RNA Inverse Folding Can Be Solved in Linear Time for Structures Without Isolated Stacks or Base Pairs

Théo Boury
Laboratoire d'Informatique de l'Ecole Polytechnique (LIX; UMR 7161),
Institut Polytechnique de Paris, France

Laurent Bulteau
LIGM, CNRS, Université Gustave Eiffel, France

Yann Ponty[1]
Laboratoire d'Informatique de l'Ecole Polytechnique (LIX; UMR 7161),
Institut Polytechnique de Paris, France

Abstract

Inverse folding is a classic instance of negative RNA design which consists in finding a sequence that uniquely folds into a target secondary structure with respect to energy minimization. A breakthrough result of Bonnet *et al.* shows that, even in simple base pairs-based (BP) models, the decision version of a mildly constrained version of inverse folding is NP-hard.

In this work, we show that inverse folding can be solved in linear time for a large collection of targets, including every structure that contains no isolated BP and no isolated stack (or, equivalently, when all helices consist of 3^+ base pairs). For structures featuring shorter helices, our linear algorithm is no longer guaranteed to produce a solution, but still does so for a large proportion of instances.

Our approach introduces a notion of modulo m-separability, generalizing a property pioneered by Hales *et al.* Separability is a sufficient condition for the existence of a solution to the inverse folding problem. We show that, for any input secondary structure of length n, a modulo m-separated sequence can be produced in time $\mathcal{O}(n2^m)$ anytime such a sequence exists. Meanwhile, we show that any structure consisting of 3^+ base pairs is either trivially non-designable, or always admits a modulo-2 separated solution ($m = 2$). Solution sequences can thus be produced in linear time, and even be uniformly generated within the set of modulo-2 separable sequences.

2012 ACM Subject Classification Applied computing → Molecular structural biology

Keywords and phrases RNA structure, String Design, Parameterized Complexity, Uniform Sampling

Digital Object Identifier 10.4230/LIPIcs.WABI.2024.19

Supplementary Material *Software (Source code)*: `https://gitlab.inria.fr/amibio/linearbpdesign` [3], archived at `swh:1:dir:73673b14e891528ae11d29515662b482f730be12`

1 Introduction

RNA inverse folding is a fascinating algorithmic problem which, given a target secondary structure T, consists of designing one or several sequences, all of which should uniquely fold into the target T according to a reference folding prediction algorithm. Considering a folding prediction algorithm as a mathematical function $\Phi : \{A, C, G, U\}^\star \to \mathcal{S} \cup \{\perp\}$ mapping an RNA sequence to a unique predicted structure (or $\perp$ if equally likely alternatives exist), inverse folding can be abstracted as the search for a preimage $w \in \Phi^{-1}(T)$ of the target structure T. This naturally generalizes into a variety of design tasks which, given a predictive algorithm implementing a function Φ, aim to create one or multiple instances

[1] Both second and third authors should be considered as corresponding authors

predicted to behave in a certain way. Such a formulation is, in general, overly broad (*e.g.* it encompasses the concept of one-way functions in cryptography) to inspire reasonable hopes for a general solution. Still, a restriction of the inverse problem to certain types of computable functions/algorithms (*e.g.* amenable to dynamic programming) appears realistic and generally relevant to (synthetic) biology, yet poorly studied to this day.

In the specific case of RNA, despite being the object of substantial attention since its formal introduction in the early 1990s [9], the complexity of RNA inverse folding has remained elusive for almost three decades. A generalization of RNA inverse folding, including the energy model as part of the input, was shown to be NP-hard by Schnall-Levin *et al.* [19]. However, their reductions critically relied on (ab)using the energy model to encode a 3SAT instance, leaving the hardness of the problem largely open for a fixed energy model. The classic complexity of inverse folding was only settled, in 2020, when Bonnet *et al.* [2] finally showed the NP-hardness of RNA folding in a classic base pairs maximization setting. Such computational intractability (retrospectively) legitimizes a very large quantity of heuristic or exponential-time methods, based on local search [9, 4, 1, 23, 17], bio-inspired metaheuristics [12, 5, 10, 13], global sampling [16, 22], constraint programming [6, 8] and, more recently, neural networks-inspired generative models [18].

In parallel to complexity studies, Hales *et al.* [7] revisited the problem from a structural angle, attempting to characterize designable or undesignable families of secondary structures. The authors showed that saturated structures, having all positions paired, are designable if and only if their multiloop degrees do not exceed 4. They also introduced a notion of separability, a sufficient, yet not necessary in general, condition for a sequence to be a design for a given target. This notion allowed them to show that any target structure either features an occurrence of a locally-undesignable motif $\{m_{3\bullet}, m_5\}$, or can always be transformed into a separable structure by adding at most one base pair per helix. More strikingly, they proposed linear-time algorithms for producing a single solution for each characterized class of designable structures, painting a – puzzling – contrasted picture of general hardness (as per Bonnet *et al.* [2]) and practical facility for inverse folding.

In this work, we further those studies and show that, while conceptually simpler, the existence of a separated design for a given structure remains NP-hard. Conversely, any structure with helices of length greater than 3 base pairs is either trivially undesignable (*i.e.* contains $\{m_{3\bullet}, m_5\}$), or separable and can be designed in linear-time. This constraint is relevant to the objectives of RNA design, as targeted secondary structures are typically stable and tend to avoid shorter – unstable – helices. This result hinges on the introduction of a modulo m version of separability, coinciding with general separability whenever $m \geq n/2$, for which we give a Fixed-Parameter Tractable (FPT) algorithm running in time $\mathcal{O}(n2^m)$. We proved that this algorithm solves all instances with minimal helix lengths of 3 BPs when invoked with $m = 2$ and, even in this restricted setting, solves many instances with shorter helices in practice. Based on an unambiguous dynamic programming, our algorithm can be adapted into a random generator of separated designs. Finally, we show through empirical studies that separated sequences, despite being only guaranteed to constitute designs with respect to base pair maximization, are also likely to represent designs in the more realistic Turner energy model, and are far superior in this setting than compatible sequences.

2 Problem statement, definitions, and prior work

Algorithmically, RNA can be abstracted as a nucleotide sequence, *i.e.* a string $w \in \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{U}\}^n$ where n denotes the length of ω. Given a length n, a (non crossing/pseudoknot-free) secondary structure is a set $T \subset [1, n]^2$ consisting of base pairs such that:

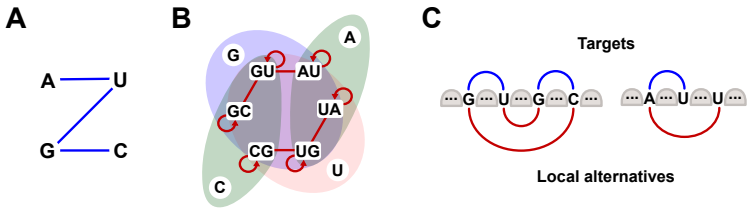


Figure 1 Local design rules. Base pair compatibility graph (A) and incompatibility graph for base pairs and unpaired nucleotides occurring within a loop (B): Connected base pairs, when jointly occurring within a loop of the target structure, can refold to form a local, an alternative structure having same number of base pairs as the target (C, left). Unpaired nucleotides may also interfere with some (A or C) or every (G or U) base pairs, leading to local alternatives (C, right).

- Each position in $[1, n]$ is involved in at most one base pair;
- Base pairs in T are pairwise non-crossing: $\forall (i, j) \neq (k, l) \in T$, $i < k$, either $i < k < l < j$ or $i < j < k < l$.
- Minimal distance in nucleotide number is parameterized by θ (default θ equals 0).

The set $\mathcal{S}_w$ of secondary structures compatible with an RNA sequence w is defined as: $\mathcal{S}_w := \{\text{Secondary structure } T \mid \forall (i, j) \in T, \{w_i, w_j\} \in \{\{\mathsf{G}, \mathsf{C}\}, \{\mathsf{A}, \mathsf{U}\}, \{\mathsf{G}, \mathsf{U}\}\}\}$.

Without loss of generality, a secondary structure can be represented as a tree $T = (V(T), E(T))$, whose nodes $V(T)$ are in bijection with base pairs (internal nodes[2]) and unpaired regions (leaves), and whose edges represent the inclusion of base pairs. Given a node $v \in V(T)$, we denote by parent(v) the parent of v in T, and by children(v) the list of children of v in T. A *loop* is the subtree restricted to node and its (direct) children. The tree is rooted in a special Root node, associated with the whole sequence interval. An *helix* of length ℓ of the tree is a maximal path $v_1, \ldots, v_\ell$ of base pair nodes such that each v_i with $i < \ell$ has a single child v_{i+1} (no leaf attached). A helix of length 1 is an *isolated base pair*. A helix of length 2 is an *isolated stack*. We define $h_{\min}$ as the minimum length over all helices of T. As the target tree is always explicit and unmodified through proofs and algorithms we do not specify it explicitly in the notations.

RNA inverse folding starts from a target secondary structure T, and attempts to construct a sequence $\omega \in \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{U}\}^n$ whose only base-pair maximizing secondary structure is T.

▶ **Problem 1** (Inverse-Folding$_{\mathsf{BP}}$).
Input: *Target secondary structure T, sequence length n*
Output: *Sequence $w \in \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{U}\}^n$ satisfying both:*
- *Compatibility with target structure: $T \in \mathcal{S}_w$;*
- *Uniqueness of the target as the optimal fold for the sequence: $\forall T' \in \mathcal{S}_w, T' \neq T, |T'| < |T|$.*

or $\perp$ if no such sequence exists.

Nevertheless, Inverse-Folding$_{\mathsf{BP}}$, mildly extended to allow further restrictions on individual sequence positions, was shown to be NP-hard by Bonnet *et al.* [2]. (The used restriction requires the inclusion of some constraints of the form "nucleotide i must be labeled by the base letter b")

[2] Base pairs may also be leaves of the tree when involving consecutive positions, which happens rarely in practice. We thus qualify as *internal node* any node in bijection with a base pair.

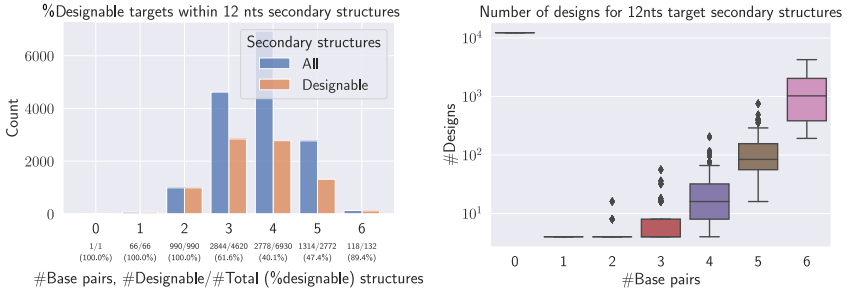


Figure 2 Exhaustive designability analysis of 12nts RNA sequences/structures. (Left) For a minimum base pair span of $\theta = 0$, there exists 15 511 secondary structures over 12 nucleotides, of which little over half (8 111) admits at least a solution to the inverse folding problem. (Right) The number of valid solutions varies substantially between targets and appears to depend on the number of base pairs. Overall, out of the 16 777 216 RNA sequences of length 12, only 399 348 ($\approx 2.4\%$) represent a valid design for some structure.

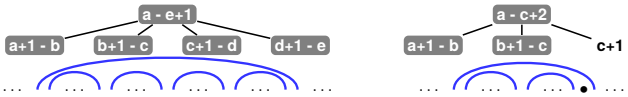


Figure 3 Forbidden motifs. Motifs m_5 (left) and $m_{3\bullet}$ (right), both shown as a tree (with a, b, c d, e arbitrary integers) and as nested base-pairs. Note that the relative order of the children base-pairs and the leaf in the $m_{3\bullet}$ pattern is irrelevant. Any assignment of base pair letters (either matching a proper coloring of the tree or not) leads to a possible local rerooting of at least two base pairs yielding an alternative thus making the structure undesignable. [7].

A sequence is called a design for a structure T if it represents a solution to the inverse folding problem for the input T. Note that the uniqueness condition can be tested in polynomial time using a variant of the Nussinov algorithm [14, 7]. In addition to showing that Inverse-Folding$_{\text{BP}}$ is in NP, such an algorithm enables, for moderate sequence lengths, a systematic folding of all sequences in order to characterize the set of structures admitting a solution. For instance, Figure 2 shows that, while only 2.4% of RNA sequences of length 12 represent a design for some target, roughly half of the secondary structure admits at least one solution sequence, and ≈ 49 on average, for the inverse folding problem.

We remind that, as noted by Halès *et al.* [7], two key motifs are not designable in a *base pair maximization* setting, see Figure 3:

- The m_5 motif consists of 5 base pairs occurring on the same loop (not counting the Root). No sequence can be designed for such a motif, since exposing 5 base pairs on a loop always allows for local refolding to have the same number of base pairs. This follows from the inspection of Figure 1, where the largest set of mutually compatible base pairs clearly has cardinality 4;
- The $m_{3\bullet}$ motif consists of 3 base pairs (excluding the Root) and at least one unpaired position. Indeed, as shown in Figure 1, the presence of an unpaired nucleotide either forbids the co-occurrence of any adjacent base pair (G or U), or only allows three (C or A). Since at most two of those base pairs can co-occur in a successful loop design, $m_{3\bullet}$ is not designable.

Any occurrence of these structures (or of any other undesignable structure, *cf* [21]) as a subgraph of an instance makes the instance undesignable.

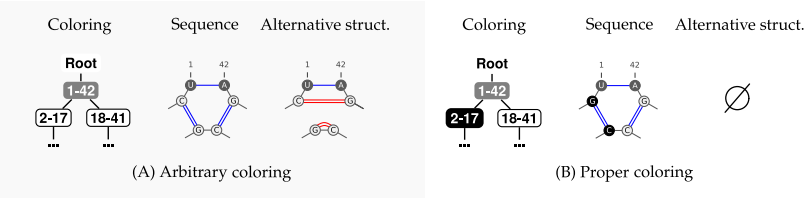


Figure 4 A proper coloring is necessary towards design. In (A), having two ○ children implies that the sequence derived from this coloring features a motif where G and C can reconfigure locally. In that case, they form an alternative structure that contains the same number of base pairs. Conversely, in (B), the proper coloring ensures that locally no alternative of equal (or better) energy exists by forcing some consecutive incompatibilities.

2.1 Inverse folding as a tree coloring problem

We start by reminding the coloring framework introduced by Halès *et al.* [7].

▶ **Definition 1** (Coloring). *A coloring of a (secondary structure) tree T is a function $\chi : V(T) \to \{●, ◍, ○, ∅\}$ associating a color to each node (except the root and the leaves which always get ∅).*

A coloring of a tree T typically induces multiple RNA sequences that are compatible with, but not guaranteed to fold into, the given secondary structure through letters assignment rules. Namely, in any sequence w derived from a coloring χ, we have for each $(i, j) \in T$:

- If $\chi((i,j)) = ● \to (w_i, w_j) = (\mathsf{G}, \mathsf{C})$;
- If $\chi((i,j)) = ○ \to (w_i, w_j) = (\mathsf{C}, \mathsf{G})$;
- If $\chi((i,j)) = ◍ \to (w_i, w_j) \in \{(\mathsf{A}, \mathsf{U}), (\mathsf{U}, \mathsf{A})\}$.

For ◍ nodes, the freedom in choosing (A, U) or (U, A) depends on the context: the choice may be unconstrained (*e.g.* when isolated within a helix), or forced (*e.g.* when two gray nodes are involved in a multiloop or stack). However, this property will only impact the number of sequences associated with the coloring, but bears no consequence on the existence of a solution to Inverse-Folding$_{\mathrm{BP}}$, since the problem asks for the production of a single sequence.

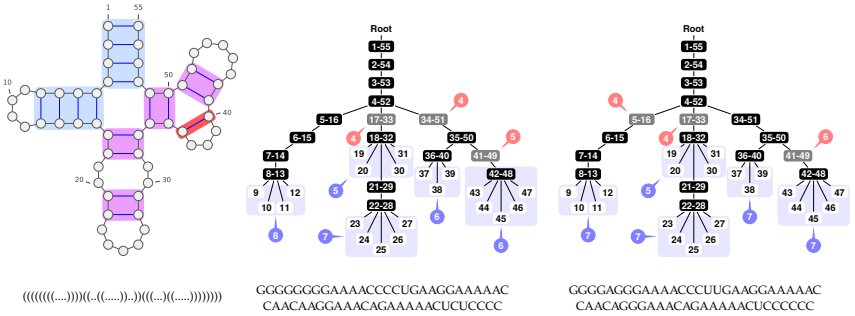
Denote by $\bar{c}$ the inverse of a color c, defined as $\overline{○} = ●$, $\overline{●} = ○$ and $\overline{◍} = ◍$. Denote by $|C|_c$ the number of occurrences of color c in vector C.

▶ **Definition 2** (Proper Coloring). *A coloring χ is proper when, for each node $v \in V(T)$, the vector of colors C, composed of the complementary color of the node concatenated with the colors of its children, respects the following constraints:*

$$|C|_● \leq 1, |C|_○ \leq 1 \text{ and } |C|_◍ \leq 2 \text{ with } C := \left[\overline{\chi(v)}\right].\left[\chi(v') \mid v' \in \mathit{children}(v)\right].$$

The use of the complementary color of v in C enables a compact definition: it forbids ● and ○ to have respectively ○ and ● children which would result in an alternative rerooting of the pairs. These conditions must also hold for the colorless Root, but with C being restricted to the colors of children(Root).

In terms of RNA design, the proper condition is necessary for an associated sequence to be a solution to inverse folding. Indeed, any coloring that is not proper will be associated with sequences that can be locally reconfigured, this without losing any base pair (see Figure 4 for an example).



1) Target structure 2) A proper coloring 3) A separated proper coloring

Figure 5 1) 2D and dot-bracket representations of a secondary structure. Helices of sizes respectively 1 (isolated base pairs), 2 (isolated stacks) and more than 3 are represented in light red, purple and blue. 2) Same secondary structure as a tree. The tree is colored and levels are represented in red and blue bubbles. The coloring is proper and non-separated as the level of the leaf 19 is the same as the level of the ● node 34-51. A non-separated coloring is not guaranteed to induce a design for its target, but may still do so, as is the case here. 3) Same secondary structure, colored in a separated (necessarily proper) manner. This coloring yields one or multiple designs (depending on the choice of AU or UA for ● nodes). Notably, this coloring is 2-separated, as leaves and ● nodes end up at odd and even levels respectively.

▶ **Definition 3** (Levels). *Given a coloring χ of a tree T, the level $L : V(T) \to \mathbb{Z}$ of a node v is $L(v) := |p|_{\bullet} - |p|_{\circ}$ where p denotes the color vector associated with the shortest node sequence from parent(v) to Root.*

On an RNA level, the concept of level helps categorize, and possibly control, the set of alternative structures to the target. Indeed, consider a sequence w generated from a coloring χ. First remark that, in order for an alternative structure to be competitive, every occurrence of C must be paired. Whenever two positions i and j interact to form a base pair, it can be shown that the inner interval $]i, j[$ interval contains $L(i) - L(j)$ more G than C. Meanwhile the outermost interval $[1, i[\cup]j, n]$ features the opposite imbalance ($L(i) - L(j)$ more C than G). In other words, any structure that contains a base pair $(i, j) \notin T$ already has $2 \times |L(i) - L(j)|$ fewer base pairs than the target structure. Thus only structures made of pairs (i, j) such that $L(i) = L(j)$ need to be considered as viable alternatives to T. This property can be exploited as a design principle, as formalized by the following property.

▶ **Definition 4** (Separated coloring). *A coloring χ is separated for a target T if and only if it is proper and the levels of ●-colored nodes and leaves do not overlap:*

$$\{L(v) \mid \chi(v) = \bullet\} \cap \{L(v) \mid v \text{ is a leaf}\} = \varnothing$$

This immediately suggests a design strategy that associates A to unpaired positions and assigns ● and ○ colors such that ● nodes end up as different levels as the leaves. Indeed, in this setting, Hales *et al.* [7] showed that the proper coloring of a saturated structure (without unpaired position) yields a sequence that uniquely folds with respect to base pair maximization. It follows that a competitive/alternative structure may only result from a base pair $(i, j) \notin T$, a position of which is a ● node while the other is a leaf. Ensuring that all ● nodes and leaves are found at different levels is thus sufficient to guarantee the designability of T, *i.e.* the existence of a solution to this instance of the inverse folding problem.

More generally, we say that a target secondary structure T is *separable* if there exists a coloring χ such that 'χ is separated for T. We recall the main results of Halès et al. [7] here.

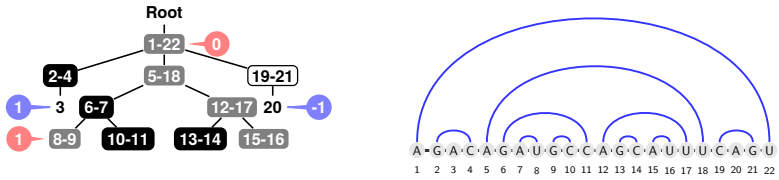


Figure 6 Designability does not imply separability. Left: A target structure that does not admit any separated coloring instance. Note that the coloring χ shown here puts the ● node 8-9 and the leaf 3 both at level 1. Right: Sequence w compatible with the coloring χ, which provably admits T as its single base pair-maximization structure (i.e. w is a design for T).

▶ **Theorem 1** (Separable $\implies$ Designable (Halès et al., 2017)). *If a tree/secondary structure T is separable, then T is designable.*

Moreover, given a separated coloring, an RNA sequence that uniquely folds into T, *i.e.* a solution to the inverse folding problem, can be found in linear time.

▶ Remark 2. Note that any design sequence w, generated through a separated coloring, avoids any alternative structure featuring GU base pair(s). Indeed, every G and C need to be paired to achieve the number of base pairs featured in the MFE. Meanwhile, the formation of any GU base pair, leaves one C and one A unpaired, resulting in the overall loss of at least one base pair. Structures featuring GU base pairs can thus be safely ignored.

3 Separability: Intrinsic and computational limits

Despite utilizing separability to explore a design of approximative instances, the work of Halès *et al.* [7] left open the complexity of searching for a separated coloring, as well as the existence of designable, yet non-separable, structures. An exhaustive search for all structures with up to 12 bases, summarized in Figure 2, shows that for such small instances, all designable instances are separable.

However, we show that non-separable designable instances can be constructed.

▶ **Proposition 1** (Designable $\not\implies$ Separable). *There exists a target structure which: i) does not admit a separated coloring; and ii) admits a solution to the inverse folding problem.*

Proof. We use the tree T of Figure 6 as a counterexample to the notion that separability fully captures designability. First, note that a separated coloring χ of T would be extremely constrained. Node $5-18$ should be ● and the nodes $2-4$ and $19-21$ are ● and ○ respectively, or vice-versa due to their respective leaf. Thus, we have two leaves at levels 1 and -1. At least, one of the two children of $5-18$, w.l.o.g $6-7$ is ● or ○. One child of $6-7$ is then necessarily ●, leading to a ● child of level $+1$ or -1. With two leaves at level $+1$ and -1, a direct consequence is that T is non-separable.

Now, we show that T is designable. We propose the sequence w of Figure 6. Using a simple dynamic programming algorithm, it is possible to check that the best folding for w is unique and corresponds to the secondary structure encoded as the tree T. Intuitively, the only competitive alternative base pair is the one corresponding to the overlap of the levels. It consists of joining the U from $8-9$ with the A at position 3. By doing so, note that the base pair $5-18$ will be disconnected with no way to pair A with another U due to the connection between 5 and 7. ◀

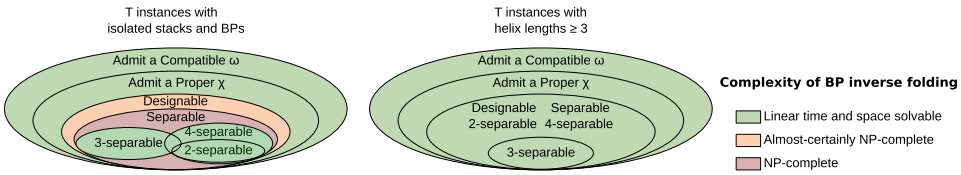


Figure 7 Instances of Inverse-Folding$_{\mathbf{BP}}$**.** For unconstrained instances (Left), Inverse-Folding$_{\mathrm{BP}}$ is likely NP-hard, as suggested by the hardness of a constrained version [2]. Finding a design for a separable target is also NP-hard but, for any fixed modular level m, m-separable targets can be designed in $\Theta(n)$ time. This suggests an algorithm, FPT on m, for all separable structures. When $h_{\min} \geq 3$ (Right), Thm 6 applies and the hierarchy collapses: any instance becomes 2-separable ($\implies$ separable and designable) and Inverse-Folding$_{\mathrm{BP}}$ can be solved in $\Theta(n)$ time.

Notice that, despite not being separated, the coloring shown in Figure 6 is compatible with a sequence that is a design for its target. This illustrates the fact that, while not being guaranteed to uniquely fold as their intended target, sequences produced from non-separated colorings may still represent solutions for the inverse folding problem.

Regarding computational complexity, although looking for a separable coloring is not directly equivalent to finding a design for a structure, we show that this decision problem (formalized below) is also NP-complete.

▶ **Problem 2** (Separability)**.**
Input: *Target tree T (without any occurrence of $m_{3\bullet}$ or m_5 motif)*
Output: *Coloring χ of the tree T such that χ is separated*

▶ **Theorem 3.** Separability *is NP-complete.*

The proof can be found in the appendix. It is obtained by reduction from Bin Packing, with a tree using one branch per item. Leaves and ● nodes enforce that items must be packed in consecutive ranges of levels (with ● levels at transitions between successive items and other levels saturated with leaves). Then, separating ● nodes are placed to enforce that series of consecutive items sum up to the target bin size, thus enforcing that items are ordered according to a correct bin packing.

4 Modulo separability as a parameterized tractable alternative

Then, we introduce a stratified version of separability, called modulo m-separability, or m-separability in short, which prescribes different modular values for the levels of ● and leaves nodes. Figure 7 describes the relative positioning of classes of instances and associated complexity results.

▶ **Definition 5** ((Modulo) m-separability)**.** *Let m be an integer. A coloring χ is* m-separated *(or* separated with modulus m*) for a target secondary structure T, if an only if χ is proper and*

$$\{L(v) \bmod m \mid \chi(v) = \bullet\} \cap \{L(v) \bmod m \mid v \text{ is a leaf}\} = \varnothing$$

using for negative levels $l < 0$ the classic $l \bmod m := (l + \lceil -x/m \rceil \times m) \mod m$.

Structure T is m-separable *if it admits an m-separated coloring.*

Clearly, modulo separability implies classic separability: if a coloring χ is m-separated for a target structure T, then χ is separated for T. Conversely, if a target structure admits a separated coloring, assigning levels in $[-a, b]$ to ● and leaf nodes, then the same coloring

is provably m'-separated for $m' := (b + a + 1)$ (since, for $l, l' \in [-a, b]$, $l \neq l'$ implies that $l \bmod m' \neq l' \bmod m'$). Note that, since there are at most $n/2$ base pairs/internal nodes in a target tree, then $0 \leq a, b \leq n/2$, and we have $m' \leq n$.

The concept of m-separability thus provides an angle to address the generation of separated colorings, so we introduce below the associated formalized algorithmic problem.

▶ **Problem 3** (Modulo Separability).
Input: *A tree T (with no $m_{3\bullet}$ or m_5 motif), a modulus $m \in \mathbb{N}$*
Output: *A coloring χ of T that is m-separated, or $\perp$ if no such coloring exists.*

As noted above, the problem specializes in the Separability problem when $m = n$, implying that Modulo Separability remains NP-complete. However, it can be efficiently solved for moderate values of m, as shown below. Practically, one may focus on small values of m since 99% of instances without isolated base pairs are separable with modulus $m \leq 6$ (*cf* Table 10).

4.1 Fixed parameter tractable algorithm for modulo-separability

We now show that, for any fixed modulus m, Modulo Separability can be solved in linear time. In particular, the problem is Fixed Parameter Tractable (FPT) for the parameter m.

Towards that goal, we consider a constrained version of Modulo Separability, where the modular values of levels are prescribed. Formally, we enforce that leaves only occur at modular levels in $\xi_L \subseteq [0, m[$, and ● nodes only occur at levels $[0, m[\backslash \xi_L$. In this constrained version of Modulo Separability, the existence of a valid solution can be solved in linear time using dynamic programming.

Namely, let us denote by $\mathrm{d}^{\xi_L}_{v \to c, \ell}$ the existence of a valid assignment (*i.e. solution*) for a subtree of T rooted at internal node v, with v occurring at level ℓ, and being assigned a prior color c. Provably, $\mathrm{d}^{\xi_L}_{v \to c, \ell}$ can be computed recursively by progressing along the tree, keeping track of the current level and checking that leaves and ● end up being assigned at modular levels ξ_L and $[0, m[\backslash \xi_L$ respectively. This leads to the following formula:

$$\mathrm{d}^{\xi_L}_{v \to c, l} = \begin{cases} \mathsf{False} & \text{if } \ell \in \xi_L \wedge c = \bullet \\ & \text{or } \ell' \notin \xi_L, \text{ and } \exists \text{ leaf in children}(v) \\ \mathsf{True} & \text{if children}(v) = \emptyset \\ \displaystyle\bigvee_{\substack{c' \text{ proper coloring of} \\ \text{children}(v) \text{ given } v \to c}} \ \bigwedge_{v' \in \text{children}(v)} \mathrm{d}^{\xi_L}_{v' \to c'(v'), \ell'} & \text{otherwise.} \end{cases}$$

$$\text{with } \ell' := \ell + \delta(c) \bmod m$$

where δ denotes the level increment induced by a color c, defined as $\delta(\bullet) = +1$, $\delta(\circ) = -1$ and $\delta(\bullet) = 0$. Moreover, in the outermost loop, the color assignment explored for children is meant to be locally proper: the colors $c(v')$ of the children, in conjunction with the color c of v must obey the conditions of Definition 2. Note that, in the absence of $m_{3\bullet}$ and m_5, the number of (proper) assignments is bounded by a constant, so this conjunctive loop does not impact the complexity. The existence of a ξ_L coloring for the full tree is then $\mathrm{Separable}_{\xi_L} := \mathrm{d}^{\xi_L}_{\mathsf{Root} \to \varnothing, 0}$.

The decision version of the problem can thus be solved in $\Theta(m.n)$ time. Indeed, the number of left-hand side terms scales in $\Theta(m.n)$, the number of proper coloring for children is bounded by a constant (since avoiding $m_{3\bullet}$ and $m_5 \implies |\mathrm{child}(v)| < 5$), and the total

number of executions of the conjunctive loops is in overall $\Theta(n)$. A backtracking procedure could also be defined to reconstruct a solution coloring in $\Theta(n)$ if such a solution exists ($\text{Separable}_{\xi_L} = \mathsf{True}$) or return $\perp$ otherwise ($\text{Separable}_{\xi_L} = \mathsf{False}$).

An algorithm for Modulo Separability can then be obtained by explicitly considering all the possible subsets of admissible modular levels for leaves:

- If T contains $m_{3\bullet}$ or m_5, return $\perp$
- For each $\xi_L \subseteq [0, m[$:
  - If $\#\text{Designs}_{\xi_L} > 0$, then backtrack to produce ξ_L-separated design
- Return $\perp$

The algorithm is correct since any ξ_L solution is also m-separated, and any m-separated coloring implies a partition of the leaves and ● nodes into disjoint levels ξ_L and $\chi_{\bullet} \subseteq [0, m[\backslash \xi_L$ respectively. A m-separated coloring is thus always found by invoking the DP algorithm over the 2^m subsets $\xi_L \in [0, m[$. The overall complexity of the algorithm is in $\Theta(n.m.2^m)$ time and $\Theta(m.n)$ memory, and we conclude with the parameterized complexity of the problem with respect to m.

▶ **Theorem 4.** Modulo Separability *is Fixed Parameter Tractable for the modulus* m

4.2 Random generation of m-separated RNA sequences

We then turn to the uniform random generation of m-separated sequences, defined as a design w for T, featuring A on unpaired positions, and such that the coloring χ_w, obtained by replacing base pairs with suitable color ($(\mathsf{G},\mathsf{C}) \to$ ●, $(\mathsf{C},\mathsf{G}) \to$ ○ and (A,U) or $(\mathsf{U},\mathsf{A}) \to$ ◉), is m-separated.

▶ **Problem 4** (Uniform Modulo Separated Generation).
Input: *Target tree* T *(with no* $m_{3\bullet}$ *or* m_5 *motif)*
Output: *RNA sequence* w*, associated with* m*-separated coloring* χ_w*, such that*

$$\mathbb{P}(w \mid \chi_w \textit{ is } m\textit{-separated}) = \frac{1}{|\{w' \mid \chi_{w'} \textit{ is } m\textit{-separated}\}|}$$

Again, we approach this problem by first solving a more constrained version where the modular levels of leaves are explicitly given as a set ξ_L. Then, in the spirit of Reinharz *et al.* [16], we adapt the above recurrence, through a simple algebra change, to count the number $\mathrm{p}^{\xi_L}_{v\to\mu,l}$ of RNA sequences, associated with a ξ_L separated coloring (for a subtree of T rooted at v, with v occurring at level l, and being assigned a nucleotide assignment μ).

$$\mathrm{p}^{\xi_L}_{v\to\mu,\ell} = \begin{cases} 0 & \text{if } \ell \in \xi_L \text{ and } \mu \in \{(A,U),(U,A)\} \\ 0 & \text{if } \ell' \notin \xi_L \text{ and } v \text{ has a leaf attached} \\ 1 & \text{if children}(v) = \varnothing \\ \displaystyle\sum_{\substack{\mu' \text{ proper assignment} \\ \text{children}(v)\to\Sigma^2\cup\{\emptyset\}}} \prod_{v'\in\text{children}(v)} \mathrm{p}^{\xi_L}_{v'\to\mu'(v'),\ell'} & \text{otherwise } (\ell' := \ell + \delta(\mu) \bmod m). \end{cases}$$

where μ' is a nucleotide assignment to the children of v, consistent with a proper coloring and additionally respecting natural constraints on the content ((A,U) or (U,A)) of pairs of ● nodes (same for both if one parent of other, different content if siblings). Once again, the colorless Root node needs to be distinguished, and the overall number of designs is given by $\#\text{Designs}_{\xi_L} := \mathrm{p}^{\xi_L}_{\mathsf{Root}\to\varnothing,0}$.

The following backtrack procedure then produces a uniform random RNA sequence that corresponds to a m-separated coloring for a fixed set ξ_L. In that case, by abuse of language, we say that the sequence is ξ_L *separated*. More precisely, $\mathsf{backtrack}(v, c, \ell)$ produces a random sequence, associated with a ξ_L separated coloring, for the subtree anchored in v, reached at height ℓ, where the root is assigned a pair of bases $\mu \in \Sigma^2$. It first picks a random proper assignment μ' for the children, weighted by the corresponding number of solutions (namely, $\prod_{v' \in \text{children}(v)} \mathrm{p}^{\xi_L}_{v' \to \mu'(v'), \ell'}$, with $\ell' := \ell + \delta(\mu) \mod m$). The resulting sequence is then

$$\prod_{v \in \text{children and leaves}(v)} \begin{cases} \mathsf{A} & \text{If } v' \text{ is a leaf} \\ b.\mathsf{backtrack}(v', \mu'(v'), \ell').b' & \text{otherwise, with } \mu'(v') = b.b' \end{cases}$$

The resulting algorithm, consisting of precomputing all $\mathrm{p}^{\xi_L}_{v \to \mu, \ell}$, followed by a sequence of k backtracks, provably returns k random, uniformly-distributed and independent designs that are ξ_L separated in time $\Theta(n.m + k.n)$.

To leverage the uniform generation for a fixed ξ_L into a uniform generation of m-separated designs, we implement a strategy (see [15, pp 77] for details), proven in Appendix C, which start by generating some ξ_L, and then uses a suitable rejection to correct the emissions probabilities of sequences compatible with several ξ_L.

▶ **Theorem 5.** Uniform Modulo Separated Generation *can be performed in an average-case complexity that is Fixed Parameter Tractable for the modulus parameter m.*

5 Structures without isolated stacks and base pairs are 2-separable

Although separability does not give a full characterization of designability in general (cf Prop. 1), we obtain a much stronger result for structures without small helices, as hinted by the fact that all counter-examples and hardness gadgets heavily use isolated base pairs in their construction. Indeed, we show that a 2-separated coloring can be constructed for *all* structures without forbidden motifs $(m_{3\bullet}, m_5)$ and $h_{\min} \geq 3$, so indeed all such structures are designable. Since avoiding $(m_{3\bullet}, m_5)$ is a necessary condition for designability, we obtain the stronger characterization stated in Corollary 9.

▶ **Theorem 6.** *Every $(m_{3\bullet}, m_5)$-avoiding target T, having $h_{\min} \geq 3$, admits a 2-separated coloring*

Proof. First, let us remark that helices can be treated as atomic objects, and compacted into the edges of a *helix tree*, whose edges are helices (sequence of consecutive BP nodes), and whose internal nodes are either:

- Multiloops, consisting of 2 or 3 children/BPs/Helices, and no leaf (so $m_{3\bullet}$ does not occur);
- Internal/Bulges/Hairpin (IBH) loops, consisting of at most 1 BP/Helix and featuring at least one leaf/unpaired node.

Remark that, while constructing a separated coloring assigning a modular level ξ_L to leaves, those two motifs are the only sources of immutable constraints:

- Any proper coloring of a multiloop features at least one ◉ node, so the levels of children/nodes need to be set to a level $\overline{\xi_L} := \xi_L + 1 \mod 2$;
- Any IBH loop features at least one leaf within its children, which needs to be set to a modular level ξ_L.

Conversely, beyond their first BP, helices may be colored with very limited constraints and can be used to *offset* multiloops and IBH loops.

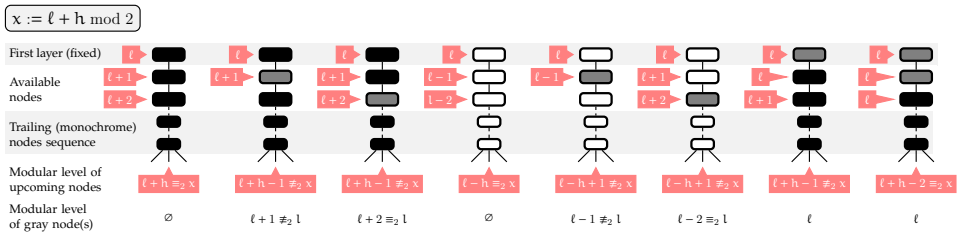


Figure 8 Alternative colorings for helices consisting of 3+ base pairs ($h_{\min} \geq 3$), such that the modular level of the following nodes is offset as needed. Such colorings can be chosen to respect a prescribed level for ● nodes and, a predetermined color for the first node/base pair of the helix.

▶ **Lemma 7.** *Let $\overline{\xi_L}$ denote the prescribed modular level for ● nodes. Consider an helix H consisting of 3 BPs or more ($h_{\min} \geq 3$), whose first BPs is assigned some color $c \in \{●, ○, ●\}$.*

Then for each modular level $l \in [0, 1]$ for the first BP of H ($c =$ ● only if $l = \overline{\xi_L}$), and targeted exit modular level $l' \in [0, 1]$, there exists a coloring for the rest of H such that:

- *The modular level of the upcoming nodes,* i.e. *those immediately following H, is l';*
- *Base pairs can only be ●-colored at modular level $\overline{\xi_L}$.*

Proof. The proof is essentially based on case decomposition, and summarized in Figure 8. We show that, for any l and $h_{\min} \geq 3$, there exists a color assignment to the first 3 nodes of the helix, such that the modular level of upcoming nodes is either 0 or 1, so l' can be reached. Moreover, if such a coloring starts with ● or ○, and uses a single ● node, then there exists an alternative coloring placing this ● node at the opposite modular level, so one of them places their ● node at the intended level $\overline{\xi_L}$. Finally, if the first node is set to ●, then the consistency condition above implies that $l \bmod 2 = \overline{\xi_L}$, so that ● nodes are naturally found at an admissible modular level. ◀

It follows that any helix tree starting with an initial helix H can be colored into a 2-separated coloring. Starting at initial level $l = 0$ and having initial BP color c ($\neq$ ● if $\xi_L = 0$), color the rest of H as shown in the proof of Lemma 7, depending on $\overline{\xi_L}$ and the type of upcoming loop (target $l' = \overline{\xi_L}$ for Multiloops; $l' = \xi_L$ for IBH loops), while ensuring that ● nodes end up at $\overline{\xi_L}$ modular level (which can always be done from Lemma 7). The remaining nodes of the loop are then colored in a proper/greedy manner, and we iterate the process recursively on the children helices of the loop (if any) until the full tree is colored.

Since its level cannot be offset, the Root node must be treated as a special case. Indeed, if the Root has at least one leaf/unpaired position, then the modular value 0 is taken by the leaf, so we must have $\xi_L = 0$. Conversely, if the Root supports at least 3 helices, then at least one needs to start with a ● node, so we must have $\xi_L = 1$. Regardless of this restriction on ξ_L, in both cases the first base pair of each helix (if any) supported by the Root can be properly colored, and helices can be independently colored using the above strategy, ultimately yielding a 2-separated coloring. ◀

▶ **Corollary 8.** Inverse Folding, *restricted to instances with $h_{\min} \geq 3$ (containing no isolated base pair and no isolated stacks) is solvable in linear time and space.*

It is a direct consequence of Theorem 6 and of the DP scheme introduced in Section 4.1. Indeed, for $m = 2$, the DP algorithm only needs to be run twice ($\xi_L = 0$ and $\xi_L = 1$) in linear time/space, to produce a 2-separated coloring whenever such a coloring exists (guaranteed by Theorem 6). The coloring can then be transformed into a design, *i.e.* a solution to the

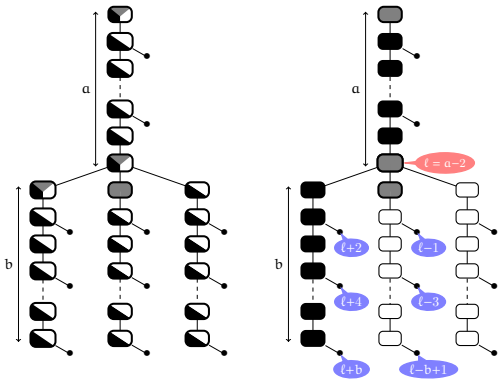


Figure 9 Main gadget used to build non-separable instances with $h_{\min} = 2$. Left: Admissible colors for each node (up to branch symmetries). Right: Example coloring and levels of a selection of leaves and ● nodes. Note that along with the ● node at level ℓ, there always exists a leaf at level $\ell + m$ or $\ell - m$ for $2 \leq m \leq b$, ruling out modulo separability for small m.

INVERSE FOLDING problem. Similarly, UNIFORM MODULO SEPARATED GENERATION can also be performed in linear expected time and space as long as input instances contain only helices of size 3 or more.

▶ **Corollary 9.** *Let T be a target structure with $h_{\min} \geq 3$, then the following are equivalent: i) T is designable; ii) T is 2-separable; and iii) T avoids $(m_{3\bullet}, m_5)$.*

With this result, the hierarchy of instances collapses as depicted on the left of Figure 7 A natural follow-up question is whether the bound 3 on the helix length is tight. Indeed, there are non-separable and designable instances with $h_{\min} = 1$ (Proposition 1), but the question remains for $h_{\min} = 2$. In Proposition 10 we give a non-separable instance without isolated base pairs, so $h_{\min} = 3$ is indeed tight to ensure separability.

▶ **Proposition 10.** *There exist non-separable structures with $h_{\min} = 2$.*

The full proof relies on a counterexample built from the gadget in Figure 9. Intuitively, $T(a, b)$ saturates all levels modulo b with leaves, so that none remains available for ● nodes. Meanwhile, the presence of multiloops forces proper colorings to use ● nodes, so a collision occurs and the gadget is not m-separable for any $m \leq b$. By assembling 5 copies of $T(a, b)$ with large b and increasing values of a, we obtain a target that is not separable for any m.

6 On the relevance of separated sequences towards realistic designs

While the existence of a linear-time algorithm for a reasonable restriction of the inverse folding problem is already notable, its practical relevance may be perceived as hindered by several limitations: our algorithms are only guaranteed to produce design solutions for helices beyond 3 base pairs; proper colorings only allows the design of highly-constrained (multi)loops; and solutions to the base pair inverse folding are not guaranteed to represent good solutions in more realistic energy models, such as the Turner nearest-neighbor model. To assess the promises of separated designs in realistic settings, we performed computational experiments, using a `Python` implementation available at `https://gitlab.inria.fr/amibio/linearbpdesign`, to assess the potential of separated colorings to inform future RNA design methods.

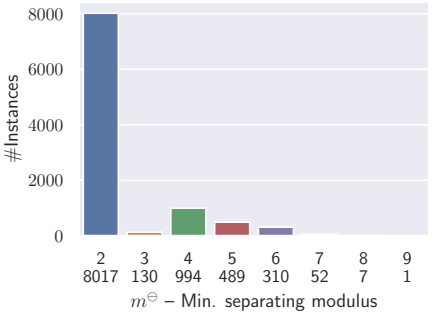


Figure 10 Minimal modulus $m^{\ominus}$ required to separate 10 000 random targets ($n = 100$; $\theta = 3$) featuring 1^+ isolated stack(s). All targets were found to be separable, with $m^{\ominus} \leq 9$.

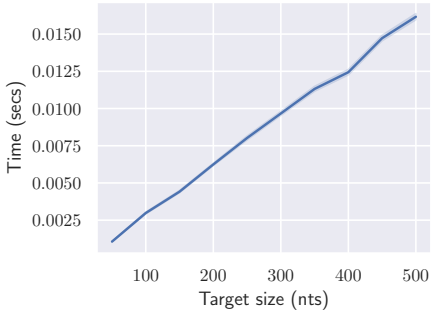


Figure 11 Average runtime of our algorithm (preprocessing + sampling of single instance) for separable instances ($h_{\min}$=3; no $m_{3\bullet}/m_5$) on a domestic laptop (AMD Ryzen 7 3700U).

6.1 Targets with isolated BPs/stacks are frequently separable

While our algorithm is only guaranteed to produce a design when $h_{\min} \geq 3$, it also produces (guaranteed correct) solutions for input with smaller helices, as long as a separated coloring exists for them. For very small targets, an exhaustive analysis is feasible, consisting of folding/testing the unicity of the MFE folding for all sequences of length $n = 12$ (see Figure 2). Moreover, once a design w is found for a target T, it is easy to test if the associated coloring χ_w is separated, and to compute minimal modulus value $m^{\ominus}$ such that χ_w is $m^{\ominus}$ separated. We found that *all of the 8 111 designable targets are also separable*, despite a very large proportion of them featuring isolated stacks and base pairs. Moreover, all designable targets admit separated solutions associated with very small values of the modulus m (7 690 for $m = 2$, 420 for $m = 3$ and $m = 1$ only for the empty structure).

To further measure the proportion of separable structures within larger targets featuring isolated stacks, we implemented a uniform random generation algorithm [15]. We produced random target secondary structures of length 100 with a min base pair span of $\theta = 3$. We used rejection to produce a synthetic dataset consisting of 10 000 targets having at least one helix of size 2 while avoiding $m_{3\bullet}$ and m_5. For each target T, we ran an in-house implementation of the algorithm in Section 4.1 with increasing modulus, to find the minimal modulus $m^{\ominus}$ such that T admits a $m^{\ominus}$ separated coloring. Table 10 summarizes our results, which we discuss below.

Remarkably, all of the 10k targets in the datasets could be designed using our algorithm, and thus admit a separable coloring. Moreover, roughly three-quarters (80%) of the targets were found to be 2-separable, and less than 1% of the targets required the consideration of values for $m^{\ominus}$ beyond 6. The max value for $m^{\ominus}$ in this dataset was 9, an order of magnitude lower than the sequence length. Clearly, since we have shown the existence of non-separable instances with isolated stacks and no isolated base pair, this observation does not generalize to arbitrary sequence lengths. However, the large size of these counterexamples suggests that the proportion of separable structures, despite ultimately decaying exponentially [21], may remain non-negligible for relevant RNA target sizes.

6.2 Separated designs are promising candidates in the Turner model

We now consider a more realistic setting, where the inverse folding problem is now considered with respect to the Turner nearest-neighbor energy model [20]. To assess the value of a sequence in the Turner model, we introduce a metrics which we call the (signed) *energy distance* $\Delta\Delta G(w,T)$ of a target T to its *most stable distant alternative* for the sequence w:

$$\Delta\Delta G(w,T) := \Delta G(w, \alpha_{d^-}(w,T)) - \Delta G(w,T), \alpha(w,T) := \min\{\Delta G(w,T') \mid |T',T| \geq d^-\}$$

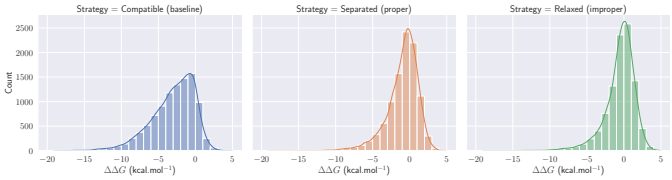
where $\Delta G(w,T)$ is the Turner free-energy, $|T,T'| := |T \triangle T'|$ denotes the base-pair distance, and d^- represents the minimum base pair distance to T. Both ΔG and $\alpha_{d^-}(w,T)$ can be obtained by appropriate calls to the `ViennaRNA` package [9], namely `RNAeval` and `RNAsubopts`, using max energy distance parameter $E = 5$ (so our estimation of $\Delta\Delta G(w,T)$ is bounded by 5). A positive energy distance confirms that w is a solution to the Turner version of inverse folding, and dominates its competitors by $\Delta\Delta G(w,T)\,\text{kcal.mol}^{-1}$. Meanwhile, a negative energy distance indicates that the target T is dominated by some alternative structure, having $\Delta\Delta G(w,T)\,\text{kcal.mol}^{-1}$ lower free-energy than the target.

We consider three strategies for sampling sequences: i) The *compatible* model uniformly generates random sequences compatible with the target (A for unpaired positions; AU, UA, GC or CG for base pairs); ii) The *separated* model uses the sampler described in Section 4.2 to generate sequences that are 2-separated and proper; iii) The *relaxed* model generates sequences that are 2-separated, but not necessarily proper by assigning uniform random pairs to the base pairs of a multiloop. The *relaxed* model enables a heuristic extension of our algorithms supporting multiloops of arbitrary degrees, noting that the local refolding (see Figure 4) occurring in the BP model for non-proper sequences are either unrealistic or outright impossible, in the Turner energy model.

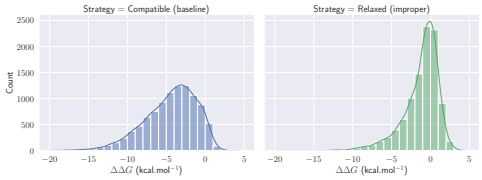
Separated sequences substantially improve over compatible random sequences. We first asked a basic question: *Are separated sequences better candidates for design in the Turner model than sequences merely compatible with the target?* The answer is not obvious since separated sequences are only guaranteed to represent designs for the BP max. model. We considered instances of size $n = 100$ admitting a solution to Inverse-Folding$_{\text{BP}}$ ($\theta = 3$; no $m_{3\bullet}/m_5$; $h_{\min} \geq 3$). We generated 10 000 random targets and, for each target, sampled a single sequence using each of the 3 strategies above and computed the energy distance.

The results, summarized in Figure 12.top suggest that separated sequences represent a substantial improvement over merely compatible sequences. Indeed, while 10% of compatible sequences ended up being good design candidates ($\Delta\Delta G > 0$), the proportion of successful designs increases to approximately one-third (35%) for separated sequences, and further to 43% for relaxed design. A similar trend can be observed for the average $\Delta\Delta G$ (distance to the first alternative/competitor) among successful designs, being of 0.79/0.98/1.06 kcal.mol^{-1} in the compatible, separated and relaxed models respectively. The surprisingly good behavior of the relaxed model, which was mostly introduced to overcome unrealistic limitations on multiloops, remains to be explained.

Relaxed sequences enable designs for multiloops having higher degrees. We also tested the capacity of the relaxed model to generate solutions for multiloops of higher degrees, noting that the avoidance of $m_{3\bullet}$ and m_5 restricts the maximum degree of a multiloop to 4. We used the above-mentioned generation algorithm to generate uniform design targets of size $n = 100$, featuring at least one (but frequently many) occurrence of $m_{3\bullet}$ and m_5. As shown in Figure 12.bottom, compatible sequences are again substantially outperformed



(a) Target structures avoiding $m_{3\bullet}$ and m_5.



(b) Target structures featuring multiloops of arbitrary degrees.

Figure 12 Comparison of compatible (baseline), separated, and relaxed models for targets having $n = 100, \theta = 3, h_{\min} = 3$. For energy distance parameters, we took $d^- = 3$ and $E = 5$.

by the relaxed separated model in this setting, with 31.5% of the separated/non-proper sequences (as opposed to only 5.1% of compatible sequences) representing successful designs ($\Delta\Delta G > 0$), on average 0.86 kcal.mol^{-1} more stable than their best competitor.

7 Conclusion

Adapting a coloring perspective initially introduced by Halès *et al.* [7], we have shown that the inverse folding problem can be solved in linear time for all target secondary structures having minimum helix length equal to 3. Towards that main result, we have established the existence of designable, yet non-separable, instances of inverse folding, and the NP-hardness of finding a separable design in the initial sense of Halès *et al.* We have also introduced concrete algorithms for the problem of finding a m modulo-separated coloring, which we have shown to be NP-hard yet FPT-solvable for m. Already for $m = 2$, the scope of our algorithms encompasses all targets without isolated base pairs and stacks, but also extends much beyond, in a way that remains to be fully characterized. Beyond base pair maximization, modulo-separated sequences may also represent a solid foundation towards concrete design methodologies. Namely, we empirically showed that, for the Turner energy model, separated sequences tend to represent better design candidates than merely compatible sequences, and that the limitations on loop degrees (intrinsic to the BP maximization model) can be overcome by relaxing our design model while retaining substantial performances.

Future work should focus on how much of designable sequences are covered by sequences obtained with (modulo)-separated colorings. More importantly, does the space of (modulo)-separated colorings always/often contain a design with respect to the nearest-neighborhood Turner energy model? Even if it unlikely to hold unconditionally, it is plausible that some extensions of separability and m-separability will achieve theoretical and practical solutions for inverse folding in more general energy models. As a first step, separability in a stacking energy model seems a relevant goal, even if less ambitious than the Turner model. It would probably require to go beyond the current coloring formalism, and motivate the introduction of more general notions of defect to capture imbalance at the dinucleotide level.

References

1 Mirela Andronescu, Anthony P. Fejes, Frank Hutter, Holger H. Hoos, and Anne Condon. A new algorithm for rna secondary structure design. *Journal of Molecular Biology*, 336(3):607–624, 2004. doi:10.1016/j.jmb.2003.12.041.

2 Édouard Bonnet, Paweł Rzążewski, and Florian Sikora. Designing rna secondary structures is hard. *Journal of Computational Biology*, 27(3):302–316, 2020. PMID:32160034. doi:10.1089/cmb.2019.0420.

3 Théo Boury, Laurent Bulteau, and Yann Ponty. LinearBPDesign. Software, version 1.0., swhId: swh:1:dir:73673b14e891528ae11d29515662b482f730be12 (visited on 2024-08-19). URL: https://gitlab.inria.fr/amibio/linearbpdesign.

4 Anke Busch and Rolf Backofen. INFO-RNA–a fast approach to inverse RNA folding. *Bioinformatics*, 22(15):1823–31, 2006.

5 Ali Esmaili-Taheri and Mohammad Ganjtabesh. ERD: a fast and reliable tool for RNA design including constraints. *BMC Bioinform.*, 16:20:1–20:11, 2015.

6 Juan Antonio Garcia-Martin, Ivan Dotu, and Peter Clote. RNAiFold 2.0: a web server and software to design custom and Rfam-based RNA molecules. *Nucleic Acids Research*, 43(W1):W513–W521, May 2015. doi:10.1093/nar/gkv460.

7 Jozef Hales, Alice Héliou, Ján Manuch, Yann Ponty, and Ladislav Stacho. Combinatorial RNA design: Designability and structure-approximating algorithm in watson-crick and nussinov-jacobson energy models. *Algorithmica*, 79(3):835–856, 2017.

8 Stefan Hammer, Wei Wang, Sebastian Will, and Yann Ponty. Fixed-parameter tractable sampling for RNA design with multiple target structures. *BMC bioinformatics*, 20:209, April 2019. doi:10.1186/s12859-019-2784-7.

9 Ivo L Hofacker, Walter Fontana, Peter F Stadler, L Sebastian Bonhoeffer, Manfred Tacker, and Peter Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie/Chemical Monthly*, 125(2):167–188, 1994.

10 Robert Kleinkauf, Martin Mann, and Rolf Backofen. antaRNA: ant colony-based RNA sequence design. *Bioinformatics*, 31(19):3114–3121, May 2015. doi:10.1093/bioinformatics/btv319.

11 William Andrew Lorenz and Yann Ponty. Non-redundant random generation algorithms for weighted context-free grammars. *Theoretical Computer Science*, 502:177–194, 2013. Generation of Combinatorial Structures. doi:10.1016/j.tcs.2013.01.006.

12 Rune B. Lyngsø, James W. J. Anderson, Elena Sizikova, Amarendra Badugu, Tomas Hyland, and Jotun Hein. Frnakenstein: multiple target inverse RNA folding. *BMC Bioinform.*, 13:260, 2012.

13 Nono S. C. Merleau and Matteo Smerlak. arnaque: an evolutionary algorithm for inverse pseudoknotted RNA folding inspired by lévy flights. *BMC Bioinform.*, 23(1):335, 2022.

14 R Nussinov and A B Jacobson. Fast algorithm for predicting the secondary structure of single-stranded rna. *Proceedings of the National Academy of Sciences*, 77(11):6309–6313, 1980. doi:10.1073/pnas.77.11.6309.

15 Yann Ponty. *Ensemble Algorithms and Analytic Combinatorics in RNA Bioinformatics and Beyond*. Habilitation à diriger des recherches, Université Paris-Saclay, May 2020. URL: https://theses.hal.science/tel-03219977.

16 Vladimir Reinharz, Yann Ponty, and Jérôme Waldispühl. A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution. *Bioinformatics*, 29(13):i308–i315, June 2013. doi:10.1093/bioinformatics/btt217.

17 Matan Drory Retwitzer, Vladimir Reinharz, Alexander Churkin, Yann Ponty, Jérôme Waldispühl, and Danny Barash. incaRNAfbinv 2.0: a webserver and software with motif control for fragment-based design of RNAs. *Bioinformatics*, 36(9):2920–2922, January 2020. doi:10.1093/bioinformatics/btaa039.

18 Frederic Runge, Danny Stoll, Stefan Falkner, and Frank Hutter. Learning to design RNA. In *Proceedings of ICLR 2019*, 2019.

19 Michael Schnall-Levin, Leonid Chindelevitch, and Bonnie Berger. Inverting the viterbi algorithm: an abstract framework for structure design. In *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 904–911. ACM, 2008.

20 Douglas H. Turner and David H. Mathews. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Research*, 38(suppl_1):D280–D282, October 2009. doi:10.1093/nar/gkp892.

21 Hua-Ting Yao, Cedric Chauve, Mireille Regnier, and Yann Ponty. Exponentially few RNA structures are designable. In *ACM-BCB 2019 - 10th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 289–298, Niagara-Falls, United States, September 2019. ACM Press. doi:10.1145/3307339.3342163.

22 Hua-Ting Yao, Jérôme Waldispühl, Yann Ponty, and Sebastian Will. Taming Disruptive Base Pairs to Reconcile Positive and Negative Structural Design of RNA. In *Proc. of the 25th Annual International Conferences on Computational Molecular Biology (RECOMB'21)*, 2021. URL: https://inria.hal.science/hal-02987566.

23 Joseph N. Zadeh, Brian R. Wolfe, and Niles A. Pierce. Nucleic acid sequence design via efficient ensemble defect optimization. *Journal of Computational Chemistry*, 32(3):439–452, 2011. doi:10.1002/jcc.21633.

A NP-completeness of general separability (Proof of Theorem 3)

Separability is clearly in NP, since any coloring (certificate) can be checked in linear time. We prove hardness by reduction from Bin Packing which we formulate as an interval packing problem.

▶ **Problem 5** (Interval Packing).
Input: *set of pairwise distinct integers* $A = \{a_1, \cdots, a_n\}$*, integers* k *and* B
Output: *function* x *from* A *to intervals of* $[0, kB-1[$ *such that:*
- $x(a_i)$ *is an interval of size* a_i
- $x(a_i)$ *and* $x(a_j)$ *are disjoint for* $i \neq j$
- $x(a_i)$ *does not contain both* $jB-1$ *and* jB *for any* i, j.

This is a reformulation of Bin Packing: fitting items for a total size of B is equivalent to finding a partition of a size-B interval into smaller intervals. The problem remains NP-hard even when input integers are encoded in unary (which corresponds to the fact that Bin Packing is strongly NP-hard). We further require that all items have size $a_i \geq 5$

Object and border gadgets. We first give the main gadgets for our reduction, see figure 13 for more details.

▶ **Definition 6.** *An* object gadget *of size* $q \geq 3$ *is a chain of* $q+3$ *nodes* $c_0, \dots c_{q+2}$ *with a child attached to* c_1 *and* c_{q+1} *and leaves attached to all other nodes* c_i.

A period-p border gadget *of size* q *is a chain of* q *nodes* $c_0, \dots c_{q-1}$ *with a child attached to* c_i *for all* $i \equiv 0 \mod p$ *and leaves attached to all other nodes* c_i.

▶ **Proposition 11.** *If an object gadget of size* q *appears in a tree with a separated coloring* χ*, with* $\ell = \min\{L(c_i) \mid 1 \leq i \leq q\}$ *such that*
- *there are* 🔘 *nodes at levels* $\ell+2$ *and* $\ell+(q+2)$
- *there are leaves at levels* $\ell+i$ *for all* $1 \leq i \leq q+3$, $i \neq 2, q+2$.

If a period-p *gadget of size* q *appears in a tree with a separated coloring* χ*, with the root at level* ℓ*, then there exists some direction* $d \in \{-1, 1\}$ *such that*
- *there are* 🔘 *nodes at levels* $\ell + d \cdot i + 1$ *for all* $1 \leq i \leq q$, $i \equiv 0 \mod p$*;*

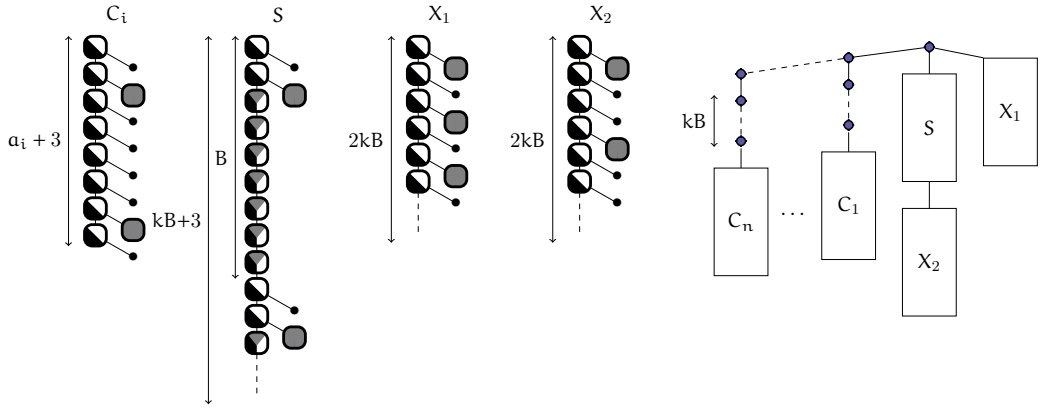


Figure 13 Left: details of the four main parts of the reduction, i.e. an object gadget C_i of size a_i (in this example with $a_i = 5$), border gadgets X_1 and X_2 with respective periods 2 and 3, and the separator chain S). Right: general layout of the tree built in the reduction.

- *there are leaves at levels* $\ell + d \cdot i$ *for all* $1 \leq i \leq q + 3$, $i \not\equiv 0 \mod p$.

Proof. First note that in either gadget, all nodes c_i have the same non-◍ color. Indeed, nodes with a leaf attached or a leaf sibling cannot be ◍, so all c_i are ● or ○. Furthermore, by the proper coloring constraints, consecutive nodes must be of the same color, so all c_i have the same color. Thus, writing ℓ_r for the root level, we have that the level below each node c_i is $\ell_r + di$, with $d = 1$ if the whole chain is ● and $d = -1$ otherwise.

Furthermore, all nodes attached to the chain must be ◍ by the proper coloring constraints. This directly gives the desired property for border gadgets. For object gadgets, the minimum level ℓ along the chain is either ℓ_r (if $d = 1$) or $\ell_r - q - 3$ (if $d = -1$), and in both cases, for each level $\ell + i$ with $1 \leq i \leq q + 3$, there is either a ◍ node ($i = 2$ or $i = q + 2$) or a leaf (otherwise). ◀

Reduction. Given an instance A, k, B of Interval Packing, we build a tree T as follows:

- We start with a chain P of $n + 1$ nodes denoted $p_0, \ldots, p_n$.
- For each $i \geq 1$ we attach a chain (denoted P_i) of Bk nodes to p_i, and an object gadget C_i of size a_i to the end of the chain.
- We attach a period-2 border gadget of size $2kB$ to p_0, denoted X_1.
- We attach a chain S of $kB + 3$ nodes to p_0 with:
  - a leaf to the $(iB + 1)$st node of S for each $0 \leq i \leq k$,
  - a second child, called *separator*, to the $(iB + 2)$nd node of S for each $0 \leq i \leq k$,
  - a period-3 border gadget of size $2kB$ at the end of S, denoted X_2.

We will now show that there exists a solution for unary bin packing if and only one can find a separated coloring for T.

From interval packing to separated coloring. In this section, we consider an interval packing x assigning an interval of $[0, kB - 1[$ to each item a_i. We write x_i such that $x(a_i) = [x_i, x_i + a_i - 1[$, and we color the tree T as follows (see Figure 14):

- All nodes c_i in object gadgets, all non-separator nodes in S and all nodes c_i in X_1 are colored ●,
- All nodes c_i in X_2 are colored ○.
- The first three nodes of P_i are colored ○ ◍ ●, and the last x_i nodes of P_i are colored ● (note that P_i has length $kB \geq x_i + 3$ since $x_i + a_i < kB$ and $a_i \geq 5$).

- All remaining nodes are colored ●.

We show that this coloring is separated, in particular, we show that the level of each ● node is of one of the following types, and that leaves are *not* of these types:

a. 0, 2 and $kB + 2$
b. $x_i + 2$ for each $1 \leq i \leq n$
c. $j - 1$ for $j \leq 0$, $j \equiv 0 \mod 2$
d. $kB + j + 4$ for $j \geq 0$, $j \equiv 0 \mod 3$

For the chain P, all nodes are ● and have level 0 (type a). For each P_i, there are ● nodes at levels -1 and 0 (types a and c), and the chain ends at level x_i. For each object gadget C_i, there are ● nodes at levels $x_i + 2$ (type b), and $x_i + a_i + 2$ (type b or a, since this corresponds to the start of the next interval or to $kB + 2$). There are also leaves in C_i at each level $x_i + j$ for $j = 1, 3, 4, \ldots a_i, a_i + 1, a_i + 3$ which are all values between 1 and $kB + 3$ and indeed do not correspond to any of the four types above. For gadget X_1, there are ● nodes at odd levels from -1 down to $-2kB + 1$ (type c), and leaves at even negative levels. For the chain S, there are ● nodes attached at levels $iB + 2$ for each $0 \leq i \leq k$, which are necessarily of the form $x_i + 2$ (type b) for some i (since each iB must be the start of some interval of x). Leaves in S are at level 1 and $kB + 1$, which are not of any type (in particular for type b, this is true since $a_i \geq 5$). Finally, for gadget X_2, the ● nodes are of type d, and the leaves occupy remaining levels beyond $kB + 4$.

From separated coloring to interval packing

Suppose now that T admits a separated coloring χ, and consider the gadget X_1. Its root is at level $\ell_{X1} \in \{-1, 0, 1\}$, and by Proposition 11, there exists some $d_{X1} \in \{-1, 1\}$ such that, for each level $\ell_{X1} + d_{X1} j$, there is a leaf (for even j) or a ● node (odd j). Without loss of generality, we assume that $d_{x1} = -1$ (i.e., the chain in X_1 is ○): if this is not the case we swap ○ and ● colors overall. Thus, there are leaves and ● nodes at alternating levels between -2 and $-2kB + 1$ (at least).

Consider the chain S. For any $0 \leq i \leq k$, the $(iB + 2)$nd node of the chain cannot be ● (since it has a leaf sibling) so one of its two children must be ●. We write $s_0 \leq s_1 \leq \ldots \leq s_k$ for the levels of such ● nodes in ascending order: from the position of the nodes we have $s_{j+1} \leq s_j + B$. Furthermore, $s_0 \leq 3$ and $s_k \leq kB + 3$ (using the distances to the root).

Consider now X_2. Its root is at most one level away from a separator, so at level ℓ_{X2} with $s_0 - 1 \leq \ell_{X2} \leq s_k + 1$. By Proposition 11, there exists some $d_{X2} \in \{-1, 1\}$ such that, for each level $\ell_{X2} + d_{X2} j$ with $1 \leq j \leq 2kB$, there is a ● node ($i \equiv 0 \mod 3$) or a leaf (otherwise). In particular, we necessarily have $d_{x2} = 1$, since otherwise there would be two consecutive ● levels among levels $\{-2, -3, -4\}$, which would raise a conflict with X_1.

For any $i \in [1, n[$, consider object gadget C_i. Its minimum level is ℓ_i with $-kB - n - a_i - 3 \leq \ell_i \leq kB + a_i + n + 3$, and by Proposition 11, for each level $\ell_i + j$ with $1 \leq j \leq a_i + 3$, there is a ● node ($j = 2, a_i + 2$) or a leaf (otherwise). In particular, $\ell_i \geq s_0 - 5$ (as otherwise there would be consecutive leaves at consecutive levels under $s_0 - 2$, in conflict with X_1) and $\ell_i + a_j \leq s_k + 5$ (otherwise there would be leaves at consecutive levels higher than $s_k + 3$, in conflict with X_2). Finally, since levels s_0 and s_k have ● nodes and $a_i \geq 5$, then for i such that $\ell_i \leq s_0 - 2$, we have $\ell_i = s_0 - 2$. Similarly, for i such that $\ell_i + a_i + 2 \geq s_k$, we have $\ell_i + a_i + 2 = s_k$. And for any i and j, if $\ell_i + 2 \leq s_j \leq \ell_i + a_i + 2$, we have $s_j \in \{\ell_i + 2, \ell_i + a_i + 2\}$.

Pick any two object gadgets $C_i, C_{i'}$ with $\ell_i \leq \ell_{i'}$. Then $\ell_i \neq \ell_{i'}$ (otherwise, since $a_i \neq a_{i'}$, there would be a conflict at level $\ell_i + \min\{a_i, a_{i'}\} + 2$), and $\ell_{i'} \geq \ell_i + a_i$ (otherwise, there would be a conflict at level $\ell_{i'} + 2$).

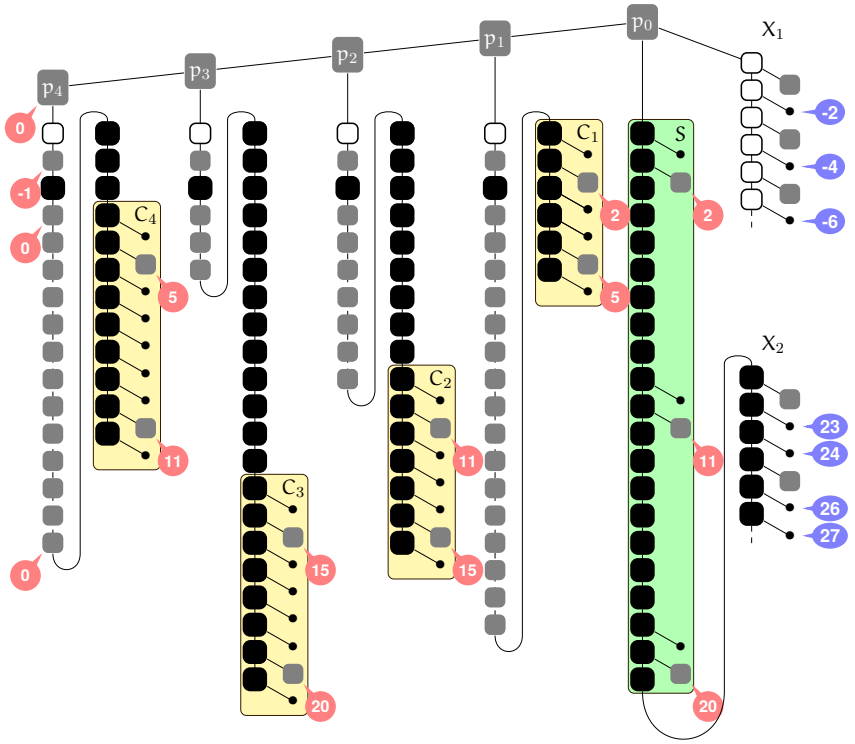


Figure 14 Example of the reduction with $n = 4$ items with sizes $\{3, 4, 5, 6\}$ to be sorted into $k = 2$ size-9 bins. A separated coloring is shown, corresponding to the solution $\{3, 6\}, \{4, 5\}$ (a selection of leaf and ● levels are depicted). Each item is mapped into a branch P_i followed by an object gadget C_i, containing 2 ● nodes separated by the size of the item. Leaves in object gadget enforce that any two gadgets may overlap only if the ● nodes are aligned. The bins are implemented using the separator sequence S, with ● nodes at every Bth position, enforcing that series of consecutive items are packed into size-B bins. Finally, border gadgets X_1 and X_2 may not overlap with any other gadget, and enforce that all object gadgets and separators are packed together in a size-kB range of levels.

We now have all the tools to build an interval packing. We write $x_i = \ell_i - s_0 + 2$ and $\sigma_j = s_j - s_0$. By the remarks above, we have that intervals $[x_i, x_i + a_i - 1[$ are pairwise disjoint. Furthermore, they are all included in interval $[0, \sigma_k - 1[$. Since they have total size $\sum_{i=1}^{n} a_i = kB$ and $\sigma_k = s_k - s_0 \leq kB$, we have $\sigma_k = kB$, which is only possible with a fully ● chain S: so we get $\sigma_j = jB$ for all $0 \leq j \leq k$. And finally, if $\sigma_j \in [x_i, x_i + a_i - 1[$, then $\ell_i + 2 \leq s_j \leq \ell_i + a_i + 2$ which yields $s_j \in \{\ell_i + 2, \ell_i + a_i + 2\}$. This translates into $\sigma_j \in \{x_i, x_i + a_i\}$, so necessarily $\sigma_j = x_i$ and $\sigma_j - 1 \notin [x_i, x_i + a_i[$. Overall gadget levels relative to the first separator s_0 give a valid partition of $[0, kB - 1[$ into pairwise disjoint size-a_i intervals non-overlapping block border positions jB, so they give a valid Interval Packing solution.

B Non-separable target w/o isolated BPs (Proof of Proposition 10)

We start with the following remark:

▶ **Proposition 2.** *If $u_0, \ldots, u_k$ is a path in T and each u_i for even i has a leaf attached to it then, for any coloring χ of the path, we have $\chi(u_0) \in \{●, ○\}$ and $\chi(u_i) = \chi(u_0)$ for all i.*

Proof. Indeed, by the proper coloring constraint, every node with an attached leaf or with a leaf sibling may not be ●, so all $\chi(u_i) \in \{●, ○\}$ for all i. Moreover, there can be no direct edge between ○ and ● nodes, so $\chi(u_i) = \chi(u_{i-1})$ for all i which gives the desired property by induction. ◀

We now build a non-separable instance I without size-1 helix nor $(m_{3\bullet}, m_5)$ motif. Let $a \geq 2$ and $b \geq 2$ be even numbers. Let $T(a, b)$ be the gadget from Fig 9, containing a length-a path from the to an internal node denoted t, and three length-b branches attached to t. Further attach a leaf to every node at an even distance from the root (except t itself). Note that all helices in $T(a, b)$ have length 2. The *level* of a copy of some $T(a, b)$ gadget is the level reached under node t of this gadget.

We build the instance I as a tree containing 5 copies of the gadget $T(a, b)$, precisely $I = (((T[10, 100], T[20, 100])), ((T[30, 100], T[40, 100])), T[50, 100])$.

First note that for a copy of gadget $T(a, b)$ at level ℓ in any separable coloring, there is a ● node at level ℓ, since the node t has three children and at least one must be ●. Also, there exist two integers u, v such that, for every $x \in [1, b[$, there is a leaf at level $\ell + ux$ if x is odd, and level $\ell + vx$ if x is even. Indeed, pick one gray child U of t, and one non-gray child V. All vertices under U form an all-white or all-black branch by Proposition 2 (we let respectively $u = -1$ and $u = 1$), and vertices at levels $l + u, l + 3u, \ldots, l + bu$ (or $l + (b-1)u$) have a pending leaf. We similarly define $v = 1$ if V is black and $v = -1$ if V is white, and vertices at levels $l + 2v, l + 4v, \ldots, l + bv$ (or $l + (b-1)v$) have a pending leaf. From the above, if there are ● nodes at levels ℓ_1 and ℓ_2 with $\ell - b \leq \ell_1 < \ell < \ell_2 \leq \ell + b$, then $\ell_1 \neq \ell_2 \mod 2$ (since otherwise, one of ℓ_1, ℓ_2 could be written as $\ell + ux$ with even x, so that level would be a leaf level).

Aiming at a contradiction, assume that I admits a separable coloring. Let $\ell_1 \leq \ell_2 \leq \ell_3 \leq \ell_4 \leq \ell_5$ be the levels of all five copies of the $T[a, b]$ gadgets of I, in ascending order. Then from the length of the branches from the root, we have $\ell_i \in [-50, 50]$ and $\ell_i \neq \ell_j$. Then by the remark above applied to the gadget with level ℓ_2, we have $\ell_1 \not\equiv \ell_3 \mod 2$, and similarly using gadgets with level ℓ_4 we have $l_3 \not\equiv l_5 \mod 2$ and $l_1 \not\equiv l_5 \mod 2$, leading to a contradiction (any three integers such as ℓ_1, ℓ_3 and ℓ_5 may not have pairwise distinct parities).

C Leveraging random generators at fixed modular levels into a uniform random generation of separated sequences

▶ **Theorem 12.** Uniform Modulo Separated Generation *can be performed in an average-case complexity that is Fixed Parameter Tractable for the modulus parameter m.*

We consider a rejection-based approach, which starts by precomputing all $\#\text{Designs}_{\xi_L}$ in time $\Theta(n.m.2^m)$ (see Section 4.2), and accumulates them into $\mathcal{Z}_m := \sum_{\xi'_L \subseteq [0,m[} \#\text{Designs}_{\xi'_L}$. It then iterates the following steps until a suitable sequence is returned:

1. Choose some $\xi_L \subset [0, m[$ with probability $\mathbb{P}(\xi_L) = \#\text{Designs}_{\xi_L}/\mathcal{Z}_m$
2. Generate a ξ_L separated sequence w
3. Compute the number Ξ_w of $\xi'_L \subset [0, m[$ such that w is ξ'_L separated
4. Accept/return w with probability $1/\Xi_w$; Reject/restart from **1.** otherwise.

Due to the full reset on each rejection, the emission probability p_w of any suitable w does not depend on the prior sequence of rejections (folklore, proven in [15, pp 77]), and we have:

$$p_w \propto \sum_{\substack{\xi_L \text{ such that } w \\ \text{is } \xi_L \text{ separated}}} \mathbb{P}(\xi_L) \times \mathbb{P}(w \mid \xi_L) \times \frac{1}{\Xi_L} = \sum_{\substack{\xi_L \text{such that } w \\ \text{is } \xi_L \text{ separated}}} \frac{\#\text{Designs}_{\xi_L}}{\mathcal{Z}_m} \times \frac{1}{\#\text{Designs}_{\xi_L}} \times \frac{1}{\Xi_w}$$

Some terms directly cancel out and, by definition, we have $\sum_{\substack{\xi_L \text{such that } w \\ \text{is } \xi_w \text{ separated}}} 1 = \Xi_w$. It follows that $p_w \propto 1/\mathcal{Z}_m$, a term that no longer depends on w, from which we conclude that the generation is uniform.

Complexity-wise, a prior accumulation of the 2^m terms $\#\text{Designs}_{\xi_L}$, each smaller than 4^m, into a suitable data structure (see Lorenz and Ponty [11] for details) enables a random choice of ξ_L (Step 1.) in $\Theta(n.m)$. Once ξ_L is chosen, the above DP algorithm uniformly generates w in time $\Theta(m.n)$ (Step 2). The computation of Ξ_w (Step 3) is trivial and consists in identifying, in time $\Theta(n+m)$, the subset $\Phi_w \subseteq [0, m[$ of modular levels that are populated by neither leaves nor ● nodes in χ_w. Indeed, those levels represent the only degrees of freedom available while choosing a compatible ξ_L, the others modular values being forced to either ● or leaves. Since such modular values can be independently chosen to be in or out of ξ_L, then we have $\Xi_w = 2^{|\Phi_w|}$. Clearly, we have $\Xi_w \leq 2^m$, so the expectation of the number of (independent) rejections admits an upper bound in 2^m, and the overall average-case complexity is in $\Theta(n.m.2^m)$.