Applied Soft Computing 139 (2023) 110269

Contents lists available at ScienceDirect

Applied Soft Computing

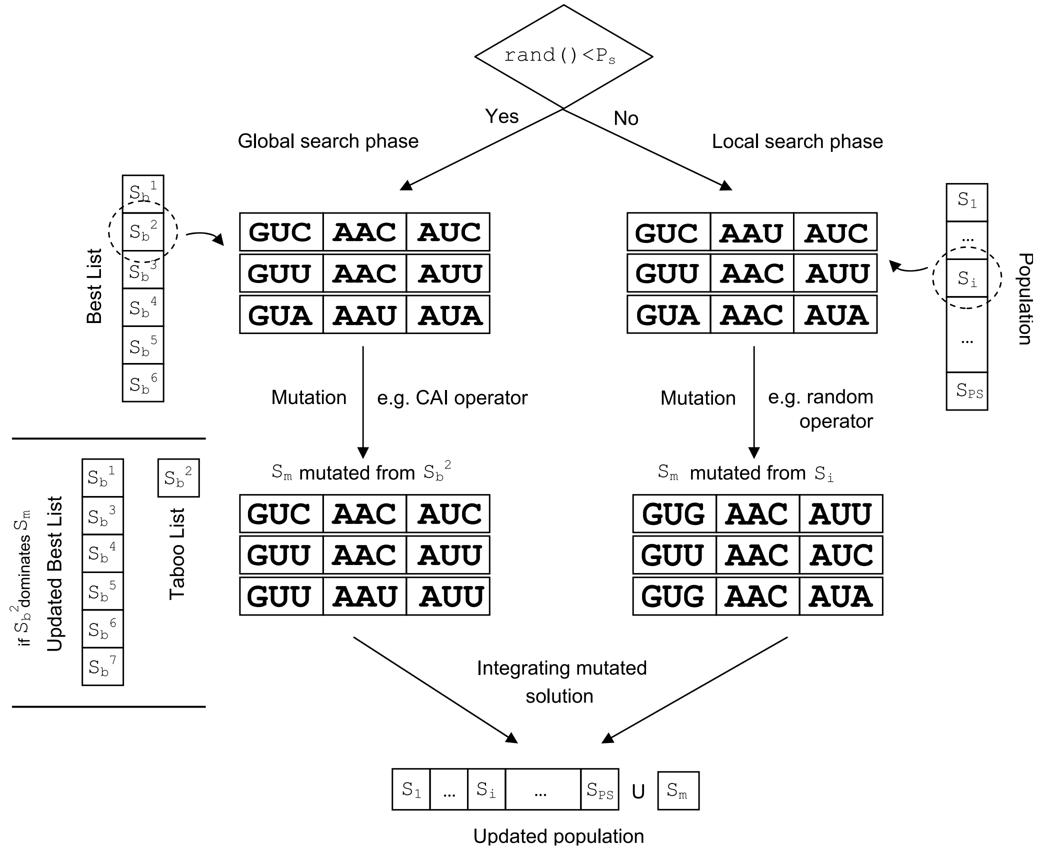
journal homepage: www.elsevier.com/locate/asoc

A multi-objective butterfly optimization algorithm for protein encoding

Belen Gonzalez-Sanchez, Miguel A. Vega-Rodríguez *, Sergio Santander-Jiménez

Escuela Politécnica, Universidad de Extremadura[1], *Campus Universitario s/n, 10003 Cáceres, Spain*

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 3 November 2022
Received in revised form 26 February 2023

ABSTRACT

The integration of multiple genes to maximize protein expression levels represents an important challenge in synthetic biology. This task relies on the definition of multiple protein-coding sequences, which must be as different as possible to avoid information loss. Proteins can be encoded in different

* Corresponding author.
E-mail addresses: belengs@unex.es (B. Gonzalez-Sanchez), mavega@unex.es (M.A. Vega-Rodríguez), sesaji@unex.es (S. Santander-Jiménez).

[1] https://ror.org/0174shg90

https://doi.org/10.1016/j.asoc.2023.110269

Accepted 23 March 2023
Available online 29 March 2023

Keywords:
Multi-objective optimization
Butterfly optimization algorithm
Protein encoding
Bioinformatics
Design of multiple genes

ways, using synonymous codons that translate into the same amino acid. Some codons are better suited to the host than others, thus being preferable the use of the most fitting ones. However, adopting only the most highly adapted codons would lead to very similar coding sequences. An additional criterion is given by the fact that the designed sequences must contain a suitable guanine–cytosine (GC) ratio in accordance with the characteristics of the host organism. Therefore, this biological task requires the simultaneous optimization of several, conflicting objectives. This work proposes a novel multi-objective approach for protein encoding, which tackles the problem according to a new formulation based on three objective functions: codon adaptation index, Hamming distance between sequences, and GC content. Our work extends the recent Butterfly Optimization Algorithm to multi-objective contexts, integrating problem-specific operators to boost solution quality by covering the different aspects required for accurate protein encoding. Two key structures, a taboo list and a best solution list, are defined to conduct improved searches attending to the potential improvements that each solution in the population can promote. Experiments conducted on nine real-world proteins reveal the attainment of relevant solutions from different evaluation perspectives, showing significant improvements over other single and multi-objective methods from the literature.

1. Introduction

One of the major research goals in synthetic biology is to maximize the expression levels of proteins, which perform a vast amount of vital functions within organisms. This task is difficult and involves multiple complex factors. An important strategy commonly found in the literature is the integration, into a host organism, of multiple genes encoding the same protein, with the aim of increasing its expression levels proportionally [1]. That is, by integrating *n* genes, the corresponding expression levels are expected to experiment an increase of *n* times. Although there are some circumstances that impact performance [2], different studies agree on the usefulness and advantages contributed by this technique. Examples of such studies include the chromosomal integration of genes based on flippase recombinase [3] and methylotrophic yeast transformants [4].

The integration of multiple genes into an organism is a complex, time-consuming, and costly procedure [3]. Traditionally, time and cost restrictions have been addressed by integrating the designed genes very close to each other within the host, as shown in the case of *Pichia pastoris* [4] and *Escherichia coli* [5]. Unfortunately, this close proximity among genes may lead to homologous recombination, a metabolic process that negatively interferes with the integration [6]. In practical terms, a homologous recombination motivates the loss of some of the genes when very similar or identical subsequences are used. For example, given five sequentially concatenated genes g_1, g_2, g_3, g_4, g_5, a homologous recombination between g_2 and g_3 will result into the loss of the information contained in these two genes, giving as a result the subsequent chain g_1, g_4, g_5. Therefore, it is essential to design genes by defining highly different protein-coding sequences (CDSs). This idea represents a first priority goal in protein encoding: each CDS must be as different as possible with regard to the others.

It is technically feasible to encode a protein through multiple, different CDSs by using synonymous codons for each specific amino acid. Most amino acids can be expressed through several synonymous codons. Consequently, it is possible to design multiple encoding sequences of a protein with differences among sequences. However, an accurate selection of synonymous codons for a target host is not straightforward, since different synonymous codons have different usage frequencies and therefore some synonymous codons have better adaptation properties than others. In order to design host-fitting CDSs and enhance expression levels, the synonymous codons with the highest adaptation index should be employed for codification purposes [7]. This point has been validated in different organisms, such as *Arachis duranensis* [8], *Saccharomyces cerevisiae* [9], *Arabidopsis thaliana* [10], bacteria [11], and viruses [12]. Nevertheless, this second priority goal is in conflict with the avoidance of homologous recombination, as the usage of the most highly adapted codons promotes higher sequence similarity.

Previous works have addressed the optimization of these goals, either as multi-objective optimization problems or as single-objective optimization problems. It can be highlighted works like [13], in which an adaptable approach named as COOL was proposed to conduct CDS optimization supporting adaptation indexes. D-Tailor [14] defines a distance-based Monte Carlo approach to perform the property-based design of synthetic DNA sequences, while OPTIMIZER [15] incorporates three different optimization methods for codon usage optimization. Regarding the usage of multi-objective metaheuristics in this problem, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) was adopted in Terai's method [16], with the aim of designing highly-adapted CDSs whose sequences are as different as possible. Following this idea, a multi-objective approach inspired by the behavior of honey bees, designated as MOABC, was proposed in [17]. Other alternative multi-objective schemes include the MOSFLA memetic algorithm [18] and the neighborhood-based trajectory method MOVNS [19].

In spite of this, these previous works do not focus on optimizing another important property in protein encoding: the guanine–cytosine (GC) content of the encoded sequences. The excess or lack of GC content (measured as the ratio of the nucleotide bases G and C in a sequence) can lead to poor protein synthesis and compromise the annealing of the synthesis [20]. Moreover, mismatches in GC content can also affect mRNA stability [21] or alter the mRNA secondary structure formation [22]. Therefore, our third priority goal will be the optimization of the GC content to encode CDSs that properly fit the GC rate of the host organism. To the best of our knowledge, this is the first time that protein encoding is addressed as a multi-objective problem targeting these three important optimization goals.

The use of bioinspired algorithms represents a useful tool to tackle complex optimization problems. An example of this kind of approaches is the Butterfly Optimization Algorithm (BOA) [23]. BOA is a recently proposed bioinspired metaheuristic, aimed at solving global optimization problems by mimicking the intelligent foraging behavior of butterflies. This metaheuristic has gained increased interest in recent years due to the relevant results reported in both benchmark and real-world problems [23]. In this sense, different works have explored the application of BOA in multi-objective contexts. In [24], BOA was applied to address constrained multi-objective optimization problems, using two main strategies: 1) an external archive to save and retrieve non-dominated solutions throughout execution, and 2) a selection strategy to identify a leader solution that guides the optimization process. An alternative design was proposed in [25], coupling BOA with non-dominated sorting to manage multi-objective optimizations. The generation of new solutions in this approach is driven by a guided search strategy, which selects three parent candidates from top 10%, bottom 10%, and middle-ranged individuals. Other butterfly-inspired strategies were studied in [26], employing the classic weighted-sum method to combine multiple objective functions into one, thus addressing the search as a single-objective optimization problem. Finally, BOA was combined with machine learning techniques (i.e. support vector machine) in [27] to conduct feature and parameter selection for the processing of cardiovascular magnetic resonance images.

This work addresses protein encoding by proposing a novel multi-objective approach based on the algorithmic scheme of BOA. More specifically, a new multi-objective adaptation of BOA, designated as Multi-Objective Butterfly Optimization Algorithm (MOBOA), is devised and implemented to accurately design CDSs with highly adapted codons while also avoiding homologous recombination and accomplishing satisfying GC ratios. In comparison to previous approaches, the search engine in MOBOA is built upon the definition of two lists that support the generation of new multi-objective solutions: a best solution list and a taboo list. The best solution list identifies the most promising non-taboo solutions, according to Pareto ranks and crowding distances, to be used to generate new solutions. If a solution in the best list did not manage to provide any further improvements, it is replaced by a different solution and migrated to the taboo list. This approach is devised to conduct improved multi-objective searches by considering the potential improvements that each solution in the population can promote. Furthermore, we define problem-oriented operators to optimize the different aspects to be considered for accurate protein encoding. The proposed MOBOA is subject to a thorough experimental evaluation on nine real proteins, measuring performance through multiple quality metrics [28], such as hypervolume, maximum spread, set coverage, and distance to the ideal point. In order to show the relevance of the proposed approach, comparisons of multi-objective and biological quality are conducted with regard to other five state-of-the-art methods.

The main contributions of this work can be summarized as follows:

- Extension of the protein encoding problem, incorporating a new optimization criterion (GC content) to be considered. A novel multi-objective formulation incorporating three conflicting objectives is described and addressed.
- Proposal of an adaptation of the recent Butterfly Optimization Algorithm to tackle multi-objective optimization problems through bioinspired search strategies, with global and local search phases. These strategies are supported by two structures (a best list and a taboo list) defined to handle the generation of new solutions.
- Application of the proposed multi-objective approach to accurately handle protein encoding optimization, defining multiple problem-specific operators that cover the different optimization goals considered in the problem formulation.
- In-depth experimental evaluation of the proposal on nine real proteins, including comparisons with other five state-of-the-art methods for protein encoding. The relevance of the obtained results is examined attending to different perspectives through four multi-objective metrics, biological comparisons, and statistical studies.

This paper is organized as follows. Section 2 describes and formulates the biological problem addressed in this work. Section 3 details the proposed multi-objective approach, illustrating the algorithmic scheme of MOBOA and the operators defined to tackle protein encoding. Section 4 explains the experimental methodology herein followed and examines the attained results, comparing multi-objective and biological quality with regard to other state-of-the-art methods. Finally, Section 5 includes concluding remarks and defines future research directions.

2. Protein encoding: Problem formulation

The protein encoding problem is aimed at designing CDSs that can be integrated into a host to generate a specific protein with improved expression levels. In this problem, a solution is given by a set of CDSs that encode in alternative ways the amino acids of the target protein using synonymous codons. These encoding sequences are represented as strings of characters belonging to the RNA alphabet $\lambda = \{A, C, G, U\}$, which denote the bases adenine, cytosine, guanine, and uracil, respectively. The length L of the CDSs depends on the number of amino acids that conform the target protein, while the expert determines the number I of CDSs desired to encode it. Fig. 1 shows an example of protein encoding with three CDSs.

The multi-objective formulation tackled in this work involves three objective functions to evaluate different important aspects of this problem. Since priority must be given to encodings that employ suitable codons with a high frequency of occurrence in the host, the first objective evaluates the codon adaptation indexes (CAI) of the solution. The second objective calculates Hamming distances (HD) to evaluate differences between CDSs, in order to avoid homologous recombination issues. The third objective examines the GC content at the third position (i.e. the third nucleotide) of the codons, with the aim of preserving the standard GC rates from the host genome. The following subsections explain these objective functions in detail.

2.1. Codon Adaptation Index (CAI)

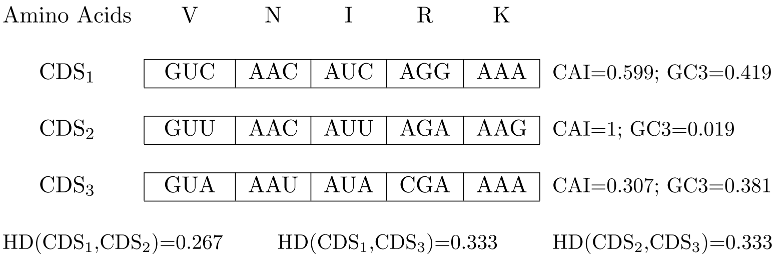
The first objective function is oriented towards evaluating the adaptation of candidate CDSs to the host. Solutions that encode the target protein using the most highly adapted codons must be preferred, that is, solutions that maximize the minimum CAI (mCAI) of their CDSs. In order to calculate mCAI, the particular CAI value for each CDS i that composes the solution must be obtained first. CAI can be calculated by using Eq. (1):

$$CAI(CDS_i) = \sqrt[N]{\prod_{n=1}^{N} W(codon_{i,n})}, \tag{1}$$

where CDS_i denotes the ith CDS in the solution, N the number of codons that compose the CDS, and $W(codon_{i,n})$ the adaptation weight associated to the nth codon of CDS_i. The adaptation weight is calculated by dividing the usage frequency of the employed synonymous codon by the highest frequency shown among all possible synonymous codons (i.e. the frequency of the most highly adapted codon). Once defined the CAI values for each CDS, the mCAI of the solution can be calculated as follows:

$$mCAI = \min_{1 \leq i \leq I} \quad CAI(CDS_i), \tag{2}$$

where I is the number of CDSs and CAI(CDS_i) the CAI value for the ith CDS. Solutions with higher mCAI denote higher adaptation, thus being preferable from this perspective. Please observe that the use of average CAI values (instead of mCAI) could potentially mask poorly adapted CDSs (with low CAI), thus not being as accurate as a more restrictive measure like mCAI.



Amino Acids	V	N	I	R	K	
CDS_1	GUC	AAC	AUC	AGG	AAA	CAI=0.599; GC3=0.419
CDS_2	GUU	AAC	AUU	AGA	AAG	CAI=1; GC3=0.019
CDS_3	GUA	AAU	AUA	CGA	AAA	CAI=0.307; GC3=0.381

HD(CDS_1,CDS_2)=0.267 HD(CDS_1,CDS_3)=0.333 HD(CDS_2,CDS_3)=0.333

Fig. 1. An example of a protein composed of 5 amino acids encoded with 3 CDSs, each one with a length of 15 nucleotides. For each CDS, the corresponding CAI and GC3 values are illustrated, also displaying the HD values for all the pairs of CDSs. The objective scores for this solution are given by the minimum CAI value ($mCAI = 0.307$), the maximum GC3 value ($MGC3 = 0.419$), and the minimum HD value ($mHD = 0.267$).

2.2. Hamming distance between CDSs (HD)

This objective function is focused on evaluating similarity at the nucleotide level between the CDSs that compose a solution. Under this perspective, priority is given to candidate solutions that maximize the minimum HD (mHD) between CDSs. For each pair (CDS_i, CDS_j) in a solution, the HD between CDS_i and CDS_j is calculated by using Eq. (3):

$$HD(CDS_i, CDS_j) = \sum_{1\leq k\leq L} \sigma(CDS_{i,k}, CDS_{j,k}), \tag{3}$$

where L is the length of the CDS in terms of number of nucleotides and $\sigma(CDS_{i,k}, CDS_{j,k})$ measures the similarity between CDS_i and CDS_j at the kth nucleotide. More specifically, σ will be 0 in case $CDS_{i,k} = CDS_{j,k}$ (both nucleotides match) and 1 otherwise (different nucleotides are observed at the kth position). After calculating the HD values for each possible pair of CDSs in the solution, mHD can be computed as follows:

$$mHD = \min_{1\leq i<j\leq I} \frac{HD(CDS_i, CDS_j)}{L}. \tag{4}$$

The optimization of solutions under this objective requires improving the pair of CDSs with the lowest HD value, that is, the pair with more nucleotides in common. Solutions with higher mHD are therefore preferred, since a higher mHD value denotes that the encoded CDSs show more nucleotide diversity. As in the mCAI case, the use of the average HD could mask hidden outliers thus not being as suitable as mHD to evaluate differences in this context.

2.3. GC content at the third nucleotide (GC3)

The third objective function allows adjusting the percentage of nucleotides G and C at the third position of the codons (GC3), in order to be coherent with the natural GC ratio of the host organism. For each CDS i in a candidate solution, a GC3 score is assigned by applying Eq. (5):

$$GC3(CDS_i) = \frac{\sum_{1\leq n\leq N} \delta(CDS_{i,n_3})}{N} - GC3_{ideal}, \tag{5}$$

where N represents the number of codons, $\delta(CDS_{i,n_3})$ a cost function that measures the presence of G/C at the third position of the nth codon in CDS_i ($\delta = 1$ if a G or C is observed, $\delta = 0$ otherwise), and $GC3_{ideal}$ is the reference GC3 ratio of the host. Once computed the GC3 score for each CDS, the maximum GC3 difference (MGC3) of the solution can be calculated in the following way:

$$MGC3 = \max_{1\leq i\leq I} |GC3(CDS_i)| . \tag{6}$$

Lower MGC3 values denote less GC divergence in the encoded CDSs with regard to the natural ratio of the host, thus being preferable in this context. Therefore, this objective must be minimized. As previously reasoned for the other objectives, using mean GC3 values instead of MGC3 could hide outliers, negatively affecting the optimization of this criterion as a consequence.

Algorithm 1 CAI mutation operator.

```
Input: $S$: solution to be mutated, $W$: usage frequency weights, $P_m$: mutation probability
Output: $S_m$: mutated solution
$S_m \leftarrow S$
$CDS \leftarrow$ Identify CDS with the Lowest CAI ($S$, $W$)
for each $Codon \in CDS$ do
    $Synonyms_{Codon} \leftarrow$ Get List of Synonymous Codons ($Codon$)
    if $Prob(P_m|100)$ and $|Synonyms_{Codon}| > 1$ and $W(Codon) \neq 1$ then
        $NewCodon \leftarrow$ Select Random Synonym ($Synonyms_{Codon}$)
        while $W(NewCodon) <= W(Codon)$ do
            $NewCodon \leftarrow$ Select Next Synonym ($Synonyms_{Codon}$)
        end while
        $S_m \leftarrow$ Replace Codon ($S_m$, $CDS$, $Codon$, $NewCodon$)
    end if
end for
```

3. Multi-objective butterfly optimization algorithm

In order to codify accurate CDSs for a target protein, we define a novel multi-objective approach that undertakes the encoding process through different problem-oriented operators coupled with bioinspired algorithmic strategies. This section details the search operators devised to generate new candidate solutions for this problem and presents the MOBOA optimization framework.

3.1. Search operators

A key element to attain accurate optimization capabilities lies on the definition of suitable operators that exploit the characteristics of the tackled problem. Following these characteristics, we have devised different search operators based on the concept of mutation. A mutation modifies randomly selected decision variables of a candidate solution in order to obtain a new solution that can potentially be better than the original one. This idea contributes to the exploration of the problem search space. The mutation operators herein proposed (four in total) are aimed at improving solutions by considering the three main aspects targeted for optimization (mCAI, mHD, MGC3), as well as introducing random variability and diversity to avoid stagnation issues. Each instantiation of the mutation procedure selects in a random way the specific operator to be applied, with the same probability for all mutations. Considering that a solution is given by a set of CDSs i.e. a set of I character strings of length L, the basic mechanism employed by the designed operators is the replacement of codons (i.e. character triplets encoding an amino acid) by one of their synonymous codons, with a mutation probability of P_m.

Algorithm 1 shows the first mutation operator, which is focused on improving the mCAI objective. This mutation targets the

Algorithm 2 HD mutation operator.

```
Input: $S$: solution to be mutated, $P_m$: mutation probability
Output: $S_m$: mutated solution
$S_m \leftarrow S$
$CDS1, CDS2 \leftarrow$ Identify CDS Pair with the Lowest HD ($S$)
for each $Codon \in CDS1$ do
  $Synonyms_{Codon} \leftarrow$ Get List of Synonymous Codons ($Codon$)
  if $Prob(P_m \mid 100)$ and $|Synonyms_{Codon}| > 1$ then
    $CurrPair\_HD \leftarrow$ Calculate Pair HD ($CDS1$, $CDS2$, $Codon$)
    $Curr\_mHD \leftarrow$ Calculate mHD ($CDS1$, $S$, $Codon$)
    $BestPair\_HD \leftarrow -1$
    $Best\_mHD \leftarrow -1$
    for each $NewSynon \in Synonyms_{Codon}$ do
      if $NewSynon \neq CurrSynon$ then
        $NewCDS1 \leftarrow$ Change Synonymous Codon ($CDS1$, $CurrSynon$, $NewSynon$)
        $NewPair\_HD \leftarrow$ Calculate Pair HD ($NewCDS1$, $CDS2$, $Codon$)
        $New\_mHD \leftarrow$ Calculate mHD ($NewCDS1$, $S$, $Codon$)
        if $New\_mHD > Curr\_mHD$ and $New\_mHD > Best\_mHD$ then
          $Best\_mHD \leftarrow New\_mHD$
          $BestSynon \leftarrow NewSynon$
        else if $Best\_mHD = -1$ and $New\_mHD = Curr\_mHD$ and $NewPair\_HD > CurrPair\_HD$ and $NewPair\_HD > BestPair\_HD$ then
          $BestPair\_HD \leftarrow NewPair\_HD$
          $BestSynon \leftarrow NewSynon$
        end if
      end if
    end for
    if $Best\_mHD \neq -1$ or $BestPair\_HD \neq -1$ then
      $S_m \leftarrow$ Replace Codon ($S_m$, $CDS1$, $CurrSynon$, $BestSynon$)
    end if
  end if
end for
```

CDS with the lowest CAI value, so each codon in this CDS will be replaced with a better adapted synonymous codon under a probability of P_m. More specifically, the new synonymous codon is randomly taken from the list of all synonymous codons with better adaptation than the original codon. While processing the CDS, if one of the codons selected for replacement is already the best one ($W(Codon) = 1$) or has not synonymous codons, the replacement of such codon will not be effective. Therefore, the result of this first mutation is a mutated solution with a better or equal mCAI value.

The second mutation operator pursues the attainment of a new mutated solution with better mHD properties than the original one. Algorithm 2 illustrates the steps of this operator. In this case, the CDSs subject to mutation are those that show the lowest HD in the solution, that is, the most similar pair of CDSs. One of the CDSs in this pair is mutated such that each codon that compose it is replaced by a different synonymous codon with a probability of P_m. Due to the fact that changing codons in a CDS has a potential impact in the mHD value of the whole solution, it must be ensured that, with each codon change, the HD values between the mutated CDS and the other ones do not experiment a worsening. The replacement will take place 1) if the mHD value of the whole solution is improved or 2) if an improvement is observed in the HD value of the targeted pair and the mHD value of the solution is not deteriorated. The result of this second mutation is consequently a mutated solution with a better or equal mHD value.

Algorithm 3 GC3 mutation operator.

```
Input: $S$: solution to be mutated, $GC3_{ideal}$: host GC3 content, $P_m$: mutation probability
Output: $S_m$: mutated solution
$S_m \leftarrow S$
$CDS, GC3 \leftarrow$ Identify CDS with the Highest GC3 Difference ($S$, $GC3_{ideal}$)
for each $Codon \in CDS$ do
  $Synonyms_{Codon} \leftarrow$ Get List of Synonymous Codons ($Codon$)
  if $Prob(P_m|100)$ and $|Synonyms_{Codon}| > 1$ then
    $NewCodon \leftarrow$ Select Random Synonym ($Synonyms_{Codon}$)
    while (($GC3 < 0$ and $NewCodon[3] \notin \{G, C\}$) or ($GC3 > 0$ and $NewCodon[3] \in \{G, C\}$)) do
      $NewCodon \leftarrow$ Select Next Synonym ($Synonyms_{Codon}$)
    end while
    $S_m \leftarrow$ Replace Codon ($S_m$, $CDS$, $Codon$, $NewCodon$)
    $GC3 \leftarrow$ Update GC3 ($CDS$, $GC3_{ideal}$)
  end if
end for
```

Algorithm 4 Random mutation operator.

```
Input: $S$: solution to be mutated, $P_m$: mutation probability
Output: $S_m$: mutated solution
$S_m \leftarrow S$
for each $CDS \in S_m$ do
  for each $Codon \in CDS$ do
    $Synonyms_{Codon} \leftarrow$ Get List of Synonymous Codons ($Codon$)

    if $Prob(P_m|100)$ and $|Synonyms_{Codon}| > 1$ then
      $NewCodon \leftarrow$ Select Random Synonym ($Synonyms_{Codon}$)

      if $NewCodon = Codon$ then
        $NewCodon \leftarrow$ Select Next Synonym ($Synonyms_{Codon}$)
      end if
      $S_m \leftarrow$ Replace Codon ($S_m$, $CDS$, $Codon$, $NewCodon$)
    end if
  end for
end for
```

The main goal of the third mutation operator, illustrated in Algorithm 3, is to improve the MGC3 value of the solution. For this purpose, the mutation targets the CDS with the largest GC3 divergence with regard to the host GC3 ratio. Each codon in this CDS is replaced by a different synonymous codon with a probability of P_m. The choice of the synonymous codon is driven by its suitability to match the host ratio: a synonymous codon with G or C at the third nucleotide will be chosen if the CDS lacks GC3 content, otherwise a synonymous codon without G or C will be used instead. If there are available multiple synonymous codons that improve or maintain the current GC3 value, the one to be used is chosen randomly. As a result, the mutated solution will show a better or equal MGC3 value.

The fourth operator is based on a random replacement approach, which is shown in Algorithm 4. This mutation considers all the CDSs in the solution, randomly replacing their codons by other synonymous codons with a probability of P_m. Since this mutation is designed to introduce variability and promote solution diversity, the new synonymous codons are chosen at random without considering any of the three aspects targeted in the previous operators.

Regarding the time complexity, the first three operators (Algorithms 1, 2, and 3) have a similar complexity: $O(Synonyms \cdot CodMut)$, where *Synonyms* is the number of synonymous codons and *CodMut* is the number of codons that are finally mutated. This number can be estimated as $CodMut = N{\cdot}P_m$, where N is the number of codons that compose the CDS and P_m is the mutation probability. Besides, the number of synonymous codons can be between 1 and 6, therefore, in the worst case, the time complexity is O(6·*CodMut*). In the case of the fourth operator (Algorithm 4), its time complexity is: O(I·*CodMut*), where I is the number of CDSs. The number of CDSs has a limited size, generally, between 2 and 10, hence, in the worst case, the time complexity is O(10·*CodMut*). In conclusion, the main factors are N (which depends on the complexity of the problem instance to solve) and the configuration parameter P_m.

3.2. MOBOA algorithmic design

The proposed method is based on BOA, a bioinspired metaheuristic that has been recently proposed for global optimization [23]. This algorithm mimics the natural behavior of butterflies, which use their sense of smell to locate food sources and mating partners. The conclusions derived from [23] point out that BOA is able to obtain significant results in comparison to other approaches on a set of 30 single-objective benchmarks and three real-world engineering problems, namely spring design, welded beam design, and gear train design. Further research has verified the relevance of this algorithm when dealing with multiple objectives in constrained optimization problems [24], four-bar truss and disk brake design [25], and feature selection for cardiovascular magnetic resonance processing [27]. Taking into account the promising search capabilities of this algorithm, we herein introduce MOBOA, a new multi-objective approach to address the protein encoding problem.

MOBOA conducts the optimization process by using a population of solutions, which are iteratively refined through different search operators in order to find new high-quality solutions to the problem. In multi-objective optimization, the goal is not to find a single optimal solution, but a Pareto set i.e. a set of solutions that optimize simultaneously the considered objective functions [29]. Candidate solutions must be evaluated according to the three objective functions considered in the problem formulation (mCAI, mHD, and MGC3) and then compared from a multi-objective perspective e.g. by using Pareto dominance. Given two solutions S_a and S_b, it is said that S_a dominates S_b if and only if S_a is better than S_b in at least one objective and it is not worse than S_b in the remaining objectives. A multi-objective optimizer will therefore pursue the set of non-dominated solutions, that is, the set of solutions that represent the best tradeoffs among objectives.

Algorithm 5 shows the algorithmic design of MOBOA. At the beginning, two data structures are declared (lines 1 and 2 in Algorithm 5): 1) a Pareto front file (*PF*) aimed at storing the non-dominated solutions found by the algorithm throughout its execution and 2) a taboo list (*TabooList*) that contains solutions that will not be employed under certain circumstances in the generation of new candidate solutions. The population structure employed by the algorithm is initialized with *PopulationSize* starter solutions (line 3), which are randomly generated with the exception of one solution that is initialized by using the codons with the best usage frequencies ($W = 1$). This initial population is sorted by calculating Pareto ranks and crowding distances (line 4), two standard measurements from [30] that are widely used in multi-objective optimization. The Pareto ranks allow classifying the solutions in different fronts considering the convergence property, using for this purpose Pareto dominance. Solutions within the same rank are ordered through the diversity-oriented crowding operator, which estimates the density of solutions surrounding each one in the front.

Algorithm 5 MOBOA pseudocode.

Input: *PopulationSize*: number of solutions in the population, *MaxCycles*: maximum number of iterations, W: usage frequency weights, $GC3_{ideal}$: host GC3 content, P_m: mutation probability, P_s: switch probability
Output: *PF*: non-dominated solutions file

```
 1: $PF \leftarrow \emptyset$
 2: $TabooList \leftarrow \emptyset$
 3: $Population \leftarrow$ Initialize Solutions ($Population$, $PopulationSize$, $W$)
 4: $Population \leftarrow$ Pareto Ranking and Crowding Sorting ($Population$, $PopulationSize$)
 5: for $Cycle \leftarrow 1$ to $MaxCycles$ do
 6:   $BestList \leftarrow$ Identify Best Solutions ($Population$, $PopulationSize$, $TabooList$)
 7:   for $i \leftarrow 1$ to $PopulationSize$ do
 8:     if $rand() < P_s$ then
 9:       $S_b \leftarrow$ Select Solution from the Best List ($BestList$)
10:       $S_m \leftarrow$ Apply Mutation ($S_b$, $W$, $GC3_{ideal}$, $P_m$)
11:       if $S_b$ dominates $S_m$ then
12:         $TabooList \leftarrow$ Update Taboo ($Population$, $TabooList \cup S_b$)
13:         $BestList \leftarrow$ Update Best Solutions ($Population$, $TabooList$, $BestList \setminus S_b$)
14:       end if
15:     else
16:       $S_m \leftarrow$ Apply Mutation ($Population[i]$, $W$, $GC3_{ideal}$, $P_m$)
17:     end if
18:     $Population \leftarrow Population \cup S_m$
19:   end for
20:   $Population \leftarrow$ Pareto Ranking and Crowding Sorting ($Population$, $2 * PopulationSize$)
21:   $PF \leftarrow$ Update Non-Dominated File ($PF$, $Population$)
22: end for
23: return $PF$
```

After these initial steps, the algorithm iterates *MaxCycles* times (line 5) over the population in order to refine it by generating new candidate solutions. Each iteration begins with the identification of the six best non-taboo solutions in the population, according to rank and crowding. These best solutions are used to define a list (*BestList*, line 6) that will be employed as a potential reference for the generation of new solutions. For each solution S_i in the population (line 7), a new candidate one is obtained by alternating the mutation procedure, with a switch probability P_s (line 8), over a solution selected from the best list (global search phase, lines 9–14) or over S_i (local search phase, lines 15–17). When applying a global search, the mutated solution is derived from a solution randomly selected from *BestList* (line 9), applying one of our mutation operators (line 10). In case the selected solution dominates the mutated one (line 11, i.e. no improvement was attained), this selected solution is moved from *BestList* to *TabooList* and the next most promising non-taboo solution from the population, following rank and crowding, is included in *BestList* to replace it (lines 12 and 13). The taboo list will be reset in case all the solutions from the population have already been considered for inclusion in *BestList*. Regarding the local search phase, the mutated solution is derived from the currently processed solution S_i (line 16). Fig. 2 depicts an example that illustrates the global and local search phases in MOBOA.

After applying the mutation, the resulting candidate solution is added to the population (line 18). These mutation steps are repeated for each solution in the population, generating *PopulationSize* new mutated solutions that must compete with the original ones. Consequently, before starting a new iteration

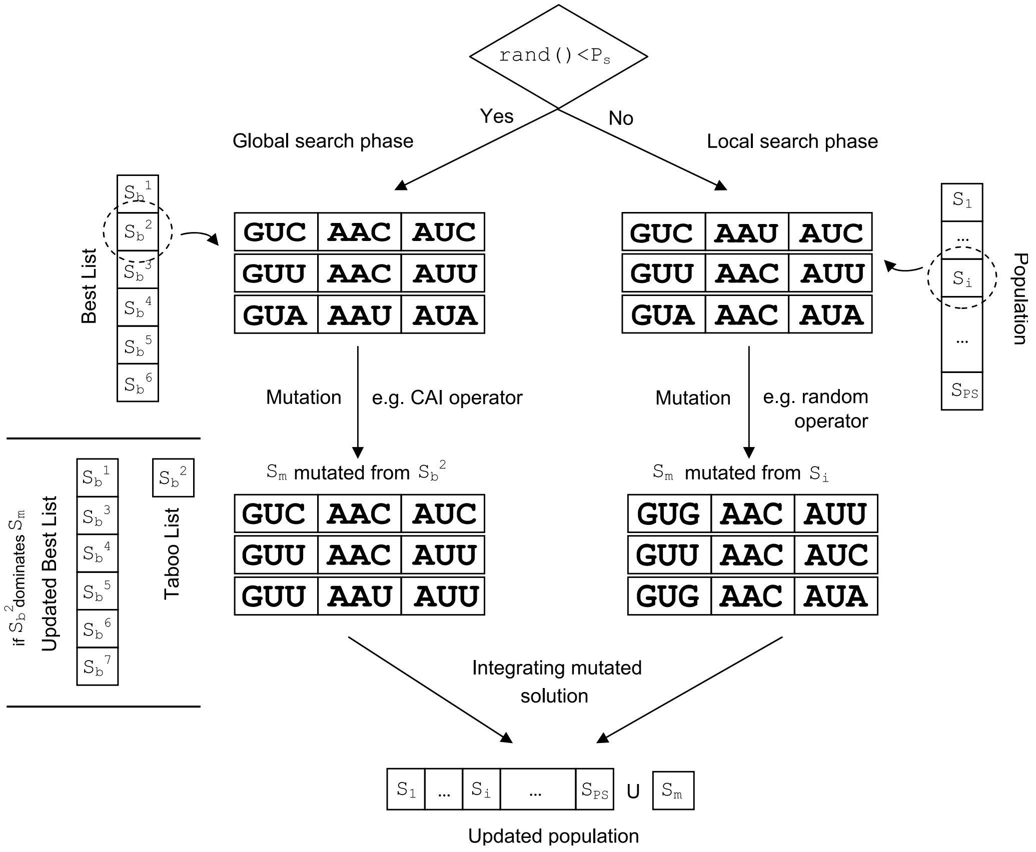


Fig. 2. Example of global and local searches in MOBOA.

of the algorithm, the population is sorted and readjusted to preserve the most promising *PopulationSize* solutions, using for this purpose Pareto ranks and crowding distances (line 20). At the end of each iteration, the *PF* structure is updated with the new non-dominated solutions available in the population (line 21). Once *MaxCycles* iterations of the main loop have been performed, the algorithm returns as output the *PF* structure containing the generated non-dominated solutions (line 23).

On analyzing the algorithm design of MOBOA, it can be observed that the multi-objective proposal inherits from BOA the switch between global and local searches, with a probability P_s, to better explore different directions of the search space and enhance the optimization process. Moreover, MOBOA includes the use of two solution lists (best and taboo solutions) coupled with Pareto-based strategies to better pursue solutions with good convergence and diversity, two key quality aspects in multi-objective optimization. Finally, the proposal incorporates problem-oriented mutation operators specifically designed to boost the quality of solutions for accurate protein encoding.

We have already detailed the exploration mechanisms used in our proposal. Regarding the exploitation mechanisms, three of the four mutation operators herein proposed improve the solutions by considering the three main aspects targeted for optimization (mCAI, mHD, MGC3), respectively. Therefore, the mutation operators clearly contribute to the exploitation by trying to improve the best solutions already known. Our proposal also uses two lists: the best list (which stores the best solutions in the population) and the taboo list (which stores solutions whose improvement failed). Both lists are very useful to lead the exploitation on the most promising solutions. The algorithm knows and maintains, in every moment, the best solutions thanks to the use of the Pareto ranks and crowding distances.

Regarding the time complexity, since, in MOBOA (Algorithm 5), 'Apply Mutation ()' is called one time per evaluation, in general, this part will govern the time complexity of MOBOA, with a final complexity of $O(10 \cdot CodMut \cdot MaxEvaluations)$, considering the worst case. In conclusion, the main factor to control is *MaxEvaluations*, that is, the maximum number of evaluations to perform.

4. Experimental results

This section provides insight into the experimental evaluation conducted to examine the results obtained by the proposal. As stated in Section 2, to the best of our knowledge, this is the first time that protein encoding is tackled as a multi-objective optimization problem involving the mCAI, mHD, and MGC3 objectives jointly. In order to conduct a comparative evaluation with other methods, we will employ as a reference other algorithmic approaches that address alternative single and multi-objective formulations of the problem. More specifically, the following state-of-the-art methods will be employed as a reference for comparisons: a web-based single-objective approach known as COOL [13], which is mainly aimed at optimizing codon adaptation indexes, and four multi-objective methods for protein encoding: Terai's method [16], MOABC [17], MOSFLA [18], and MOVNS [19]. To effectively evaluate the search strategies proposed in this paper with regard to other alternative approaches, we consider for each competing method the operators originally defined and reported in the corresponding works.

4.1. Datasets

In order to evaluate the proposal over different problem instances, nine real-world proteins, described in Table 1, have been employed in the experimentation.

Table 1
Protein instances used in the experiments.

Code	Name	CDSs	Length (AA)	CDSs*Length
Q5VZP5	DUS27_HUMAN	2	1158	2316
A4Y1B6	FADB_SHEPC	3	716	2148
B3LS90	OCA5_YEAS1	4	679	2716
B4TWR7	CAIT_SALSV	5	505	2525
Q91X51	GORS1_MOUSE	6	446	2676
Q89BP2	DAPE_BRADU	7	388	2716
A6L9J9	TRPF_PARD8	8	221	1768
Q88X33	Y1415_LACPL	9	114	1026
B7KHU9	PETG_CYAP7	10	38	380

Table 2
Parameter settings for MOBOA.

Parameter	Description	Value
PopulationSize	Number of solutions in the population	100
MaxCycles	Maximum number of iterations	100
P_m	Mutation probability	(CDSs*Length)/100%
P_s	Switch probability	50%

Table 3
Nadir and ideal values used for normalization.

Objective	Nadir value	Ideal value
mCAI	0	1
mHD	0	0.5
MGC3	0.6	0

These proteins have been selected according to two important characteristics: length (in terms of amino acids, AA) and number of CDSs needed to encode them. Due to the importance of both attributes in protein encoding complexity, the choice of these proteins was driven by the idea of maintaining a balance between both characteristics while also providing a suitable range of different problem sizes in both attributes. Each protein, in FASTA format, was obtained from the Universal Protein Resource (UniProt) website.[2] The host organism targeted in these experiments was *Saccharomyces cerevisiae*, in accordance with previous research [16] from which codon usage frequencies were taken. The Kazusa DNA Research Institute establishes the reference GC3 ratio in this host genome to 38.10% [31].

4.2. Implementation, experimentation and parameter settings

The configuration of the methods tested in this work considered the following guidelines. For the case of population-based methods (including MOBOA), the size of the population was established to 100 individuals (solutions). The stop criterion was set to 10000 fitness evaluations for all the methods under comparison. In this way, we ensure that all the evaluated methods perform the same number of fitness evaluations to guarantee the fairness of the comparison. As studied in [16], a stop criterion involving 10000 fitness evaluations is enough to allow the attainment of competitive results in this problem, in terms of multi-objective and biological quality. The rest of the parameters for each method were set according to the values recommended by their authors. Table 2 summarizes the parameter settings employed in MOBOA, which were configured via parametric studies.

Throughout the experimentation, 31 independent runs per protein instance were performed to ensure the statistical reliability of the attained results. Table 3 provides the ideal and nadir values for each objective function, which were used to normalize objective scores to the range [0,1] with the aim of avoiding potential bias. This value normalization procedure has been used when calculating the metrics employed in this experimental evaluation.

Regarding implementation details, the C/C++ programming language has been used to implement MOBOA.

4.3. Results and comparative evaluation

As previously introduced, the results obtained by MOBOA are validated through a comparative evaluation with different state-of-the-art tools, including both single-objective approaches (COOL [13]) and multi-objective methods (Terai's method [16], MOABC [17], MOSFLA [18], and MOVNS [19]). These comparisons have been performed by evaluating the results reported by each method using the three objective functions considered in the tackled problem formulation (mCAI, mHD, and MGC3). In order to evaluate the attained solutions attending to different properties, four widely adopted performance metrics are employed in this experimental evaluation, namely hypervolume (HV), maximum spread (MS), set coverage (SC), and the distance to the ideal point [28]. Furthermore, the biological assessment of the solutions is conducted by inspecting how each method is able to satisfy the different biological properties herein considered.

Our evaluation methodology also involves statistical tests [32] to determine if the proposed method is able to attain statistically significant improvements over the state of the art, with a confidence level of 95%. More specifically, the parametric test of analysis of variance (ANOVA) is applied in case the compared samples verify two conditions: 1) they follow a normal distribution (according to the Kolmogorov–Smirnov test) and they show equal variances (according to the Levene test). Otherwise, the nonparametric test of Mann–Whitney U is applied if any of these conditions is not satisfied. Due to the characteristics of the considered performance metrics, these statistical tests will be applied to examine HV and MS samples, while SC and the distances to the ideal point will be calculated by using as a reference the median-hypervolume fronts.

4.3.1. Hypervolume

HV is a widely-adopted multi-objective metric that quantifies the region of the objective space covered by the output of a multi-objective optimizer. Since we are dealing with a problem with three objective functions, HV in our case calculates the volume of a three-dimensional objective space covered by the reported Pareto solutions. Higher HV values denote better multi-objective quality. Given a set of solutions $\mathcal{A}$ reported by a multi-objective algorithm, HV can be calculated by using Eq. (7):

$$HV(\mathcal{A}, r) = Leb\left(\bigcup_{i=1}^{|\mathcal{A}|} h(a_i, r)\right), \tag{7}$$

where Leb denotes the Lebesgue measure, $|\mathcal{A}|$ the number of solutions in $\mathcal{A}$, and $h(a_i, r)$ is the volume covered by the ith solution in $\mathcal{A}$ with regard to the reference point r, which is formed with the nadir values of each objective.

Table 4 reports, for each protein instance, the median HV results obtained by MOBOA and the state-of-the-art methods, along with the corresponding quartile deviations. It can be observed that the proposed MOBOA method is able to obtain the best HV values in all the considered protein instances, with an average score of 46.49%. Although improvements are observed in all the tested proteins, it is specially remarkable the performance obtained by MOBOA for Q5VZP5, where the method reports a median HV value of 74.86% that noticeably improves the results of the competing methods COOL (0.18%), Terai (45.83%), MOABC (46.86%), MOSFLA (50.83%), and MOVNS (59.06%). Therefore, these results denote that MOBOA provides more satisfying

[2] https://www.uniprot.org/uniprot/

Table 4
Comparisons of median hypervolume results (format: median$_{\pm quartile\_deviation}$) with Terai [16], MOABC [17], MOSFLA [18], MOVNS [19], and COOL [13]. The best values for each protein instance are highlighted in bold.

Protein	MOBOA	Terai	MOABC
Q5VZP5	**74.86%**$_{\pm 0.001\%}$	45.83%$_{\pm 0.002\%}$	46.86%$_{\pm 0.004\%}$
A4Y1B6	**50.12%**$_{\pm 0.002\%}$	40.67%$_{\pm 0.001\%}$	41.05%$_{\pm 0.002\%}$
B3LS90	**49.72%**$_{\pm 0.001\%}$	40.24%$_{\pm 0.000\%}$	41.71%$_{\pm 0.002\%}$
B4TWR7	**43.10%**$_{\pm 0.001\%}$	36.37%$_{\pm 0.001\%}$	38.17%$_{\pm 0.001\%}$
Q91X51	**44.67%**$_{\pm 0.002\%}$	38.71%$_{\pm 0.003\%}$	39.76%$_{\pm 0.003\%}$
Q89BP2	**42.80%**$_{\pm 0.002\%}$	36.87%$_{\pm 0.002\%}$	38.06%$_{\pm 0.003\%}$
A6L9J9	**40.35%**$_{\pm 0.001\%}$	35.03%$_{\pm 0.003\%}$	35.91%$_{\pm 0.003\%}$
Q88X33	**36.98%**$_{\pm 0.002\%}$	30.01%$_{\pm 0.002\%}$	31.06%$_{\pm 0.003\%}$
B7KHU9	**35.82%**$_{\pm 0.004\%}$	26.81%$_{\pm 0.002\%}$	28.79%$_{\pm 0.005\%}$
Average	**46.49%**	36.73%	37.93%
Protein	MOSFLA	MOVNS	COOL
Q5VZP5	50.83%$_{\pm 0.002\%}$	59.06%$_{\pm 0.004\%}$	0.18%$_{\pm 0.000\%}$
A4Y1B6	41.88%$_{\pm 0.003\%}$	43.55%$_{\pm 0.003\%}$	0.33%$_{\pm 0.000\%}$
B3LS90	38.67%$_{\pm 0.002\%}$	42.47%$_{\pm 0.003\%}$	0.38%$_{\pm 0.000\%}$
B4TWR7	38.67%$_{\pm 0.002\%}$	37.51%$_{\pm 0.001\%}$	0.51%$_{\pm 0.000\%}$
Q91X51	40.02%$_{\pm 0.001\%}$	40.28%$_{\pm 0.004\%}$	0.21%$_{\pm 0.000\%}$
Q89BP2	38.33%$_{\pm 0.002\%}$	38.17%$_{\pm 0.003\%}$	0.25%$_{\pm 0.000\%}$
A6L9J9	36.41%$_{\pm 0.003\%}$	35.92%$_{\pm 0.003\%}$	0.51%$_{\pm 0.000\%}$
Q88X33	31.94%$_{\pm 0.003\%}$	29.85%$_{\pm 0.003\%}$	1.14%$_{\pm 0.000\%}$
B7KHU9	30.82%$_{\pm 0.005\%}$	28.32%$_{\pm 0.007\%}$	9.44%$_{\pm 0.008\%}$
Average	38.62%	39.46%	1.44%

solutions from the HV perspective, as they manage to dominate a larger volume of the objective space. In order to illustrate this point, Fig. 3 depicts the median Pareto fronts obtained by each method for two of the tested proteins, Q5VZP5 and A4Y1B6.

In order to verify if the observed improvements are statistically significant, we have performed a statistical analysis of the attained hypervolume samples. Tables 5 and 6 report the results of the different steps of the statistical procedure (Kolmogorov–Smirnov, Levene, ANOVA/Mann Whitney), along with the conclusions on statistical significance. For all the protein instances under evaluation, MOBOA achieves statistically significant improvements (p-values < 0.05) over the state-of-the-art methods. Therefore, it can be concluded that MOBOA is able to obtain significant results attending to this first performance metric.

4.3.2. Maximum spread

Secondly, the attained results are evaluated by using the MS metric, which is focused on examining the range of objective function values found by a multi-objective method. This metric is calculated by using the extreme points of the generated Pareto front $\mathcal{A}$, as expressed in Eq. (8):

$$MS(\mathcal{A}) = \sqrt{\sum_{m=1}^{M} (\max_{i=1}^{|\mathcal{A}|} f_m^i - \min_{i=1}^{|\mathcal{A}|} f_m^i)^2}, \tag{8}$$

where M refers to the number of objective functions, $|\mathcal{A}|$ the size of the non-dominated solutions set $\mathcal{A}$, $\max_{i=1}^{|\mathcal{A}|} f_m^i$ the solution showing the maximum score for the objective m, and $\min_{i=1}^{|\mathcal{A}|} f_m^i$ the solution with the minimum score for m. Higher MS scores are preferable since they denote a more diversified and extended front.

The median MS results and quartile deviations obtained by MOBOA and the state-of-the-art methods are shown in Table 7. According to this metric, our proposal is able to attain the most satisfying MS properties in overall terms, achieving the best score in 6 out of 9 protein instances while also reaching the best average value (0.78).

Statistical tests over the obtained MS samples were applied to better examine the differences observed with the other methods. The results of this statistical analysis are given in Tables 8 and 9. Comparing method by method, it can be observed that MOBOA reports statistically significant MS improvements with regard to Terai's method in 5 protein instances. Statistically significant differences are also observed in comparison to MOABC and MOSFLA in 6 instances. In this sense, it is worth remarking that the improved MS scores reported by the MOABC tool in the $Q5VZP5$ protein were found to be non-significant (p-value = 0.83), which denotes the attainment of comparable MS performance in this scenario when using MOBOA. Finally, the proposed MOBOA statistically outperforms MOVNS and COOL in 8 and 9 instances, respectively. It can then be concluded that MOBOA also accomplishes relevant performance according to this second metric.

4.3.3. Set coverage

The SC metric is a binary indicator that directly compares the fronts reported by two multi-objective optimizers. Given two solution sets $\mathcal{A}$ and $\mathcal{B}$, $SC(\mathcal{A}, \mathcal{B})$ determines how many solutions from $\mathcal{B}$ are covered by at least one solution in $\mathcal{A}$, that is, the percentage of solutions from $\mathcal{B}$ that are weakly dominated ($\succeq$) by $\mathcal{A}$. Eq. (9) illustrates this calculation:

$$SC(\mathcal{A}, \mathcal{B}) = \frac{|\{b \in \mathcal{B};\ \exists\ a \in \mathcal{A} : a \succeq b\}|}{|\mathcal{B}|}, \tag{9}$$

where $|\mathcal{B}|$ is the number of solutions in $\mathcal{B}$. A $SC(\mathcal{A}, \mathcal{B})$ value of 1 (100%) denotes that all the solutions in $\mathcal{B}$ are covered by at least one solution from $\mathcal{A}$. Conversely, if $SC(\mathcal{A}, \mathcal{B})$ is equal to 0 (0%), the solutions from $\mathcal{A}$ did not manage to cover any of the solutions in $\mathcal{B}$. As this metric is not symmetric ($SC(\mathcal{B}, \mathcal{A}) \neq 1 - SC(\mathcal{A}, \mathcal{B})$), both $SC(\mathcal{A}, \mathcal{B})$ and $SC(\mathcal{B}, \mathcal{A})$ have to be calculated to verify the degree of coverage of each optimizer over the other.

Tables 10 and 11 report the SC values derived from the pairwise comparison between MOBOA and the literature methods, using for this purpose the median-hypervolume fronts. More specifically, Table 10 includes the percentage of solutions from the state-of-the-art methods that are covered by MOBOA, while Table 11 shows the percentage of solutions from MOBOA that are covered by the remaining approaches.

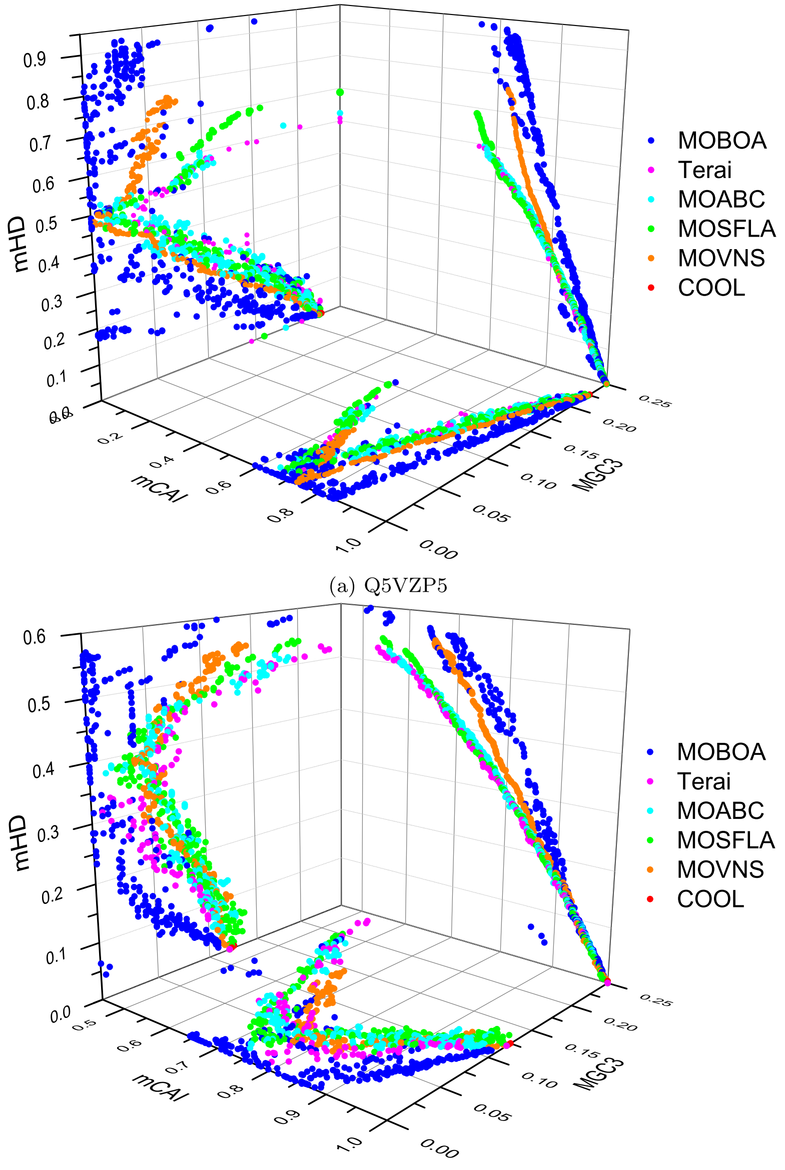
Focusing first on Table 10, MOBOA is able to significantly dominate the solutions from Terai's method, MOABC, MOSFLA, and MOVNS in most of the tested protein instances. In average terms, MOBOA covers a range between 68.10% and 97.04% of the solutions reported by these approaches. Regarding COOL, it is worth clarifying that this tool reports a small number of different solutions due to its single-objective nature, thus impacting the SC scores as expressed in Eq. (9).

On the other side, the SC values in Table 11 highlight that the state-of-the-art methods only managed to cover very low percentages of the solutions reported by MOBOA. Particularly, average coverage percentages between 0% (by COOL) and 1.16% (by MOSFLA) are observed, which emphasize the high quality of the solution sets achieved by the proposal.

4.3.4. Distance to the ideal point

In multi-objective optimization, the ideal point is defined as a utopian point that has the best scores for each objective function separately. Since the objectives are in conflict with each other, it is impossible to reach it. However, the concept of ideal point can be employed to evaluate the performance of multi-objective optimizers by identifying which method generates the closest solution to the ideal point. The information provided by this metric is also valuable to select a solution of the front for decision-making purposes.

Taking into account that all the objective values are normalized to the range [0,1], the ideal point in our case would be (1,1,0), that is, mCAI = 1, mHD = 1, and MGC3 = 0. The distance to



(a) Q5VZP5

(b) A4Y1B6

Fig. 3. Scatter plots of the median Pareto fronts from MOBOA and the state-of-the-art methods for Q5VZP5 and A4Y1B6. The plots show the different projections by pairs of objectives.

this ideal point can be measured by taking the objective scores for each solution in $\mathcal{A}$ and calculating afterwards the Euclidean distances with regard to (1,1,0). Lower distances denote better approximations to the ideal point.

For each method, the evaluation of minimum distances to the ideal point can be found in Table 12. It can be observed that MOBOA achieves the best performance from this perspective, reaching the closest solutions to the ideal point in all the scenarios herein examined. Average distances of 0.58 to the ideal point can be attained by using MOBOA, which improves the results from the alternative multi-objective tools (0.69 to 0.60) and COOL (0.99).

In conclusion, the four multi-objective metrics employed in the evaluation (HV, MS, SC, and distance to the ideal point) agree on the significant solution quality obtained by the proposed approach. The combination of bioinspired searches, Pareto-based strategies supported by the explicit distinction between best and taboo solutions, and problem-specific operators defines a suitable framework to conduct the multi-objective optimization of protein encodings, in accordance with the results achieved with regard to the state of the art.

4.3.5. Biological comparisons

The final step in this comparative analysis involves the evaluation of the three key aspects considered in protein encoding (mCAI, mHD, and MGC3), in order to inspect the biological quality of the solutions reported by each method. The extreme points of the median-hypervolume Pareto fronts are employed as a reference to conduct this comparison. Table 13 shows the results obtained for the mCAI. In this particular case, all the methods attain comparable performance, since COOL is focused on optimizing this aspect and the remaining methods include greedy solutions specifically adjusted to match the host adaptation requirements. Regarding mHD, Table 14 shows that MOBOA achieves the best results in 7 out of 9 proteins, attaining the best mHD performance in average terms (0.58). In this sense, the other methods report average mHD results that range from 0.02 (COOL) to 0.54 (MOSFLA). Finally, the comparison of MGC3 scores

Table 5
Statistical testing of hypervolume samples for Q5VZP5, A4Y1B6, B3LS90, B4TWR7, and Q91X51. "–" values denote that the distribution of samples did not meet the conditions to undergo Levene test.

Protein	Q5VZP5	A4Y1B6	B3LS90	B4TWR7	Q91X51
Kolmogorov–Smirnov test					
MOBOA	0.200	0.019	0.006	0.200	0.091
Terai's method	0.200	0.200	0.200	0.200	0.200
MOABC	0.200	0.146	0.200	0.200	0.055
MOSFLA	0.062	0.036	0.200	0.200	0.200
MOVNS	0.200	0.200	0.200	0.200	0.018
COOL	0.000	0.000	0.000	0.000	0.001
Levene test					
MOBOA vs Terai's method	0.000	–	–	0.094	0.026
MOBOA vs MOABC	0.001	–	–	0.505	0.740
MOBOA vs MOSFLA	0.000	–	–	0.923	0.055
MOBOA vs MOVNS	0.000	–	–	0.000	–
MOBOA vs COOL	–	–	–	–	–
ANOVA or Mann–Whitney p-value					
MOBOA vs Terai's method	0.000	0.000	0.000	0.000	0.000
MOBOA vs MOABC	0.000	0.000	0.000	0.000	0.000
MOBOA vs MOSFLA	0.000	0.000	0.000	0.000	0.000
MOBOA vs MOVNS	0.000	0.000	0.000	0.000	0.000
MOBOA vs COOL	0.000	0.000	0.000	0.000	0.000
Statistical significance?					
MOBOA vs Terai's method	Yes	Yes	Yes	Yes	Yes
MOBOA vs MOABC	Yes	Yes	Yes	Yes	Yes
MOBOA vs MOSFLA	Yes	Yes	Yes	Yes	Yes
MOBOA vs MOVNS	Yes	Yes	Yes	Yes	Yes
MOBOA vs COOL	Yes	Yes	Yes	Yes	Yes

Table 6
Statistical testing of hypervolume samples for Q89BP2, A6L9J9, Q88X33, and B7KHU9. "–" values denote that the distribution of samples did not meet the conditions to undergo Levene test.

Protein	Q89BP2	A6L9J9	Q88X33	B7KHU9
Kolmogorov–Smirnov test				
MOBOA	0.200	0.050	0.068	0.200
Terai's method	0.200	0.200	0.200	0.167
MOABC	0.200	0.200	0.200	0.060
MOSFLA	0.200	0.200	0.102	0.200
MOVNS	0.000	0.000	0.006	0.200
COOL	0.000	0.001	0.001	0.166
Levene test				
MOBOA vs Terai's method	0.037	0.021	0.350	0.631
MOBOA vs MOABC	0.012	0.000	0.219	0.199
MOBOA vs MOSFLA	0.000	0.024	0.114	0.231
MOBOA vs MOVNS	–	–	–	0.008
MOBOA vs COOL	–	–	–	0.000
ANOVA or Mann–Whitney p-value				
MOBOA vs Terai's method	0.000	0.000	0.000	0.000
MOBOA vs MOABC	0.000	0.000	0.000	0.000
MOBOA vs MOSFLA	0.000	0.000	0.000	0.000
MOBOA vs MOVNS	0.000	0.000	0.000	0.000
MOBOA vs COOL	0.000	0.000	0.000	0.000
Statistical significance?				
MOBOA vs Terai's method	Yes	Yes	Yes	Yes
MOBOA vs MOABC	Yes	Yes	Yes	Yes
MOBOA vs MOSFLA	Yes	Yes	Yes	Yes
MOBOA vs MOVNS	Yes	Yes	Yes	Yes
MOBOA vs COOL	Yes	Yes	Yes	Yes

in Table 15 reveals that MOBOA obtains the best results in all the tested proteins, thus going a step further with regard to the state-of-the-art methods also in this aspect.

In conclusion, the biological comparisons support the idea that MOBOA represents a valuable approach to undertake the multi-objective design of CDSs with suitable host adaptation and GC content rates, while also showing significant encoding differences to avoid homologous recombination.

Table 7
Comparisons of maximum spread median results (format: $\text{median}_{\pm quartile\_deviation}$) with Terai [16], MOABC [17], MOSFLA [18], MOVNS [19], and COOL [13]. The best values for each protein instance are highlighted in bold.

Protein	**MOBOA**	**Terai**	**MOABC**
Q5VZP5	$1.0695_{\pm 0.163}$	$1.1885_{\pm 0.003}$	$\mathbf{1.1991}_{\pm 0.002}$
A4Y1B6	$\mathbf{0.7943}_{\pm 0.011}$	$0.7344_{\pm 0.003}$	$0.7128_{\pm 0.009}$
B3LS90	$\mathbf{0.7751}_{\pm 0.004}$	$0.7225_{\pm 0.004}$	$0.7127_{\pm 0.005}$
B4TWR7	$\mathbf{0.7449}_{\pm 0.006}$	$0.7373_{\pm 0.002}$	$0.7113_{\pm 0.008}$
Q91X51	$\mathbf{0.8375}_{\pm 0.004}$	$0.7932_{\pm 0.005}$	$0.7810_{\pm 0.004}$
Q89BP2	$\mathbf{0.8010}_{\pm 0.005}$	$0.7610_{\pm 0.006}$	$0.7492_{\pm 0.006}$
A6L9J9	$\mathbf{0.7201}_{\pm 0.009}$	$0.7052_{\pm 0.002}$	$0.6882_{\pm 0.012}$
Q88X33	$0.6121_{\pm 0.008}$	$0.6387_{\pm 0.010}$	$0.6291_{\pm 0.015}$
B7KHU9	$0.6822_{\pm 0.012}$	$0.7151_{\pm 0.005}$	$0.7241_{\pm 0.026}$
Average	**0.7819**	0.7773	0.7675
Protein	**MOSFLA**	**MOVNS**	**COOL**
Q5VZP5	$0.8666_{\pm 0.188}$	$0.8660_{\pm 0.004}$	$0.0022_{\pm 0.000}$
A4Y1B6	$0.7239_{\pm 0.007}$	$0.6529_{\pm 0.005}$	$0.0036_{\pm 0.000}$
B3LS90	$0.7299_{\pm 0.007}$	$0.6508_{\pm 0.006}$	$0.0028_{\pm 0.000}$
B4TWR7	$0.7285_{\pm 0.010}$	$0.6184_{\pm 0.010}$	$0.0042_{\pm 0.000}$
Q91X51	$0.7936_{\pm 0.005}$	$0.7001_{\pm 0.009}$	$0.0000_{\pm 0.000}$
Q89BP2	$0.7608_{\pm 0.006}$	$0.6591_{\pm 0.005}$	$0.0000_{\pm 0.000}$
A6L9J9	$0.7108_{\pm 0.011}$	$0.5944_{\pm 0.009}$	$0.0000_{\pm 0.000}$
Q88X33	$\mathbf{0.6609}_{\pm 0.015}$	$0.5667_{\pm 0.018}$	$0.0162_{\pm 0.015}$
B7KHU9	$\mathbf{0.8092}_{\pm 0.015}$	$0.6729_{\pm 0.016}$	$0.1448_{\pm 0.013}$
Average	0.7538	0.6646	0.0193

5. Conclusions and future work

In this work, we introduced a novel multi-objective optimization method for encoding proteins with multiple genes. Unlike previous efforts to address this problem, the multi-objective approach herein proposed, designated as MOBOA, is the first that considers for optimization the codon adaptation, Hamming distance, and GC nucleotide content together, which represent important biological quality aspects for accurate protein encoding. MOBOA contributes with a search engine that combines bioinspired global and local searches, Pareto-based optimization supported by lists of best and taboo solutions, and a re-defined

Table 8
Statistical testing of maximum spread samples for Q5VZP5, A4Y1B6, B3LS90, B4TWR7, and Q91X51. "–" values denote that the distribution of samples did not meet the conditions to undergo Levene test.

Protein	Q5VZP5	A4Y1B6	B3LS90	B4TWR7	Q91X51
Kolmogorov–Smirnov test					
MOBOA	0.000	0.124	0.192	0.200	0.200
Terai's method	0.200	0.085	0.200	0.080	0.200
MOABC	0.200	0.200	0.200	0.200	0.200
MOSFLA	0.000	0.200	0.200	0.138	0.043
MOVNS	0.200	0.200	0.200	0.200	0.200
COOL	0.000	0.000	0.000	0.000	0.000
Levene test					
MOBOA vs Terai's method	–	0.476	0.680	0.929	0.799
MOBOA vs MOABC	–	0.212	0.741	0.632	0.876
MOBOA vs MOSFLA	–	0.215	0.282	0.272	–
MOBOA vs MOVNS	–	0.001	0.464	0.096	0.003
MOBOA vs COOL	–	–	–	–	–
ANOVA or Mann–Whitney p-value					
MOBOA vs Terai's method	0.929	0.000	0.000	0.063	0.000
MOBOA vs MOABC	0.827	0.000	0.000	0.000	0.000
MOBOA vs MOSFLA	0.000	0.000	0.000	0.000	0.000
MOBOA vs MOVNS	0.000	0.000	0.000	0.000	0.000
MOBOA vs COOL	0.000	0.000	0.000	0.000	0.000
Statistical significance?					
MOBOA vs Terai's method	No	Yes	Yes	No	Yes
MOBOA vs MOABC	No	Yes	Yes	Yes	Yes
MOBOA vs MOSFLA	Yes	Yes	Yes	Yes	Yes
MOBOA vs MOVNS	Yes	Yes	Yes	Yes	Yes
MOBOA vs COOL	Yes	Yes	Yes	Yes	Yes

Table 9
Statistical testing of maximum spread samples for Q89BP2, A6L9J9, Q88X33, and B7KHU9. "–" values denote that the distribution of samples did not meet the conditions to undergo Levene test.

Protein	Q89BP2	A6L9J9	Q88X33	B7KHU9
Kolmogorov–Smirnov test				
MOBOA	0.015	0.200	0.200	0.200
Terai's method	0.200	0.047	0.200	0.200
MOABC	0.200	0.200	0.061	0.200
MOSFLA	0.200	0.042	0.200	0.200
MOVNS	0.040	0.200	0.200	0.200
COOL	0.000	0.001	0.200	0.200
Levene test				
MOBOA vs Terai's method	–	–	0.662	0.123
MOBOA vs MOABC	–	0.038	0.057	0.005
MOBOA vs MOSFLA	–	–	0.038	0.246
MOBOA vs MOVNS	–	0.074	0.019	0.391
MOBOA vs COOL	–	–	0.769	0.017
ANOVA or Mann–Whitney p-value				
MOBOA vs Terai's method	0.000	0.025	0.000	0.006
MOBOA vs MOABC	0.000	0.000	0.014	0.000
MOBOA vs MOSFLA	0.000	0.137	0.000	0.000
MOBOA vs MOVNS	0.000	0.000	0.000	0.442
MOBOA vs COOL	0.000	0.000	0.000	0.000
Statistical significance?				
MOBOA vs Terai's method	Yes	Yes	Yes	Yes
MOBOA vs MOABC	Yes	Yes	Yes	Yes
MOBOA vs MOSFLA	Yes	No	Yes	Yes
MOBOA vs MOVNS	Yes	Yes	Yes	No
MOBOA vs COOL	Yes	Yes	Yes	Yes

Table 10
Set coverage results (SC) attained by MOBOA over Terai [16], MOABC [17], MOSFLA [18], MOVNS [19], and COOL [13].

Protein	SC(MOBOA, Terai)	SC(MOBOA, MOABC)	SC(MOBOA, MOSFLA)	SC(MOBOA, MOVNS)	SC(MOBOA, COOL)
Q5VZP5	96.84%	95.14%	97.29%	99.55%	50.00%
A4Y1B6	98.95%	93.73%	100.00%	95.20%	0.00%
B3LS90	97.80%	76.42%	82.33%	83.56%	0.00%
B4TWR7	100.00%	91.05%	85.90%	74.43%	0.00%
Q91X51	97.70%	87.28%	94.12%	78.80%	0.00%
Q89BP2	100.00%	73.97%	91.71%	70.05%	0.00%
A6L9J9	98.51%	74.53%	82.53%	42.31%	0.00%
Q88X33	92.31%	73.33%	75.66%	45.45%	0.00%
B7KHU9	91.23%	63.64%	37.89%	23.53%	100.00%
Average	97.04%	81.01%	83.05%	68.10%	16.67%

Table 11
Set coverage results (SC) attained by the state-of-the-art methods Terai [16], MOABC [17], MOSFLA [18], MOVNS [19], and COOL [13] over MOBOA.

Protein	SC(Terai, MOBOA)	SC(MOABC, MOBOA)	SC(MOSFLA, MOBOA)	SC(MOVNS, MOBOA)	SC(COOL, MOBOA)
Q5VZP5	1.13%	0.91%	1.13%	0.45%	0.00%
A4Y1B6	0.35%	0.35%	0.35%	0.35%	0.00%
B3LS90	0.57%	1.14%	1.71%	0.57%	0.00%
B4TWR7	0.53%	0.53%	0.53%	0.53%	0.00%
Q91X51	0.55%	0.55%	0.55%	0.55%	0.00%
Q89BP2	0.55%	0.55%	0.55%	0.55%	0.00%
A6L9J9	0.83%	0.83%	0.83%	1.65%	0.00%
Q88X33	1.22%	1.22%	1.22%	1.22%	0.00%
B7KHU9	3.57%	3.57%	3.57%	3.57%	0.00%
Average	1.03%	1.07%	1.16%	1.05%	0.00%

problem-oriented mutation procedure to satisfy biological quality requirements.

In order to evaluate the results obtained by the proposed algorithm, comparisons were conducted with other five alternative methods from the state of the art. For this purpose, each method was experimentally tested on nine representative real-world proteins, which were chosen according to their balance between lengths and number of CDSs. The attained results were thoroughly evaluated by using multiple performance metrics (HV, MS, SC, and distances to the ideal point) and also examining the three key biological aspects of the problem (mCAI, mHD, and MGC3). From a multi-objective perspective, the employed metrics agreed on the significant quality of the Pareto fronts reported

Table 12
Comparisons of minimum distances to the ideal point (1,1,0) with Terai [16], MOABC [17], MOSFLA [18], MOVNS [19], and COOL [13]. The best values for each protein instance are highlighted in bold.

Protein	MOBOA	Terai	MOABC	MOSFLA	MOVNS	COOL
Q5VZP5	**0.3537**	0.5937	0.5818	0.5664	0.4233	1.0230
A4Y1B6	**0.5239**	0.6416	0.6355	0.6284	0.5688	1.0041
B3LS90	**0.5362**	0.6297	0.6156	0.6138	0.5595	0.9963
B4TWR7	**0.6186**	0.6813	0.6647	0.6605	0.6244	0.9951
Q91X51	**0.6065**	0.6670	0.6506	0.6530	0.6125	1.0356
Q89BP2	**0.6065**	0.6670	0.6573	0.6597	0.6125	1.0325
A6L9J9	**0.6431**	0.6827	0.6713	0.6762	0.6434	1.0056
Q88X33	**0.6606**	0.7163	0.6935	0.6933	0.6750	0.9883
B7KHU9	**0.6845**	0.9415	0.7302	0.7145	0.6850	0.9002
Average	**0.5815**	0.6912	0.6556	0.6518	0.6005	0.9979

Table 13
Comparison of mCAI quality with Terai [16], MOABC [17], MOSFLA [18], MOVNS [19], and COOL [13]. The best values for each instance are highlighted in bold.

Protein	MOBOA	Terai	MOABC	MOSFLA	MOVNS	COOL
Q5VZP5	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9999
A4Y1B6	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9999
B3LS90	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9999
B4TWR7	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9999
Q91X51	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9999
Q89BP2	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9998
A6L9J9	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9997
Q88X33	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9989
B7KHU9	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9676
Average	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9962

Table 14
Comparison of mHD quality with Terai [16], MOABC [17], MOSFLA [18], MOVNS [19], and COOL [13]. The best values for each instance are highlighted in bold.

Protein	MOBOA	Terai	MOABC	MOSFLA	MOVNS	COOL
Q5VZP5	**0.9309**	0.6079	0.6402	0.6989	0.7737	0.0023
A4Y1B6	**0.6006**	0.5288	0.5298	0.5496	0.5559	0.0037
B3LS90	**0.6028**	0.5390	0.5479	0.5658	0.5557	0.0039
B4TWR7	**0.5333**	0.4977	0.5017	0.5126	0.4990	0.0053
Q91X51	**0.5830**	0.5426	0.5471	0.5580	0.5366	0.0030
Q89BP2	**0.5515**	0.5172	0.5189	0.5383	0.5258	0.0034
A6L9J9	**0.5128**	0.4796	0.4887	0.5015	0.4887	0.0060
Q88X33	0.4503	0.4210	0.4269	**0.4575**	0.4211	0.0117
B7KHU9	0.4912	0.4561	0.4737	**0.5134**	0.4737	0.1404
Average	**0.5840**	0.5100	0.5194	0.5440	0.5367	0.0200

Table 15
Comparison of MGC3 quality with Terai [16], MOABC [17], MOSFLA [18], MOVNS [19], and COOL [13]. The best values for each instance are highlighted in bold.

Protein	MOBOA	Terai	MOABC	MOSFLA	MOVNS	COOL
Q5VZP5	**0.0003**	0.0055	**0.0003**	0.0012	0.0012	0.2248
A4Y1B6	**0.0005**	0.0065	0.0028	0.0042	0.0019	0.1229
B3LS90	**0.0007**	0.0253	0.0204	0.0179	0.0228	0.0204
B4TWR7	**0.0013**	0.0119	0.0020	0.0046	0.0046	0.0244
Q91X51	**0.0003**	0.0221	0.0115	0.0184	0.0147	0.2800
Q89BP2	**0.0007**	0.0308	0.0207	0.0179	0.0207	0.2699
A6L9J9	**0.0015**	0.0317	0.0287	0.0241	0.0211	0.1448
Q88X33	**0.0063**	0.0375	0.0375	0.0375	0.0375	0.0083
B7KHU9	**0.0210**	0.1106	0.1106	0.1106	0.1106	0.0668
Average	**0.0036**	0.0313	0.0261	0.0263	0.0261	0.1291

by MOBOA, achieving statistically significant improvements over the state-of-the-art methods. The biological comparisons also suggested the relevance of the contributed solutions according to the three biological criteria herein considered.

Our future research directions are aimed at further exploiting the capabilities of the proposed MOBOA in other experimental scenarios. Taking into account the relevant results obtained for protein encoding, the proposed approach will be adapted and evaluated in other important bioinformatics problems, such as phylogenetic reconstruction and RNA inverse folding. Moreover, the application of MOBOA to other problems, which are not bioinformatic, is also interesting. For example, it will be analyzed its possible application to heuristic augmentation for machine/deep learning [33], a hot topic at present, where data augmentation is improved thanks to the use of heuristics.

CRediT authorship contribution statement

Belen Gonzalez-Sanchez: Conceptualization, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Visualization. **Miguel A. Vega-Rodríguez:** Conceptualization, Methodology, Resources, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition. **Sergio Santander-Jiménez:** Conceptualization, Writing – review & editing, Visualization, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was partially funded by the MCIU (Ministry of Science, Innovation and Universities, Spain), the AEI (State Research Agency, Spain), and the ERDF (European Regional Development Fund, EU), under the contract PID2019-107299GB-I00/AEI/10.13039/501100011033 (Multi-HPC-Bio project). It was also partially funded by the Junta de Extremadura (Spain) and the ERDF under the contract IB20047 (iARN project).

References

[1] A. Vassileva, D.A. Chugh, S. Swaminathan, N. Khanna, Expression of hepatitis B surface antigen in the methylotrophic yeast pichia pastoris using the GAP promoter, J. Biotechnol. 88 (1) (2001) 21–35, http://dx.doi.org/10.1016/S0168-1656(01)00254-1.

[2] H. Hohenblum, B. Gasser, M. Maurer, N. Borth, D. Mattanovich, Effects of gene dosage, promoters, and substrates on unfolded protein stress of recombinant Pichia pastoris, Biotechnol. Bioeng. 85 (4) (2004) 367–375, http://dx.doi.org/10.1002/bit.10904.

[3] P. Gu, F. Yang, T. Su, Q. Wang, Q. Liang, Q. Qi, A rapid and reliable strategy for chromosomal integration of gene(s) with multiple copies, Sci. Rep. 5, Article number 9684 (2015) 1–9, http://dx.doi.org/10.1038/srep09684.

[4] C.A. Scorer, J.J. Clare, W.R. McCombie, M.A. Romanos, K. Sreekrishna, Rapid selection using G418 of high copy number transformants of pichia pastoris for high-level foreign gene expression, Bio/Technology 12 (1994) 181–184, http://dx.doi.org/10.1038/nbt0294-181.

[5] K.E.J. Tyo, P.K. Ajikumar, G. Stephanopoulos, Stabilized gene duplication enables long-term selection-free heterologous pathway expression, Nature Biotechnol. 27 (2009) 760–765, http://dx.doi.org/10.1038/nbt.1555.

[6] R. Aw, K.M. Polizzi, Can too many copies spoil the broth? Microb. Cell Factories 12 (2013) 128, http://dx.doi.org/10.1186/1475-2859-12-128, 1–9.

[7] J. Athey, A. Alexaki, E. Osipova, A. Rostovtsev, L.V. Santana-Quintero, U. Katneni, V. Simonyan, C. Kimchi-Sarfaty, A new and updated resource for codon usage tables, BMC Bioinformatics 18 (2017) 391, http://dx.doi.org/10.1186/s12859-017-1793-7, 1–10.

[8] H. Song, H. Gao, J. Liu, P. Tian, Z. Nan, Comprehensive analysis of correlations among codon usage bias, gene expression, and substitution rate in Arachis duranensis and Arachis ipaënsis orthologs, Sci. Rep. 7 (2017) 14853, http://dx.doi.org/10.1038/s41598-017-13981-1, 1–12.

[9] S. Chen, K. Li, W. Cao, J. Wang, T. Zhao, Q. Huan, Y.-F. Yang, S. Wu, W. Qian, Codon-resolution analysis reveals a direct and context-dependent impact of individual synonymous mutations on mRNA level, Mol. Biol. Evol. 34 (11) (2017) 2944–2958, http://dx.doi.org/10.1093/molbev/msx229.

[10] S. Sahoo, S.S. Das, R. Rakshit, Codon usage pattern and predicted gene expression in Arabidopsis Thaliana, Gene 721 (2019) 100012, http://dx.doi.org/10.1016/j.gene.2019.100012, 1–8.

[11] S. Vasanthi, J.F.P. Dass, Comparative genome-wide analysis of codon usage of different bacterial species infecting Oryza sativa, J. Cell. Biochem. 119 (11) (2018) 9346–9356, http://dx.doi.org/10.1002/jcb.27214.

[12] W. Wang, X. Cheng, P.J. Buske, J.A. Suzich, H. Jin, Attenuate newcastle disease virus by codon modification of the glycoproteins and phosphoprotein genes, Virology 528 (2019) 144–151, http://dx.doi.org/10.1016/j.virol.2018.12.017.

[13] J.X. Chin, B.K.-S. Chung, D.-Y. Lee, Codon optimization OnLine (COOL): A web-based multi-objective optimization platform for synthetic gene design, Bioinformatics 30 (15) (2014) 2210–2212, http://dx.doi.org/10.1093/bioinformatics/btu192.

[14] J.C. Guimaraes, M. Rocha, A.P. Arkin, G. Cambray, D-tailor: Automated analysis and design of DNA sequences, Bioinformatics 30 (8) (2014) 1087–1094, http://dx.doi.org/10.1093/bioinformatics/btt742.

[15] P. Puigbò, E. Guzmán, A. Romeu, S. Garcia-Vallvé, OPTIMIZER: A web server for optimizing the codon usage of DNA sequences, Nucleic Acids Res. 35 (suppl_2) (2007) W126–W131, http://dx.doi.org/10.1093/nar/gkm219.

[16] G. Terai, S. Kamegai, A. Taneda, K. Asai, Evolutionary design of multiple genes encoding the same protein, Bioinformatics 33 (11) (2017) 1613–1620, http://dx.doi.org/10.1093/bioinformatics/btx030.

[17] B. Gonzalez-Sanchez, M.A. Vega-Rodríguez, S. Santander-Jiménez, J.M. Granado-Criado, Multi-objective artificial Bee colony for designing multiple genes encoding the same protein, Appl. Soft Comput. 74 (2019) 90–98, http://dx.doi.org/10.1016/j.asoc.2018.10.023.

[18] B. Gonzalez-Sanchez, M.A. Vega-Rodríguez, S. Santander-Jiménez, Multi-objective memetic meta-heuristic algorithm for encoding the same protein with multiple genes, Expert Syst. Appl. 136 (2019) 83–93, http://dx.doi.org/10.1016/j.eswa.2019.06.031.

[19] B. Gonzalez-Sanchez, M.A. Vega-Rodríguez, S. Santander-Jiménez, Multi-objective protein encoding: Redefinition of the problem, new problem-aware operators, and approach based on variable neighborhood search, Inform. Sci. 500 (2019) 173–189, http://dx.doi.org/10.1016/j.ins.2019.05.088.

[20] G. Sanli, S.I. Blaber, M. Blaber, Reduction of wobble-position GC bases in corynebacteria genes and enhancement of PCR and heterologous expression, J. Mol. Microbiol. Biotechnol. 3 (1) (2001) 123–126.

[21] G. Kudla, L. Lipinski, F. Caffin, A. Helwak, M. Zylicz, High guanine and Cytosine content increases mRNA levels in Mammalian cells, PLoS Biol. 4 (6) (2006) 933–942, http://dx.doi.org/10.1371/journal.pbio.0040180.

[22] M.H. de Smit, J. van Duin, Secondary structure of the ribosome binding site determines translational efficiency: A quantitative analysis, Proc. Natl. Acad. Sci. 87 (19) (1990) 7668–7672, http://dx.doi.org/10.1073/pnas.87.19.7668.

[23] S. Arora, S. Singh, Butterfly optimization algorithm: A novel approach for global optimization, Soft Comput. 23 (3) (2019) 715–734, http://dx.doi.org/10.1007/s00500-018-3102-4.

[24] M.M. Ahmed, A.E. Hassanien, M. Tang, Multi-objective butterfly optimization algorithm for solving constrained optimization problems, in: LISS 2021, Springer, Singapore, 2022, pp. 389–400, http://dx.doi.org/10.1007/978-981-16-8656-6_36.

[25] S. Sharma, N. Khodadadi, A.K. Saha, F.S. Gharehchopogh, S. Mirjalili, Non-dominated sorting advanced butterfly optimization algorithm for multi-objective problems, J. Bionic Eng. 20 (2023) 819–843, http://dx.doi.org/10.1007/s42235-022-00288-9.

[26] D. Rodrigues, V.H.C. de Albuquerque, J.P. Papa, A multi-objective artificial butterfly optimization approach for feature selection, Appl. Soft Comput. 94 (2020) 106442, http://dx.doi.org/10.1016/j.asoc.2020.106442.

[27] M. Muthulakshmi, G. Kavitha, N. Aishwarya, Multi-objective butterfly optimization for feature and classifier parameter's selection in diagnosis of heart failure types using CMR images, in: 2022 IEEE Global Conference on Computing, Power and Communication Technologies, GlobConPT, 2022, pp. 01–06, http://dx.doi.org/10.1109/GlobConPT57482.2022.9938325.

[28] G.G. Yen, Z. He, Performance metric ensemble for multiobjective evolutionary algorithms, IEEE Trans. Evol. Comput. 18 (1) (2014) 131–144, http://dx.doi.org/10.1109/TEVC.2013.2240687.

[29] K. Deb, Multi-objective evolutionary algorithms, in: Springer Handbook of Computational Intelligence, Springer, 2015, pp. 995–1015, http://dx.doi.org/10.1007/978-3-662-43505-2_49.

[30] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.

[31] Kazusa DNA Research Institute, Saccharomyces cerevisiae GC contents, 2023, URL https://www.kazusa.or.jp/codon/cgi-bin/showcodon.cgi?species=4932 (Accessed 26 February 2023).

[32] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, fifth ed., Chapman & Hall/CRC, NY, USA, 2011, http://dx.doi.org/10.1201/9780429186196.

[33] D. Połap, M. Woźniak, A hybridization of distributed policy and heuristic augmentation for improving federated learning approach, Neural Netw. 146 (2022) 130–140, http://dx.doi.org/10.1016/j.neunet.2021.11.018.