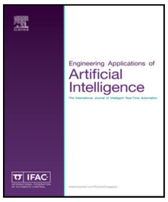
Engineering Applications of Artificial Intelligence 157 (2025) 111189

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Research paper

Exploring multiobjective evolutionary algorithms for designing Ribonucleic Acid sequences: An experimental analysis

Álvaro Rubio-Largo [a],*, Nuria Lozano-García [a], José M. Granado-Criado [b]

[a] Department of Computers and Telematics Systems Engineering, University of Extremadura, Escuela Politécnica, 10003, Cáceres, Spain
[b] Department of Computer and Communications Technologies, University of Extremadura, Escuela Politécnica, 10003, Cáceres, Spain

ARTICLE INFO

Keywords:
Bioinformatics
Ribonucleic Acid inverse folding
Multiobjective evolutionary algorithm
Multiobjective Optimization Problem
Genetic operators
Performance analysis

ABSTRACT

Evolutionary algorithms have proven effective in addressing the Ribonucleic Acid (RNA) inverse folding problem, a critical challenge in Biomedical Engineering. This problem, involving the discovery of a nucleotide RNA sequence that folds into a desired secondary structure, is formulated as a Multiobjective Optimization Problem. In this study, we introduce an approach incorporating three objective functions (Partition Function, Ensemble Diversity, and Nucleotides Composition) and a constraint (Similarity), utilizing a real-valued chromosome encoding.

The primary focus is on analyzing and comparing the performance of four multiobjective evolutionary algorithms. We explore various crossover (Simulated Binary, Differential Evolution, One-Point, Two-Point, K-Point, and Exponential) and selection (Random and Tournament) operators, coupled with a fixed mutation operator (Polynomial). Our investigation involves 48 distinct algorithm-operator combinations, with the aim of solving a well-known benchmark set.

This research makes a significant contribution to the field of Artificial Intelligence by addressing a complex problem through the lens of Multiobjective Optimization. The proposed framework not only advances our understanding of RNA inverse folding but also demonstrates the versatility of evolutionary algorithms in tackling real-world challenges in Biomedical Engineering. Our findings provide valuable insights into the behavior of different algorithmic elements and combinations, identifying optimal and suboptimal performers for future research and practical applications.

1. Introduction

Although the RNA molecule is mainly known for its role as a coding mRNA, in recent years increasing attention has been paid to functional non-coding RNAs (ncRNAs), whose functions include, among others, regulation of gene expression, splicing, translation or epigenetic control of chromatin (Hombach and Kretz, 2016). As a consequence, synthetic RNAs have found a place in areas such as drug and therapeutic agents as construction of ribozymes and riboswitches (Busch and Backofen, 2006), nano-biotechnology in the context of building self-assembling structures from RNA molecules (Qiu et al., 2013), or synthetic biology (Meyer et al., 2015). These practical applications require the design of a specific RNA molecule that performs a desired function.

A related and well-studied problem is RNA folding, which consists in predicting the most likely secondary structure of a given RNA sequence of four nucleotides: Adenine (A), Guanine (G), Cytosine (C), and Uracil (U) (primary RNA structure). The hydrogen bonds established between two specific nucleotides lead to canonical Watson–Crick base pairs (AU, UA, GC, CG) (Seeman et al., 1976; Rosenberg et al., 1976) and fundamental UG/GU wobble base pairs (Varani and McClain, 2000).

The biological function that a ncRNA molecule performs is largely determined by its 3D structure (tertiary structure), which depends on how it folds due to its base-pairing interactions (secondary structure) (Tinoco and Bustamante, 1999). Consequently, to obtain an ncRNA that fulfills a desired function in a biological system, it must spontaneously fold into the specific structure appropriate for that function. Therefore, it will be necessary to discover an ncRNA nucleotide sequence that is predicted to fold into that target structure. This matter is known as the RNA inverse folding problem (Hofacker et al., 1994). In addition, the method used for that RNA design must complete the task in a reasonable amount of time.

Abbreviations: CA, Convergence-oriented Archive; CV, Constraint Violation; DA, Diversity-oriented Archive; DBPs, Disruptive Base Pairs; ED, Ensemble Defect; HV, Hypervolume; NED, Normalized Energy Distance; SAP, Single-Action-Playout

* Corresponding author.
E-mail addresses: arl@unex.es (Á. Rubio-Largo), nulogar@unex.es (N. Lozano-García), granado@unex.es (J.M. Granado-Criado).

https://doi.org/10.1016/j.engappai.2025.111189
Received 15 December 2023; Received in revised form 15 April 2025; Accepted 14 May 2025
Available online 4 June 2025

A first approach to this problem would be to solve it by brute force (Churkin et al., 2018), although with an unapproachable growing complexity of 4^n (being n the length of the target structure). A closer inspection of the problem allows us to reduce this complexity, since the paired positions have to form the aforementioned Watson–Crick and wobble base pairs. Therefore, the number of valid sequences will be $6^{\frac{p}{2}}4^u$. As an example, in dot-bracket notation (dots symbolize unbound sites and opening and closing brackets base-pairs) we show a 17-nucleotides-long tiny structure that is composed of $p = 10$ and $u = 7$ paired and unpaired nucleotides (respectively).

```
((....))(((...)))
```

The number of RNA-compatible sequences would be $\approx 1.3 \cdot 10^8$. As we can see, despite this reduction in complexity, we need to find better ways to deal with the RNA inverse folding problem than brute force.

Evolutionary algorithms have been employed in various fields, being especially applicable to optimization, scheduling, planning, design, and management problems. As indicated in Slowik and Kwasnicka (2020) the main areas of application are: Engineering electrical electronics, Computer science artificial intelligence, Computer science theory methods, Computer science interdisciplinary applications, Automation control system, Computer science information systems, and Operations research management science. Since RNA inverse folding can be understood as an optimization problem, it is a good candidate to be solved by means of EA, so, as we will see below and in the next Section, some methods based on them have already been developed.

In a previous publication, we presented the RNA inverse folding method m2dRNAs (Rubio-Largo et al., 2019), a Multiobjective Optimization Evolutionary Algorithm (MOEA). This algorithm is based on an innovative definition of the RNA inverse folding problem as a Multiobjective Optimization Problem (MOOP). Starting from that formulation, we explore in this paper the effect of some specific modifications applied to the MOEA. Since the MOOP is the same and certain features of the MOEA remain unchanged, some theoretical explanations are very similar to those presented in Rubio-Largo et al. (2019). However, they are included here again to provide a complete theoretical framework. In addition, some of them have been expanded and/or reorganized, and new and more extensive examples are provided.

As main contributions of this work we can mention:

- A wide explanation of our formulation of the RNA inverse folding problem as a Multiobjective Optimization Problem, objective functions to be minimized, constraint, and chromosome representation, with extensive examples for a better understanding.
- Description of the elements studied: multiobjective evolutionary algorithms (NSGA-II, SMS-EMOA, NSGA-III, and C-TAEA), crossover operators (Simulated Binary, Differential Evolution, One-Point, Two-Point, K-Point, and Exponential), selection operators (Random and Tournament), and the mutation operator Polynomial.
- A comparative study of the performance of the 48 possible combinations of algorithms + operators applied to solve the RFAM benchmark set, together with an extensive set of tables, boxplots and convergence graphs highlighting the different features studied.
- From the previous point, identification of some characteristics on the behavior of the different elements studied and those that perform better and worse.
- An objective ranking of the 48 combinations studied, calculated on the basis of the results of the performance indicators, which makes it possible to determine which is the best.

2. State of the art

The first RNA inverse folding algorithm was developed in 1994 (Hofacker et al., 1994). Subsequently, it was followed by a series of methods to solve the RNA design problem, based on many different points of view. The period from 1994 to 2016 was reviewed in Rubio-Largo et al. (2019), based on which we make here a brief summary of the tools published: RNAinverse (Hofacker et al., 1994) utilizes an adaptive random walk to minimize the Hamming distance between the target structure and the Minimum Free Energy (MFE) secondary structure of the candidate RNA sequence; RNA secondary structure designer (RNA-SSD) (Andronescu et al., 2004) makes use of a stochastic local search after selecting an initial RNA sequence by means of a greedy initialization; an algorithm for the INverse FOlding of RNA (INFO-RNA) (Busch and Backofen, 2006) comprises two fundamental stages: one employing a dynamic programming approach to generate favorable initial sequences, followed by an enhanced stochastic local search; Multiobjective design of nucleic acids (MODENA) (Taneda, 2010, 2012, 2015) is based on the Non-Dominated Sorting Genetic Algorithm (NSGA-II) and the objective functions structure stability and similarity; The NUPACK software suite (Zadeh et al., 2010b,a), contains an RNA designer with resemblant features to the approach of RNA-SSD; fRNAkenstein (Lyngsø et al., 2012) is a genetic algorithm designed to solve the multitarget variant of the problem, which means simultaneously discovering one or multiple target structures; DSS-Opt, dynamics in sequence space optimization algorithm (Matthies et al., 2012) utilizes Newtonian dynamics within the sequence space, incorporating a negative design factor and applying simulated annealing to optimize a sequence's folding into the target structure; RNAiFOLD (García-Martín et al., 2013; García-Martín et al., 2015) is based on constraint programming; The EteRNA ensemble algorithm (Lee et al., 2014) is a folding method derived from the collective strategies employed by tens of thousands of EteRNA players and other RNA design software; Evolutionary RNA design (ERD) (Esmaili-Taheri et al., 2014; Esmaili-Taheri and Ganjtabesh, 2015) constructs an initial RNA sequence compatible with the specified target structure starting from pools of various components of varying lengths. Subsequently, it employs an evolutionary algorithm to enhance the quality of the subsequence segments corresponding to these components; Finally, antaRNA (Kleinkauf et al., 2015b,a) is a colony optimization algorithm that designs RNA structures managing multiple constraints.

Since then new tools have been published, from new and already proposed points of view. SIMARD (Erhan et al., 2016; Sav et al., 2016), which is based on the simulated annealing paradigm and makes use of Hamming distance and free energy to determine the fitness of a candidate sequence, has been extended in several publications. Dynamic Exploration Strategy (DES) (Hampson and Tsang, 2018) permits SIMARD to explore a broader scope of RNA sequences with minimal runtime cost, as it increases the number of mutations between evaluations (folding, computationally costly). In McBride and Tsang (2020), the performance of the adaptive AARTs, non-adaptive geometric, linear, and logarithmic simulated annealing cooling schedules is compared and analyzed on SIMARD. Finally, SIMARD-LinearFold (McBride and Tsang, 2021) tries to avoid the limiting problem of the long execution time of commonly used folding algorithms (which is due to the exponential number of possible folds) by integrating a technique called LinearFold into SIMARD. LinearFold is an approximate RNA folding algorithm, the first to run in linear time, that utilizes concepts drawn from context-free parsers from natural language processing and beam search techniques (Huang et al., 2019).

MoiRNAiFold (Minuesa et al., 2021) is an enhanced version of RNAiFold (García-Martín et al., 2013; García-Martín et al., 2015) that incorporates new modeling concepts, design constraints and quality measures necessary to design complex functional ncRNAs, as well as new heuristics and restart strategies designed for Large Neighborhood Search.

Some attempts have been made to apply the moves and strategies of participants in the EteRNA (Lee et al., 2014) RNA design game to improve automated computational RNA design using machine learning methods. SentRNA (Shi et al., 2018) comprises a fully-connected Neural Network (NN) that has been trained on solutions submitted by players, combined with an adaptive walk algorithm that integrates straightforward human design strategies to refine the initial solution sequence predicted by the NN. In Koodli et al. (2019) the authors trained a Convolutional NN (CNN) architecture using 30,477 moves from the best 72 players on a selected collection of complex puzzles. As a means of correcting some errors in the series of CNN movements located by visual inspection, a Single-Action-Playout (SAP) of six canonical strategies gathered by human players was added to the pipeline, leading to the EternaBrain-SAP method.

The Monte Carlo Search algorithms were introduced by MCTS-RNA (Yang et al., 2017). This method is based on a Monte Carlo Tree Search (MCTS) together with a Hamming distance-dependent reward function between the target and predicted secondary structures. Both are applied to find the set of essential bases that determines the secondary structure. More recently, in Cazenave and Fournier (2021) the authors adapt and evaluate different Monte Carlo Search algorithms. The first algorithm is the Nested Rollout Policy Adaptation (NRPA), that is an MCTS based recursive algorithm with adaptive rollout policy. A score function that combines the fitness of the base chain with the target structure and the dissimilitude between the target structure and the folded chain structure is used to evaluate candidate solutions. Other algorithms studied are the generalized NRPA (GNRPA) with restarts, the stabilized GNRPA and the beam GNRPA.

Reinforcement Learning (RL) defines a problem from the point of view of an agent (a NN) interacting with an environment. Two methods have been developed that apply it to design RNA sequences. In Eastman et al. (2018), where the NN is made of convolutional layers, the actions of the agent modify one by one the type of the single bases or base pairs of a candidate sequence, whereas LEARNA's (Runge et al., 2019) policy network sequentially designs an entire RNA sequence, locally adapts it and uses the Hamming distance of the result to the target structure as an error indication for the RL agent. Moreover, meta-learning on a great corpus of biological RNA sequences was applied to obtain the extension Meta-LEARNA, which creates a unique RNA design policy that can be executed in one go to solve novel RNA Design tasks. Along the training process, the architecture of the policy network was optimized.

Still in the field of machine learning, Yan et al. (2021)'s proposal is a graph-based deep generative approach to simultaneously embed and generate RNA sequences and structures, together with three interrelated benchmark tasks for RNA representation and generation: Unsupervised generation, semi-supervised learning and targeted generation. Three generative models are presented and evaluated, which are based on the variational autoencoder (VAE) framework.

There are also less conventional proposed methods. The contribution of Bellaousov et al. (2018) was to build RNA sequence databases of pre-selected helices and loops that could be used to accelerate RNA design using state-of-the-art inverse folding methods. To guide the selection of the set of helices, their thermodynamic features in natural RNA structures were used. Moreover, sequences in both databases were selected to minimize cross-hybridization. RNARedPrint (Hammer et al., 2019) is a method for multi-target RNA design. Sequences are designed while targeting specific complex features: free energies of multiple target structures and GC-content. This method couples a Fixed-Parameter Tractable (FPT) sampling algorithm with multidimensional Boltzmann sampling across distributions regulated by expressive RNA energy models. It can be used both to generate a series of sequences that fit specific target values within configurable tolerances, and to generate high quality seed sequences proper for starting RNA inverse folding methods. Finally, RNAPOND (Yao et al., 2021) incorporates positive and negative design objectives and is powered by a FPT algorithm for sequence sampling. It focuses on iteratively identifying recurring Disruptive Base Pairs (DBPs) and preventing their appearance in following rounds by incorporating suitable constraints. DBPs are base pairs and structural motifs prone to interfere with the desired folding of candidate sequences generated from positive design principles.

In addition to the formerly presented MODENA, fRNAkenstein and ERD algorithms, more Evolutionary Algorithms (EA) have been introduced. In m2dRNAs Rubio-Largo et al. (2019) the RNA inverse folding problem is formulated as a multiobjective optimization problem. Therefore, a MOEA is applied to solve it. The similarity between target and predicted structures is considered a constraint, and three objective functions are optimized simultaneously: (1) partition function (free energy of the ensemble) (McCaskill, 1990); (2) ensemble diversity (Lorenz et al., 2016); and (3) nucleotides composition. The chromosome encoding of the individual is a real-valued vector of length $|B|+|U|$, being B and U the sets of positions of base-pairs and unpaired nucleotides respectively.

MCROiRNA (Afnan et al., 2020) is a multiobjective metaheuristic algorithm adapted from Chemical Reaction Optimization (CRO) which adds non-dominated sorting and has an algorithmic scheme similar to NSGA-II (Deb et al., 2002). The objective functions and similarity as a constraint were taken from Rubio-Largo et al. (2019), the authors state. A novel operator called Repair Function was included in the CRO operators to remove invalid RNA sequences from the solution space, and the conventional ones were redesigned.

aRNAque (Merleau and Smerlak, 2021) is an evolutionary algorithm with three objective functions to minimize: Hamming distance from the target structure, Normalized Energy Distance (NED), and Ensemble Defect (ED). Local mutations step depends on the nucleotide and canonical base pair probability distribution. Instead of combining the objective functions to form a multi-objective function, they are used separately at different levels. NED and Hamming distance are used as selection weights for the sequences that will be mutated, and to choose ten best sequences that will always move to the next generation, respectively. Hence, the selection method is roulette wheel selection (Lipowski and Lipowska, 2012) (*fitness proportionate selection)*. ED is used to walk through the neutral network of the first found sequence that folds into the given target by minimizing it. An updated version of aRNAque can be found in Merleau and Smerlak (2022). It applies a Lévy flight mutation scheme (Mandelbrot, 1963), which are random walks characterized by a step size distribution that exhibits a heavy tail. Such a scheme allows exploration at different scales (local search coupled with occasional big jumps). The distribution of the number of point mutations at every step is taken to follow a Zipf distribution (Newman, 2005).

eM2dRNAs: Enhanced Multiobjective Metaheuristic for RNA Sequence Design (Rubio-Largo et al., 2023) extends the previously mentioned m2dRNAs. The key enhancement involves decomposing the target structure into smaller, more manageable substructures through a recursive process. These smaller substructures simplify the problem-solving process compared to the original structure. The decomposition generates a directed acyclic graph that captures the dependencies between the substructures. Each substructure is solved independently using an adapted version of m2dRNAs, and the results are then integrated to form a complete sequence solution. Furthermore, since the recursive decomposition employed by eM2dRNAs does not always produce an optimal dependency graph, an Evolutionary Strategy (ES) was subsequently integrated to optimize the decomposition process and improve the performance of the core MOEA, resulting in ES+eM2dRNAs (Rubio-Largo et al., 2024).

GREED-RNA (Lozano-García et al., 2024) is based on a simple greedy evolutionary strategy. Its main feature is the use of several dynamically adapting objective functions, such as base-pair distance, Hamming distance, probability over ensemble, partition function, ensemble defect, and GC-content. The weights of these objectives are adjusted according to the state of the algorithm, which changes as the process progresses. The algorithm incorporates greedy initialization and mutation steps to facilitate the generation of sequences that fold into the target structure. To expand the search space and avoid local minima, the mutation mode switches to random when stagnation is detected. In addition, GREED-RNA allows specifying a range of GC content for solution sequences, providing greater flexibility in sequence design.

An innovative optimization paradigm is Structure-Aware Multifrontier Ensemble Optimization (SAMFEO) (Zhou et al., 2023) optimizes either the equilibrium probability or the ensemble defect through an iterative multifrontier search process. This approach generates a diverse set of valid RNA sequences, including both MFE and unique MFE solutions, as byproducts. The optimization process begins with a targeted initialization, followed by iterative sampling, structured mutation, and updating. Throughout these phases, both structural and ensemble-level information are utilized.

As seen throughout this review, different functions can be used in RNA design to evaluate the quality of the results. A comparative study of the performance of some of them can be found in Ward et al. (2023).

3. RNA inverse folding problem

The Gibbs free energy model approximates the free energy of an RNA molecule by presuming that the energy of the full three-dimensional structure only depends on the secondary structure. Moreover, this spin can be decomposed into a sum of independent contributions from each loop of the secondary structure (Lyngsø, 2008b).

The RNA inverse folding problem consists in identifying an RNA sequence x of nucleotides that would fold into a specific RNA secondary structure S. Schnall-Levin et al. (2008) proved the NP-hardness of the RNA secondary structure design problem.

Let x represent an RNA sequence of n nucleotides (A, C, G, U) (Lyngsø, 2008b). A base pair between bases x_i and x_j, where $1 \leq i < j \leq n$, is defined as $i \cdot j$. A secondary structure for an RNA sequence x is a set of base pairs $S = \{i \cdot j | 1 \leq i < j \leq n \wedge i < j - 3\}$. For all base pairs $i_1 \cdot j_1, i_2 \cdot j_2 \in S$ with $i_1 \cdot j_1 \neq i_2 \cdot j_2$:

1. $\{i_1, j_1\} \cap \{i_2, j_2\} = \emptyset$;
2. $\{x_{i_1}, x_{j_1}\} \in \{\{A, U\}, \{U, A\}, \{C, G\}, \{G, C\}, \{U, G\}, \{G, U\}\}$, i.e., only canonical Watson–Crick and wobble base pairs are allowed.
3. $i_1 < i_2 < j_1 < j_2$ (meaning crossed base pairs are not allowed).

As we defined in Rubio-Largo et al. (2019), the RNA inverse folding problem can be formulated as a Multiobjective Optimization Problem (MOOP) (Collette and Siarry, 2004), whose objective is to optimize three objective functions at the same time. These functions are:

1. *Partition Function (f_1) (McCaskill, 1990):* As exposed in Hofacker (2003), optimization through this function generates sequences with an intense preference for the target structure. The partition function for the set of all possible secondary structures of a given RNA sequence x can be calculated as in Eq. (1):

   $$f_1(x) = \sum_{S \epsilon S'(x)} e^{\frac{-\Delta G(S)}{RT}} \tag{1}$$

   where $-\Delta G$ denotes the Gibbs' free energy change, R represents the universal gas constant, T is for absolute temperature (37 °C), and $S'(x)$ is the batch of all possible secondary structures, over which the summation is performed. A complete definition can be found in Lyngsø (2008a).
2. *Ensemble Diversity (f_2) (Lorenz et al., 2016):* This function is a recognized measure of the reliability of a prediction. This metric essentially represents the average base pair distance (number of pairs present in one, but not both structures) between all structures in the Boltzmann ensemble, which is the simplest distance measure between two structures, and can be expressed in terms of base pair probabilities p_{ij} as it is shown in Eq. (2):

   $$f_2(x) = \sum_{(i,j) \epsilon x} p_{ij} \cdot (1 - p_{ij}) \tag{2}$$

   For a detailed mathematical formulation of the ensemble diversity, please refer to Lorenz et al. (2016). This type of information is crucial to deal with uncertainty in prediction (Lorenz et al., 2016). Moreover, ensemble diversity is also recommended in the literature (Wilm et al., 2008) for long RNA sequences.
3. *Nucleotides Composition (f_3):* With the objective of obtaining diversity in the set of solutions, this function helps to avoid strong biases in the composition of the designed sequences. From the designed RNA sequence x, its composition is studied in terms of: (1) base-pairs percentages ($\%GC$: GC/CG, $\%AU$: AU/UA, and $\%GU$: GU/UG), (2) unpaired bases percentages ($\%uA$, $\%uC$, $\%uG$, and $\%uU$), and (3) total bases distribution ($\%A$, $\%C$, $\%G$, and $\%U$). The first category shows the distribution of the three types of base pairs along the paired positions in the target structure, the second category shows the nucleotides distribution in unpaired positions of the target structure, and the last category shows the total nucleotides distribution in the entire designed sequence. Based on these categories, the nucleotides composition objective function is calculated as shown in Eq. (3):

   $$\begin{aligned} f_3(x) = {} & \max\{\%GC, \%AU, \%UG\} \\ & + \max\{\%uA, \%uC, \%uG, \%uU\} \\ & + \max\{\%A, \%C, \%G, \%U\} \end{aligned} \tag{3}$$

   For each category (base-pairs, unpaired, and total), the maximum percentage is acquired. Since f_3 is to be minimized, well-balanced RNA sequences are designed in terms of nucleotide composition.

In this analysis, to calculate f_1 and f_2, the ViennaRNA package 2 (v2.5.1) (python library) (Lorenz et al., 2011) was utilized.

As we stated in Rubio-Largo et al. (2019), a mandatory constraint for each designed RNA is defined: *Similarity* (σ) (Taneda, 2012): It evaluates the similitude between the predicted structure of x and the target structure. To calculate this, Eq. (4) can be used:

$$\sigma(x) = \frac{n - d}{n} \tag{4}$$

where n is the length of x, and d the number of nucleotide positions whose structure in the designed sequence do not correspond with that in the target structure. If $\sigma(x) = 1$ the predicted and the target structures are identical.

Unlike other MOEAs, in our proposal similarity to the target structure is defined as a constraint rather than an objective function. Forcing the similarity to be equal to 1 guarantees that the solutions offered by the algorithm fold into the same structure as the target structure, which is not the case in other algorithms where it is not a constraint. Thus, it is designed to maintain structural similarity while optimizing structural stability (optimize the frequency of the target structure within the thermodynamic ensemble), prediction reliability, and nucleotide composition. This approach not only allows for the generation of stable RNA sequences, but also provides reliable structure predictions while minimizing compositional biases. The objective functions are tailored to capture the key features of an optimal solution, so the solutions should perform well for other objectives.

4. Multiobjective metaheuristics

4.1. Multiobjective optimization

The RNA inverse folding problem may be formulated as a MOOP with 3 objective functions for minimization (Rubio-Largo et al., 2019):

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

where x is the vector of variables in the set Ω, $F : \Omega \rightarrow Y \subset R^3$ is a vector of 3 objective functions, and Y is the *objective space*. For the case of the RNA inverse folding problem, it is needed to add the following clauses to the previously defined function $F(x)$ to minimize:

$$\text{subject to } \sigma(x) = 1$$
$$x \in \Omega$$

In this problem, a solution x is a succession of letters within an RNA molecule $(A, C, G, U) \; : \; x = \{x_1, \dots, x_n\}$. The solution x must satisfy $\sigma(x) = 1$. All the feasible values (A, C, G, U) of each component of x compose Ω *(decision space)*, which includes all RNA sequences of length n, i.e., $|\Omega| = 6^{\frac{p}{2}}4^u$. For each solution $x = \{x_1, \dots, x_n\}$ in the decision space, there is a corresponding point $y = \{y_1, y_2, y_3\}$ in the objective space.

In multiobjective optimization, one requirement is to have an objective metric to decide which solution is *better* than the others. To that end, two solutions will be compared using the criterion of *dominance*. A solution x_1 *dominates* other solution x_2, if and only if the following conditions are accomplished:

1. $f_i(x_1) \le f_i(x_2)$ for all $i \in \{1, 2, 3\}$ (the solution x_1 is no worse than x_2 in any objective function).
2. $f_i(x_1) < f_i(x_2)$ for at least one index $i \in \{1, 2, 3\}$ (the solution x_1 is strictly better than x_2 in at least one objective function).

A solution x^* is denominated as *Pareto-optimal* or *non-dominated solution* if no solution in the set of solutions P dominates x^*. The ensemble of all non-dominated solutions in P is named as *Pareto-optimal set* or simply *Pareto set*, and its graphical representation as *Pareto front*.

We offer a comprehensive example to illustrate our definition of the RNA inverse folding problem as a MOOP. The structure for which it would be desired to solve this problem will be:

$$S = \texttt{((....))(((...)))}$$

in dot-bracket notation.

For this target structure we consider five candidate sequences (obviously they must be of the same length as S):

$$x_1 = \texttt{GGGGGACCGCCGUGGGC}$$
$$x_2 = \texttt{GGGAAACCGGGAAACCC}$$
$$x_3 = \texttt{GCGACAGCGGGAAACCC}$$
$$x_4 = \texttt{GGGAAACCGGGAAACUC}$$
$$x_5 = \texttt{GCGGGACCGCCGUGGGC}$$

Since we have defined a constraint, the starting point is to check whether the candidate sequences meet it. For this purpose, it is necessary to obtain their predicted secondary structures and then calculate their similarities against S. For the first requirement in this paper we use *RNAfold* from the ViennaRNA package:

$$x_1 \Longrightarrow \texttt{((....))(((...)))}$$
$$x_2 \Longrightarrow \texttt{((....))(((...)))}$$
$$x_3 \Longrightarrow \texttt{((....))(((...)))}$$
$$x_4 \Longrightarrow \texttt{((....)).........}$$
$$x_5 \Longrightarrow \texttt{((((.....))))....}$$

A simple visual inspection allows us to see that only x_1, x_2 and x_3 have the same secondary structure as S, thus satisfying the constraint. Still, we will show the calculations of similarity. Nucleotide positions whose category (base-pairs or unpaired) do not correspond between the predicted and the target structures are highlighted in red:

$$x_1, x_2, x_3 \Longrightarrow \texttt{((....))(((...)))}$$
$$x_4 \Longrightarrow \texttt{((....)).........}$$
$$x_5 \Longrightarrow \texttt{((((.....))))....}$$

then:

$$\sigma(x_1) = \sigma(x_2) = \sigma(x_3) = \frac{17 - 0}{17} = 1$$
$$\sigma(x_4) = \frac{17 - 6}{17} \neq 1$$
$$\sigma(x_5) = \frac{17 - 12}{17} \neq 1$$

In Fig. 1, we show the target structure S and the predicted secondary structures for the RNA sequences considered (displayed by Forna Kerpedjiev et al., 2015).

Considering that the constraint is mandatory, x_4 and x_5 are discarded as candidate sequences. Thus, the objective functions are calculated only for x_1, x_2 and x_3. This is achieved by means of the ViennaRNA package for Partition Function(f_1) and Ensemble Diversity(f_2), obtaining the following values:

$$f_1(x_1) = -3.74$$
$$f_1(x_2) = -3.84$$
$$f_1(x_3) = -3.73$$
$$f_2(x_1) = 2.90$$
$$f_2(x_2) = 2.94$$
$$f_2(x_3) = 2.72$$

To evaluate Nucleotides Composition(f_3) of the candidate sequences, the percentages composition of the three categories mentioned in Section 3 (base-pairs, unpaired and total) is calculated, and the maximums % are located. S has 5 base-pairs, 7 unpaired bases and 17 total bases. For x_1:

$$\text{Base-pairs} \Longrightarrow \texttt{GC,GC,GC,CG,CG}$$
$$\text{Unpaired} \Longrightarrow \texttt{G,G,G,A,G,U,G}$$
$$\text{Total} \Longrightarrow \texttt{G,G,G,G,G,A,C,C,G,C,C,G,U,G,G,G,C}$$

from which the percentages are derived (maximums bolded):

$$\text{Base-pairs} \Longrightarrow \%\mathbf{GC} = \mathbf{100}, \%AU = 0, \%GU = 0$$
$$\text{Unpaired} \Longrightarrow \%uA = 14.29, \%uC = 0, \%\mathbf{uG} = \mathbf{71.43}, \%uU = 14.29$$
$$\text{Total} \Longrightarrow \%A = 5.88, \%C = 29.41, \%\mathbf{G} = \mathbf{58.82}, \%U = 5.88$$

and the f_3 value is:

$$f_3(x_1) = 100 + 71.43 + 58.82 = 230.25$$

Similarly, for x_2:

$$\text{Base-pairs} \Longrightarrow \texttt{GC,GC,GC,GC,GC}$$
$$\text{Unpaired} \Longrightarrow \texttt{G,A,A,A,A,A,A}$$
$$\text{Total} \Longrightarrow \texttt{G,G,G,A,A,A,C,C,G,G,G,A,A,A,C,C,C}$$

To show an alternative method to obtain f_3, instead of calculating all the percentages now we count all items in each category (maximums in bold):

$$\text{Base-pairs} \Longrightarrow \mathbf{GC/CG} = \mathbf{5}, AU/UA = 0, GU/UG = 0$$
$$\text{Unpaired} \Longrightarrow \mathbf{uA} = \mathbf{6}, uC = 0, uG = 1, uU = 0$$
$$\text{Total} \Longrightarrow \mathbf{A} = \mathbf{6}, C = 5, \mathbf{G} = \mathbf{6}, U = 0$$

from which the maximums percentages are directly derived, and the f_3 value calculated:

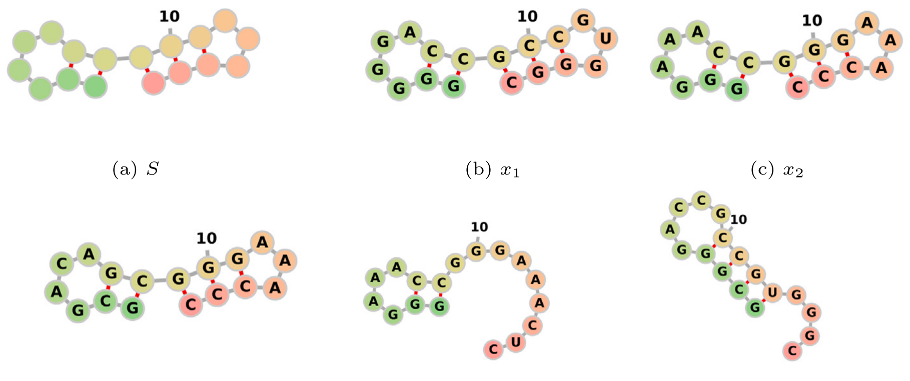
$$f_3(x_2) = 100 + 85.71 + 35.29 = 221$$

Following the same steps of either x_1 or x_2 we obtain the value of f_3 for x_3:

$$f_3(x_3) = 100 + 71.43 + 35.29 = 206.72$$

Now that the values of all the objective functions are computed, it is possible to check the dominance. The procedure implies comparing candidate sequences by pairs:

- x_1 vs. x_2: $f_2(x_1) < f_2(x_2)$, but $f_1(x_1) > f_1(x_2)$ and $f_3(x_1) > f_3(x_2)$. Consequently, neither dominates the other.



(a) S

(b) x_1

(c) x_2

(d) x_3

(e) x_4

(f) x_5

Fig. 1. Target structure S (a) and the predicted secondary structure for the RNA sequences x_1, x_2, x_3, x_4 and x_5 (b to f). Sequences x_4 and x_5 do not fit S. Nucleotides are colored according to their position in the linear primary structure. Lower numbered nucleotides are closer to the 5' end and are colored green. Nucleotides in the middle are colored yellow whereas nucleotides near the 3' end are colored red.

- x_1 vs. x_3: $f_1(x_1) < f_1(x_3)$ but $f_2(x_1) > f_2(x_3)$ and $f_3(x_1) > f_3(x_3)$, so there is no dominance.
- x_2 vs. x_3: $f_1(x_2) < f_1(x_3)$, but $f_2(x_2) > f_2(x_3)$ and $f_3(x_2) > f_3(x_3)$. Once again neither one dominates the other.

From these results it can be concluded that x_1, x_2 and x_3 is a set of non-dominated solutions.

There are many performance indicators to evaluate the quality of a given *Pareto set* in the Multiobjective Optimization domain. In this work, we employ the following indicators:

- Hypervolume (HV) (Zitzler and Thiele, 1999) (*S-metric*), determines the volume (in objective function space) covered by the solutions of a non-dominated set $A = \{a_1, \dots, a_n\}$ with respect to a predefined bounding reference point $r = (r_1, r_2, r_3)$. The hypercube h of each member a_i of set A is calculated as $h(a_i) = [a_{i1}, r_1] \times [a_{i2}, r_2] \times [a_{i3}, r_3]$. Therefore, the hypervolume HV of A is the union of these $|A|$ hypercubes. As a consequence, overlapping hypercubes are only counted once. To calculate this, Eq. (5) can be used:

$$HV(A, r) = L\left(\bigcup_{i=1}^{|A|} h(a_i) | a_i \in A\right) \tag{5}$$

  being L the Lebesgue measure.
- Inverted Generational Distance (IGD) (Coello and Sierra, 2004). It was introduced as an advancement over the Generational Distance (GD) metric by flipping the order of the fronts used as input. GD is defined as the distance between each objective vector in a given approximation front A (solution points) and the closest objective vector in a reference front R, which may be the true Pareto front or a highly accurate estimation of it, averaged over the size of A. Hence, $IGD(A, R) = GD(R, A)$. This means that IGD is essentially the same as GD, but with the distances between each objective vector at the reference front and its closest counterpart at the approximation front averaged over the size of the reference front.

4.2. Chromosome representation

The selection of the chromosome encoding of individuals has significant impact on the implementation of the algorithm, as it determines how the RNA inverse folding problem is structured within the algorithm, as well as the behavior of the evolutionary algorithm, which limits the operators that can be applied to the chromosome. For this study, we use the same encoding as in Rubio-Largo et al. (2019), since it permits to apply any crossover and mutation operators for continuous optimization problems.

The representation of the chromosome X was designed as a real-valued vector of length $|B| + |U|$, being B and U the sets of positions in the target structure S of the base-pairs or unpaired bases respectively.

$$X = \{\rho_1, \dots, \rho_{|B|}, \rho_{|B|+1}, \dots, \rho_{|B|+|U|}\}$$

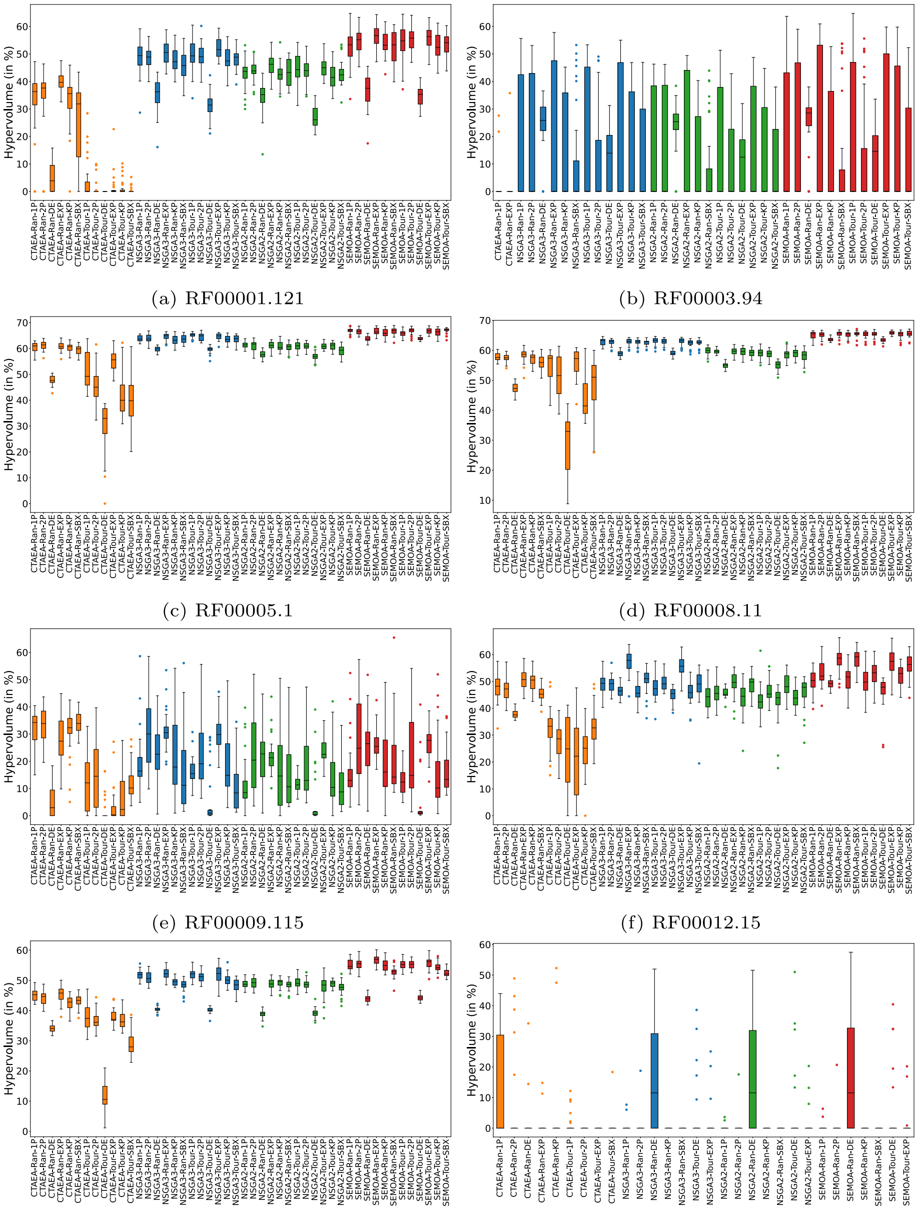
where ρ is a real value in the range [0,1] that codifies the type of base-pair or unpaired nucleotide. The $|B|$ first elements store the ρ values of the base-pairs, whereas from $|B|+1$ to $|B|+|U|$ are ρ values of unpaired positions.

To obtain B and U, the target structure S in dot-bracket notation is processed as follows:

1. S is traversed iteratively from the first to the last position.
2. Whether the considered position is unpaired (dot "”.), it is stored in U.
3. On the contrary, if it is the first position of a base-pair (opening bracket “(”) it is saved. The walk continues until the second position of the base-pair (closing bracket “)”) is found, when they are both stored in B.
4. The positions visited while searching for the corresponding closing bracket are processed the same way, that is, storing it in U if is an unpaired position, or saving it until its matching closing bracket is found (when they both will be stored in B) if is an opening bracket.

With this chromosome encoding, it is mandatory to have a procedure to translate a real-valued input chromosome (X) into an RNA sequence (x) The translated RNA sequence is the element that will actually be evaluated to verify if it fulfills the similarity constraint and to calculate its objective functions. In Rubio-Largo et al. (2019), we defined the following procedure:

1. X is traversed iteratively from the first to the last element.
2. The $|B|$ first elements correspond to base-pairs. Therefore, for the element being considered, from B the positions in S (and consequently in the RNA sequence x being constructed) of both components of the base-pair are obtained.
3. The ρ value is translated as a base-pair:
   - GC if $0 \le \rho < 1/6$
   - CG if $1/6 \le \rho < 2/6$
   - AU if $2/6 \le \rho < 3/6$
   - UA if $3/6 \le \rho < 4/6$
   - GU if $4/6 \le \rho < 5/6$
   - UG if $5/6 \le \rho \le 1$
4. The base-pair nucleotides are placed in the positions obtained in step 3.



(a) RF00001.121

(b) RF00003.94

(c) RF00005.1

(d) RF00008.11

(e) RF00009.115

(f) RF00012.15

(g) RF00026.1

(h) RF00028.1

Fig. 2. Boxplots of final HV values obtained with all alg+oper combinations for a selection of RFAM structures. Note that the color code used refers to the different algorithms used, specifically: 🟧 **C-TAEA**, 🟦 **NSGA-III**, 🟩 **NSGA-II**, 🟥 **SMS-EMOA**.

5. The remaining $|U|$ elements correspond to unpaired nucleotides. Therefore, for the element being considered, from U the position in S (and consequently in the RNA sequence x being constructed) is obtained.
6. The ρ value is translated as a nucleotide:
   - C if $0 \leq \rho < 1/4$
   - G if $1/4 \leq \rho < 2/4$
   - A if $2/4 \leq \rho < 3/4$
   - U if $3/4 \leq \rho \leq 1$
7. The obtained nucleotide is placed in the position found in step 5.

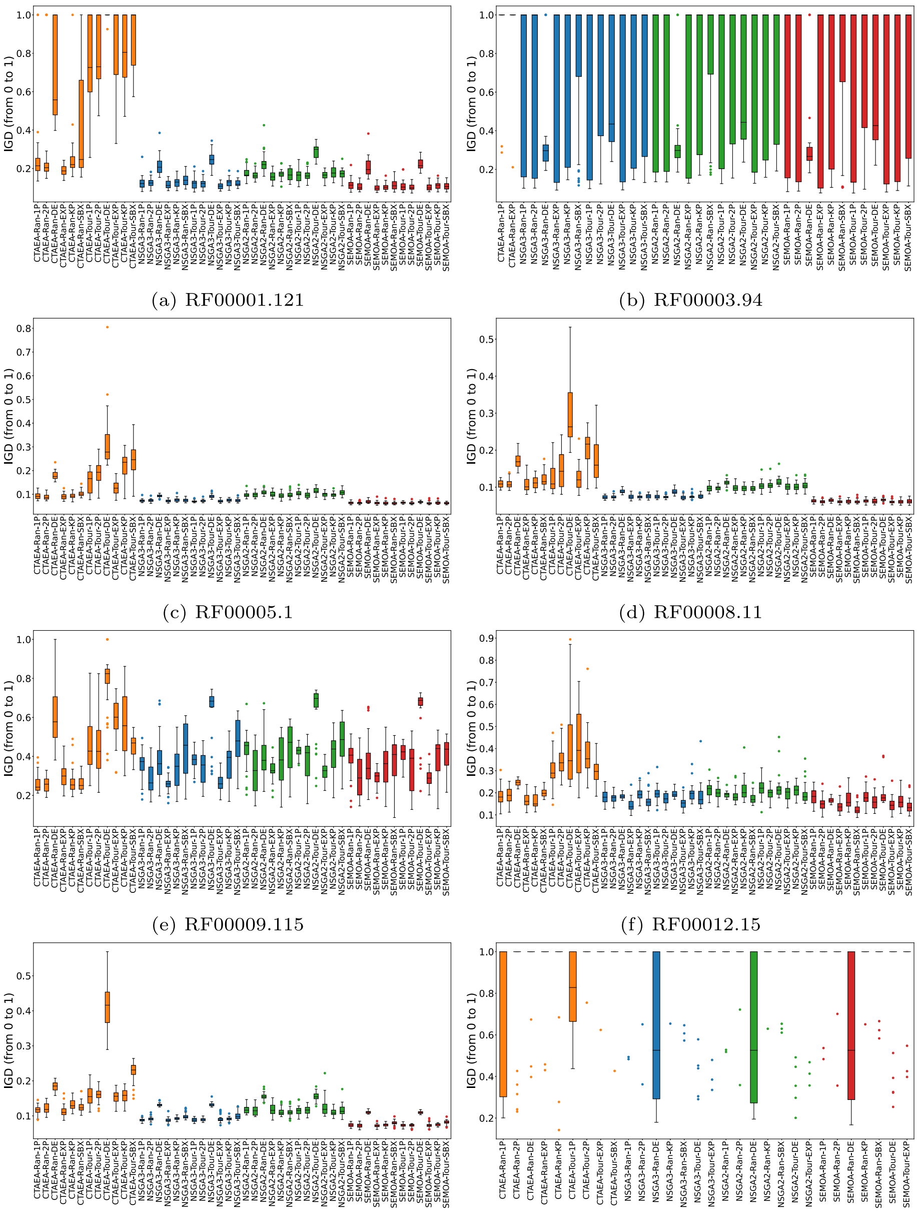
Initialization of the individuals in the starting population is conducted in a manner that minimizes the probability of creating unnecessary loops in the structure of the new RNA sequence by taking into account possible related base-pairs when selecting the nucleotide for each unpaired position (U set). In a small percentage of individuals, this initialization is performed completely randomly.

Using this chromosome encoding and initialization of individuals ensures a certain level of greediness, which allows finding sequences that fold into the target structure faster.

For a better understanding of the proposed representation, we show it here applied to our previous example structure S. To start, positions are numbered and B and U defined.

$$S = ((....))(((...)))$$
$$= (_1(_2\cdot_3\cdot_4\cdot_5\cdot_6)_7)_8(_9(_{10}(_{11}\cdot_{12}\cdot_{13}\cdot_{14})_{15})_{16})_{17}$$
$$B = \{(1,8),(2,7),(9,17),(10,16),(11,15)\}$$
$$U = \{3,4,5,6,12,13,14\}$$

Therefore, in this case the representation of a chromosome will be a vector of twelve elements ($|B| = 5 + |U| = 7$), where the elements 1 to 5 encode the paired nucleotides that will be located in the positions



(a) RF00001.121

(b) RF00003.94

(c) RF00005.1

(d) RF00008.11

(e) RF00009.115

(f) RF00012.15

(g) RF00026.1

(h) RF00028.1

Fig. 3. Boxplots of final IGD values obtained with all alg+oper combinations for a selection of RFAM structures. Note that the color code used refers to the different algorithms used, specifically: 🟧 **C-TAEA**, 🟦 **NSGA-III**, 🟩 **NSGA-II**, 🟥 **SMS-EMOA**.

stored in the B set, and the elements 6 to 12 encode the unpaired nucleotides that will be located in the positions stored in the U set. As already stated, their values are real numbers in the range [0,1]. Let us suppose that we have the following chromosomes X_1, X_2 and X_3:

$$X_1 = \{0.15, 0.10, 0.05, 0.21, 0.32, 0.27, 0.42, 0.48, 0.60, 0.35, 0.91, 0.29\}$$

$$X_2 = \{0.02, 0.15, 0.08, 0.02, 0.13, 0.47, 0.51, 0.73, 0.62, 0.51, 0.67, 0.69\}$$

$$X_3 = \{0.01, 0.18, 0.16, 0.02, 0.11, 0.30, 0.55, 0.22, 0.71, 0.69, 0.62, 0.74\}$$

We decode them into RNA sequences using the translation process explained above. For example, considering the first element of X_1 (0.15), by its position we know that it corresponds to a base-pair. Looking for this value in the translation process, as $0 \leq 0.15 < 1/6$, this base pair is going to be GC. The position of these nucleotides in x_1 is obtained by looking at the first position of B, thus obtaining (1,8). So G will be placed at position 1 and C at position 8 of x_1. Similarly, the ninth item in X_1 (0.60) is an unpaired nucleotide, whose position corresponds to the fourth element in U (6). Checking the values displayed in the translation process, as $2/4 \leq 0.60 < 3/4$, this nucleotide corresponds to an A, to be placed at position 6 of x_1.

By applying the process to whole chromosomes, we obtain the sequences x_1, x_2 and x_3 considered in Section 4.1:

$$x_1 = G_1G_2G_3G_4G_5A_6C_7C_8G_9C_{10}C_{11}G_{12}U_{13}G_{14}G_{15}G_{16}C_{17}$$

$$x_2 = G_1G_2G_3A_4A_5A_6C_7C_8G_9G_{10}G_{11}A_{12}A_{13}A_{14}C_{15}C_{16}C_{17}$$

$$x_3 = G_1C_2G_3A_4C_5A_6G_7C_8G_9G_{10}G_{11}A_{12}A_{13}A_{14}C_{15}C_{16}C_{17}$$

4.3. Description of the multiobjective metaheuristics

The aim of this section is to briefly describe the different multiobjective metaheuristics used in this work to solve the RNA inverse folding problem: NSGA-II (used as the core algorithm in m2dRNAs Rubio-Largo et al., 2019), SMS-EMOA, NSGA-III, and C-TAEA.

NSGA-II (Deb et al., 2002) is a well-known method that aims to generate a new offspring population from an existing parent population.

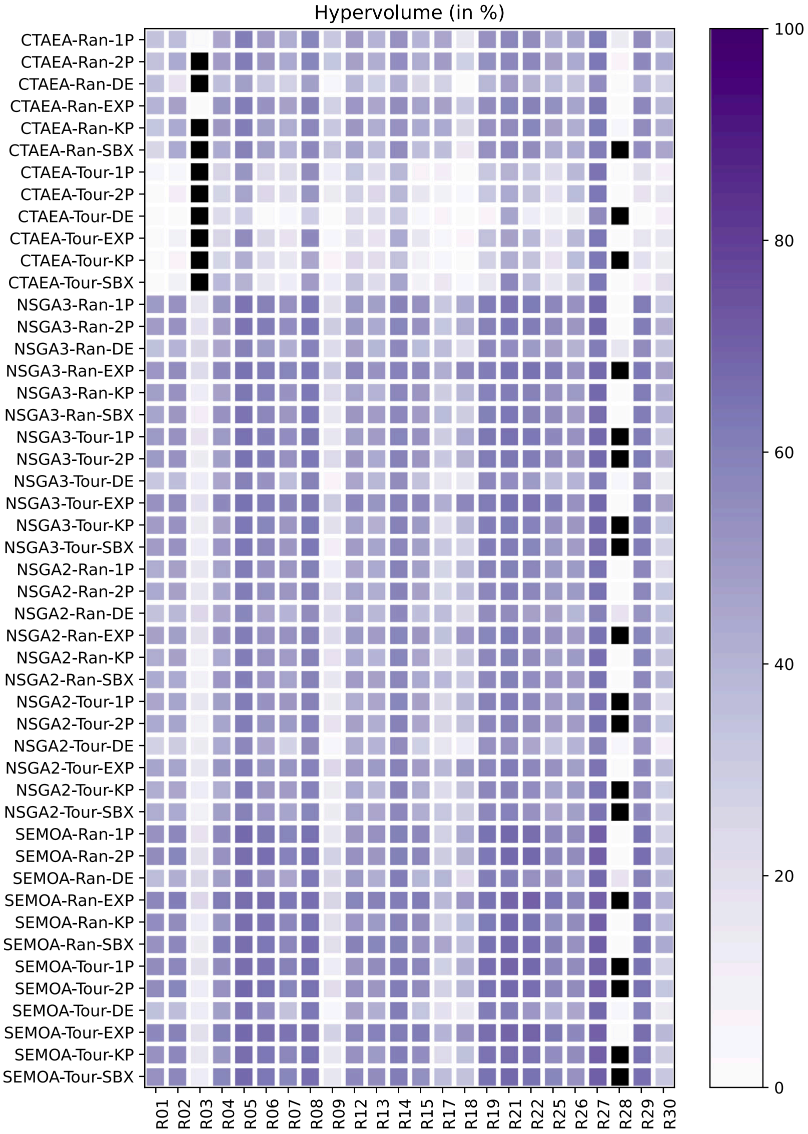


Fig. 4. Heat map of average final HV values of RFAM structures. NA values are represented as ■.

For this purpose, traditional genetic operations of selection, crossing and mutation are used. This offspring population is combined with the parent population to create a new population that is then sorted into categories (ranked fronts) based on the dominance relationships established between individuals, calculated by applying a non-dominated sorting function. Individuals classified as the best first half will become the parent population of the next generation. If the individuals of a front are to be only partially selected, their crowding distances are computed to determine the best ones. This strategy of comparing crowding distances (utilized in both tournament selection and population reduction stages) allows NSGA-II to incorporate diversity among the non-dominant solutions.

SMS-EMOA (*S-Metric Selection* Evolutionary Multi-Objective Algorithm) (Beume et al., 2007) employs a steady-state algorithm characterized by the utilization of non-dominated sorting as a ranking criterion, and the application of HV (*S-Metric*) to select the individual to eliminate, since it will be the one with the lowest HV contribution to the lowest-ranked front. Fleischer (2003) demonstrated that within a finite search space and with a specified reference point, maximizing the HV is tantamount to discovering the Pareto set, so the main idea here is to choose potential solutions based on their impact on the dominated HV. The SMS-EMOA algorithm was designed to encompass a maximum HV using a finite set of points and to mitigate the challenge of selecting the appropriate reference point for the HV calculation.

From an initial population comprising μ individuals, a novel individual is created through randomized crossover and mutation operators. This fresh individual will join the subsequent population solely if the substitution of another individual for it results in an improved population quality in terms of HV. To this end, non-dominated sorting is applied to the $\mu + 1$ individuals, and the one with the lowest contribution to HV in the worst ranked front is eliminated. Thus, individuals that maximize the *S-metric* value within the population are retained, ensuring that the population's encompassed HV does not decrease as the generations advance.

Since calculating HV is computationally expensive, a steady-state selection scheme is used. Since only one individual will be removed from the population in each generation, the selection operator will calculate a maximum of $\mu + 1$ HV values.

The reference point is dynamically updated during the optimization process. In each generation, it is recalculated as a vector of the current worst-case target values, incremented by 1.0.

The basic framework of NSGA-III (Deb and Jain, 2014) is similar to the original NSGA-II algorithm. To achieve diversity among the individuals in the population, NSGA-III uses a predefined set

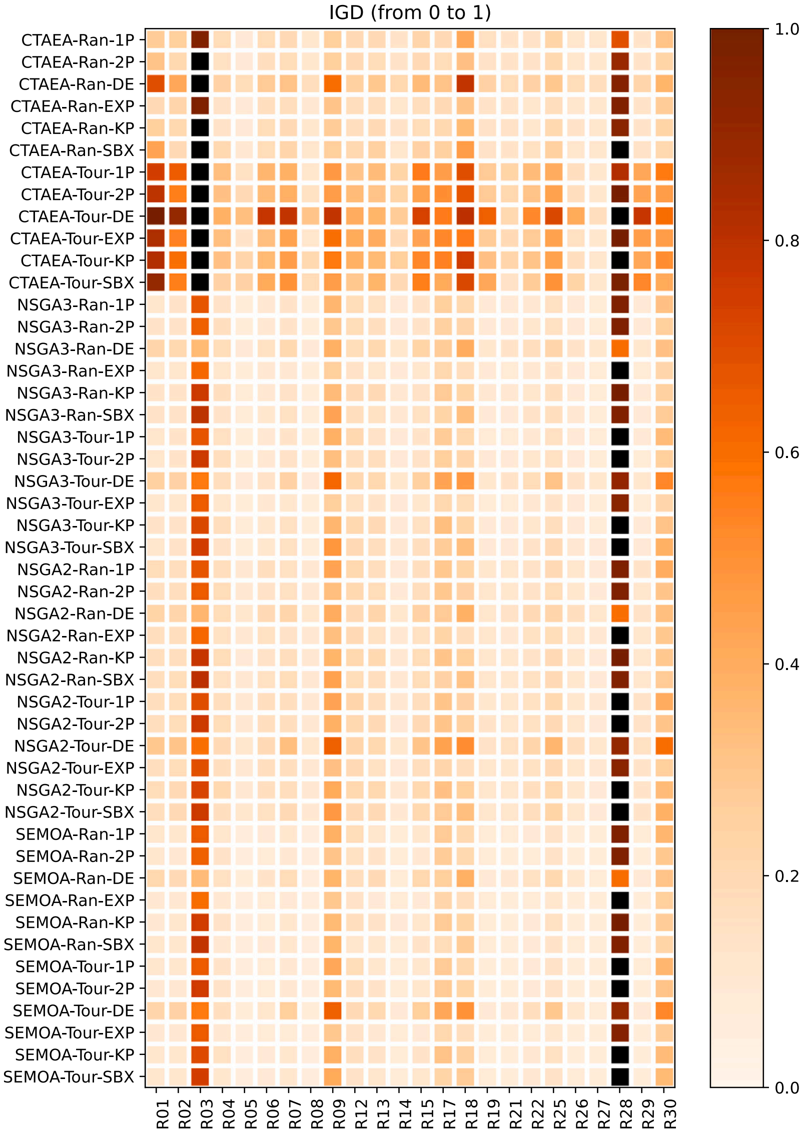


Fig. 5. Heat map of average final IGD values of RFAM structures. NA values are represented as ■.

of well-distributed reference points that are placed on a normalized hyper-plane. The selected reference points may either come in a pre-established structured format or be provided by the user. A commonly used method to place reference points is Das and Dennis's systematic approach (Das and Dennis, 1998). This method settles points on a normalized hyper-plane that exhibits equal inclination towards all objective axes and intersects each axis at a value of one. The total number of reference points is determined by the number of objectives and the number of divisions considered for each objective.

NSGA-III places a strong emphasis not only on non-dominated solutions but also on population members linked to each reference point. As the reference points are broadly distributed throughout the entire normalized hyper-plane, the resulting solutions are also expected to be widely distributed along or in close proximity to the Pareto-optimal front.

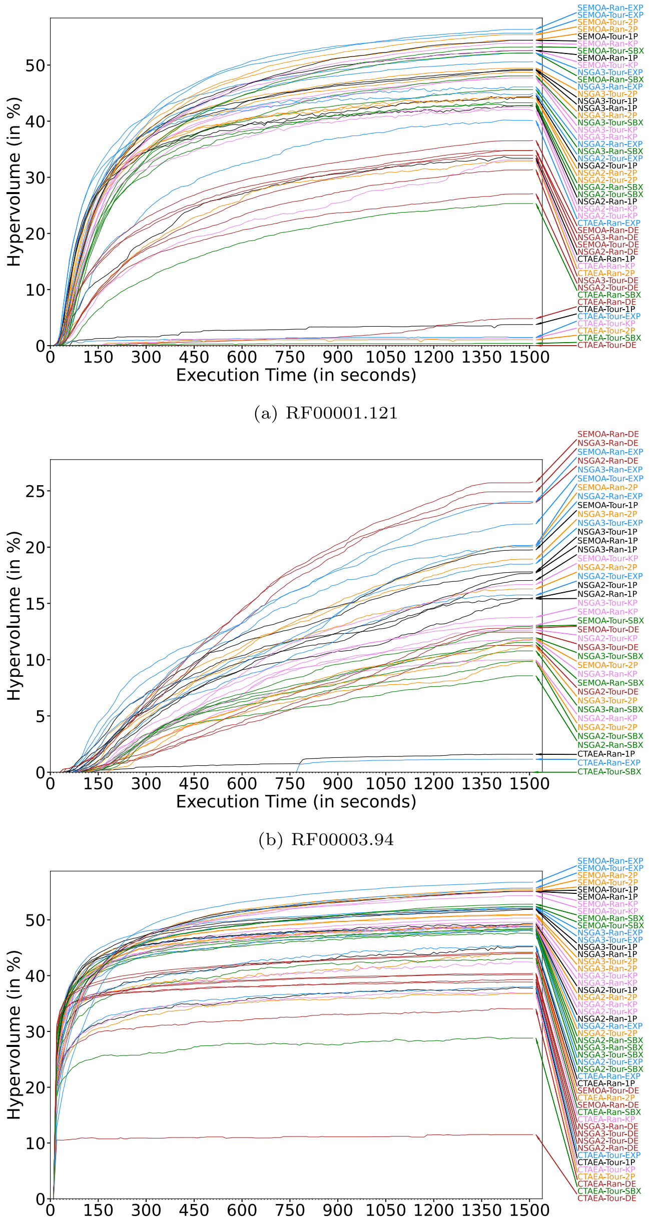
The objective functions of the population members are adaptively normalized. The method involves first determining the ideal point of the population and then identifying the extreme points on each objective axis to construct a hyper-plane. This process is performed in each generation, employing the extreme points identified from the start of the simulation.

In order to link each population member with a reference point, a reference line is established for each reference point by connecting it to the origin. Subsequently, the perpendicular distance between each population member and each of these reference lines is computed. A population member is then associated with the reference point whose reference line exhibits the closest proximity to the population member within the normalized objective space.

To cover all available population positions of the next generation, the underrepresented reference directions are prioritized, assigning solutions to them first. If no solutions are allocated to a reference direction, then the surviving solution is the one with the smallest perpendicular distance in the normalized objective space. Whether a second solution is added to this reference line, it is randomly assigned.

C-TAEA (Li et al., 2019) is a constraint handling method, without parameters, for constrained multiobjective optimization. It simultaneously upholds two cooperative and complementary archives, named convergence-oriented archive (CA) and diversity-oriented archive (DA). The primary attributes of C-TAEA can be described as follows:

1. CA plays a fundamental role in sustaining both the convergence and feasibility aspects of the evolutionary process, exerting selection pressure towards the Pareto Front.
2. In contrast, DA mainly focuses on maintaining the convergence and diversity of the evolution process, without considering feasibility. The DA explores areas that have not been exploited by

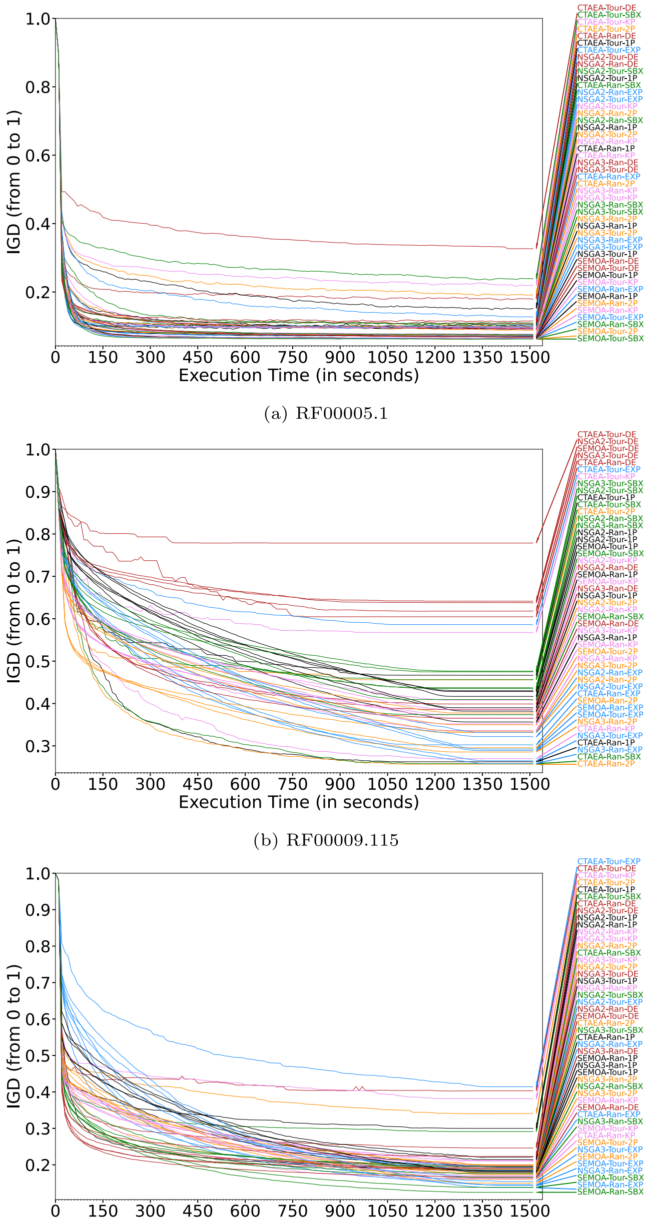


(a) RF00001.121

(b) RF00003.94

(c) RF00026.1

Fig. 6. Convergence (HV) plots of all alg+oper combinations for a selection of RFAM structures. Note that the color code used refers to the different crossover operators used, specifically: ⬛ **1P**, 🟧 **2P**, 🟥 **DE**, 🟦 **EXP**, 🟪 **KP**, and 🟩 **SBX**.



(a) RF00005.1

(b) RF00009.115

(c) RF00012.15

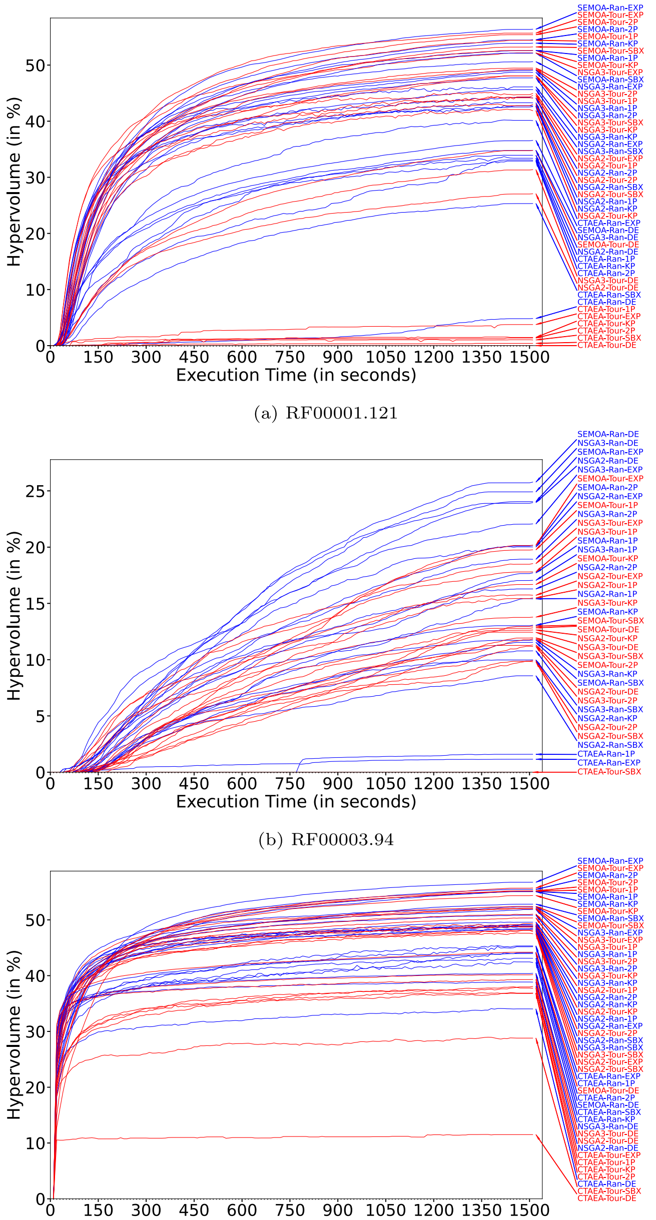
Fig. 7. Convergence (IGD) plots of all alg+oper combinations for a selection of RFAM structures. Note that the color code used refers to the different crossover operators used, specifically: ⬛ **1P**, 🟧 **2P**, 🟥 **DE**, 🟦 **EXP**, 🟪 **KP**, and 🟩 **SBX**.

the CA. This improves the diversity of the CA population within the currently investigated feasible region and helps overcome local optima or locally feasible regions, even exploring areas considered unfeasible by CA.

3. To take advantage on the complementary effect and valuable data contained in these two collaborative archives, a restricted mating selection mechanism is implemented. This mechanism independently selects suitable parent pairs from the CA and DA, taking into account their individual evolutionary statuses.

To facilitate density estimation, the objective space is divided into subregions, each of which is represented by a unique weight vector on the canonical simplex. Then each solution of a population is associated with a unique subregion.

Both CA and DA populations are updated by combining them with the offspring population (becoming H_c and H_d respectively). H_c prioritizes feasible solutions, and makes use of fast non-dominated sorting, density information of the subregions and the distance between each solution and its nearest neighbor to pick the individuals to construct the new CA. If feasible solutions in (H_c) do not fill the new CA, the number of needed infeasible solutions are selected by means of fast non-dominated sorting and Constraint Violation (CV). On the other hand, the update mechanism of H_d possesses two distinctive traits: It disregards CV, and employs the most current CA as a reference set, thus complementing the CA's performance by investigating its less-explored regions. This is achieved by taking into account the density estimation of each subregion to decide whether or not a solution is going to survive.

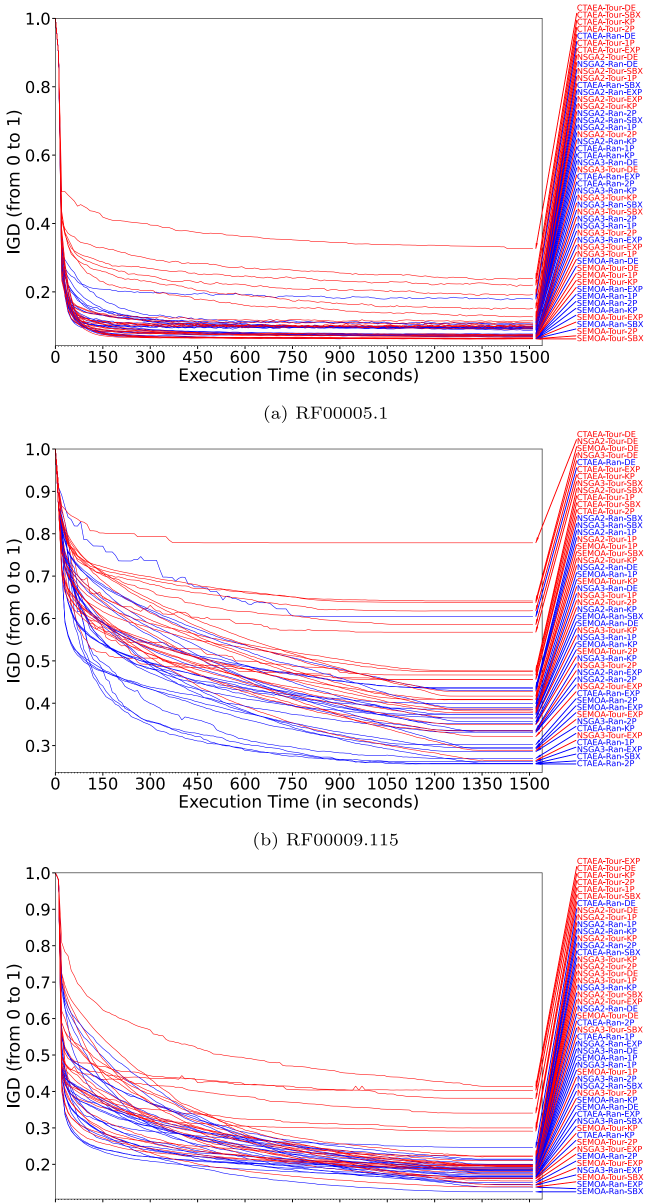


(a) RF00001.121

(b) RF00003.94

(c) RF00026.1

Fig. 8. Convergence (HV) plots of all alg+oper combinations for a selection of RFAM structures. Note that the color code used refers to the different selection operators used, specifically: 🟦 **R**, 🟥 **T.**



(a) RF00005.1

(b) RF00009.115

(c) RF00012.15

Fig. 9. Convergence (IGD) plots of all alg+oper combinations for a selection of RFAM structures. Note that the color code used refers to the different selection operators used, specifically: 🟦 **R**, 🟥 **T.**

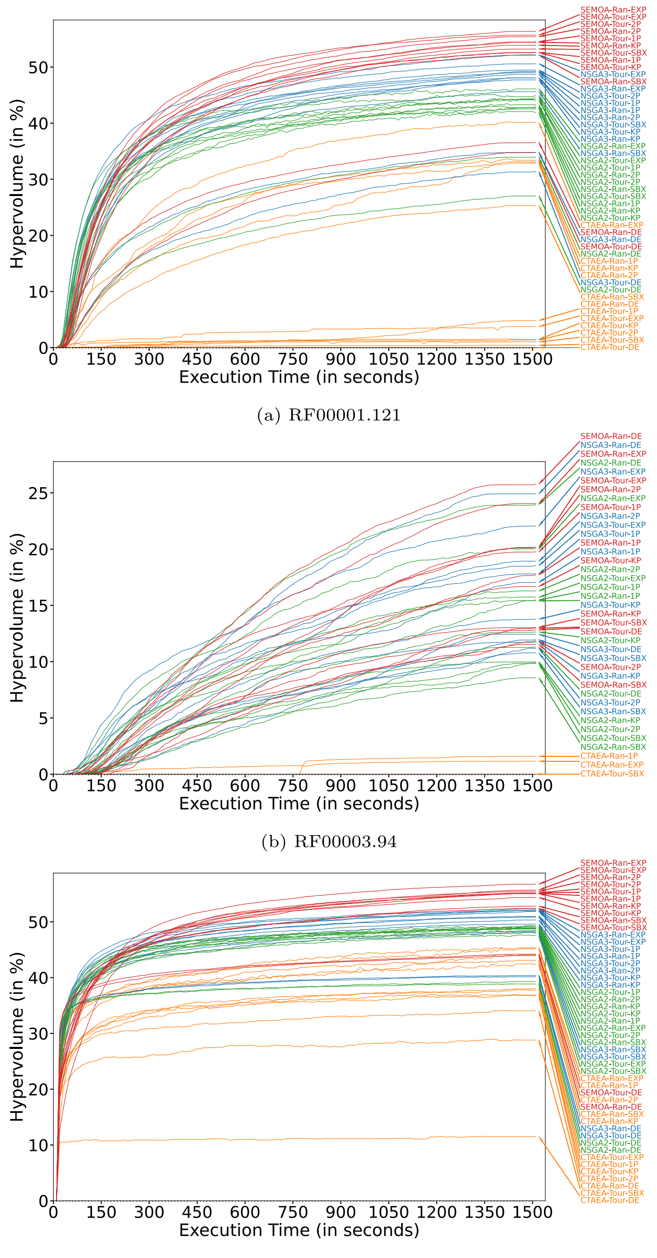
NSGA-II, NSGA-III, and SMS-EMOA have in common the use of non-dominated sorting to categorize individuals into ranked fronts after merging the parent and offspring populations, but NSGA-II and NSGA-III create an offspring population of the same size as the parent population, while SMS-EMOA generates only one individual as offspring. These three metaheuristics differ in the criteria used to introduce diversity among members of the same rank and choose which individuals will be discarded to form the next generation, as NSGA-II is based on crowding distance, NSGA-III uses a predefined set of well-distributed reference points to which population members are associated, and SMS-EMOA relies on HV. Similar to NSGA-III, C-TAEA makes use of reference points/vectors to divide the objective space into subregions, to which individuals are associated. C-TAEA is also the most different of all, since it is based on two archives (CA and DA) that collaborate with each other and makes use of a restricted mating selection mechanism. In this last metaheuristic, diversity is the responsibility of the DA, and non-dominated sorting is also involved in the algorithm, due to its role in the construction of the next generation CA.

4.4. *Description of the operators*

In this section we take a look at the versions of the classical genetic operators selection, crossover and mutation applied to the selected metaheuristics.

4.4.1. *Selection*

Selection operators determine which individuals from a population should be chosen to become parents for the next generation. The



(a) RF00001.121

(b) RF00003.94

(c) RF00026.1

Fig. 10. Convergence (HV) plots of all alg+oper combinations for a selection of RFAM structures. Note that the color code used refers to the different algorithms used, specifically: 🟧 **C-TAEA**, 🟦 **NSGA-III**, 🟩 **NSGA-II**, 🟥 **SMS-EMOA**.

selection operator is responsible for biasing the selection process towards fitter individuals, in order to improve the overall fitness of the population over time.

- Random (RandS). Individuals are randomly selected from the current population to be used as parents. The implementation utilized here makes use of a permutation to prevent any individual from being chosen more than once.
- Tournament (used by m2dRNAs Rubio-Largo et al., 2019). Tournament pressure is helpful for faster convergence. The process of tournament selection starts with the selection of a few individuals at random from the population, followed by the running of several "tournaments" among them. The tournament winner will be the one with the best fitness (the specific fitness function varies among metaheuristics), which will then be selected for crossover.

  By altering the tournament size, the selection pressure can be adjusted. A larger tournament size reduces the chances of weaker individuals being selected as parents, since stronger individuals are more likely to be included in the same tournament. In this study a binary tournament (two individuals) is always used when this operator is selected.

4.4.2. *Crossover*

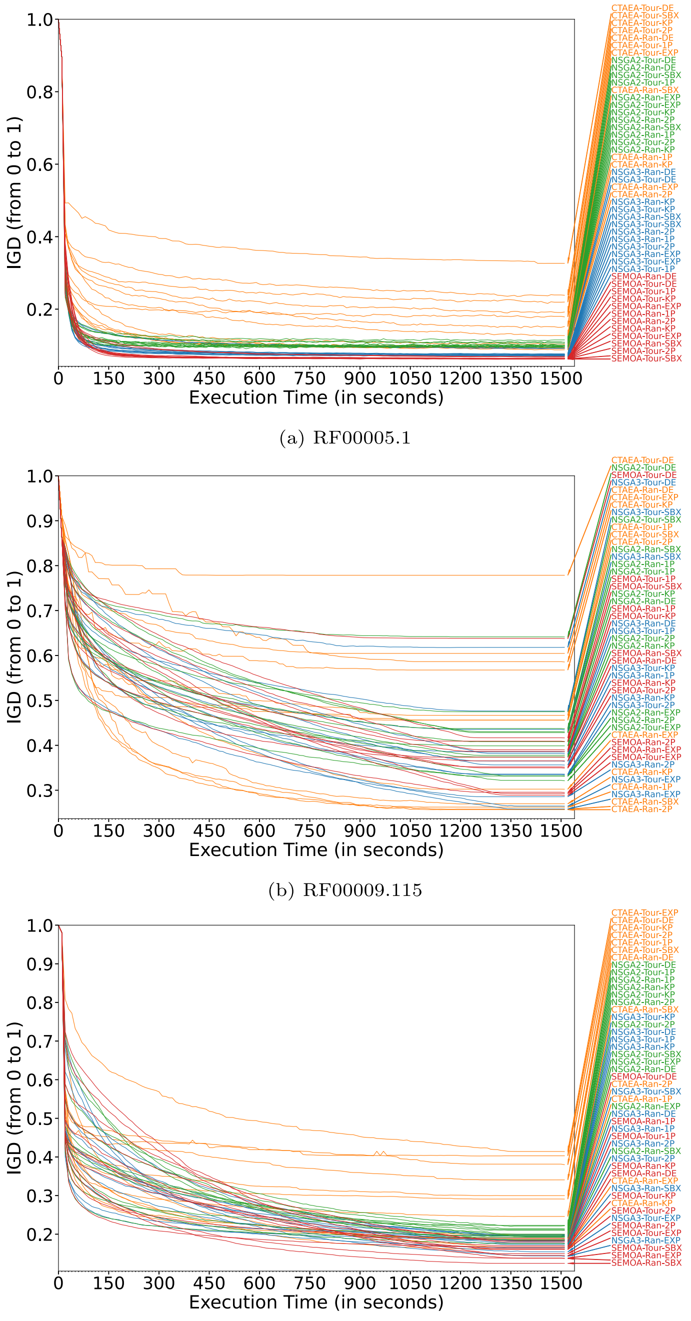
The crossover operator takes two or more parent individuals and selects one or more points in their chromosomes. The genetic material between these points is then exchanged between the parents, resulting in new offspring that have a combination of genetic information from both parents.

- Simulated Binary (SBX) (used by m2dRNAs Rubio-Largo et al., 2019). This operator simulates the functioning of the single-point crossover operator on binary strings but for real-valued chromosomes, achieving a similar search power. It makes use of a probability distribution to generate new values for each gene in the offspring solutions based on the values of the corresponding genes in the two parent individuals. This probability distribution ensures that the new values are similar to the values of the parents, but with some variation to allow for potential improvements. SBX requires a user-selected parameter called distribution index (η_c), which is a non-negative real number. A large value of η_c gives a higher probability for creating offspring close to the parents, whereas a small value of η_c allows creating distant solutions (more diverse search).
- Differential Evolution (DE). It is the crossover operator used in Differential Evolution (Storn and Price, 1997) algorithm. In this operation, genes either from the target chromosome or a previously computed donor chromosome are selected based on some probability distribution to form a trial (child) chromosome, ensuring that the trial chromosome receives at least one gene from the donor chromosome. It is used a *Crossover Probability* parameter. The donor is created from randomly selected parental solutions from the population, which are combined to generate a new candidate solution (donor) by adding a scaled difference vector to one of the parents.
- One-Point (1Point). In single-point crossover, a single point is randomly selected in the genetic material of each of the two parents, and the genetic information is exchanged beyond that point.
- Two-Point (2Point). In two-point crossover, two points are randomly selected, and the genetic material between those points is exchanged.
- K-Point (KPoint). This method is similar to the two previous, but with a fixed and user-defined number of points (K, greater than two).
- Exponential (Exp). To begin, a starting index is chosen randomly, followed by the inclusion of the subsequent gene to be mutated with a predetermined probability (p_{exp}). In the event that the final gene is reached, the process wraps around to the first variable.

4.4.3. *Mutation*

This operator introduces random changes in the chromosomes of individuals in a population. The purpose of the mutation operator is to maintain genetic diversity in the population and avoid premature convergence towards suboptimal solutions. Mutation is applied with a low probability rate to avoid drastic changes that may lead to poor solutions.

- Polynomial (PM) (used by m2dRNAs Rubio-Largo et al., 2019). A polynomial probability distribution is used to perturb a solution in its neighborhood. This mutation operator usually follows the common method of attempting to mutate each gene on an individual's chromosome one at a time with a predefined mutation



(a) RF00005.1

(b) RF00009.115

(c) RF00012.15

Fig. 11. Convergence (IGD) plots of all alg+oper combinations for a selection of RFAM structures. Note that the color code used refers to the different algorithms used, specifically: 🟧 **C-TAEA**, 🟦 **NSGA-III**, 🟩 **NSGA-II**, 🟥 **SMS-EMOA**.

probability p_m. A random number $u \in [0,1]$ is created for each gene in the individual. Whether $u \leq p_m$ the gene is selected to be mutated, so a value calculated by a polynomial function is added to its current value. The polynomial function determines the distribution of the random values to be added to the gene, with higher order polynomials leading to more extreme values.

5. Experimental results

The objective of this section is to present a comparative analysis of the performance of four selected MOEAs and various parameter combinations in addressing the RNA inverse folding problem.

5.1. Previous considerations

Here we describe the methodology followed in the experimental analysis of the MOEAs. We will compare the behavior of the four multiobjective metaheuristics explained in Section 4.3 (NSGA-II, SMS-EMOA, NSGA-III, and C-TAEA) when used to solve the RNA inverse folding problem with the RFAM benchmark set (Taneda, 2010), which was created from the Rfam 9.0.21 seed alignments and contains a total of 29 target structures. In addition, these MOEAs have been tested with the different combinations of the appropriate genetic operators for real-valued chromosomes selected for this study. Specifically it is used:

- Selection. RandS is applied to all the metaheuristics. Tournament is tested with NSGA-II, SMS-EMOA, NSGA-III, and for the case of C-TAEA it is used in the context of its restricted mating selection mechanism.
- Crossover. All selected crossover operators (SBX, DE, 1Point, 2Point, KPoint and Exp) were tested with each of the metaheuristics and with the parameter $p_c = 0.9$ (probability of a crossover). Other parameters are:
  - SBX: $\eta_c = 10$
  - DE: Crossover constant, $CR = 0.7$
  - KPoint: $K = 4$
  - Exp: $p_{exp} = 0.9$
- Mutation. Only PM is selected. Its parameters are $\eta_m = 5$ (distribution index) and $p_m = 1/l$ (where l is the length of the chromosome).

For the implementation we have used the framework pymoo: Multi-objective Optimization in Python (v0.6.1) and ViennaRNA (v2.5.1). The experiments were carried out on a 2x Intel(R) Xeon(R) Gold 6238 CPU @ 2.10 GHz and 196 GB of RAM with Ubuntu 18.04 and Python 3.7.5, running each combination 31 independent times, with population size of 52 individuals, a time limit of 25 min and Turner2004 energy parameters. From each execution, final and intermediate results (every 10 s.) were obtained.

To encode the input structure, obtain the length of the chromosome, as well as the indices corresponding to the paired and unpaired positions, we use the process to obtain B and U described in Section 4.2. Following the pymoo interface and meeting its requirements for the definition of the chromosome, number of objective functions, and equality constraint (similarity), the problem is defined as an *ElementwiseProblem*, within which the evaluation is performed. For this, the chromosome must be translated from a vector of real values to an RNA sequence, using the translation process also explained in Section 4.2, implemented as an isolated function. To evaluate the sequence, the objective functions *f1* (Partition Function), *f2* (Ensemble Diversity), and the similarity are calculated with the help of the ViennaRNA Package.

In Pymoo, functions subject to equality constraints must be defined in such a way that, to satisfy the constraint, the function value must be 0. Therefore, instead of similarity, dissimilarity is used in the implementation, since a dissimilarity of 0 indicates that the structures are identical—just as a similarity value of 1 would. To calculate this dissimilarity, the structure chains are compared and the non-matching positions are counted. Thus, if the structures are the same, the value will be 0.

For the C-TAEA and NSGA-III algorithms, the reference directions are created following the structured approach of Das and Dennis with n_partitions=8 (number of partitions). For all algorithms, if the selection operator used is RandS, this is indicated to the algorithm in its call, while if it is Tournament no specific parameter is needed, since Tournament is the default selection operator for all. The crossover and mutation operators are explicitly stated, along with the parameters indicated above, also in the algorithm call. Duplicate individuals in the population are not eliminated. A *Callback* object is used to obtain the intermediate results. The run number (from 1 to 31) is used as a seed for the call to minimize. Any other parameter not specified here uses the default value of the pymoo version specified above.

To compare the results of each combination of algorithm + operators (alg+oper) IGD and HV performance indicators are used (0–1 normalized). Since a reference Pareto front is needed to calculate IGD, an approximation was constructed for each RFAM target structure by extracting its non-dominated solutions from all corresponding final solution sets.

Furthermore, an objective ranking was computed as follows. For each RFAM structure, the calculated average final HV and IGD values were ordered from best to worst. For both indicators, the first to the sixth combinations obtained the following scores: 10, 8, 6, 4, 2 and 1 points. The seventh and subsequent combinations did not score any points. The scores corresponding to all the Rfam structures were added together to obtain two overall scores, one for HV and the other for IGD. Finally, both scores were summed to obtain a total score.

Table 1
Reference table to RFAM structures acronyms.

RFAM dataset	Acronym
RF00001.121	R01
RF00002.2	R02
RF00003.94	R03
RF00004.126	R04
RF00005.1	R05
RF00006.1	R06
RF00007.20	R07
RF00008.11	R08
RF00009.115	R09
RF00012.15	R12
RF00013.139	R13
RF00014.2	R14
RF00015.101	R15
RF00017.90	R17
RF00018.2	R18
RF00019.115	R19
RF00021.10	R21
RF00022.1	R22
RF00025.12	R25
RF00026.1	R26
RF00027.7	R27
RF00028.1	R28
RF00029.107	R29
RF00030.30	R30

Throughout the text, alg+oper combinations are named according to the algorithm and operators used, matching the following structure: Algorithm_selection _mutation_crossover. For figures and tables the corresponding acronyms are used, which can be found in Tables 1 and 2, thus becoming generically alg-sel-mut-cross.

5.2. RFAM convergence

Five RFAM structures (RF00010.253, RF00011.18, RF00016.15, RF00020.107, RF00024.16) were not solved by any alg+oper combination. Therefore, they were excluded from further analysis, meaning that there were 24 RFAM structures left to work with.

Our previously published multiobjective metaheuristic to design RNA sequences (m2dRNAs) corresponds to the alg+oper combination NSGA-II_ Tournament_PM_SBX, as can be checked in Rubio-Largo et al. (2019). Therefore, throughout this discussion we will look at that combination to test its performance.

As a first approach to analyzing the behavior and performance of the algorithm+operator (alg+oper) combinations, Figs. 2 and 3 show boxplots of the final HV and IGD values obtained by each combination, allowing their distribution to be easily examined. Each boxplot represents the results for one RFAM structure. Only a selection is shown here; the complete set is available in the supplementary material. Values of 0 and 1 were assigned to the HV and IGD indicators, respectively, for repetitions that did not return results. If none of the repetitions for a given combination returned results, the corresponding boxplot is not displayed. To aid interpretation, the boxplots are color-coded by algorithm. Additionally, the means of the final and intermediate HV and IGD values from the 31 repetitions were computed for each alg+oper combination across all RFAM structures, using the same criteria for assigning values in cases with no results. This ensured that averages were always calculated over 31 values. Figs. 4 and 5 present heat maps based on the final mean values of HV and IGD.

At first glance some observations can be made. Along the majority of the graphs it can be seen that C-TAEA usually is the worst algorithm,

Table 2
Algorithms and operators acronyms reference tables.

(a) Reference table to crossover operators acronyms

Crossover operator	Acronym
One Point (1Point)	1P
Two Point (2Point)	2P
K Point (KPoint)	KP
Differential Evolution (DE)	DE
Exponential (Exp)	EXP
Simulated Binary (SBX)	SBX

(b) Reference table to mutation operators acronyms

Mutation operator	Acronym
Polynomial (PM)	PM or absent

(c) Reference table to selection operators acronyms

Selection operator	Acronym
Random (RandS)	Ran or R
Tournament	Tour or T

(d) Reference table to algorithms acronyms

Algorithm	Acronym
Two-archive evolutionary algorithm for constrained multi objective optimization (Li et al., 2019)	C-TAEA, CTAEA or CT
Non-dominated Sorting Genetic Algorithm III (Deb and Jain, 2014)	NSGA-III, NSGA3 or N3
Fast Non-dominated Sorting Genetic Algorithm (Deb et al., 2002)	NSGA-II, NSGA2 or N2
Multiobjective selection based on dominated hypervolume (Beume et al., 2007)	SMS-EMOA, SEMOA or SE

especially in its combination with the Tournament selection operator. Exceptions to this observation include the RandS selection operator half of RF00009.115, RF00012.15, RF00017.90 and RF00030.30, as well as RF00028.1 where the loss of 18 combinations and the high diversity of the remaining make difficult to establish comparisons. Another observation that can be easily detected is the common lower performance of DE when compared to its fellow crossover operator partners (using the same algorithm + selection operator), since it is always the worst or almost the worst. The exceptions are RF00003.94 where, leaving aside C-TAEA as it does not return results with it, DE is the best crossover operator among its partners, both in median value and dispersion. Both effects are more noticeable when using RandS than Tournament. For RF00014.2 DE is the worst among its fellows always but with SMS-EMOA, where the performance values are very similar. Considering RF00012.15, this effect is only seen with HV and IGD for CTAEA in combination with RandS, and only with HV for NSGA-III and SMS-EMOA. In RF00009.115, RF00017.90 and RF00030.30 DE is clearly the worst among its partners always with C-TAEA, and in combination with NSGA-III, NSGA-II and SMS-EMOA when using Tournament, but not in their RandS versions. Similar to the previous group, for RF00028.1 DE with RandS seems the best combination of operators for NSGA-III, NSGA-II and SMS-EMOA, but not for C-TAEA. Some boxplots of RFAM structures show a large dispersion of values (RF00003.94, RF00028.1 and to a lesser extent RF00030.30 or even RF00009.115). This effect is probably due to the higher difficulty of these structures, which has a negative impact on the number of repetitions solving the problem, thus widening the dispersion. In addition, C-TAEA tends to have a larger dispersion than the other algorithms. Since it appears worse than them, this is probably showing the same effect of fewer repetitions managing to solve the structure. Along the RFAM structures NSGA-II_Tournament_PM_SBX boxplot does not show any remarkable behavior. Its values are usually in the middle of those in the NSGA-II boxplots.

In Tables 3 and 4, the final average HV and IGD values of the 48 alg+oper combinations are shown. If none of the repetitions of a combination returns results, its performance values are represented as “NA”. In addition, the best three values of each RFAM structure are highlighted and colored. These numerical tables allow for more precise comparisons than boxplots and heat maps. As expected, in most cases the highlighted values coincide in both tables, but not always. They tend to be concentrated in the SMS-EMOA area, which points to a better performance of this algorithm. Moreover, the combinations with the highest number of outstanding values are both versions of SMS-EMOA + Exp, which allows us to suspect that one of them is going to be the best combination. In addition, outside of the SMS-EMOA area there is a positive trend to the crossover operator Exp. These last two observations make Exp a solid candidate to be the best crossover operator. Inspection of the boxplots allowed us to see that C-TAEA tends to be the worst algorithm, but with some exceptions including RF00009.115, RF00017.90 and RF00030.30 when using the RandS selection operator. When looking at these tables we can realize that not only they are not the worst, but in some cases they are among the three best: C-TAEA_RandS_PM_1Point (RF00009.115, only for HV), C-TAEA_RandS_PM_2Point (RF00009.115 and RF00017.90), C-TAEA_RandS_PM_SBX (RF00009.115 and RF00030.30) and C-TAEA_RandS_PM_Exp (RF00017.90). Another observation from the boxplots is that for RF00003.94, DE is the best crossover operator among its partners (except for C-TAEA, where there are few results), this tendency is more pronounced with RandS than with Tournament. This can be checked in the tables, and is reflected in the fact that the values of the combinations using DE and RandS are highlighted, except for SMS-EMOA_RandS_PM_Exp, which ranks third for HV, and is closely followed by NSGA-II_RandS_PM_DE (240.5 and 239.6 E-03 respectively), which is the best among its partners as well (and is third when looking at IGD). Surprisingly, for HV and Tournament selection operator, DE is not the best crossover operator with this metric, but is in the middle of its partners. This can be explained by the high dispersion of these combinations, since the infrequent but very high HV values of the partners may be attracting the calculated average, making it higher. The combination NSGA-II_Tournament_PM_SBX is not highlighted in either table. In addition, it does not return results for RF00028.1.

From the means of the final and intermediate HV and IGD values of the 31 repetitions, convergence plots with different color codes have

Table 3
Average HV values of RFAM structures. Values are expressed in scientific notation(E-03).

Combination	R01	R02	R03	R04	R05	R06	R07	R08	R09	R12	R13	R14	R15	R17	R18	R19	R21	R22	R25	R26	R27	R28	R29	R30
CT-R-PM-1P	333.9	366.6	15.9	443.1	603.9	487.7	422.1	577.4	🟨 **314.3**	480.9	399.0	534.7	390.7	446.1	165.7	526.4	573.3	548.7	415.7	451.9	623.3	134.5	547.5	311.3
CT-R-PM-2P	328.8	440.0	NA	485.2	611.4	509.3	438.8	574.9	🟩 **326.8**	470.1	420.0	541.4	429.3	🟦 **481.6**	282.0	536.9	567.0	571.7	466.2	441.0	633.3	57.9	555.0	432.7
CT-R-PM-DE	48.3	174.3	NA	355.9	477.0	319.2	279.6	473.4	53.0	378.7	282.9	412.8	247.3	280.8	8.7	385.6	484.9	402.7	364.9	340.3	541.0	15.7	397.8	278.4
CT-R-PM-EXP	401.2	471.0	11.5	511.9	608.4	517.0	464.0	582.8	279.1	506.3	441.9	549.2	457.2	🟩 **463.7**	305.5	545.9	579.8	594.6	536.7	453.3	631.6	8.4	563.1	380.4
CT-R-PM-KP	330.6	439.0	NA	496.9	603.2	479.5	428.2	571.2	310.0	499.1	428.0	532.8	431.1	433.9	249.3	530.2	552.1	577.5	463.8	423.6	622.8	32.2	551.8	421.2
CT-R-PM-SBX	253.3	437.2	NA	431.2	592.9	480.0	401.7	558.5	🟦 **343.8**	453.0	362.1	553.6	356.2	351.9	144.7	534.9	572.7	543.6	424.5	431.0	613.4	NA	541.9	🟨 **447.3**
CT-T-PM-1P	37.6	39.8	NA	261.2	515.0	231.9	226.9	550.0	126.1	328.5	217.1	386.6	68.8	96.5	17.0	319.6	405.4	314.0	214.3	377.4	637.4	12.6	197.5	94.8
CT-T-PM-2P	10.1	96.3	NA	264.1	459.8	240.3	215.5	514.0	143.1	278.9	243.7	386.7	156.0	79.7	25.4	324.1	438.7	338.5	186.4	368.2	635.6	0	179.0	169.3
CT-T-PM-DE	0	4.2	NA	219.9	291.7	15.6	25.3	289.5	18.8	237.4	215.5	323.8	38.0	61.2	6.7	56.2	450.9	124.6	72.4	115.1	551.5	NA	15.0	97.2
CT-T-PM-EXP	14.7	93.6	NA	258.3	551.2	255.4	175.8	557.8	34.4	215.7	172.5	436.7	161.8	53.7	56.0	346.4	467.8	377.9	186.9	380.6	633.0	0	165.2	161.4
CT-T-PM-KP	13.5	69.8	NA	263.5	415.7	225.5	160.8	441.8	66.5	240.3	207.2	343.9	93.1	41.5	15.0	271.7	421.4	337.0	182.4	368.8	626.1	NA	186.8	134.6
CT-T-PM-SBX	4.0	81.4	NA	371.6	404.7	163.0	119.2	482.7	119.4	333.6	211.6	453.7	75.0	145.9	22.9	169.8	564.3	362.5	151.5	288.1	626.5	5.9	97.5	207.1
N3-R-PM-1P	490.9	531.5	170.5	515.2	639.0	591.9	534.0	628.0	196.8	493.8	476.2	582.5	530.2	316.3	430.8	616.6	640.2	622.2	563.2	518.5	673.8	4.4	618.8	322.1
N3-R-PM-2P	487.0	523.1	189.2	482.6	637.9	607.6	540.1	625.9	300.4	488.5	439.4	580.8	522.6	324.8	428.0	593.8	626.7	610.4	552.6	508.4	673.9	6.1	611.6	409.1
N3-R-PM-DE	348.1	395.6	🟩 **249.2**	441.5	596.9	493.5	410.7	588.6	223.5	463.7	388.6	555.0	371.0	364.2	229.8	565.6	542.1	490.0	466.5	404.2	614.7	🟨 **169.8**	551.2	334.5
N3-R-PM-EXP	506.0	553.9	220.7	558.6	645.6	613.5	581.6	627.1	313.2	🟨 **572.2**	518.6	587.2	568.4	421.5	🟦 **565.2**	605.7	643.8	644.6	🟨 **610.8**	523.1	675.0	NA	628.1	🟩 **469.2**
N3-R-PM-KP	477.3	525.3	117.4	469.2	631.7	589.7	519.1	626.3	237.1	460.6	401.6	571.9	510.4	287.5	379.7	586.7	610.9	589.9	534.7	495.9	668.7	0	615.7	412.3
N3-R-PM-SBX	456.3	499.2	107.9	523.9	636.8	572.5	509.8	625.3	158.3	503.5	486.4	568.1	481.4	370.0	286.8	597.8	634.1	584.2	545.7	483.2	665.8	0	609.9	400.0
N3-T-PM-1P	492.4	527.2	178.0	495.4	650.3	603.5	541.8	630.4	162.3	476.9	483.6	579.3	520.8	293.1	438.1	632.0	652.1	628.0	557.1	519.8	673.9	NA	613.6	293.2
N3-T-PM-2P	494.7	526.4	111.7	473.3	645.7	604.1	533.6	627.6	233.7	498.7	435.4	593.9	508.6	305.3	405.0	619.4	629.2	610.2	554.3	510.2	675.2	NA	622.2	407.7
N3-T-PM-DE	313.6	353.1	124.3	446.4	595.8	516.2	361.9	590.5	58.7	449.2	385.4	547.3	341.4	170.6	156.6	565.4	542.9	488.6	358.0	401.7	606.8	38.7	550.8	137.2
N3-T-PM-EXP	521.6	551.6	185.4	556.1	646.0	615.3	583.0	629.6	303.4	556.7	514.4	595.5	559.8	417.9	🟩 **564.0**	621.0	648.2	637.3	604.7	521.1	672.6	17.7	629.9	🟦 **472.4**
N3-T-PM-KP	480.9	519.3	137.8	463.1	636.4	581.9	510.9	622.6	197.8	458.1	406.9	590.1	491.3	219.3	379.7	606.5	624.7	586.5	518.4	503.4	668.5	NA	610.1	330.1
N3-T-PM-SBX	481.0	520.1	119.5	507.6	636.4	570.7	498.7	625.8	108.4	483.0	452.3	572.9	474.2	308.6	274.5	604.6	615.9	580.0	494.6	481.3	666.1	NA	606.1	270.1
N2-R-PM-1P	427.1	472.2	154.0	460.1	613.3	522.6	510.4	598.2	117.4	443.7	441.3	590.4	474.5	265.3	379.4	592.6	599.6	572.2	490.5	488.5	648.6	2.0	565.2	232.0
N2-R-PM-2P	441.4	468.4	162.8	452.4	609.3	537.8	510.2	596.0	231.2	459.9	425.4	584.0	468.0	264.5	353.3	564.5	606.7	567.0	494.9	490.7	651.2	5.7	572.2	324.3
N2-R-PM-DE	339.3	378.2	239.6	441.9	576.1	447.0	390.7	549.6	214.2	457.6	380.9	556.6	353.2	362.7	252.4	545.9	546.9	471.2	462.9	388.4	590.9	🟩 **174.0**	513.7	328.0
N2-R-PM-EXP	461.2	473.0	200.2	517.6	610.4	532.7	529.8	596.6	218.3	493.8	485.5	578.4	503.1	349.5	504.1	578.2	624.1	566.3	531.1	487.8	644.3	NA	579.4	353.3
N2-R-PM-KP	425.9	464.1	99.8	439.7	608.9	530.5	486.7	594.7	180.4	436.7	393.9	580.9	445.7	257.1	333.3	564.2	600.9	552.9	477.5	490.6	655.8	0	577.4	337.8
N2-R-PM-SBX	432.7	441.4	85.7	504.2	604.3	506.7	459.1	591.2	149.9	485.9	468.0	571.4	438.7	372.9	312.0	561.7	601.3	542.7	507.3	485.1	648.0	0	568.1	376.6
N2-T-PM-1P	445.7	460.0	154.8	480.1	609.8	515.9	500.5	591.1	116.7	429.0	446.3	574.3	463.7	251.7	375.3	591.2	614.8	559.6	485.3	493.0	639.7	NA	548.5	225.5
N2-T-PM-2P	441.2	454.3	99.3	453.6	609.0	514.9	490.7	587.4	179.1	462.8	414.3	594.7	444.2	256.2	346.3	564.5	596.0	550.1	484.0	487.2	645.4	NA	554.4	324.4
N2-T-PM-DE	270.6	300.3	113.0	433.8	569.3	443.4	269.2	550.4	49.1	421.7	392.2	551.9	284.9	165.9	131.4	528.9	531.2	442.4	322.3	393.1	590.6	47.7	507.3	97.5
N2-T-PM-EXP	449.5	459.3	157.2	514.5	608.7	514.4	514.8	588.0	238.6	486.2	490.7	595.9	480.5	378.3	513.2	572.5	616.6	560.7	520.0	479.1	637.2	13.4	557.9	377.8
N2-T-PM-KP	418.7	445.6	126.4	424.4	609.3	510.2	481.6	589.4	139.6	441.0	378.1	582.8	427.8	223.5	338.7	565.2	583.3	539.2	456.9	488.8	648.6	NA	551.2	282.8
N2-T-PM-SBX	429.5	434.0	98.2	479.4	591.8	504.0	449.9	580.4	114.7	459.5	441.5	562.9	421.7	322.4	294.0	565.1	586.6	522.3	444.8	475.1	640.0	NA	555.3	271.5
SE-R-PM-1P	526.0	568.5	177.1	559.3	🟩 **669.4**	632.0	578.4	650.6	159.2	500.6	533.5	601.3	566.1	280.8	440.5	644.1	673.1	661.7	564.8	550.8	695.7	3.3	644.2	276.5
SE-R-PM-2P	545.1	578.2	201.6	535.1	665.0	🟨 **653.9**	588.7	651.6	279.0	532.5	499.7	597.7	572.3	295.0	406.0	628.5	670.4	🟨 **678.9**	570.1	🟨 **554.5**	696.9	6.7	🟩 **653.7**	357.5
SE-R-PM-DE	365.8	414.8	🟦 **257.8**	477.2	638.0	525.3	445.8	636.1	248.5	492.8	419.5	🟩 **612.2**	388.3	389.3	243.9	612.6	607.7	534.7	497.3	439.5	676.7	🟦 **175.0**	590.9	353.2
SE-R-PM-EXP	🟦 **564.0**	🟦 **602.8**	🟨 **240.5**	🟦 **620.4**	666.2	🟦 **662.0**	🟩 **625.1**	🟩 **655.1**	260.4	🟩 **578.8**	🟦 **594.7**	607.0	🟦 **615.8**	381.9	506.7	642.8	🟨 **680.5**	🟦 **696.7**	🟦 **636.6**	🟦 **567.4**	696.8	NA	🟨 **653.4**	399.4
SE-R-PM-KP	539.1	551.5	130.6	514.1	660.8	639.6	568.7	652.2	204.0	506.9	490.7	591.3	547.8	280.6	370.8	621.4	668.7	644.6	538.4	550.6	696.2	0	649.4	371.1
SE-R-PM-SBX	521.4	562.2	115.4	🟩 **608.8**	666.2	634.7	572.6	653.6	204.3	🟦 **582.1**	🟨 **574.7**	599.0	527.2	🟨 **448.8**	357.1	640.9	679.2	656.1	584.4	528.1	🟩 **700.7**	0	643.5	435.6
SE-T-PM-1P	544.2	572.7	197.6	545.2	657.7	639.3	🟨 **592.5**	651.6	125.5	502.5	548.2	606.9	🟨 **573.8**	283.8	410.5	🟩 **655.0**	🟦 **686.6**	668.8	563.2	551.5	693.6	NA	643.9	275.1
SE-T-PM-2P	🟨 **554.2**	🟨 **580.6**	118.4	532.7	🟨 **668.1**	649.7	581.0	652.3	209.1	529.2	508.5	🟨 **609.2**	553.8	275.1	383.2	🟨 **649.1**	671.4	656.0	563.4	552.0	695.7	NA	650.9	330.8
SE-T-PM-DE	347.9	358.5	128.3	473.0	638.2	541.6	339.1	634.1	43.4	464.5	413.7	608.9	330.4	184.5	149.6	608.9	609.1	506.6	391.0	442.0	673.3	44.5	586.0	141.4
SE-T-PM-EXP	🟩 **557.1**	🟩 **596.1**	201.7	🟨 **598.8**	668.1	🟩 **657.6**	🟦 **637.1**	🟦 **658.5**	269.2	570.4	🟩 **587.9**	🟦 **618.9**	🟩 **608.4**	394.2	🟨 **522.2**	🟦 **663.6**	🟩 **681.3**	🟩 **694.6**	🟩 **632.1**	🟩 **557.1**	🟨 **697.7**	12.3	🟦 **657.4**	377.9
SE-T-PM-KP	526.0	560.1	166.8	514.9	663.9	636.3	573.9	651.7	162.5	519.9	491.0	604.3	540.7	237.0	349.6	637.2	664.9	638.2	542.4	543.7	696.4	NA	650.9	296.4
SE-T-PM-SBX	532.5	567.5	129.8	579.9	🟦 **669.8**	636.6	574.5	🟨 **654.7**	162.7	558.0	546.5	606.8	535.6	361.4	339.9	641.6	672.7	641.8	574.3	524.1	🟦 **701.1**	NA	652.5	312.6

Note that the color code used refers to the top three values, specifically: 🟦 First, 🟩 Second, 🟨 Third.

been created to highlight all the studied features (crossover operators in Figs. 6 and 7, selection operators in Figs. 8 and 9 and algorithms in Figs. 10 and 11). As in the case of the boxplots, only a selection of them is shown, and the complete set can be consulted in the supplementary material. These plots allow us to perform an analysis of convergence, in order to evaluate if the convergence has been reached, and the behavior of the alg+oper combinations as they advance in time. It should be noted that some runs missed several intermediate points in the execution and returned fewer intermediate results than expected. When this was the case, the last value was prolonged to the end, which in some plots generates a "false plateau" effect, easily detectable due to the completely flat slope of the lines at the end except for the last point, which is always the one corresponding to the final solutions. A first visual inspection of these plots shows that convergence may have been reached (or is close) for some RFAM structures (RF00001.121, RF00002.2, RF00005.1, RF00006.1, RF00008.11, RF00014.2, RF00019.115, RF000021.10, RF00022.1, RF00026.1, RF000027.7 and RF000029.107), but clearly not for others (RF00003.94, RF00009.115, RF00017.90, RF000018.2, RF000025.12, RF00028.1 and RF00030.30). Since the last point represented in these plots has the same value than Tables 3 and 4, the same conclusions reached from these tables can be seen here in a more visual way when looking at the end of the lines. For example, our two presumed candidates for the best alg+oper combination (both versions of SMS-EMOA + Exp) are usually found among the top performers. The NSGA-II_Tournament_PM_SBX combination always ranks in the middle or worse half, but does not reach the bottom positions (its worst position is 5th from the bottom, with RF00003.94 for HV).

Focusing on crossover operators (Figs. 6 and 7), we can see how Exp tends to be in high-performing positions and DE in low-performing ones (but not always), with the obvious exceptions of RF00003.94 and RF00028.1. The rest of the crossover operators do not exhibit any recognizable pattern. From inspection of Figs. 8 and 9 (colored by selection operators), it is not possible to come to any firm conclusions, but there seems to be a tendency for RandS to be in better positions and improve performance faster than Tournament, although it is not very marked. Color patterns are more evident in Figs. 10 and 11 (by algorithms), especially in those RFAM structures for which we have said that convergence seems to have been achieved, and leaving aside those that clearly have not. From this observation we can conclude that, when there is enough time to be close to convergence, the most important alg+oper combination parameter for performance is algorithm. Based on their line positions, the best algorithm would be SMS-EMOA and the worst by far C-TAEA, as we indicated above. NSGA-III and NSGA-II have not been ranked so far, but from these plots it is possible to conclude that NSGA-III is, in general, better than NSGA-II. We will now turn our attention to some of the detected exceptions to the aforementioned observations, looking for an explanation in these graphs:

- From inspection of Tables 3 and 4, we saw that some combinations that use C-TAEA+RandS are among the top three for RF00009.115, RF00017.90 and RF00030.30. Above we classified these RFAM structures as not reaching the convergence. This is especially clear for the other algorithms besides C-TAEA (in fact, C-TAEA could have converged). If we look at the slope of the combinations (omitting the false plateau part), it is clear that C-TAEA in combination with RandS improves performance more quickly (with some exceptions), but stagnates earlier than other algorithms. With more time, many of these would improve their performance, outperforming C-TAEA. Furthermore, from boxplots (Figs. 2 and 3) we identified RF00012.15 as another exception of C-TAEA in combination with RandS which is not the worst algorithm (besides the three RFAM structures aforementioned),

Table 4
Average IGD values of RFAM structures. Values are expressed in scientific notation (E-03).

Combination	R01	R02	R03	R04	R05	R06	R07	R08	R09	R12	R13	R14	R15	R17	R18	R19	R21	R22	R25	R26	R27	R28	R29	R30
CT-R-PM-1P	270.3	260.1	955.0	183.9	94.0	180.0	194.1	108.8	262.2	185.4	189.5	130.0	217.1	194.4	415.3	148.0	130.6	143.0	234.0	116.7	116.5	675.8	134.8	314.0
CT-R-PM-2P	305.6	210.2	NA	152.0	87.9	161.2	181.9	109.0	256.7	187.3	177.4	127.2	187.4	180.3	320.3	137.7	129.8	127.1	198.1	120.7	111.2	889.5	132.5	226.4
CT-R-PM-DE	685.2	419.8	NA	242.6	178.8	284.6	312.3	170.7	604.6	246.0	294.4	208.7	336.8	313.1	781.5	239.7	183.9	239.1	288.7	184.1	168.7	952.2	225.0	366.6
CT-R-PM-EXP	188.9	219.1	974.5	138.3	91.3	157.5	166.3	104.9	302.4	166.5	172.4	120.0	173.5	**185.6**	302.8	133.2	126.5	119.7	166.6	113.8	110.9	964.1	132.7	273.2
CT-R-PM-KP	260.2	211.4	NA	144.5	93.7	180.8	188.5	111.9	269.9	162.0	185.8	129.1	184.4	199.6	353.9	143.3	136.6	124.6	203.6	132.6	121.5	938.9	134.2	230.9
CT-R-PM-SBX	428.8	187.3	NA	198.9	103.2	177.8	217.1	122.4	**257.9**	199.6	251.1	116.0	244.0	245.4	464.7	143.0	127.8	150.0	248.1	124.9	122.4	NA	142.8	215.0
CT-T-PM-1P	736.6	646.6	NA	325.8	152.0	360.9	377.8	124.4	466.6	299.9	344.0	225.5	559.7	452.5	685.4	283.7	232.4	335.7	399.5	158.9	110.8	815.9	423.0	565.8
CT-T-PM-2P	794.0	551.7	NA	324.4	192.2	337.2	386.2	149.4	456.1	340.6	312.6	231.6	437.7	499.8	674.9	281.2	206.9	307.7	437.1	160.5	109.2	992.1	427.7	459.1
CT-T-PM-DE	997.6	895.0	NA	379.6	326.4	775.0	777.9	301.6	778.6	403.0	371.1	271.3	726.2	561.4	801.3	635.3	204.1	522.9	709.5	414.1	158.5	NA	773.6	603.5
CT-T-PM-EXP	833.6	539.5	NA	334.0	127.0	331.5	436.2	122.6	585.6	413.9	403.9	194.0	433.1	518.2	560.5	267.8	191.9	276.4	439.4	152.9	113.4	987.9	447.3	456.6
CT-T-PM-KP	815.2	588.5	NA	322.4	219.6	352.1	447.8	199.1	567.7	380.0	350.4	262.8	521.3	544.9	755.0	319.2	221.1	306.5	439.7	158.5	120.0	NA	419.7	500.6
CT-T-PM-SBX	897.0	552.4	NA	241.2	238.4	413.1	490.4	175.3	456.2	290.9	369.8	186.8	551.4	407.4	707.6	415.8	140.8	267.5	494.3	226.4	119.0	981.5	533.1	413.5
N3-R-PM-1P	124.0	128.5	667.2	139.3	73.5	104.3	129.1	72.8	356.1	182.0	147.0	91.8	126.3	261.1	209.9	86.3	86.9	102.8	146.8	88.1	78.4	967.0	90.0	323.4
N3-R-PM-2P	125.4	133.6	642.6	155.3	74.3	98.2	128.8	74.2	285.8	176.6	163.9	93.2	128.8	253.4	219.3	93.9	88.2	111.7	151.5	91.2	79.2	968.2	89.8	259.4
N3-R-PM-DE	219.3	205.5	**346.6**	176.5	92.5	152.1	202.2	88.6	385.4	184.4	214.8	109.7	225.2	273.4	396.4	107.9	132.8	174.3	218.5	131.4	108.5	**601.2**	110.0	323.8
N3-R-PM-EXP	116.9	118.1	605.9	122.1	72.3	96.0	104.4	74.2	**258.3**	143.0	123.1	88.3	105.4	207.4	174.7	90.4	81.8	98.5	130.5	87.7	78.9	NA	83.5	**224.2**
N3-R-PM-KP	127.9	132.6	760.8	161.0	77.0	103.8	142.0	73.9	335.9	195.0	189.4	98.3	138.8	281.2	232.2	97.1	96.1	119.4	163.2	92.5	83.3	988.8	87.4	251.5
N3-R-PM-SBX	141.6	137.9	790.8	127.2	75.1	109.0	145.7	75.7	434.4	165.3	147.4	97.2	151.7	228.5	325.2	92.4	85.7	126.2	159.0	99.6	83.1	962.1	88.6	276.8
N3-T-PM-1P	124.6	128.5	667.9	152.5	70.3	100.8	125.6	74.5	381.2	195.7	142.9	94.7	132.6	271.6	210.5	81.8	83.1	103.1	149.5	88.5	78.7	NA	91.9	342.6
N3-T-PM-2P	120.8	131.0	755.5	168.5	72.3	98.9	136.4	74.1	334.4	172.4	167.7	88.3	136.3	264.0	221.7	85.1	87.6	112.7	151.3	89.1	78.9	NA	84.7	257.6
N3-T-PM-DE	246.4	239.3	574.4	174.1	91.8	136.7	238.3	88.4	618.3	195.9	214.9	112.1	245.3	425.7	474.3	107.8	130.8	176.5	314.2	132.3	113.7	905.3	111.1	528.2
N3-T-PM-EXP	107.6	121.5	664.8	124.5	71.3	95.6	106.2	72.7	264.7	154.2	125.0	88.0	110.3	210.5	181.1	84.3	83.7	100.2	134.0	91.7	81.9	941.9	82.7	**221.7**
N3-T-PM-KP	126.5	132.3	712.3	174.2	75.2	106.7	147.2	74.8	364.3	198.6	187.7	89.6	149.3	325.6	233.8	88.3	90.7	124.4	171.0	91.3	84.0	NA	89.7	315.0
N3-T-PM-SBX	124.8	133.0	744.7	137.0	74.7	109.9	149.3	76.0	476.3	185.6	165.0	95.1	160.1	269.4	334.2	89.7	93.9	128.6	188.9	100.9	85.6	NA	91.4	378.2
N2-R-PM-1P	176.1	170.0	669.1	174.9	97.7	156.2	148.3	96.8	430.1	214.0	170.4	90.1	163.5	289.9	230.9	110.1	121.2	136.4	188.6	117.9	95.0	969.2	132.5	396.5
N2-R-PM-2P	164.5	172.4	661.1	177.9	98.3	143.6	152.8	97.2	331.2	200.1	180.3	93.4	165.4	289.7	244.3	121.0	109.9	139.3	181.1	114.9	92.1	970.3	120.9	302.2
N2-R-PM-DE	229.5	223.9	**358.1**	183.4	109.1	185.2	220.9	112.0	398.9	191.5	223.7	110.2	243.0	281.0	379.2	124.8	136.2	192.6	223.2	155.4	116.9	594.8	144.8	334.7
N2-R-PM-EXP	157.1	169.9	619.0	149.8	100.5	145.7	144.0	98.7	332.2	185.1	150.7	96.9	152.9	233.5	174.6	117.9	102.7	145.3	176.4	118.5	95.9	NA	118.6	286.6
N2-R-PM-KP	170.2	173.5	773.1	184.4	94.7	140.2	166.7	96.6	374.1	211.7	202.6	95.9	182.9	296.9	259.4	119.7	109.7	141.1	192.4	110.8	90.3	988.0	118.0	287.9
N2-R-PM-SBX	166.9	183.2	804.1	147.7	97.9	151.0	181.7	97.5	437.0	175.3	168.1	96.7	183.3	236.3	310.9	121.4	112.0	148.3	185.7	111.4	92.4	964.3	119.8	278.1
N2-T-PM-1P	167.1	176.2	685.0	168.3	104.7	157.6	162.9	103.5	428.2	221.7	173.7	100.1	174.4	300.3	237.1	114.8	115.8	149.1	197.4	114.3	101.4	NA	142.0	400.2
N2-T-PM-2P	166.1	184.3	762.6	182.0	97.0	156.1	165.0	104.9	377.7	197.0	192.4	89.4	187.0	295.1	252.3	125.0	123.4	152.3	193.7	116.3	94.8	NA	135.6	304.7
N2-T-PM-DE	288.4	295.4	591.3	189.9	115.2	186.8	326.4	115.1	641.2	222.9	214.7	113.7	303.5	444.9	507.7	137.2	146.6	219.9	359.7	155.3	117.7	896.5	151.9	597.0
N2-T-PM-EXP	164.5	177.4	681.6	157.2	99.7	159.3	156.8	101.7	321.2	193.7	152.7	88.2	167.7	218.2	**173.3**	123.1	113.4	153.0	188.0	124.7	98.8	943.2	134.9	262.2
N2-T-PM-KP	178.4	185.3	722.7	201.0	98.3	158.1	171.5	101.0	408.0	209.9	210.6	95.6	195.2	320.6	256.6	120.9	123.4	157.2	209.5	109.6	94.2	NA	134.0	344.3
N2-T-PM-SBX	169.5	188.2	760.6	163.5	107.9	151.2	188.3	104.9	474.9	194.1	185.2	104.5	198.8	275.4	326.7	122.3	120.7	166.0	231.4	116.5	94.0	NA	129.4	377.1
SE-R-PM-1P	117.2	107.9	661.4	116.3	64.2	87.6	105.1	62.8	389.9	182.7	122.6	84.6	113.1	280.9	195.9	78.8	76.2	88.4	138.3	73.7	68.8	968.4	76.3	358.0
SE-R-PM-2P	105.2	103.9	640.1	128.3	63.6	**79.3**	98.4	61.2	295.0	149.4	133.3	84.3	**104.6**	268.6	214.2	80.8	74.3	**80.2**	137.4	72.8	68.6	969.6	74.4	288.7
SE-R-PM-DE	207.1	192.8	341.6	155.0	69.2	131.8	178.7	64.4	365.4	166.5	194.4	82.1	214.2	257.9	387.5	88.1	101.3	146.5	196.6	110.7	80.4	**598.8**	91.9	312.5
SE-R-PM-EXP	99.8	96.2	597.3	**98.4**	64.3	78.7	**85.1**	**60.8**	292.3	**137.3**	94.7	81.5	89.3	215.9	**170.1**	77.7	**71.3**	77.5	105.8	71.7	68.9	NA	74.5	256.6
SE-R-PM-KP	104.5	118.3	748.1	140.2	63.2	82.8	110.8	61.6	351.5	169.7	141.5	86.3	119.6	279.2	234.0	80.8	74.1	94.1	151.4	73.7	69.7	988.7	74.5	270.1
SE-R-PM-SBX	116.2	111.5	781.2	96.4	**63.1**	85.5	112.5	61.2	373.4	124.6	**110.6**	78.7	125.4	**181.0**	274.0	77.0	71.4	89.7	**130.4**	79.8	67.1	963.6	74.6	234.6
SE-T-PM-1P	109.4	107.1	660.1	128.1	64.7	84.5	**97.9**	62.9	416.6	179.6	113.1	81.0	105.1	277.5	211.6	**74.5**	**71.2**	86.4	136.4	**72.7**	70.6	NA	76.9	360.8
SE-T-PM-2P	**102.6**	**103.5**	754.2	137.1	**62.9**	80.7	103.9	61.7	349.3	160.4	130.6	81.7	115.1	284.3	225.9	**74.8**	74.6	89.8	138.7	**72.3**	**68.1**	NA	**73.2**	305.8
SE-T-PM-DE	220.3	239.2	572.2	159.6	67.7	122.2	261.9	66.0	637.5	189.2	199.7	85.5	262.1	421.4	493.0	91.4	101.3	166.9	292.0	109.6	82.6	897.0	93.4	532.0
SE-T-PM-EXP	**103.2**	**97.2**	657.8	**102.8**	63.1	**78.8**	81.3	**60.7**	288.5	144.6	**98.0**	**78.7**	**94.6**	207.7	168.4	73.7	70.0	**78.3**	**110.1**	73.2	68.8	947.4	72.7	267.1
SE-T-PM-KP	112.8	113.5	695.6	145.5	64.6	85.1	109.2	60.6	386.0	162.2	139.5	84.0	126.0	312.1	246.4	76.6	75.0	98.3	150.0	74.8	69.1	NA	73.4	335.9
SE-T-PM-SBX	109.3	106.7	738.6	108.5	62.4	85.1	110.1	61.8	408.6	**138.1**	123.2	**80.1**	122.8	228.1	284.3	77.3	74.2	95.8	137.8	81.7	**67.5**	NA	**73.2**	345.5

Note that the color code used refers to the top three values, specifically: 🟥 First, 🟪 Second, 🟧 Third.

Table 5
Average HV and IGD values of crossover operators. Values are expressed in scientific notation (E-03).

(a) HV

Op.	R01	R02	R03	R04	R05	R06	R07	R08	R09	R12	R13	R14	R15	R17	R18	R19	R21	R22	R25	R26	R27	R28	R29	R30
1P	**412.2**	442.3	**131.0**	**470.0**	**619.8**	**528.1**	**488.3**	**609.7**	164.8	457.0	**443.2**	557.0	**448.6**	279.2	**332.2**	**572.2**	**605.7**	**571.9**	**481.8**	**493.9**	**660.8**	**19.6**	547.4	253.8
2P	**412.8**	**458.4**	110.4	459.9	**613.3**	**539.7**	**487.3**	**603.7**	**237.8**	**465.1**	423.3	**561.0**	**456.8**	**285.3**	**328.7**	**560.1**	600.8	**572.9**	**484.0**	**489.0**	**663.4**	**9.5**	**549.9**	**344.5**
DE	254.2	297.4	**139.0**	411.2	547.9	412.7	315.3	539.0	113.6	420.7	359.8	521.1	294.3	247.4	147.4	483.6	539.5	432.6	366.9	365.6	605.7	**83.2**	464.1	220.9
EXP	**434.4**	**475.2**	**152.2**	**517.0**	**625.6**	**546.0**	**513.9**	**611.9**	**239.6**	**497.5**	**475.8**	**571.1**	**494.4**	**357.6**	**442.1**	**572.0**	**617.7**	**596.6**	**532.4**	**496.2**	**661.0**	6.5	**554.3**	**374.0**
KP	401.5	**446.8**	97.4	448.2	603.7	524.2	466.2	593.7	**187.2**	445.3	399.7	549.8	436.0	247.6	302.0	547.9	590.9	558.2	464.3	483.2	660.4	4.0	549.2	323.3
SBX	388.8	442.9	82.0	**500.8**	600.4	508.5	448.2	596.5	170.2	**482.3**	**442.9**	**561.1**	413.8	**335.3**	254.0	539.6	**615.8**	554.2	465.9	462.0	657.7	0.74	**534.4**	**340.1**
Best	EXP	EXP	EXP	EXP	EXP	EXP	EXP	EXP	EXP	EXP	EXP	EXP	EXP	EXP	EXP	1P	EXP	EXP	EXP	EXP	2P	DE	EXP	EXP

Note that the color code used refers to the top three values, specifically: 🟦 **First**, 🟩 **Second**, 🟨 **Third**.

(b) IGD

Op.	R01	R02	R03	R04	R05	R06	R07	R08	R09	R12	R13	R14	R15	R17	R18	R19	R21	R22	R25	R26	R27	R28	R29	R30
1P	**228.2**	215.6	**745.7**	**173.6**	**90.1**	154.0	**167.6**	**88.3**	391.4	207.6	**175.4**	112.2	**199.0**	**291.0**	**299.6**	**122.3**	114.7	**143.1**	**198.8**	**103.9**	**90.0**	**924.5**	145.9	382.6
2P	**235.5**	**198.8**	777.0	178.2	**93.6**	**144.4**	**169.2**	**91.5**	**335.8**	**198.0**	**182.3**	**111.1**	**182.8**	291.9	**296.6**	**125.0**	**111.9**	**140.1**	**198.6**	**104.7**	**87.8**	973.7	**142.4**	**300.5**
DE	386.7	338.9	**598.0**	207.6	131.3	246.8	314.8	125.8	553.7	224.9	241.0	136.7	319.5	372.3	527.6	191.5	142.1	229.8	325.3	174.1	118.4	**805.7**	212.7	449.8
EXP	**221.4**	**192.4**	**725.1**	**153.4**	**86.2**	**142.9**	**160.0**	**87.0**	**330.6**	**192.3**	**165.1**	**104.4**	**165.9**	**249.6**	**238.2**	**121.0**	**105.2**	**131.1**	**181.4**	**104.3**	**89.7**	**973.1**	**143.4**	**281.0**
KP	237.0	206.9	801.6	184.2	98.3	**151.2**	185.5	97.4	**382.2**	211.1	200.9	117.7	202.2	320.0	321.4	130.7	115.8	145.7	210.1	105.5	91.5	988.1	**141.4**	317.0
SBX	269.2	**200.0**	827.5	**152.5**	102.8	160.3	199.4	96.8	414.8	**184.2**	190.1	**106.9**	217.2	**259.0**	378.4	142.4	**103.3**	146.5	222.0	117.7	91.4	983.9	156.6	**314.9**
Best	EXP	EXP	DE	SBX	EXP	EXP	EXP	EXP	EXP	SBX	EXP	EXP	EXP	EXP	EXP	EXP	SBX	EXP	EXP	1P	2P	DE	KP	EXP

Note that the color code used refers to the top three values, specifically: 🟥 **First**, 🟪 **Second**, 🟧 **Third**.

but with the difference that in this structure none of the C-TAEA + RandS combinations are among the top three. Moreover, we have not classified this structure as convergent or non-convergent because it is uncertain. Therefore, if we consider it in a middle-way, where it has not achieved convergence but is not so far from it as to be classified as clearly non-convergent, this would be consistent with the characteristics identified, since there would have been enough time for some of their partners to achieve better performances but not all of them, thus not being the worst but not being among the best either.

- Looking at the boxplots (in Figs. 2 and 3) we identified DE as the worst crossover operator, as it is always the worst or almost the worst among its partners. There were some exceptions to this observation, such as NSGA-III, NSGA-II and SMS-EMOA when using RandS in RF00009.115, RF00017.90 and RF00030.30, and RF00012.15 for CTAEA in combination with Tournament, NSGA-II, and with IGD for NSGA-III and SMS-EMOA. As before, the not achieved convergence and slope differences among those algorithms + Rands when using DE and their partners would explain this exception.

Table 6
Average HV and IGD values of selection operators. Values are expressed in scientific notation (E-03).

(a) HV

Op.	R01	R02	R03	R04	R05	R06	R07	R08	R09	R12	R13	R14	R15	R17	R18	R19	R21	R22	R25	R26	R27	R28	R29	R30
R	418.8	472.2	131.1	493.6	619.7	545.3	491.5	603.5	230.1	489.2	448.0	568.3	466.5	353.8	330.7	576.0	607.8	580.2	512.6	479.0	651.6	33.1	583.5	365.2
T	349.2	382.2	106.2	442.1	583.8	474.5	414.9	581.3	141.0	433.4	400.2	538.7	381.4	230.3	271.5	515.8	582.3	515.2	419.2	450.9	651.3	8.0	482.9	253.7
Best	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note that the color code used refers to the top three values, specifically: 🟦 **First**, 🟩 **Second**, 🟨 **Third**.

(b) IGD

Op.	R01	R02	R03	R04	R05	R06	R07	R08	R09	R12	R13	R14	R15	R17	R18	R19	R21	R22	R25	R26	R27	R28	R29	R30
R	200.4	174.5	724.9	153.9	87.5	136.4	159.1	89.9	349.2	179.2	172.9	103.8	169.9	247.3	303.0	113.0	107.0	129.8	179.8	108.1	94.5	911.8	111.3	287.1
T	325.7	276.4	766.8	195.9	113.3	196.8	239.8	105.7	453.7	226.8	212.0	125.9	259.0	347.4	384.3	164.7	124.0	182.3	265.5	128.6	95.1	971.2	202.8	394.8
Best	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note that the color code used refers to the top three values, specifically: 🟥 **First**, 🟪 **Second**, 🟧 **Third**.

Table 7
Average HD and IGD values of algorithms. Values are expressed in scientific notation (E-03).

(a) HV

Alg.	R01	R02	R03	R04	R05	R06	R07	R08	R09	R12	R13	R14	R15	R17	R18	R19	R21	R22	R25	R26	R27	R28	R29	R30
CT	148.0	226.1	2.3	363.6	511.3	327.0	279.8	514.5	177.9	368.5	300.1	454.7	242.0	244.7	108.2	379.0	506.5	424.4	305.5	369.9	614.6	22.3	333.2	261.3
N3	462.5	502.2	159.3	494.4	633.2	580.0	510.4	620.6	207.8	492.1	449.1	577.0	490.0	316.6	378.2	601.2	617.6	589.3	530.0	489.2	661.2	19.7	605.7	354.8
N2	415.2	437.6	140.9	466.8	601.7	506.7	466.1	584.4	162.4	456.5	429.8	577.0	433.8	289.2	344.5	566.2	592.3	537.2	473.1	470.6	636.7	20.2	554.2	294.3
SE	510.3	542.8	172.1	546.6	661.0	625.7	556.5	650.2	194.0	528.2	517.4	605.3	530.0	317.7	373.3	637.1	663.8	639.9	554.8	530.1	693.4	20.1	639.7	327.3
Best	SE	SE	SE	SE	SE	SE	SE	SE	N3	SE	SE	SE	SE	SE	N3	SE	SE	SE	SE	SE	SE	CT	SE	N3

Note that the color code used refers to the top three values, specifically: 🟦 **First**, 🟩 **Second**, 🟨 **Third**.

(b) IGD

Alg.	R01	R02	R03	R04	R05	R06	R07	R08	R09	R12	R13	R14	R15	R17	R18	R19	R21	R22	R25	R26	R27	R28	R29	R30
CT	601.1	440.1	994.1	249.0	158.7	309.3	348.0	150.0	438.7	272.9	285.2	183.6	381.1	358.5	568.6	262.3	169.4	243.4	354.9	172.0	123.5	933.2	327.2	385.4
N3	142.2	145.1	661.1	151.0	76.7	109.3	146.3	76.7	374.6	179.1	165.7	95.5	150.8	272.7	267.8	92.1	95.1	123.2	173.2	98.7	86.2	944.5	91.7	308.5
N2	183.2	191.7	674.0	173.3	101.8	157.6	182.1	102.5	412.9	201.4	185.4	97.9	193.1	290.1	279.4	121.5	119.6	158.4	210.6	122.1	98.6	943.9	131.9	347.6
SE	125.6	124.8	654.0	126.3	64.4	90.2	121.2	62.1	379.5	158.7	133.4	82.4	132.7	267.9	258.8	79.3	77.9	99.3	152.1	80.6	70.9	944.5	77.4	322.3
Best	SE	SE	SE	SE	SE	SE	SE	SE	N3	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE	CT	SE	N3

Note that the color code used refers to the top three values, specifically: 🟥 **First**, 🟪 **Second**, 🟧 **Third**.

To study separately which option among the studied features is the best, from Tables 3 and 4, we calculated the averages of all the alg+oper combinations in which the studied option was used (e.g., for the 1Point crossover operation, its average is calculated among C-TAEA_RandS_PM_1Point, C-TAEA_Tournament_PM_1Point, NSGA-III_RandS_PM_1Point, NSGA-III_Tournament_PM_1Point, NSGA-II_RandS_PM_1Point, NSGA-II_Tournament_PM_1Point, SMS-EMOA_ RandS_PM_1 Point and SMS-EMOA_Tournament_PM_1Point). The best three values of each RFAM structure are highlighted and colored. For each RFAM structure we determined which option achieves the best value for both HV and IGD. These results are shown in Tables 5–7. As we can see, for both HV and IGD indicators, Exp is the best performing crossover operator for more RFAM structures (21 and 16 respectively). Between the two selection operators, RandS is always the best. Finally, SMS-EMOA wins among the algorithms for HV and IGD (20 and 21 respectively). Also, from the distribution of highlighted positions we can infer that the worst would be C-TAEA, and NSGA-III is better than NSGA-II. These results are consistent with previous observations. Since NSGA-II_Tournament_PM_SBX does not use any of the best options, incorporating them would improve m2dRNAs.

As explained in Section 5.1, an objective ranking was created. It allows us to determine which alg+oper combination achieved the best performance. This ranking is shown in Table 8. The best combination is SMS-EMOA_Tournament_ PM_Exp, followed closely by SMS-EMOA_RandS_PM_Exp. Both combinations were our presumed candidates for best overall performance. The combination NSGA-II_ Tournament_PM_SBX (which, as mentioned above, corresponds to m2dRNAs) obtains 0 points, which means a tie from the 27th to the last position (48th).

6. Conclusions and future work

In this paper we have presented an experimental Analysis of Multiobjective Evolutionary Algorithms for solving the RNA inverse folding problem. We have tested four evolutionary algorithms: NSGA-II, SMS-EMOA, NSGA-III and C-TAEA; two selection operators: Random (RandS) and Tournament; six crossover operators: Simulated Binary (SBX), Differential Evolution (DE), One-Point (1Point), Two-Point (2Point), K-Point (KPoint) and Exponential (Exp), all of them combined with the same mutation operator: Polynomial (PM). The 48 possible combinations of algorithms + operators were run to solve the RFAM benchmark set, and their performances were calculated to develop a comparative study with them.

Throughout the different comparisons developed with the RFAM benchmark set we have established some observations on the studied features: SMS-EMOA is the best algorithm among those compared, and C-TAEA is the worst by far, which makes it not a good choice to solve this multiobjective problem; Looking at selection operators RandS performs better in general than Tournament; Finally, DE would be the worst among the crossover operators and Exp the best. Focusing on alg+oper combinations, the best is SMS-EMOA_Tournament_PM_Exp, followed at a short distance by SMS-EMOA_ RandS_PM_Exp. The analysis of the results has not revealed synergistic relationships between the components of the combinations. What has been observed is, within each type of combination element (algorithms, crossover operators, and selection operators), which of the tested alternatives performs best or worst. Thus, the best combinations do not appear to be so because their components interact better with each other than with others, but because they are the best in their categories (this could be summarized as “the whole is not better than the sum of its parts”). Of the three types of elements tested in the combinations, the algorithm appears to carry

Table 8
Alg+oper combinations ranking. For each RFAM structure, the calculated average final HV and IGD values were ordered from best to worst. The first to the sixth combinations obtained the following scores: 10, 8, 6, 4, 2 and 1 points. The seventh and subsequent combinations did not score any points. The scores corresponding to all the Rfam structures were added together. Both scores were summed to obtain the total score.

Combination	HV								IGD								Total
	1	2	3	4	5	6	7+	Score	1	2	3	4	5	6	7+	Score	score
SE-T-PM-EXP	5	9	2	2	0	2	4	144	5	8	3	1	2	0	5	140	284
SE-R-PM-EXP	9	3	3	0	3	1	5	139	8	4	2	0	2	0	8	128	267
SE-R-PM-SBX	2	1	2	3	2	1	13	57	3	2	3	2	1	2	11	76	133
SE-T-PM-2P	0	0	6	2	2	0	14	48	0	3	3	1	1	1	15	49	97
SE-T-PM-1P	1	1	2	2	3	2	13	46	0	2	2	4	0	3	13	47	93
SE-R-PM-2P	0	1	3	5	0	0	15	46	0	0	3	4	3	2	12	42	88
SE-T-PM-SBX	1	1	1	2	2	3	14	39	2	1	2	1	1	2	15	48	87
N3-R-PM-EXP	1	1	2	1	2	3	14	41	0	0	2	2	2	4	14	28	69
CT-R-PM-2P	1	1	0	0	2	0	20	22	2	0	0	1	0	1	20	25	47
SE-R-PM-DE	2	1	0	0	0	0	21	28	1	1	0	0	0	0	22	18	46
N3-T-PM-EXP	1	1	0	1	1	2	18	26	0	1	0	0	2	3	18	15	41
CT-R-PM-SBX	1	0	1	0	0	0	22	16	1	1	0	0	0	0	22	18	34
SE-R-PM-1P	0	1	0	1	4	4	14	24	0	0	0	0	3	0	21	6	30
N2-R-PM-DE	0	1	0	1	0	1	21	13	1	0	1	0	0	0	22	16	29
N3-R-PM-DE	0	1	1	0	0	0	22	14	0	1	1	0	0	0	22	14	28
CT-R-PM-1P	0	0	1	2	0	0	21	14	0	0	0	3	0	0	21	12	26
SE-T-PM-KP	0	0	0	0	0	1	23	1	1	0	0	2	0	1	20	19	20
SE-R-PM-KP	0	0	0	0	1	2	21	4	0	0	0	1	3	3	17	13	17
CT-R-PM-EXP	0	1	0	0	0	0	23	8	0	0	1	0	0	0	23	6	14
CT-R-PM-KP	0	0	0	0	2	1	21	5	0	0	0	0	2	1	21	5	10
SE-T-PM-DE	0	0	0	1	0	0	23	4	0	0	0	1	0	0	23	4	8
N2-T-PM-DE	0	0	0	1	0	1	22	5	0	0	0	0	0	1	23	1	6
N2-T-PM-EXP	0	0	0	0	0	0	24	0	0	0	1	0	0	0	23	6	6
N2-R-PM-EXP	0	0	0	0	0	0	24	0	0	0	0	1	0	0	23	4	4
CT-T-PM-1P	0	0	0	0	0	0	24	0	0	0	0	0	1	0	23	2	2
N3-T-PM-DE	0	0	0	0	0	0	24	0	0	0	0	0	1	0	23	2	2
CT-R-PM-DE	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
CT-T-PM-2P	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
CT-T-PM-DE	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
CT-T-PM-EXP	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
CT-T-PM-KP	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
CT-T-PM-SBX	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N3-R-PM-1P	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N3-R-PM-2P	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N3-R-PM-KP	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N3-R-PM-SBX	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N3-T-PM-1P	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N3-T-PM-2P	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N3-T-PM-KP	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N3-T-PM-SBX	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N2-R-PM-1P	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N2-R-PM-2P	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N2-R-PM-KP	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N2-R-PM-SBX	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N2-T-PM-1P	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N2-T-PM-2P	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N2-T-PM-KP	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0
N2-T-PM-SBX	0	0	0	0	0	0	24	0	0	0	0	0	0	0	24	0	0

the most weight, followed by the crossover operator, while the selection operator appears to have little importance (which explains why the best combination does not include the best operator of this type). From a practical point of view, this suggests that the greatest efforts in selecting or creating elements for this problem should be directed primarily to the algorithm, followed by the crossover operator.

To test the performance of methods for solving the RNA inverse folding problem it is important to select structures where it is possible to reach or nearly reach convergence in affordable times. This can be achieved either by simply selecting structures that have already been used in other studies and are therefore approximately known to be easy/difficult to solve, depending on what proportion of methods have achieved this and the time required, or by taking into account certain structural features, as described in Anderson-Lee et al. (2016). As is the case here for RF00009.115, RF00017.90, and RF00030.30 some exceptions to the general observations may simply be a consequence of this. Also, a large dispersion of values makes it difficult to establish comparisons among methods (here seen with RF00003.94, RF00028.1) and can lead to inaccuracies. Both observations point to the need to create boxplots and convergence plots to check whether the structures used meet these requirements, since the exceptions to the general observations may be due to a lack of sufficient time to reach convergence. As can be seen in the convergence graphs, the lines often cross, meaning that the combinations that improve HV and IGD values most quickly at the beginning of the process are not necessarily those that return the best results at the end. This means that if the results are collected before convergence has been reached, the behavior of the combinations may appear inconsistent. This effect may be seen in structures that take longer to solve, either due to their greater length or the difficulty to solve their structural components.

As future work, it would be useful to modify m2dRNAs to use the alg+oper combination and/or the individual characteristics identified here as the best (SMS-EMOA, Exp, Rands), either by changing only one of them or by using all possible combinations of these and the original ones. With these variations, we would perform a comparative study between them and with other methods similar to those performed in the articles presenting m2dRNAs and other RNA inverse folding algorithms, more focused on comparing the ability of the different methods to solve the structures of the benchmark(s) used. To compare, we would identify how many and which structures are solved, in addition to using other interesting metrics for this comparison, such as success rate and run time.

Other related lines of future work could include further studying why there is no greater difference between the number of structures solved by m2dRNAs and the best combination, introducing other features such as different chromosome encodings or mutation operators, testing other performance indicators, or modifying the definition of the MOOP. In addition, other potential directions could include improving algorithms, optimizing operators, and using additional benchmark sets.

CRediT authorship contribution statement

Álvaro Rubio-Largo: Writing – review & editing, Visualization, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Nuria Lozano-García:** Writing – original draft, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **José M. Granado-Criado:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization.

Funding

This work was partially funded by the Junta de Extremadura (Spain) and the ERDF (European Regional Development Fund, EU), under the contract IB20047 (iARN project). This work was also partially supported by the MCIN (Ministry of Science and Innovation, Spain), the AEI (State Research Agency, Spain), and the ERDF (European Regional Development Fund, EU), under the project PID2022-137275NA-I00 funded by MCIN/AEI/10.13039/501100011033/FEDER, EU (X-BIO project).

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.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.engappai.2025.111189.

Data availability

Data will be made available on request.

References

Afnan, M.R., Ashraf, N.B., Islam, M.R., 2020. Multiobjective computational RNA design using chemical reaction optimization. In: 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP). IEEE, pp. 1–6. http://dx.doi.org/10.1109/icccsp49186.2020.9315262.

Anderson-Lee, J., Fisker, E., Kosaraju, V., Wu, M., Kong, J., Lee, J., Lee, M., Zada, M., Treuille, A., Das, R., 2016. Principles for predicting RNA secondary structure design difficulty. J. Mol. Biol. 428 (5, Part A), 748–757. http://dx.doi.org/10.1016/j.jmb.2015.11.013, Challenges in RNA structural modeling and design, URL https://www.sciencedirect.com/science/article/pii/S0022283615006567.

Andronescu, M., Fejes, A.P., Hutter, F., Hoos, H.H., Condon, A., 2004. A new algorithm for RNA secondary structure design. J. Mol. Biol. 336 (3), 607–624. http://dx.doi.org/10.1016/j.jmb.2003.12.041.

Bellaousov, S., Kayedkhordeh, M., Peterson, R.J., Mathews, D.H., 2018. Accelerated RNA secondary structure design using preselected sequences for helices and loops. RNA 24 (11), 1555–1567. http://dx.doi.org/10.1261/rna.066324.118.

Beume, N., Naujoks, B., Emmerich, M., 2007. SMS-EMOA: Multiobjective selection based on dominated hypervolume. European J. Oper. Res. 181 (3), 1653–1669. http://dx.doi.org/10.1016/j.ejor.2006.08.008.

Busch, A., Backofen, R., 2006. INFO-RNA - a fast approach to inverse RNA folding. Bioinformatics 22 (15), 1823–1831. http://dx.doi.org/10.1093/bioinformatics/btl194.

Cazenave, T., Fournier, T., 2021. Monte Carlo inverse folding. In: Cazenave, T., Teytaud, O., Winands, M.H.M. (Eds.), Monte Carlo Search. Springer International Publishing, Cham, pp. 84–99. http://dx.doi.org/10.1007/978-3-030-89453-5_7.

Churkin, A., Retwitzer, M.D., Reinharz, V., Ponty, Y., Waldispühl, J., Barash, D., 2018. Design of RNAs: comparing programs for inverse RNA folding. Brief. Bioinform. 19 (2), 350–358. http://dx.doi.org/10.1093/bib/bbw120.

Coello, C.A.C., Sierra, M.R., 2004. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In: MICAI 2004: Advances in Artificial Intelligence. Springer Berlin Heidelberg, pp. 688–697. http://dx.doi.org/10.1007/978-3-540-24694-7_71.

Collette, Y., Siarry, P., 2004. Multiobjective Optimization. Springer Berlin Heidelberg, http://dx.doi.org/10.1007/978-3-662-08883-8.

Das, I., Dennis, J.E., 1998. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J. Optim. 8 (3), 631–657. http://dx.doi.org/10.1137/s1052623496307510.

Deb, K., Jain, H., 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. IEEE Trans. Evol. Comput. 18 (4), 577–601. http://dx.doi.org/10.1109/tevc.2013.2281535.

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

Eastman, P., Shi, J., Ramsundar, B., Pande, V.S., 2018. Solving the RNA design problem with reinforcement learning. In: Chen, S.-J. (Ed.), PLOS Comput. Biology 14 (6), e1006176. http://dx.doi.org/10.1371/journal.pcbi.1006176.

Erhan, H.E., Sav, S., Kalashnikov, S., Tsang, H.H., 2016. Examining the annealing schedules for RNA design algorithm. In: 2016 IEEE Congress on Evolutionary Computation. CEC, IEEE, pp. 1295–1302. http://dx.doi.org/10.1109/cec.2016.7743937.

Esmaili-Taheri, A., Ganjtabesh, M., 2015. ERD: a fast and reliable tool for RNA design including constraints. BMC Bioinform. 16 (1), 20. http://dx.doi.org/10.1186/s12859-014-0444-5.

Esmaili-Taheri, A., Ganjtabesh, M., Mohammad-Noori, M., 2014. Evolutionary solution for the RNA design problem. Bioinformatics 30 (9), 1250–1258. http://dx.doi.org/10.1093/bioinformatics/btu001.

Fleischer, M., 2003. The measure of Pareto optima applications to multi-objective metaheuristics. In: Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 519–533. http://dx.doi.org/10.1007/3-540-36970-8_37.

García-Martín, J.A., Clote, P., Dotú, I., 2013. RNAiFold: a constraint programming algorithm for RNA inverse folding and molecular design. J. Bioinform. Comput. Biology 11 (02), 1350001. http://dx.doi.org/10.1142/s0219720013500017.

García-Martín, J.A., Dotú, I., Clote, P., 2015. RNAiFold 2.0: a web server and software to design custom and Rfam-based RNA molecules. Nucleic Acids Res. 43 (W1), W513–W521. http://dx.doi.org/10.1093/nar/gkv460.

Hammer, S., Wang, W., Will, S., Ponty, Y., 2019. Fixed-parameter tractable sampling for RNA design with multiple target structures. BMC Bioinform. 20 (1), 209. http://dx.doi.org/10.1186/s12859-019-2784-7.

Hampson, D.J.D., Tsang, H.H., 2018. Incorporating dynamic exploration strategy for RNA design. In: 2018 IEEE Symposium Series on Computational Intelligence. SSCI, IEEE, pp. 1041–1048. http://dx.doi.org/10.1109/ssci.2018.8628681.

Hofacker, I., 2003. Vienna RNA secondary structure server. Nucleic Acids Res. 31 (13), 3429–3431. http://dx.doi.org/10.1093/nar/gkg599.

Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, L.S., Tacker, M., Schuster, P., 1994. Fast folding and comparison of RNA secondary structures. Monatshefte Für Chem. Chem. Mon. 125 (2), 167–188. http://dx.doi.org/10.1007/bf00818163.

Hombach, S., Kretz, M., 2016. Non-Coding RNAs in Colorectal Cancer. Springer International Publishing, pp. 3–17. http://dx.doi.org/10.1007/978-3-319-42059-2_1.

Huang, L., Zhang, H., Deng, D., Zhao, K., Liu, K., Hendrix, D.A., Mathews, D.H., 2019. LinearFold: linear-time approximate RNA folding by 5'-to-3' dynamic programming and beam search. Bioinformatics 35 (14), i295–i304. http://dx.doi.org/10.1093/bioinformatics/btz375.

Kerpedjiev, P., Hammer, S., Hofacker, I.L., 2015. Forna (force-directed RNA): Simple and effective online RNA secondary structure diagrams. Bioinformatics 31 (20), 3377–3379. http://dx.doi.org/10.1093/bioinformatics/btv372.

Kleinkauf, R., Houwaart, T., Backofen, R., Mann, M., 2015a. antaRNA – multi-objective inverse folding of pseudoknot RNA using ant-colony optimization. BMC Bioinform. 16 (1), 389. http://dx.doi.org/10.1186/s12859-015-0815-6.

Kleinkauf, R., Mann, M., Backofen, R., 2015b. antaRNA: ant colony-based RNA sequence design. Bioinformatics 31 (19), 3114–3121. http://dx.doi.org/10.1093/bioinformatics/btv319.

Koodli, R.V., Keep, B., Coppess, K.R., Portela, F., Das, R., 2019. EternaBrain: Automated RNA design through move sets and strategies from an internet-scale RNA videogame. In: Chen, S.-J. (Ed.), PLOS Comput. Biology 15 (6), e1007059. http://dx.doi.org/10.1371/journal.pcbi.1007059.

Lee, J., Kladwang, W., Lee, M., Cantu, D., Azizyan, M., Kim, H., Limpaecher, A., Gaikwad, S., Yoon, S., Treuille, A., Das, R., 2014. RNA design rules from a massive open laboratory. Proc. Natl. Acad. Sci. 111 (6), 2122–2127. http://dx.doi.org/10.1073/pnas.1313039111.

Li, K., Chen, R., Fu, G., Yao, X., 2019. Two-archive evolutionary algorithm for constrained multiobjective optimization. IEEE Trans. Evol. Comput. 23 (2), 303–315. http://dx.doi.org/10.1109/tevc.2018.2855411.

Lipowski, A., Lipowska, D., 2012. Roulette-wheel selection via stochastic acceptance. Phys. A 391 (6), 2193–2196. http://dx.doi.org/10.1016/j.physa.2011.12.004.

Lorenz, R., Bernhart, S.H., zu Siederdissen, C.H., Tafer, H., Flamm, C., Stadler, P.F., Hofacker, I.L., 2011. ViennaRNA package 2.0. Algorithms Mol. Biology 6 (1), http://dx.doi.org/10.1186/1748-7188-6-26.

Lorenz, R., Wolfinger, M.T., Tanzer, A., Hofacker, I.L., 2016. Predicting RNA secondary structures from sequence and probing data. Methods 103, 86–98. http://dx.doi.org/10.1016/j.ymeth.2016.04.004, Advances in RNA structure determination, URL https://www.sciencedirect.com/science/article/pii/S1046202316300743.

Lozano-García, N., Rubio-Largo, Á., Granado-Criado, J.M., 2024. A simple yet effective greedy evolutionary strategy for RNA design. IEEE Trans. Evol. Comput. 1. http://dx.doi.org/10.1109/TEVC.2024.3461509.

Lyngsø, R.B., 2008a. RNA secondary structure Boltzmann distribution. In: Encyclopedia of Algorithms. Springer US, pp. 777–779. http://dx.doi.org/10.1007/978-0-387-30162-4_345.

Lyngsø, R.B., 2008b. RNA secondary structure prediction by minimum free energy. In: Encyclopedia of Algorithms. Springer US, pp. 782–785. http://dx.doi.org/10.1007/978-0-387-30162-4_347.

Lyngsø, R.B., Anderson, J.W.J., Sizikova, E., Badugu, A., Hyland, T., Hein, J., 2012. FRNAkenstein: multiple target inverse RNA folding. BMC Bioinform. 13 (1), 260. http://dx.doi.org/10.1186/1471-2105-13-260.

Mandelbrot, B.B., 1963. The variation of certain speculative prices. J. Bus. 36, 371–418.

Matthies, M.C., Bienert, S., Torda, A.E., 2012. Dynamics in sequence space for RNA secondary structure design. J. Chem. Theory Comput. 8 (10), 3663–3670. http://dx.doi.org/10.1021/ct300267j.

McBride, R., Tsang, H.H., 2020. Examination of annealing schedules for RNA design. In: 2020 IEEE Congress on Evolutionary Computation. CEC, IEEE, pp. 1–8. http://dx.doi.org/10.1109/cec48606.2020.9185702.

McBride, R., Tsang, H.H., 2021. SIMARD-LinearFold: Long sequence RNA design with simulated annealing. In: 2021 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 2234–2241. http://dx.doi.org/10.1109/cec45853.2021.9504978.

McCaskill, J.S., 1990. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. Biopolymers 29 (6–7), 1105–1119. http://dx.doi.org/10.1002/bip.360290621, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/bip.360290621.

Merleau, N.S.C., Smerlak, M., 2021. A simple evolutionary algorithm guided by local mutations for an efficient RNA design. In: Proceedings of the Genetic and Evolutionary Computation Conference. GECCO'21, ACM, pp. 1027–1034. http://dx.doi.org/10.1145/3449639.3459280.

Merleau, N.S.C., Smerlak, M., 2022. aRNAque: an evolutionary algorithm for inverse pseudoknotted RNA folding inspired by Lévy flights. BMC Bioinform. 23 (1), 335. http://dx.doi.org/10.1186/s12859-022-04866-w.

Meyer, S., Chappell, J., Sankar, S., Chew, R., Lucks, J.B., 2015. Improving fold activation of small transcription activating RNAs (STARs) with rational RNA engineering strategies. Biotechnol. Bioeng. 113 (1), 216–225. http://dx.doi.org/10.1002/bit.25693.

Minuesa, G., Alsina, C., García-Martín, J.A., Oliveros, J.C., Dotú, I., 2021. MoiRNAiFold: a novel tool for complex in silico RNA design. Nucleic Acids Res. 49 (9), 4934–4943. http://dx.doi.org/10.1093/nar/gkab331.

Newman, M., 2005. Power laws, Pareto distributions and Zipf's law. Contemp. Phys. 46 (5), 323–351. http://dx.doi.org/10.1080/00107510500052444.

Qiu, M., Khisamutdinov, E., Zhao, Z., Pan, C., Choi, J.-W., Leontis, N.B., Guo, P., 2013. RNA nanotechnology for computer design and in vivo computation. Philos. Trans. R. Soc. A: Math, Phys Eng Sci. 371 (2000), 20120310. http://dx.doi.org/10.1098/rsta.2012.0310.

Rosenberg, J.M., Seeman, N.C., Day, R.O., Rich, A., 1976. RNA double-helical fragments at atomic resolution: II. The crystal structure of sodium guanylyl-3',5'-cytidine nonahydrate. J. Mol. Biol. 104 (1), 145–167. http://dx.doi.org/10.1016/0022-2836(76)90006-1.

Rubio-Largo, Á., Escobar-Encinas, L., Lozano-García, N., Granado-Criado, J.M., 2024. Evolutionary strategy to enhance an RNA design tool performance. IEEE Access 12, 15582–15593. http://dx.doi.org/10.1109/ACCESS.2024.3358426.

Rubio-Largo, Á., Lozano-García, N., Granado-Criado, J.M., Vega-Rodríguez, M.A., 2023. Solving the RNA inverse folding problem through target structure decomposition and multiobjective evolutionary computation. Appl. Soft. Comput. 147, 110779. http://dx.doi.org/10.1016/j.asoc.2023.110779, URL https://www.sciencedirect.com/science/article/pii/S1568494623007974.

Rubio-Largo, Á., Vanneschi, L., Castelli, M., Vega-Rodríguez, M.A., 2019. Multiobjective metaheuristic to design RNA sequences. IEEE Trans. Evol. Comput. 23 (1), 156–169. http://dx.doi.org/10.1109/TEVC.2018.2844116.

Runge, F., Stoll, D., Falkner, S., Hutter, F., 2019. Learning to design RNA. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=ByfyHh05tQ.

Sav, S., Hampson, D.J.D., Tsang, H.H., 2016. SIMARD: A simulated annealing based RNA design algorithm with quality pre-selection strategies. In: 2016 IEEE Symposium Series on Computational Intelligence. SSCI, IEEE, pp. 1–8. http://dx.doi.org/10.1109/ssci.2016.7849957.

Schnall-Levin, M., Chindelevitch, L., Berger, B., 2008. Inverting the viterbi algorithm: an abstract framework for structure design. In: Proceedings of the 25th International Conference on Machine Learning. ICML '08, Association for Computing Machinery, New York, NY, USA, pp. 904–911. http://dx.doi.org/10.1145/1390156.1390270.

Seeman, N.C., Rosenberg, J.M., Suddath, F., Kim, J.J.P., Rich, A., 1976. RNA double-helical fragments at atomic resolution: I. The crystal and molecular structure of sodium adenylyl-3',5'-uridine hexahydrate. J. Mol. Biol. 104 (1), 109–144. http://dx.doi.org/10.1016/0022-2836(76)90005-x.

Shi, J., Das, R., Pande, V.S., 2018. SentRNA: Improving computational RNA design by incorporating a prior of human design strategies. arXiv:1803.03146.

Slowik, A., Kwasnicka, H., 2020. Evolutionary algorithms and their applications to engineering problems. Neural Comput. Appl. 32, 12363–12379. http://dx.doi.org/10.1007/s00521-020-04832-8.

Storn, R., Price, K., 1997. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11 (4), 341–359. http://dx.doi.org/10.1023/a:1008202821328.

Taneda, A., 2010. MODENA: a multi-objective RNA inverse folding. Adv. Appl. Bioinform. Chem. 2011 (4), 1–12. http://dx.doi.org/10.2147/aabc.s14335.

Taneda, A., 2012. Multi-objective genetic algorithm for pseudoknotted RNA sequence design. Front. Genet. 3, http://dx.doi.org/10.3389/fgene.2012.00036.

Taneda, A., 2015. Multi-objective optimization for RNA design with multiple target secondary structures. BMC Bioinform. 16 (1), 280. http://dx.doi.org/10.1186/s12859-015-0706-x.

Tinoco, I., Bustamante, C., 1999. How RNA folds. J. Mol. Biol. 293 (2), 271–281. http://dx.doi.org/10.1006/jmbi.1999.3001.

Varani, G., McClain, W.H., 2000. The G·U wobble base pair. EMBO Rep. 1 (1), 18–23. http://dx.doi.org/10.1093/embo-reports/kvd001.

Ward, M., Courtney, E., Rivas, E., 2023. Fitness functions for RNA structure design. Nucleic Acids Res. 51 (7), e40. http://dx.doi.org/10.1093/nar/gkad097, arXiv:https://academic.oup.com/nar/article-pdf/51/7/e40/54399611/gkad097.pdf.

Wilm, A., Higgins, D.G., Notredame, C., 2008. R-coffee: a method for multiple alignment of non-coding RNA. Nucleic Acids Res. 36 (9), e52. http://dx.doi.org/10.1093/nar/gkn174.

Yan, Z., Hamilton, W.L., Blanchette, M., 2021. Neural representation and generation for RNA secondary structures. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=snOgiCYZgJ7.

Yang, X., Yoshizoe, K., Taneda, A., Tsuda, K., 2017. RNA inverse folding using Monte Carlo tree search. BMC Bioinform. 18 (1), 468. http://dx.doi.org/10.1186/s12859-017-1882-7.

Yao, H.-T., Waldispühl, J., Ponty, Y., Will, S., 2021. Taming disruptive base pairs to reconcile positive and negative structural design of RNA. In: RECOMB 2021 - 25th International Conference on Research in Computational Molecular Biology. Padova, France, URL https://hal.inria.fr/hal-02987566.

Zadeh, J.N., Steenberg, C.D., Bois, J.S., Wolfe, B.R., Pierce, M.B., Khan, A.R., Dirks, R.M., Pierce, N.A., 2010a. NUPACK: Analysis and design of nucleic acid systems. J. Comput. Chem. 32 (1), 170–173. http://dx.doi.org/10.1002/jcc.21596.

Zadeh, J.N., Wolfe, B.R., Pierce, N.A., 2010b. Nucleic acid sequence design via efficient ensemble defect optimization. J. Comput. Chem. 32 (3), 439–452. http://dx.doi.org/10.1002/jcc.21633.

Zhou, T., Dai, N., Li, S., Ward, M., Mathews, D.H., Huang, L., 2023. RNA design via structure-aware multifrontier ensemble optimization. Bioinformatics 39 (Supplement_1), i563–i571. http://dx.doi.org/10.1093/bioinformatics/btad252.

Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. 3 (4), 257–271. http://dx.doi.org/10.1109/4235.797969.