Permutation-based Estimation of Distribution Algorithms for the RNA folding problem*

Julen Benages[1], Leticia Hernando[1] and Roberto Santana[1,*]

[1]University of the Basque Country (UPV/EHU)

Abstract

Ribonucleic acid (RNA) sequences fold into 3D structures that play diverse roles in living organisms. In this paper, we introduce a new permutation-based representation for the RNA secondary structure prediction problem and evaluate the use of Estimation of Distribution Algorithms (EDAs) for this problem. We address the search for secondary structures as a bi-objective problem using a decomposition-based approach adapted to EDAs. Our goal is twofold: first, to investigate the difficulty of the RNA secondary structure prediction search space, and second, to evaluate different types of permutation-based probabilistic models on a relevant and challenging problem. Specifically, we evaluate EDAs based on various permutation-based models using a subset of the Rfam database, which provides experimentally determined RNA secondary structures for diverse RNA families. Our results show that permutation-based EDAs can lead the search to RNA secondary structures of minimal energy and that the choice of the model plays a relevant role in search efficiency.

Keywords

Estimation of Distribution Algorithms, RNA secondary structure, combinatorial optimisation

1. Introduction

Finding the 3D structure of RNA sequences is important to determine their functionality, a critical task in domains such as vaccine design, analysis of biological process and other. Usually, an intermediate step is the identification of the RNA's secondary structure. A large variety of computational approaches have been used to determine the secondary RNA structures. Dynamic programming approaches [1] can produce exact solutions but their complexity significantly increases with the sequence dimensions. Prediction methods based on machine learning algorithms [2], particularly neural networks [3, 4], have been increasingly proposed. While they can predict good initial folds, they may not be sufficiently accurate to predict structures with pseudoknots. Metaheuristics approaches such as evolutionary algorithms [5, 6] can not guarantee optimality but they can provide a variety of different solutions that approximate the best structure and find structures with pseudoknots.

Since secondary structures can be characterised by different properties (e.g., energy, folding probability, stability), there are many scenarios in which it is convenient to have a set of RNA secondary structure candidates rather than a single one. Following this rationale, in this paper we approach the RNA secondary structure prediction problem as a bi-objective problem. To follow this path, a critical question is the choice of solution representation. One representational approach previously proposed is the permutation representation of stems [5, 6], in which all possible stems of the RNA sequence are first computed, and then a permutation defines the order in which stems are progressively added to build the secondary structure, while ignoring stems that violate structural constraints with those already present in the solution.

One limitation of the stem representation is that the number of stems can rapidly increase with the RNA sequence size reaching thousands of stems, therefore making the search in a space of permutations infeasible. We introduce a more direct base-pair permutation where permutations are defined in the sequence size n and permutations indicate a ranking of preferences at the time of considering

ITAT'26: Information Technologies – Applications and Theory, September 25–29, 2026, Biele Karpaty, Slovakia

*Corresponding author.

jbenages001@ikasle.ehu.eus (J. Benages); leticia.hernando@ehu.eus (L. Hernando); roberto.santana@ehu.eus (R. Santana)

0009-0004-1302-9707 (J. Benages); 0000-0003-2828-0748 (L. Hernando); 0000-0002-1005-8535 (R. Santana)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-09

CEUR-WS.org/Vol-4323/paper14.pdf

pair-based connections. Using this representation, we then approach the search using Estimation of Distribution Algorithms (EDAs) [7, 8] whose models are defined on the permutation representation [9]. The bi-objective search is framed using the multi-objective decomposition approach [10] in which a probabilistic model is learned for each problem [11]. To evaluate the impact of the characteristics of the specific class of probabilistic model, we evaluate six different permutation-based EDAs in their ability to find secondary structures of minimal energy.

The paper is organised as follows: The next section provides the necessary background on multi-objective optimisation and RNA folding. Section 3 introduces the two permutation representations used in the paper. Section 4 presents the different EDAs used in the paper. Work related to our research is analysed in Section 5. The experimental framework and results from the experiments are presented in Section 6. The final remarks of the paper and directions for future work are presented in Section 7.

2. Multi-objective optimisation in the permutation domain

Let S_n denote the symmetric group of all $n!$ permutations of the set $\{1, \dots, n\}$. A permutation $\sigma \in S_n$ is a bijection $\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$, where $\sigma(i) = j$ means that element j is placed at position i. We represent a population as a set of permutations $\{\sigma^1, \dots, \sigma^N\} \subset S_n$, where N is the population size. Similarly, σ_i^l denotes the value at position i of the l-th permutation in the population.

A general multi-objective optimization problem (MOP) can be defined as follows [12]:

$$\text{minimise} \quad F(\sigma) = \big(f_1(\sigma), \dots, f_m(\sigma)\big); \qquad \text{subject to } \sigma \in S_n \tag{1}$$

where $\sigma \in S_n$ is the decision variable, $F : S_n \to \mathbb{R}^m$ consists of m objective functions and $\mathbb{R}^m$ is the objective space.

Pareto optimality is defined as follows. Let $\sigma, \tau \in S_n$. σ is said to dominate τ if and only if $f_i(\sigma) \leq f_i(\tau)$ for all $i \in \{1, \dots, m\}$ and $f_i(\sigma) < f_i(\tau)$ for at least one i. A solution $\sigma^* \in S_n$ is called Pareto optimal if there is no other $\sigma \in S_n$ which dominates σ^*. The set of all Pareto optimal solutions is called the Pareto set (PS) and the solutions mapped in the objective space are called the Pareto front (PF), i.e., $PF = \{F(\sigma) \mid \sigma \in PS\}$.

2.1. Objective functions

Let $\mathcal{S}(\mathbf{x})$ denote the set of all valid secondary structures of $\mathbf{x}$, that is, the sets of base pairs in which every position is paired at most once, each pair joins complementary (Watson–Crick or wobble) nucleotides, the pairs are properly nested (no pseudoknots), and a minimum hairpin-loop size is respected. The Turner nearest-neighbor model assigns to each structure $S \in \mathcal{S}(\mathbf{x})$ a free energy $\Delta G(S, \mathbf{x})$ obtained by summing tabulated contributions of its loops (stacked pairs, hairpins, bulges, internal loops, and multiloops). The minimum free energy (MFE) structure is the single valid structure that minimises this energy,

$$s(\mathbf{x}) = \mathscr{F}(\mathbf{x}) = \operatorname*{arg\,min}_{S \in \mathcal{S}(\mathbf{x})} \Delta G(S, \mathbf{x}).$$

The Vienna RNA package computes $s(\mathbf{x})$ exactly with the cubic-time Zuker dynamic-programming algorithm [13]; the result is therefore a feasible secondary structure.

We use a bi-objective approach for searching the space of RNA structures. The first objective is the thermodynamic free energy of the decoded structure under the nearest-neighbor model:

$$f_1(\sigma) = \Delta G\big(\hat{s}(\sigma),\, \mathbf{x}\big), \tag{2}$$

Values are in kcal/mol; lower (more negative) values correspond to more thermodynamically stable structures, with the minimum-free-energy (MFE) structure as the global optimum.

The second objective function is the ensemble defect [14] (NED) that quantifies the expected number of incorrectly paired nucleotides when a structure is drawn at random from a Boltzmann ensemble:

$$f_2(\sigma) = \frac{1}{n}\left(n \; - \; \sum_{i=1}^{n}\sum_{j=0}^{n} \mathbb{1}[s^*_{ij}]\, p_{ij}(\mathbf{x})\right), \tag{3}$$

where $s^* = \hat{s}(\sigma)$ is the structure decoded from σ, $p_{ij}(\mathbf{x})$ is the base-pair probability between positions i and j (with $j = 0$ denoting the unpaired state), and $\mathbb{1}[s^*_{ij}]$ is 1 if position i is paired with position j in s^* (for $j \geq 1$), or if position i is unpaired in s^* (for $j = 0$). Values lie in $[0, 1]$; $f_2(\sigma) = 0$ means that, on average, every nucleotide adopts its target pairing state across the Boltzmann ensemble, and f_2 is minimised, exactly as f_1. NED returns the unnormalised count, dividing it by n gives $f_2 \in [0, 1]$ directly.

We use these f_1 and f_2, implemented via the ViennaRNA package [15], to search for RNA secondary structures. The structure that minimises the energy is not necessarily the one that minimises NED. Therefore, there is a trade of between these two metrics and it is convenient to keep non-dominated candidate secondary structures according to these two objectives.

There are other ways to evaluate the structure. As an additional objective, that is not used for evolution but only to evaluate the quality of the found secondary structures we use the distance to the MFE structure $s_{\mathrm{MFE}}(\mathbf{x})$:

$$f_d(\sigma) = d_{\mathrm{BP}}\big(\hat{s}(\sigma),\, s_{\mathrm{MFE}}(\mathbf{x})\big) = \big|\mathscr{B}(\hat{s}(\sigma)) \vartriangle \mathscr{B}(s_{\mathrm{MFE}}(\mathbf{x}))\big|, \tag{4}$$

where $\mathscr{B}(s)$ is the set of (i, j) base pairs encoded by structure s and $\vartriangle$ denotes the symmetric difference. Intuitively, $f_d(\sigma)$ counts the number of base pairs by which the candidate structure differs from the MFE prediction.

Let $\hat{s}(\sigma)$ denote the secondary structure decoded from permutation $\sigma \in S_n$ for a given RNA sequence $\mathbf{x}$ of length n (the decoding procedures are detailed in Section 3).

3. Base-pair and stem permutation representations

3.1. Base-pair representation

In the base-pair permutation representation, a candidate secondary structure for an RNA sequence of length n is encoded as a permutation σ of $\{0, \ldots, n-1\}$ (here S_n acts on zero-based indices to match sequence indexing). In an exact (lossless) encoding σ is an involution: $\sigma(i) = j$ means nucleotide i is paired with nucleotide j, and $\sigma(j) = i$ by symmetry; $\sigma(i) = i$ means nucleotide i is unpaired. Because standard evolutionary operators can generate arbitrary permutations rather than involutions, a deterministic greedy decoder maps every permutation to a valid pseudoknot-free structure:

1. Compute the rank vector r from σ: $r(i)$ is the position at which element i appears in σ.
2. Enumerate all candidate Watson-Crick or GU-wobble pairs (i, j) with $i < j$ such that $\mathrm{seq}[i]$ and $\mathrm{seq}[j]$ belong to $\{\mathrm{A{-}U}, \mathrm{U{-}A}, \mathrm{G{-}C}, \mathrm{C{-}G}, \mathrm{G{-}U}, \mathrm{U{-}G}\}$ *and* $j - i > \tau$, where τ is the minimum hairpin-loop size.
3. Sort candidates by $r(i)$ in ascending order (lower rank = higher priority).
4. Greedily include pairs in priority order, skipping any pair whose addition would create a crossing with a previously accepted pair.

By construction, this decoder guarantees that every decoded structure satisfies ViennaRNA's minimum loop-length requirement. In addition, a mutation step is added that allows to change two variables that are unpaired ($\sigma(i) = i$ and $\sigma(j) = j$) to be paired, and similarly, paired variables can become un-paired. In addition, the initial population is generated using the stem-based initialisation described below: a random stem permutation is decoded, and the resulting base-pair involution is used as the starting permutation.

3.2. Stem representation and population initialisation

The stem representation encodes a candidate structure as an ordering of maximal helical stems. A maximal stem anchored at the outermost pair (s_5, s_3) with $s_5 < s_3$ consists of ℓ consecutive Watson-Crick/wobble pairs:

$$(s_5,\ s_3),\ (s_5{+}1,\ s_3{-}1),\ \ldots,\ (s_5{+}\ell{-}1,\ s_3{-}\ell{+}1).$$

A stem is maximal if it cannot be extended outward (i.e. positions $s_5 - 1$ and $s_3 + 1$ do not form a valid pair, or a sequence boundary is reached). Let M denote the total number of maximal stems; the search space is then S_M. A permutation $\sigma \in S_M$ is decoded by reading stem indices in the order given by σ and greedily adding each stem's base pairs provided both nucleotides remain unpaired and no pseudoknot would be introduced, following the approach of [16, 5].

Since the number of maximal helical stems can be significantly high with respect to the sequence length, a search based on this representation becomes infeasible. However, we use the computation of stems as a way to initialise the population in the following way:

To build one individual of the base-pair-representation population, a permutation of the M stem indices, $\pi \in S_M$, is drawn uniformly at random and decoded exactly as described above: stems are visited in the order given by π and each stem's ℓ base pairs are accepted together, as a block, provided every position they touch is still unpaired and none of them would cross an already-accepted pair. The resulting set of base pairs is then re-encoded, losslessly, as a base-pair involution $\sigma \in S_n$ by setting $\sigma(i) = j$ for every accepted pair (i, j) and $\sigma(p) = p$ for every position p left unpaired; this σ is the individual's starting permutation. Repeating this procedure independently for every individual yields the initial population which is them evolved by the EDA.

The rationale behind using this stem-based initialisation is to seed the base-pair search with solutions which are already composed of complete, contiguous helices rather than isolated pairs assembled one at a time. Since a stem's ℓ pairs are accepted or rejected as a single block, the initial population automatically respects the geometric regularity of real secondary structures (stacked, unbroken runs of pairs). It would be more difficult for purely position-by-position random or compatibility-biased construction to reach this type of structure by chance.

4. Bi-objective EDAs for permutation optimisation

The Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) [10] decomposes a MOP into a number of scalar optimisation subproblems and optimises them simultaneously. Each subproblem i is associated to a weight vector λ^i that represents the contribution of each objective to the scalarized objective. A number of decomposition approaches have been proposed for MOEA/D. In this paper we focus on bi-objective problems and implement the EDAs variants that follow the MOEA/D framework using the Tchebycheff (Chebyshev) scalarisation approach.

A characteristic feature of MOEA/D is the way in which the neighborhood relation among the subproblems is defined. The neighborhood for each subproblem i is defined according to the Euclidean distance of its weight vectors. The set of weight vectors should be such that the optimal solution of their corresponding subproblems is uniformly distributed along the PF. In this paper, the method used by [10] to set the weight vectors is applied. The relationship of the neighbors is used for the selection of parents solutions and the replacement of old solutions (also called update mechanism).

The size of the neighborhood for selection and replacement plays a vital role in MOEA/D to exchange information among the subproblems. Given the set of neighbors, solutions are evaluated according to the scalarised function and the top half solutions are used to learn the permutation-based probabilistic model. In addition, an external population is used to maintain all Pareto optimal solutions found during the search. The multi-objective decomposition-based EDAs that we describe in the following section share the same structure as MOEA/D except the way models are learned and solutions are sampled.

4.1. Permutation-based EDAs

One of our goals is to investigate how the choice of probabilistic models defined on permutations impact search efficiency. While permutation-based models have been applied to classical combinatorial problems, their application to other real-world problems is scarse. We investigate six permutation-based EDAs: MEDA-MK, MEDA-PL, MEDA-NHM, MEDA-Kendall, MEDA-GMCayley and MEDA-GMKendall.

MEDA-MK uses Kernels of Mallows models [11]. In Mallows-based distributions [9], given a set of permutations, the learning process consists of estimating the consensus (central) permutation (σ_0) and the spread parameter (θ). These parameters are in turn computed according to a pre-defined distance among permutations, in this paper the Cayley distance. MEDA-MK does not learn the model from a population sample. Instead it uses a Mallows Kernel, a fixed, pre-parameterised distribution centred on the current solution of each subproblem. For subproblem k, the centre σ_0 is the current best solution found so far for that subproblem. The concentration parameter θ is passed at construction and fixed (theta=1.0 by default), not estimated from data.

MEDA-PL learns a Plackett-Luce ranking model [17] from the top half of each subproblem's neighbourhood: every value $1, \dots, n$ is assigned a latent weight, estimated by maximum likelihood via the Minorisation-Maximisation algorithm, and a new permutation is sampled by repeatedly drawing, without replacement, a value with probability proportional to its estimated weight among those not yet placed. Unlike MEDA-MK, both the n weights and the resulting distribution are re-estimated from the selected population at every generation, rather than being fixed a priori.

MEDA-NHM uses a Node Histogram Model [18]: for every absolute position of the permutation, a histogram over which value appears there is tabulated across the selected population, with a smoothing/prior term (controlled by a β-ratio parameter) to avoid zero probabilities, and a new permutation is sampled position by position, drawing from each position's histogram and renormalising over the values not yet used so that the sample remains a valid permutation.

MEDA-Kendall, unlike MEDA-MK, learns a proper Mallows distribution under the Kendall-τ distance [19, 9] from the selected population at every generation: both the consensus permutation σ_0 and a single, global spread parameter θ are estimated from data, rather than θ being fixed and σ_0 simply copied from the current best solution as in MEDA-MK; new candidates are then sampled from the resulting Mallows-Kendall distribution.

MEDA-GMCayley and MEDA-GMKendall both use the Generalised Mallows model [19], which replaces the single spread parameter θ of the standard Mallows model with a vector of $n-1$ position-specific spread parameters, letting the learned distribution capture that some rank positions vary far less across the selected population than others. The two algorithms differ only in the permutation distance the model is built on: Cayley distance (minimum number of transpositions separating two permutations) for MEDA-GMCayley, and Kendall-τ distance (number of pairwise inversions) for MEDA-GMKendall.

5. Related work

While our focus is on the investigation of MEDAs, we review previous works that applied optimization algorithms using permutation-based representations to the RNA folding as well as applications to related problems.

5.1. EA with permutation representations for RNA Folding

The permutation-based genetic algorithm (GA) was introduced in [16], encoding the RNA secondary structure as an integer permutation of all possible helices. In the same work, different genetic operators are investigated, revealing that Cycle Crossover (CX), which preserves the absolute position of elements, generally outperforms other permutation operators like PMX and OX in the RNA domain. Subsequent research formalised the RnaPredict algorithm, an evolutionary approach capable of directly incorporating sophisticated thermodynamic models during the optimisation process [20]. RnaPredict accurately evaluates the free energy contribution of stacked base pairs and terminal mismatches.

The permutation representation has also been successfully integrated into simulated annealing (SA) frameworks, most notably in the SARNA-Predict algorithm [5] which leverages a swap mutation operator on the permutation of helices to perturb candidate structures in a controlled manner, while relying on distinct annealing schedules to systematically escape local optima [5].

5.2. Other EA applications to RNA and protein folding

Beyond permutation-based encodings, conventional GAs remain a popular heuristic choice for representing complex RNA structural features, such as pseudoknots, which are generally excluded or severely restricted by computationally expensive dynamic programming methods [6]. For example, tools like GAknot deploy customised GAs to explore complex folding pathways, enabling the successful prediction of secondary structures containing pseudoknots while maintaining reasonable computational time [6].

In recent years, EDAs have emerged as a powerful evolutionary alternative to standard GAs in various bioinformatics optimisation tasks [21]. EDAs have seen direct application in single-sequence RNA secondary structure prediction [22]. For instance, the Co-EDAs method utilises a collaborative learning strategy integrating two distinct EDAs, one tailored for global search and one restricted for local search via a mutation operator, that share a single probability vector [22]. More recently, EDAs approaches with discrete representation have been applied to the RNA design problem [23].

Finally, the utility of EDAs extends broadly into protein folding and computational protein design. They have been successful when applied to simplified lattice models, such as the 2D and 3D hydrophobic-polar (HP) models [24, 25], where locating the minimum energy conformation is an NP-complete problem.

6. Experiments

The goal of experiments is twofold: First, to evaluate the performance of the permutation-based EDAs to find low-energy RNA secondary structure configurations. Second, to determine differences in the behavior of the algorithms since we would like to know how different approaches to probabilistic modeling in the space of permutations can produce advantages for this problem.

6.1. Experimental benchmark

We use a subset of the RFAM database [26], which provides experimentally determined RNA secondary structures for diverse RNA families. The initial collection contains 1015 instances in bpseq format, each specifying the RNA sequence and its annotated base pairs as the groundtruth structure. DNA thymine residues (T) are silently converted to RNA uracil (U).

For each instance we initially computed, using ViennaRNA: (i) the MFE structure and its free energy E_{MFE}, and (ii) the free energy E_{GT} of the groundtruth structure. We retain only instances satisfying $|E_{\mathrm{MFE}} - E_{\mathrm{GT}}| < 20\,\mathrm{kcal/mol}$. This threshold excludes structures whose groundtruth is energetically very far from the MFE fold, as energy-driven optimisation is unlikely to recover them regardless of the representation. Sequences longer than 500 nucleotides are also excluded since ViennaRNA's partition function scales as $O(n^3)$ and is impractical for long sequences within the run-time budget. We also excluded instances with a very large stem catalogue. After these filters, 123 instances remain.

Each combination of (instance, algorithm) is repeated over 5 independent runs. All algorithms share: population size $N = 500$, number of generations $T = 500$, neighbourhood size $|\mathscr{B}| = 20$, and stem-based population initialisation. MEDA-MK is run with its fixed Mallows concentration parameter $\theta = 1.0$. For the implementation of the permutation-based EDAs, the `perm_pateda` library was used[1].

All experiments were executed on a heterogeneous computing cluster under a 24-hour wall-clock limit per run. Five of the 3690 (instance, algorithm, seed) combinations – all involving MEDA-Kendall on

[1] https://github.com/rsantana-isg/perm_pateda

Table 1
Mean ± SD of Best Free Energy (BFE) and Best NED, and of F1 score vs. groundtruth for the best-energy (F1-MFE) and best-NED (F1-NED) solutions, for valid structures found per run. Lower is better for BFE/NED; higher is better for F1-MFE/F1-NED.

Algorithm	BFE (kcal/mol)	NED	F1-MFE	F1-NED
MK	**-12.58 ± 11.16**	**0.0031 ± 0.0019**	**0.559 ± 0.315**	**0.586 ± 0.307**
PL	-9.24 ± 11.68	0.0035 ± 0.0020	0.512 ± 0.324	0.562 ± 0.305
NHM	-8.61 ± 11.63	0.0035 ± 0.0019	0.526 ± 0.313	0.563 ± 0.299
Kendall	6.24 ± 12.79	0.0061 ± 0.0019	0.295 ± 0.283	0.419 ± 0.250
GMCayley	6.78 ± 13.63	0.0061 ± 0.0020	0.297 ± 0.286	0.413 ± 0.252
GMKendall	-0.98 ± 12.25	0.0049 ± 0.0020	0.400 ± 0.303	0.485 ± 0.273

two of the longer instances – exceeded this limit and were cancelled by the scheduler before completion; they are excluded from the aggregates reported below, which are accordingly computed over 610 MEDA-Kendall runs and 615 runs for every other algorithm.

For each run, we store the complete Pareto front and record the following summary statistics. The *best energy* is the minimum f_1 value on the Pareto front (reported only for runs that produce at least one finite-energy structure). The *lowest* $_{ned}$ is the minimum f_2 value. We also compute *precision* (P), *recall* (R), and *F1* score by comparing the base-pair set of the best solution against the MFE and the determined groundtruth structure. Finally, we computed distances between the best-energy solution on the Pareto front and the MFE structure. All results are aggregated over the 5 runs and 123 RNA sequences.

6.2. Analysis of the algorithms performance

Table 1 reports, for each algorithm, the mean best free energy (BFE) and best NED together with the mean F1 score against the experimentally determined groundtruth structure achieved by the best-energy solution (F1-MFE) and by the best-NED solution (F1-NED).

An analysis of Table 1 shows that MEDA-MK attains the best mean value on all four columns (BFE −12.58 kcal/mol, NED 0.0031, F1-MFE 0.559, F1-NED 0.586), with MEDA-PL and MEDA-NHM close behind on every column and clearly separated from the remaining three. MEDA-Kendall and MEDA-GMCayley are the weakest throughout, with a mean BFE around +5 to +7 kcal/mol. MEDA-GMKendall sits in between: its per-position spread vector allows it to improve steadily where the single-θ MEDA-Kendall stalls, but it is not able to produce the same performance as MK/PL/NHM within 500 generations.

A consistent pattern emerges across all six algorithms: F1-NED is always higher than F1-MFE (e.g. MEDA-MK: 0.586 vs. 0.559; MEDA-GMKendall: 0.485 vs. 0.400, the best-NED solution is, on average, a better predictor of the true biological structure than the best-energy solution, even though neither objective targets the groundtruth directly. This is consistent with NED being computed from the full Boltzmann ensemble rather than from a single point estimate. This illustrates the convenience of following a bi-objective approach. A solution favoured by NED already reflects some of the structural robustness that a real (evolved, functional) structure tends to have, whereas a solution favoured purely by minimum free energy, while directly indicating the solution quality, can be a fragile, isolated point in the energy landscape.

Figure 1 shows a summary of the distributions for the main performance metrics evaluated. The boxplots of best energy and best NED (panels a-b) mirror the ordering in Table 1, and the F1-vs-groundtruth violin plots (panels c-d) show the same three-tier split, with MK/PL/NHM reaching F1 ≈ 1 on a visible fraction of instances while Kendall/GMCayley have a much larger mass of runs at F1 ≈ 0, and panel (d) (F1-NED) is visibly shifted upward relative to panel (c) (F1-MFE) for every algorithm, the same gap seen in the table.

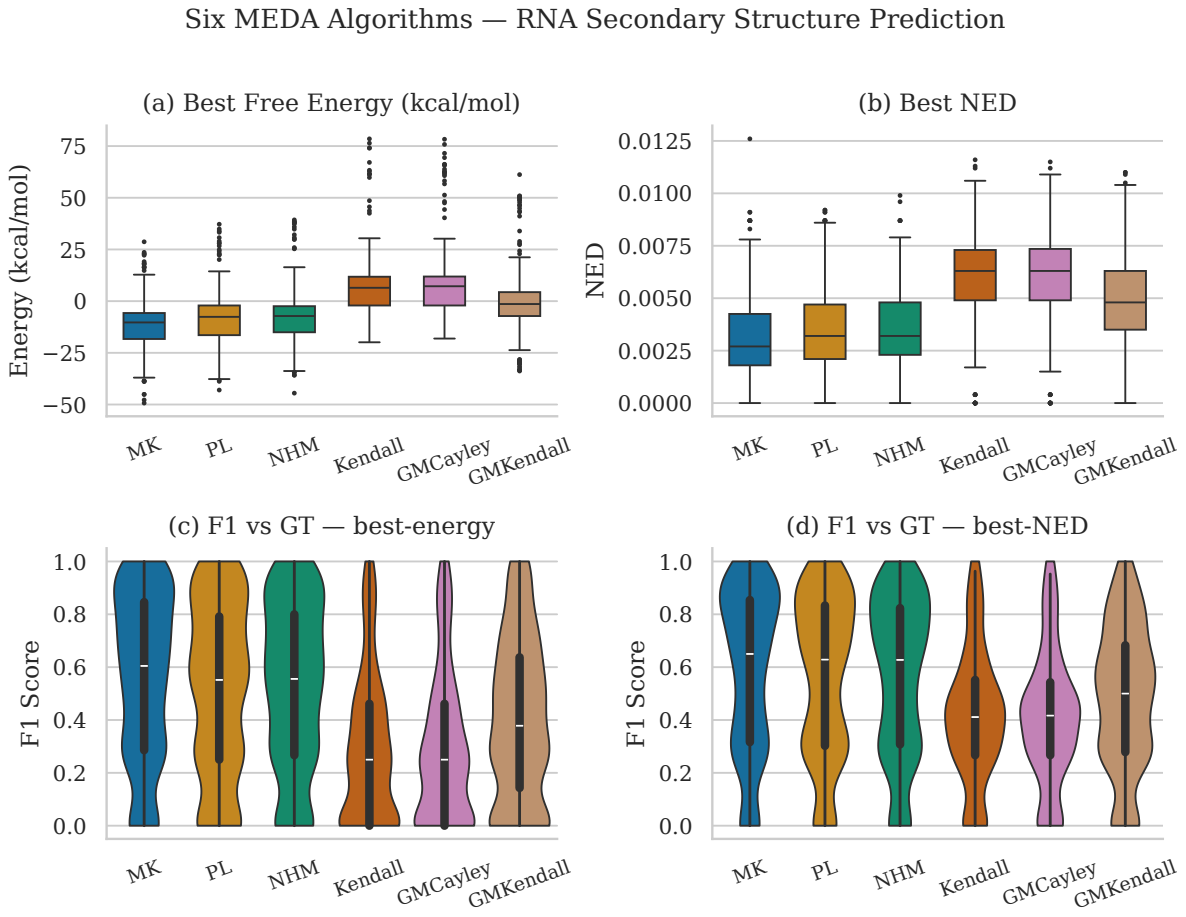


Figure 1: Per-run distributions of best free energy, best NED, and F1 vs. groundtruth (best-energy and best-NED solutions) for the six algorithms, over all completed runs.

6.3. Analysis of the algorithms evolution

Figure 2 shows the mean best free energy against generation number, averaged over all instances. MEDA-MK converges fastest and to the lowest final value, closely followed by MEDA-PL and MEDA-NHM, which converge at a near-identical rate to each other after an initial few generations; MEDA-GMKendall improves throughout the full 500 generations without plateauing, suggesting it would likely continue to close the gap under a larger generation budget; MEDA-Kendall and MEDA-GMCayley drop quickly for the first ~20-30 generations and then flatten, showing signs of early convergence. This is consistent with the three algorithms' underlying models: MEDA-MK's kernel is centred directly on each subproblem's current best solution every generation giving it a strong exploitation bias, whereas MEDA-Kendall and MEDA-GMCayley must re-estimate a consensus permutation from the selected population at every generation, which is both noisier and slower to sharpen around a specific solution.

6.4. Analysis of the solutions in the Pareto set

Beyond the best single solution, it is relevant to analyse what additional valuable information contained in the PS approximations learned by the algorithms. The two natural, sequence-only reference structures, the MFE structure and the structure that minimises NED globally for the sequence, i.e. that maximises expected pairing accuracy under the Boltzmann ensemble, are by definition extreme solutions of the PS and can be computed using the Vienna package. We refer to the structure at which the NED is minimised as the "NED structure".

For each algorithm and instance we pool the five seeds' final-generation Pareto fronts and re-filter for

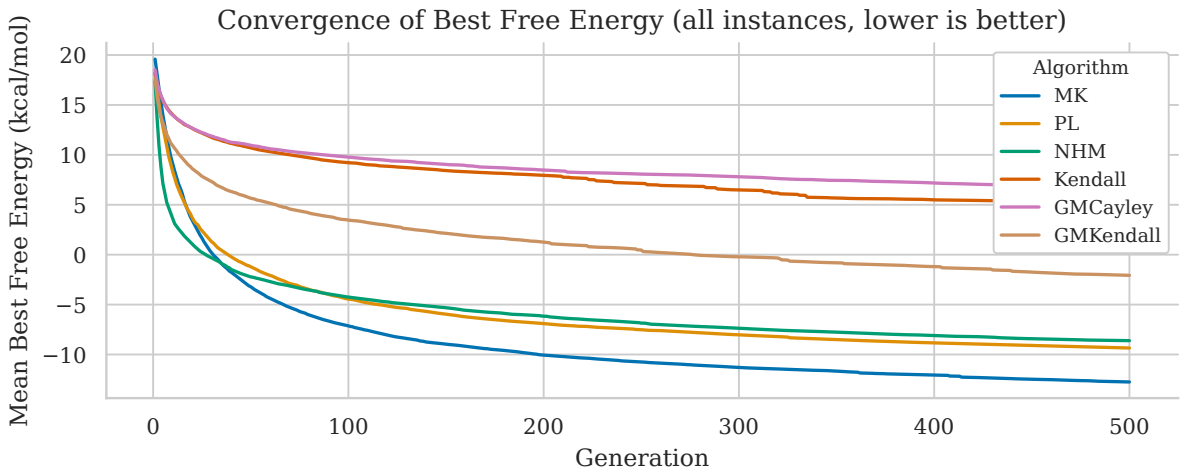


Figure 2: Convergence of the mean best free energy against generation number, averaged over all completed runs and instances.

Table 2
Analysis of the joint non-dominated set S obtained by pooling the five seeds' final-generation Pareto fronts, per algorithm and RNA instance. **MFE** $\in S$: number of instances for which the sequence's MFE structure is a member of S. **NED** $\in S$: number of instances for which the sequence's NED-optimal structure is a member of S. $|S|$: mean number of non-dominated solutions in S. **Better found**: number of instances for which some solution in S, other than the MFE and NED structures, has strictly lower base-pair distance to the groundtruth than both of them. l is the number of instances with at least one completed seed for that algorithm.

Algorithm	**MFE** $\in S$	**NED** $\in S$	$\lvert S\rvert$ **(mean)**	**Better found**	l
MK	33/123	40/123	3.93	18/123	123
PL	25/123	26/123	4.23	20/123	123
NHM	25/123	28/123	4.08	21/123	123
Kendall	6/123	5/123	5.07	28/123	123
GMCayley	7/123	5/123	5.19	26/123	123
GMKendall	15/123	17/123	5.61	20/123	123

non-domination to obtain one joint set S. Table 2 reports, per algorithm: how often the MFE structure and the NED structure are themselves members of S; the mean size of S; and how often S contains some *other* solution, closer to the groundtruth than both references.

The analysis of Table 2 confirms that MEDA-MK recovers the true MFE and NED structures far more often than any other algorithm (33/123 and 40/123, respectively), consistent with its lead on every other metric. Counter-intuitively, however, the "Better found" column is highest for the weakest algorithms on energy/NED (MEDA-Kendall 28/123, MEDA-GMCayley 26/123) and lowest for MEDA-MK (18/123). The mean size of S follows the same inverse pattern (MEDA-MK's is the smallest, at 3.93; MEDA-GMKendall's the largest, at 5.61). A plausible explanation is that MEDA-MK's search is the most tightly exploitative (Section 4), converging its population onto a narrow region around the energy optimum and therefore producing a small, energy-concentrated S; the weaker algorithms' broader, less-converged search instead produces a larger and more diverse S, which – even though it rarely contains the true energy/NED optima – has more chances of also containing some structurally different candidate that happens to sit closer to the groundtruth. This suggests that a strategy combining MEDA-MK's efficiency at approaching the thermodynamic optimum with a mechanism for preserving more diversity across the Pareto set could be a promising direction.

Table 3
Mean ± SD of wall-clock time (seconds) per run (efficiency; lower is better).

Algorithm	Mean ± SD	n
MK	8142.1 ± 3759.3	615
PL	3708.3 ± 2280.3	615
NHM	2418.2 ± 2318.5	615
Kendall	14268.5 ± 11950.4	610
GMCayley	**2195.0** ± **2324.9**	615
GMKendall	4516.1 ± 4072.1	615

6.5. Computational time of the algorithms

For completeness, Table 3 compares the computational time taken by the algorithms. The fastest methods are MEDA/D-GMCayley and MEDA/D-NHM, while MEDA/D-Kendall takes the longest time. The computational effort required by MEDA/D-MK is higher than that of all the other methods except MEDA/D-Kendall.

7. Conclusions

In this paper, we have introduced a direct permutation-based representation for the RNA folding problem. In comparison with previous stem-based representations, the size of the permutation equals the RNA sequence size. We have also evaluated different permutation-based multi-objective EDAs on 123 RNA sequences for which the secondary structure is known, and which often differ from the structures predicted by the extensively used Vienna package. Our results show that MEDA-MK produces the best results according to all defined metrics. However, the weakest algorithms are able to include in the final set of non-dominated solutions a higher number of secondary structures that are closer to the ground truth than the MFE and NED structures.

7.1. Future work

While we have focused on the investigation of EDAs, in order to assess the efficiency of the permutation representation, a comparison of the proposed approaches against other established multi-objective optimization paradigms is required. Pareto-dominance algorithms like NSGA-II [27] or standard MOEAs employing classical permutation crossover and mutation operators should be included in the comparison.

As future research, we consider the possibility of investigating other types of permutation-based EDAs [28, 17]. We also plan to extend the direct permutation representation to handle pseudoknots, which were excluded from the present work by the non-crossing constraint. Addressing secondary structures with pseudoknots would broaden applicability to biologically important RNA classes such as ribozymes and ribosomal RNAs. Finally, scaling to longer sequences and conducting systematic evaluations across a broader range of Rfam families would provide a more comprehensive picture of the scalability and generality of permutation-based approaches to RNA folding.

Acknowledgments

We acknowledge support from the Spanish Ministry of Science, Innovation and Universities (Projects ID2023-149195NB-I00 and PID 2022-137442NB-I00), and the Basque Government (Grant Nos. KK-2024/00030, KK-2026/00055, and IT2109-26). This work benefited from the Dagstuhl Seminar 25092 "Estimation-of-Distribution Algorithms: Theory and Applications" held at Schloss Dagstuhl–Leibniz Zentrum für Informatik in February 2025. Roberto Santana thanks Schloss Dagstuhl-Leibniz-Zentrum für Informatik for their support and hospitality.

Declaration on Generative AI

For the implementation of the programming code used for the experiments, analysis of the results and improving the writing of the paper, AI tools, in particular *claude code* with different AI models, have been used.

References

[1] B. Marchand, et al., Automated design of dynamic programming schemes for RNA folding with pseudoknots, Algorithms for Molecular Biology 18 (2023).

[2] M. Budnik, J. Wawrzyniak, Ł. Grala, M. Kadziński, N. Szostak, Deep dive into RNA: a systematic literature review on RNA structure prediction using machine learning methods, Artificial Intelligence Review 57 (2024) 254.

[3] M. Justyna, M. Antczak, M. Szachniuk, Machine learning for RNA 2d structure prediction benchmarked on experimental data, Briefings in Bioinformatics 24 (2023) bbad153.

[4] H. Zhu, P. Xiong, S. K. Zhou, Deep generalizable prediction of RNA secondary structure via base pair motif energy, Nature Communications (2025).

[5] H. H. Tsang, K. C. Wiese, Sarna-predict: A study of RNA secondary structure prediction using different annealing schedules, in: 2007 IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, IEEE, 2007, pp. 239–246.

[6] K.-K. Tong, K.-Y. Cheung, K.-H. Lee, K.-S. Leung, Gaknot: RNA secondary structures prediction with pseudoknots using genetic algorithm, in: 2013 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2013, pp. 136–142.

[7] P. Larrañaga, J. A. Lozano (Eds.), Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation, Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.

[8] H. Mühlenbein, G. Paaß, From recombination of genes to the estimation of distributions I. Binary parameters, in: H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature - PPSN IV, volume 1141 of *Lectures Notes in Computer Science*, Springer, Berlin, 1996, pp. 178–187.

[9] J. Ceberio, A. Mendiburu, J. A. Lozano, Introducing the Mallows model on estimation of distribution algorithms, in: International Conference on Neural Information Processing, Springer, 2011, pp. 461–470.

[10] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (2007) 712–731.

[11] M. Zangari-de Souza, A. Mendiburu, R. Santana, A. Pozo, Multiobjective decomposition-based Mallows models estimation of distribution algorithm. A case of study for permutation flowshop scheduling problem, Information Sciences 397–398 (s2017) 137–154.

[12] C. Coello, G. Lamont, D. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-objective Problems, Springer-Verlag New York Inc, 2007.

[13] M. Zuker, P. Stiegler, Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information, Nucleic Acids Research 9 (1981) 133–148.

[14] J. N. Zadeh, C. D. Steenberg, J. S. Bois, B. R. Wolfe, M. B. Pierce, A. R. Khan, R. M. Dirks, N. A. Pierce, NUPACK: Analysis and design of nucleic acid systems, Journal of Computational Chemistry 32 (2011) 170–173.

[15] R. Lorenz, S. H. Bernhart, C. Höner zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, I. L. Hofacker, ViennaRNA package 2.0, Algorithms for Molecular Biology 6 (2011) 26.

[16] K. C. Wiese, E. Glen, A permutation-based genetic algorithm for the RNA folding problem: a critical look at selection strategies, crossover operators, and representation issues, Biosystems 72 (2003) 29–41.

[17] J. Ceberio, A. Mendiburu, J. A. Lozano, The Plackett-Luce ranking model on permutation-based

optimization problems, in: Evolutionary Computation (CEC), 2013 IEEE Congress on, IEEE, 2013, pp. 494–501.

[18] S. Tsutsui, Node histogram vs. edge histogram: A comparison of probabilistic model-building genetic algorithms in permutation domains, in: 2006 IEEE Congress on Evolutionary Computation (CEC), IEEE, Vancouver, BC, Canada, 2006.

[19] J. Ceberio, E. Irurozki, A. Mendiburu, J. A. Lozano, A review of distances for the Mallows and generalized Mallows estimation of distribution algorithms, Computational Optimization and Applications 62 (2015) 545–564.

[20] K. Wiese, A. Deschenes, A. Hendriks, Rnapredict—an evolutionary algorithm for RNA secondary structure prediction, IEEE/ACM Transactions on Computational Biology and Bioinformatics 5 (2008) 25–41.

[21] R. Armañanzas, I. Inza, R. Santana, Y. Saeys, J. L. Flores, J. A. Lozano, Y. Van de Peer, R. Blanco, V. Robles, C. Bielza, P. Larrañaga, A review of estimation of distribution algorithms in bioinformatics, BioData Mining 1 (2008) doi:10.1186/1756–0381–1–6. URL: http://www.biodatamining.org/content/1/1/6.

[22] S. Srikamdee, P. Chongstitvatana, Collaborative learning of estimation of distribution algorithm for RNA secondary structure prediction, ECTI Transactions on Computer and Information Technology (ECTI-CIT) 14 (2020) 92–102.

[23] R. Santana, Multi-valued estimation of distribution algorithms for RNA design, in: Proceedings of 24th International Conference of the Italian Association for Artificial Intelligence (AIxIA-2026), Springer LNCS, 2026. Accepted for publication.

[24] B. Chen, L. Li, J. Hu, A novel EDAs based method for HP model protein folding, in: Proceedings of the 2009 Congress on Evolutionary Computation CEC-2009, IEEE Press, Norway, 2009, pp. 309–315.

[25] R. Santana, P. Larrañaga, J. A. Lozano, Protein folding in simplified models with estimation of distribution algorithms, IEEE Transactions on Evolutionary Computation 12 (2008) 418–438. URL: http://dx.doi.org/10.1109/TEVC.2007.906095.

[26] N. Ontiveros-Palacios, E. Cooke, E. P. Nawrocki, S. Triebel, M. Marz, E. Rivas, S. Griffiths-Jones, A. I. Petrov, A. Bateman, B. Sweeney, Rfam 15: Rna families database in 2025, Nucleic acids research 53 (2025) D258–D267.

[27] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.

[28] J. Ceberio, E. Irurozki, A. Mendiburu, J. A. Lozano, A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems, Progress in Artificial Intelligence 1 (2012) 103–117.