An algorithm for the energy barrier problem without pseudoknots and temporary arcs

Chris Thachuk, Ján Maňuch, Arash Rafiey, Leigh-Anne Mathieson, Ladislav Stacho and Anne Condon

Department of Computer Science, University of British Columbia, Canada
School of Computing Science, Simon Fraser University, Canada
Department of Mathematics, Simon Fraser University, Canada

We make two new contributions to the problem of calculating pseudoknot-free folding pathways with minimum energy barrier between pairs $(\mathcal{A}, \mathcal{B})$ of RNA secondary structures. Our first contribution pertains to a problem posed by Morgan and Higgs: find a min-barrier *direct* folding pathway for a simple energy model in which each base pair contributes -1. In a direct folding pathway, intermediate structures contain only base pairs in $\mathcal{A}$ and $\mathcal{B}$ and are of length $|\mathcal{A} \triangle \mathcal{B}|$ (the size of the symmetric difference of the two structures). We show how to solve this problem exactly, using techniques for deconstructing bipartite graphs. The problem is NP-hard and so our algorithm requires exponential time in the worst case but performs quite well empirically on pairs of structures that are hundreds of nucleotides long. Our second contribution shows that for the simple energy model, repeatedly adding or removing a base pair from $\mathcal{A} \cup \mathcal{B}$ along a pathway is not useful in minimizing the energy barrier. Two consequences of this result are that (i) the problem of determining the min-barrier pseudoknot-free folding pathway from the space of direct pathways with repeats is NP-hard and (ii) our new algorithm finds the min-barrier pathway not only from the space of direct folding pathways but in fact from the space of direct pathways with repeats.

Keywords: RNA secondary structure; RNA folding pathways; energy barrier problem

1. Introduction

We present new algorithms for *exactly* computing direct folding pathways between two RNA structures that have minimum energy barrier, for a simple energy model. We first briefly motivate the energy barrier computation problem, describe previous work on energy barrier calculation and summarize our results.

Motivation. RNA molecules play vital roles in the cell, not only because of the diverse structures they can form but also because of their ability to fold into alternative structures under changing environmental conditions.[1–6] Thus knowledge of folding pathways between pairs of alternative RNA structures is very valuable for inferring RNA function in such environments, and is valuable also for predicting RNA structure, e.g., in light of co-transcriptional folding.[2,7–10]

Computational approaches for predicting folding pathways focus on secondary structure— the set of base pairs that form when an RNA molecule folds. As illustrated in Fig. 1, the folding pathway from an initial to a final structure is a sequence of intermediate structures, each differing from the previous one by a single base pair (or equivalently by a single arc in the arc diagram representation). Much focus to date has been on pathways of pseudoknot-free secondary structures— structures in which no base pairs cross. Since folding is a thermodynamically-driven probabilistic process, folding pathways tend to avoid high-energy structures. As a result, many methods for predicting folding pathways or energy landscapes— particularly course-grained methods designed to work for long structures which do not attempt to model the complete energy landscape— are guided by calculations of the *energy barrier*.[2,11] Energy barrier calculations have also been useful in constructing barrier trees and to study properties of disordered systems in statistical physics.[12–14]

If $\pi = \mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_m$ is a folding pathway from structure $\mathcal{A}$ to structure $\mathcal{B}$ (i.e., $\mathcal{A} = \mathcal{P}_0$ and $\mathcal{B} = \mathcal{P}_m$), the *energy barrier* of π is $\max_{1 \leq i \leq m} E(\mathcal{P}_i) - E(\mathcal{P}_0)$, where $E(\mathcal{S})$ denotes the free energy change of secondary structure S. In this paper we focus on a simple energy model in which each base pair contributes -1 to the total energy. The energy barrier of pair $(\mathcal{A}, \mathcal{B})$ is the lowest energy barrier, taken over all pathways from $\mathcal{A}$ to $\mathcal{B}$. Note that there is always a folding pathway from $\mathcal{A}$ to $\mathcal{B}$ in which first all arcs (base pairs) of $\mathcal{A}$ are removed and then all arcs of $\mathcal{B}$ are added. The question is whether there is another pathway that avoids such a high energy barrier, by adding arcs of $\mathcal{B}$ before all arcs of $\mathcal{A}$ are removed. Throughout, we consider only pseudoknot-free pathways, in which all intermediate structures are pseudoknot-free.

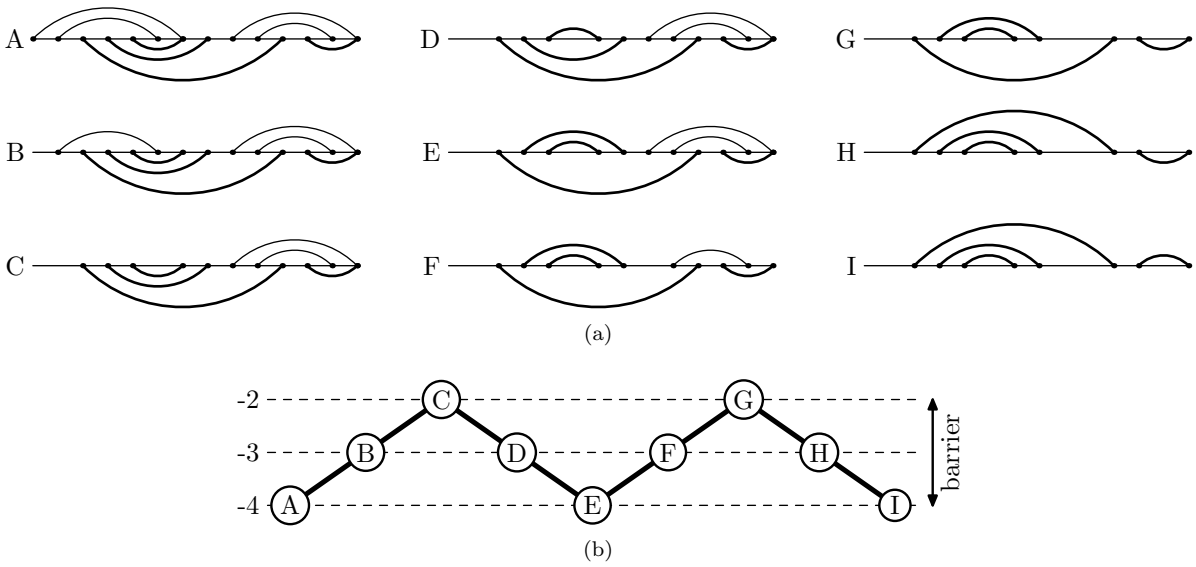

Fig. 1: (a) A possible pseudoknot-free folding pathway is shown for an initial structure A transitioning through intermediate structures $(B, C, \ldots)$ until the final structure I is reached. Structures are represented as arc diagrams. For a particular position in the pathway, the top of the arc diagram denotes the base pairs of the structure and the bottom of the diagram denoes the base pairs that are currently not in the structure but need to be in the final structure. Each structure along the pathway differs from its neighbours by at most one arc. Furthermore, each arc is either removed (if in A) or added (if in I) at most once along the pathway and thus the pathway is *direct*. (b) The corresponding energy plot. The barrier in this example is two.

Background and previous work. There is a rich literature on the problem of predicting folding pathways and energy landscapes; see the recent work of Geis at al.,[2] Tang et al.[11] and the references therein. We focus here on algorithms for energy barrier calculation which are an important component of many approaches to estimation of folding pathways and energy landscapes. Such methods have been proposed, for example, by Morgan and Higgs,[13] Wolfinger et al.,[12,14] Flamm et al.[15] and Geis et al.[2]

Several versions of the energy barrier problem have been studied, which are distinguished by properties of the intermediate structures. Morgan and Higgs focus on *direct* folding pathways from structure $\mathcal{A}$ to structure $\mathcal{B}$ in which intermediate structures contain only arcs in $\mathcal{A} \cup \mathcal{B}$ and such that the total pathway length is $|\mathcal{A} \triangle \mathcal{B}|$. In such pathways, each arc from the initial structure not also in the final structure is removed exactly once and each arc from the final structure not also in the initial structure is added exactly once along the pathway. A larger class of pathways is obtained by allowing the the length of the pathway to exceed $|\mathcal{A} \triangle \mathcal{B}|$. We call such pathways *direct-with-repeats* pathways since an arc from $\mathcal{A}$ or $\mathcal{B}$ may be added or removed multiple times along the pathway. An even more general class of pathways allow intermediate structures to contain "temporary" base pairs which are neither in $\mathcal{A}$ nor in $\mathcal{B}$. Morgan and Higgs call such pathways *indirect*. Thus, direct pathways are a subclass of direct-with-repeats pathways, which in turn are a subclass of indirect pathways.

Morgan and Higgs assume the simple energy model in which each base pair contributes -1 to the total free energy. Using a randomized greedy approach, they construct several low-barrier direct pathways and take the minimum energy barrier of these as their estimate. (They also construct indirect pathways using a "single link clustering" method.) Wolfinger et al. use a barrier tree to represent the folding landscape; identifying nodes in the tree (which are called saddle points) is analogous to calculating energy barriers. Flamm et al.'s method[15] for approximating energy barriers explores direct pathways by performing a breadth-first search, maintaining the best m candidate solutions at each step. As m becomes large, the search does become

exhaustive, yielding an exact solution, however exponential runtime and memory are required. The program `barriers`[12] is capable of computing exact direct and indirect pathways, provided a complete sample of low energy states separating the two structures is provided. However, this approach is also exponential in runtime and space and thus unsuitable for medium or large problem instances.

For their Kinwalker folding pathway predictor, Geis et al.[2] describe a heuristic which explores the space of possible direct pathways in a more sophisticated manner than does the Morgan-Higgs heuristic, incorporating a parameter lookahead technique to avoid excessive runtimes. While their method uses the Turner energy model to evaluate energy barriers, it relies on simple addition and removal of base pairs (and thus the simple energy model) while generating putative low-barrier pathways.

In summary, all current methods are either heuristic in nature and thus are not guaranteed to find the exact energy barrier between two structures, or are exponential in both runtime and space, precluding their use on even medium sized problem instances. Thus there is strong motivation for finding a *fast* method which can *exactly* compute the energy barrier between two structures. Indeed, the Geis et al. method can estimate energy barriers for structures of long sequences (1,500nt or more) but the authors note that "as the performance of Kinwalker crucially depends on approximating saddle heights, further improvements to the Morgan-Higgs heuristic as well as alternative approaches will be investigated".

Finally, in earlier work, we showed that the problem of finding the energy barrier of direct pathways between two structures is NP-hard.[16] Thus we do not expect to find an algorithm for energy barrier calculation whose running time is bounded by polynomial in its input, in the worst case.

Our results. Our main contributions are new approaches for exactly calculating the energy barrier between two RNA secondary structures along direct, pseudoknot-free pathways. To develop a sound theoretical basis for energy barrier calculation we follow the approach of Morgan and Higgs: we assume a simple energy model in which each base pair contributes -1 to the total energy and we focus on the problem of finding min-barrier pathways between structures with minimum free energy. Structure $\mathcal{S}$ is a *minimum free energy* (MFE) structure for a given sequence if no other structure for the sequence has lower energy than $\mathcal{S}$. A sequence may have more than one MFE structure.

Our methods exploit elegant algorithms for bipartite graphs to split a problem into independent subproblems where possible. The theoretical run-time complexity of our algorithms is exponential in the worst case; this is not surprising since the problem is NP-hard.[16] However, our empirical analysis shows that implementations of our algorithms can often find the energy barrier on problem instances with hundreds of nucleotides in seconds. Our algorithms are highly amenable to parallelization and have potential to work with more sophisticated energy models that include Turner parameters for base stacking, for example. Furthermore, our methods could be integrated with current heuristics in order to improve energy barrier estimation on structures with thousands of nucleotides.

Our second contribution addresses the following question: for the simple Morgan-Higgs energy model could direct-with-repeats pathways have lower energy barriers than direct pathways? We show that the answer to this question is no: repeated arcs cannot help to lower the energy barrier of a pathway. Two consequences of this contribution are that (i) the problem of determining the min-barrier pseudoknot-free folding pathway from the space of direct-with-repeats pathways is NP-hard and (ii) our algorithm finds the min-barrier pathway not only from the space of direct folding pathways but in fact from the space of direct-with-repeats folding pathways.

In Section 2 we present our result that repeated arcs do not help to decrease the energy barrier in direct pathways. In Section 3 we present our new algorithmic techniques for exactly computing min-barrier pseudoknot-free folding pathways. We present our empirical analysis in Section 4. We provide a brief discussion and suggest directions for future work in Section 5.

2. Repeated base pairs do not lower the energy barrier

In this section we show that on a folding pathway from structure $\mathcal{A}$ to structure $\mathcal{B}$, repeatedly adding or removing arcs from $\mathcal{A}$ or $\mathcal{B}$ along the pathway cannot help to reduce the energy barrier. This enables us to consider only direct (i.e., repeat-free) pathways in our later algorithms without loss of generality.

It is convenient in this section to slightly generalize the definition of a folding pathway, so that a structure on the pathway differs from its predecessor by at most one (rather than exactly one) arc. A sequence $T = T[1], T[2], \ldots, T[m]$ is a *transformation sequence* for a direct-with-repeats pathway $\pi = \mathcal{P}_0, \ldots, \mathcal{P}_m$ if for all $i \in [1, m]$, one of the following holds:

- $T[i] = +a$, $\mathcal{P}_i = \mathcal{P}_{i-1} \cup \{a\}$, and $a \notin \mathcal{P}_{i-1}$;
- $T[i] = -a$, $\mathcal{P}_i = \mathcal{P}_{i-1} \setminus \{a\}$, and $a \in \mathcal{P}_{i-1}$;
- $T[i] =$ no-op and $\mathcal{P}_i = \mathcal{P}_{i-1}$;

where $a \in \mathcal{P}_0 \cup \mathcal{P}_m$. We call each $T[i]$ an *operation*. Let $\hat{T}[i] = a$ if $T[i] \in \{+a, -a\}$.

Theorem 2.1. *For any direct-with-repeats pseudoknot-free pathway π from structure $\mathcal{A}$ to structure $\mathcal{B}$ there is a direct pseudoknot-free pathway from $\mathcal{A}$ to $\mathcal{B}$ with energy barrier at most that of π.*

Proof. Consider a direct-with-repeats pathway $\pi = \mathcal{P}_0, \ldots, \mathcal{P}_m$ from $\mathcal{A} = \mathcal{P}_0$ to $\mathcal{B} = \mathcal{P}_m$ and let $T[1], \ldots, T[m]$ be a transformation sequence for π. For each $a \in \mathcal{A}$, we call each occurrence of $+a$ or $-a$ in T, except for the *last* $-a$, a *repeat operation*. Similarly, for each $b \in \mathcal{B}$, each occurrence of $+b$ or $-b$ in T except for the last $+b$ is a repeat operation.

If π is direct we are done, so suppose that π is not direct. We will show how to construct a pathway π' and corresponding transformation sequence which has fewer repeats than does T, with the barrier of π' being at most that of π. The proof follows since we can iterate this construction until we obtain a direct transformation sequence from $\mathcal{A}$ to $\mathcal{B}$.

Let $I^+_{\mathcal{A}}$ and $I^-_{\mathcal{A}}$ be the subset of indices $i \in [1, m]$ for which $T[i]$ is a repeat operation of the form $+a$ and $-a$, respectively, where $a \in \mathcal{A}$. Similarly, let $I^+_{\mathcal{B}}$ and $I^-_{\mathcal{B}}$ be the subset of indices $i \in [1, m]$ for which $T[i]$ is a repeat operation of the form $+b$ and $-b$, respectively, where $b \in \mathcal{B}$. To construct a new pathway π' with a smaller number of repeats we consider two cases:
Case 1: $|I^-_{\mathcal{A}}| \geq |I^+_{\mathcal{B}}|$.
Case 2: $|I^-_{\mathcal{A}}| < |I^+_{\mathcal{B}}|$.

We will construct π' only in Case 1. The construction in Case 2 follows by symmetry. Indeed, by reversing the pathway π and its transformation sequence T and replacing each $+$ by a $-$ and vice versa in T, we obtain a pathway $\bar{\pi}$ from $\bar{\mathcal{A}} = \mathcal{B}$ to $\bar{\mathcal{B}} = \mathcal{A}$ and a corresponding transformation sequence $\bar{T}$. Now we have that $|I^-_{\bar{\mathcal{A}}}| = |I^+_{\mathcal{B}}| > |I^-_{\mathcal{A}}| = |I^+_{\bar{\mathcal{B}}}|$. Therefore, the construction in Case 1 produces a transformation sequence from $\mathcal{B}$ to $\mathcal{A}$ with a smaller number of repeat operations. Its reversal, with $+$'s changed to $-$'s and vice versa, is a transformation sequence from $\mathcal{A}$ to $\mathcal{B}$ which has a smaller number of repeat operations than does T.

We use the following notation. For each $i \in I^-_{\mathcal{A}}$ (i.e., $T[i] = -a$ for some $a \in \mathcal{A}$ and is not the last such $-a$), find the smallest index $i' > i$ for which $T[i'] = +a$. We say that i is a partner of i' and vice versa, and write $\text{partner}(i) = i'$ and $\text{partner}(i') = i$. Similarly, every $j \in I^+_{\mathcal{B}}$ is partnered with the smallest $j' > j$ for which $T[j'] = -b$. For any $J \subseteq [1, m]$, let $\text{partner}(J) = \{\text{partner}(j); j \in J\}$.

Consider Case 1. We will identify subsets $I \subseteq I^-_{\mathcal{A}}$ and $J \subseteq I^+_{\mathcal{B}}$ such that the transformation sequence $T(I, J)$ obtained by replacing all operations in T at positions $I \cup \text{partner}(I) \cup J \cup \text{partner}(J)$ with no-op's has a smaller number of repeat operations and corresponds to a valid pathway π' from $\mathcal{A}$ to $\mathcal{B}$. By valid, we mean that π' is pseudoknot-free and has barrier no greater than that of π. The sets $I^-_{\mathcal{A}}$ and $I^+_{\mathcal{B}}$ satisfy a useful property. To describe this, we say that i *conflicts with* j if $T[i] = -a$, $T[j] = +b$, arcs a and b cross and $j \in [i, i']$ where $i' = \text{partner}(i)$. For any $I \subseteq I^-_{\mathcal{A}}$, let $\text{conflict}(I)$ be the subset of $[1, m]$ that conflicts with indices in I. Note that if i conflicts with j, it must be that $[j, \text{partner}(j)] \subset [i, \text{partner}(i)]$. The useful property we have is that $\text{conflict}(I^-_{\mathcal{A}}) \subseteq I^+_{\mathcal{B}}$.

We initially set $I = I^-_{\mathcal{A}}$ and $J = I^+_{\mathcal{B}}$. We will cull sets I and J until $T(I, J)$ is a valid transformation sequence. After each culling step sets I and J will satisfy the following two properties: (a) $|I| > |J|$ and (b) conflict$(I) \subseteq J$. Before each culling step (including the first one), we have $|I| \geq |J|$ and conflict$(I) \subseteq J$.

A culling step is as follows. For any $r \in [1, m]$ let I_r be the subset of $I \cap [1, r]$ whose partners are after position r and let J_r be the subset of $J \cap [1, r]$ whose partners are after position r. Suppose that for some r,

$$|I_r| < |J_r|. \tag{1}$$

Let $I' = I - I_r$ and let $J' = J - J_r$. The sets I' and J' are results of the culling step. If there is no r such that $|I_r| < |J_r|$, no culling step is performed.

We will show that after the culling step, I' and J' satisfy properties (a) and (b). We immediately have that $|I'| > |J'|$ because of the fact that the removed subsets satisfy inequality (1) and that $|I| \geq |J|$. For property (b), we have that conflict$(I') \subseteq$ conflict$(I) \subseteq J$. Hence, to prove that conflict$(I') \subseteq J'$ it is enough to show that conflict$(I') \cap J_r$ is empty. Assume to the contrary that some $i \in I'$ conflicts with $j \in J_r$. Since $j \in J_r$, $r \in [j, j']$, where $j' =$ partner(j). Since i and j conflict, $[j, j'] \subset [i, i']$, where $i' =$ partner(i). Hence, $r \in [i, i']$, a contradiction with the assumption that $i \in I' = I - I_r$.

Now set $I = I'$ and $J = J'$. Repeat the culling step for the resulting (new) sets I and J, until there no longer is any r with $|I_r| < |J_r|$. Note that there can only be a finite number of culling steps, since each removes a non-empty subset from J (since, $|J_r| > |I_r|$, J_r, the set which is removed from J, is non-empty). Thus, at the end of the culling process, we have sets I and J with the following properties:

(1) $I \cup J$ is non-empty,
(2) conflict$(I) \subseteq J$ and
(3) for all $i \in [1, m]$, $|I_i| \geq |J_i|$.

Since $I \cup J$ is non-empty, $T(I, J)$ has fewer repeats than does T. Using property (2) we can show that the pathway π' corresponding $T(I, J)$ is pseudoknot free. Finally, property (3) implies that the energy barrier of π' is at most that of π (details will appear in the full paper). □

We have shown that the direct pseudoknot-free energy barrier problem is NP-complete.[16] The above result implies the following corollary.

Corollary 2.1. *The direct-with-repeats pseudoknot-free energy barrier problem is NP-complete.*

Example.

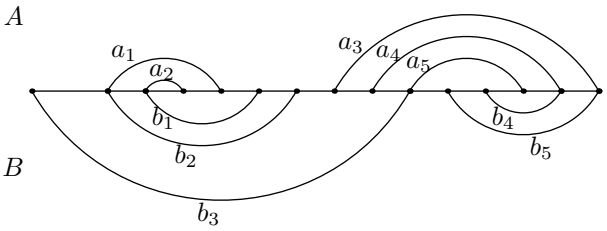

Fig. 2: Initial structure $\mathcal{A}$ and final structure $\mathcal{B}$

Consider the pair of structures $(\mathcal{A}, \mathcal{B})$ in Fig. 2 and the following transformation sequence for a pathway from $\mathcal{A}$ to $\mathcal{B}$:

$-a_1$	$-a_2$	$+b_1$	$+b_2$	$-b_1$	$+a_2$	$-b_2$	$-a_3$	$-a_4$	$+b_2$	$-a_2$	$+b_1$
$T[1]$	$T[2]$	$T[3]$	$T[4]$	$T[5]$	$T[6]$	$T[7]$	$T[8]$	$T[9]$	$T[10]$	$T[11]$	$T[12]$
$-a_5$	$+b_3$	$-b_3$	$+a_3$	$+a_4$	$+a_5$	$-a_3$	$-a_4$	$-a_5$	$+b_3$	$+b_4$	$+b_5$
$T[13]$	$T[14]$	$T[15]$	$T[16]$	$T[17]$	$T[18]$	$T[19]$	$T[20]$	$T[21]$	$T[22]$	$T[23]$	$T[24]$

We begin by identifying the sets $I^-_{\mathcal{A}} = \{2, 8, 9, 13\}$ and $I^+_{\mathcal{B}} = \{3, 4, 14\}$. Now, we find partners for the indexes in these sets. First, $I^-_{\mathcal{A}}$: partner$(2) = 6$, partner$(8) = 16$, partner$(9) = 17$, and partner$(13) = 18$. Now, $I^+_{\mathcal{B}}$: partner$(3) = 5$, partner$(4) = 7$ and partner$(14) = 15$.

Initially, we set $I = I^-_{\mathcal{A}} = \{2, 8, 9, 13\}$ and $J = J^+_{\mathcal{B}} = \{3, 4, 14\}$. Note that the conditions $|I| = 4 > 3 = |J|$ and conflict$(I) = \{14\} \subseteq J$ are true before the first culling step. Now, there is an r such that $|I_r| < |J_r|$: in particular, if $r = 7$, $I_r = \{2\}$ and $J_r = \{3, 4\}$. Hence, the new I is set to $I - I_r = \{8, 9, 13\}$ and the new J to $J - J_r = \{14\}$. We can now set $I = I'$ and $J = J'$. After this step we can see that I is non-empty, conflict$(I) = \{14\} \subseteq J$ and for all $i \in [1, m], |I_i| \geq |J_i|$ since the only element in J occurs after all the elements in I. We now replace operations at positions $I \cup$ partner$(I) \cup J \cup$ partner(J) with no-ops. We will denote no-ops by ϵ. The transformation sequence is now:

$-a_1$	$-a_2$	$+b_1$	$+b_2$	$-b_1$	$+a_2$	$-b_2$	ϵ	ϵ	$+b_2$	$-a_2$	$+b_1$
$T[1]$	$T[2]$	$T[3]$	$T[4]$	$T[5]$	$T[6]$	$T[7]$	$T[8]$	$T[9]$	$T[10]$	$T[11]$	$T[12]$
ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	$-a_3$	$-a_4$	$-a_5$	$+b_3$	$+b_4$	$+b_5$
$T[13]$	$T[14]$	$T[15]$	$T[16]$	$T[17]$	$T[18]$	$T[19]$	$T[20]$	$T[21]$	$T[22]$	$T[23]$	$T[24]$

Once again, we identify the sets $I^-_{\mathcal{A}} = \{2\}$ and $I^+_{\mathcal{B}} = \{3, 4\}$. Since $|I^-_{\mathcal{A}}| < |I^+_{\mathcal{B}}|$ we are now in Case 2. We proceed by reversing the transformation sequence and replacing each $+$ with a $-$, and vice versa to obtain a transformation sequence $\bar{T}$ from $\bar{\mathcal{A}} = \mathcal{B}$ to $\bar{\mathcal{B}} = \mathcal{A}$. This results in the following transformation sequence:

$-b_5$	$-b_4$	$-b_3$	$+a_5$	$+a_4$	$+a_3$	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
$T[1]$	$T[2]$	$T[3]$	$T[4]$	$T[5]$	$T[6]$	$T[7]$	$T[8]$	$T[9]$	$T[10]$	$T[11]$	$T[12]$
$-b_1$	$+a_2$	$-b_2$	ϵ	ϵ	$+b_2$	$-a_2$	$+b_1$	$-b_2$	$-b_1$	$+a_2$	$+a_1$
$T[13]$	$T[14]$	$T[15]$	$T[16]$	$T[17]$	$T[18]$	$T[19]$	$T[20]$	$T[21]$	$T[22]$	$T[23]$	$T[24]$

Now our new $I^-_{\bar{\mathcal{A}}} = \{13, 15\}$ and $I^+_{\bar{\mathcal{B}}} = \{14\}$, so we are back in case 1. Next, we identify the partners of these repeats: partner$(13) = 20$, partner$(15) = 18$, and partner$(14) = 19$. As before, we initially set $I = I^-_{\bar{\mathcal{A}}} = \{13, 15\}$ and $J = I^+_{\bar{\mathcal{B}}} = \{14\}$. We know that there is no r such that $|I_r| < |J_r|$ because J's only element occurs after an element in I, so $|I_r|$ and $|J_r|$ need to be at least equal for any r. So, sets I and J do have all three expected properties: I is non-empty, conflict$(I) = \{14\} \subseteq J$ and for all $i \in [1, m], |I_i| \geq |J_i|$. We now replace operations at positions $I \cup$ partner$(I) \cup J \cup$ partner(J) with no-ops, obtaining the new transformation sequence (here, for brevity, we collapse $T[7]$ through $T[20]$ since they are all no-ops:

$-b_5$	$-b_4$	$-b_3$	$+a_5$	$+a_4$	$+a_3$	ϵ	$-b_2$	$-b_1$	$+a_2$	$+a_1$
$T[1]$	$T[2]$	$T[3]$	$T[4]$	$T[5]$	$T[6]$	$T[7-20]$	$T[21]$	$T[22]$	$T[23]$	$T[24]$

Since we are interested in a pathway from $\mathcal{A}$ to $\mathcal{B}$, we need to reverse the transformation sequence and replace every $+$ with a $-$.The transformation sequence is now repeat-free:

$-a_1$	$-a_2$	$+b_1$	$+b_2$	ϵ	$-a_3$	$-a_4$	$-a_5$	$+b_3$	$+b_4$	$+b_5$
$T[1]$	$T[2]$	$T[3]$	$T[4]$	$T[5-18]$	$T[19]$	$T[20]$	$T[21]$	$T[22]$	$T[23]$	$T[24]$

3. Algorithms for exactly computing energy barriers

In this section we describe our algorithms for finding the min-barrier pathway between two structures. We model the problem in terms of bipartite graphs.

For a pair of pseudoknot-free structures for the same RNA sequence, we define the *conflict graph* to be a bipartite graph $G = (A, B; E)$ where A is the set of arcs from the first structure, B is the set of arcs from the second structure and there is an edge in E between arc $a \in A$ and arc $b \in B$ if and only if a and b are crossing. Throughout, we denote the neighbours (in A) of a subset B' of B by $N(B')$. Also, we denote the subgraph of G induced by subsets $A' \subseteq A$ and $B' \subseteq B$ by $G/(A', B')$. We need a notion analogous to that of a pair of MFE structures in the context of bipartite graphs. We say that G is *pairwise-optimal* if the size

of the maximum independent set in the graph is $|A|(=|B|)$. If A and B are MFE structures then G must be pairwise-optimal; otherwise the largest independent in the conflict graph G would be a set of arcs with lower free energy than either A or B. We let $[i,j]$ represent the interval (set of integers) $\{i, i+1, \ldots, j\}$.

Let $G = (A, B; E)$ be a pairwise-optimal bipartite graph. A *set pathway* for G is a sequence of independent sets $S_0, \ldots, S_m$, each of which is a subset of $A \cup B$, such that (i) $S_0 = A$, (ii) $S_m = B$ and (iii) for every $i = 1, \ldots, m$, $|S_{i-1} \triangle S_i| = 1$ (the size of symmetric difference is one, i.e., at each step one arc is either added or removed). The *transformation sequence* corresponding to this set pathway is the sequence of singletons $S_0 \triangle S_1, \ldots, S_{m-1} \triangle S_m$. The set pathway is *direct* if its corresponding transformation sequence has no repeating elements. The *barrier* of the pathway (or its corresponding transformation sequence) is $k = \max_i |A| - |S_i|$. (Since A is a maximum independent set of $(A, B; E)$, it must be that $|A| - |S_i| \geq 0$ for all $i, 1 \leq i \leq m$.) We say that a set pathway is a $(\leq k)$-barrier set pathway or a k-barrier pathway if its barrier is $\leq k$ or $= k$, respectively. A *min-barrier* set pathway is a set pathway whose barrier is less than or equal to the barriers of any other set pathway for G. Consider the following problem:

Direct Set Barrier Problem. Given a pairwise-optimal bipartite graph $G = (A, B; E)$ and an integer k, find a direct set pathway with barrier at most k for G if one exists.

An instance of the Direct Energy Barrier Problem can be mapped to an instance of the Direct Set Barrier Problem by contructing its conflict graph. However, the Direct Set Barrier Problem is actually a more general problem since not every bipartite graph is realizable by a pair of pseudoknot-free structures.

Algorithm Overview. Our algorithm for the Direct Set Barrier Problem uses two key ideas. The first is a *splitting strategy*: if for some proper non-empty subset B_1 of B the induced subgraph $G/(A_1, B_1)$ is pairwise-optimal where $A_1 = N(B_1)$ then we can determine the solution for $(A, B; E)$ by recursively solving the problem on the induced subgraphs $G/(A_1, B_1)$ and $G/(A \setminus A_1, B \setminus B_1)$ and combining the solutions to the subproblems. Specifically, if $G/(A_1, B_1)$ is pairwise-optimal then we can show three properties described in the following lemma (due to space limitations we omit details of the proof of correctness of our algorithm here and elsewhere but these will be provided in the full paper):

Lemma 3.1. *Let $G = (A, B; E)$ be a pairwise-optimal bipartite graph and let $G_1 = G/(A_1, B_1)$ be pairwise-optimal where B_1 is a proper non-empty subset of B and $A_1 = N(B_1)$. Let $G' = G/(A \setminus A_1, B \setminus B_1)$. Then*

(1) G' is pairwise-optimal,
(2) if T_1 and T' are $(\leq k)$-barrier transformation sequences for G_1 and G' respectively then T_1, T' is a $(\leq k)$-barrier transformation sequence for G and
(3) G has a $(\leq k)$-barrier set pathway only if both G_1 and G' do.

For efficiency reasons, our implementation generalizes this splitting idea by splitting the problem into as many "minimal" pairwise-optimal subproblems as possible. We say that $(A, B; E)$ is *minimal pairwise-optimal* if $|A| = |B|$ and the only independent sets in $(A, B; E)$ of size $|A|$ are A and B.

The second key idea is a *cutting strategy* for reducing the size of minimal pairwise-optimal problem instances. We have developed two cutting strategies. The first, the *two-sided cutting strategy*, generates the subgraphs $G/(A \setminus \{a\}, B \setminus \{b\})$ for each choice of $a \in A$ and $b \in B$ and recursively solves each of the resulting subproblems with the barrier set to $k - 1$. The following lemma states that if we do this for all possible choices of a and b, we are guaranteed to find a $(\leq k)$-barrier set pathway for G if one exists.

Lemma 3.2. *Let $G = (A, B; E)$ be minimal pairwise-optimal. Then*

(1) $G/(A \setminus \{a\}, B \setminus \{b\})$ is pairwise-optimal for all $a \in A$ and $b \in B$,
(2) if $G/(A \setminus \{a\}, B \setminus \{b\})$ has a transformation sequence T' with barrier at most $k-1$ then $T = \{a\}, T', \{b\}$ is a transformation sequence for $(A, B; E)$ with barrier at most k and
(3) G has a transformation sequence with barrier at most k only if $G/(A \setminus \{a\}, B \setminus \{b\})$ has a transformation sequence with barrier at most $k - 1$ for some $a \in A$ and $b \in B$.

The second cutting strategy is a *one-sided cutting strategy*. Suppose that in some ($\leq k$)-barrier pathway for G, b is the first node of B removed in the corresponding transformation sequence T. Let A' be the set of nodes of A which conflict with b (i.e., $N(b') = A'$) and let B' be the set of nodes of B such that $N(B') \subseteq A'$. Then we can assume without loss of generality that all nodes of A' are first removed in T and immediately thereafter all nodes of B' are added. Abusing notation slightly, we can write that $T = -A', +B', \ldots$ where we mean that the singleton subsets of A' are listed first in an unspecified order, followed by the singleton subsets of B' in an unspecified order. Given A' and B', we'd like to use the algorithm recursively on the remaining subgraph $G/(A - A', B - B')$. However this subgraph is not pairwise-optimal and in fact $|A - A'| < |B - B'|$ (otherwise we would be able to apply the splitting strategy). To circumvent this problem we create a new bipartite graph $G'(b)$ by adding $k' = |A'| - |B'|$ "artificial" nodes to G and connecting them to all nodes in $B - B'$. To obtain a ($\leq k$)-barrier pathway for G, we then recursively obtain the transformation sequence T' of a ($\leq k$)-barrier pathway for the new graph $G'(b)$, remove the singletons of T' that involve the k' extra nodes and finally append what's left of T' to A', B'. The following lemma states that if we do this for all possible choices for b, we are guaranteed to find a ($\leq k$)-barrier set pathway for G if one exists.

Lemma 3.3. *Let $G = (A, B; E)$ be minimal pairwise-optimal. Then*

(1) $G'(b)$ is pairwise-optimal for all $b \in B$,
(2) if T' is a ($\leq k$)-barrier transformation sequence for $G'(b)$ for some b then the transformation sequence T obtained from T' via the cutting strategy is a ($\leq k$)-barrier transformation sequence for G and
(3) G has ($\leq k$)-barrier transformation sequence only if for some $b \in B$, $G'(b)$ has a ($\leq k$)-barrier transformation sequence.

The **Direct-SetBarrier** algorithm, presented as Algorithm 3.1 below, incorporates the splitting and two-sided cutting strategy. First, the input graph $(A, B; E)$ is split to yield one or more subproblems (line 3), via a call to procedure Split$(A, B; E)$. We describe later how this procedure works. The subproblems produced by this split procedure cannot be split further, so our cutting strategy reduces these to smaller subproblems (lines 12-19), unless they are already trivial to solve (lines 9-10), and concatenates the solutions to the subproblems. By changing the inner loop, the one-sided cutting strategy can be implemented.

In the rest of this section we provide more details of the Split procedure. We conclude with a note on the theoretical run-time and space complexity of the algorithm. In Section 4 we provide an empirical analysis of the performance of our algorithm.

Details of the Split procedure. Given as input a pairwise-optimal bipartite graph $G = (A, B; E)$, our Split procedure produces a sequence of non-empty bipartite graphs $G_1, G_2, \ldots, G_p$ via the following steps.

First find a maximum matching $\mathcal{M}$ in $(A, B; E)$ using the Hopcroft-Karp algorithm.[17] (As we note in our correctness proof, well-known results on bipartite graphs show that such a matching exists.) Second, create the *precedence graph* for G and $\mathcal{M}$. By precedence graph, we mean the directed bipartite graph $D = (A, B; E')$ where $E' = \{(b, a) | b \in B \wedge (b, a) \in E\} \cup \{(a, b) | a \in A \wedge (a, b) \in \mathcal{M}\}$. Third, using Tarjan's algorithm,[18] find the strongly connected components in the precedence graph. Finally, create a total order of these components in a manner that is consistent with their topological ordering in the associated *condensation graph*, i.e. the directed acyclic graph in which each strongly connected component is condensed into a single node. Let (A_i, B_i) be the set of nodes in the ith component in this total ordering. Then the graph G_i is chosen to be the subgraph of $G/(A - \cup_{j=1}^{i-1} A_j, B - \cup_{j=1}^{i-1} B_j)$ induced by the nodes (A_i, B_i). The next lemma summarizes important properties of the Split procedure.

Lemma 3.4. *Given as input a pairwise-optimal bipartite graph $G = (A, B; E)$ the* Split *procedure produces a sequence $G_1 = (A_1, B_1; E_1), G_2 = (A_2, B_2; E_2), \ldots, G_p = (A_p, B_p; E_p)$ such that*

(1) G_i is minimal pairwise-optimal for each $i, 1 \leq i \leq p$ and
(2) $A_i = N(B_i)$ in the graph $G'_i = G/(A - \cup_{j=1}^{i-1} A_j, B - \cup_{j=1}^{i-1} B_j)$ for all i, $1 \leq i \leq p$.

Algorithm 3.1 Direct Set Barrier Algorithm **Direct-SetBarrier**$((A, B; E), k)$

```
// INPUT: a non-empty pairwise-optimal bipartite graph (A, B; E) and a barrier k ≥ 0
// OUTPUT: a direct transformation sequence from A to B with barrier at most k, or "no solution"
call procedure SPLIT(A, B; E) to obtain subproblems G_1 = (A_1, B_1; E_1), ..., G_p = (A_p, B_p; E_p)
T ← ∅ // empty sequence
if k ≤ 0 then
  return "no solution"
else
  for i = 1 to p do
    if |A_i| ≤ k then
      append A_i, B_i to T
    else
      // the inner loop:
      for all a ∈ A_i and b ∈ B_i do
        call DIRECT-SETBARRIER(G_i/(A_i \ {a}, B_i \ {b}), k − 1)
        if a sequence T' was returned then
          append {a}, T', {b} to T
          exit the inner loop
        end if
      end for
      if no transformation sequence was found in the inner loop then
        return "no solution"
      end if
    end if
  end for
  return T
end if
```

Run-time and space complexity. The run-time of the SPLIT$(A, B; E)$ procedure is dominated by the time needed to find a maximum matching in a bipartite graph. This time is $O(n^{2.5})$, or more precisely $O(n^{1/2}m)$ using the Hopcroft-Karp algorithm[17] where $n = |A| = |B|$ and $m = |E|$. The remaining steps of the SPLIT$(A, B; E)$ procedure take $O(n + m)$ time.

The DIRECT-SETBARRIER$((A, B; E), k)$ algorithm with the two-sided cutting strategy has theoretical run-time complexity $O(n^{2k+2.5})$. Recall that k is the input which specifies the allowable barrier, i.e. the algorithm determines whether $(A, B; E)$ has a set with barrier at most k. The worst case arises when every call to the SPLIT procedure produces just one subproblem, because in this case we are not able to split a problem (or its recursive subproblems) into smaller independent problems. In this case the inner loop is called $O(n^2)$ times on a subproblem with allowable barrier $k - 1$ and the recursion bottoms out when k reaches 0. We therefore have $O(n^{2k})$ calls to the DIRECT-SETBARRIER$((A, B; E), k)$ algorithm (Algorithm 3.1), each taking time $O(n^{2.5})$ for a total running time of $O(n^{2k+2.5})$. Note that the run-time is exponential in k. However, since the problem is NP-hard,[16] we cannot hope for an algorithm whose run-time is polynomial in n, m and k in the worst case.

The worst-case run-time complexity of the algorithm with the one-sided cutting strategy is $n^{O(n)}$ since in this case k is not reduced on recursive calls and so the depth of recursion depends on the degree to which the problem size is reduced at each level of recursion.

Thus which cutting strategy is best depends on the input. When k is small, the two-sided cutting strategy dominates but when the one-sided cutting strategy succeeds in reducing the problem size consistently, then the latter is more effective. Our empirical analysis sheds more insight on the trade-offs.

All of the techniques used by the algorithm use $O(n + m)$ space. Finally we note that Direct-SetBarrier takes as input a specific candidate k for the min-barrier. To find the barrier when k is unknown, we can use a divide and conquer approach. Initially, we know k is in the range $[1, |A|]$. We can try $k = |A|/2$; if we find a solution we then we know the range is $[1, |A|/2]$. If not, we know the range is $[|A|/2 + 1, k]$. We continue to reduce the range by a factor of 2 until the range is 1, at which point we know the min-barrier. This increases the run-time by a factor that is logarithmic in the input size.

4. Empirical Results

We implemented both of our algorithms for the Direct-SetBarrier problem, in order to study their efficiency in practice on biologically motivated data. Here we describe our experimental setup and protocol for generating problem instances and describe our results on the performance of our algorithm.

Implementation and experimental environment. Both algorithms were coded in C++ and compiled using g++ (GCC version 4.2.1). All experiments were run on our reference PCs with 2.4Ghz Intel Pentium IV processors with 256KB L2 cache and 1GB RAM, running SUSE Linux version 10.3.

Generation of problem instances. With the motivation of studying algorithm performance across a variety of problem instances, we randomly sampled five sequences for each of four different classes of non-coding RNA—*Transfer RNA, Transfer Messenger RNA, Ribonuclease P RNA,* and *5S Ribosomal RNA,*—found in the RNA STRAND database.[19] For each sequence, five MFE structures—with respect to number of base pairs—were determined using a modified version of the Nussinov-Jacobsen algorithm.[20] The modified algorithm stored all optimal paths within the traceback matrix. In this way, we were able to randomly sample five different MFE structures for the same sequence. Identical structures were discarded. Every possible pairing of structures for the same sequence formed a new problem instance. Thus, ten problem instances were created for each sequence, resulting in 200 problem instances overall. The distribution of sequence length and the resulting number of conflicting base pairs between paired structures can be seen in Fig. 3. In general, and as expected, the number of conflicting bases pairs increases with sequence length.

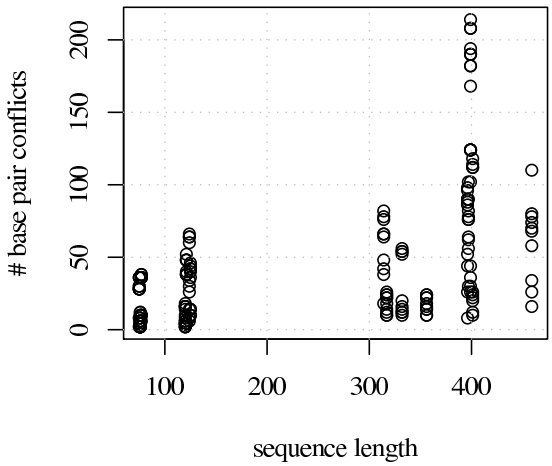

Fig. 3: Distribution of conflicting base pairs for generated problem instances.

Algorithm runtime performance. Both algorithms were run for a maximum of 1 CPU hour on each of the 200 hundred problem instances. The $n^{O(n)}$ algorithm found solutions to 183 instances, while the $O(n^{2k+2.5})$ algorithm found solutions to 184 instances. Interestingly, the $n^{O(n)}$ algorithm found solutions to three instances not found by the $O(n^{2k+2.5})$ algorithm; likewise, four instances were found by the $O(n^{2k+2.5})$

algorithm not found by the $n^{O(n)}$ algorithm. Of the instances that were solved, optimal barriers were found within 1 CPU second by both algorithms in 90% of the cases with barrier height ranging from 1 to 8. The barrier of harder instances ranged from 6 to 11, with a mean of 9. In general, the $O(n^{2k+2.5})$ algorithm was the best performing for harder instances. However, as can be seen in Fig. 4, both algorithms excelled for certain instances relative to one another.

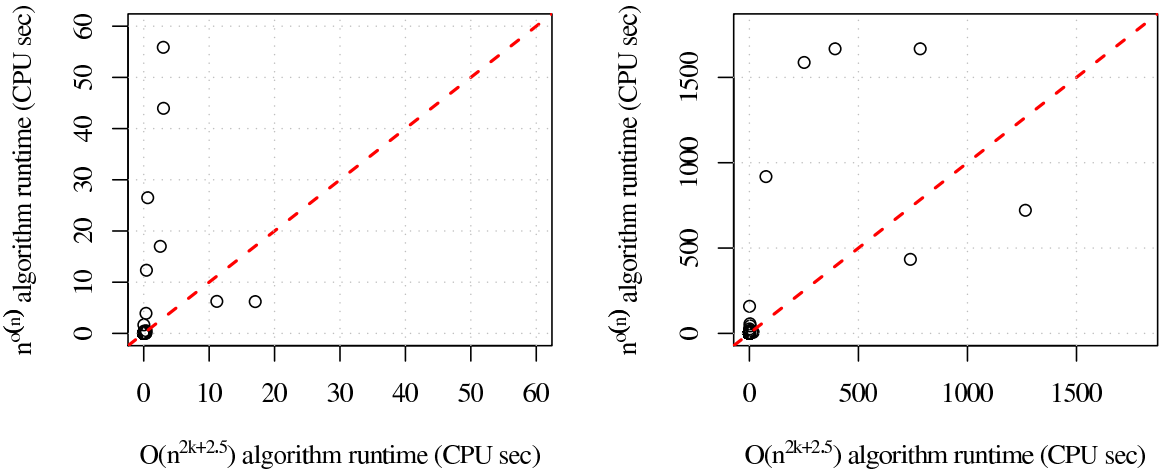

Fig. 4: The required time to find an optimal barrier pathway shown for two time scales.

The instances which failed to be solved within our cut-off time tended to have the highest number of conflicting base pairs. We also found that the instances which failed to be solved tended to have the largest minimally pairwise-optimal subproblems generated by the Split procedure. For each problem instance, we recorded the size of the maximum subproblem, as well as the average size of all subproblems, produced by the Split procedure at the top level of recursion, We measured size as number of base pairs. Fig. 5 shows the frequency of problem instances which have given maximum (left) and average (right) subproblem size. The problem instances which have maximum subproblems of size 200 or more were those that failed to be solved. Alternative methods for splitting or recursing on such subproblems would clearly be valuable.

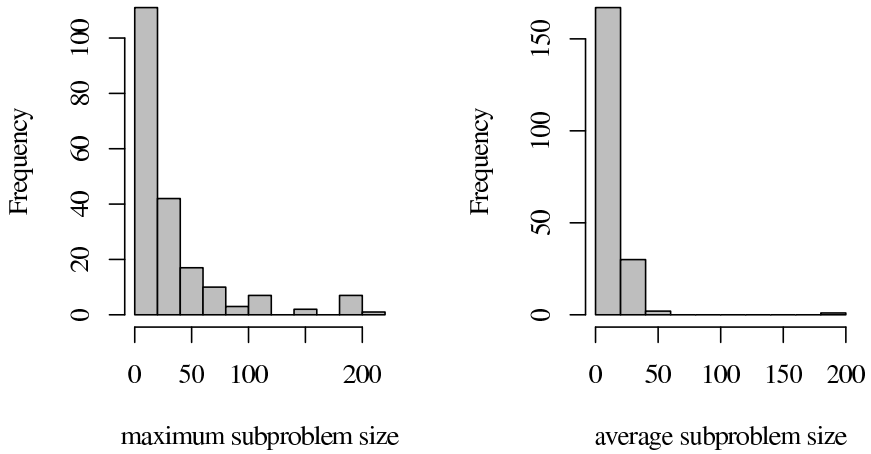

Fig. 5: Frequency of maximum (left) and average (right) subproblem sizes, measured as number of base pairs in the subproblem produced by the first call to the Split procedure for a given instance. The max and average are taken over all subproblems generated for a given instance.

5. Conclusion

We have presented new approaches for calculating energy barriers between RNA secondary structures using classical bipartite graph algorithms, and can prove that our methods find the min-barrier direct folding pathway between two MFE secondary structures. Our algorithms run very efficiently on structures of length up to 300nt and solved the vast majority of our generated instances of length up to 500nt.

Our methods show much promise for further development. They can be generalized to find energy barriers between non-MFE structures by exploiting the introduction of "artificial" nodes as in the one-sided cutting strategy. Our algorithms are highly amenable to parallelization: subproblems generated both by the Split procedure and the cutting strategy can be solved independently. Furthermore, there is more potential for creative means to split or otherwise reduce the size of a problem.

Even on instances that are intractable for our algorithm, preliminary analysis suggests that our algorithm might yield tight approximate bounds. For example, while the algorithm may be slow on some long instances when given the "true" min-barrier k, it can be fast when given $k-1$ and $k+1$, thereby narrowing the range to 2. The ability to establish a narrow range could be sufficient for some methods that approximate RNA energy landscapes.[2,11] We plan to investigate this further.

Other important directions for future work are to determine whether low-barrier pathways produced by our methods for the simple energy model are competitive with pathways found by state-of-the-art heuristics using a thermodynamic energy model, or whether our techniques can be extended to obtain exact energy barriers with more realistic energy models. Also of interest would be a follow-up of the Morgan and Higgs study to determine how barrier heights scale with sequence length, particularly for long sequences. Further testing of our method on more biologically-relevant pairs of structures will be necessary to answer these questions. Finally, development of efficient exact methods for determining min-barrier *indirect* pathways, i.e. pathways that allow temporary edges, is an interesting unresolved challenge.

Acknowledgments

We thank the anonymous reviewers for their insightful comments and suggestions for future work.

References

1. T. Baumstark, A. R. Schroder and D. Riesner, *EMBO J.* **16**, 599 (1997).
2. M. Geis, C. Flamm, M. T. Wolfinger, A. Tanzer, I. L. Hofacker, M. Middendorf, C. Mandl, P. F. Stadler and C. Thurner, *J. Mol. Biol.* **379**, 160 (2008).
3. A. Roth and R. R. Breaker, *Annual Rev. Biochem.* **78**, 305 (2009).
4. E. A. Schultes and D. P. Bartel, *Science* **289**, 448(July 2000).
5. T. B. J and R. R. Breaker, *Curr. Opin. Struct. Biol.* **15**, p. 342 (2005).
6. C. Yanofsky, *RNA* **13**, 1141 (2007).
7. S.-J. Chen and K. A. Dill, *Proc. Nat. Acad. Sci.* **97**, 646(January 2000).
8. R. Russell, X. Zhuang, H. Babcock, I. Millett, S. Doniach, S. Chu and D. Herschlag, *Proc. Nat. Acad. Sci.* **99**, 155 (2002).
9. I. Shcherbakova, S. Mitra, A. Laederach and M. Brenowitz, *Curr. Opin. Chem. Biol.* **12**, 655 (2008).
10. D. K. Treiber and J. R. Williamson, *Curr. Opin. Struc. Biol.* **11**, 309 (2001).
11. X. Tang, S. Thomas, L. Tapia, D. P. Giedroc and N. M. Amato, *J. Mol. Biol.* **381**, 1055 (2008).
12. C. Flamm, I. L. Hofacker, P. F. Stadler and M. T. Wolfinger, *Zeitschrift für Physikalische Chemie* **216**, 155 (2002).
13. S. R. Morgan and P. G. Higgs, *J. Phys. A: Math. Gen.* **31**, 3153 (1998).
14. M. T. Wolfinger, The energy landscape of RNA folding, Master's thesis, University Vienna (2001).
15. C. Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler and M. Zehl, *RNA* **7**, 254 (2001).
16. J. Maňuch, C. Thachuk, L. Stacho and A. Condon, *Proc. of the 15th Intl. Meeting on DNA Computing and Molecular Programming (DNA15)* (2009).
17. J. E. Hopcroft and R. M. Karp, *SIAM J. Comput.* **2**, 225 (1973).
18. R. Tarjan, *SIAM J. Comput.* **1**, 146 (1972).
19. M. Andronescu, V. Bereg, H. H. Hoos and A. Condon, *BMC Bioinformatics* **9**, 340 (2008).
20. R. Nussinov and A. B. Jacobson, *Proceedings of the National Academy of Sciences of the United States of America* **77**, 6309 (1980).