A New Parametrization for Independent Set Reconfiguration and Applications to RNA Kinetics

Laurent Bulteau
LIGM, CNRS, Univ Gustave Eiffel, F77454 Marne-la-vallée, France

Bertrand Marchand
LIX, CNRS UMR 7161, Ecole Polytechnique, Institut Polytechique de Paris, France
LIGM, Marne-la-Valée, France

Yann Ponty
LIX, CNRS UMR 7161, Ecole Polytechnique, Institut Polytechique de Paris, France

Abstract

In this paper, we study the Independent Set (IS) reconfiguration problem in graphs. An IS reconfiguration is a scenario transforming an IS L into another IS R, inserting/removing vertices one step at a time while keeping the cardinalities of intermediate sets greater than a specified threshold. We focus on the *bipartite* variant where only start and end vertices are allowed in intermediate ISs. Our motivation is an application to the *RNA energy barrier* problem from bioinformatics, for which a natural parameter would be the difference between the initial IS size and the threshold.

We first show the para-NP hardness of the problem with respect to this parameter. We then investigate a new parameter, the *cardinality range*, denoted by ρ which captures the maximum deviation of the reconfiguration scenario from optimal sets (formally, ρ is the maximum difference between the cardinalities of an intermediate IS and an optimal IS). We give two different routes to show that this problem is in XP for ρ: The first is a direct $O(n^2)$-space, $O(n^{2\rho+2.5})$-time algorithm based on a separation lemma; The second builds on a parameterized equivalence with the directed pathwidth problem, leading to a $O(n^{\rho+1})$-space, $O(n^{\rho+2})$-time algorithm for the reconfiguration problem through an adaptation of a prior result by Tamaki [20]. This equivalence is an interesting result in its own right, connecting a reconfiguration problem (which is essentially a *connectivity* problem within a *reconfiguration network*) with a *structural* parameter for an auxiliary graph.

We demonstrate the practicality of these algorithms, and the relevance of our introduced parameter, by considering the application of our algorithms on random small-degree instances for our problem. Moreover, we reformulate the computation of the energy barrier between two RNA secondary structures, a classic hard problem in computational biology, as an instance of bipartite reconfiguration. Our results on IS reconfiguration thus yield an XP algorithm in $O(n^{\rho+2})$ for the energy barrier problem, improving upon a partial $O(n^{2\rho+2.5})$ algorithm for the problem.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Applied computing → Bioinformatics

Keywords and phrases reconfiguration problems - parameterized algorithms - RNA bioinformatics - directed pathwidth

Digital Object Identifier 10.4230/LIPIcs.IPEC.2021.11

Related Version *Full Version*: `https://hal.inria.fr/hal-03272963`

Supplementary Material *Software*: `https://gitlab.inria.fr/bmarchan/bisr-dpw`

Acknowledgements The authors would like to thank H. Tamaki and Y. Kobayashi for fruitful email exchanges.

16th International Symposium on Parameterized and Exact Computation (IPEC 2021).
Editors: Petr A. Golovach and Meirav Zehavi; Article No. 11; pp. 11:1–11:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Reconfiguration problems. Reconfiguration problems informally ask whether there exists, between two *configurations* of a system, a *reconfiguration pathway* entirely composed of *legal* intermediate configurations, connected by legal *moves*. In a thoroughly studied sub-category of these problems, configurations correspond to *feasible solutions* of some *optimization problem*, and a feasible solution is legal when its quality is higher than a specified threshold.

Examples of optimization problems for which reconfiguration versions have been studied include Dominating Set, Vertex Cover, Shortest Path or Independent Set, which is our focus in this article. Associated complexities range from polynomial (see [23] for examples) to NP-complete (for bipartite independent set reconfiguration [13]), and even **PSPACE**-complete for many of them [13, 9]. Such computational hardness motivates the study of these problems under the lens of *parametrized complexity* [18, 14, 15, 9], in the hope of identifying tractable sub-regimes. Typical parameters considered by these studies focus on the value of the *quality threshold* (typically a *solution size* bound) defining legal configurations and the length of the reconfiguration sequences.

Directed pathwidth. *Directed pathwidth*, originally defined in [1] and attributed to Robertson, Seymour and Thomas, represents a natural extension of the notions of path-width and path decompositions to directed graphs. Like its undirected restriction, it may alternatively be defined in terms of *graph searching* [24], *path decompositions* [4, 6] or *vertex separation number* [11, 20]. An intuitive formulation can be stated as the search for a visit order of the directed graph, using as few active vertices as possible at each step, and such that no vertex may be deactivated until all its in-neighbors have been activated. Although an FPT algorithm is known for the undirected pathwidth [2], it remains open whether computing the directed pathwidth admits a FPT algorithm. XP algorithms [20, 11] are known, and have been implemented in practice [19, 12].

RNA energy barrier. RNAs are single-stranded biomolecules, which fold onto themselves into 2D and 3D structures through the pairing of nucleotides along their sequence [22]. Thermodynamics then favors low-energy structures, and the RNA energy barrier problem asks, given two structures, whether there exists a re-folding pathway connecting them that does not go through unlikely high-energy intermediate states [17, 21]. Interestingly, the problem falls under the wide umbrella of reconfiguration problems described above, namely the reconfiguration of solutions of optimization problems (here, energy minimization). An important specificity of the problem is that the probability of a refolding pathway depends on the energy difference between intermediate states and the *starting point* rather than the absolute energy value. Another aspect of this problem is that, since some pairings of the initial structure may impede the formation of new pairings for the target structure, it induces a notion of *precedence constraints*, and may therefore also be treated as a *scheduling* problem, as carried out in [8, 10].

Problem statement. In our work, we focus on independent set reconfigurations where only vertices from the start or end ISs (L and R) are allowed within intermediate ISs. This amounts to considering the induced subgraph $G[L \cup R]$, bipartite by construction. We write $\alpha(G)$ for the size of a maximum independent set of G (recall that $\alpha(G)$ can be computed in polynomial time on bipartite graphs).

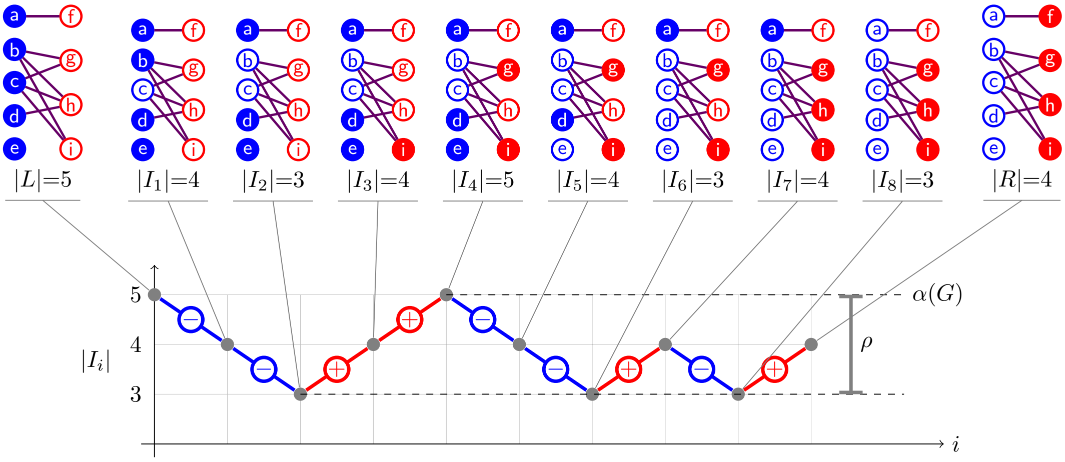

Figure 1 Example of a bipartite independent set reconfiguration from vertices in L (blue) to R (red). Selected vertices at each step have a filled background. All intermediate ISs have size at least 3, and the optimal IS has size 5, so this scenario has a range of 2; it can easily be verified that it is optimal.

Bipartite Independent Set Reconfiguration (BISR)
Input: Bipartite graph $G = (V, E)$ with partition $V = L \cup R$; integer ρ
Parameter: ρ
Output: True if there exists a sequence $I_0 \cdots I_\ell$ of independent sets of G such that

- $I_0 = L$ and $I_\ell = R$;
- $|I_i| \geq \alpha(G) - \rho, \forall i \in [0, \ell]$;
- $|I_i \triangle I_{i+1}| = 1, \forall i \in [0, \ell - 1]$.

False otherwise.

Figure 1 shows an example of an instance of BISR and a possible reconfiguration pathway. We introduce the *cardinality range* (or simply *range*) $\rho = \max_{1 \leq i \leq \ell} \alpha(G) - |I_i|$ as a natural parameter for this problem, since it measures a distance to optimality. As mentioned above, the related parameter in RNA reconfiguration is the *barrier*, denoted k, and defined as $k = \max_{1 \leq i \leq \ell} |L| - |I_i|$. Intuitively, k measures the size difference from the starting point rather than from an "absolute" optimum. Note that $k = \rho - (\alpha(G) - |L|)$, so one has $0 \leq k \leq \rho$. Both parameters are obviously similar for instances where L is close to being a maximum independent set, which is generally the case in RNA applications, but in theory the range ρ can be arbitrarily larger than the barrier k.

Our results. We first prove that in general, the barrier k may not yield any interesting parameterized algorithm, since BISR is Para-NP-hard for this parameter. We thus focus on the range parameter for Bipartite Independent Set Reconfiguration, and prove that it is in XP by providing two distinct algorithmic strategies to tackle it.

Our first algorithmic strategy stems from a parameterized equivalence we draw between BISR and the problem of computing the directed pathwidth of directed graphs. Within this equivalence, the range parameter ρ maps exactly to the directed pathwidth. This allows to apply XP algorithms for Directed Pathwidth to BISR while retaining their complexity, such as the $O(n^{\rho+2})$-time, $O(n^{\rho+1})$-space algorithm from Tamaki [20] (with $n = |V|$). This equivalence between directed pathwidth and bipartite independent set reconfiguration is itself an interesting result, as it connects a *structural* problem, whose parameterized complexity is open, with a reconfiguration problem of the kind that is routinely studied in parameterized complexity [18, 14, 15, 9].

We also present another more direct algorithm for BISR, with a time complexity of $O(n^{2\rho}\sqrt{n}m)$ (with $m = |E|$) but using only $O(n^2)$ space. It relies on a separation lemma involving, if it exists, a *mixed* maximum independent set of G containing at least one vertex from both parts of the graph. In the specific case of bipartite graphs arising from RNA reconfiguration, we improve the run-time of the subroutine computing a mixed MIS to $O(n^2)$ (rather than $O(\sqrt{n}m)$), with a dynamic programming approach.

We present benchmark results for both algorithms, on random instances of general bipartite graphs as well as instances of the RNA Energy Barrier problem. The approach based on directed pathwidth yields reasonable solving times for RNA strings of length up to ~ 150.

Outline. To start with, Section 2 presents some previously known results related to BISR, as well as some alternative formulations or parameters. Then, Section 3 shows that BISR is in fact equivalent to the computation of *directed pathwidth* in directed graphs. We first present a parameterized reduction from bipartite independent set reconfiguration to an input-restricted version, on graphs allowing for a perfect matching. Then, this version of the problem is shown to be simply equivalent to the computation of directed pathwidth on general directed graphs.

Section 4 presents our direct algorithm for bipartite independent set reconfiguration. More precisely, Section 4.2 presents the separation lemma on which the divide-and-conquer approach of the algorithm is based, while Section 4.3 details the algorithm and its analysis.

To finish, Section 5 explains some optimizations specific to RNA reconfiguration instances, and presents our numerical results.

2 Preliminaries

Previous results. Bipartite Independent Set Reconfiguration was proven NP-complete in [13], through the equivalent k-Vertex Cover Reconfiguration problem. Formulated in terms of RNAs, and restricted to secondary structures (i.e. the subset of bipartite graphs that can be obtained in RNA reconfiguration instances), it was independently proven NP-hard in [17]. To the authors' knowledge, its parameterized complexity remains open.

Independent set reconfiguration in an unrestricted setting (allowing vertices which are outside from the start or end independent sets, i.e. in possibly non-bipartite graphs) when parameterized by the minimum allowed size of intermediate sets has been proven W[1]-hard [18, 9], and fixed-parameter tractable for planar graphs or graphs of bounded degree [14]. Whether this more general problem is in XP for this parameter remains open. We note that in this setting, parameter ρ seems slightly less relevant since it involves computing a maximal independent set in a general graph (i.e. testing if there exists a reconfiguration from $\emptyset$ to $\emptyset$ with range ρ is equivalent to deciding if $\alpha(G) \geq \rho$).

As for algorithms for BISR, the closest precedent is an algorithm by Thachuk et al. [21]. It is restricted to RNA secondary structure conflict graphs, and additionally to conflict graphs for which both parts L and R are *maximum independent sets* of G. In this restricted setting, although it is not stated as such, [21] provides an XP algorithm with respect to the barrier parameter k which then coincides with the range parameter ρ that we introduce.

Restriction to the monotonous case. A reconfiguration pathway for BIPARTITE INDEPENDENT SET RECONFIGURATION is called *monotonous* or *direct* if every vertex is added or removed exactly once in the entire sequence. The length of a monotonous sequence is therefore necessarily: $\ell = |L \cup R| = |L| + |R|$.

Theorem 2 from [13] tells us that if G, ρ is a yes-instance of bipartite independent set reconfiguration, then there exists a *monotonous* reconfiguration between L and R respecting the constraints. We will therefore restrict without loss of generality our study to this simpler case. In the more restricted set studied in [21], this was also independently shown.

Hardness for the barrier parameter. In the general case where L is not necessarily a maximal independent set, the range and barrier parameters (respectively ρ and $k = \rho - (\alpha(G) - |L|)$ may be arbitrarily different. The following result motivates our use of parameter ρ for the parameterized analysis of BISR.

▶ **Proposition 1.** BISR *is* Para-NP*-hard for the energy barrier parameter (i.e.* NP*-hard even with* $k = 0$*).*

Proof. We use additional vertices in R to prove this result. Informally, such a vertex may always be inserted first in a realization: it improves the starting IS from $|L|$ to $|L| + 1$, so the lower bound on the rest of the sequence is shifted from $|L| - k$ to $|L| - (k - 1)$, effectively reducing the barrier without simplifying the instance. Thus, we build a reduction from the general version of BISR: given a bipartite graph G with parts L and R and an integer ρ, we construct a new instance G' with parts $L' = L$ and R' equal to $R \cup N_R$ and $\rho' = \rho$. N_R is composed of $|L| - (\alpha(G) - \rho)$ isolated vertices (we can assume without loss of generality that this quantity is non-negative, otherwise (G, ρ) is a trivial no-instance), completely disconnected from the rest of the graph.

Note that $\alpha(G') = \alpha(G) + |N_R| = |L| + \rho$, so the barrier in (G', ρ') is $k = \rho - (\alpha(G') - |L|) = 0$. A realization for (G, ρ) can be transformed into a realization for (G', ρ) by inserting vertices from N_R first, and conversely, vertices from N_R can be ignored in a realization for (G', ρ) to obtain a realization for (G, ρ). Therefore, since BISR is NP-Complete, it is also Para-NP-hard w.r.t the barrier k. ◀

Permutation formulation and ρ-realizations. An equivalent representation of a monotonous reconfiguration pathway $I_0 \dots I_\ell$ from L to R for a graph G is a *permutation* P of $L \cup R$. The i-th vertex of the permutation is the vertex that is *processed* (i.e. added or removed) between I_{i-1} and I_i (this formulation lightens the representation of a solution, from a list of vertex sets to a list of vertices). Given a subset X of vertices, we write $\delta(X) = |L \cap X| - |R \cap X|$ and $I(X) = (L \setminus X) \cup (R \cap X) = L \Delta X$ for the set obtained from L after processing vertices from X. Then $|I(X)| = |L| - \delta(X)$. We say that X is *licit* if $I(X)$ is an independent set. For any prefix p of P of length i, we write $V(p)$ (or simply p if the context is clear) for the set of vertices appearing in p, and $I_i = I(V(p))$. A permutation P is licit if $V(p)$ is licit for each prefix p of P; note that P is licit if and only if $\forall r \in R$, the neighborhood $N(r)$ of r in G appears before r in P. Last, we say that P is a *ρ-realization* that is licit and such that for each prefix p, $|I(p)| \geq \alpha(G) - \rho$ (i.e. $\delta(V(p)) \leq \rho + |L| - \alpha(G)$).

3 Connection to Directed Pathwidth

3.1 Definitions

Parameterized reduction. In this section, we provide a definition of directed pathwidth, and then prove its parameterized equivalence to the bipartite independent set reconfiguration problem. We say two problems $\mathcal{P}_1$ and $\mathcal{P}_2$ are parametrically equivalent when there exists both a *parameterized reduction* from $\mathcal{P}_1$ to $\mathcal{P}_2$ and another from $\mathcal{P}_2$ to $\mathcal{P}_1$. A parameterized reduction [5] from problem $\mathcal{P}$ to problem $\mathcal{Q}$ is a function φ from instances of $\mathcal{P}$ to instances of $\mathcal{Q}$ such that (i) $\varphi(x)$ is a yes-instance of $\mathcal{Q} \Leftrightarrow x$ is a yes-instance of $\mathcal{P}$, (ii) $\varphi(x)$ can be computed in time $f(k) \cdot |x|^{O(1)}$, where k is the parameter of x, and (iii) if k is the parameter of x and k' is the parameter of $\varphi(x)$, then $k' \leq g(k)$ for some (computable) function g.

Interval representation. Our definition of directed pathwidth relies on interval embeddings. Alternative definitions can be found, for instance in terms of directed path decomposition or directed vertex separation number [24, 20, 11].

▶ **Definition 2** (Interval representation). *An* interval representation *of a directed graph H associates each vertex $u \in H$ with an interval $I_u = [a_u, b_u]$, with a_u, b_u integers. An interval representation is* valid *when $(u, v) \in E \Rightarrow a_u \leq b_v$. I.e, the interval of u must start before the interval of v ends. If m, M are such that $\forall u, m \leq a_u, b_u \leq M$, we define the* width *of an interval representation as* $\max_{m \leq i \leq M} |\{u | i \in I_u\}|$

▶ **Definition 3** (directed pathwidth). *The* directed pathwidth *of a directed graph H is the minimum possible width of a valid interval representation of H. We note this number $dpw(H)$.*

Nice interval representation. An interval representation is said to be *nice* when no more than one interval bound is associated to any given integer, and the integers associated to interval bounds are exactly $[1 \dots 2 \cdot |V(H)|]$. Any interval representation may be turned into a nice one without changing the width by introducing new positions and "spreading events". See the full version of the article for more details.

Directed graph from perfect matching. Given a bipartite graph G allowing for a perfect matching M, we construct an associated directed graph H in the following way: the vertices of H are the edges of the matching, and $(l, r) \to (l', r')$ is an arc of H iff $(l, r') \in G$. Alternatively, H is obtained from G, M by orienting the edges of G from L to R, and then contracting the edges of M. We will denote this graph $H(G, M)$, and simply call it the *directed graph associated to G, M*. Such a construction is relatively standard and can be found in [7, 25], for instance.

3.2 Directed pathwidth ⇔ Bipartite independent set reconfiguration

Perfect matching case. Our main structural result is the following. Its proof, relying on interval representations, is quite straightforward and can be found in the full version of the article.

▶ **Proposition 4.** *Let G be a bipartite graph allowing for a perfect matching M, and let $H(G, M)$ be the directed graph associated to G, M. Then G allows for a ρ-realization iff $dpw\,(H(G, M)) \leq \rho$.*
Conversely, given any directed graph H, there exists a bipartite graph G allowing for a perfect matching M such that $H = H(G, M)$ is the directed graph associated to G, M and G allows for a ρ-realization iff $dpw(H) \leq \rho$.

The first half of Proposition 4 is a parameterized reduction from an input-restricted version of BIPARTITE INDEPENDENT SET RECONFIGURATION to directed pathwidth. The restriction is on bipartite graphs allowing for a perfect matching. The second half is a parameterized reduction in the other direction. In both cases, the parameter value is directly transferred, which allows to retain the same complexity when transferring an algorithm from one problem to the other.

Non-perfect-matching case. In the case where G does not allow for a perfect matching, we construct an equivalent instance G' allowing for a perfect matching M', through the addition of new vertices. Specifically, with a bipartite graph G with sides L, R, a maximum matching M of G, and the set U of unmatched vertices in G, we extend G with $|U|$ new vertices in two sets N_L, N_R, giving a new graph G', with sides $L' = L \cup N_L, R' = R \cup N_R$, in the following way ($M'$ is initialized to M):

- For each $u \in L \cap U$, we introduce a new vertex $r(u) \in N_R$, connect it to *all* vertices of L', and add the edge $(u, r(u))$ to M'.
- Likewise, for each $v \in R \cap U$, we introduce $l(v) \in N_L$, connect it to all vertices of R' and add $(v, l(v))$ to M'.

Note that M' is a perfect matching of the extended bipartite graph G'.

▶ **Proposition 5.** *With G, G' defined as above, we have that G allows for a ρ-realization iff G' allows for a ρ-realization.*

Proof. First note that by König's Theorem, $\alpha(G') = |M'| = |M| + |U| = \alpha(G)$, so it suffices to ensure that any realization for G can be transformed into a realization for G' where independent sets are lower-bounded by the same value, and vice versa.

Let P be any ρ-realization of G, then $P' = N_L \cdot P \cdot N_R$ is a ρ-realization for G', with N_L and N_R laid out in any order. Indeed, P' satisfies the precedence constraint, and any intermediate set I in P' satisfies one of the following cases: $L \subseteq I$, $R \subseteq I$, or I is an intermediate set from P, so in any case it has size at least $\alpha(G) - \rho = \alpha(G') - \rho$.

Conversely, because of the all-to-all connectivity between N_L and R and between L and N_R, a realization for G' needs to have N_L before any vertex from R, and have N_R after all vertices from L. Without loss of generality, it is therefore of the form $N_L \cdot P \cdot N_R$ with P a realization of G, and G allows for a ρ-realization. ◀

The construction above in fact yields a parameterized reduction from BIPARTITE INDEPENDENT SET RECONFIGURATION to its input-restricted version on bipartite graphs, allowing for a perfect matching. This input-restricted version is in turn parametrically equivalent to directed pathwidth by Proposition 4. Hence the following corollary:

▶ **Corollary 6.** BIPARTITE INDEPENDENT SET RECONFIGURATION *is parametrically equivalent to* DIRECTED PATHWIDTH

4 An XP algorithm for independent set reconfiguration

4.1 Definitions

We use the permutation representation of reconfiguration scenarios, i.e. licit permutations of vertices. Note that the intersection, as well as the union, of two licit set of vertices are licit. Given a realization P of G and a set of vertices X, we write $P \cap X$ for the sub-sequence of P consisting of the vertices of X, without changing the order. Likewise, $P \setminus X$ denotes the sub-sequence of P consisting of vertices *not* in X.

A *mixed maximum independent set* I of G is an independent set of G of maximum cardinality containing at least a vertex from both parts. Note that not every bipartite graph contains such a set. A *separator* X is a subset of $L \cup R$ such that $I(X)$ is a mixed maximum independent set of G.

4.2 Separation lemma

The separation lemma on which our algorithm is based is proved using the following "modularity" property of the imbalance functions. Interestingly, it is almost the same property (sub-modularity), on a different quantity (the in-degrees of vertices) on which rely the XP algorithm for directed pathwidth [20].

▶ **Lemma 7** (modularity). *The function associating a licit subset to its corresponding independent set $I(X)$ verifies:*

$$|I(X)| + |I(Y)| = |I(X \cup Y)| + |I(X \cap Y)|$$

Proof. We have $I(X) = (L \setminus X) \cup (R \cap X)$. Therefore, $|I(X)| = |L \setminus X| + |R \cap X| = |L| - |L \cap X| + |R \cap X|$. Furthermore, $|(X \cup Y) \cap L| = |(X \cap L) \cup (Y \cap L)| = |X \cap L| + |Y \cap L| - |X \cap Y \cap L|$, and likewise for R. The result stems from a substraction of one equation to the other, and an addition of $|L|$. ◀

Based on this "modularity", the following separation lemma is shown by "re-shuffling" a solution into another one going through a mixed MIS.

▶ **Lemma 8** (separation lemma). *Let X be a separator of G. If P is a ρ-realization for G, then $(P \cap X) \cdot (P \setminus X)$ is also a ρ-realization for G.*

Proof. Let P be a ρ-realization for G and $P' = (P \cap X) \cdot (P \setminus X)$ a reshuffling, where X is processed first.

Consider p' a prefix of P'. There are two cases:

1. p' is included in (or equal to) $P \cap X$. In this case, $\exists p$ prefix of P such that: $p' = p \cap X$. We therefore have $|I(p')| = |I(p)| + |I(X)| - |I(p \cup X)|$, and since $|I(X)|$ is a maximum independent set of G, $|I(p')| \geq |I(p)| \geq \alpha(G) - \rho$.
2. $P \cap X$ is included in p. In that case, $\exists p$ prefix of P such that $p' = p \cup X$. We have, likewise, $|I(p')| = |I(p)| + |I(X)| - |I(p \cap X)|$ and conclude the same way. ◀

The separation allows for a divide-and-conquer approach: if we identify a separator X in G, i.e. a licit subset of G such that $I(X)$ is a mixed independent set, then we may independently solve the problem of finding a ρ-realization from L to $I(X)$ and then from $I(X)$ to R. If no solution is found for one of them, then the converse of Lemma 8 implies that no ρ-realizations exists for G. The algorithm of the following section is based on this approach.

4.3 XP algorithm

Algorithm details. We present here a direct algorithm for Bipartite Independent Set Reconfiguration, detailed in Algorithm 1. The main function `Realize` is recursive. Its sub-calls arise either from a split with a mixed MIS I (in which case it is called on a smaller graph but with the same parameter), or from the loop over all possible starting points in the case where no separator is found (lines 13-18), in which case the parameter does reduce. The overall runtime is dominated by this loop, and is analyzed in Proposition 9 below.

Mixed MIS algorithm. The sub-routine allowing to find, if it exists, a maximum independent set intersecting both L and R is based on concepts from *matching theory* [16], namely the *Dulmage-Mendelsohn* decomposition [3, 16], as well as the decomposition of bipartite graphs with a perfect matching into *elementary subgraphs* [16](part 4.1). Its full details are described in the full version of the article.

Algorithm 1 XP algorithm for Bipartite Independent Set Reconfiguration.

```
Input  : bipartite graph $G$ (with sides $L$ and $R$), integer $\rho$
Output : a $\rho$-realization for $G$, if it exists
Function Realize($G, \rho$):
    // Terminal cases:
    if $\rho < 0$ then return $\bot$
    if $|L \cup R| = \emptyset$ then return $\emptyset$
    // Isolated vertices:
    if $\exists \ell \in L$ s.t $N(\ell) = \emptyset$) then return Realize($G \setminus \{\ell\}, \rho - 1$) $\cdot\, l$
    if $\exists r \in R$ s.t $N(r) = \emptyset$) then return $r \,\cdot$ Realize($G \setminus \{r\}, \rho - 1$)
    // Trying to find a separator (see full version of the article for details)
    $I$ = MixedMIS($G$)
    if $I \neq \bot$ then
        $S = (L \setminus I) \cup (R \cap I)$                // intermediate point.
        return Realize($G[S], \rho$)$\cdot$ Realize($G[V \setminus S], \rho$)
    else
        // Iterating over all possible start/end point pairs.
        for $(\ell, r) \in L \times R$ do
            if Realize($G \setminus \{\ell, r\}, \rho - 1$)$\neq \bot$ then
                return $\ell \,\cdot$ Realize($G \setminus \{\ell, r\}, \rho - 1$) $\cdot\, r$
        return $\bot$
```

▶ **Proposition 9.** *Algorithm 1 runs in $O(|V|^{2\rho}\sqrt{|V|}|E|)$ time, while using $O(|V|^2)$ space, where ρ is the difference between the minimum allowed and maximum possible independent set size, along the reconfiguration.*

Proof. Let us start with space: throughout the algorithm, one needs only to maintain a description of G and related objects (independent set I, maximum matching M, associated directed graph $H(G, M)$) for which $O(|V|^2)$ is enough.

As for time, let $C(n_1, n_2, \rho)$ be the number of recursive calls of the function *Realize* of Algorithm 1 when initially called with $|L| = n_1$, $|R| = n_2$, and some value of ρ. We will show by induction that $C(n_1, n_2, \rho) \leq (n_1 + n_2)^{2\rho}$. Since each call involves one computation of a maximum matching, this will prove our result.

Given (n_1, n_2, ρ), suppose therefore that $\forall (n_1', n_2', \rho') \neq (n_1, n_2, \rho)$ with $n_1' \leq n_1, n_2' \leq n_2, \rho' \leq \rho$ we have $C(n_1', n_2', \rho') < (n_1' + n_2')^{2\rho'}$

1. If G allows for a mixed maximum independent set, the instance is split into two smaller instances, yielding $C(n_1, n_2, \rho) = C(n_1', n_2', \rho) + C(n_1'', n_2'', \rho)$ with $n_1' + n_1'' = n_1$ and $n_2 = n_2' + n_2''$. And $C(n_1, n_2, \rho) \leq \left((n_1' + n_2')^{2\rho} + (n_1'' + n_2'')^{2\rho}\right) \leq (n_1' + n_1'' + n_2' + n_2'')^{2\rho} \leq (n_1 + n_2)^{2\rho}$.

2. else, we have the following relation: $C(n_1, n_2, \rho) = n_1 n_2 \cdot C(n_1 - 1, n_2 - 1, \rho - 1)$. Which yields:

$$\begin{aligned} C(n_1, n_2, \rho) &= n_1 n_2 \cdot C(n_1 - 1, n_2 - 1, \rho - 1) \\ &\leq n^2 \cdot n^{2(\rho-1)} \quad \text{by induction hypothesis} \\ &\leq n^{2\rho} \end{aligned}$$ ◀

The exponential part ($O(n^{2\rho})$) of the worst case complexity of Algorithm 1 is in fact tight, as it is met with a complete bi-clique $K_{n,n}$ with sides of size n. Indeed, in this case, no mixed MIS is found in any of the recursive calls.

5 Benchmarks and Applications

In this section, we report benchmark results for both algorithmic approaches. We first explain some details about the algorithm we implemented for directed pathwidth. Then, we present a general benchmark of our algorithms on random (Erdös-Rényi) bipartite graphs. Last, we give some background related to RNA bioinformatics and the application of our algorithm to the barrier energy problem.

Code availability. The code used for our benchmarks, including a Python/C++ implementation of our two algorithms, is available at `https://gitlab.inria.fr/bmarchan/bisr-dpw`

5.1 Implementation details

Directed pathwidth. We implemented and used an algorithm from Tamaki [20], with a runtime of $O(n^{\rho+2})$. This algorithm was originally published in 2011 [20]. In 2015, H.Tamaki and other authors described this algorithm as "flawed" in [11], and replaced it with another XP algorithm for directed pathwidth, with a run-time of $O(\frac{mn^{2\rho}}{(\rho-1)!})$.

Upon further analysis from our part, and discussions with H. Tamaki and the corresponding author of [11], it appears a small modification allowed to make the algorithm correct. In a nutshell, the algorithm involves *pruning* actions, and these need to be carried out *as soon as they are detected*. In [20], temporary solutions were accumulated before a general pruning step. With this modification, the analysis presented in [20] applies without modification, and yields a time complexity of $O(n^{\rho+2})$. The space complexity is unchanged at $O(n^{\rho+1})$. For completeness, a detailed re-derivation of the results of [20] is included in the full version of the article.

Mixed-MIS algorithm implementation. On Figure 2, the "m-MIS"-curve, corresponds to our mixed-MIS-based algorithm in $O(n^{2\rho}\sqrt{|V|}|E|)$. Compared to the algorithm presented in Algorithm 1, a more efficient rule is used in the non-separable case: we loop over all possible $r \in R$ and add $N(r) \cdot r$ to the schedule (instead of a single vertex $\ell \in L$).

5.2 Random bipartite graphs

Benchmark details. Figure 2 shows, as a function of the number of vertices, the average execution time of both our algorithms (top panel), as well as the distribution of parameter values (ρ - bottom panel), on a class of random bipartite graphs. These graphs are generated according to an Erdös-Rényi distribution (each pair of vertices has a constant probability p of forming an edge). We use a connection probability of d/n, dependent on the number of vertices. It is such that the average degree of vertices is d. The data of our benchmark (Figure 2) has been generated with $d = 5$.

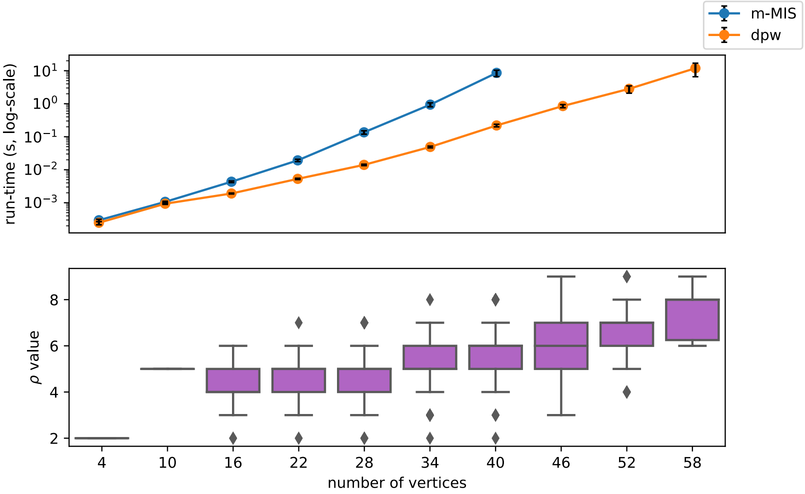

Figure 2 (top panel) Average run-time (seconds, log-scale) of our algorithms on random Erdös-Rényi bipartite graphs, with a probability of connection such that the average degree of a vertex is 5 (i.e $p = 5/n$). (bottom panel) Average parameter value of generated instances, as a function of input size.

Comments on Figure 2. The difference in trend between the execution times of the two algorithms is quite coherent with the difference in their exponents ($n^{\rho+2}$ vs. $n^{2\rho+2.5}$).

5.3 Computing energy barriers in RNA kinetics

In this section, we give more detail about how our algorithms may apply to a bioinformatics problem, the RNA barrier energy problem. We present benchmark results, on a random class of RNA instances, showing the practicality of our approach.

RNA basics. RiboNucleic Acids (RNAs) are biomolecules of outstanding interest for molecular biology, which can be represented as strings over an alphabet $\Sigma := \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{U}\}$ (in this context, n denotes the length of the string). Importantly, these strings may *fold* on themselves to adopt one or several conformation(s). A conformation is typically described by a set of base pairs $(i, j), i < j$. Then, a standard class of conformations to consider in RNA bioinformatics are *secondary structures*, which are pairwise non-crossing ($\nexists(i, j), (k, l) \in S$ such that $i \leq k \leq j \leq l$, in particular, they involve distinct positions). In this section, we more precisely work on the problem of finding a reconfiguration pathway between two *secondary structures* (i.e conflict-free sets of base pairs). The reconfiguration may only involve secondary structures, and remain of energy as low as possible. We work with a simple energy model consisting of the opposite of *number of base pairs* in a configuration ($-N_{bps}$). The RNA Energy-Barrier problem can then be stated as such:

RNA Energy-Barrier
Input: Secondary structures L and R; Energy barrier $k \in \mathbb{N}^+$
Output: True if there exists a sequence $S_0 \cdots S_\ell$ of secondary structures such that

- $S_0 = L$ and $S_\ell = R$;
- $|S_i| \geq |L| - k, \forall i \in [0, \ell]$;
- $|S_i \triangle S_{i+1}| = 1, \forall i \in [0, \ell - 1]$.

False otherwise.

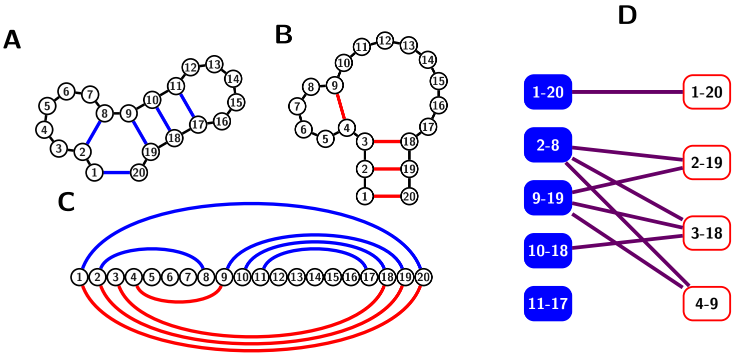

Figure 3 Conflict bipartite graph (D) associated with an instance of the RNA Energy-Barrier problem, consisting of an initial (A) and final (B) structure, both represented as an arc-annotated sequence (C). The sequence of valid secondary structures, achieving minimum energy barrier can be obtained from the solution given in Figure 1.

Bipartite representation. Given two secondary structures L and R, represented as sets of base pairs, we define a *conflict graph* $G(L,R)$ such that: the vertex set of $G(L,R)$ is $L \cup R$; and two vertices $(i,j),(k,l)$ are connected if they are crossing (see Figure 3). Since base pairs in both L and R are both pairwise non-crossing, $G(L,R)$ is bipartite with parts L and R. In this context, a maximum independent set of $G(L,R)$ is a *minimum free-energy structure* of the RNA, and we write $\mathrm{MFE}(L,R) = \alpha(G(L,R))$. We then see how the RNA Energy-Barrier problem is simply Bipartite Independent Set Reconfiguration restricted to a specific class of bipartite graphs: the conflict graphs of secondary structures, with a range of $\rho = k + \mathrm{MFE}(L,R) - |L|$.

Problem motivation. Since the number of secondary structures available to a given RNA grows exponentially with n, RNA energy landscapes are notoriously *rugged*, *i.e.* feature many local minima, and the folding process of an RNA from its *synthesis* to its theoretical final state (a thermodynamic equilibrium around low energy conformations) can be significantly slowed down. Consequently, some RNAs end up being degraded before reaching this final state. This observation motivates the study of RNA kinetics, which encompass all time-dependent aspects of the folding process. In particular, it is known (Arrhenius law) that the energy barrier is the dominant factor influencing the transition rate between two structures, with an exponential dependence.

Related works in bioinformatics. The problem was shown to be NP-hard by Maňuch *et al* [17]. Thachuk *et al* [21] also proposed an XP algorithm in $O(n^{2k+2.5})$ parameterized by the energy barrier k, restricted to instances such that the maximum independent set of $G(L,R)$ has cardinality equal to $|L|$ and $|L| = |R|$.

Benchmark details. Figure 4 shows (top panel) the average execution time of our algorithms on random RNA instances. The bottom panel shows the parameter distribution as a function of the length of the RNA string. Random instances are generated according to the following model: two secondary structures L, R are chosen *uniformly* at random (within the space of all possible secondary structure). Base pairs are constrained to occur between nucleotides separated by a distance of at least $\theta = 5$.

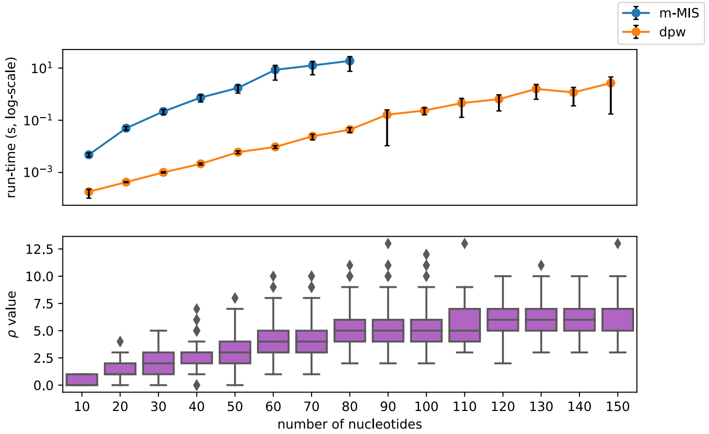

Figure 4 Execution time of our algorithms on random RNA reconfiguration instances (top panel). On the bottom panel, the distribution of the parameter value (ρ) is plotted against the length of the RNA string. Error bars (top panel) are obtained using a bootstrapping method.

5.4 RNA specific optimizations

Dynamic Programming and RNA. Given two secondary structures L and R, a mixed MIS of $G(L, R)$ is a *maximum conflict-free* subset of $L \cup R$, containing at least a base pair from L and R. As is the case for many algorithmic problems involving RNA, the fact that RNAs are *strings* and that base pairs define *intervals* suggests a dynamic programming approach to the mixed maximum independent set problem in RNA conflict graphs. Subproblems will correspond to *intervals* of the RNA string. Let us start with a simple dynamic programming scheme allowing to compute an unconstrained MIS.

Unconstrained MIS DP scheme. A maximum conflict-free subset of $L \cup R$ can be computed by dynamic programming, using the following DP table: for each $1 \leq i \leq j \leq n$, let $MCF_{i,j}$ be the size of a maximum conflict-free subset of all base pairs included in $[i, j]$.

▶ **Lemma 10.** *$MCF_{1,n}$ can be computed in time $O(n^2)$*

Proof. We have the following recurrence formula:

$$MCF_{i,i'} = 0, \forall i' < i$$

$$MCF_{i,j} = \max \begin{cases} MCF_{i+1,j} \\ \max_{(i,k) \in L \cup R} 1 + MCF_{i+1,k-1} + MCF_{k+1,j} \end{cases}$$

Note that the last *max* is over at most two possible pairs (i, k) (1 from L and 1 from R), per the fact that L and R are both conflict-free. ◀

Mixed MIS DP scheme. The following modifications to the DP scheme above allow to compute a mixed MIS of $G(L, R)$ while retaining the same complexity. In addition to the interval, we index the table by Boolean α and β which, when true, further restricts the optimization to subsets with > 0 pair from L (iff $\alpha =$ True) or R (iff $\beta =$ True):

$$MCF_{i,i'}^{\alpha,\beta} = \begin{cases} 0 & \text{if } (\alpha,\beta) = (\mathsf{False},\mathsf{False}) \\ -\infty & \text{otherwise} \end{cases}, \forall i' < i$$

$$MCF_{i,j}^{\alpha,\beta} = \max \begin{cases} MCF_{i+1,j}^{\alpha,\beta} \\ \max\limits_{\substack{(i,k)\in E \\ \alpha',\alpha'',\beta',\beta''\in\mathbb{B}^4}} 1 + MCF_{i+1,k-1}^{\alpha',\beta'} + MCF_{k+1,j}^{\alpha'',\beta''} & \left| \begin{array}{l} \text{if } \neg\alpha \vee \alpha' \vee \alpha'' \vee ((i,k) \in L) \\ \text{and } \neg\beta \vee \beta' \vee \beta'' \vee ((i,k) \in R) \end{array} \right. \end{cases}$$

Through a suitable memorization, the system can be used to compute in $\mathcal{O}(n^2)$ the maximum cardinality $MCF_{1,n}^{\mathsf{True},\mathsf{True}}$ of a subset over the whole sequence. A backtracking procedure is then used to rebuild the maximal subset.

6 Conclusion

Our work so far sheds a new light on both Bipartite Independent Set Reconfiguration and Directed Pathwidth problems. The former can thus be solved with a parameterized algorithm, having important applications in RNA kinetics since the range parameter is particularly relevant in this context. We hope the newly drawn connection will help settle the fixed parameter tractability of computing the directed pathwidth. A slightly more accessible open problem would be to design an FPT algorithm for BISR in the context of secondary structure conflict graphs (i.e. those graphs arising in RNA reconfiguration).

References

1 János Barát. Directed path-width and monotonicity in digraph searching. *Graphs and Combinatorics*, 22(2):161–172, 2006.

2 Hans L Bodlaender. Fixed-parameter tractability of treewidth and pathwidth. In *The Multivariate Algorithmic Revolution and Beyond*, pages 196–227. Springer, 2012.

3 Jianer Chen and Iyad A Kanj. Constrained minimum vertex cover in bipartite graphs: complexity and parameterized algorithms. *Journal of Computer and System Sciences*, 67(4):833–847, 2003.

4 David Coudert, Dorian Mazauric, and Nicolas Nisse. Experimental evaluation of a branch-and-bound algorithm for computing pathwidth and directed pathwidth. *Journal of Experimental Algorithmics (JEA)*, 21:1–23, 2016.

5 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 5. Springer, 2015.

6 Joshua Erde. Directed path-decompositions. *SIAM Journal on Discrete Mathematics*, 34(1):415–430, 2020.

7 Komei Fukuda and Tomomi Matsui. Finding all the perfect matchings in bipartite graphs. *Applied Mathematics Letters*, 7(1):15–18, 1994.

8 Marinus Gottschau, Felix Happach, Marcus Kaiser, and Clara Waldmann. Budget minimization with precedence constraints. *CoRR*, abs/1905.13740, 2019. `arXiv:1905.13740`.

9 Takehiro Ito, Marcin Kamiński, Hirotaka Ono, Akira Suzuki, Ryuhei Uehara, and Katsuhisa Yamanaka. Parameterized complexity of independent set reconfiguration problems. *Discrete Applied Mathematics*, 283:336–345, 2020.

10 Jeff Kinne, Ján Manuch, Akbar Rafiey, and Arash Rafiey. Ordering with precedence constraints and budget minimization. *CoRR*, abs/1507.04885, 2015. `arXiv:1507.04885`.

11 Kenta Kitsunai, Yasuaki Kobayashi, Keita Komuro, Hisao Tamaki, and Toshihiro Tano. Computing directed pathwidth in $o(1.89^n)$ time. *Algorithmica*, 75(1):138–157, 2016.

12 Yasuaki Kobayashi, Keita Komuro, and Hisao Tamaki. Search space reduction through commitments in pathwidth computation: An experimental study. In *International Symposium on Experimental Algorithms*, pages 388–399. Springer, 2014.

13 Daniel Lokshtanov and Amer E Mouawad. The complexity of independent set reconfiguration on bipartite graphs. *ACM Transactions on Algorithms (TALG)*, 15(1):1–19, 2018.

14 Daniel Lokshtanov, Amer E Mouawad, Fahad Panolan, MS Ramanujan, and Saket Saurabh. Reconfiguration on sparse graphs. *Journal of Computer and System Sciences*, 95:122–131, 2018.

15 Daniel Lokshtanov, Amer E Mouawad, Fahad Panolan, and Sebastian Siebertz. On the parameterized complexity of reconfiguration of connected dominating sets. *arXiv preprint arXiv:1910.00581*, 2019.

16 László Lovász and Michael D Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

17 Ján Maňuch, Chris Thachuk, Ladislav Stacho, and Anne Condon. Np-completeness of the energy barrier problem without pseudoknots and temporary arcs. *Natural Computing*, 10(1):391–405, 2011.

18 Amer E Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. *Algorithmica*, 78(1):274–297, 2017.

19 Hisao Tamaki. A directed path-decomposition approach to exactly identifying attractors of boolean networks. In *2010 10th International Symposium on Communications and Information Technologies*, pages 844–849. IEEE, 2010.

20 Hisao Tamaki. A polynomial time algorithm for bounded directed pathwidth. In Petr Kolman and Jan Kratochvíl, editors, *Graph-Theoretic Concepts in Computer Science*, pages 331–342, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

21 Chris Thachuk, Jan Manuch, Arash Rafiey, Leigh-Anne Mathieson, Ladislav Stacho, and Anne Condon. An Algorithm for the Energy Barrier Problem Without Pseudoknots and Temporary Arcs. In *Biocomputing 2010*, pages 108–119. World Scientific, October 2009. `doi:10.1142/9789814295291_0013`.

22 Ignacio Tinoco Jr and Carlos Bustamante. How rna folds. *Journal of molecular biology*, 293(2):271–281, 1999.

23 Jan van den Heuvel. The complexity of change. *Surveys in combinatorics*, 409(2013):127–160, 2013.

24 Boting Yang and Yi Cao. Digraph searching, directed vertex separation and directed pathwidth. *Discrete Applied Mathematics*, 156(10):1822–1837, 2008.

25 Zan-Bo Zhang, Xiaoyan Zhang, and Xuelian Wen. Directed hamilton cycles in digraphs and matching alternating hamilton cycles in bipartite graphs. *SIAM J. Discret. Math.*, 27(1):274–289, 2013. `doi:10.1137/110837188`.