Multiobjective Metaheuristic to Design RNA Sequences

Álvaro Rubio-Largo
NOVA Information Management School, Universidade Nova de Lisboa, Lisbon, Portugal

Leonardo Vanneschi
NOVA Information Management School, Universidade Nova de Lisboa, Lisbon, Portugal

Mauro Castelli
NOVA Information Management School, Universidade Nova de Lisboa, Lisbon, Portugal

This is the final, accepted version of the article published by IEEE


© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Multiobjective Metaheuristic to Design RNA Sequences

Álvaro Rubio-Largo, Leonardo Vanneschi, Mauro Castelli, Miguel A. Vega-Rodríguez

Abstract—RNA inverse folding problem is a bioinformatics problem where the objective is to find an RNA sequence that folds into a given target secondary structure. In this work, we use Evolutionary Computation to solve a new and innovative multiobjective definition of this problem. In this new multiobjective definition of the problem, we have considered the similarity between target and predicted structures as a constraint, and the three objective functions: (i) Partition Function (free energy of the ensemble), (ii) Ensemble Diversity and (iii) Nucleotides Composition. The Multiobjective Metaheuristic To Design RNA Sequences (m2dRNAs) proposed in this paper is compared against other RNA inverse folding methods published in the literature, such as RNAinverse, RNA-SSD, INFO-RNA, MODENA, NUPACK, (RNAkenstein, DSS-Opt, RNAiFOLD, antaRNA, ERD, and Eterna players). After a comprehensive comparative study on two well-known benchmarks (RFam and Eterna100), we conclude that m2dRNAs is capable of obtaining very promising results in terms of both quality of RNA designs and required runtime. The source code of m2dRNAs is available at http://arxiv.unex.es/arl/m2dRNAs-source_code.zip.

Index Terms—Multiobjective optimization, metaheuristic, RNA, inverse folding.

I. INTRODUCTION

Given a particular RNA secondary structure (target structure), finding a sequence of base pairs that would fold into this structure is known as the RNA inverse folding problem. It is a common practice to make use of this bioinformatics problem for designing non-coding RNAs, which are involved in gene regulation, chromosome replication and/or RNA modification [1], [2]. In the drug and therapeutic agents area, the design of RNA sequences is also applicable to the construction of ribozymes and riboswitches [3]. We can also find applications in nano-biotechnology in the context of building self-assembling structures from RNA molecules [4].

According to [5], the RNA inverse folding problem can be solved by using brute force. Unfortunately, its complexity grows exponentially as \(4^n\), where \(n\) is the length of the target structure. This upper bound may be refined if we take into account that paired positions have to form valid base pairs, such as: AU, UA, GC, CG, GU or UG. Therefore, given a target structure with \(u\) unpaired nucleotides and \(p\) paired nucleotides, the number of valid sequences will be \(6^{p/2}4^u\). For example, see the following secondary target structure:

\[
\text{\{}\text{\{}\text{\{}\text{\{}\text{\{}\text{\{}\text{\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{}\{\}
stochastic local search that uses an effective neighbor selection method. In the initialization, by using dynamic programming they generate a set of RNA sequences that fold into the input target structure; however, only the sequence with the lower possible energy is selected. For the selection of neighbors during the local search, they employ a kind of lookahead of one selection step applying an additional energy-based criterion. After a comprehensive comparative study, the authors conclude that INFO-RNA performed better than RNAinverse and in most cases, better than RNA-SSD. The main disadvantage of INFO-RNA is that the returned RNA sequences present a high GC-content and tend to have little resemblance with biologically active RNA.

In [13], A. Taneda applied the well-known Fast Non-Dominated Sorting Genetic Algorithm (NSGA-II) [14] to this problem: MODENA (Multi-Objective Design of Nucleic Acids). This multiobjective approach explores the approximate set of weak Pareto optimal solutions in the objective function space of two objective functions: (i) a structure stability (energy of the MFE structure of the proposed sequence) and (ii) structure similarity (distance between the MFE structure for the candidate sequence and the target structure). In [15], the author extends MODENA with pseudoknot prediction methods, such as Pfold [16] and HooKnots [17]. A new version of MODENA appears in [18]. This new version allows MODENA to design RNA sequences which fold into multiple target secondary structures.

NUPACK is a suite of programs for computational nucleic acid analysis and includes an RNA designer [19]. NUPACK uses a similar method to that of RNA-SSD, but, the goal of NUPACK is to find an RNA sequence that minimizes the ensemble defect instead of the energy of the MFE structure. The ensemble defect is the average number of nucleotides that are incorrectly paired at equilibrium relative to the specified target structure, evaluated over the Boltzmann-weighted ensemble of secondary structure [20].

In [21], the authors present a genetic algorithm for the design of RNA sequences: RNAKenstein. The authors developed RNAKenstein with the aim of finding one or more target structures at the same time, that is, to solve the multi-target inverse folding version of the problem. In comparison with other evolutionary techniques (such as MODENA). According to [21], RNAKenstein was designed with little focus on running time, choosing Python as implementation language for the ease of development and flexibility it offers. In addition, its mutation and recombination processes provide additional computational burden.

Matthes et al. [22] proposed the Dynamics in Sequence Space Optimization (DSS-Opt) algorithm. DSS-Opt uses Newtonian dynamics in the sequence space, with a negative design term and simulated annealing to optimize a sequence such that it folds into the desired secondary structure. DSS-Opt makes use of multiple scoring functions, such as (i) Nearest-Neighbor Model Energies for Sequence Compositions, (ii) Negative Design, (iii) Sequence Heterogeneity and (iv) Composition Constraint. A detailed explanation of these scoring functions appears in [22].

RNAiFOLD is a constraint programming approach developed by J. A. García-Martín et al. [23], [24]. It allows the user to specify one of three different optimization criteria: MFE, the free energy or the ensemble defect. Furthermore, it provides a wide variety of design constraints, such as base-pairs bounds, specified motifs, etc.

The EteRNA Ensemble Algorithm (also known as EteRNA Bot) is a folding approach based on strategies made by tens of thousand EteRNA players and other RNA design software (RNAInverse, INFO-RNA, and RNA-SSD) [25]. It first creates an ensemble classifier from all player strategies in the market. Then, the ensemble classifier is passed over to a sequence designer which first creates a sequence with 60% GC pairs and then keeps changing bases at random positions until it finds a sequence that gets high ensemble classifier score. Unfortunately, the EteRNA Bot webpage is no longer available. From this project, the authors introduce the Eternal100 benchmark [26], a standard set of structures to be used for challenging and evaluating the next generation of automated RNA design algorithms.

In [27], Esmaeil-Tahteri et al. introduced an evolutionary algorithm for the RNA inverse folding problem: Evolutionary RNA Design (ERD). This algorithm starts reconstructing RNA sub-sequences (pools) which corresponds to different components with different lengths. Using these pools, it builds an initial RNA sequence that is compatible with the given target structure. Then it makes use of an evolutionary algorithm to improve the quality of the sub-sequences corresponding to the components. In [28], the authors present a new version of ERD and extend it to handle some sequence and energy constraints. In the sequence constraints, one can restrict sequence positions to a fixed nucleotide or to a subset of nucleotides. As for the energy constraint, one can specify an interval for the free energy ranges of the designed sequences.

In 2015, Kleinkauf et al. proposed an ant colony optimization algorithm to design RNA structures (antaRNA) [29], [30]. Besides the structural constraint, antaRNA realizes the usage of sequence constraints and provides the user to specify a GC value constraint. In [30], the authors presents antaRNA as multi-objective; however, antaRNA only handles multiple constraints but not multiple objectives.

In this work, we follow the research line opened by A. Taneda (MODENA) [13], [15], i.e. Multiobjective Optimization and Evolutionary Algorithms (MOEA) for solving the RNA inverse folding problem. We refer to our approach as Multiobjective Metaheuristic To Design RNA Sequences (m2dRNAs).

As we have mentioned before, MODENA focuses on optimizing the structure stability and the structure similarity of the RNA sequences. However, m2dRNAs is constrained by the structural similarity and optimizes the structural stability, the prediction reliability, and the nucleotide composition. This problem formulation allows m2dRNAs not only to provide a set of stable RNA sequences, but also ensuring reliability in the prediction of structures as well as avoiding strong biases in their composition. In m2dRNAs, the objective functions

1http://www.eternagenome.org/
2http://eternabot.org.
capture certain aspects of a good solution; thus, it should be expected that the desired solutions will score relatively highly under all of the 'proxy' objectives. In the literature, we find a wide variety of problems successfully solved by using multiobjective optimization, e.g. [34], [35], [36], [37], [38].

The rest of the paper is organized as follows. Section II formulates the new multiobjective formulation of the RNA Inverse Folding problem. A detailed description of the m2dRNAs algorithm is presented in Section III. Section IV is devoted to the analysis of the experiments carried out and also a comparison with other heuristics and metaheuristics published in the literature. Finally, Section V summarizes the conclusions of the paper and discusses possible lines of future work.

II. MULTIOBJECTIVE RNA INVERSE FOLDING PROBLEM

Let x denote an RNA sequence of n nucleotides (A, C, G, U) [43]. We use i, j to define a base pair between bases xi and xj, where 1 ≤ i < j ≤ n.

A secondary structure for an RNA sequence x is a set of base pairs S = {i, j | 1 ≤ i < j ≤ n ∧ i < j − 3}. For all base pairs i, j, k, l ≤ i < k < j < l ≤ n.

i. {i, j} ∩ {k, l} = ∅.
ii. (xii, xij), (xik, xij), (xij, xij), (xik, xij), (xij, xij), (xij, xij) i.e. only Watson-Crick and wobble base pairs are allowed.

III. Bases 1 < i < j < 2 (crossed base pairs are not allowed). According to [43], the model of the Gibbs free energy approximates the free energy by postulating that the energy of the full three-dimensional structure only depends on the secondary structure, and that this turn can be broken into a sum of independent contributions from each loop in the secondary structure. In a secondary RNA structure, loops are classified by the number of interior base pairs they contain: Hairpin loop (no interior base pairs). Stacked Pairs (one interior base pair separated from the closing base pair on either side), Bulges loops (one interior base pair separated from the closing base pair on one side), Internal loops (one interior base pair separated from the closing base pair on both sides), Multiloops (two or more interior base pairs). Note that bases not accessible from any base pair are called external or unpaired.

In Figure 1, we can see an illustrative representation of the aforementioned RNA loops.

The RNA inverse folding problem consists in, given a particular RNA secondary structure S, finding an RNA sequence x of base pairs that would fold into S. In [44], Schnall-Levin et al. demonstrated that the RNA secondary structure design problem is NP-hard, please refer to [44] for more details about the NP-hardness of the problem.

In this work, we have formulated the RNA inverse folding problem as a Multiobjective Optimization Problem [45], where the final goal is to simultaneously optimize the following objective functions:

- Partition Function (f1) [46]. Given an RNA sequence x, the partition function for the ensemble of all possible secondary structures can be calculated as follows:

\[
f_1(x) = \sum_{S \in \mathcal{S}(x)} e^{-\frac{-\Delta G(S)}{RT}}
\]

where \( \Delta G \) is the Gibb's Free Energy change, R the universal gas constant, T is the absolute temperature (37°C), and \( \mathcal{S}(x) \), over which the summation is performed, is the set of all possible secondary structures. This partition function is determined by nonredundant and exhaustive recursions, a complete definition may be found in [47]. In this work, to calculate f1, we have used the ViennaRNA Package 2 (v2.3.2) (C code library) [48].

- Ensemble Diversity (f2). According to [49], an important approach to deal with uncertainty in prediction, is to provide reliability information that informs us how trustworthy a prediction is. A well-known reliability measure is known as ensemble diversity. The simplest distance measure between two structures is the base pair distance which counts the number of pairs present in one, but not both structures. Basically, the ensemble diversity represents the average base pair distance between all structures in the Boltzmann ensemble and can be expressed in terms of base pair probabilities p_{ij}:

\[
f_2(x) = \sum_{(i,j) \in x} p_{ij} \cdot (1 - p_{ij})
\]

In this work, the ViennaRNA Package 2 (v2.3.2) was used to compute f2. For a detailed mathematical formulation of the Ensemble Diversity, please refer to [49].

- Nucleotides Composition (f3). Given the designed RNA sequence x, we analyze its composition in terms of base-pairs percentages (%GC: GC/GC, %AU: AU/AU, %GU: GU/UG), unpaired bases (%uA, %uC, %uG, %uU), and total bases distribution (%A, %C, %G, %U). The first group shows the distribution over the three types of base pairs in paired positions in the target, the second group shows the nucleotide distribution for unpaired positions in the target, and the last group shows the overall nucleotide distribution in the sequence. In this way, the nucleotides composition objective function is calculated as follows:

\[
f_3(x) = \max\{\%GC, \%AU, \%UG\} + \max\{\%uA, \%uC, \%uG, \%uU\} + \max\{\%A, \%C, \%G, \%U\}
\]
As we can see, for each group (base-pairs, unpaired and total), we obtain a maximum percentage. Since $f_3$ needs to be minimized, we obtain well-balanced RNA sequences in terms of nucleotides composition.

Furthermore, every RNA sequence designed must satisfy the following constraint:

- Similarity ($\sigma$) [15]. It measures the similarity between the predicted structure of $x$ and the input target structure:

$$\sigma(x) = \frac{n-d}{n}$$

where $n$ is the total number of nucleotides in $x$, and $d$ corresponds to the number of the nucleotide positions whose structure in the designed sequence are different from that in the target structure. If $\sigma(x)=1$, then $\sigma$ indicates a perfect consensus between the predicted and the target structures.

In a more formal way, the RNA inverse folding problem may be formulated as a Multiobjective Optimization Problem:

$$\text{minimize} \quad F(x) = (f_1(x), f_2(x), f_3(x))$$

subject to

$$\sigma(x) = 1$$

$x \in \Omega$

where a solution $x$ is a succession of letters within an RNA molecules $(A, C, G, U)$: $x = \{x_1, \ldots, x_n\}$. The solution $x$ must satisfy $\sigma(x)=1$. All the possible values $(A, C, G, U)$ of each component of $x$ constitute $\Omega$ (decision space), which contains all the RNA sequences of length $n$, i.e. $|\Omega| = 4^n$ sequences.

The three objective functions ($f_1$, $f_2$ and $f_3$) need to be minimized, constituting a multi-dimensional space $F: \Omega \rightarrow \mathbb{R}^3$, which is commonly known as objective space ($\mathbb{R}^3$). For each solution $x = \{x_1, \ldots, x_n\}$ in the decision space, there exists a point $z = \{z_1, z_2, z_3\}$ in the objective space.

In multiobjective optimization is necessary to compare solutions in order to decide which one is better. Thus, two solutions are compared on the basis of whether one dominates the other solution or not. Therefore, a solution $x_1$ is said to dominate the other solution $x_2$, if and only if the following conditions are satisfied:

i. The solution $x_1$ is no worse than $x_2$ in all objective functions, or $f_i(x_1) \leq f_i(x_2)$ for all $i \in \{1, 2, 3\}$.

ii. The solution $x_1$ is strictly better than $x_2$ in at least one objective function, or $f_i(x_1) < f_i(x_2)$ for at least one index $i \in \{1, 2, 3\}$.

Given a set of solutions $P$, we denominate a solution $x^*$ as Pareto-optimal or non-dominated solution if no solution in $P$ dominates $x^*$. The set of all non-dominated solutions in $P$ is known as Pareto set, and its graphical representation as Pareto front.

MODENA optimizes two objective functions: (i) Minimum Free Energy (MFE), (ii) Similarity between the predicted and the target structures. Therefore, whereas MODENA minimized the dissimilarity between the predicted MFE structure (objective 1), our first objective focuses on optimizing the frequency of the target structure in the thermodynamic ensemble (free energy). According to [39], optimization via the Partition Function produces sequences with a very strong preference for the target structure. The second objective function (ensemble diversity) is also recommended in the literature [40] for long RNA sequences, mainly because MFE-prediction (MODENA) decreases with sequence length [41]. Our last objective function (Nucleotides Composition) tries to avoid strong biases in the composition of the designed sequences, allowing m2dRNAs to obtain diversity in the set of solutions returned.

An illustrative example will help to understand the aforementioned definition of the problem. Let us consider the following target structure (in dot-bracket notation):

$$S = (((((. . . . .))) .))$$

and the following three candidate RNA sequences:

$$x_1 = GGACUAGCGGACC$$
$$x_2 = GGCUGCGCGGACC$$
$$x_3 = ACGCCAGAAGUGA$$

The first step is to obtain their predicted secondary structures and measure its similarity against $S$:

$$\sigma(x_1) = \frac{(13-0)}{13} \rightarrow (((((. . . . .))) .))$$
$$\sigma(x_2) = \frac{(13-0)}{13} \rightarrow (((((. . . . .))) .))$$
$$\sigma(x_3) = \frac{(13-4)}{13} \rightarrow \ldots (((((. . . . .))) .))$$

As we can see, only $x_1$ and $x_2$ satisfy the similarity constraint, therefore, we continue only with them.

By using the ViennaRNA package, we compute $f_1$ and $f_2$ for the two feasible solutions ($x_1$ and $x_2$):

$$f_1(x_1) = -1.48 \quad f_2(x_1) = 1.22$$
$$f_1(x_2) = -3.47 \quad f_2(x_2) = 0.42$$

Then, we analyze the composition of $x_1$ and $x_2$ in order to compute the third objective function ($f_3$). In the following, we have highlighted the maximum percentage (%) within each group:
Recently, a new version of this approach (NSGA-III) has been proposed by Deb and Jain [51] to handle many-objective optimization problems. Ishibuchi et al. [52] compare both versions (NSGA-II and NSGA-III) on various many-objective test problems. For three-objective optimization problems (like the RNA inverse folding problem), the authors conclude that similar solution sets are obtained by the two algorithms. The main disadvantage of NSGA-III is the \( \rho \) parameter. It is not clear how \( \rho \) is calculated and this is missing from the methodology [51]. On the other hand, NSGA-II is extremely well established and has proven across an enormous number of problems. As NSGA-II and NSGA-III have been shown to be comparable for related problems [52], in this work, we have applied the NSGA-II to solve this bioinformatics problem.

### B. Chromosome Encoding and Genetic Operators

The chromosome encoding of the individual determines how the RNA inverse folding problem is structured in the proposed algorithm. It gives us the necessary knowledge to understand the behaviour of the evolutionary algorithm.

For example, MODENA [13] encodes the solutions as a string of characters defined over the RNA alphabet, making use of one crossover operator (structural n-point crossover) and two mutation operators (point accepted mutation and error diagnosis mutation). In MODENA, the three genetic operators are invoked with an equal probability.

In this work, given a target structure \( S \) in dot-bracket notation, the first step is to create two sets: base-pairs (\( B \)) and unpaired (\( U \)). As we can see in Algorithm 1, we process the target structure, storing in \( B \) and \( U \) the positions of the base-pairs or unpaired nucleotides, respectively. In Algorithm 1, the notation \(|\cdot|\) indicates the number of elements in a set.

Note that, although Algorithm 1 does not significantly affect the overall running time of m2dRNAs, it is possible to use a stack to extract the required information in \( O(n) \) time complexity.

A chromosome \( X \) is then encoded as a real-valued vector of length \(|B|+|U|\):

\[
X = \{\rho_1, \ldots, \rho_{|B|}, \rho_{|B|+1}, \ldots, \rho_{|B|+|U|}\}
\]

where \( \rho \) is a real value in the range \([0,1]\) and indicates the type of base-pair or unpaired nucleotides, depending on the position. In Algorithm 2, we present the procedure to convert an input chromosome \( X \) to an RNA sequence \( x \) that will be evaluated to check if it satisfies the similarity constraint and to compute their values of objective functions.

The initialization of individuals is performed randomly for paired positions (\( B \) set) but for selecting each unpaired position (\( U \) set), we take into account its possible related base-pairs, so, we minimize the probability of creating unnecessary loops in the structure of the new RNA sequence.

The main advantage of this chromosome encoding is to ensure a certain level of greed, allowing m2dRNAs to find sequences that fold into the target structure in a faster way.

For example, given the following target structure:

\[
S = ( ( ( . . . ) ) . )
\]

1 2 3 4 5 6 7 8 9 10 11 12 13

1. fast non-dominated sorting genetic algorithm (NSGA-II)

As a revised version of NSGA [50], the Fast Non-Dominated Sorting Genetic Algorithm (NSGA-II) is a well-known MOEA created by Deb et al. [14], the source code is available at the Kanpur Genetic Algorithms Laboratory’s webpage.

NSGA-II tries to obtain a new population (offspring population) from an original one (parent population) by applying classical genetic operators, such as selection, crossover and mutation. Then, both populations, offspring and parent, are mixed into a new population. This new population is sorted into categories (ranks) according to their relationship of dominance. After that, the best individuals are selected to create a new parent population for the next generation. In the case of having to choose among individuals with the same rank, the crowding distance of the individuals belonging to the same rank is calculated, in order to decide which are the best individuals. In NSGA-II, the diversity among non-dominated solutions is introduced by using the crowding comparison procedure, which is used in the tournament selection and during the population reduction phase.

1. [https://www.iitk.ac.in/kangal/codes.shtml](https://www.iitk.ac.in/kangal/codes.shtml)

IEEE Transactions on Evolutionary Computation
Algorithm 1: Process target structure

Input: a target structure in dot-bracket notation (S);
1 B ← ∅;
2 U ← ∅;
3 for i ← 1 to |S| do
4 if S_i = ‘(’ then U ← U ∪ i;
5 if S_i = ‘)’ then c ← 0;
6 for j ← i to |S| do
7 if S_j = ‘(’ then c ← c + 1;
8 if S_j = ‘)’ then c ← c - 1;
9 if c < 0 then B ← B ∪ (i,j);

Algorithm 2: Chromosome encoding to RNA sequence

Input: a real-valued vector (X);
Output: an RNA sequence (x);
1 x ← ∅;
2 foreach (i,j) in B do
3   x_i ← U;
4   x_j ← G;
5   if X_i < 1/6 then x_i ← G;
6   if X_i < 2/6 then x_i ← C;
7   if X_i < 4/6 then x_i ← A;
8   if X_i < 5/6 then x_i ← U;
9   if X_i < 1/4 then x_i ← C;
10  if X_i < 2/4 then x_i ← G;
11  if X_i < 3/4 then x_i ← A;
12 return x

After processing the structure S (see Algorithm 1), we obtain the following two sets:

B = { (1,13), (2,12), (3,10), (4,9) }

U = { 5, 6, 7, 8, 11 }.

As we can see, the chromosome will be a vector of 9 real-values in the range [0, 1]. For example, if we consider the following two chromosomes:

X1 = { 0.12, 0.09, 0.43, 0.23, 0.81, 0.62, 0.14, 0.31, 0.55 }

X2 = { 0.15, 0.12, 0.31, 0.27, 0.79, 0.31, 0.05, 0.29, 0.66 }

then, we could evaluate them after a proper conversion into an RNA sequence (see Algorithm 2):

\[ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \]
\[ x_1 = G \ G \ G \ A \ C \ U \ A \ C \ G \ G \ U \ A \ C \ C \]
\[ x_2 = G \ G \ C \ C \ C \ U \ G \ C \ G \ C \ G \ A \ C \ C \]

With the proposed chromosome encoding, any crossover and mutation operators for continuous optimization problems may be applied within m2dRNAs. In this work, we have selected the crossover and mutation operators proposed by Deb et al. [14]: simulated binary crossover (SBX) and polynomial mutation.

IV. EXPERIMENTAL RESULTS

This section is devoted to present a comparative study of our proposal (m2dRNAs) and well-known approaches for the RNA inverse folding problem. In the first place, we describe the methodology followed in the study. Then, we compare the performance of m2dRNAs against other approaches published in the literature.

A. Methodology

We have evaluated the performance of m2dRNAs and other RNA inverse folding approaches with two benchmarks: Rfam [13] and Eterna100 [26]. The first one is a benchmark created from the seed alignments of Rfam 9.0.21, containing a total of 29 target structures. The second benchmark (Eterna100) contains a total of 100 secondary structure design challenges that span a large range in design difficulty, from short hairpins to complex 400-nucleotide designs.

The parameter configuration of m2dRNAs is: population size of 52 individuals (m2dRNAs requires a population size multiple of 4 to perform the binary tournament), stopping criterion based on iterations (50 iterations), simulated binary crossover (η=10, p_c=0.9), polynomial mutation (η_m=5, p_m=10/n', where n' is the total number of base pairs and unpaired positions). Therefore, after 50 iterations, m2dRNAs outputs up to 52 RNA sequences; however, to perform a fair comparative study, we only select 50 sequences by using the crowding-distance procedure. It was compiled using g++ (GCC) 4.9.3.

The methods involved in the comparative study are: RNAInverase v2.3.3 [9], RNA-SSD v02/2004 [12], INFO-RNA v2.1.2 [3], MODENA v0.0.67 [13], [15], NUPACK v3.0.6 [19], iRNAkentstein v1.22/07/12 [21], DSS-Opt v01/2014 [22], RNAfold v3.1 [23], [24], anatRNA v114 [29],[30], ERD v2.0.0 [28], and Eterna players [26].

In Rfam benchmark, to make a fair comparison, for each structure, all the methods involved in the comparison were configured to return a total of 50 RNA designs. Some of these methods do not allow user to define a number of output RNA designs (RNAInverse, INFO-RNA, NUPACK, DSS-Opt), therefore, they were run 50 times to obtain a set of 50 RNA designs (like the other methods). In addition, all the methods were run with default parameters configuration and with no total time limit. To evaluate the quality of the output RNA designs obtained by the approaches in terms of partition function and ensemble diversity, we have used two well-known multiobjective metrics: the Hypervolume (HV) [53] quality indicator and the Set Coverage [54] indicator. In order to extract some useful conclusions with a certain level of statistical confidence (a significance level of 0.5%), 31 independent runs were performed per experiment by each method (therefore, 31 sets of 50 RNA designs were output by each approach), and the output set of RNA designs with the
median value of HV was used. All the approaches were run on an Intel Xeon E5-2630 v3 2.4GHz and 40GB RAM with Ubuntu 14.04.

For the Eternaf100 benchmark, we have followed the same methodology as Anderson-Lee et al. [26], that is, all structures were attempted five times by each algorithm with a total time limit of twenty-four hours.

In our comparative study, we have performed a statistical analysis of the results obtained [55]. In the first place, a test for calculating residual normality is applied, in our case, Kolmogorov-Smirnov. The main objective of this test is to check whether the values of the results follow a gaussian distribution or not. For non-gaussian distributions, we perform a non-parametric analysis, such as Kruskal-Wallis. However, if the values follow a gaussian distribution, a test to check the homogeneity of the variances (Levene test) is also carried out. Finally, if this is positive, we apply an ANOVA analysis; otherwise, we perform the Kruskal-Wallis analysis.

### B. Comparative study

This comparative study is divided into two parts. The first one compares the performance of md2RNAs and other approaches when dealing with the Rfam benchmark in terms of multibjective quality and required running time. The second one is focused on comparing the success of the approaches when solving the Eternaf100 benchmark.

We start comparing our proposal (md2RNAs) with ERD, RNAiFOLD, DSS-Opt, MODENA, INFO-RNA, RNAKenstein, antaRNA, NUPACK and RNAInverse. Unfortunately, we cannot include RNA-SSD (it is only available to run via web-server) and Eternai (it is no longer available) in this first comparison.

As we can see in Table I, there exists a total of five Rfam structures (RF00010.253.ss, RF00011.18.ss, RF00016.15.ss, RF00020.107.ss, and RF00024.16.ss) in which no approach is able to obtain a successful RNA design (in median), that is a structure with similarity ($\rho$) equal to one. According to a formal proof given in [7], these structures contain two kinds of motifs that are impossible to design for any algorithm based on the current thermodynamic model.

As we mentioned in the methodology, the HV metrics measures the quality (in terms of partition function and ensemble diversity) of the set of up to 50 successful RNA designs obtained by an algorithm. After performing 31 independent runs for each Rfam structure, in Table I, we present the median value of HV (in %) obtained by the ten approaches under study. In terms of median HV, md2RNAs is able to obtain the highest values in 21 out of the 29 Rfam structures. If we compare md2RNAs and MODENA (two multibjective approaches), we can conclude that md2RNAs overcomes the results obtained by MODENA in all feasible Rfam structures.

The three best approaches in terms of structures successfully solved are md2RNAs, ERD and RNAiFOLD. As we can observe, only md2RNAs is able to solve the remaining 24 feasible Rfam structures, whereas ERD fails in RF00018.2.ss and RNAiFOLD in RF00003.94.ss. The rest of approaches fails in three or more Rfam structures, highlighting the particularly poor performance of RNAInverse, which is only able to successfully solve 15 Rfam structures (in median).

In Figure S1, we present, for each Rfam structure, a box plot...
showing the distribution of HV obtained by each of the ten approaches under study (those structures considered impossible-to-folding were excluded). As we may see, there are some RNA inverse folding methods, such as RNAfold, NUPACK, or RNAlkab, with a high deviation of HV, which is translated in a non-reliable behavior. For our proposal (m2dRNAs), we observe that, in the vast majority of cases, a good distribution of HV in 31 independent runs is obtained.

### TABLE II

**SETS OF ALGORITHMS WHERE THE DIFFERENCES OF HV ARE STATISTICALLY NOT SIGNIFICANT (p-VALUE > 0.005).** Note that, we have highlighted (in bold) those sets of algorithms in which m2dRNAs appears.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SC(A, B) (%)</th>
<th>Algorithm</th>
<th>SC(A, B) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>nt:alpha</td>
<td>53.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>INFO-RNA</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>M2DENA</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>m2dRNAs</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>RNAfold</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>NUPACK</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>RNAinverse</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>RNAlkab</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>DSS-Opt</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>ERD</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
</tbody>
</table>

**Notation:**
- (a) nt:alpha
- (b) INFO-RNA
- (c) M2DENA
- (d) m2dRNAs
- (e) RNAfold
- (f) RNAinverse
- (g) NUPACK
- (h) RNAlkab
- (i) DSS-Opt
- (j) ERD

Due to the stochastic behavior of the RNA inverse folding methods considered in this study, we have performed a statistical analysis (see Section IV-A) to demonstrate that the differences of HV among the methods are statistically significant. The confidence level considered in this work is always 95% in the statistical tests (a significance level of 0.5% or p-value under 0.005). A summary of the statistical analysis is presented in Table II (note that the five impossible to design structures were not taken into account). More precisely, for each RFam structure, we present the sets of algorithms in which the differences of HV in 31 independent runs are not statistically significant. As we can see, there is no significant difference between m2dRNAs and INFO-RNA in one structure (RF00012.15.ss). Compared with DSS-Opt, we can see that their differences of HV are not significant in five structures: RF00004.126.ss, RF00005.1.ss, RF00007.20.ss, RF00012.15.ss, and RF00015.101.ss.

We continue comparing the methods by using the Set Coverage (SC) metrics. If we suppose two sets of non-dominated solutions A and B, the SC(A,B) indicates the fraction of non-dominated solutions in B, which are covered by the non-dominated solutions in A. To make a fair comparison among the approaches, we compare the set of non-dominated solutions (RNA designs) with median value of HV in 31 independent runs. In Table III, we present the coverage relation between every pair of methods involved in this study. As we may see, the non-dominated solutions obtained by m2dRNAs are poorly dominated by the non-dominated solutions obtained by the other methods. More precisely, we find that only DSS-Opt and INFO-RNA dominates a low percentage of solutions of m2dRNAs. In contrast, m2dRNAs dominates a large fraction of non-dominated solutions obtained by the other approaches, 68.61% in average.

In Figure 3, we present a runtime comparison among the different methods involved in the study. Given an input target structure, the running time of each method corresponds with the time spent by each method in obtaining 50 RNA sequences. First, we analyze the required runtime of each method depending on the length of the target sequences, see Figure 3(a). As we may observe, with small and medium length (length<120), all the methods except antarNA and NUPACK require a similar amount of time, highlighting the two fastest approaches: ERD and M2DENA. With medium-large target structures (length>150), the fastest approaches resulted to be

---

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SC(A, B) (%)</th>
<th>Algorithm</th>
<th>SC(A, B) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>nt:alpha</td>
<td>53.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>INFO-RNA</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>M2DENA</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>m2dRNAs</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>RNAfold</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>NUPACK</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>RNAinverse</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>RNAlkab</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>DSS-Opt</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>ERD</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
</tbody>
</table>

### TABLE III

**AVGVERAGE COVERAGE RELATION AMONG THE APPROACHES WHEN SOLVING THE RPAM STRUCTURES.**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SC(A, B) (%)</th>
<th>Algorithm</th>
<th>SC(A, B) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>nt:alpha</td>
<td>53.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>INFO-RNA</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>M2DENA</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>m2dRNAs</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>RNAfold</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>NUPACK</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>RNAinverse</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>RNAlkab</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>DSS-Opt</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
<tr>
<td>ERD</td>
<td>59.0</td>
<td>nt:alpha</td>
<td>53.0</td>
</tr>
</tbody>
</table>

**Notation:**
- (a) nt:alpha
- (b) INFO-RNA
- (c) M2DENA
- (d) m2dRNAs
- (e) RNAfold
- (f) RNAinverse
- (g) NUPACK
- (h) RNAlkab
- (i) DSS-Opt
- (j) ERD

---

IEEE Transactions on Evolutionary Computation
Fig. 3. Runtime Analysis. In (a), we show the runtime spent by each method depending on the length of the RNA structure. In (b), we present the total runtime required by each method in designing all the Rfam structures (in seconds).

MODENA, RNAiFOLD and m2dRNAs. These conclusions are corroborated with the plot shown in Figure 3(b), where it is clear that RNAiFOLD, MODENA and m2dRNAs are clearly the three fastest RNA inverse folding methods.

In Figure 4, we analyze the RNA sequences designed for the RF00009.115.ss structure. This specific structure was selected after considering the five longest Rfam sequences, which are: RF000024.16.ss (452 nucleotides), RF00011.18.ss (383 nucleotides), RF00018.2.ss (361 nucleotides), RF00010.253.ss (358 nucleotides), and RF00009.115.ss (349 nucleotides).

Discarding those impossible-to-design Rfam structures (RF00024.16.ss, RF00011.18.ss, and RF00010.253.ss), only two remain: RF00018.2.ss and RF00009.115.ss. Taking into account that both structures are of a very similar length (361 and 349 nucleotides, respectively), we decided to choose RF00009.115.ss because it was successfully solved by 7 out of the 10 algorithms under study, whereas RF00018.2.ss was only successfully solved by 2 algorithms (m2dRNAs and RNAiFOLD).

Figures 4(a) to 4(g) show the predicted structure of a selected RNA design found by the aforementioned methods, where the color scheme represents the base pair probabilities and the mountain plot represents a secondary structure in a plot of height versus position, where the height m(k) is given by the number of base pairs enclosing the base at position k. As we can see, the top-3 methods for RF00009.115.ss are: m2dRNAs, RNAiFOLD, and MODENA—This highlights the particularly good performance of our approach.

In Figure 5, we study the RNA sequences found by the different approaches for the RF00025.12.ss structure. According to Table I, it is the least favourable structure for our method. As we may observe in Figures 5(a) to 5(g), the two best methods for RF00025.12.ss are DSS-Opt and m2dRNAs, which their quality is far from the quality of the other ones.

We have analyzed so far the quality of the different methods in terms of partition function and ensemble diversity (IV and SC metrics); however, the nucleotides distribution of base pairs and nucleotides in the successful designs for each method needs to be studied in order to determine which method produce RNA sequences with distribution closer to the natural distribution. Like other authors [21], we present in Table IV

TABLE IV
AVERAGE NUCLEOTIDE DISTRIBUTION IN THE GENERATED SEQUENCES
FOUND BY EACH METHOD IN RFAM STRUCTURES

| RNAiFOLD | 66.01 | 28.91 | 6.08 | 0.1064 |
| RNAiApart | 64.64 | 30.30 | 5.05 | 0.1350 |
| MODENA | 68.54 | 23.51 | 7.95 | 0.1418 |
| m2dRNAs | 43.46 | 38.17 | 18.37 | 0.1689 |
| RNAiFOLD | 59.11 | 40.89 | 0.00 | 0.0709 |
| RNAiApart | 74.24 | 25.76 | 0.00 | 0.2200 |
| MODENA | 76.18 | 23.82 | 0.00 | 0.2398 |
| DSS-Opt | 92.29 | 7.71 | 0.00 | 0.4372 |
| RNAiFOLD | 91.85 | 5.05 | 3.15 | 0.4400 |
| INFO-RNA | 94.18 | 4.21 | 1.61 | 0.4665 |

Natural [21] 57.00 30.00 13.00

* Euclidean distance to the Natural RNA sequence reported in [21]

Unpaired (%) | A | C | G | U |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RNAiApart</td>
<td>29.46</td>
<td>23.77</td>
<td>20.59</td>
<td>26.19</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>31.74</td>
<td>25.93</td>
<td>17.44</td>
<td>24.89</td>
</tr>
<tr>
<td>INFO-RNA</td>
<td>36.58</td>
<td>21.10</td>
<td>23.93</td>
<td>18.40</td>
</tr>
<tr>
<td>m2dRNAs</td>
<td>38.60</td>
<td>19.25</td>
<td>41.52</td>
<td>7.92</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>34.30</td>
<td>29.65</td>
<td>12.80</td>
<td>23.24</td>
</tr>
<tr>
<td>RNAiApart</td>
<td>47.29</td>
<td>31.31</td>
<td>30.34</td>
<td>19.24</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>42.76</td>
<td>27.46</td>
<td>12.30</td>
<td>17.48</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>57.23</td>
<td>7.48</td>
<td>5.67</td>
<td>5.67</td>
</tr>
<tr>
<td>DSS-Opt</td>
<td>89.43</td>
<td>1.31</td>
<td>9.24</td>
<td>0.01</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>95.89</td>
<td>1.18</td>
<td>1.23</td>
<td>1.69</td>
</tr>
</tbody>
</table>

Natural [21] 30.00 30.00 30.00 30.00

* Euclidean distance to the Natural RNA sequence reported in [21]

Total (%) | A | C | G | U |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RNAiApart</td>
<td>23.17</td>
<td>22.65</td>
<td>26.57</td>
<td>27.61</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>25.85</td>
<td>24.12</td>
<td>24.06</td>
<td>26.17</td>
</tr>
<tr>
<td>m2dRNAs</td>
<td>26.21</td>
<td>26.25</td>
<td>24.37</td>
<td>23.16</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>29.80</td>
<td>23.33</td>
<td>29.36</td>
<td>17.60</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>27.90</td>
<td>29.63</td>
<td>20.31</td>
<td>22.15</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>28.02</td>
<td>32.38</td>
<td>24.24</td>
<td>15.35</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>17.02</td>
<td>36.08</td>
<td>37.28</td>
<td>9.62</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>44.40</td>
<td>23.41</td>
<td>22.93</td>
<td>9.26</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>44.19</td>
<td>23.40</td>
<td>28.20</td>
<td>2.22</td>
</tr>
<tr>
<td>RNAiFOLD</td>
<td>46.57</td>
<td>24.76</td>
<td>25.63</td>
<td>3.04</td>
</tr>
</tbody>
</table>

Natural [21] 23.00 24.00 28.00 24.00

* Euclidean distance to the Natural RNA sequence reported in [21]
Fig. 4. Analysis of the RNA sequences designed for the RF00005.115.s.s structure. Figures (a) to (g) show the structure of the RNA sequence found by INFO-RNA, MODENA, m2dRNA, RNAiFOLD, RNAkneistein, DSS-Opt, and ERD, where the color scheme represents the base pair probabilities and the mountain plot represents a secondary structure in a plot of height versus position, where the height \( m(k) \) is given by the number of base pairs enclosing the base at position \( k \). Note that antiRNA, RNAinverse, and NUPACK did not find any successful RNA design for RF00005.115.s.s.

Fig. 5. Analysis of the RNA sequences designed for the RF00025.12.s.s structure. Figures (a) to (g) show the structure of the RNA sequence found by INFO-RNA, MODENA, m2dRNA, RNAiFOLD, antiRNA, DSS-Opt, and ERD, where the color scheme represents the base pair probabilities and the mountain plot represents a secondary structure in a plot of height versus position, where the height \( m(k) \) is given by the number of base pairs enclosing the base at position \( k \). Note that RNAkneistein, RNAinverse, and NUPACK did not find any successful RNA design for RF00025.12.s.s.

the average nucleotide distribution in their designed RNA sequences. In Table IV, the first group indicates the distribution over the three types of base pairs in paired positions, the second group shows the nucleotide distribution for unpaired positions, and the last group shows the overall nucleotide distribution in the sequences. For each of the aforementioned groups we calculate the Euclidean distance between the natural distribution [21] and each method involved in this study. As we may observe, INFO-RNA, DSS-Opt, and RNAiFOLD are heavily biased towards GC base pairs, whereas m2dRNA, RNAkneistein, and ERD are closer methods to the natural distribution of base pairs. If we focus on the second group (unpaired), we find that RNAiFOLD, DSS-Opt, and MODENA are heavily biased towards A nucleotide. In contrast, we can see that our proposal (m2dRNA) is not biased towards any nucleotide, showing a distribution reasonably close to the natural distribution. Finally, if we compare the overall nucleotide distributions, we conclude that the sequences obtained by RNAinverse, RNAkneistein, and m2dRNA are very well-distributed among the four bases, and their distributions are close to the natural one.

The last comparison shows the reliability of each method.
As we mentioned in Section IV.A (Methodology), all the methods involved in the comparison were configured to return a total of 50 RNA designs; therefore, the reliability metric indicates how many RNA sequences were successfully designed by each method (in percentage).

In Figure 6, we present a comparison between m2dRNAs and the other contestant techniques in terms of reliability. As we can see, m2dRNAs and ERD are able to obtain 50 different RNA designs in almost all Pfam datasets (excluding the five impossible-to-design structures), etnaRNA, RNAfold, MODENA, DSS-Opt, and INFO-RNA output approximately half of the structures requested. Finally, RNAInverse, iRNA-kenstein, and NUPACK are able to obtain a poor pool of different solutions. The average reliability of each method for the 29 Pfam datasets is presented in Table V.

All in all, we can say that our Multiobjective Metaheuristic To design RNA Sequences is not only able to obtain a reliable set of RNA designs in a single run but also the nucleotide distribution of its designed sequences is reasonably natural.

The second benchmark tested is Eternase100. In this case, we compare our proposal (m2dRNAs) with Eternase players, ERD, iRNA-kenstein, MODENA, antaRNA, INFO-RNA, DSS-Opt, NUPACK, RNAfold, RNA-SSD, and RNAInverse. As we mentioned in Section IV-A, we have followed the same methodology as Anderson-Lee et al. [26], that is, all structures of Eternase100 were attempted five times by each algorithm with a total time limit of twenty-four hours.

In Table VI, we present the number of Eternase100 structures successfully solved by each method under study. As
we can see, the notation used for successes and failures is: ‘■’ and ‘□’, respectively. The Eterna players found at least one successful solution for almost every Eterna100 structures; however, the percentage of players capable of solving these structures is very low (<1% in 64 out of the 100 structures). As pointed by Anderson-Lee et al. [26], many Eterna100 structures were only solved by the creator. If we focus on the RNA inverse folding methods, we can see that m2DRNAs outperforms the other methods obtaining successful RNA designs in 73 structures, whereas the second and third best approaches (ERD and RNAkneinstein) solved 66 and 57 structures, respectively.

Next, we compare the required runtime of the different approaches. As we have mentioned before, for each Eterna dataset unsuccessfully solved, an algorithm spent twenty-four hours, therefore, the total runtime of the algorithms will be affected by the number of dataset failed. Note that, the results of Eterna players, MODENA, INFO-RNA, DSS-Opt, NUPACK, RNA-SSD, and RNAinverse were obtained from [26]; therefore, their runtime are not available. The total runtime, in hours:minutes:seconds format, spent by the remaining methods are: 1368:07:55 (RNAiFold), 1269:12:15 (antRNA), 1042:00:28 (RNAkneinstein), 819:22:40 (ERD), and 649:20:27 (m2DRNAs). In Table VI, we present, for each Eterna dataset successfully solved, the computational time spent by m2DRNAs (in seconds).

As we did with the Rfam benchmark, we conclude by measuring the reliability of m2DRNAs and compare it against the available contestant techniques (ERD, RNAiFold, RNAkneinstein, and antaRNA). As we can see in Figure 7, m2DRNAs is more reliable than the other methods in the vast majority of Eterna datasets, therefore, it is not only able to solve more Eterna datasets (see Table VI), but it also outputs a higher number of solutions. The average reliability of each method for the 100 Eterna datasets is presented in Table VII.

V. CONCLUSIONS AND FUTURE WORKS

To find a sequence of base pairs that would fold into a given target structure is known as the RNA Inverse Folding problem.

In this work, a Multiobjective Evolutionary Algorithm (MOEA) has been applied for solving this problem, we refer to our approach as Multiobjective Metaheuristic To Design RNA Sequences (m2DRNAs). This multiobjective approach is focused on solving a new multiobjective definition of the RNA inverse folding problem. In this new multiobjective definition of the problem, we have considered the similarity between target and predicted structures as a constraint, and focused on optimizing the following three objective functions simultaneously: (i) Partition Function (free energy of the ensemble), (ii) Ensemble Diversity, (iii) Nucleotides Composition.

We have evaluated the performance of m2DRNAs and other RNA inverse folding approaches with two benchmarks: Rfam and Eterna100. The methods involved in the comparative study are: RNAinverse, RNA-SSD, INFO-RNA, MODENA, NUPACK, RNAkneinstein, DSS-Opt, RNAiFold, antaRNA, ERD, and Eterna players. From this comparison, we can conclude that m2DRNAs is capable of obtaining very promising results, outperforming the results obtained by other methods published in the literature. In addition, the required runtime of m2DRNAs is also an advantage when the length of the target structure starts to grow.

In the last years, several MOEAs have been proposed. We may classify them into three groups: (i) dominance-based, (ii) indicator-based and (iii) decomposition-based. Therefore, a straightforward line of future work is to analyze the performance of different MOEAs when solving the multiobjective version of the RNA inverse folding problem proposed in this work. Another interesting line of future work is to study the behaviour of m2DRNAs by using different chromosome encodings (e.g. an integer encoding), mutation and crossover operators, as well as a deeper statistical analysis focusing on different objectives optimized: not only the ones used by m2DRNAs but also those used by other well-known approaches. In addition, m2DRNAs will be extended for allowing multiple target structures, variety of design constraints (base-pairs bounds, specified motifs) and pseudoknot prediction methods, such as IPKnot [16] and HotKnots [17]. Finally, another challenging future work is to analyze the scalability of diverse algorithms when they are applied to the RNA inverse folding problem using structures of increasing length.

REFERENCES


Álvaro Rubio-Largo received the Ph.D. degree in Computer Engineering from the University of Extremadura, Extremadura, Spain, in 2013. He is currently working at NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa. He has authored or coauthored 60+ publications including 25+ Journal Citation Report (JCR) papers. His main research interests include big data, machine learning and the use of evolutionary computation for solving real-world multiobjective optimization problems. He has co-organized several international workshops and formed part of technical program committee in a number of international conferences. He has also co-edited a special issue of an international JCR-indexed journal. In addition, he is an active reviewer of diverse international JCR-indexed journals.

Leonardo Vanneschi took his University degree (Laurea) in Computer Science at the University of Studies of Pisa (Italy) in 1996 (summa cum Laude) and his PhD in Computer Science at the University of Lausanne (Switzerland) in 2004 (PhD thesis honours with the Excellence Award of the Science Faculty of the University of Lausanne). He is currently an Associate Professor (Tenure) at NOVA Information Management School (Lisbon, Portugal). His main research interests concern Machine Learning, Complex Systems, Data Mining, and in particular Evolutionary Computation. He has published about 160 scientific contributions, among which 40 have appeared in top-ranked scientific journals, and 10 have been honoured with international awards. He is member of the editorial board of two internationally renowned scientific journals. He is member of the steering committee and program committee of various international conferences. He has been the editor of several international conference proceedings and of two scientific journal special issues.

Mauro Castelli obtained his Master degree in Computer Science at the University of Milano Bicocca (Italy) in 2008 ("summa cum Laude"), and his PhD at the University of Milano Bicocca in 2012. He is assistant professor at NOVA IMS, Universidade Nova de Lisboa (Portugal). His main research interests are in the field of Artificial Intelligence (in particular, Evolutionary Computation and Genetic Programming) and in the application of Machine Learning techniques to solve complex real-life problems, especially in the field of biology and medicine.

Miguel A. Vega-Rodriguez received the PhD degree in computer engineering from the University of Extremadura, Spain, in 2003. He is currently Associate Professor (accredited as Full Professor) of computer architecture in the Department of Computer and Communications Technologies, University of Extremadura. He has authored or coauthored more than 580 publications including journal papers (more than 100 JCR-indexed journal papers), books chapters, and peer-reviewed conference proceedings, for which he got several awards - such as ISDA'11 Best Paper Award, IBERGRID’11 Best Paper Award, ICEC’09 Best Paper Award, and IBA-AM’08 Best Paper Award. He has contributed to the organization of several international conferences and workshops, namely as general chair or co-chair. He has edited 10 special issues of international JCR-indexed journals. In addition, he is an editor and a reviewer of diverse international JCR-indexed journals. His main research interests include parallel and distributed computing, evolutionary computation, reconfigurable and embedded computing, and bioinformatics.

IEEE Transactions on Evolutionary Computation
Supplementary Material for “Multiobjective Metaheuristic to Design RNA Sequences”

Álvaro Rubio-Largo, Leonardo Vanneschi, Mauro Castelli, Miguel A. Vega-Rodríguez

Fig. S1. Box plots with the 31 independent runs (in terms of % HV) for each of the ten approaches in Rfam structures. Note that, we have not included the impossible-to-design Rfam structures (RP000153.ss, RP000111.18.ss, RP000161.15.ss, RP000201.107.ss, and RP000241.16.ss).