utilized for future extension of the proposed methodology. The limitation of the proposed work comes into picture whenever the suspected drive is encrypted or encoded; as a result every sector is identified as significant sector. The encrypted drive can make the proposed method ambiguous. The enhancement of the proposed approach to user interface is also left for future work. The study of the affect of sector read size other than 512 bytes over the proposed methodology is also another dimension of research.

6 Conclusion

In this chapter we presented a new trade-off rectangle to reveal present requirements towards the development of existing DF facilities. The chapter also proposes a methodology that extract data storage pattern using storage drive's structural information and DE algorithm that can help investigator in planning further course of action. Significant saving in investigation time and resource utilization can be achieved by the investigator with the help of proposed technique. The proposed approach also identifies the significant data sectors within storage drives. Although modern HDD provides good storage capability, the proliferation of these devices has also created new hurdles for digital investigators. Presently, the traditional digital forensic process requires considerable time for performing investigative task. This chapter emphasizes on the identification of data relevant sector regions in HDD using computationally intelligent DE algorithm to accelerate the overall DF process. The proposed method enables the examiner with insight of relevant data sector location and data intensity that is present within HDD such that, the investigator can directly access the specified data regions for investigative examination and analysis. Since, high entropy reflect maximum information content the proposed approach provides data locations which have globally high entropy among all the sectors addressed within HDD for particular iteration of DE algorithm. It is preferable to increase the number of iterations with the increase in size of HDD and vice-versa. The proposed approach provides acceptable regions of the suspected HDD, which is under investigative consideration. The further search and analysis can be narrowed down and continued manually once the data relevant regions are identified. Hence, the proposed technique can be of great help to the digital forensic community for fast evidence examination.

References

1. M.G. Williams, A risk assessment on Raspberry Pi using NIST standards. Int. J. Comput. Sci. Netw. Secur. (IJCSNS) **15**(6), 22 (2015)
2. D. Quick, K.K.R. Choo, Big forensic data reduction: digital forensic images and electronic evidence. Springer Cluster Comput. 1–18 (2016)
3. D. Quick, K.K.R. Choo, Impacts of increasing volume of digital forensic data: a survey and future research challenges. Elsevier Digit. Investig. **11**(4), 273–294 (2014)
4. V. Roussev, C. Quates, R. Martell, Real-time digital forensics and triage. Elsevier Digit. Investig. **10**(2), 158–167 (2013)
5. A. Shaw, A. Browne, A practical and robust approach to coping with large volumes of data submitted for digital forensic examination. Elsevier Digit. Investig. **10**(2), 116–128 (2013)
6. J. Grier, G.G. Richard, Rapid forensic imaging of large disks with sifting collectors. Elsevier Digit. Investig. **14**, S34–S44 (2015)
7. S.L. Garfinkel, Carving contiguous and fragmented files with fast object validation. Elsevier Digit. Investig. **4**, 2–12 (2007)
8. S.L. Garfinkel, A. Nelson, Fast Disk Analysis with Random Sampling (2010)
9. N. Kishore, B. Kapoor, Faster file imaging framework for digital forensics. Procedia Comput. Sci. **49**, 74–81 (2015)
10. F. Adelstein, Live forensics: diagnosis your system without killing it first. Commun. ACM **49**(2), 63–66 (2006)
11. D. Ayers, A second generation computer forensic analysis system. Elsevier Digit. Investig. **6**, S34–S42 (2009)
12. S.L. Garfinkel, Digital forensics research: the next 10 years. Elsevier Digit. Investig. **7**, S64–S73 (2010)
13. N. Beebe, J. Clark, Dealing with Terabyte Data Sets in Digital Investigations, in *Advances in Digital Forensics* (Springer, 2005), pp. 3–16
14. J. Dalins, C. Wilson, M. Carman, Monte-carlo filesystem search—a crawl strategy for digital forensics. Elsevier Digit. Investig. **13**, 58–71 (2015)
15. G. Palmer et al., A Roadmap for Digital Forensics Research, in *Forst Digital Forensics Research Workshop*, Utica, New York (2001), pp. 27–30
16. R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
17. S. Das, A. Konar, Two-dimensional IIR filter design with modern search heuristics: a comparative study. Int. J. Comput. Intell. Appl. **6**(03), 329–355 (2006)
18. P.K. Liu, F.S. Wang, Inverse problems of biological systems using multi-objective optimization. J. Chin. Inst. Chem. Eng. **39**(5), 399–406 (2008)
19. T. Rogalsky, S. Kocabiyik, R. Derksen, Differential evolution in aerodynamic optimization. Can. Aeronaut. Space J. **46**(4), 183–190 (2000)
20. M.G. Omran, A.P. Engelbrecht, A. Salman, in *2005 IEEE Congress on Differential Evolution Methods for Unsupervised Image Classification*, vol. 2 (IEEE, 2005), pp. 966–973
21. J. Vesterstrom, R. Thomsen, A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, in *Congress on Evolutionary Computation, 2004. CEC2004*, vol. 2 (2004), pp. 1980–1987. doi:10.1109/CEC.2004.1331139
22. J. Kennedy, R. Eberhart, Particle swarm optimization, in *IEEE International Conference on Neural Networks, 1995. Proceedings*, vol. 4 (1995), pp. 1942–1948. doi:10.1109/ICNN.1995.488968
23. J.H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (U Michigan Press, 1975)
24. A.P. Engelbrecht, *Computational Intelligence: An Introduction* (Wiley, 2007)