Applying Hybrid Simulated Annealing Algorithm to the Information Sets Search Problem

ANDERSON ANDREI DE BONA,WALTER GODOY JÚNIOR+
Department of Electronic and Telecommunication Engineering
Federal University of Technology of Paraná
Av. Sete de Setembro, 3165, Rebouças, 80230-901 - Curitiba - Pr
BRAZIL
debone@gmail.com and godoy3009@gmail.com

Abstract: - In error-correcting codes, due to the complexity and the processing time of maximum likelihood decoding (MLD) techniques are used which has lower complexity and processing time, one of the techniques used is the decoding algorithm for information set (IS). In this paper an exploratory analysis is performed, about the use of the technique by using IS hybrid heuristic of Simulated Annealing with Genetic Algorithm, with in order to perform comparative search on how many IS performance is capable of covering the modified solution compared to the IS in its original form and genetic approach using IS solution because the use of heuristics is that if there is no defined search patterns IS and the use of heuristic is aimed at improving the outcome of IS in its original form, which has close to the maximum likelihood algorithm results, but with lower complexity and processing time, aiming at greater performance.

Key-Words: - Error correcting code, Soft Decision Algorithm, Information Set Decoding Algorithm, Simulated Annealing, Genetic Algorithm.

1 Introduction

Error correcting code represents a technique introduced in digital systems to increase reliability in transmission operations and data storage. During these operations, errors may occur, due mostly to noise or interference in the communication channel or imperfections on the storage media. How these codes are able to detect and correct errors (according to its detection capacity / correction), they are widely used in satellite communication systems, digital telephony, local area networks of computers, laser discs, control and automation. The correction capability of a code due to a coding process in which they are added to the original message bits, bits of redundancy. According to Shannon, the maximum channel capacity (bits/ second) are respected, errors occurring in data transmission can be reduced to any desired rate through appropriate coding and decoding [1] [2]. The maximum likelihood decoding (MLD) is considered the ideal technique and most powerful decoding techniques [3] [4], but the decoding process is related to the size of the code because all code words need to be analyzed, as well as larger code size, it becomes increasingly complicated decoding, as the computational requirement because usually the time for decoding becomes infeasible. Due to this limitation of MLD, there is constant research of technical soft decoding (soft-decision decoding) that enables the demodulation and decoding interact using information from reliable demodulation in the decoding process, aiming to reduce the decode time of MLD [5].

One of the techniques used is that of Information Set Decoding is a direct decoding technique, in which a collection of information sets is used to generate code words candidates. The process then selects a decoded word that is closest to the received sequence [6]. But there is no known constructive procedure to find optimal Information Sets (IS) collection, beyond the call brute force. Can also be used a search with some techniques to build the IS during the decoding process (such, using information available reliability), or use a "pre-determined" set.

A new approach in [7] has been proposed for obtaining coverage of a set of sequentially generating a set of (n-k) such that each new standard pattern adds a minimum number of error patterns not previously covered. However, this is time consuming computational algorithm for determining solutions. In [8], was also proposed the use of the IS algorithm using a genetic algorithm-based approach, but as shown by [9], there is hybrid genetic algorithm solutions that are interesting alternatives to genetic algorithms, but in [9], the algorithm was applied to the famous traveling salesman problem (TSP), and in this work we propose a new approach to Simulated Annealing with Genetic algorithm in search of information sets.

2 Information Sets

Information Sets decoding was first suggested by Prange [6] for decoding cyclic codes and has been extensively studied and modified. Their functioning is based on a code (n, k), one Information Set (SI) is defined as any set of k linearly independent vectors (LI) of the matrix [3]. The remaining positions are the n-k parity set. So if the generator matrix for the code can be written in canonical form, the first k columns (starting from left to right) are a Information Set. Any other set of vectors form an Information Set if possible, through elementary operations with the lines, make the corresponding column of the generator matrix has unit weight. For example, consider the Hamming code C(7,4) having the following generator matrix:

$$G_0=\begin{pmatrix}1&0&0&0&1&1&0\\0&1&0&0&0&1&1\\0&0&1&0&1&1&1\\0&0&0&1&1&0&1\end{pmatrix} \qquad (1)$$

The first k positions G are a Information Set that can be represented by $I0 = \{1, 2, 3, 4\}$. Another generator to the same code matrix is obtained by adding the first row to the third and fourth, where the resulting matrix is:

$$G_1=\begin{pmatrix}1&0&0&0&1&1&0\\0&1&0&0&0&1&1\\1&0&1&0&0&0&1\\1&0&0&1&0&1&1\end{pmatrix} \qquad (2)$$

The result of these elementary operations meant that the columns 1 and 5 were exchanged in place and modified 6. Now the columns 2, 3, 4 and 5 form the set of information ($I1 = \{2, 3, 4, 5\}$). Similarly, a third generator matrix ($G2$) can be obtained for the same code C(7,4), where $I2 = \{1, 2, 6, 7\}$. For this exchange column positions is possible, it is necessary that the two columns have the value 1 in the same row.

$$G_2=\begin{pmatrix}1&0&1&1&1&0&0\\0&1&1&0&1&0&0\\0&1&1&0&1&0&0\\0&0&0&1&1&0&1\end{pmatrix} \qquad (3)$$

As the symbols contained in the information set can be specified independently, they only define a codeword. If no errors exist in these positions, the remaining symbols in the transmitted codeword can be reconstructed. This property provides the basis for all decoding algorithms that use Information in Set.

3 Greedy Algorithm for Search IS

In [7], it was displayed a greedy algorithm (IS-GU) to find a collection of information sets that covers all error patterns of a particular type. The generation of optimized collection is made taking into consideration two factors:
1) An exhaustive search is performed in order to obtain the greatest possible distance between the Information Sets chosen, using the metric of the Hamming distance.
2) The collection of information sets should cover all the error patterns of weight ($dHmin$ - 1) with the least possible overlap.

4 Simulated Annealing Algorithm

In condensed matter physics, annealing is a thermal process used for obtaining low energy states in a solid. This process consists of two stages: first, the temperature of the solid is increased to a maximum value at which it melts, and the second, the temperature is slowly reduced until the material to solidify. In the second phase, the cooling must be done very slowly, allowing the atoms making up the material, enough to organize themselves into a uniform structure with minimum energy time. If the solid is cooled abruptly, its atoms will form a irregular and weak structure, with high energy as a result of internal effort spent. Annealing can be seen as a stochastic process of determining the organization of atoms in solid that presents a minimum energy. At high temperatures, the atoms move freely and with great probability can move to positions that will increase the total energy of the system.
When the temperature is lowered, the atoms gradually move toward a regular structure, and only with small probability will increase your energy. This process was simulated in computer successfully, by [10].
The algorithm used was based on Monte Carlo methods, and generated a sequence of states of a solid as follows: Given a current state i of the solid with energy Ei, a subsequent state was generated by applying a perturbation mechanism, which transforming the current state to a next state for a small distortion, for example, by displacement of a single particle. The energy in the next stage becomes Ei. If the energy difference was less than or equal to zero, the state j was accepted as the

current state. If the variation was greater than zero, the state j was accepted with a probability given by *exp ((Ei - Ej) / (KB * T))* where *t* is the current temperature of the system and *KB* is a physical constant known as the Constant Boltzmann. This rule is known as acceptance criteria and the Metropolis algorithm also takes its name. In the early 80s, [11] developed a general-purpose algorithm analogous to the Metropolis algorithm called Simulated Annealing with (SA). In this algorithm, used as a criterion for acceptance of a new solution, the function:

$$p_{ck}(accept\ j) = \left\{ \exp\left(\frac{-[g(j)-g(i)]}{\mathrm{ck}}\right) \begin{matrix} if, g(j) \le g(i) \\ if, g(j) > g(j) \end{matrix} \right\} \quad (4)$$

Where *g* is the function to be optimized (minimized), *i* and *j* are the current solution and a candidate solution, respectively, and *ck* a parameter representing the temperature T. According to this criterion, if a candidate solution *j* is better than the current solution *i*, ie *(g (j) ≤ g (i))*, it is accepted with probability 1. Otherwise, the candidate solution is accepted with a given probability. Therefore, this probability is greater than the extent of the variation is less energy defined by *(g (j) g (i))*. At the same time, as there is a decreasing temperature *ck*, the algorithm becomes more selective, becoming less often still accept solutions which exhibit large increase in the energy change, i.e. solutions that are much worse the current solution. This probability tends to zero as the temperature approaches the freezing point.
The simulated annealing algorithm can be considered as an extension of the original local search method. Local search requires only the definition of a neighborhood scheme, and a method of assessing the cost of a particular solution, always presents a final solution. It is understood by neighborhood scheme the appropriate mechanism dependent on the problem being treated, through which it obtains a new solution, which also belongs to the space of solutions of the problem, making a minor change in the current solution.
The simulated annealing algorithm can be considered as an extension of the original local search method. The local search requires only the definition of a neighborhood scheme, is another important property of the simulated annealing algorithm, as shown in (Fig. 1) through a pseudocode is its computational simplicity of implementation.

```
get the constant α and the number of repetitions NR;
    S ← S0;
    T ← LS;          // Upper limit
    TMIN ← LI;       // Lower limit
    while (T > TMIN) {
     for I de 1 to NR do
       generate a solution S' de N(S);
       ΔE = f (S') – f (S);
         if (ΔE <= 0) {
           S ← S';
         else
           gerarate  rand ∈ random [ 0, 1 ];
             if (rand < exp (-ΔE / T)) {
             S ← S';
             }
         }
      }
        T ← T * α;
  }
```

Fig. 1 Pseudocode of the *Simulated Annealing* [11].

Looking avoid premature convergence to a local minimum, the algorithm starts with a relatively high T value. This parameter is gradually decreased and, for each of its values, several attempts (NR) to achieve a better solution in the neighborhood of the current solution are performed.
In (Fig. 1), the expression ($T \leftarrow T * \alpha$) corresponds to the process of lowering the temperature, where α is the reduction factor of temperature. In our tests the 0.95 was considered using the parameterization proposed in [12].
Mathematically, the Simulated Annealing algorithm can be modeled using the theory of Markov chains. Using this model, several important results dealing with sufficient conditions for convergence, have appeared in the literature. The vast majority of these studies, however, does not take into account the number of iterations required to achieve this convergence. Since the size of the solution space and grows exponentially with the size of the problem, the running time of an algorithm of this type can reach viable levels. A very important result is given in work published by [13], which are necessary and sufficient for the asymptotic convergence of the algorithm to a set of solutions not only the global optimal conditions are provided, but the number of iterations required for this convergence can occur.

5 Genetic Algorithms

Genetic Algorithms (GA) is a search for the stochastic variation of beam first specified in the

successor states are generated by combining two others by [14]. The idea of the process involved in GA is inspired by the theory of natural selection proposed by Darwin in his book "The Origin of Species," which describes a dynamic in nature where only the fittest survive. Thus, the nomenclature used comes from biology, such as chromosomes, genes, crossover and mutation [15].
The elements are called search and individuals are represented by a structure that is called a chromosome. Although in nature an individual may have multiple chromosomes, in representation of the search strategy only one chromosome identifies an individual. Keeping the analogy, we can think of individuals uni-chromosome. These chromosomes are data structures with encoded information of the problem and represents a chain of elements, genes. The states of the gene are called alleles [15].
The process begins by randomly generating individuals k named this set of population. A relation evaluates the degree to which an individual the requirements of the problem, and is called the fitness function. A new generation of individuals is produced by genetic operations. The crossover is the process in which there is the "intersection" between individuals, selecting the pairs according to their fitness value, being an individual created by exchange of genetic material from chromosome structure parents. The most k elements of fitness are preserved and the process repeats until they begin to converge, ie, individuals begin to repeat variation in fitness or not start to be significant. Other stopping conditions can also be prescribed as a maximum number of generations [15].
Successive generations may converge to a local maximum in the spectrum of optimality. This can be avoided with the use of a genetic operation called mutation process in which can be introduced, with each new generation, random changes in genes in a small percentage of individuals, allowing the possibility of new points of convergence [15].

5.1 Chromosomal representation

The chromosome representation is the first step in the implementation of a genetic algorithm in solving a problem. It consists in determining a way to represent each possible solution S of the search space, as a sequence of symbols generated from a finite alphabet A. In the general case, both the evaluation method as the genetic alphabet depend on each problem. But since the most appropriate structure defined to represent the solution of the problem, it must be able to represent all possible solutions of the problem uniquely, and remain unchanged in the process [16].
According to [17], using some of the metaphors employed by theorists and practitioners of GA, the most used terms are:

- Population: a group of individuals (solution set of the problem);
- Chromosome: represents an individual in the population (a setting or solution);
- Gene: is a component of the chromosome (a variable of the problem);
- Allele: describes the possible states of an individual attribute (possible values of a variable of the problem);
- Locus: is the position of the attribute in the chromosome;
- Phenotype: denotes the decoded chromosome, and
- Genotype: represents the structure of the encoded chromosome;

In most genetic algorithms assume that each individual consists of a single chromosome (which does not occur in natural genetics), which is why it is common to use the terms interchangeably chromosomes and individuals. Most genetic algorithms proposed in the literature, uses a fixed-size population with constant size also chromosomes [18].

5.2 Population

It is a set of chromosomes representing candidate solutions to suffer the effects of genetic over evolutionary process operators. Populations move toward formation of a population of more appropriate solutions to the problem being solved. The initial population consists of n individuals, usually created randomly, from which other more refined will be determined. Thus, this initial population should be as diverse as possible so that the various points of the search space can be sampled. [15]

5.3 Genetic operators

They are classified as the mechanisms responsible for modifications made by individuals in a population, being directly responsible for generating new solutions. The genetic operators of a GA must be defined considering the form of representation adopted and the nature of the problem in question. According to [19], we find three classes of operators present in most implementations, which are: mutation, recombination and selection, considered as the basic operators of GA.

5.4 Mutation operators
Classified as operators that aims to simulate natural phenomenon of genetic mutation of individuals, which in its basic version consists of the operator to change the value of some randomly selected genes.
These operators lead to the exploration of new regions of the search space of the problem, generating new genetic information, and to suffer mutation, a chromosome mapping becomes a new point in the search space, possibly in an area not yet explored by transferring the search undertaken by chromosome for this area. These operators consist of an important mechanism for maintaining population diversity and coverage of the search space. When individuals in a population will become very similar, the effect of the crossover operator will gradually nullifying causing populations in successive generations become increasingly similar. Among the various mutation operators, we highlight the Swap and based Mutation operator (MUT) [20].

5.5 Operators recombination, crossovers
These operators simulate the natural process of sexual reproduction and are responsible for transferring genetic background of the parents to their offspring. Cutoff points are selected on chromosomes parents and by combining the resulting fragments are formed their descendants. Whatever the variant chosen, these operators act on the search space by performing a refinement of solutions encoded by the parent chromosomes, since recombination "preserves" the genetic information of the same good quality as well, setting a local search to from the parents [17]. Among the most sophisticated and widely used operators can mention: Order Crossover (OX), Cycle Crossover (CX), Partially-Mapped Crossover (PMX), but this proposal is being submitted only operator Order Crossover (OX), because as [15] this operator has better performance.
The OX operator acts on the parent chromosomes through two distinct stages. In (Fig. 2) is exemplified the operation of the operator. First, select two points to cut the parental chromosomes randomly and then copies the genes located between these points of the first parent chromosome to chromosome first child fully, keeping the positions and orders of the genes. In (Fig. 2), this behavior relates to (step 1). Subsequently the remaining positions are filled with this descendant genes of the second parent, the second cutoff point onwards (step 2) After this step, this procedure shall operate from its first position, setting up a cycle being terminated when all descending positions are filled. [17].

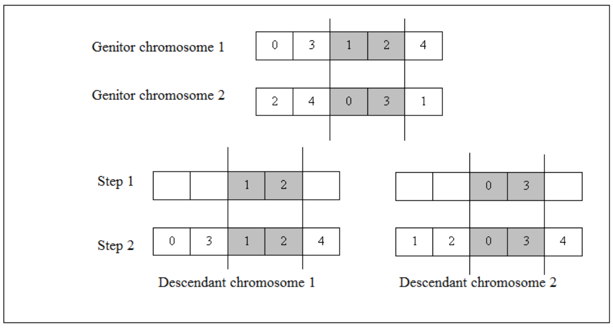

Fig. 2 Operator OX (Order Crossover). [21]

6 Proposed Model
For the development of the proposed model was adopted that a mask IS (Information set) is a binary vector of length k, used to represent an IS. So the fitness function is the number of error patterns covered by the mask in position i (and not covered by a mask in any other position $j < i$).
Our main reason for the development of Simulated Annealing, the fact that the algorithm is probabilistic, allowing for each execution of the algorithm, a different result is achieved. Thus the execution of multiple instances of the algorithm brings us the advantage that we can quickly analyze various results and choose the lowest cost, but also allows us to apply some features of Genetic algorithms seeking to improve the results. The Simulated Annealing algorithm is part of an initial high temperature, and this goes to the cooling system freezing. The aim is thus a balance to every standby performed, higher energy states can be accepted on the condition defined accepted, thus seeking to escape from local minimum of the search space worked. Thus to implement the Simulated Annealing algorithm needs to define some parameters, such as the search space, initial temperature , role acceptance , cooling program and completion criteria. Will be adopted for the proposed that the initial parameters are the same for all instances of the simulated annealing algorithm implemented model because they are optimizing the same initial problem presented. The proposed algorithm is based on the Simulated Annealing in its canonical form. Its implementation was based on the original model proposed in [16]. The basic structure of the developed method is a simulated annealing algorithm, which was presented in (Fig. 1). As the pseudocode (Fig. 3), the model implemented worked as follows: a number of runs is defined (in order to facilitate the evaluation of the outcome of the solutions obtained) after starting the execution, during execution the process will stabilize the recommended temperature, so the process will

incrementing variable and if they have not reached the last step (frozen system), is brought to a condition called locked state. If the process ended the amount of steps to be performed, so no blocks and terminates its processing incrementing the counter variable.

```
get the constant α and the number of repetitions NR;
    S ← S0;
    T ← LS;          // Upper limit
    TMIN ← LI;       // Lower limit
    while (T > TMIN) {
      for I de 1 to NR do
        generate a solution S' de N(S);
        ΔE = f (S') – f (S);
          if (ΔE <= 0) {
            S ← S';
          else
            gerarate   rand ∈ random [ 0, 1 ];
              if (rand < exp (-ΔE / T)) {
                select_results ();
                apply_crossover_ox ();
                apply_mutation ();
//Fitness function evaluates the children produced
                evaluate ();
//S receives S 'influence after the GA.
                S ← S'; // new S'
              }
          }
      }
        T ← T * α;
    }
```

Fig. 3 Pseudocode Algorithm with Simulated Annealing Genetic - IS-SAGA

After freezing is checked if the instance has reached the freezing of the algorithm. If no results are reported, apply the genetic operators crossover type OX, updates the instance as the new parameters and unlocks the same for execution, thus returning back to its locked state. With the freezing of instances, the process evaluates the results, and releases instances SA allocated, communicating the main application on the end of the optimization. Each interaction, the instance will have a set of n solutions generated by *n* instances SA allocated. Thus receives all intermediate solutions, it calculates the cost of each individual, organizes them in order of increasing cost, thus preparing a set of solutions to be treated by genetic operations.

The Elitism indicates the number of better solutions that will be no change, ie, that do not participate in the existing intersections. These solutions will be saved and together with the results of the crosses of others solutions will form the new set of basis solutions for the next iteration of SA instances. The selection of the best solutions without changing them, is due to the fact we need to keep at least one set of optimal solutions to be presented as the final solution if the crossings do not produce future success. The crossing itself will be applied to all other individuals who were not included in the elite set. Processing is to select pairs of random solutions [17] form, and perform the crossover OX. During the selection of random pairs of solutions to be crossed, the solutions will be allowed to participate only once for crossing. After crossing, we have four solutions (two parents and two children), where the instance must select only two solutions that will be the next iteration of the instances, along with the elite solutions and selected the other crossings solutions. The selection by the instance of the two solutions will be based on the best choice and best father's son.

7 Methodology

The definition of the SA parameters determining the behavior of the condition of acceptance of the solutions generated within each temperature range as well as the termination condition (freezing) in the system, since all instances obey to the same parameters for each interaction (new temperature). We have the following parameters: Parameters Accepted: a) Initial temperature: indicates whether to accept or reject solutions. High temperatures are more likely to accept solutions, while low temperatures will not b) Alpha - α: indicates the degree of convergence of the system, a high value will reduce the temperature more slowly, and a low value reduces the temperature shorter.

We got to alpha: $\alpha \subset (0.1)$. 2. Parameters of the End) Steps: Indicates the amount of interactions to the freezing of the model, b) Attempts: Maximum number of rejected at certain temperature solutions c) Changes: Maximum number of accepted solutions in certain temperature. The default values as parameters to be used Simulated Annealing proposed by [22]. Have the parameters applied to the operators of genetic algorithm, determine how they will be treated intermediate solutions generated by different instances of SA executed when the balance of the system occur at each temperature.

Intermediate results generated will be sorted according to their cost, and by choosing "elitism", the "n" first solutions lower cost, they will not suffer genetic alterations will be selected and will be fully passed the "n" instances SA to continue processing. For other solutions, pairs of solutions will be randomly selected and performed the genetic operation including crossover type OX. The new generated individuals will be evaluated along with their parents, adopting one of the following ways: Choose better father and a better son, using the crossover OX - Order Crossover, based on the implementation of [10].

Table 1. Comparative Results, IS-GA, IS-GU and IS-SAGA applied in problem C(15,7), C(23,12) and C(24,12)

	C(15,7)	C(23,12)	C(24,12)
Numbers of Information Sets (IS) [n]	43	676	1500
Patterns of error correcting code	1941	145499	1391040
Processing Time [IS -GA]	20s	1,6h	19h
Processing Time [IS -GU]	0,040s	2,1h	36h
Processing Time [IS-SAGA]	18s	1,4h	16h

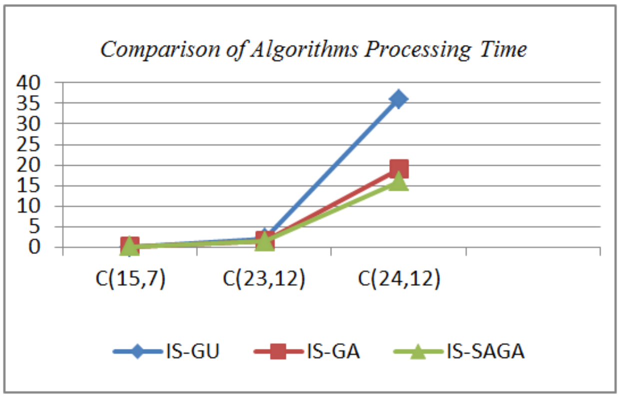

Fig. 4 Comparison Performance IS-GU, IS-GA and IS-SAGA

The applied metrics based on the complexity of the problem, so it can be noted that the IS-SAGA algorithm achieved superior performance to the others, since due to the fitness function (the accepted solution) converge faster than the population of IS-GA, and not perform the exhaustive search performed by IS-GU, besides the use of type X crossover contributed to the results achieved as shown in Table 1. And Fig. 4 , we also highlight that a substantial improvement in the quality of the cross used compared to IS-GA , since as studied in [15] this crossover operator that has superior performance . It can be seen that despite the good results obtained by IS-GA proposed in [4], it used only one type of crossover cutoff point , and perform all generations to 100% crossover. For the model proposed in this work the crossover two cutoff points, which according to [15], is extremely higher than a cutoff point, was performed and we also emphasize that besides the influence of performance of simulated annealing, the two crossovers points higher than the performances have a cutoff point, and consequently the hybrid algorithm has achieved significantly better results, therefore because of these differences this hybrid solution proposed in this paper.

8 Analysis of the results

To validate and compare the algorithms implementation was used to generate information sets of optimized for the following codes: C(15,7) Golay code C(23,12) Golay code and the extended C(24,12), used for confront the algorithm developed in [7] and [8], the tests were performed using coding in Java, a computer AMD FX 9370 8-Core 4.7 Ghz with 16 GB of RAM, running Linux Fedora 19 operating system.

The applied metrics based on the complexity of the problem, so it can be noted that the IS-SAGA algorithm achieved superior performance to the others, seeing due to the fitness function (the accepted solution) converge faster than the population of IS-GA, and not perform the exhaustive search performed by IS-GU, also by the utilization of crossover OX of I-SAGA made it possible improvement in the quality of the crossing, seeing that according to [15] this is the operator that has better performance, as compared with IS-GA in [4], it only a has used a crossover cutoff point beyond which crossover tracks always 100%, which according to [15], this type of crossover values by up to 70% must be used , not 100%, so we also highlight that the mutation with values of 1% made it possible that the algorithm could escape from local optimal, which were perceived at the time of implementation of IS-GA also worth noting that the convergence of the simulated annealing be faster assistance in obtaining the best results.

9 Conclusion

The proposed and implemented in this paper model was based on the use of two known techniques for solving search problems of information sets, which are Genetic Algorithms (IS-GA) and (IS-GU) Greedy algorithm. The proposal basically had as primary objective to increase the capability of searching for information sets using a hybrid algorithm Simulated Annealing with Genetic (IS-SAGA) through the same approach with the crossover operator of genetic type OX. To evaluate the proposed model known problems, such as C(15,7) Golay code C(23,12) Golay code and the extended C(24,12) were used.

The results obtained in simulations and design a capability of the hybrid algorithm implemented higher than the Genetic Algorithm and implemented in [8] and Greedy algorithm [7], in certain issues and certain situations.

It is noteworthy that in other studies such as [9], [23] and [24] The hybrid SA algorithm has slightly better performance, but as explained there are many other studies that indicate that genetic algorithms have better performance, since a difficulty in the algorithm genetic is the setting of the parameters, because unlike simulated annealing, they are more diverse and difficult calibration, this may have somehow influenced the performance of the IS-GA implemented in [7]. Overall the IS-SAGA expected performance had a slightly higher compared to the others here, but it is noteworthy that a better adaptation of the parameters to each long code can be done in future executions to make the best performance that have already gotten.

The results show that the IS-SAGA algorithm was higher in the search information sets, with some interesting results. Adjustments to the performance parameters of IS-GA can be performed to try to outperform the IS-GA. It can be seen that the IS-SAGA for small codes , close to the result obtained IS-GA is less than the IS-GU, but when the code is considered to be larger, it could find a solution (number of IS) in less time than the previous two.

Thus we may say that the IS-SAGA algorithm can be an alternative to maximum likelihood algorithm (MLD) when the processing time is one of the variables considered for the decoding of long codes, and we can say that new parameters are applied to the IS-GA, one can obtain new results superior to those obtained previously with the IS-GA it should not be disregarded since there is a huge amount of parameters of genetic algorithms have not been applied in the same by [7].

The algorithm implemented for long codes, might have to go through parameter settings of the heuristics and genetic and simulated annealing, the major problem of the heuristic parameters is hit uniformly for all problems and to cover a large amount of code, or codes than those tested, only a finding the correct parameters for each class or set of codes, but would have to conduct tests with new long codes to legitimize the current parameters, aiming at uniformity calibration of the current parameters, but a major problem is that the sensitivity of the parameters directly influences the result.

So it is worth noting that the IS-SAGA, use an approach based mainly on simulated annealing, and the influence of genetic parameters used are easily calibrated, thus due to ease of understanding of the adjustments of simulated annealing compared with genetic algorithms in its original form, be more easily refined, since the complexity of the parameters is extremely lower, we see a potential of the algorithm implemented in this work compared the approaches of [8] and [7], as the adjustments are easier to be calibrated and easy to understand, since the complexity of SA is extremely lower than the GA.

The major contribution of this new approach is that it enables the discovery of long codes, compared to [7] and [8], the proposed implementation achieves better performance, less time searching and lower implementation complexity, and ease of calibration the heuristic parameters, facilitating the search for long codes closest or nearly identical performance to the maximum likelihood (MLD).

As already noted earlier, the MLD is almost impractical for long codes, it becomes computationally impractical these days, so one could make changes in the proposed implementation, aiming a search sub-optimal, since the generated assemblies cover more than 90% of error patterns, making the total time for decoding were adjusted depending on the assumptions and constraints which are present for each specific problem, or class code. It should be noted that the search for information sets of in the form of masks is represented as (*k 1s* and *nk zeros*) is equivalent to a search for non-linear binary codes whose words have, among other properties, the weight constant equal to *k (w = k)*.

Thus, the algorithm developed (or set of algorithms that we call this system), with slight modifications can be used also in the binary search codes. The converse is also true, that is, for a few special cases as explained in [3] the information sets can be extracted from the linear for long codes words. This study is not conclusive, but foments a new implementation technique for information sets using hybrid heuristics the low implementation complexity. Associated with the method studied in this work, could indicate how alternative studies new crossover operations (with new operators), differential rates of mutation, new forms of hybridization, and apply the new technique to other types of decoding problems to demonstrate the potential of the proposed approach especially in long codes aiming suboptimal implementation.

References:

[1] M.J. Castiñeira, P.G. Farrell, *Essentials of Error-Control Coding*. Chichester: J. Wiley, 2006.

[2] P. Sweeney *Error Control Coding: From Theory to Practice*. Chichester: J. Wiley, 2002.

[3] G.C. Clark Jr., J.B. Cain, *Error-Correction Coding for Digital Communications*, New York, Plenum Press, 1981.

[4] D.G. Hoffman, *Coding Theory: The Essentials. New York*, Marcel Dekker Inc., 1991.

[5] D. Chase, A Class of Algorithms for Decoding Block Codes with Channel Measurement Information, *IEEE Transactions on Information Theory*, IT-18, 170-182, 1972.

[6] E. Prange, The use of information sets in decoding cyclic codes, *IRE Transactions*, Vol. IT-8, S5-S9, 1962.

[7] W. Godoy Jr, E.C.G. Wille, R.P. Jasinski. A Simple Algorithm for Decoding of Binary Block Codes based on Information Sets, *International Communications Conference ICC2010*, Cape Town, 2010.

[8] J.R. Hoffman, M.M Tenório, E.C.G Wille, W. Godoy Jr, Applying genetic algorithms to the information sets search problem, *SBRT, Communications LATINCOM, IEEE Latin-American Conference*, 2011.

[9] A.A. De Bona, *Algoritmos de otimização combinatorial: uma proposta híbrida utilizando os algoritmos simulated annealing e genético em ambiente multiprocessado, M.S.* Thesis, Federal University of Santa Catarina, UFSC, 2005.

[10] W. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, Equation of State Calculations by Fast Computing Machines, *Journal of Chemical Physics*, vol. 21, 1953.

[11] S. Kirkpatrick, D.C. Gellat, M.P. Vecchi, Optimization by Simulated Annealing. *Science*, pp. 671-680., 1983.

[12] H.A. Araújo, *Algoritmo Simulated Annealing: Uma nova abordagem*, Dissertação de Mestrado, Florianópolis, PPGEP/UFSC, 2001.

[13] B. Hajek, Cooling Schedules for Optimal Annealing, *Mathematics of Operations Research*, 13, p. 311-329. 1988.

[14] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning. Massachusets*, Addilson Wesley Publisher Company, 1989.

[15] R. Linden, *Algoritmos Genéticos – Uma importante ferramenta de Inteligência Computacional* 3º Edição, Rio de Janeiro, Editora Brasport, 2012.

[16] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Department of Computer Science, University of North Carolina, Charlotte, USA, 1999.

[17] M.C Goldberg, M.C. Goldberg, *Otimização Combinatória e Programação Linear: Modelos e Algoritmos*. Rio de Janeiro, Editora Campos, 2005.

[18] J. Tanomaru, Motivação, fundamentos e aplicações de algoritmos genéticos. *II Congresso Brasileiro de Redes Neurais, UFSC- SC*, 1995.

[19] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning. Massachusets*, Addilson Wesley Publisher Company, 1989.

[20] R. Drechsler, B. Becker, N. Gockerl, A genetic algorithm for variable ordering of obdds. *ACM/IEEE International Workshop on Logic Synthesis*, 1995.

[21] S. Russel, P. Norvig, *Artificial Intelligence A Modern Approach*, Third Edition, 2013.

[22] P.B. Hansen, Java's Insecure Parallelism, *ACM Sigplan*, v. 34, n.4, pp. 38-45,1999.

[23] L. Yamamoto, *Uso de Simulated Annealing e Algoritmo Genético no problema de reconfiguração de uma rede de distribuição de energia elétrica*, M.S. Thesis, Federal University of Parana, UFPR, 2004.

[24] E. Hopper, B. C. H. Turton, An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. *European Journal of Operational Research* 128/1, 34-57, 2000.