AI & Soc (2016) 31:265–274
DOI 10.1007/s00146-015-0591-x

OPEN FORUM

# On the applicability of diploid genetic algorithms

**Harsh Bhasin**[1] · **Sushant Mehta**[2]

Received: 1 July 2013 / Accepted: 13 July 2015 / Published online: 28 August 2015


**Abstract** The heuristic search processes like simple genetic algorithms help in achieving optimization but do not guarantee robustness so there is an immediate need of a machine learning technique that also promises robustness. Diploid genetic algorithms ensure consistent results and can therefore replace Simple genetic algorithms in applications such as test data generation and regression testing, where robustness is more important. However, there is a need to review the work that has been done so far in the field. It is also important to analyse the applicability of the premise of the dominance techniques applied so far in order to implement the technique. The work presents a systematic review of diploid genetic algorithms, examines the premise of the dominance relation used in different works and discusses the future scope. The work also discusses the biological basis of evaluating dominance. The work is important as the future of machine learning relies on techniques that are robust as well as efficient.

**Keywords** Diploid genetic algorithms · Robustness · Dominance · Allele

✉ Harsh Bhasin
i_harsh_bhasin@yahoo.com

Sushant Mehta
sushant0523@gmail.com

[1] Department of Computer Science, Jamia Hamdard University, New Delhi, India

[2] Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani, India

## 1 Introduction

The scientific community has developed numerous computational procedures and algorithmic approaches to accomplish assorted tasks from development of automated vehicles to the analysis of patterns to implement artificial creativity. The next step is to make software that implements these algorithms and approaches as efficiently as possible. One of the prime goals of designing these softwares is to achieve optimization. For example, in order to fetch results, a crawler evaluates similarity of a web page in its own repository with a query that is entered by the user. The pages are then ranked in the order of similarity evaluated by the crawler. The crawler displays those web pages that have higher similarity first, thus making it an optimization problem. There are many such situations in which the prime aim is either to maximize the profit or to minimize the cost. In such situations, simple genetic algorithms (SGAs) come to our rescue. SGAs are heuristic search processes, which are based on theory of the survival of the fittest (Mitchell 1996; Goldberg 1989).

However, in this quest for optimization, the thirst of robustness has as yet not been quenched. In order to achieve robustness, another variant of SGAs is used, which is referred to as diploid genetic algorithm (DGA). DGAs are known to be robust and thus are capable of giving consistent results. Robustness is needed in many practical situations. For example, automated vehicles and assembly line scheduling call for a very high degree of consistency. Robustness will ensure that proper execution of the task is given priority over mere speed. Therefore, robustness is much more important than optimization in such cases. The implementation of DGA requires dominance to be evaluated. However, there are ambiguities in the determination of dominance in DGAs. In the literature review, presented

in the following sections, it was observed that random or pseudo-random techniques were adopted in several cases to determine dominance. Such techniques do not lend credibility to the results obtained. The proper implementation of DGAs and their comparison with SGAs will be possible only if such ambiguities are resolved.

The work intends to summarize and critically evaluate the existing works in the domain. Hence, finding out the gaps and suggesting some possible solutions are the primary goals of the review. The work also presents the future scope and suggests the applicability of DGAs in various fields including software engineering.

One of the potential applications of DGAs can be test data generation. Test data generation is needed in order to test software. However, manual test data generation requires a lot of time; therefore, automated test data generation becomes important. In automated test data generation test, cases are crafted as per the paths of the control flow graph of the program under test. Since there can be many paths and hence the number of test cases can be too large, SGAs may be used in order to facilitate the search process and achieve optimization. In one of our earlier works, genetic algorithms were used to generate test cases (Bhasin and Singla 2013a). The problem, however, was that the results were not consistent. During the implementation of the work, it was realized that SGA can only provide optimization. The present work indents to provide a framework to overcome that hurdle. It may be noted that in test case generation, robustness is more important as compared to optimization. This is because the generation of a complete and comprehensive test case suite is a much more important issue than the time taken for generation. The optimization algorithms can save time but in that process quality at compromised. The work discusses many such possible applications of DGAs.

The rest of the paper has been organized as follows: Section 2 presents the research methods, Sect. 3 explains the biological background of DGAs, Sect. 4 explains the review process, Sect. 5 answers the research questions, Sect. 6 presents the future scope, and Sect. 7 concludes the review.

## 2 Research method

### 2.1 The need of review

The present study intends to summarize the works that concentrate on the concept of DGAs and their applications. The study also intends to analyse the benefits of DGAs against SGAs. In order to do so, the existing approaches have been studied and analysed. The study also intends to explore the basis of the applicability of DGAs in fields such as software engineering.

The review is undertaken in accordance with the techniques proposed by Kitchenham (2012). These techniques lay emphasis on clearly establishing the need for review, conducting a rigorous assessment of existing literature, explicit documentation of the search strategy, formulation of the right research questions and avoiding publication bias. The reason for following these guidelines is to reduce the probability of the review being biased and to ensure the quality of the review. During the course of the study, many variations in the studies were observed. The study also points out the probable causes of these variations.

It may be pointed out at this stage that DGAs is a relatively new technique and it has seldom been applied in algorithmic or testing problems. However, it is important to summarize the existing evidence in order to streamline the future studies. Therefore, the work presents the state of the art and explores the trends in the discipline.

### 2.2 Sources of information

In order to have a clear, unbiased, broader and complete perspective, many sources have been searched. The need of exploring the databases has been explained by Kitchenham and others (2012). The following databases have been searched:

- IEEE Explore
- ACM Digital
- Science Direct
- Springer
- Wiley Online

The journals with high impact factors and quality searches were considered. The review also takes into account the conference proceedings. However, except for the papers in the above databases, some other papers were also considered owing to their importance. It may be noted that many duplicates cropped up during the search process, which were manually removed.

### 2.3 Search criteria

The initial search criterion was to include all the papers related to DGA. The keyword used was <Diploid Genetic Algorithms>. To further search within these search results, the keywords <Computer Science>, <Dynamic Environment>, <Optimization>, <Dominance> and <Alleles> were used. Papers written in English were studied. Irrelevant results pertaining to theoretical biology, job scheduling problems, etc. were filtered. The results are summarized as follows:

Search of the keyword <Diploid Genetic Algorithms> displayed 429 results in ACM Digital Library (10 September 2014). The papers were ordered with respect to the year of publication. On further examination, all these papers but four were filtered out as they were from the fields such as speciation, molecular biology, haplotyping algorithms and molecular genetics. Figure 1 depicts the selection process.

Wiley Online Library displayed 3579 results when <Diploid Genetic Algorithms> was searched (3230 in journals, 333 books and 16 laboratory protocols). The keywords <Computer Science> and <Algorithms> were used to further refine the search. This resulted in 1395 results. Papers pertaining to irrelevant fields such as cytometry, ecology, cancer and plant breeding were removed, and the final set had 41 results. The abstract of these papers were studied, and almost all the papers were deemed irrelevant. Figure 2 depicts the selection process. When the search was carried out in publication title, there was no match. However, on searching the keywords for <Diploid Genetic Algorithms> 2 results were found, out of which one was selected.

Science Direct showed 40,353 results with the keyword <Diploid Genetics> (33,558 journals, 6082 books and 713 reference works). Adding the keywords <Algorithms> and <Computer Science> refined the search to 1014 results (895 journals, 187 books and 23 reference works). Excluding results from journals such as cell, phylogenetics, theoretical biology and immunology resulted in a final set of 12 articles (10 journals and 2 books). The abstract of these journals were read, and finally one relevant paper was selected. The process is depicted in Fig. 3. It may be stated here that the search with the keyword <Diploid Genetic Algorithms> did not show any result.

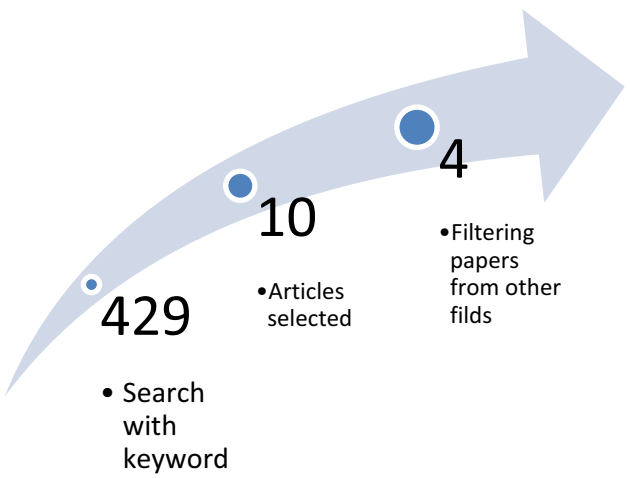


**Fig. 1** Selection of papers from ACM Digital Library

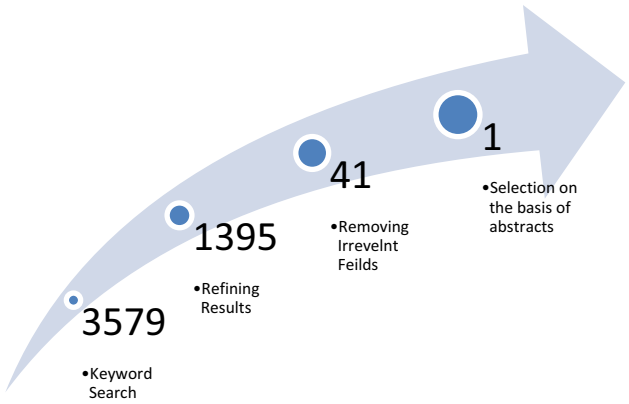


**Fig. 2** Selection of papers from Wiley database

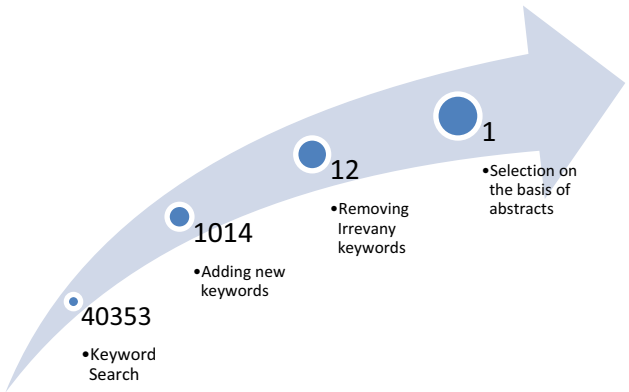


**Fig. 3** Selection of papers from Science Direct database

IEEE showed 36 results with the keyword <Diploid Genetic Algorithms> (32 conferences and 4 journals). The papers pertaining to irrelevant results pertaining to topics such as scheduling and hydro-thermal systems were filtered out. The process is depicted in Fig. 4. Here, it is worth stating that the count of the papers published in 2014 (as on 10 September 2014) was just one.

The search of the keyword <Diploid Genetic Algorithm> on Springer generated 4416 results. However, only a few results were found to be relevant after filtering.

A group of experts was formed in order to select relevant papers from among the above papers. Finally, 13 papers, using diploid genetic algorithms in computer science, were selected.

### 2.4 Study selection

Papers were filtered on the basis of titles. This was followed by segregation on the basis of keywords. Then, the abstract of the remaining papers was studied. The final set of papers was finalized after reading the complete text. The final set had 13 relevant papers. The predominant techniques that were employed were based on adaptive dominance, dominance

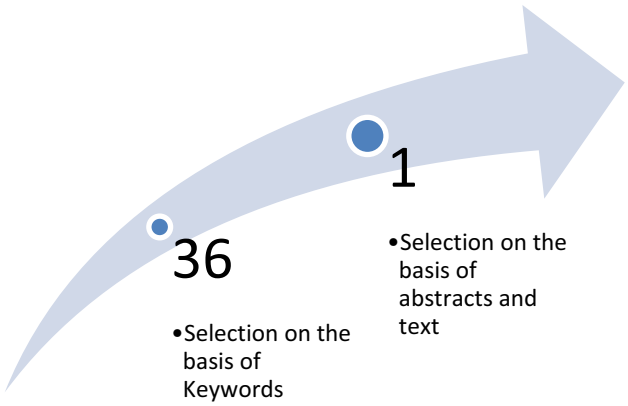


**Fig. 4** Selection of papers from IEEE database

learning mechanisms and random dominance. Twelve techniques were extracted out of these 13 papers.

### 2.5 Data extraction

As per the guidelines by Kitchenham, data were collected in the most scientific way. The quality of the papers was the foremost criterion in the paper selection. Twelve techniques were identified from 13 papers for a detailed comparative study. The data were classified as explained in Sect. 4.

### 2.6 Research questions

The systematic review intends to classify the work in the diploid genetic algorithms by proposing the answer to the set of questions presented below. The papers were selected in accordance with the following research questions. The aim of the work is to identify the gaps in the existing methodologies and propose answers to fill the gaps. The questions we wish to answer are as follows.

*RQ1* What is the present state of the art?

*RQ2* Can the applicability of the technique be theoretically justified?

*RQ3* What is the percentage of papers published of DGAs in various journals and what are the applications of these DGAs?

*RQ4* What are the advantages, if any, of using DGAs over SGAs?

The above questions form the basis of the review.

## 3 Biological basis for dominance

The understanding of dominance is required in order to implement DGAs, and therefore, it is essential to examine the biological basis of dominance. This section briefly explains these concepts.

Deoxyribose nucleic acid (DNA) is a polymer that is responsible for long-term storage of genetic information. It consists of complementary chains of nucleotides. Each nucleotide consists of a sugar molecule, a phosphate group and any one of the four bases—adenine, cytosine, guanine and thymine (Russell 2001; Saenger 1984).

The chemical polarity of DNA is determined by the order in which nucleotides are arranged together. Gene is a name given to a stretch of DNA that codes for a polypeptide (Watson and Crick 1953; Alberts et al. 2002). Genes in DNA preserve biological information for the next generation. It may be stated at this point that the mapping between the nucleotides and amino acids of the protein is intricate and has taken decades to decode.

Ribose nucleic acid (RNA) is a biological molecule that is single-stranded and, like DNA, consists of four types of nucleotides, with uracil replacing thymine (Higgs 2000; Tinoco and Bustamante 1999).

Information in DNA is not converted to proteins directly. First, DNA is transcribed to messenger RNA (mRNA), a special type of RNA. This process is known as transcription (Hausner and Michael 2001; Berg et al. 2006; Solomon et al. 2004). Proteins are then synthesized with the help of mRNA. Various other bio-molecules such as ribosomes are also involved in this process (Nissen et al. 2000). This process is known as translation (Campbell 1996; Freeman and Griffiths 2008). The whole process of transcription and translation is referred to as the central dogma of molecular biology (Crick 1970).

Genotype is the genetically stored information that carries the instructions for formation and maintenance of an organism. For example, Tt may denote the genotype of an organism, where T denotes the physical characteristic of tallness and vice versa.

Phenotype is the outward, physical manifestation of the genotype. For example, the genotype of Tt results in the phenotype of tallness, because the allele 'T' exhibits something known as dominance, which is explained below.

DNA is transcribed to RNA, which is translated to form proteins. This is how genotype is mapped to phenotype, since phenotype depends primarily on proteins (Russell 2001). Dominance results when one of the two alleles at a locus is non-functional. Thus, only the protein product of the other gene is formed and results in the corresponding phenotype.

For example, when the genotype is Tt, where T denotes the phenotypic characteristic of tall height and vice versa, T gene is transcribed to RNA which is further translated to a protein, which plays some role in increasing height, for instance HGH (human growth hormone). As a result of formation of this protein, person gains height. Hence, the gene T is said to be dominant because its corresponding

phenotype is shown in the organism. Figure 5 shows homologous pairs of chromosomes.

Another dominance scheme observed in nature is called codominance. Codominance is observed when a heterozygous organism shows traits of all alleles. The concept of dominance can be understood by the example given in Fig. 6, in which B is dominant and A is recessive. Here, in the table depicting the first cross, the cross would always lead to AB. However, in the second cross when A and A combine the result would be an AA and when B and B combine the answer would be BB.

The theoretical basis of DGAs in computer applications has already been elaborated in one of our earlier works (Bhasin et al. 2015).

## 4 Review

An extensive review was carried out in order to find out the work that has been done in the discipline. It is necessary in order to place diploid genetics in the right perspective and explore the possibility of it being used as another major machine learning tool, which will provide robustness as well. The following section briefly examines the various theories and experiments carried out taking DGAs as the base.

Goldberg was among the pioneers in this field (1987a, b, 1989). As he explains, in complex organisms, diploid chromosomes are present in which each genotype contains one chromosome or a pair of chromosomes (Goldberg 1989). It is therefore important to understand the mechanism, which decides which allele should be selected. Different

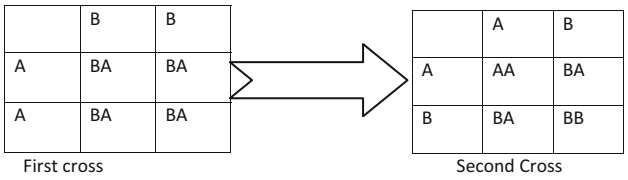


**Fig. 6** Dominance

researchers have proposed different mechanisms in the case of ambiguity. A genetic operator called dominance helps to decide which allele is dominant. It was observed by many eminent scientists that diploidy provides a way of remembering alleles that were previously useful. This is one of the prime reasons of the survival of organisms as their genes remember the favourable and unfavourable conditions and hence help them to adapt. One of the first applications of diploidy and mapping of genotype to phenotype was explained in Bagley's dissertation (1967). He proposed a scheme that could adapt with the environment but did not compare diploid and haploid schemes. Rosenberg conducted a biologically oriented study in which dominance was a part of biochemical interaction (1967, 1970a, b).

In 1971, the genes were segregated into two types: the modified gene and the functional gene. Functional gene had values 0 and 1, whereas the modified gene had values M and m (Hollstein 1971).

Yukiko and Nobue (1984) proposed a pseudo-meiosis genetic algorithm that preserves population diversity. Their algorithm was applied to non-stationary travelling salesman problem. Travelling salesman problem calls for finding the minimum cost Hamiltonian cycle in a given graph.

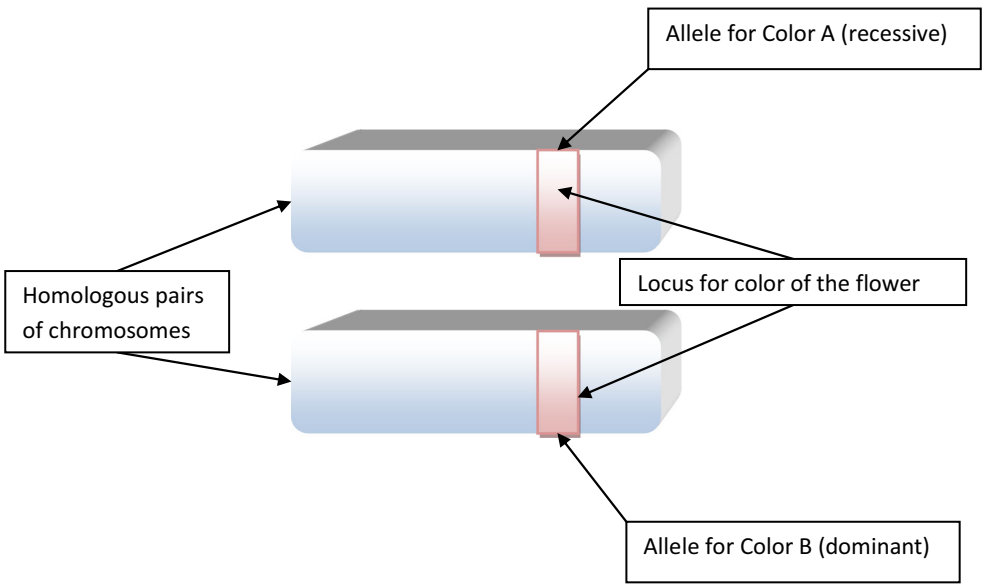


**Fig. 5** Homologous pairs of chromosomes

Actually, it is a problem that requires the selection of the correct permutation. The task has been accomplished by genetic algorithms, as they are heuristic search processes inspired by the selection of the fittest. The problem is also a search problem, and fitness can also be assigned to every path depending upon the cost associated with it. As per the work, high mutation and immigration can increase the population diversity. Redundancy was deliberately introduced so as to keep the population diverse.

As stated earlier, the choice of dominance scheme is the most crucial part of the implementation of diploid genetic algorithms. Ryan (1994) devised a dominance scheme, which was called additive diploidy. Here, the result is obtained by the application of the pseudo-arithmetic to the numbers denoting alleles to obtain the phenotype. His experiments proved that the technique outperforms other methods of binary genetic algorithms. The technique also allowed the use of non-stationary set of test cases. It may be noted at this point that polygenetic inheritance is made out to be superior to simple additive diploidy, but sufficient number of experiments were not carried out in order to prove this hypothesis.

In the experiment carried out by him, five alleles and two gene pairs were used. The calculation resulted in 70 different phenotypes, and the threshold value was taken as nine. His work was a continuation of another work, which was suitable only for variable length implementation (Hillis 1992).

There are many applications where getting a very good result a few number of times is not as important as getting a good average result. Here, consistency matters. The best result denotes optimization, and consistency means robustness. Lee and Rowlands (2005) stress the importance of robustness over optimization in solving real-world problems. Diploid genetic algorithms are used to select a more robust, local optimal solution over a less robust, more optimal one. The results, which use a bi-modal fitness function, demonstrate that the more robust, local optima are found almost 100 % of the time, which strengthens our premise that DGAs are the best candidates for solving problems, which require a high degree of robustness.

Schafer (2009) also asserts that SGAs are not good at continuous adaptation in a dynamic environment. He employs DGAs to a very novel class of problems, wherein maintaining population diversity is as important as maintaining fitness. The results show the ability of DGAs to create dynamic systems, which exhibit stability as well as fitness. Performance of SGAs, on the other hand, was shown to be significantly lesser than DGAs when frequent environment readjustments were made.

The Markov model explains a randomly changing system. The state of a system, in a Markov model, depends on the present state. The need of such model was realized, and Liekens et al. (2003) formally described DGAs. A Markov model of DGAs is provided, and a DGA is constructed by reusing the genetic operators of a haploid SGA. Conversely, transformation of a DGA into a SGA is also demonstrated. Thus, two ways are shown to design a DGA. Either it can be built directly according to the definition of a DGA, or it can be simulated inside a SGA. It is also proved that the behaviour of these two implementations, if implemented as described, will be same.

As mentioned earlier, Yukiko and Nobue (1984) deliberately introduced redundancy to increase population diversity. On somewhat similar lines, Weicker and Weicker (2001) explored the effects of redundancy on mutation and recombination. They investigated the combined effects of mutation and randomness on local optima and plateau points. The effect of redundancy on population diversity was observed in great detail. It was observed that redundancy encourages recombination and hence increases diversity to some extent.

To address optimization in dynamic environments, Yang et al. (2007) proposed an adaptive dominance mechanism. According to him, the cardinality of the genotypic alleles and the uncertainty in the dominance scheme are the two key factors that affect the performance of the technique (Yang 2006). He carried out experiments with the help of a tool he developed in one of his earlier works (Yang 2003; Yang and Yao 2005) and proved that his dominance scheme is better than the previous ones. It may be stated at this point that his dominance scheme was based on learning. A dominance probability vector is employed for mapping genotype to phenotype. The dominance probability vector starts off with a value of 0.5 for each locus, implying that alleles have equal chances of expression. Then, the dominance probability vector is updated every generation according to the current population. Learning rate can be altered by changing the sensitivity.

The effect of variation of learning parameter on the performance of his algorithm has also been demonstrated. However, the dynamic environment was created by his tool, and hence, the results do not instil confidence in the work.

It may be noted at this point that each gene may have several alleles. In order to carry out crossover, each individual offspring is independently subjected to bitwise mutation. The mutation may or may not result in change in the phenotype. The example of non-stationary knapsack problem also proves the point that fixed dominance or haploid structure does not perform as well as tri-allelic dominance.

Ng and Wong (1995) used 4 genotypic alleles: 2 dominant and 2 recessive. If the diploid genotype had 2 recessive or 2 dominant alleles, one of them was randomly chosen to be expressed. There are 4 genotypic

combinations with 1 dominant and 1 recessive gene. Out of these, 2 combinations were prohibited in the sense that if they occurred, the recessive gene was promoted to be dominant. The following table presents the summary of the review (Table 1).

## 5 Answers to research questions

*RQ1*

During the review, it was found that DGAs have not been explored in detail. In general, the Springer had the highest number of papers pertaining to DGAs. It may be noted that none of the papers reviewed applied this technique in the field of software engineering. The number of papers selected from various journals is mentioned in Sect. 2.

On searching diploid genetic algorithms in Springer Link (21 April 2015) and limiting the search to computer science-related literature, 355 links of chapters and 219 links of articles were displayed. Only 6 reference work entries and 4 protocols came up. As per the sub-discipline is concerned, 317 papers were on theoretical computer science, 262 on system biology, 255 on bioinformatics, 250 on artificial intelligence and 180 on computational science and engineering. Out of these papers, those published in the last 2 years explain the applications of diploidity in various fields.

The number of papers published (Springer Link) in the last two decades is shown in Fig. 7. It may be stated though that the trend of the number of papers on applications of diploid genetic does not necessarily indicate the decline in the interest. However, the trend is an indicative of the need of a comprehensive review and the summarization of the existing methodologies. This paper proposes to accomplish the above task.

As a matter of research, it was found that Branke's work is one of the most cited as per this topic is concerned (Branke 2001). The work by Lewis explains the concept of diploid genetic algorithms for non-stationary problems. The conclusions, though useful, are not necessarily applicable to dynamic environments (Lewis et al. 2004).

*RQ2*

There are some ambiguities associated with the calculation of dominance. In most of the cases, randomness or pseudo-arithmetic was used to determine dominance (Ryan 1994; Ng and Wong 1995). It is desired to end this ambiguity and come up with a dominance scheme that is theoretically sound, and inspired by nature at the same time. In our future work, it is intended to test this technique on dynamic environments and compare it with the existing

**Table 1** Summary of the review

| Ref. no. | Author | Technique proposed |
|---|---|---|
| 9 | C. Ryan | The work proposed additive diploidy and used pseudo-arithmetic to determine dominance |
| 18 | Ng, K.P. and Wong, K.C. | The work used four genotypic alleles and employed randomness when both alleles were dominant or both were recessive |
| 15 | S. Yang | The works states that in diploidy genetic algorithms, there are two key design factors: the cardinality of genotypic alleles and the uncertainty in the dominance scheme. The work investigated the effect of these two factors on the performance of diploidy genetic algorithms in dynamic environments |
| 14 | S. Yang et al. | The work proposed an adaptive dominance mechanism for DGAs in dynamic environments |
| 13 | A.M.L. Liekens et al. | The work formally described DGAs and the mechanism for transformation of DGA to SGA and vice versa |
| 12 | R. Schafer | The work used DGAs to stabilize dynamic environments and demonstrates the superior performance of DGAS over SGAs in dynamic environments |
| 11 | S. Lee and H. Rowlands | The work used DGAs to select a more robust, less optimal solution over a less robust, more optimal one. DGAs find the correct solution almost 100 % of the time |
| 6 | J.D. Bagley | The work proposed a novel algorithm that could work in changing environments |
| 7 | R.B. Hollstein | The work proposed division of gene into two types: modified and functional. Functional gene was assigned values of 0 and 1, while modified gene was assigned values of m and M. Used this model in computer control systems |
| 8 | Y. Yukiko and A. Nobue | The work modelled GAs on the lines of meiosis and applies it on dynamic travelling salesman problem. It stressed on the importance of high mutation rate |
| 5 | K. Weicker and N. Weicker | The work explored the effects of redundancy on mutation, recombination and most importantly population diversity |
| 47–49 | Rosenberg | The work proposed use of biochemical properties to determine dominance |

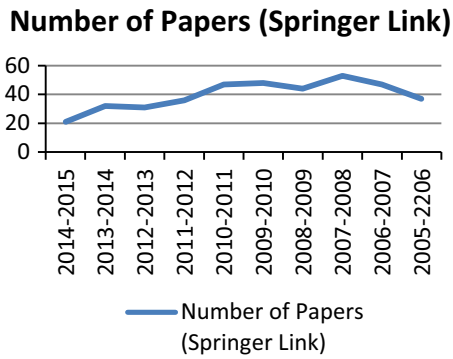


**Fig. 7** Variation of the number of papers and the year (Springer Link)

techniques. The three major techniques, discussed in the previous section, have been implemented and applied to dynamic travelling salesman problem (Bhasin et al. 2015). The results affirm the applicability of DGAs in dynamic environments.

*RQ3*

Most of the papers that were reviewed focussed on devising techniques to determine dominance (Ryan 1994; Ng and Wong 1995; Yang 2006). Some of the papers suggested that DGAs could be applied in dynamic environments. In our future work, it is desired to find a better way to determine dominance and apply DGAs in software engineering in general, and testing in particular. The chart in Fig. 8 shows the applicability of the various papers reviewed.

*RQ4*

SGAs aim for optimization, even if it comes at the cost of robustness. This attribute makes them unsuitable for a wide variety of applications. This fact has also been strongly asserted in some of papers, which were reviewed (Liekens et al. 2003). DGAs on the other hand achieve a very high degree of robustness, which may be required by such applications. They significantly outperform SGAs in dynamic environments because of their superior ability to 'remember' previous results (Lee and Rowlands 2005).

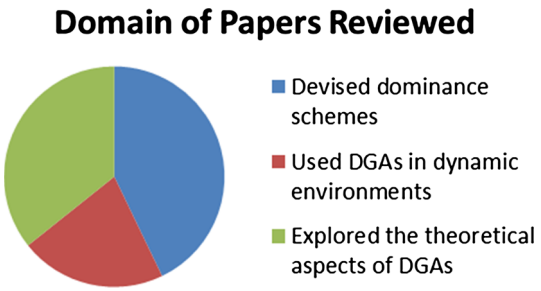


**Fig. 8** Domain of reviewed papers

## 6 Future scope

DGAs have been used in diverse fields and applications. However, the focus of the review was to find the applicability of DGAs in sub-disciplines of computer science. The applications of the techniques in testing, path finding and NP problems have been discussed as follows. However, the application in say chemistry and petroleum is beyond the scope of this review.

### 6.1 Testing

Testing is an indispensible part of software engineering (Bertolino 2008; Bartolini et al. 2011). Testing can be classified as either white box or black box. In white box, testing is done via the code, whereas in black box, testing is done via the program's inputs and outputs (Bertolino 2008). Heuristic search techniques are used, as in white box testing when there are too many paths, while in black box testing they are used when there are too many controls. In both these cases, use of DGAs could lead to significantly better results.

It may be noted that though DGAs have the potential to significantly improve white box and black box testing, we feel that their best application is in the field of regression testing as stated below:

#### *6.1.1 Regression testing*

When changes are made in the software, then the old test cases need to rerun after every change. This is called regression testing. As an increasing number of changes are made, the time taken for development is less while testing time progressively increases as pointed out by Yoo and Harman (2012) and Agrawal et al. (1993). DGAs can provide an answer to this conundrum.

### 6.2 Test case generation

Manual test case generation is an error prone and tedious process. Hence, there is a need to make this process automated. But the solution should cover the complete range of test cases even if it has to compromise on time. This is why SGAs are not suitable for this task because their main concern is optimization, not robustness. Certain novel techniques have been applied to this problem (Bhasin and Singla 2013b). The technique used in the above work used genetic algorithms to produce test cases. The population of genetic algorithms has chromosomes. A cell in a chromosome depicts the inclusion of non-inclusion of a node of the corresponding flow graph of the program under test. Each chromosome depicts some path. Each path was assigned some fitness value depending on the coverage. The selection of the path was followed by the assignment of test data to the

variables used in the path. However, we feel that because DGA prioritizes robustness over optimization, they are the best candidate for automated test case generation.

### 6.3 Path finding in automated vehicles

In real-life applications such as automated vehicles, robustness and consistency are much more important than optimization. These applications, at their core, rely on Dijkstra's algorithm (1959), which focuses on finding the shortest path. In real life though, this may not be our top priority. For example, it is more prudent to travel by a route that is safer, but longer. DGAs are much better suited for applications like these because they have mechanisms to remember and reuse the best solutions. In one of our works, diploid genetic algorithms have been applied to dynamic travelling salesman problem (Bhasin et al. 2014). The results have been compared to the greedy approach and the simple genetic algorithms. The works also present a hybrid approach, namely greedy genetic approach. The results of the experiments established that robustness can be achieved by diploidity. In the experiments carried out, the three variants of dominance were implemented and 115 trials bought forth the point that though haploid and greedy approaches do not outperform the other, diploid is the best bet for dynamic environments.

### 6.4 NP problems

The problems can be classified as P and NP. P problems are those problems which can be solved by a deterministic machine in polynomial time. NP problems can be solved by non-deterministic machines. The decision problems that are NP are referred to as NP complete. The NP hard problems are the problems, which cannot be solved in polynomial time, and neither there exist algorithms that can verify the answer in polynomial time. Thus, finding the solutions of such problems becomes infeasible with current computational resources. SGAs have been applied in this area with some success (Bhasin and Singla 2012a, b). However, as mentioned before, SGAs fail to provide robustness. DGAs will prove to be much more useful in this regard as they guarantee a very high level of consistency and robustness. In one of the related works, DGAs have been applied to dynamic travelling salesman problem (Bhasin et al. 2014).

## 7 Conclusion

While conducting the review, it was observed that simple genetic algorithms have been widely used in NP complete or NP hard problems, which is not the case with diploid genetic algorithms. This might be due to the availability of sufficient number of tools employing simple genetic algorithms and also due to the surplus papers that explain the theoretical basis for simple genetic algorithms.

It has also been observed that the applicability of DGAs in software engineering remains largely unexplored. This review will help the practitioners in understanding the basic principles of diploid genetic algorithms. The paper also puts forward all the major techniques that have been used to determine dominance, the most precarious issue in DGAs.

DGAs have been applied to travelling salesman problem (Bhasin et al. 2014), and the results prove that DGAs are more competent in providing robustness as compared to SGAs. In one of our ongoing works, it is intended to propose and apply a new dominance scheme on the dynamic knapsack problem. The framework involving DGAs in test data generation is also being developed. In one of our ongoing experiments, DGAs are being employed in dynamic knapsack problem. DGAs in the near future would pave way of robustness in the fields such as testing and NP hard problem-solving. This work intends to provide an overview to understand the fascinating concept of DGAs.

## References

Agrawal H, Horgan JR, Krauser EW, London S (1993) Incremental regression testing. In: Proc. ICSM, pp 348–357

Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walters P (2002) Molecular biology of the cell, 4th edn. Garland Science, New York. ISBN 0-8153-3218-1, OCLC 145080076 48122761 57023651 69932405

Bagley JD (1967) The behaviour of adaptive systems which employ genetic and correlation algorithms. PhD thesis, University of Michigan, Ann Arbor, MI (University Microfilms No. 68-7556)

Bartolini C, Bertolino A, Sebastian G, Elbaum S, Marchett E (2011) Bringing white-box testing to service oriented architectures through a service oriented approach. J Syst Softw 84(4):655–668

Berg J, Tymoczko JL, Stryer L (2006) Biochemistry, 6th edn. W. H. Freeman, San Francisco. ISBN 0-7167-8724-5

Bertolino A (2008) Software testing forever: old and new processes and techniques for Validating Today's Applications, Keynote at 9th international conference product-focused software process improvement (PROFES 2008), Monte PorzioCatone, June 2008, LNCS 5089, p 1

Bhasin H, Singla N (2012a) Genetic based algorithm for N-puzzle problem. Int J Comput Appl 51(22):44–50

Bhasin H, Singla N (2012b) Harnessing cellular automata and genetic algorithms to solve travelling salesman problem. In: Proceedings of international conference on information computing and telecommunication, pp 72–77

Bhasin H, Singla N (2013a) Cellular Automata based test data generation. ACM Sigsoft Softw Eng Notes 38(4):1–7

Bhasin H, Singla N (2013b) Cellular genetic test data generation. ACM Sigsoft Softw Eng Notes 38(5):1–7

Bhasin H, Behal G, Aggarwal N, Saini RK, Choudhary S (2014) On the applicability of diploid genetic algorithms in dynamic

environments. In: Soft computing and machine intelligence (ISCMI), 2014 international conference on, pp 94, 97. doi:10.1109/ISCMI.2014.27
Bhasin H, Behal G, Aggarwal N, Saini RK, Choudhary S (2015) On the applicability of diploid genetic algorithms in dynamic environments. Soft Comput. doi:10.1007/s00500-015-1803-5
Branke J (2001) Evolutionary optimization in dynamic environments. Kluwer, Norwell. ISBN:0792376315
Campbell N (1996) Biology, fourth edition. The Benjamin/Cummings Publishing Company, p 309, 310. ISBN 0-8053-1940-9
Crick F (1970) Central dogma of molecular biology. Nature 227(5258):561–563
Dijkstra EW (1959) A note on two problems in connection with graphs. Numer Math 1:269–271. doi:10.1007/BF01386390
Freeman WH, Griffiths A (2008) Introduction to genetic analysis, 9th edn. W.H. Freeman and Company, New York, pp 335–339. ISBN 978-0-7167-6887-6 and Company pp 826 ISBN 0-7167-4684-0
Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison and Wesley, Reading
Goldberg DE, Richardson J (1987a) Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the second international conference in genetic algorithms, pp 41–49
Goldberg DE, Smith R (1987b) Nonstationary function optimization using genetic algorithms with dominance and diploidy. In: Proceedings of the second international conference of genetic algorithms, pp 59–68
Hausner W, Michael T (2001) Events during initiation of archaeal transcription: open complex formation and DNA–protein interactions. J Bacteriol 183(10):3025–3031
Higgs PG (2000) RNA secondary structure: physical and computational aspects. Q Rev Biophys 33(3):199–253
Hillis D (1992) Coevolving parasites improve simulated evolution as an optimization procedure. Artificial Life II
Hollstein RB (1971) Artificial genetic adaptation in computer control systems Ph.D. Thesis University of Michigan
Kitchenham BA (2012) Systematic review in software engineering: where we are and where we should be going. In: Proceedings of the 2nd international workshop on evidential assessment of software technologies, pp 1–2
Lee S, Rowlands H (2005) Finding robust optima with a diploid genetic algorithm. Int J Simul 6(9):73–80
Lewis J, Hart E, Ritchiew G (2004) A comparison of dominance mechanisms and simple mutation on non-stationary problems. In: The PPSN V proceedings of the 5th international conference on parallel problem solving from nature, pp 139–148
Liekens AML, Eikelder HMMT, Hilbers PAJ (2003) Modelling and simulating diploid simple genetic algorithms foundations of genetic algorithms VII (FOGA VII) Malaga, Spain
Mitchell M (1996) An introduction to genetic algorithms. MIT Press, Cambridge
Ng KP, Wong KC (1995) A new diploid scheme and dominance change mechanism for non-stationary function optimisation. In: Proceedings of the 6th international conference on genetic algorithms, pp 159–166
Nissen P, Hansen J, Ban N, Moore PB, Steitz TA (2000) The structural basis of ribosome activity in peptide bond synthesis. Science 289(5481):920–930
Rosenberg RS (1967) Simulation of genetic populations with biochemical properties. Doctoral Dissertation, University of Michigan Dissertation Abstracts International, volume 28, issue no. 7, p 2732B
Rosenberg RS (1970a) Simulation of genetic populations with biochemical properties: I. The model. Math Biosci 7:223–257
Rosenberg RS (1970b) Simulation of genetic populations with biochemical properties: II. Selection of crossover probabilities. Math Biosci 8:1–37
Russell P (2001) Genetics. Benjamin Cummings, New York. ISBN 0-8053-4553-1
Ryan C (1994) The degree of oneness. In: Proceedings of the 1994 ECAI workshop on genetic algorithms
Saenger W (1984) Principles of nucleic acid structure. Springer, New York. ISBN 0-387-90762-9
Schafer R (2009) Using a diploid genetic algorithm to create and maintain a complex system in dynamic equilibrium, Stanford University
Solomon EP, Berg LR, Martin DW (2004) Biology, 8th edition, international student edition. Thomson Brooks/Cole. ISBN 978-0-495-30978-8
Tinoco I, Bustamante C (1999) How RNA folds. J Mol Biol 293(2):271–281
Watson JD, Crick FHC (1953) A structure for deoxyribose nucleic acid. Nature 171(4356):737–738
Weicker K, Weicker N (2001) Burden and benefits of redundancy. Found Genet Algorithms 6:313–333
Yang S (2003) Non-stationary problem optimization using the primal-dual genetic algorithm. In: Proceedings of the 2003 congress on evolutionary computation, volume 3, pp 2246–2253
Yang S (2006) On the design of diploid genetic algorithms for problem optimization in dynamic environments evolutionary computation, IEEE congress
Yang S, Yao X (2005) Experimental study on population-based incremental learning algorithms for dynamic optimization problems. Soft Comput 9(11):815–834
Yang S, Zeng SLY, Zhang Q, Kang L (2007) Learning the dominance in diploid genetic algorithms for changing optimization problems. In: Proceedings of the 2nd International Symposium on Intelligence Computation and Applications, pp 157–162
Yoo S, Harman M (2012) Regression testing minimization, selection and prioritization: a survey. Softw Test Verif Reliab 22(2):67–120. doi:10.1002/stv.430
Yukiko Y, Nobue A (1984) A diploid genetic algorithm for preserving population diversity—Pseudo Meiosis GA. In: Parallel problem solving from nature lecture notes in computer science, volume 886, pp 36–45