Multi-objective genetic algorithms in the study of the genetic code's adaptability

Lariza Laura de Oliveira[a], Alex A. Freitas[c], Renato Tinós[b]

[a] *Center of Health Informatics and Information, School of Medicine of Ribeirão Preto, University of São Paulo, Ribeirão Preto, Brazil*
[b] *Department of Computing and Mathematics, University of São Paulo, Ribeirão Preto, Brazil*
[c] *School of Computing, University of Kent, Canterbury, Kent, UK*

Abstract

Using a robustness measure based on values of the polar requirement of amino acids, Freeland and Hurst [11] showed that less than one in one million random hypothetical codes are better than the standard genetic code. In this paper, instead of comparing the standard code with randomly generated codes, we use an optimisation algorithm to find the best hypothetical codes. This approach has been used before, but considering only one objective to be optimised. The robustness measure based on the polar requirement is considered the most effective objective to be optimised by the algorithm. We propose here that the polar requirement is not the only property to be considered when computing the robustness of the genetic code. We include the hydropathy index and molecular volume in the evaluation of the amino acids using three multi-objective approaches: the weighted formula, lexicographic and Pareto approaches. To our knowledge, this is the first work proposing multi-objective optimisation approaches with a non-restrictive encoding for studying the evolution of the genetic code. Our results indicate that multi-objective approaches considering the three amino acid properties obtain better results than those obtained by single objective approaches reported in the literature. The codes obtained by the multi-objective approach are more robust and structurally more similar to the standard code.

Keywords: genetic code, genetic algorithms, lexicographic approach, multi-objective genetic algorithm

1. Introduction

The genetic code is responsible for mapping the four-letter DNA alphabet to the 20-letter protein alphabet. Almost all organisms use a unique standard code; non standard codes are very rare in nature [5, 42, 20, 27]. However, the differences between these non-standard codes are considerably small and the similarities between the codes allow us to assume that all codes have a common origin [40].

The evolutionary context of the standard genetic code's origins has been an intriguing question [39]. Many approaches have been proposed in order to investigate the adaptation of the genetic code [24]. There are three main theories that are most accepted today. The first one is the stereochemical theory, which claims that the genetic code structure was determined by the physicochemical affinity between amino acids and codons or anti-codons [14, 21, 23, 25].

The second one, adopted here, is the adaptive theory. This theory suggests that the genetic code acquired its standard form due to selective pressure to minimise the effects of errors introduced in the production of proteins [6, 42, 23]. In this theory, the genetic code evolved towards a frozen state or, in optimisation terminology, towards a local or global optimum.

The third theory, called co-evolution [44], claims that the standard code evolved under the influence of the pathways of amino acid biosynthesis, together with the first species. The three theories, which are not mutually exclusive [10], can be used to explain the robustness of the standard Genetic Code (SGC).

One of the most evident features of the SGC is its robustness against errors or mutations. The robustness has been used as evidence to support the hypothesis that the genetic code has evolved [5]. Considering this hypothesis, two approaches have been used to investigate the relationship between the robustness and the evolution of the code [33]. The first one is the statistical approach, which estimates the number of random codes better than the standard genetic code by randomly generating many different codes [11, 18]. The codes are evaluated by a robustness measure. A code A is better than a code B if the evaluation of the former, denoted $f(A)$, is better than the evaluation of the latter, $f(B)$. The other approach is the engineering approach, where the best genetic codes are obtained using an optimisation algorithm [33]. The problem with the statistical approach is that it is usually hard to find a significant number of hypothetical random codes better than the

standard one by random sampling. On the other hand, when optimisation algorithms are used, it is generally easy to find hypothetical codes more robust than the *SGC*.

The engineering approach also needs a measure to evaluate the codes. Usually, like in the statistical approach, a robustness function based on one amino acid property is employed. Using the properties: polar requirement [43], hydropathy index [26], molecular volume [15] or isoeletric point [1], Haig and Hurst [18] showed that the standard genetic code is more robust than most random codes for the first three properties, with better results for the first one. In fact, Santos and Monteagudo, using the engineering approach, also concluded that the isoeletric point is the only property that is not good to compute the robustness of the genetic codes [33], and that polar requirement is the best measure. It is important to highlight that the results presented in both papers were obtained by using the amino acid properties individually, i.e, using a single objective approach.

Santos and Monteagudo [33] employed a Genetic Algorithm (GA) to optimise the robustness function based on the amino acids' polar requirement [42, 34].

Other works used other objectives in the engineering approach. In this sense, the code is also optimised for the kinetic energy in polypeptide chains [17], compensation between codon-anticodon mismatches and tRNA misacylation [38], and secondary structure formation by mRNAs [19]. In [41], some intriguing questions about the genome structure are raised and discussed in the context of gene expression error minimisation.

The polar requirement was shown to be important to determine the organisation of the genetic code [42, 43, 11]. However, probably it was not the only factor considered during the evolutionary process. In this context, here we propose a multi-objective approach to investigate the robustness of the genetic code. We use a genetic algorithm (GA) as an optimisation algorithm to obtain hypothetical genetic codes and compare them to the standard genetic code. It is important to highlight that other optimisation algorithms could be used, but GAs, due to their intrinsic characteristics, e.g., the use of a population of candidate solutions, are natural approaches to deal with multi-objective problems [3].

In [31], a multi-objective Pareto approach was used to investigate the *SGC*'s robustness, but with a restrictive encoding. In the restrictive encoding, each amino acid is associated to a set of codons and the sets are the same found in the *SGC*. Hence, this encoding significantly reduces the

search space and uses *a priori* information about the SGC.

In a more recent paper, Santos and Monteagudo [35] included the fitness sharing technique to explore the fitness landscape of the problem, considering a robustness function based only on the amino acids' polar requirement. They concluded that the SGC is not a deep local minimum in the fitness landscape. Also, their findings show that robustness based only on the polar requirement cannot explain the SGC's structure by itself.

According to [13], when dealing with multi-objective problems, we can use three main approaches: (a) the weighted formula approach, which transforms the multi-objective problem into a single objective one; (b) the lexicographical approach, where the objectives are ranked in a priority order; and (c) the Pareto approach, which considers a set of non-dominated solutions (details will be given in Section 2). In this context, the main objective of this article is to investigate the hypothesis that a multi-objective optimisation approach is useful to study the genetic code's adaptability, since intuitively it is more biologically plausible to consider evolution as a multi-objective optimisation process than a mono-objective one. We compared the three proposed multi-objective approaches, considering their pros and cons. We also used a non-restrictive encoding with three amino acid properties which seem to be relevant to the computation of robustness. Regarding implementation, we used the well-known NSGA-II algorithm as the Pareto-based genetic algorithm, and implemented the weighted formula and lexicopraphic approaches using a standard genetic algorithm [7].

When comparing our results with previous ones [33], we found better values of fitness, which means that the best hypothetical solutions evolved by the GA are closer to the SGC in terms of the used evaluation function. In addition, the solutions found by the multi-objective approach have frequencies of codons associated with amino acids more similar to the SGC than those found by the single-objective approach. This result also indicates that it is not necessary to use a restrictive encoding to reduce the search space of the problem – a restrictive encoding is frequently used in the literature [31]. Also, it is important to highlight that the multi-objective approach seems to be more realistic, because it does not seem plausible that the robustness of the standard genetic code was optimised considering only polar requirement.

2. Methods

In this work we propose a multi-objective genetic algorithm where the candidate genetic codes are evaluated by simultaneously considering their robustness for three properties: polar requirement, hydropathy index, and molecular volume. Three multi-objective approaches are tested: an weighted evaluation function, the lexicographic approach, and the Pareto approach.

2.1. Amino acid properties

The values of polar requirement for the amino acids were first defined by Woese in 1965 [42] from chromatographic experiments. Since then, polar requirement has been used to explain the genetic code's structure [33, 18, 12, 8]. By analysing the standard genetic code organisation and the values of polar requirement for each amino acid (Table 1), one can observe that, when a codon is mutated, most often the new codon will codify the same amino acid or one with a similar polar requirement. The robustness of the code occurs for other amino acids properties too, but it is more evident for polar requirement.

Here we propose a multi objective approach where genetic codes are evaluated by simultaneously considering their robustness for three properties: polar requirement, hydropathy index, and molecular volume. The hydropathy index is based on the amino acids' free energy transfer in vapour and the side chain distribution [26], while the molecular volume is calculated as the volume of the amino acid residue minus a peptide's volume [15]. Table 1 shows the polar requirement (PR) values, the hydropathy index (HI) and the molecular volume (MV) values for each amino acid. All these properties are dimensionless.

2.1.1. Solution representation

In [33], two types of encoding for genetic codes were used. In the non-restrictive encoding, each position of the string associated with an individual (candidate solution representing the genetic code) of the GA takes one out of 20 labels, each one representing an amino acid. The stop codons are not considered, i.e., they remain fixed in the same codons associated with them in the standard code. This explains why there are 61 positions rather than 64, since the 3 positions related to the stop codons are not encoded in the individuals.

According to [36], considering all possible code combinations there will be more than $1,51 \times 10^{84}$ possible codes. In the restrictive encoding, in order

Table 1: Polar Requirement (PR), Hydropathy Index (HI) and Molecular Volume (MV) values [18].

Amino acid	PR	HI	MV
Ala (A)	7	1.8	31
Arg (R)	9.1	-4.5	124
Asp (D)	13	-3.5	54
Asn (N)	10	-3.5	56
Cys (C)	4.8	2.5	55
Glu (E)	12.5	-3.5	83
Gln (Q)	8.6	-3.5	85
Gly (G)	7.9	-0.4	3
His (H)	8.4	-3.2	96
Ile (I)	4.9	4.5	111
Leu (L)	4.9	3.8	111
Lys (K)	10.1	-3.9	119
Met (M)	5.3	1.9	105
Phe (F)	5	2.8	132
Pro (P)	6.6	-1.6	32.5
Ser (S)	7.5	-0.8	32
Thr (T)	6.6	-0.7	61
Trp (W)	5.2	-0.9	170
Tyr (Y)	5.4	-1.3	136
Val (V)	5.6	4.2	84

Figure 1: Representation of a fragment of a hypothetical genetic code (individual of the GA).

to reduce this huge search space, the same groups of codons found in the standard genetic code are considered. Each group of codons is associated with one amino acid. Hence, the restrictive encoding uses information in the standard code to generate hypothetical codes. On the other hand, as described in [30], the use of this information avoids the main problem with the non-restrictive encoding, that is: the best hypothetical codes are those where almost all of the codons are associated with a few amino acids and most amino acids are associated with only one codon.

In this work, we use the non-restrictive encoding to represent the genetic codes. We will show in the results section that the unbalanced frequency problem is solved by using the multi-objective approach. Figure 1 shows an example of a fragment of genetic code represented using the non-restrictive encoding approach.

2.2. Robustness-based evaluation fitness function

In order to evaluate the robustness of the genetic code C considering an amino acid property, we compute the mean square change for the values of this property. The mean square change, represented by $M_S(C)$ and used in both the statistical and engineering approaches, computes all possible changes in the codons for a given code C [11, 18, 33, 9, 16], i.e.:

$$M_s(C) = \frac{\sum_{ij}(X(i,C) - X(j,C))^2}{\sum_{ij} N(i,j,C)} \tag{1}$$

where $X(i,C)$ is the amino acid property value for the amino acid codified by the i-th codon of the genetic code C, and $N(i,j,C)$ is the number of possible replacements between codons i and j, i.e. the total number of nucleotide replacements necessary to get codon i from codon j.

A lower $M_s(C)$ means that code C is more robust, i.e., a change in a codon base will not cause a drastic change in the amino acid property. In robust codes, when a codon is mutated, the new amino acid is generally the same or one with a similar property value.

Table 2: Weights used in M_{st} calculation.

Weight	First base	Second base	Third base
Transitions	1	0.5	1
Transversions	0.5	0.1	1

It is also important to observe that, intuitively, the most robust codes are those where most codons are associated with the amino acids that are most important for minimizing Eq. 1, i.e., those with the shortest mean distances to all others [30]. For this reason, when using a non-restrictive encoding in a single objective algorithm, the frequencies of some amino acids increase too much, while the frequencies of some amino acids remain too small.

Although experimental data show that errors in the translational process occur in a complex manner [32], all base positions have the same impact when computing Eq. 1. Freeland and Hurst [11] propose weighting the impact of mistranslations according to the base position when computing the mean squared error.

2.2.1. Mistranslation and base position errors

Nucleotides are composed by a nitrogenous base, a pentose, and a phosphate. The nitrogenous bases are classified in purines and pyrimidines according to their structure [28]. The purines Adenine (A) and Guanine (G) have a pair of fused rings, while Cytosine (C), Thymine (T), and Uracil (U) contain a single ring [29]. Transition errors occur when a purine is replaced by another purine or a pyrimidine is replaced by a pyrimidine. On the other hand, transversion errors occur when a purine is replaced by a pyrimidine or vice versa. Freeland and Hurst [11] summarised this knowledge as follows:

- Mistranslations of the second base are much less frequent than mistranslations of the other two bases, whereas mistranslations of the first base are less frequent than mistranslations of the third base.
- Most mistranslations of the second base are transitional.
- Most mistranslations of first base are transitional.
- The transition bias is very small in the third base mistranslation.

In order to use this knowledge, Freeland and Hurst proposed a mistranslation weight matrix (Table 2). The weights in Table 2 are used in Equation 2 considering whether a mutation from codon i to j requires first, second and/or third base mutations, and whether these are transitions or transversions. The $M_s(C)$ computed with mistranslation weights, called $M_{st}(C)$ here, is given by:

$$M_{st}(C) = \frac{\sum_{ij} w(i,j)(X(i,C) - X(j,C))^2}{\sum_{ij} N(i,j,C)} \tag{2}$$

where the weight $w(i,j)$ between the amino acids codified by the i-th and j-th codons for code C is given in Table 2.

The robustness of the standard genetic code, when compared to other random codes, is better when Eq. 2 is used to evaluate the codes. In this paper, we use M_{st} to evaluate the genetic codes.

2.3. Multi-objective genetic algorithms

Due to their intrinsic characteristics, like the use of a set (a population) of solutions to be optimized in parallel, GAs have been seen as a natural approach to solve multi-objective optimisation problems [4, 3]. According to [13], three different approaches are generally used to deal with multi-objective problems, they are: the weighted formula, the lexicographic and the Pareto approaches. These three approaches are described next.

2.3.1. The weighted formula approach

This approach transforms a multi-objective problem into a single-objective one by assigning a weight to each objective. Hence, if we have n objectives the weighted formula has the form:

$$f(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + \ldots w_n f_n(\mathbf{x}) \tag{3}$$

where w_i is the $i-th$ weight and obj_i is the $i-th$ objective. The main problem with this approach is to determine the values of w_i, which are usually *ad-hoc* values.

2.3.2. The lexicographic approach

This approach gives different priorities to the objectives. When two solutions are evaluated, each objective is evaluated according to its priority. Hence, if a solution A is better than another B according to the objective

with the highest priority and this difference is significant, then it is not necessary to compare A and B according to the other objectives, solution A is considered better than B. On the other hand, if the difference between the evaluations of solutions A and B is not significant when considering the highest-priority objective, then the solutions are compared according to the next objective, and so on. Note that in the lexicographic approach it is necessary to have some *a priori* knowledge about the objectives' priorities [2], but there is no need to specify *ad-hoc* weight values. Users normally find it much easier to specify the qualitative order of objectives' priorities than the numerical values of all objective weights.

2.3.3. The Pareto approach

In this approach the multi-objective problem is not turned into a single objective problem. Usually more than one non-dominated solutions are found by the optimisation method, and the set of non-dominated solutions is called the Pareto front. In order to find the Pareto front, this type of approach uses the concept of Pareto dominance, where a solution A is considered to dominate a solution B if A is better than B in at least one of the objectives, and A is not worse than B in any of the objectives. This approach is more complex than the other ones, and its complexity increases with the number of objectives. Furthermore, it is difficult to choose the best solution among the set of final non-dominated solutions [7, 4]. There are several multi-objective optimisation algorithms [4]; in ths work, we have chosen the Nondominated Sorting Genetic Algorithm II (NSGA-II), specially because it has a good performance when number of objectives is not high [7]. NSGA-II presents a worst-case time complexity of $O(MN^2)$, where M is the population size and N is the number of objectives. NSGA-II also has a mechanism for the maintenance of solution diversity and is based on elitism. The NSGA-II algorithm can be summarized by the following steps [7]:

- An initial population $P_{(0)}$ is generated and sorted in layers (fronts) according to dominance. Hence, the first layer represents the solutions which are not dominated by other solutions. i.e., the best Pareto optimal solution set found so far.

- The population at time t, $P_{(t)}$ is submitted to selection and transformation operators, generating another population $Q_{(t)}$. Then, a population $P_{(t)} + Q_{(t)}$ is generated and sorted according to dominance.

- A new population $P_{(t+1)}$ is created by merging the layers of $P_{(t)} and Q_{(t)}$. When the number of individuals in the last layer exceeds the population size, the values of the crowding distance are used to choose the most diverse individuals. The crowding distance is also used to choose the most diverse solutions between individuals in the same layer.

The pseudo-code for NSGA-II is shown in Algorithm 1.

Algorithm 1: NSGA-II (Nondominated Sorting Genetic Algorithm II)

$P_{(0)} \leftarrow InitializePopulation()$;
// *The population* $P_{(0)}$ *is created and initialized, the size of* $P_{(0)}$ *is* N
$ObjectiveFunctions(P_{(0)})$;
// *Use objective functions to evaluate the population*
$FastNondominatedSort(P_{(0)})$;
// *The population* $P_{(0)}$ *is sorted in fronts based on Pareto dominance*
$Q_{(0)} \leftarrow SelectAndTransform(P_{(0)})$;
// *Operators of selection and transformation are applied to* $P_{(0)}$ *and*
// *another population* $Q_{(0)}$ *is generated*
$ObjectiveFunctions(Q_{(0)})$;
for *t=0* **to** NumberOfGenerations **do**
    $P_{(t+1)} \leftarrow Merge(P_{(t)} + Q_{(t)})$;
    // *The size of* $P_{(t+1)}$ *is* $2N$
    $ObjectiveFunctions(P_{(t+1)})$;
    $FastNondominatedSort(P_{(t+1)})$;
    for *i=1* **to** NumberOfFronts **do**
        if $(Size(P_{(t+1)}) + Size(Front_i) < N)$ **then**
            $P_{(t+1)} \leftarrow P_{(t+1)} + Front_i$;
        else $CrowdingDistance(P_{(t+1)}, Front_i)$;
        $i \leftarrow NumberOfFronts$;
    end
    $Q_{(t+1)} \leftarrow SelectAndTransform(P_{(t+1)})$;
    $ObjectiveFunctions(Q_{(t+1)})$;
end

The crowding distance orders the population according to each of their objectives, as shown in Algorithm 2. In order to do this, each solution of

the population is associated with a distance value. The boundary solutions with maximum and minimum values have their distance value set to an infinite distance value. All intermediate solutions are associated with a distance value which is equal to the normalized absolute value of the difference between the function values of the two neighbouring solutions. In Algorithm 2 $I[i].m$ refers to the m-th objective of the i-th individual and f^m_{max} and f^m_{min} are the maximum and minimum values of the m-th objective. After associating each solution with a distance value, it becomes possible to compare two solutions according to their proximity to the others. When the last layer of the population is ordered, according to the values of the crowding distance, the individuals with the greatest distance, i.e., the most diverse individuals, are added into P_{t+1} until the population size becomes equal to N. The pseudocode of the crowding distance procedure can be seen in Algorithm 2.

Algorithm 2: Crowding distance algorithm

for each *solution of the front* F_j **do**
  $I[i].distance \leftarrow 0$;
end
for each *objective function* $m = 1, 2, ..., M$ **do**
  Order the solutions;
  Add the solutions to the list I_m ;
end
for each *border solutions (min and max)* **do**
  $I[1].distance \leftarrow I[l].distance \leftarrow \infty$;
end
for each *intermediate solutions* **do**
  $I[i].distance = I[i].distance + (I[i+1].m - I[i-1].m)/f^m_{max} - f^m_{min})$;
end

2.3.4. Objective functions employed in this work

As mentioned before, it is known that the robustness of the genetic code is higher when the polar requirement is used. The second highest robustness level is achieved with the hydropathy index and the third one with molecular volume [18, 33]. We use this information about different robustness levels in order to determine the priorities of the objectives in the proposed lexicographic approach. Hence, for each code to be evaluated, we compute Eq. 2

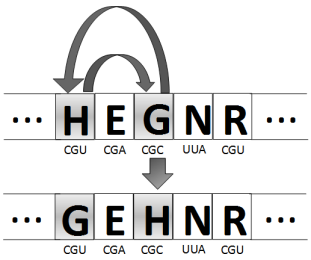

Figure 2: Example of application of the swap operator.

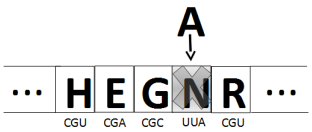

Figure 3: Example of application of the mutation operator.

three times: for polar requirement (f_1); for hydropathy index (f_2), and for molecular volume (f_3). In this way, when two codes are compared for the lexicographic approach, we first compute f_1 using the two codes. If the difference is not significant we check f_2, and finally, if the difference regarding f_2 is not significant, we check f_3. For the weighted formula approach, we define weights w_1 for f_1, w_2 for f_2, and w_3 for f_3. For the Pareto approach, the 3 objectives are considered: f_1, f_2, and f_3.

2.3.5. GA operators

The GA uses two reproduction operators: swap and mutation. These two operators were also used in [33]. The first one interchanges amino acids associated with two codons, i.e., two positions in an individual are randomly selected and their amino acids are swapped as shown in Figure 2.

In the mutation operator, a position is selected in the code (individual) and its corresponding amino acid is replaced by another one, selected among the 20 possible amino acids (Figure 3). The position and the new amino acid are randomly selected using a uniform distribution.

In order to select the individuals to be reproduced, tournament selection is employed. In this technique, a percentage of individuals is randomly selected and the best individual is chosen. In order to choose the best individual between two solutions (A and B) in the lexicographic approach, we compare the evaluations of the first, highest-priority objective (f_1, i.e., M_{st}

for polar requirement) for each solution. If the absolute value of the difference between the evaluations is higher than one standard deviation, then the solution with the lowest value of the first objective is chosen. However, when the difference is not higher than one standard deviation, the evaluations of the second objective (f_2, i.e., M_{st} for hydropathy index) for the two solutions are compared. Then, if this difference is lower than one standard deviation, the evaluations of the last objective (f_3, i.e., M_{st} for molecular volume) for the two solutions are compared. If this difference is not larger than one standard deviation, then the solution with the lowest value of the first objective is chosen, regardless of how large the difference is. The standard deviation value of each objective is calculated over the population in the current generation. Hence, at the beginning of the simulation, the standard deviations will be generally higher than at the end. Furthermore, elitism is used to preserve the best individual found in the previous generation.

2.4. Single-objective genetic algorithm

In order to provide a comparison with the previously described multi-objective GA, we also tested a single-objective GA. In this case, we used a standard GA and each individual is evaluated using M_{st} for polar requirement as the objective to be optimized. The individual representation, mutation and swap operators are the same as for the multi-objective GA.

2.5. Analysis of the solutions found by the genetic algorithms

In order to compare the standard genetic code to the best codes obtained by the GA, we use four measures:

- Percentage of Minimization Distance (*pmd*), as described in [8];

  The *pmd* for objective i is computed as follows:

$$pmd_i = 100\frac{\left|\bar{f}_i - f_i(C_{\mathrm{C}})\right|}{\left|\bar{f}_i - f_i(C)\right|} \tag{4}$$

  where $\bar{f}_i$ is the estimated average evaluation of objective i for all possible genetic codes, $f_i(C)$ is the evaluation of objective i for genetic code C (i.e., the code being evaluated), and C_{C} is the standard genetic code. The evaluation f_i is given by Eq. 2, considering one of the three properties of the amino acids (polar requirement, hydropathy, and molecular

volume). The value of $\bar{f}_i$ is computed as the mean evaluation of objective i for a large number of random codes (100 million codes were generated). Higher values of pmd_i imply greater proximity between the evaluations of objective i for codes C (standard code), relative to the estimated average evaluation for all possible codes. When computing pmd, we assume that $f_i(C) < \bar{f}_i$, i.e., the solutions found by the GA (at the end of each GA run) always have smaller fitness values than the estimated average of random codes' fitness values. We also include the absolute operator in the formula; it is necessary since the hypothetical codes generated in a multi-objective approach are not always better than the standard code. When this happens we will indicate whether or not the solution is better than the standard code in terms of fitness. In order to compare the solutions found by the multi-objective approach proposed here to the solutions found by the single-objective algorithm, in the proposed approach pmd_i is computed only for the robustness against mutations considering the polar requirement (to simplify notation, pmd will be used to denote pmd_i in the rest of the paper).

- Improvement, as described in [33];

  This measure, which is related to pmd, gives the relative improvement of the best code's fitness in relation to the standard genetic code's fitness, i.e.,

  $$imp_i = 100\frac{f_i(C_{\mathrm{c}}) - f_i(C)}{f_i(C_{\mathrm{c}})} \tag{5}$$

  Improvement should decrease as pmd increases and it provides a measure of how the best codes found by the GA improved the evaluation (fitness) of the solution compared to the standard code's evaluation. Like pmd, improvement will be computed for the robustness against mutations considering only the polar requirement (referred to as imp). The improvement can be a negative value, since the generated hypothetical codes are not always better than the SGC. Values of improvement close to zero mean greater proximity between the evaluation of the i-th objective of the hypothetical code being analysed and the SGC.

- Number of Matches;

This measure computes the number of amino acids codified by the same codon in the standard genetic code and in the evaluated code (the best code produced by a GA run).

- Entropy;

  This measure computes the entropy of a hypothetical code.

$$S(C) = -\sum_{k} p(k, C) \log p(k, C) \tag{6}$$

  where $p(k, C)$ is the relative frequency of the k-th amino acid in the genetic code C [30].

  Higher values of entropy mean that the distribution of codons related to amino acids is more uniform. The opposite situation occurs when the entropy is low and we have few amino acids related to large groups of codons. The number of codons related to amino acids is not controlled by the algorithm in the non-restrictive encoding. Hence, the size of the groups of codons can fluctuate. However, the frequency of codons related to amino acids in the standard genetic code (SGC) is almost uniform, i.e., it has a high entropy. We employ the entropy as a measure to compare the distribution of hypothetical codes with the distribution of the *SGC*. Higher values of entropy means a more uniform distribution of codons, i.e., more similar to the *SGC*'s distribution.

3. Results and discussion

The multi-objective and single-objective approaches were implemented in C++. To adjust the simulation parameters, we chose a reasonably diverse range of swap rate (50%, 70% and 90%) and two mutation rates (1% and 5%), which means on average from 0 to 3 mutations per individual.

Table 3 presents the robustness results found considering polar requirement. The first column represents the swap and mutation rates used. One can observe that the lowest values were those for 1% of mutation rate.

We performed the statistical non-parametric Wilcoxon signed-rank test, in order to compare the results for different swap and mutation rates. The p-values found can seen in Table 4.

According to Table 4, we found statistically significant differences between the experiments with the same swap rate and different mutation rates (1%

Table 3: Mean, standard deviation and minimum values of robustness values (when polar requirement is considered) calculated for the best codes found by the single-objective GA in the last generation (over 30 runs).

swap/mutation	mean ± std	min.
50%/1%	1.21±0.13	0.98
70%/1%	1.21±0.52	0.99
90%/1%	1.31±0.17	1.07
50%/5%	1.56±0.24	1.21
70%/5%	1.51±0.17	1.25
90%/5%	1.61±0.24	1.25

Table 4: P-values obtained using the Wilcoxon signed rank test, comparison of swap rates. The robustness values were calculated for the best codes found by the single-objective GA.

swap/mut.	50% 1%	70% 1%	90% 1%	50% 5%	70% 5%	90% 5%
50% 1%		0.7172	0.01788	3.725e-08		
70% 1%	0.7172		0.02906		3.725e-09	
90% 1%	0.01788	0.02906				3.278e-07
50% 5%	3.725e-08				0.7655	0.6391
70% 5%		3.725e-09		0.7655		0.09633
90% 5%			3.278e-07	0.6391	0.09633	

and 5%), at the significance level of 5%. Comparing the results for different swap rates for mutation rate 1%, we did not find statistically differences only between the results for swap rates of 50% and 70%. The robustness values (Table 3) show that the experiments with 50% and 70% of swap and 1% of mutation present the same mean value of robustness, but the experiment with 50% of swap has slightly lower minimum value of robustness. Hence, the following experiments, presented bellow, were performed considering 50% of swap rate and 1% of mutation. The population size used was 100 and the tournament size was set to 3% of the population size.

Each algorithm was executed 10 times during 1000 generations with different random seeds used to create the initial population. After publication the source code of the three approaches will be freely available in Github at: https://github.com/larizalaura/Genetic-Algorithms-employed-to-Genetic-Code-Adaptability-Study.

Table 5 presents the *pmd* and improvement values obtained by the single-

Table 5: Mean, standard deviation and best values of *pmd* (when polar requirement is considered) and improvement calculated for the best codes found by the single-objective GA in the last generation (over 30 runs).

	mean ± std	best
robustness	1.21±0.13	0.98
pmd	87.10 ± 1.07%	90.11%
imp.	55.47 ± 9.55%	39.95%
matches	4.47 (7.33%) ± 2.88	12 (19.67%)

objective GA using only M_{st} with polar requirement, considering 1% of mutation rate and 50% of swap rate. The best values of *pmd* are the highest ones, close to 100%, while the best values of improvement are those close to zero.

The results presented in Table 5 are close to the values reported in the literature. Santos and Monteagudo found a *pmd* value of 85% and an improvement of 63%.

The Pareto approach was used in two scenarios, considering 2 and 3 objectives. The lexicographic approach was used with the following objective priority order: polar requirement, hydropathy index and molecular volume. This choice was based on the robustness level of each property [18, 33]. To set up the weighted-formula approach we also used the same priority order used by lexicographic approach, doing experiments with different weight values, increasing or decreasing the weights of each property in a way that respected their priority order.

Table 6 shows M_{st} values for polar requirement (PR), hydropathy index (HI) and molecular volume (MV) obtained at the end of the simulation for all algorithms. The values of M_{st} obtained for the standard genetic code are also provided. One can observe that the code generated by the single-objective approach presents the lowest value for M_{st} with PR, which is expected since only PR was minimised in the optimisation process. The multi-objective approaches, specially the Pareto one, obtain higher values of M_{st} for HI and for MV. This happens mainly because the M_{st} for PR influences optimisation. The lexicographic, weighted-formula (all of its versions) and Pareto (optimising PR and HI) approaches obtain M_{st} values for HI better than the value for the *SGC*. Also, all optimisation approaches, except Pareto with PR and HI, obtain M_{st} values for MV lower than the one for the *SGC*.

Still observing Table 6, the lowest M_{st} obtained by a multi-objective ap-

proach was obtained by a code generated by the weighted-formula approach, with weights: $w_1 = 0.8$, $w_2 = 0.15$ e $w_3 = 0.05$, which gives the highest weight to M_{st} considering PR. The main problem with this approach is that the weights are arbitrarily chosen. Comparing the Pareto approaches, note that the best values of M_{st} were obtained when only HI was used along with PR. When M_{st} considering MV is used, higher values of M_{st} for PR were obtained, i.e. minimising that objective (MV) has some negative effect on M_{st} considering PR optimisation.

Table 7 shows the M_{st} values for the best hypothetical codes obtained by all the approaches. The M_{st} value for the *SGC* is also shown. In the single-objective, lexicographic and weighted-formula approaches, only one hypothetical code is obtained at the end of the simulation, whereas, in the Pareto approach, all the non-dominated solutions are shown. Considering the concept of Pareto dominance (Section 2.3), we classified all the solutions according to their dominance with respect to the *SGC*. The column dominance shows whether or not the code dominates the *SGC*, considering its values of M_{st}. One can observe that the hypothetical codes generated by the single-objective, lexicographic and weighted-formula approaches dominate the *SGC*. Considering Pareto solutions with two objectives, i.e, Pareto(PR and HI) and Pareto(PR and MV), one can observe that all these solutions do not dominate the *SGC*. When considering Pareto(PR, HI and MV), one can observe that 4 of 9 solutions dominate the *SGC*.

Figure 4 shows the solutions found by the Pareto approaches along with the *SGC*. Solutions dominated by the *SGC* were removed. One can observe that the solutions generated by the Pareto approach with PR and HI seem to be closer to the *SGC* than the ones generated with the three properties.

Table 8 shows the *pmd* values of M_{st} considering PR. The *pmd* values show how close the hypothetical codes' fitness values are from the *SGC*'s fitness. The mean best values found were obtained by the lexicographic, weighted-formula and Pareto approaches; all of them with *pmd* values higher than 90%. The best values of *pmd* found in [34] were around 85%, close to the values obtained for the single objective algorithm here.

Table 9 presents the values of improvement (Section 2.5). Improvement indicates how a hypothetical code improves the fitness value compared to the *SGC*. If the hypothetical codes' fitness is worse than the *SGC*'s fitness, the improvement value is negative.

Improvement and *pmd* are complementary measures. For example, the single-objective GA obtained the lowest value of *pmd* and the highest value

Table 6: Results of M_{st} for the best individuals obtained by the algorithms. The mean results are averaged over 10 runs. The M_{st} is computed considering polar requirement (PR), hydropathy index (HI) and molecular volume (MV).

		mean ± std	best
Single-objective	M_{st} (PR)	1.16± 0.14	0.92
Single-objective	M_{st} (HI)	5.35 ± 1.14	3.92
Single-objective	M_{st} (MV)	1331.63 ± 257.99	989.50
Lexicographic	M_{st} (PR)	1.86 ± 0.25	1.40
Lexicographic	M_{st} (HI)	3.71 ± 0.72	2.56
Lexicographic	M_{st} (MV)	1590.46 ± 339.79	996.13
Weighted(0.4/0.35/0.25)	M_{st} (PR)	1.80 ± 0.30	1.17
Weighted(0.4/0.35/0.25)	M_{st} (HI)	2.75 ± 0.44	1.86
Weighted(0.4/0.35/0.25)	M_{st} (MV)	760.10 ± 111.14	527.67
Weighted(0.6/0.3/0.1)	M_{st} (PR)	1.49 ± 0.20	1.14
Weighted(0.6/0.3/0.1)	M_{st} (HI)	2.20 ± 0.39	1.57
Weighted(0.6/0.3/0.1)	M_{st} (MV)	972.09 ± 163.03	743.17
Weighted(0.8/0.15/0.05)	M_{st} (PR)	1.40 ± 0.25	1.05
Weighted(0.8/0.15/0.05)	M_{st} (HI)	2.38 ± 0.34	1.79
Weighted(0.8/0.15/0.05)	M_{st} (MV)	969.97 ± 168.21	691.97
Pareto(PR and HI)	M_{st} (PR)	1.96 ± 0.61	1.57
Pareto(PR and HI)	M_{st} (HI)	3.65 ± 0.86	2.29
Pareto(PR and HI)	M_{st} (MV)	2755.20 ± 704.21	2008.49
Pareto(PR and MV)	M_{st} (PR)	6.84 ± 0.68	6.38
Pareto(PR and MV)	M_{st} (HI)	14.73 ± 1.67	13.69
Pareto(PR and MV)	M_{st} (MV)	1084.79 ± 50.47	1027.73
Pareto(PR, HI and MV)	M_{st} (PR)	5.05 ± 1.97	3.20
Pareto(PR, HI and MV)	M_{st} (HI)	7.46 ± 2.18	5.72
Pareto(PR, HI and MV)	M_{st} (MV)	1569.18 ± 353.34	1027.73
SGC	M_{st} (PR)		2.63
SGC	M_{st} (HI)		4.61
SGC	M_{st} (MV)		1766.77

Table 7: M_{st} values of non-dominated solutions found by all approaches and M_{st} values of *SGC*.

Approach	PR	HI	MV	Dominance
Single-objective	0.9	4.1	1684.7	yes
Lexicographic	1.4	2.9	1298.3	yes
Weighted(0.4/0.35/0.25)	1.3	2.0	711.0	yes
Weighted(0.6/0.3/0.1)	1.4	1.7	743.2	yes
Weighted(0.8/0.15/0.05)	1.1	2.1	692.0	yes
Pareto(PR and HI)	3.8	2.3	4209.7	no
Pareto(PR and HI)	2.4	2.6	3230.5	no
Pareto(PR and HI)	2.5	2.4	3849.2	no
Pareto(PR and HI)	1.7	3.4	2649.0	no
Pareto(PR and HI)	1.6	4.2	2320.6	no
Pareto(PR and HI)	1.6	4.2	2375.5	no
Pareto(PR and HI)	1.7	3.6	2519.3	no
Pareto(PR and HI)	1.6	4.6	2298.4	no
Pareto(PR and HI)	1.6	4.2	2333.6	no
Pareto(PR and HI)	1.6	4.7	2008.5	no
Pareto(PR and HI)	1.6	4.1	2294.9	no
Pareto(PR and HI)	1.6	4.7	2188.7	no
Pareto(PR and HI)	1.8	3.3	2545.8	no
Pareto(PR and HI)	2.1	2.7	3749.1	no
Pareto(PR and MV)	6.4	13.7	1123.6	no
Pareto(PR and MV)	6.5	13.8	1103.1	no
Pareto(PR and MV)	7.6	16.7	1027.7	no
Pareto(PR, HI and MV)	4.0	7.0	1496.0	no
Pareto(PR, HI and MV)	6.6	7.3	1342.2	no
Pareto(PR, HI and MV)	7.7	11.1	1097.2	no
Pareto(PR, HI and MV)	7.7	11.3	1149.7	no
Pareto(PR, HI and MV)	6.2	7.0	1352.5	no
Pareto(PR, HI and MV)	3.4	5.8	1906.1	yes
Pareto(PR, HI and MV)	3.2	5.8	1932.6	yes
Pareto(PR, HI and MV)	3.4	5.7	1922.4	yes
Pareto(PR, HI and MV)	3.2	6.2	1924.0	yes
SGC	2.6	4.6	1766.8	

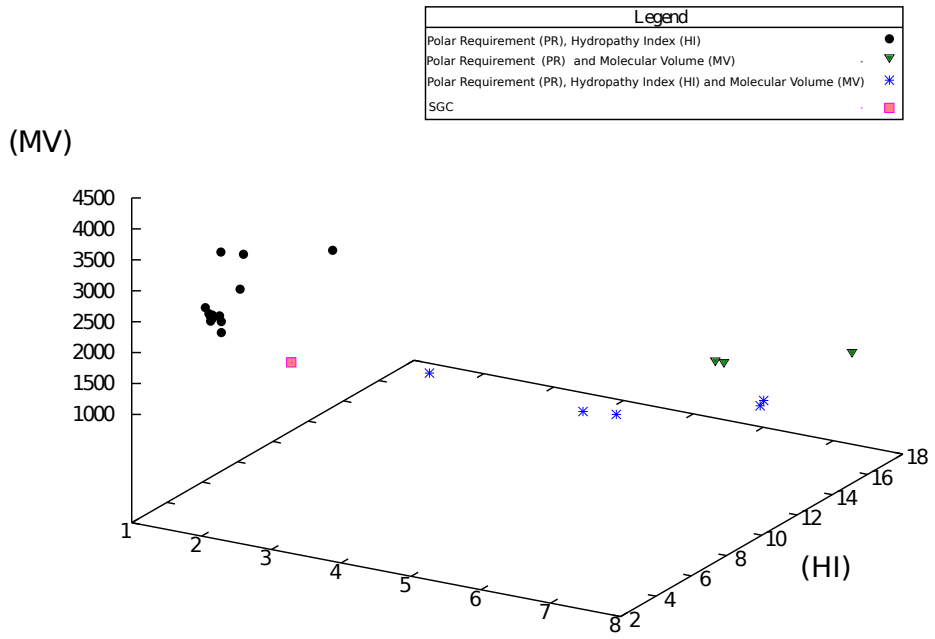

Figure 4: Pareto Frontier, considering all non-dominated solutions.

Table 8: Values of *pmd* for all tested approaches.

		mean ± std	maximum
Single-objective	*pdm*	86.75±1.10%	89.75%
Lexicographic	*pdm*	92.59±2.29%	98.04%
Weighted(0.4/0.35/0.25)	*pdm*	92.03± 2.70%	96.45%
Weighted(0.6/0.3/0.1)	*pdm*	89.33± 1.67%	93.50%
Weighted(0.8/0.15/0.05)	*pdm*	88.54± 2.11%	93.09%
Pareto(PR and HI)	*pdm*	91.92±3.22%	98.98%
Pareto(PR and MV)	*pdm*	55.32± 7.19%	60.16%
Pareto(PR, HI and MV)	*pdm*	74.29± 20.84%	93.85%

Table 9: Improvement values for all tested approaches.

	mean ± std	best
Single-objective	55.94± 5.23%	41.08%
Lexicographic	19.04± 9.44%	7.19%
Ponder(0.4/0.35/0.25)	31.49± 11.58%	13.22%
Ponder(0.6/0.3/0.1)	43.11± 7.44%	25.00%
Ponder(0.8/0.15/0.05)	46.81± 9.55%	26.72%
Pareto(PR and HI)	25.32±3.79%	23.35%
Pareto(PR and MV)	-136.23± 61.27	-69.76%
Pareto(PR, HI and MV)	-92.49± 74.98%	-22.14%

of improvement, i.e., its solution is the most distant one from the *SGC*, in terms of fitness. In this case, the lexicographic GA's solutions present the best improvement values. The best values of improvement found in [34] were about 63%.

Table 10 shows the number of matches between the best hypothetical codes of each approach and the *SGC*. It is important to note that the number of coincidences was small. One explanation is that the used encoding does not relate codons to their respective amino acids, that is, as long as the robustness function is minimised, the amino acid that binds to a given codon is not taken into account. Note that the codon-amino acid associations that emerge from the obtained solutions are different from the associations present in the *SGC*. A plausible explanation is that, obviously, during the evolution of the *SGC*, several other factors led to the emergence of these associations, factors that are not being considered for the robustness function used in this study.

Table 11 presents the entropy results. The entropy of the *SGC* is 2.87. The approaches with solutions presenting higher entropy were the Pareto and lexicographic approaches. Note that the single-objective approach presents the lowest entropy value, which is expected, since the combination of non-restrictive encoding and single-objective approach generates codes with some amino acids associated with a large number of codons. Despite the fact that the entropy is not directly used in the evaluation function, the use of a multi-objective approach solves the unbalanced frequency problem. This result shows that it is not necessary to use a restrictive encoding, as it has been done in the literature. Also, codes with higher entropy are more bi-

Table 10: Number of structural coincidences for all approaches tested.

	mean ± std	best
Single-objective	4.97 ± 3.93	15(24.59%)
Lexicographic	3.10 ± 2.51	9 (14.75%)
Weighted(0.4/0.35/0.25)	3.57 ± 2.76	9 (14.75%)
Weighted(0.6/0.3/0.1)	3.33 ± 3.92	15 (24.59%)
Weighted(0.8/0.15/0.05)	3.57 ± 2.76	13 (21.31%)
Pareto(PR and HI)	3.86±2.07	9 (14.75%)
Pareto(PR and MV)	3.33±0.58	4 (6.56%)
Pareto(PR, HI and MV)	3.71±2.36	6 (9.84%)

Table 11: Entropy values for all approaches tested.

	mean ± std	best
Single-objective	0.75 ± 0.05	0.84
Lexicographic	2.70 ± 0.11	2.90
Weighted(0.4/0.35/0.25)	2.42 ± 0.09	2.63
Weighted(0.6/0.3/0.1)	2.40 ± 0.12	2.62
Weighted(0.8/0.15/0.05)	2.41 ± 0.10	2.56
Pareto(PR and HI)	2.64 ± 0.03	2.71
Pareto(PR and MV)	2.89 ± 0.03	2.91
Pareto(PR, HI and MV)	2.89 ± 0.01	2.92
SGC	2.87	

ologically plausible. Figure 5 shows a hypothetical code, generated by the single-objective approach, in which 18 codons are associated with the amino acid Serine and 14 with Alanine, i.e. this code presents a low entropy value. In the *SGC*, the higher number of codons associated with a single amino acid is 6. A code generated by the lexicographic approach can be seen in Figure 6. This code presents higher entropy, and is more similar to the *SGC* and biologically more plausible.

4. Conclusion

In this paper, we proposed three different multi-objective approaches for the study of the genetic code's adaptability. Unlike the approaches used in

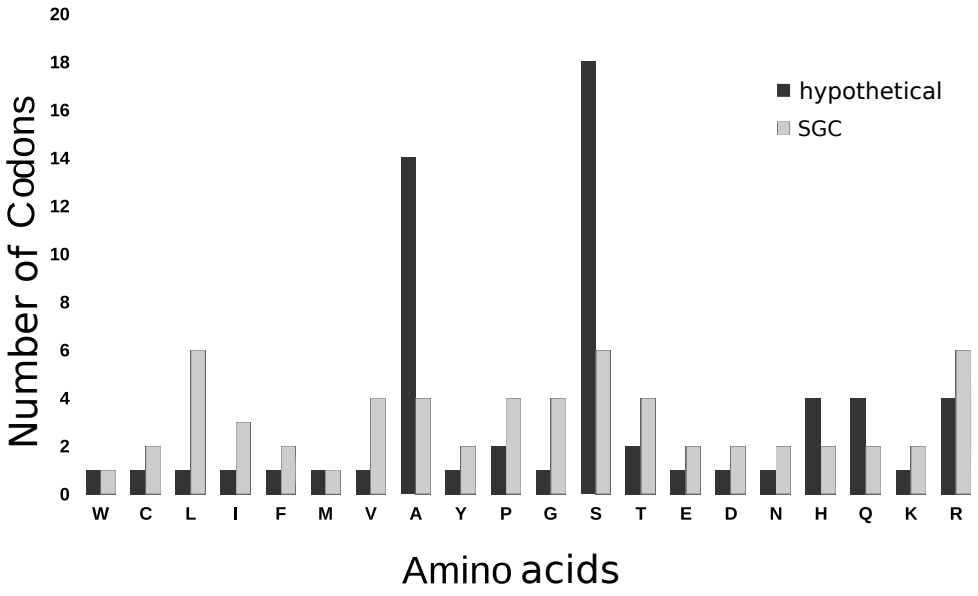

Figure 5: Frequencies of codons related to amino acids for the best hypothetical code found by the single-objective approach.

the literature where only one property is used to compute the fitness of a genetic code, in this study three amino acid properties are used to compare the genetic codes. More precisely, we use robustness measures based on polar requirement, hydropathy index and molecular volume as objectives to be optimised.

Using all multi-objective approaches, we found higher *pmd* values than those reported in the literature. We also found solutions whose frequencies of codons associated with amino acids are more similar to the frequencies in the standard code than those found by the single-objective approach. This result also indicates that it is not necessary to use a restrictive encoding to reduce the search space of the problem. Hence, using multi-objective optimisation to study the genetic code's adaptability seems to be a promising approach. Also, it seems a more realistic approach, because it does not seem plausible that the robustness of the genetic code was optimised considering only one amino acid property.

The best values of *pmd* and improvement were found using the lexicographic GA, which means that the fitness values of the codes found by the

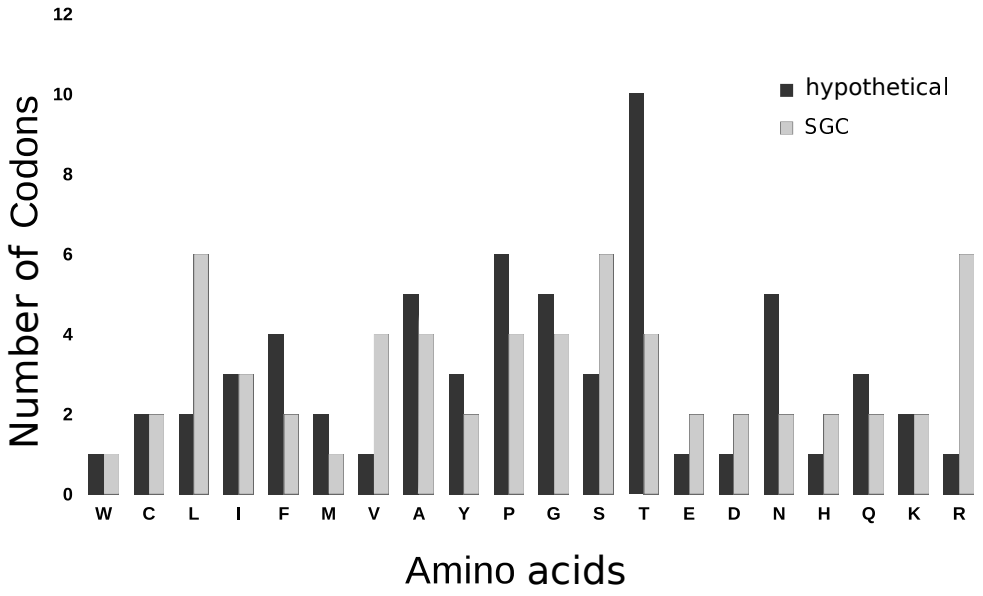

Figure 6: Frequencies of codons related to amino acids for the best hypothetical code found by lexicographic approach.

lexicographic GA are closer to the fitness of the SGC than the fitness values of the codes found by the other approaches. The best entropy values were found by the Pareto and lexicographic approaches. The results also showed that it is possible to obtain good solutions using the weighted-formula approach, choosing a good combination of weights. However, the choice of weights is arbitrary.

Furthermore, the approach based on the robustness measures adopted here, and in other papers in the literature, does not seem to be able to explain, by itself, the structure of the standard genetic code. One could suppose that the SGC is located in a local optimum in the search space, which would explain these results. Santos and Montegudo, as well as Knight and colleagues, also suggested that [34, 22] in previous studies. Their experiments resulted in hypothetical codes with structures similar to the SGC, but still different from it. Hence, future studies should address also other objectives to be considered in the study of the genetic code's adaptability. In addition, other characteristics of the problem should be addressed, like studying the role of the stop codons in the optimisation problem and the relation between

each codon and the amino acids associated with that codon. There are several studies considering the role of stop codons in the genetic code, for instance: [23]; [37]. However, there is no study combining the robustness functions used here with the stop codon information; this could be an interesting future research direction.

5. Acknowledgments

The authors would like to thank the Brazilian research-funding agencies Fapesp (process number 2012/24559-4) and CNPq for the financial support.

References

[1] Alff-Steinberger C (1969) The genetic code and error transmission. *Proceeding National Academy of Sciences of the United States of America* 64, 584–591

[2] Basgalupp MP, Barros RC, de Carvalho ACPLF, Freitas AA, Ruiz DD (2007) Legal-tree: a lexicographic multi-objective genetic algorithm for decision tree induction. *Proceedings of the 2009 ACM symposium on Applied Computing* 1, 1085–1090

[3] Coello CAC, Van Veldhuizen DA, Lamont GB (2002) *Evolutionary algorithms for solving multi-objective problems*. Springer

[4] Coello CAC, Lamont GB (2004) *Applications of multi-objective evolutionary algorithms*. World Scientific

[5] Crick FH (1968) The origin of the genetic code. *J. Mol. Biol.* 38, 367–379

[6] Crick FH (1964) Streptomycin, Suppression, and the Code. *Proceedings of the National Academy of Sciences* 51(5):883890

[7] Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: N*SGC*-II. *IEEE Transactions on Evolutionary Computation* 6(2):182–197

[8] Di Giulio M (1989) The extension reached by the minimization of the polarity distances during the evolution of the genetic code. *Journal of Molecular Evolution* 29, 288–293

[9] Di Giulio M, Capobianco MR, Medugno M (1994) On the optimisation of the physicochemical distances between amino acids in the evolution of the genetic code. *Journal of Theoretical Biology* 168, 43–51

[10] Di Giulio M (2005) The origin of the genetic code: theories and their relationships, a review. *Biosystems* 2, 175–184

[11] Freeland SJ, Hurst LD (1998) The genetic code is one in a million. *Journal of Molecular Evolution* 47, 238–248

[12] Freeland SJ (2002) The Darwinian genetic code: an adaptation for adapting?*Genetic Programming and Evolvable Machines* **2**, 113–127

[13] Freitas A (2004) A critical review of multi-objective optimisation in data mining: a position paper. *ACM SIGKDD Explorations* 6, 77–86

[14] Gamow G (1954) Possible relation between deoxyribonucleic acid and protein structures. *Nature* 173:318

[15] Grantham R (1974) Amino acid difference formula to help explain protein evolution. *Science* 185, 862–864

[16] Goldman N (1993) Further results on error minimization in the genetic code. *Journal of Molecular Evolution* 37, 662–664

[17] Guilloux A, Jestin JL (2012) The genetic code and its optimisation for kinetic energy conservation in polypeptide chains *Biosystems* 2:141–144

[18] Haig D, Hurst LD (1991) A quantitative measure of error minimization in the genetic code. *Journal of Molecular Biology* 33, 412–417

[19] Itzkovitz S, Alon U (2007) The genetic code is nearly optimal for allowing additional information within protein-coding sequences. *Cold Spring Harbor Lab Genome Research*, 4:405–412.

[20] Keeling PJ (2016) Genomics: evolution of the genetic code. *Current Biology*. Elsevier 26(18), 851–853

[21] Knight R, Landweber L, Landweber LF (1999) Tests of a stereochemical genetic code. *Landes Bioscience*, 2000–2013

[22] Knight R, Freeland SJ, and Landweber LF (2004) *Selection, history and chemistry: the three faces of the genetic code*. The Genetic Code and the Origin of Life. Springer US 24, 201-220.

[23] Koonin EV, Novozhilov AS (2009) Origin and evolution of the genetic code: the universal enigma. *Wiley IUBMB life* 2:99–111

[24] Koonin EV, Novozhilov AS (2017) Origin and evolution of the universal genetic code *Annual Review of Genetics* 51(1)

[25] Kumar B, Supreet S (2016) Analysis of the optimality of the standard genetic code. *Molecular BioSystems* 12(8):2642–2651

[26] Kyte J, Doolittle RF (1982) A simple method for displaying the hydropathic character of a protein. *J Mol Biol* 157:105–132

[27] Lajoie MJ, Söll D, Church GM (2016) *Overcoming challenges in engineering the genetic code. Journal of molecular biology* 428(5):1004–1021

[28] Lehninger AL, Nelson DL, Cox MM (2005) *Lehninger Principles of Biochemistry*. WH Freeman

[29] Lodish H, Berk A, Zipursky S, Lawrence, Kaiser, Chris A, Krieger M, Scott MP, Bretscher A, Ploegh H, Matsudaira P (2007) *Molecular Cell Biology*. W H Freeman

[30] Oliveira LL, Tinós R (2014) Entropy-based evaluation function in a multi-objective approach for the investigation of the genetic code robustness. *Springer Memetic Computing* 6:157–170

[31] Oliveira LL, Tinós R (2015) A multiobjective approach to the genetic code adaptability problem.*BMC Bioinformatics* 16(1):52

[32] Parker J (1989) Errors and alternatives in reading the universal genetic code. *Microbiology and Molecular Biology Reviews* 53, 273

[33] Santos J, Monteagudo Á (2010) Study of the genetic code adaptability by means of a genetic algorithm.*Journal of Theoretical Biology* 264, 854–865

[34] Santos J, Monteagudo Á (2011) Simulated evolution applied to study the genetic code optimality using a model of codon reassignments. *BMC Bioinformatics* 12, 56

[35] Santos J, Monteagudo Á (2017) Inclusion of the fitness sharing technique in an evolutionary algorithm to analyze the fitness landscape of the genetic code adaptability. *BMC Bioinformatics* 18(1), 195

[36] Schoenauer S, Clote P (1997) How optimal is the genetic code. In: IEEE Proceedings of the German Conference on Bioinformatics (GCB'97) 65–67

[37] Seligmann H, Pollock DD (2004) The ambush hypothesis: hidden stop codons prevent off-frame gene reading. *DNA and Cell Biology* 10:701–705

[38] Seligmann H (2010) Do anticodons of misacylated tRNAs preferentially mismatch codons coding for the misloaded amino acid? *BMC Molecular Biology*, 11:41

[39] Sengupta S, Higgs PG (2015) Pathways of genetic code evolution in ancient and modern organisms *Journal of molecular evolution*, 80(5-6), 229

[40] van der Gulik PTS, Hoff WD (2016) Anticodon modifications in the tRNA set of LUCA and the fundamental regularity in the standard genetic code *PloS one*, 11(7)

[41] Warnecke T, Hurst LD (2011) Error prevention and mitigation as forces in the evolution of genes and genomes *Nature Reviews Genetics*, 12:875–881

[42] Woese, CR (1965) On the evolution of the genetic code. *Proceedings of the National Academy of Sciences of the United States of America*. 54, 1546–1552

[43] Woese CR, Dugre DH, Saxinger WC, Dugre SA(1966) The molecular basis for the genetic code. *Proceedings of the National Academy of Sciences of the United States of America* 55(4), 966

[44] Wong JT (1975) A co-evolution theory of the genetic code. *National Academy of Sciences* 72(5), 273