amcs

TABU SEARCH FOR THE RNA PARTIAL DEGRADATION PROBLEM

AGNIESZKA RYBARCZYK [a,b], ALAIN HERTZ [c], MARTA KASPRZAK [a,b], JACEK BLAZEWICZ [a,b,*]

[a] Institute of Computing Science
Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland
e-mail: jblazewicz@cs.put.poznan.pl

[b] Institute of Bioorganic Chemistry
Polish Academy of Sciences, Noskowskiego 12, 61-704 Poznan, Poland

[c] Department of Mathematics and Industrial Engineering
Polytechnique Montreal/GERAD, Montreal, Canada

In recent years, a growing interest has been observed in research on RNA (ribonucleic acid), primarily due to the discovery of the role of RNA molecules in biological systems. They not only serve as templates in protein synthesis or as adapters in the translation process, but also influence and are involved in the regulation of gene expression. The RNA degradation process is now heavily studied as a potential source of such riboregulators. In this paper, we consider the so-called RNA partial degradation problem (RNA PDP). By solving this combinatorial problem, one can reconstruct a given RNA molecule, having as input the results of the biochemical analysis of its degradation, which possibly contain errors (false negatives or false positives). From the computational point of view the RNA PDP is strongly NP-hard. Hence, there is a need for developing algorithms that construct good suboptimal solutions. We propose a heuristic approach, in which two tabu search algorithms cooperate, in order to reconstruct an RNA molecule. Computational tests clearly demonstrate that the proposed approach fits well the biological problem and allows to achieve near-optimal results. The algorithm is freely available at http://www.cs.put.poznan.pl/arybarczyk/tabusearch.php.

Keywords: RNA degradation, tabu search, bioinformatics.

1. Introduction

In the last two decades, there has been a rapid progress in computational molecular biology. Many problems that have arisen in this discipline have been classified as computationally hard (i.e., unlikely to be solved optimally in polynomial time). We consider one of them, namely, the *RNA partial degradation problem* (*RNA PDP* for short), proved to be strongly NP-hard, in which the primary actor is the ribonucleic acid (RNA) subjected to a nonenzymatic hydrolysis experiment (Blazewicz *et al.*, 2011).

RNA molecules play an essential role in a large variety of biological processes (Zok *et al.*, 2015), such as regulation of gene expression, protein synthesis or RNA degradation (Deutscher, 2003; Jankowiak *et al.*,

2004; 2005; Podkowinski *et al.*, 2009; Szostak *et al.*, 2014; Rybarczyk *et al.*, 2015; Kuppusamy and Mahendran, 2016). RNA degradation (cleavage of RNA into fragments) is a major component of RNA metabolism. It secures the balance between transcription and RNA decay pathways and provides cell homeostasis (Nowacka *et al.*, 2012). In fulfilling its role, the RNA degradation machinery has to distinguish between a set of molecules being unnecessary at certain conditions or defective and those essential for a proper cell functioning. Unfortunately, it still remains to be established how RNA degradation pathways control such higher level functions, namely, which specific RNAs involved in cellular differentiation and functions are targeted by RNA degradation machinery and which stay intact (Chanfreau, 2015).

What is more, it has been shown that not

*Corresponding author

all redundant RNA fragments are rapidly removed (Jackowiak *et al.*, 2011). Some of the cleavage products are stable and display regulatory functions through acting as translational inhibitors or signaling molecules (Zhang *et al.*, 2009; Bibillo *et al.*, 1999; 2000; Kierzek, 1992; 2001; Ender *et al.*, 2008; Haussecker *et al.*, 2010). These findings promote further research on RNA degradation which is essential for broadening our knowledge on physiological functions of RNA.

The biological process described above is analyzed at the biochemical level. However, the data it generates must be also studied at the computational level because their quantity and interdependence make it unfeasible for biochemists to analyze them manually.

Our focus is on biochemical experiments that use *in vitro* systems, since it is not possible to study all aspects of this process in a living organism using methods currently available. Blazewicz *et al.* (2011) analyzed the degradation patterns of two artificial RNA molecules applying commonly used experimental methods (Dutkiewicz and Ciesiolka, 2005; Rybarczyk *et al.*, 2016; Adachi and Yu, 2014). As a result of the partial degradation process, many copies of an RNA molecule are cleaved into a collection of fragments of the original molecule. Based on the data obtained, they formulated (on a computational level) a new strongly NP-hard problem, called the RNA PDP, which is to reconstruct an RNA molecule using the results of the biochemical analysis of its degradation. The same authors developed an exact algorithm for the RNA PDP. Given that this exact algorithm is not capable of handling large instances, we propose here to solve the problem with a heuristic method, based on two cooperative tabu search algorithms. We assume that the available results of the degradation process possibly contain false negatives (i.e., missing elements) and false positives (i.e., falsely reported elements).

The organization of the paper is as follows. In Section 2 we give a precise definition of the RNA PDP, the proposed heuristic algorithm is presented in Section 3, and computational results are given in Section 4.

2. RNA partial degradation problem

The degradation process of an RNA molecule can be summarized as follows. The input molecule of the full length (in many copies) is first broken at *primary cleavage sites*, which gives rise to a collection of *primary fragments*. These fragments are then broken at *secondary cleavage sites*, what creates *secondary fragments*. Hence, the result of the degradation process is a set of primary and secondary fragments, which comes from two separate experiments: involving multi-labeled RNA, where labeled nucleotides are randomly introduced along the RNA molecule and single-labeled RNA, which

contain labeled $5'$ end of the RNA molecule (let us say, its "left" end). Each primary fragment is assumed to cleave at most once. The only information available for every fragment is its length, and whether or not it contains the "left" end of the input RNA molecule. It is not known whether a fragment is primary or secondary. The objective of the RNA PDP is to reconstruct the original molecule by determining from this limited information the exact positions of the primary and secondary cleavage sites. More details can be found in the work of Blazewicz *et al.* (2011).

We now give a mathematical formulation of the problem. Assume that the analyzed molecule has length L, and that we are given the multiset (where multiple occurrences of elements are allowed) $\mathcal{D}$ of fragment lengths resulting from the degradation process, as well as its subset $\mathcal{Z} \subseteq \mathcal{D}$ containing the lengths of those fragments having the "left" end of the input RNA molecule. Missing elements (i.e., false negatives) in $\mathcal{D}$ and $\mathcal{Z}$ are allowed, but (for the moment) not false positives, and we assume that each primary fragment cleaves at most once. We aim to determine two disjoint sets $\mathcal{P}_1$ and $\mathcal{P}_2$ of integers, where $\mathcal{P}_1$ stands for the set of primary cleavage sites, and $\mathcal{P}_2$ for the set of secondary ones. For a set $\mathcal{P}$ of integers in $\{1, \ldots, L-1\}$, let $\mathcal{R}(\mathcal{P})$ denote the set of pairs $(x, y) \neq (0, L)$ such that $x, y \in \mathcal{P} \cup \{0, L\}$ and $x < y$. If $\mathcal{P}$ is a set of primary cleavage sites, then $\mathcal{R}(\mathcal{P})$ is the set of primary fragments (x, y), where x stands for the "left" end of the fragment, and y for the "right" end.

Definition 1. Let L be a positive integer, C a non-negative integer, and let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two sets of integers such that $0 < x < L$ for all $x \in \mathcal{P}_1 \cup \mathcal{P}_2$. The pair $(\mathcal{P}_1, \mathcal{P}_2)$ is *C-consistent* with $\mathcal{D}$ and $\mathcal{Z}$ if the following constraints are satisfied:

There is a function $f : \mathcal{R}' \to \mathcal{P}_2$
$\quad$ between a subset $\mathcal{R}' \subseteq \mathcal{R}(\mathcal{P}_1)$ and $\mathcal{P}_2$
$\quad$ such that $x < f(x, y) < y$, $\forall (x, y) \in \mathcal{R}'$, $\quad$ (1)

$$\mathcal{D} \subseteq \mathcal{D}' = \bigcup_{(x,y)\in\mathcal{R}(\mathcal{P}_1)} \{y - x\}, \qquad (2)$$

$$\mathcal{Z} \subseteq \mathcal{Z}' = \mathcal{P}_1 \cup \bigcup_{(0,y)\in\mathcal{R}'} \{f(0, y)\}$$
$$\cup \bigcup_{(x,y)\in\mathcal{R}'} \{y - f(x, y), f(x, y) - x\}, \qquad (3)$$

$$|\mathcal{D}'| - |\mathcal{D}| + |\mathcal{Z}'| - |\mathcal{Z}| \le C. \qquad (4)$$

Set $\mathcal{R}'$ in (1) contains the primary fragments that broke into smaller secondary fragments. For every

$(x, y) \in \mathcal{R}'$, $f(x, y) \in \mathcal{P}_2$ is the location of the secondary cleavage on fragment (x, y). The fact that f is a function enforces the requirement that primary fragments cleave at most once. Multiset $\mathcal{D}'$ in (3) contains the lengths of all primary fragments in $\mathcal{R}(\mathcal{P}_1)$, and of all secondary fragments $(x, f(x, y))$ and $(f(x, y), y)$ resulting from a secondary cleavage at position $f(x, y)$ on $(x, y) \in \mathcal{R}'$. Since we assume no false positive, it is imposed that $\mathcal{D}'$ contains multiset $\mathcal{D}$ of fragment lengths resulting from the degradation process. Set $\mathcal{Z}'$ in (2) contains the lengths of all primary and secondary segments with the left end (position 0) of the the input RNA molecule. Finally, since missing elements in $\mathcal{D}$ and $\mathcal{Z}$ are allowed, we aim to minimize the total number of false negatives. Constraint (4) imposes an upper bound C on the number of missing elements in $\mathcal{D}$ and $\mathcal{Z}$. If $C = 0$ we get the *ideal problem* with no false negatives allowed. It is worth noting that, if an element of $\mathcal{Z}'$ is missing both in $\mathcal{Z}$ and $\mathcal{D}$, then the error is counted twice since $\mathcal{D}'$ contains $\mathcal{Z}'$.

Note that the lack of a secondary cleavage site in a primary fragment is not treated as an error. If every primary fragment is assumed to degrade into two secondary fragments (a case also observed in biology) we set $\mathcal{R}' = \mathcal{R}(\mathcal{P}_1)$ in constraint (1). Also, if false positives are allowed, then we do not impose $\mathcal{D} \subseteq \mathcal{D}'$ and $\mathcal{Z} \subseteq \mathcal{Z}'$ in constraints (3) and (2), while constraint (4) becomes

$$|\mathcal{D}'| + |\mathcal{D}| - 2|\mathcal{D}' \cap \mathcal{D}| + |\mathcal{Z}'| \\ + |\mathcal{Z}| - 2|\mathcal{Z}' \cap \mathcal{Z}| \le C. \quad (4')$$

The RNA PDP can now be formulated as follows.

RNA PDP.
Instance: A positive integer L, non-negative integer C, multiset $\mathcal{D}$ and set $\mathcal{Z}$ of integers such that $0 < x < L$ for all $x \in \mathcal{D}$, and $\mathcal{Z} \subseteq \mathcal{D}$.
Objective: Find two sets $\mathcal{P}_1$ and $\mathcal{P}_2$ such that $(\mathcal{P}_1, \mathcal{P}_2)$ is C-consistent with $\mathcal{D}$ and $\mathcal{Z}$.

The following example illustrates the problem.

Example 1. Consider the parameter $L = 4653$, $C \ge 5$, $\mathcal{Z} = \{11, 435, 1248, 1254, 4554\}$ and $\mathcal{D} = \{11, 16, 83, 154, 424, 435, 886, 890, 1002, 1035, 1248, 1254, 1269, 1694, 2216, 2271, 2283, 2370, 3233, 3300, 4119, 4218, 4554\}$. We assume here that all primary fragments have broken into exactly two parts due to the secondary cleavages.

A possible solution is depicted in Fig. 1, with $\mathcal{P}_1 = \{435, 2283, 4554\}$ as a set of primary cleavage sites, and $\mathcal{P}_2 = \{11, 1248, 1254, 2129, 2651, 3552, 3668, 3763, 4637\}$ as a set of secondary ones. The pair $(\mathcal{P}_1, \mathcal{P}_2)$ is 5-consistent with $\mathcal{D}$ and $\mathcal{Z}$, since we have five missing fragment lengths: one in $\mathcal{Z}$ (2283) and four in $\mathcal{D}$ (99,

1480, 1848, 2002). $\blacklozenge$

The decision version of the RNA PDP is to determine whether there is a C-consistent pair $(\mathcal{P}_1, \mathcal{P}_2)$ with $\mathcal{D}$ and $\mathcal{Z}$. It was proved by Blazewicz *et al.* (2011) that the problem is strongly NP-complete when no errors are allowed (i.e., when $C = 0$). The computational complexity of the modified problem with $\mathcal{R}' = \mathcal{R}(\mathcal{P}_1)$ is not formally determined yet, but presumably it remains strongly NP-complete even without any errors allowed. The main difficulty of the basic problem (constraints (1)–(4)) lies in coupling secondary fragments, even if we know all of them and the set of intervals they should fit in. In the modified problem we stay with the same task, it is a similar situation as in the strongly NP-complete problem numerical matching with target sums (Garey and Johnson, 1979).

The idea behind the RNA partial degradation problem, which consists in exploiting information about lengths of fragments defined by pairs of cut points located within a nucleic acid sequence, makes the problem somewhat similar to DNA mapping problems: the partial digest problem (PDP) and its newer version, the simplified partial digest problem (SPDP) (Blazewicz *et al.*, 2001). For the moment, a dependence between the combinatorial PDP and RNA PDP, as well as the SPDP, that could have an impact on determining computational complexity of the former problem (open from the computational complexity point of view), is not yet known and is an interesting question for further studies.

3. Tabu search approach for the RNA PDP

The proposed heuristic algorithm for solving the RNA PDP is based on the tabu search metaheuristic, which is one of the most frequently used in combinatorial optimization (Glover, 1990; Glover *et al.*, 1995; Glover and Laguna, 1997; Bilski and Wojciechowski, 2016; Yao *et al.*, 2014). This choice has been motivated by high-quality results this metaheuristic reached in solving a problem of reconstructing a DNA sequence with false negatives and false positives (Blazewicz *et al.*, 2005). We
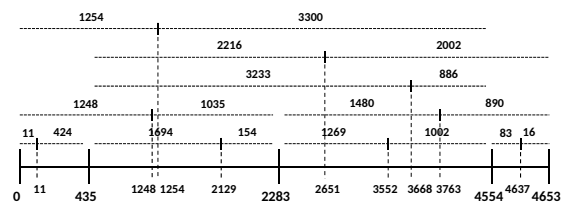

Fig. 1. Possible solution to the RNA PDP for the example considered. The primary cleavage sites (elements of $\mathcal{P}_1$) are represented by vertical solid lines while the secondary ones (elements of $\mathcal{P}_2$) by vertical dashed lines.

suppose that every primary fragment breaks into smaller fragments. As mentioned at the end of this section, this assumption can easily be modified for dealing with the case where not all primary fragments have a secondary cleavage site.

Let $\mathcal{D}$ be the multiset of (primary and secondary) fragment lengths resulting from the degradation process of an RNA molecule of length L, and let $\mathcal{Z}$ be its subset containing the lengths of those fragments having the "left" end of the input RNA molecule. Let $S = (P_1^S, P_2^S)$ be a solution to the RNA PDP with a set $\mathcal{P}_1^S = \{p_1, \ldots, p_v\}$ of v primary cleavage sites. Assume $p_i < p_j$ for all $i < j$, and let $p_0 = 0$ and $p_{v+1} = L$. The primary cleavage sites in S create a set of $r = 2v + v(v-1)/2 = v(v+3)/2$ primary fragments. Note, that the input RNA molecule with left end p_0 and right end p_{v+1} is not considered a fragment. Remember that $\mathcal{R}(\mathcal{P}_1^S)$ is the set of pairs $(p_i, p_j) \neq (0, L)$ such that $0 \leq i < j \leq v+1$. For every primary fragment $(p_i, p_j) \in \mathcal{R}(\mathcal{P}_1^S)$, let s_{ij} be the position of the secondary cleavage site, which implies $p_i < s_{ij} < p_j$. We thus have a set $\mathcal{P}_2^S = \{s_{ij} : (p_i, p_j) \in \mathcal{R}(\mathcal{P}_1^S)\}$ of r secondary cleavage sites which give rise to a set of $2r$ secondary fragments. We denote by $\mathcal{D}_S$ the multiset of primary and secondary fragment lengths, which result from solution S, while $\mathcal{Z}_S$ contains only those with the "left" end at position 0. Hence, $|\mathcal{D}_S| = 3r$ and $|\mathcal{Z}_S| = 2v$.

To evaluate the quality of a solution S, we consider two functions : $F(S)$ is the number of elements that appear in $\mathcal{D}$ but not in $\mathcal{D}_S$ plus the number of elements that appear in $\mathcal{Z}$ but not in $\mathcal{Z}_S$; $G(S)$ is the number of elements that appear in $\mathcal{D}_S$ but not in $\mathcal{D}$ plus the number of elements that appear in $\mathcal{Z}_S$ but not in $\mathcal{Z}$. Following (4') we see that for solution S, the pair $(\mathcal{P}_1^S, \mathcal{P}_2^S)$ is $(F(S)+G(S))$-consistent with $\mathcal{D}$ and $\mathcal{Z}$.

The proposed algorithm is executed several times with various numbers v of primary cleavage sites, which we set in the following manner. Since the given sets $\mathcal{D}$ and $\mathcal{Z}$ possibly have false positives and false negatives, we can only estimate v. In an ideal situation, without false positives or negatives, we should have $|\mathcal{Z}| = 2v$ and $|\mathcal{D}| = 3v(v+3)/2$, which gives two estimates for v, v_1 based on the cardinality of $\mathcal{Z}$ and v_2 based on $\mathcal{D}$:

$$v_1 = \left\lfloor \frac{|\mathcal{Z}|}{2} + \frac{1}{2} \right\rfloor, \quad v_2 = \left\lfloor \frac{-9 + \sqrt{81 + 24|\mathcal{D}|}}{6} + \frac{1}{2} \right\rfloor.$$

If $v_1 \leq v_2$, we first apply our algorithm for $v = v_1 - c$ and $v = v_2 + c$, where c is a constant. Otherwise, we consider $v = v_2 - c$ and $v = v_1 + c$. Let v^* denote the value that provides a better solution. It is considered the starting point for further analysis. More precisely, it is decreased as long as better solutions are obtained. Next, the number of primary cleavage sites is set back to v^* and increased as long as better solutions are found.

For each value of v considered, we apply two tabu search algorithms: the first one, $TS_{primary}$, is dedicated to finding primary cleavage sites, while the second one, $TS_{secondary}$, looks for secondary cleavage sites. The former does not take into account secondary cleavage sites and the latter considers primary cleavage sites as fixed. In addition, we apply two heuristic algorithms, $IS_{primary}$ and $IS_{secondary}$, which role is to provide initial solutions to the tabu search algorithms.

Algorithm 1. General scheme of the method.

Input: $\mathcal{D}, \mathcal{Z}, L$, the range of values of v
Output: S_{best}

1: Set $F_{best} \leftarrow \infty, G_{best} \leftarrow \infty$
2: **for** every given number v of primary cleavage sites **do**
3:      Generate an initial set $\mathcal{P}_1$ of primary cleavage sites ($IS_{primary}$)
4:      Try to get a better set $\mathcal{P}_1^*$ of primary cleavage sites ($TS_{primary}$)
5:      Generate an initial set $\mathcal{P}_2$ of secondary cleavage sites, considering the primary ones in $\mathcal{P}_1^*$ as fixed ($IS_{secondary}$)
6:      Try to get a better set $\mathcal{P}_2^*$ of secondary cleavage sites, without modifying the primary ones ($TS_{secondary}$), and let $S_v^* = (\mathcal{P}_1^*, \mathcal{P}_2^*)$ be the resulting solution
7:      **if** $F(S_v^*) + G(S_v^*) < F_{best} + G_{best}$ **then**
8:          Set $F_{best} \leftarrow F(S_v^*), G_{best} \leftarrow G(S_v^*), S_{best} \leftarrow S_v^*$
9:      **end if**
10: **end for**

Note, that for a fixed number v of primary cleavage sites and solution S, the addition to $\mathcal{D}_S$ of an element in $\mathcal{D} \setminus \mathcal{D}_S$ decreases both $F(S)$ and $G(S)$. Similarly, an increase of $F(S)$ results in an increase of $G(S)$. We therefore use only $F(S)$ to compare solutions with the same number of cleavage sites. Function $G(S)$ helps to determine the best solution among all those obtained with various values of v.

Each time a number is inserted/deleted to/from a set or multiset, this means that its single occurrence is inserted or deleted.

$IS_{primary}$ builds an initial set $\mathcal{P}_1$ of primary cleavage sites as follows. Each time an element d is added to $\mathcal{P}_1$, the fragment lengths d and $L-d$ are not further considered in $\mathcal{D}$, and d is not further considered in $\mathcal{Z}$. Starting from the empty set, elements are added to $\mathcal{P}_1$ in the following order. First, every element $z \in \mathcal{Z}$ is considered a primary cleavage site if its complement $L - z$ belongs to $\mathcal{D}$. Then, if there are $z \in \mathcal{Z}$ and $d \in \mathcal{D}$ such that both $d' = z + d$ and $L - d'$ belong to $\mathcal{D}$, we add position d' to $\mathcal{P}_1$. Next, every element d of $\mathcal{D}$ such that $L - d$ also belongs to $\mathcal{D}$ is added to $\mathcal{P}_1$. Finally, if necessary, elements of $\mathcal{Z}$ and $\mathcal{D}$ are added (in the order of non-increasing values of the

elements) to $\mathcal{P}_1$. The algorithm stops when $\mathcal{P}_1$ contains v elements.

Algorithm 2. $\text{IS}_{\text{primary}}$ (Generate an initial set $\mathcal{P}_1$ of primary cleavage sites).

Input: $\mathcal{D}, \mathcal{Z}, L, v$
Output: $\mathcal{P}_1$
1: Set $num \leftarrow 0, \mathcal{P}_1 \leftarrow \emptyset, \mathcal{D}_0 \leftarrow D, \mathcal{Z}_0 \leftarrow Z$
2: **while** $num < v$ and there are $z \in Z_0$ and $d \in D_0$ such that $z + d = L$ **do**
3:     Add z to $\mathcal{P}_1$ and set $num \leftarrow num + 1$
4:     Remove z from $\mathcal{Z}_0$ and $\mathcal{D}_0$ and d from $\mathcal{D}_0$
5: **end while**
6: **while** $num < v$ and there are $d, d', d'' \in D_0$ and $z \in Z_0$ such that $z + d = d'$
    and $d' + d'' = L$ **do**
7:     Add d' to $\mathcal{P}_1$ and set $num \leftarrow num + 1$
8:     Remove d' and d'' from $\mathcal{D}_0$
9: **end while**
10: **while** $num < v$ and there are $d, d' \in D_0$ such that $d + d' = L$ **do**
11:     Add $\max\{d, d'\}$ to $\mathcal{P}_1$ and set $num \leftarrow num + 1$
12:     Remove d and d' from $\mathcal{D}_0$
13: **end while**
14: **while** $num < v$ and $\mathcal{Z}_0 \neq \emptyset$ **do**
15:     Add the largest element z of $\mathcal{Z}_0$ to $\mathcal{P}_1$ and set $num \leftarrow num + 1$
16:     Remove z from $\mathcal{Z}_0$ and $\mathcal{D}_0$
17: **end while**
18: **while** $num < v$ and $\mathcal{D}_0 \neq \emptyset$ **do**
19:     Add the largest element d of $\mathcal{D}_0$ to $\mathcal{P}_1$ and set $num \leftarrow num + 1$
20:     Remove d from $\mathcal{D}_0$
21: **end while**

In what follows, $S(\mathcal{P})$ denotes the solution having $\mathcal{P}$ as a set of primary cleavage sites (and no secondary cleavage). Let $\mathcal{P}_1 = \{p_1, \ldots, p_v\}$ be the output of the $\text{IS}_{\text{primary}}$ algorithm, with $p_i < p_j$ for all $i < j$. The tabu search algorithm $\text{TS}_{\text{primary}}$ tries to generate a better set $\mathcal{P}_1^*$. This is done as follows. Moves to neighbor solutions are defined as a shift of a cleavage site to the left or to the right within the RNA molecule. The given new locations for a cleavage currently at position p_i are all integers in $[p_{i-1} + 1, p_{i+1} - 1]$, except for p_i. We try all such moves and perform the best "non-tabu" one. The tabu restrictions are contained in matrix M with v rows and $L - 1$ columns, where $M_{i,j}$ denotes the iteration number before which it is forbidden to move the i-th primary cleavage to position j. Initially, all $M_{i,j}$ are set to 0, and if the i-th primary cleavage (currently at position p_i) is moved to a new position at iteration $Iter$, we set M_{i,p_i} equal to $Iter + \lceil \sqrt{v} \rceil$ to prevent cycling, i.e., endless executing the same sequence of moves (revisiting the same solutions). The tabu status of a move is canceled

if the solution resulting from such a move is better than the current best known solution. $\text{TS}_{\text{primary}}$ stops after $|\mathcal{D}|$ iterations.

Algorithm 3. $\text{TS}_{\text{primary}}$ (Try to get a better set $\mathcal{P}_1^*$ of primary cleavage sites).

Input: $\mathcal{D}, L, v, \mathcal{P}_1 = \{p_1, \ldots, p_v\}$ with $p_i < p_j$ for all $i < j$
Output: $\mathcal{P}_1^*$
1: Set $\mathcal{P}_1^* \leftarrow \mathcal{P}_1, p_0 \leftarrow 0, p_{v+1} \leftarrow L$
2: Initialize the tabu matrix M with zero entries
3: **for** $Iter = 1$ to $|\mathcal{D}|$ **do**
4:     Set $F' \leftarrow \infty$ (best value of a neighbor solution)
5:     **for** every $i = 1, \ldots, v$ **do**
6:         **for** every $q \in \{p_{i-1} + 1, \ldots, p_i - 1\} \cup \{p_i + 1, \ldots, p_{i+1} - 1\}$ **do**
7:             Let S_{iq} be the solution obtained by replacing p_i with q
8:             **if** $M_{iq} \leq Iter$ **or** $\text{F}(S_{iq}) < \text{F}(S(\mathcal{P}_1^*))$ **then**
9:                 **if** $\text{F}(S_{iq}) < F'$ **then**
10:                     Set $F' \leftarrow \text{F}(S_{iq}), \mathcal{P}_1' \leftarrow (\mathcal{P}_1 \setminus \{p_i\}) \cup \{q\}$
11:                 **end if**
12:             **end if**
13:         **end for**
14:     **end for**
15:     **if** $\text{F}(S(\mathcal{P}_1')) < \text{F}(S(\mathcal{P}_1^*))$ **then**
16:         Set $\mathcal{P}_1^* \leftarrow \mathcal{P}_1'$
17:     **end if**
18:     Set $\mathcal{P}_1 \leftarrow \mathcal{P}_1'$ and update the tabu matrix M
19: **end for**

The output $\mathcal{P}_1^*$ of $\text{TS}_{\text{primary}}$ is now considered a fixed set of primary cleavage sites. They give rise to the set $\mathcal{R}(\mathcal{P}_1^*)$ of primary fragments. We now determine secondary cleavage sites by defining $f(x, y)$ for a subset $\mathcal{R}'$ of $\mathcal{R}(\mathcal{P}_1^*)$ so that $x < f(x, y) < y$ for all $(x, y) \in \mathcal{R}'$. Let $\mathcal{D}_0 \leftarrow \mathcal{D} \setminus \{y - x : (x, y) \in \mathcal{R}(\mathcal{P}_1^*)\}$ be the multiset of fragment lengths in $\mathcal{D}$ that do not correspond to lengths of primary fragments. Let $\mathcal{E}$ be the subset of primary fragment lengths that can be obtained by summing two elements of $\mathcal{D}_0$. For every $e \in \mathcal{E}$, let m_e denote the number of different pairs d, d' of elements of $\mathcal{D}_0$ with $d \leq d'$ and $d + d' = e$. The elements e of $\mathcal{E}$ are then considered by non-decreasing values of m_e. For every $e \in \mathcal{E}$, we look for two elements d, d' in $\mathcal{D}_0$ and a primary fragment (x, y) not yet in $\mathcal{R}'$ such that $e = d + d' = y - x$. If we succeed, we remove d, d' from $\mathcal{D}_0$, add (x, y) to $\mathcal{R}'$, and fix a secondary cleavage site on (x, y): if $x = 0$ and d or d' belongs to $\mathcal{Z}$, say d, we set $f(0, y) = d$; otherwise, we set $f(x, y) = x + \min\{d, d'\}$.

We start the $\text{TS}_{\text{secondary}}$ algorithm with the set $\mathcal{P}_1^*$ of primary cleavage sites produced by $\text{TS}_{\text{primary}}$ and with the set $\mathcal{P}_2 = \{f(x, y) : (x, y) \in \mathcal{R}'\}$ of secondary cleavage sites produced by $\text{IS}_{\text{secondary}}$, where $\mathcal{R}' \subseteq \mathcal{R}(\mathcal{P}_1^*)$. We

Algorithm 4. $IS_{secondary}$ (Generate an initial set $\mathcal{P}_2$ of secondary cleavage sites).

Input: $\mathcal{D}, \mathcal{Z}, \mathcal{P}_1^*$
Output: $S = (\mathcal{P}_1^*, \mathcal{P}_2)$
1: Set $\mathcal{D}_0 \leftarrow D \setminus \{y - x : (x,y) \in \mathcal{R}(\mathcal{P}_1^*)\}$, $\mathcal{R}_0 \leftarrow \mathcal{R}(\mathcal{P}_1^*)$ and $\mathcal{E} \leftarrow \emptyset$
2: **for** every $d, d' \in \mathcal{D}_0$ with $d \leq d'$ and $e = d + d' \in \{y - x : (x,y) \in \mathcal{R}_0\}$ **do**
3:     **if** $e \in \mathcal{E}$ **then**
4:        Set $m_e \leftarrow m_e + 1$
5:     **else**
6:        Add e to $\mathcal{E}$ and set $m_e \leftarrow 1$
7:     **end if**
8: **end for**
9: Order $\mathcal{E}$ so that $\mathcal{E} = \{e_1, \ldots, e_{|\mathcal{E}|}\}$ with $m_{e_i} \leq m_{e_j}$ for all $i < j$
10: **for** $i = 1, \ldots, |\mathcal{E}|$ **do**
11:     **if** there are $d \leq d' \in \mathcal{D}_0$ and $(x,y) \in \mathcal{R}_0$ with $d + d' = y - x = e_i$ **then**
12:        **if** $x = 0$ and $\{d, d'\} \cap \mathcal{Z} \neq \emptyset$ **then**
13:           Choose the largest $\tilde{d}$ in $\{d, d'\} \cap \mathcal{Z}$ and set $f(0, y) \leftarrow \tilde{d}$
14:        **else**
15:           Set $f(x, y) \leftarrow x + d$
16:        **end if**
17:        Remove d, d' from $\mathcal{D}_0$ and (x, y) from $\mathcal{R}_0$
18:     **end if**
19: **end for**

try to improve $\mathcal{P}_2$ by using the tabu search metaheuristic for $2|\mathcal{D}|$ iterations. At each iteration, we generate three sets $\mathcal{N}_1(S)$, $\mathcal{N}_2(S)$ and $\mathcal{N}_3(S)$ of solutions in the neighborhood of the current solution S. These sets are defined as follows, where

$$
\begin{aligned}
\mathcal{D}_0 &= \mathcal{D} \setminus \Big(\bigcup_{(x,y) \in \mathcal{R}(\mathcal{P}_1^*)} \{y - x\} \\
&\quad \cup \bigcup_{(x,y) \in \mathcal{R}'} \{y - f(x,y), f(x,y) - x\} \Big), \\
\mathcal{Z}_0 &= \mathcal{Z} \setminus \Big(\mathcal{P}_1^* \cup \bigcup_{(0,y) \in \mathcal{R}'} \{f(0,y)\} \Big)
\end{aligned}
$$

are the sets of fragment lengths in $\mathcal{D}$ and $\mathcal{Z}$, respectively, that are not yet used by primary or secondary fragments.

(i) The solutions in $\mathcal{N}_1(S)$ are obtained from S by removing a secondary cleavage site on a primary fragment $(x,y) \in \mathcal{R}'$ and by adding a secondary cleavage site on two primary fragments (x', y') and (x'', y'') not belonging to $\mathcal{R}'$. This is done only if there exist two integers d, d' in $\mathcal{D}_0$ such that $f(x,y) - x + d' = y' - x'$ and $y - f(x,y) + d'' = y'' - x''$. If these conditions are met, we replace (x,y) by (x', y')

and (x'', y'') in $\mathcal{R}'$ and we fix the secondary cleavage sites on (x', y') and (x'', y'') as follows: if $x' = 0$ and $d' \in \mathcal{Z}_0$, we set $f(x', y') = d'$, otherwise we set $f(x', y') = x' + \min\{f(x,y) - x, d'\}$; similarly, if $x'' = 0$ and $d'' \in \mathcal{Z}_0$, we set $f(x'', y'') = d''$, otherwise we set $f(x'', y'') = x'' + \min\{y - f(x,y), d''\}$.

(ii) The solutions in $\mathcal{N}_2(S)$ are obtained from S by adding a secondary cleavage site on a primary fragment $(x,y) \notin \mathcal{R}'$. If $x = 0$, this is done only if there are $z \in \mathcal{Z}_0$ and $d \in \mathcal{D}_0$ such that $z + d = y$, in which case we add $(0, y)$ to $\mathcal{R}'$ and fix the secondary cleavage site at $f(x,y) = z$. If $x > 0$, we have to find two integers d, d' in $\mathcal{D}_0$ such that $d + d' = y - x$, and if we succeed, we add (x, y) to $\mathcal{R}'$ and fix the secondary cleavage site at $f(x,y) = x + \min\{d, d'\}$.

(iii) The solutions in $\mathcal{N}_3(S)$ are obtained from S by removing a secondary cleavage site on a primary fragment $(x,y) \in \mathcal{R}'$ and adding one on $(x', y') \notin \mathcal{R}'$. If $x' = 0$, this is done if there is $z \in \mathcal{Z}_0$ such that $z + f(x,y) - x$ or $z + y - f(x,y)$ is equal to y', in which case we replace (x,y) by $(0, y')$ in $\mathcal{R}'$ and fix the secondary cleavage site on $(0, y')$ at $f(0, y') = z$. If $x' > 0$, we have to find an integer $d \in \mathcal{D}_0$ such that $d + f(x,y) - x$ or $d + y - f(x,y)$ is equal to $y' - x'$, and if we succeed, we replace (x,y) by (x', y') in $\mathcal{R}'$ and fix the secondary cleavage site on (x', y') at $f(x', y') = x' + d$.

Let $\mathcal{D} = \{d_1, \ldots, d_{|\mathcal{D}|}\}$. The tabu restrictions are contained in matrix M' with $|\mathcal{D}|$ rows and $|\mathcal{D}|$ columns, where M'_{ij} denotes the iteration number before which it is forbidden to combine two secondary fragments of lengths d_i and d_j, respectively, to obtain a primary fragment. Initially, all M'_{ij} are set to 0. Then, if the chosen move at iteration $Iter$ involves the removal of a primary fragment (x,y) from $\mathcal{R}'$, we set $M'_{ij} = M'_{ji} = Iter + 10$ for i, j such that $d_i = f(x,y) - x$ and $d_j = y - f(x,y)$. The tabu status of a move to a neighbor solution is canceled if the solution resulting from such a move is better than the current best known solution.

If all moves are tabu (have the tabu status as defined above), we delete one of the most rarely (since the start of the $TS_{secondary}$ procedure) relocated secondary cleavage sites from S, say $f(x,y)$, which means that (x,y) is removed from $\mathcal{R}'$. Otherwise, the best non-tabu neighbor S' in $\mathcal{N}_1(S) \cup \mathcal{N}_2(S) \cup \mathcal{N}_3(S)$ becomes the new current solution for the next iteration.

Let S_v^* denote the best solution found after $2|\mathcal{D}|$ iterations, let $\mathcal{R}_0^*$ be the set of primary fragments in S_v^* with no secondary cleavage site, and let $\mathcal{D}_0^*$ and $\mathcal{Z}_0^*$ be the set of fragment lengths in $\mathcal{D}$ and $\mathcal{Z}$, respectively, that are not used by primary or secondary fragments in S_v^*.

If $\mathcal{R}_0^* \neq \emptyset$ and $\mathcal{D}_0^* \cup \mathcal{Z}_0^* \neq \emptyset$, we do the following. While there are primary fragments $(0, y)$ in $\mathcal{R}_0^*$ and z in $\mathcal{Z}_0^*$ with $z < y$, we fix a secondary cleavage site on $(0, y)$ at $f(0, y) = z$, and we remove $(0, y)$ from $\mathcal{R}_0^*$ and z from $\mathcal{Z}_0^*$. Then, while there are primary fragments (x, y) in $\mathcal{R}_0^*$ and d in $\mathcal{D}_0^*$ with $d < y - x$, we fix a secondary cleavage site on (x, y) at $f(x, y) = x + d$, and we remove (x, y) from $\mathcal{R}_0^*$ and d from $\mathcal{D}_0^*$.

Algorithm 5. $\text{TS}_{\text{secondary}}$ (Try to get a better set $\mathcal{P}_2^*$ of secondary cleavage sites).

Input: $\mathcal{D}, \mathcal{Z}, S = (\mathcal{P}_1^*, \mathcal{P}_2)$
Output: $S_v^* = (\mathcal{P}_1^*, \mathcal{P}_2^*)$

1: Set $S_v^* \leftarrow S$
2: Initialize the tabu matrix M' with zero entries
3: **for** $iter = 1$ to $2|\mathcal{D}|$ **do**
4:     Let $\mathcal{N}(S)$ be the set of non-tabu solutions in $\mathcal{N}_1(S) \cup \mathcal{N}_2(S) \cup \mathcal{N}_3(S)$.
5:     **if** $\mathcal{N}(S) \neq \emptyset$ **then**
6:        Let S' be a solution in $\mathcal{N}(S)$ with smallest value $\text{F}(S')$
7:        **if** $\text{F}(S') < \text{F}(S_v^*)$ **then**
8:           Set $S_v^* \leftarrow S'$
9:        **end if**
10:    **else**
11:        Set S' equal to the solution obtained from S by removing the most rarely relocated secondary cleavage site
12:    **end if**
13:    $S \leftarrow S'$ and update the tabu matrix M'
14: **end for**
15: Assign unused elements of $\mathcal{Z}$ to primary segments $(0, y)$ with no secondary cleavage site, and then assign unused elements of $\mathcal{D}$ to primary segments (x, y) with no secondary cleavage site

---

As already explained in the general scheme, the four algorithms $\text{IS}_{\text{primary}}$, $\text{TS}_{\text{primary}}$, $\text{IS}_{\text{secondary}}$, and $\text{TS}_{\text{secondary}}$ are applied sequentially for different numbers v of primary fragments. We illustrate the whole process using the instance from Example 1.

Example 2. As a reminder, we have $L = 4653$, $\mathcal{Z} = \{11, 435, 1248, 1254, 4554\}$, and $\mathcal{D} = \{11, 16, 83, 154, 424, 435, 886, 890, 1002, 1035, 1248, 1254, 1269, 1694, 2216, 2271, 2283, 2370, 3233, 3300, 4119, 4218, 4554\}$. Length 2283 is missing in $\mathcal{Z}$, while lengths 99, 1480, 1848, and 2002 are missing in $\mathcal{D}$, for a total of 5 false negatives.

Since $|\mathcal{Z}| = 5$ and $|\mathcal{D}| = 23$, we get $v_1 = v_2 = 3$. Assuming $c = 0$, we first set $v = 3$. $\text{IS}_{\text{primary}}$ chooses primary cleavages at positions 435 (instruction 2), 2283 (instruction 6), and 4554 (instruction 14), which gives $\mathcal{Z}_0 = \{11, 1248, 1254\}$, $\mathcal{D}_0 = \{11, 16, 83, 154, 424, 886, 890, 1002, 1035, 1248, 1254, 1269, 1694, 2216,$

3233, 3300$\}$ and a solution S of value $\text{F}(S) = 19$. $\text{TS}_{\text{primary}}$ does not modify this set $\mathcal{P}_1 = \{435, 2283, 4554\}$ of primary cleavage sites. Then, $\text{IS}_{\text{secondary}}$ chooses secondary cleavages at the following positions:

- 11 on primary fragment $(0, 435)$, since $11 + 424 = 435$;

- 589 (which corresponds to position 154 on primary fragment $(435, 2283)$) since $2283 - 435 = 1848 = 154 + 1694$;

- 1248 on primary fragment $(0, 2283)$, since $1248 + 1035 = 2283$;

- 1254 on primary fragment $(0, 4554)$, since $1254 + 3300 = 4554$;

- 1321 (which corresponds to position 886 on primary fragment $(435, 4554)$), since $4554 - 435 = 4119 = 886 + 3233$;

- 3285 (which corresponds to position 1002 on primary fragment $(2283, 4554)$), since $4554 - 2283 = 2271 = 1002 + 1269$;

- 4570 (which corresponds to position 16 on primary fragment $(4554, 4653)$), since $4653 - 4554 = 99 = 16 + 83$.

Thus, we get $\mathcal{Z}_0 = \emptyset$, $\mathcal{D}_0 = \{890, 2216\}$ and a solution S with $\text{F}(S) = 2$. $\text{TS}_{\text{secondary}}$ adds the two following secondary cleavage sites:

- 3173 (which corresponds to position 890 on primary fragment $(2283, 4653)$);

- 2651 (which corresponds to position 2216 on primary fragment $(435, 4653)$).

As a result, we get a solution S_3^* with $\text{F}(S_3^*) = 0$, while $\text{G}(S_3^*) = 5$ since there are 5 false negatives (one in $\mathcal{Z}$ and 4 in $\mathcal{D}$). Note, that although the positions of the secondary cleavage sites in S_3^* are not identical to those in Fig. 1, the assignment of secondary fragments to primary ones is the same. The difference is due to the lack of information about the order of the secondary fragments on the primary ones.

The four algorithms are then executed again with $v = 2$, and we obtain the solution S_2^* with $\text{F}(S_2^*) = 11$, $\text{G}(S_2^*) = 2$. In order to compare S_2^* with S_3^*, we use the sum of the two functions F and G. Since $0 + 5 < 11 + 2$, S_3^* is considered better than S_2^*.

The algorithm is then executed again with $v = 4$ and produces a solution S_4^* with $\text{F}(S_4^*) = 0$ and $\text{G}(S_4^*) = 22$. Hence S_3^* is again a better solution, and the output of the whole process is therefore S_3^*, which is the optimal solution.   ◆

As mentioned at the beginning of this section, the proposed algorithm assumes that all primary fragments break into smaller ones. If this is not the case, we propose the following modifications. The estimates v_1 and v_2 should be adjusted according to the probability that a primary fragment breaks into smaller ones. For example, if this probability is 0.5, we get the estimated values $|\mathcal{Z}| = 3v/2$ and $|\mathcal{D}| = v(v+3)$, which results in

$$v_1 = \left\lfloor \frac{2|\mathcal{Z}|}{3} + \frac{1}{2} \right\rfloor, \quad v_2 = \left\lfloor \frac{-3 + \sqrt{9 + 4|\mathcal{D}|}}{2} + \frac{1}{2} \right\rfloor.$$

We also recommend to increase the value of constant c extending the range of v. Algorithms $\text{IS}_{\text{primary}}$, $\text{TS}_{\text{primary}}$ and $\text{IS}_{\text{secondary}}$ do not require any modification, while instruction 15 of $\text{TS}_{\text{secondary}}$ can be removed. Note that this instruction has no impact on the total value $F(S) + G(S)$ of solution S. It divides a primary fragment into two secondary ones and the lengths of these secondary fragments are such that one is in $\mathcal{D} \cup \mathcal{Z}$ while the other is outside this set. Hence, $F(S)$ is decreased while $G(S)$ is increased by the same amount.

4. Computational results

In this section, we report computational experiments made on random instances, using a machine with the Intel Xeon E5-2670, 2.60 GHz processor, 16 GB of RAM and Linux operating system. The algorithms were implemented in C++.

We have generated RNA molecules of length $L = 5000$ and with numbers $p = 5, 10, 15$ or 20 of primary cleavage sites, based on values met in biological experiments (Blazewicz *et al.*, 2011; Jackowiak *et al.*, 2011; Rybarczyk *et al.*, 2016). The positions of the primary cleavages were chosen using a uniform distribution in the interval $[1, 4999]$. Also, for every instance and every primary fragment (x, y), we have generated a secondary cleavage site using a uniform distribution in the interval $[x + 1, y - 1]$.

The first data set considered contains instances without any error in the input sets $\mathcal{D}$ and $\mathcal{Z}$. The second data set contains instances with $5, 10, 15$ or 20 false negatives, these errors being obtained by randomly deleting elements from $\mathcal{D} \cup \mathcal{Z}$. The third set contains instances with $5, 10, 15$ or 20 false positives, where elements have been added to $\mathcal{D} \cup \mathcal{Z}$ using a uniform distribution in $[1, 4999]$. The last set contains instances with $e = 5, 10, 15$ or 20 false negatives, and the same number, respectively, of false positives, for a total of $2e$ errors. The heading of the columns of Tables 1–4 has the following meaning:

p : number of primary cleavage sites in the tested instance

Neg : number of false negatives

Pos : number of false positives

F_{best} : average value F_{best} obtained at the end of the algorithm

G_{best} : average value G_{best} obtained at the end of the algorithm

$v_1 - v_2$: initial range of values of v we apply our algorithm with (where constant c is equal to 0)

v : numbers of primary cleavage sites considered by the algorithm

F : average value $F(S_v^*)$ of the best solutions S_v^* obtained with v primary cleavage sites

G : average value $G(S_v^*)$ of the best solutions S_v^* obtained with v primary cleavage sites

$Hits$: number of instances, among the 10 tested ones, for which the best solution S_{best} was equal to S_v^*.

For each set of parameters (p, Neg, Pos) 10 random instances were generated and solved, and the presented results are mean values. Note, that the values in columns F and G do not necessarily correspond to average values taken on 10 instances. Particular instances can be solved with different ranges of values of v, only v_1 and v_2 are guaranteed to be used for all 10 instances.

Table 1 contains the results for the instances without any error. Most of these instances are solved optimally. The only exception from reaching ideal solutions appears for the largest instances, but even then the number of reconstructed cleavage sites is correct.

The results for instances with false negatives (and no false positive) are shown in Table 2 and Figure 2. We observe that the algorithm often produces solutions with the right number p of primary cleavage sites. Some solutions have only $p - 1$ primary cleavage sites, and a very limited number have $p - 2$ ones. This deviation of 2 units is observed only for the smallest instances with the largest number of false negatives ($p = 5$, $Neg = 20$) and is due to the fact that there is a big percentage of lacking elements in $\mathcal{D}$ and $\mathcal{Z}$. Figure 2 clearly illustrates the fact that our global criterion $F_{\text{best}} + G_{\text{best}}$ is almost equal to the number of errors for instances with up to 15 primary cleavages sites and at most 10 errors. Instances with a larger number of primary cleavage sites or with more false negatives appear to be more challenging.

The results are even better for instances with false positives (and no false negative). They appear in Table 3 and Figure 3. Hits are almost always associated with the proper value v of primary cleavage sites, and the total value $F_{\text{best}} + G_{\text{best}}$ is always very close to the total

Table 1. Results for instances without any error.

p	Neg	Pos	F_{best}	G_{best}	$v_1 - v_2$	v	F	G	$Hits$
5	0	0	0.0	0.0	5–5	4	18.0	0.0	0
						5	0.0	0.0	10
						6	0.0	23.0	0
10	0	0	0.0	0.0	10–10	9	33.0	0.0	0
						10	0.0	0.0	10
						11	0.0	37.7	0
15	0	0	0.0	0.0	15–15	14	48.0	0.0	0
						15	0.0	0.0	10
						16	0.1	52.4	0
20	0	0	2.2	3.5	20–20	19	62.1	0.5	0
						20	2.2	3.5	10
						21	0.1	68.3	0

number of errors. This better performance in comparison to the case with false negatives was previously observed on another problem from the bioinformatics area, namely sequencing by hybridization (Blazewicz and Kasprzak, 2012). Although both variants of the latter problem (with only false negatives and with only false positives) are strongly NP-hard, typical instances of both kinds are not equally hard to be processed by a sequencing algorithm (see, e.g., Blazewicz *et al.*, 1999; 2002). The reason is that a random false positive error is usually easier to be handled since it may not fit to the rest of the instance, while a false negative error makes the task more complex to guide the search towards an optimal solution.

The most general case with errors of both kinds is represented in Table 4 and Fig. 4. This case cumulates difficulties associated with both kinds of errors. Again, our algorithm often predicts the proper numbers of cleavage sites and finds most of the secondary cleavage sites.

Since the size of $\mathcal{D}$ is quadratic with respect to the number of primary cleavage sites, $F(S_v)$ is a convex decreasing function of v, while $G(S_v)$ is a convex increasing function of v. Hence, $F(S_v) + G(S_v)$ is a convex function and we are looking for its minimal value. A typical shape of these functions is shown in Fig. 5 for $p = 10$ and $Neg = Pos = 20$. The optimal solution we are looking for is approximately at the intersection of the curves $F(S_v)$ and $G(S_v)$. Note that we choose one of $v_1 - c, v_2 + c$ (or $v_2 - c, v_1 + c$) as a starting point for v which is then decreased and increased until we do not get any improvement. Since we try both directions, one of them moves the search towards the optimal value. The influence of IS_{primary}, TS_{primary}, $IS_{\text{secondary}}$, and $TS_{\text{secondary}}$ on the total process can be seen in Fig. 6, where we represent the values $F(S)$ reached by the four algorithms for instances with $Neg = 20$ false negatives and no false positive. We observe very good performance of
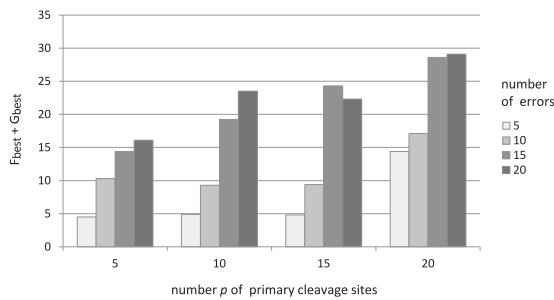

Fig. 2. Values of $F_{\text{best}} + G_{\text{best}}$ for instances with false negatives.
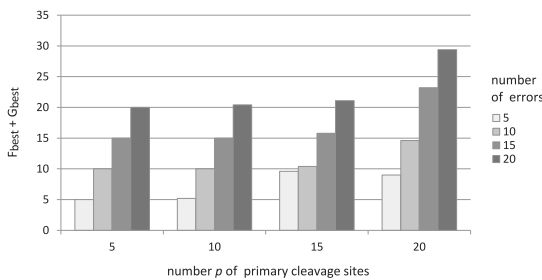

Fig. 3. Values of $F_{\text{best}} + G_{\text{best}}$ for instances with false positives.
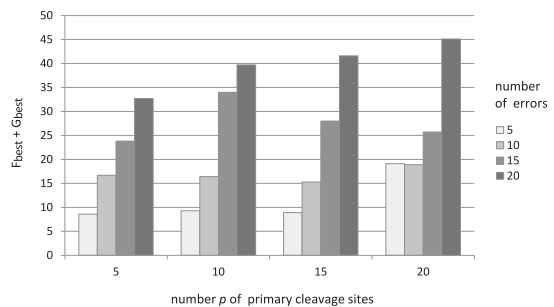

Fig. 4. Values of $F_{\text{best}} + G_{\text{best}}$ for instances with false negatives and false positives.

Table 2. Results for instances with false negatives.

p	Neg	Pos	F_{best}	G_{best}	$v_1 - v_2$	v	F	G	$Hits$
5	10	0	0.4	9.9	5–5	4	12.6	4.0	0
						5	0.4	9.9	10
						6	0.0	31.8	0
	15	0	4.0	10.4	4–4	3	22.6	2.3	0
						4	10.8	5.9	4
						5	0.8	14.7	6
						6	0.2	36.7	0
	20	0	7.3	8.8	4–4	2	29.0	1.0	0
						3	19.1	3.1	1
						4	9.2	8.2	7
						5	0.9	19.2	2
						6	0.0	40.0	0
10	10	0	0.0	9.3	10–10	9	27.4	3.7	0
						10	0.0	9.3	10
						11	0.0	46.9	0
	15	0	7.5	11.7	9–10	8	52.5	2.8	0
						9	26.0	6.6	3
						10	3.4	17.5	7
						11	0.4	52.0	0
	20	0	9.3	14.2	9–9	8	51.0	5.2	0
						9	24.1	8.9	4
						10	3.6	22.2	6
						11	1.7	57.9	0
15	10	0	0.0	9.4	15–15	14	41.9	3.3	0
						15	0.0	9.4	10
						16	0.0	61.8	0
	15	0	12.2	12.1	14–15	13	81.5	2.4	0
						14	39.1	5.0	3
						15	6.2	20.8	7
						16	3.1	70.0	0
	20	0	4.1	18.2	14–15	13	79.6	4.1	0
						14	38.3	8.3	1
						15	2.2	21.0	9
						16	0.5	71.6	0
20	10	0	6.1	11.0	20–20	18	113.0	0.0	0
						19	55.8	4.0	1
						20	3.0	14.3	9
						21	0.2	78.3	0
	15	0	12.8	15.8	19–20	18	111.3	3.9	0
						19	53.7	6.4	2
						20	7.9	23.5	8
						21	5.8	88.6	0
	20	0	4.4	24.7	19–20	19	51.6	8.6	0
						20	4.4	24.7	10
						21	0.8	88.4	0

the initial heuristics. Although the output of IS_{primary} is not markedly corrected by TS_{primary} (in the sense of the criterion function value), it finally appears to be quite appropriate to get near-optimal solutions. The role of $TS_{\text{secondary}}$ is better visible. While three of the curves increase with p, we see that $TS_{\text{secondary}}$ has values almost independent of p, which is our goal since the solution S we are looking for has values $F(S) = 0$ and $G(S) = 20$

that do not depend on the number of primary cleavage sites. Similar curves can be drawn for instances with both false negatives and false positives.

Figure 7 represents the total computing time needed to solve instances with negative and positive errors. Blazewicz *et al.* (2011) have developed an exact exponential-time algorithm for instances with false

Table 3. Results for instances with false positives.

p	Neg	Pos	F_{best}	G_{best}	$v_1 - v_2$	v	F	G	$Hits$
5	0	10	10.0	0.0	5–5	4	28.0	0.0	0
						5	10.0	0.0	10
						6	1.8	14.6	0
	0	15	15.0	0.0	5–6	4	32.8	0.1	0
						5	15.0	0.0	10
						6	6.5	13.8	0
	0	20	20.0	0.0	6–6	4	37.0	0.0	0
						5	20.0	0.0	10
						6	11.0	13.2	0
10	0	10	10.0	0.0	10–10	9	43.0	0.0	0
						10	10.0	0.0	10
						11	0.0	27.7	0
	0	15	15.0	0.0	10–10	9	47.8	0.0	0
						10	15.0	0.0	10
						11	1.0	23.2	0
	0	20	20.0	0.4	11–11	9	52.5	0.1	0
						10	20.0	0.4	10
						11	5.6	23.1	0
15	0	10	10.2	0.2	15–15	14	58.1	0.1	0
						15	10.2	0.2	10
						16	0.0	42.2	0
	0	15	15.3	0.5	15–15	14	62.8	0.0	0
						15	15.3	0.5	10
						16	0.2	37.4	0
	0	20	20.4	0.7	15–16	14	67.6	0.0	0
						15	20.4	0.7	10
						16	1.1	33.3	0
20	0	10	11.7	2.9	20–20	19	71.8	0.2	0
						20	11.7	2.9	10
						21	0.0	58.3	0
	0	15	15.0	8.2	20–20	19	76.8	0.3	0
						20	18.5	5.1	9
						21	0.8	53.7	1
						22	0.0	122.0	0
	0	20	20.6	8.8	20–21	19	83.9	2.6	0
						20	24.3	5.8	9
						21	0.8	48.9	1
						22	0.2	118.5	0

negatives, but no false positive. Computing times are shown in Fig. 8, using a logarithmic scale, for instances with $p = 5$ (curve 5-Ex) and $p = 10$ (curve 10-Ex) primary cleavage sites. The exact algorithm is not able to solve larger instances. For comparison, we also present the computing times of our algorithms.

5. Conclusion

In this paper, we have developed a heuristic algorithm, with two cooperating tabu search procedures, for the solution of the RNA partial degradation problem. The proposed algorithm can deal with both kinds of errors: false negatives and false positives. Computational tests have clearly shown that the solutions produced by our algorithm are of good quality, with numbers of cleavage sites close to the optimal ones. It should be stressed that the parameters used to generate the instances (number of cleavage sites, number of errors) are those met in the real world. Hence, the proposed algorithm will perform well in practice and will be useful in supporting analysis of biochemical data.

An exact algorithm exists for the case of only false negatives, but computing times become unacceptable for instances with more than 10 primary cleavage sites. Hence, the proposed algorithm is the only option for solving the problem with a lot of cleavage sites, and with false positives.

Table 4. Results for instances with false negatives and false positives.

p	Neg	Pos	F_{best}	G_{best}	$v_1 - v_2$	v	F	G	$Hits$
5	10	10	7.6	9.1	5–5	4	22.4	3.7	0
						5	8.9	8.3	9
						6	0.6	22.6	1
						7	0.0	45.0	0
	15	15	13.4	10.4	5–5	3	37.0	2.0	0
						4	25.5	5.9	1
						5	12.5	11.4	9
						6	4.6	25.9	0
	20	20	17.8	14.9	5–5	3	38.0	2.0	0
						4	28.7	8.2	3
						5	18.1	16.0	5
						6	8.7	29.1	2
						7	0.0	46.5	0
10	10	10	8.6	7.8	10–10	9	36.9	3.1	0
						10	8.6	7.8	10
						11	0.0	37.0	0
	15	15	16.9	17.1	10–10	8	66.7	3.0	0
						9	40.3	6.0	3
						10	23.0	22.4	4
						11	5.0	41.2	3
						12	0.3	77.5	0
	20	20	23.8	15.9	10–10	8	68.3	2.7	0
						9	43.6	8.7	3
						10	23.5	21.8	6
						11	11.3	46.8	1
						12	2.0	80.0	0
15	10	10	8.0	7.3	15–15	14	51.2	2.5	0
						15	8.0	7.3	10
						16	0.1	51.6	0
	15	15	16.8	11.2	15–15	13	95.0	1.0	0
						14	53.2	4.1	1
						15	14.5	13.8	9
						16	3.0	54.3	0
	20	20	23.8	17.8	15–15	13	99.5	4.0	0
						14	56.8	6.9	2
						15	24.8	23.5	7
						16	6.8	57.5	1
						17	9.5	115.0	0
20	10	10	8.8	10.1	20–20	19	65.0	3.3	0
						20	8.8	10.1	10
						21	0.0	68.0	0
	15	15	12.4	13.3	20–20	19	70.4	8.2	0
						20	12.4	13.3	10
						21	2.7	70.2	0
	20	20	15.6	29.5	20–20	19	72.6	10.1	0
						20	22.7	23.4	8
						21	8.1	75.4	2
						22	0.0	136.0	0

As mentioned above, the proposed algorithm can easily be modified to handle the case where not all primary fragments break into smaller secondary ones. As a continuation of the research reported in this paper, one may consider the analysis of not only secondary but also further products of the spontaneous RNA degradation, which are observed in biology. Taking them into account is a real challenge.
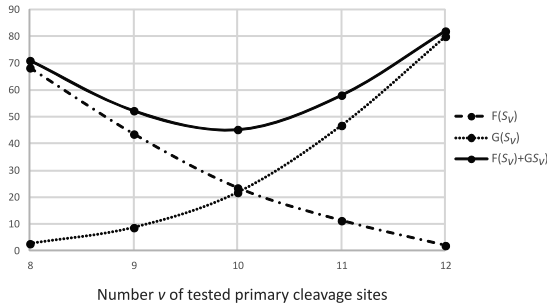
Fig. 5. Values of $F(S_v), G(S_v)$ and $F(S_v) + G(S_v)$ for instances with $p = 10$ primary cleavage sites, $Neg = 20$ false negatives, and $Pos = 20$ false positives.
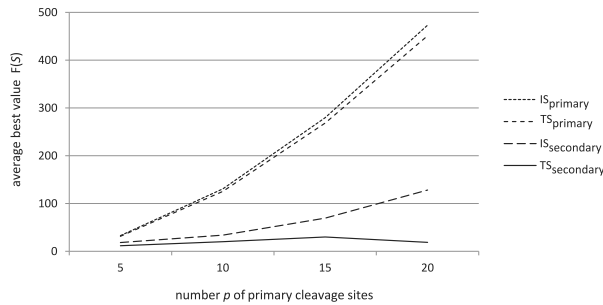

Fig. 7. Total computing time of tabu search for instances with false negatives and false positives.

Fig. 6. Average best values $F(S)$ produced by the four subroutines for instances with 20 false negatives.
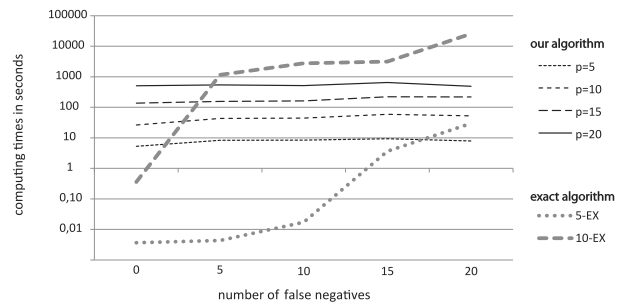
Fig. 8. Total computing time of both algorithms for instances with false negatives.

Acknowledgment

References

Adachi, H. and Yu, Y. (2014). Purification of radiolabeled RNA products using denaturing gel electrophoresis, *Current Protocols in Molecular Biology* **105**: 4.20.1–4.20.13, DOI: 10.1002/0471142727.mb0420s105.

Bibillo, A., Figlerowicz, M. and Kierzek, R. (1999). The non-enzymatic hydrolysis of oligoribonucleotides. VI: The role of biogenic polyamines, *Nucleic Acids Research* **27**(19): 3931–3937, DOI: 10.1093/nar/27.19.3931.

Bibillo, A., Figlerowicz, M., Ziomek, K. and Kierzek, R. (2000). The nonenzymatic hydrolysis of oligoribonucleotides. VII: Structural elements affecting hydrolysis, *Nucleosides Nucleotides Nucleic Acids* **19**(5–6): 977–994, DOI: 10.1080/15257770008033037.

Bilski, A. and Wojciechowski, J. (2016). Automatic parametric fault detection in complex analog systems based on a method of minimum node selection, *International Journal of Applied Mathematics and Computer Science* **26**(3): 655–668, DOI: 10.1515/amcs-2016-0045.
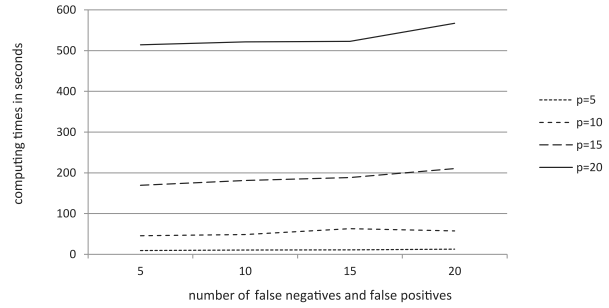
Blazewicz, J., Figlerowicz, M., Kasprzak, M., Nowacka, M. and Rybarczyk, A. (2011). RNA partial degradation problem: Motivation, complexity, algorithm, *Journal of Computational Biology* **18**(6): 821–834, DOI: 10.1089/cmb.2010.0153.

Blazewicz, J., Formanowicz, P., Guinand, F. and Kasprzak, M. (2002). A heuristic managing errors for DNA sequencing, *Bioinformatics* **18**(5): 652–660, DOI: 10.1093/bioinformatics/18.5.652.

Blazewicz, J., Formanowicz, P., Kasprzak, M., Jaroszewski, M. and Markiewicz, W. (2001). Construction of DNA restriction maps based on a simplified experiment, *Bioinformatics* **17**(5): 398–404, DOI: 10.1093/bioinformatics/17.5.398.

Blazewicz, J., Formanowicz, P., Kasprzak, M., Markiewicz, W. and Weglarz, J. (1999). DNA sequencing with positive and negative errors, *Journal of Computational Biology* **6**(1): 113–123, DOI: 10.1089/ cmb.1999.6.113.

Blazewicz, J., Glover, F. and Kasprzak, M. (2005). Evolutionary approaches to DNA sequencing with errors, *Annals of Operations Research* **138**(67): 67–78, DOI: 10.1007/s10479-005-2445-2.

Blazewicz, J. and Kasprzak, M. (2012). Complexity issues in computational biology, *Fundamenta Informaticae* **118**(4): 385–401, DOI: 10.3233/FI-2012-721.

Chanfreau, G. (2015). Two degrading decades for RNA, *RNA* **21**(4): 584–585, DOI: 10.1261/rna.050146.115.

Deutscher, M. (2003). Degradation of stable RNA in bacteria, *Journal of Biological Chemistry* **278**(46): 45041–45044, DOI: 10.1074/jbc.R300031200.

Dutkiewicz, M. and Ciesiolka, J. (2005). Structural characterization of the highly conserved 98-base sequence at the 3' end of HCV RNA genome and the complementary sequence located at the 5' end of the replicative viral strand, *Nucleic Acids Research* **33**(2): 693–703, DOI: 10.1093/nar/gki218.

Ender, C., Krek, A., Friedlander, M., Beitzinger, M., Weinmann, L., Chen, W., Pfeffer, S., Rajewsky, N. and Meister, G. (2008). A human snoRNA with microRNA-like functions, *Molecular Cell* **32**(4): 519–528, DOI: 10.1016/j.molcel.2008.10.017.

Garey, M. and Johnson, D. (1979). *Computers and Intractability. A Guide to the Theory of NP-Completeness*, W.H. Freeman & Co., New York, NY.

Glover, F. (1990). Tabu search: A tutorial, *Interfaces* **20**: 74–94, DOI: 10.1287/inte.20.4.74.

Glover, F., Kelly, J. and Laguna, M. (1995). Genetic algorithms and tabu search: Hybrids for optimization, *Computers and Operations Research* **22**(1): 111–134, DOI: 10.1016/0305-0548(93)E0023-M.

Glover, F. and Laguna, M. (1997). *Tabu Search*, Kluwer Academic Publishers, Norwell, MA.

Haussecker, D., Huang, Y., Lau, A., Parameswaran, P., Fire, A. and Kay, M. (2010). Human tRNA-derived small RNAs in the global regulation of RNA silencing, *RNA* **16**(4): 673–695, DOI: 10.1261/rna.2000810.

Jackowiak, P., Nowacka, M., Strozycki, P. and Figlerowicz, M. (2011). RNA degradome—ITS biogenesis and functions, *Nucleic Acids Research* **39**(17): 7361–7370, DOI: 10.1093/nar/gkr450.

Jankowiak, K., Lesicka, J., Pacak, A., Rybarczyk, A. and Szweykowska-Kulinska, Z. (2004). A comparison of group II introns of plastid tRNALys UUU genes encoding maturase protein, *Cellular and Molecular Biology Letters* **9**(2): 239–251.

Jankowiak, K., Rybarczyk, A., Wyatt, R., Odrzykoski, I., Pacak, A. and Szweykowska-Kulinska, Z. (2005). Organellar inheritance in the allopolyploid moss rhizomnium pseudopunctatum, *Taxon* **54**(2): 383–388, DOI: 10.2307/25065367.

Kierzek, R. (1992). Hydrolysis of oligoribonucleotides: influence of sequence and length, *Nucleic Acids Research* **20**(19): 5073–5077, DOI: 10.1093/nar/20.19.5073.

Kierzek, R. (2001). Nonenzymatic cleavage of oligoribonucleotides, *Methods in Enzymology* **341**: 657–675.

Kuppusamy, L. and Mahendran, A. (2016). Modelling DNA and RNA secondary structures using matrix insertion–deletion systems, *International Journal of Applied Mathematics and Computer Science* **26**(1): 245–258, DOI: 10.1515/amcs-2016-0017.

Nowacka, M., Jackowiak, P., Rybarczyk, A., Magacz, T., Strozycki, P., Barciszewski, J. and Figlerowicz, M. (2012). 2D-PAGE as an effective method of RNA degradome analysis, *Molecular Biology Reports* **39**(1): 139–146, DOI: 10.1007/s11033-011-0718-1.

Podkowinski, J., Zmienko, A., Florek, B., Wojciechowski, P., Rybarczyk, A., Wrzesinski, J., Ciesiolka, J., Blazewicz, J., Kondorosi, A., Crespi, M. and Legocki, A. (2009). Translational and structural analysis of the shortest legume ENOD40 gene in Lupinus luteus, *Acta Biochimica Polonica* **56**(1): 89–102.

Rybarczyk, A., Jackowiak, P., Figlerowicz, M. and Blazewicz, J. (2016). Computational prediction of nonenzymatic RNA degradation patterns, *Acta Biochimica Polonica* **63**(4): 745–751, DOI: 10.18388/abp.2016_1331.

Rybarczyk, A., Szostak, N., Antczak, M., Zok, T., Popenda, M., Adamiak, R., Blazewicz, J. and Szachniuk, M. (2015). New in silico approach to assessing RNA secondary structures with non-canonical base pairs, *BMC Bioinformatics* **16**: 276, DOI: 10.1186/s12859-015-0718-6.

Szostak, N., Royo, F., Rybarczyk, A., Szachniuk, M., Blazewicz, J., del Sol, A. and Falcon-Perez, J. (2014). Sorting signal targeting mRNA into hepatic extracellular vesicles, *RNA Biology* **11**(7): 836–844, DOI: 10.4161/rna.29305.

Yao, B., Hu, P., Zhang, M. and Jin, M. (2014). A support vector machine with the tabu search algorithm for freeway incident detection, *International Journal of Applied Mathematics and Computer Science* **24**(2): 397–404, DOI: 10.2478/amcs-2014-0030.

Zhang, S., Sun, L. and Kragler, F. (2009). The phloem-delivered RNA pool contains small noncoding RNAs and interferes with translation, *Plant Physiology* **150**(1): 378–387, DOI: 10.1104/pp.108.134767.

Zok, T., Antczak, M., Riedel, M., Nebel, D., Villmann, T., Lukasiak, P., Blazewicz, J. and Szachniuk, M. (2015). Building the library of RNA 3D nucleotide conformations using the clustering approach, *International Journal of Applied Mathematics and Computer Science* **25**(3): 689–700, DOI: 10.1515/amcs-2015-0050.

Agnieszka Rybarczyk received her MSc degree in molecular biology from Adam Mickiewicz University (UAM), Poland, and her MSc and PhD (2010) degrees in computing science from the Poznan University of Technology (PUT), Poland. She is currently working as an assistant professor in the Institute of Computing Science, PUT, and in the Institute of Bioorganic Chemistry, Polish Academy of Sciences. She has attended many scientific and technical courses and has participated in many national and international grants as a co-investigator. Moreover, she has published many research papers in various international journals. Her research interests are focused on bioinformatics (computational biology), systems biology, algorithms design and computational complexity.

Alain Hertz is the holder of a diploma in mathematical engineering. He obtained a PhD in operations research at École Polytechnique Fédérale de Lausanne. Since 2001, he has been a professor at the Department of Mathematics and Industrial Engineering at École Polytechnique in Montreal. He is also a member of the multidisciplinary GERAD research group that includes nearly sixty researchers and experts in operations research and discrete mathematics. He is the author of about more than 200 scientific publications His main research domains are combinatorial optimization, graph theory, algorithmics, and the development of decision aid systems for scheduling and distribution problems.

Marta Kasprzak received her PhD degree in computer science at the Poznan University of Technology, Poland, in 1999. In 2015, she obtained the scientific title of a professor. She focuses her scientific research on bioinformatics/computational biology, with the emphasis on theoretical analysis of problems. She has published 61 articles, 37 of them indexed by the Web of Science, and has worked in 15 national and international research projects. She is a founding member of the Polish Bioinformatics Society.

Jacek Blazewicz is a professor at the Poznan University of Technology. His research interests include algorithm design, computational complexity, scheduling, combinatorial optimization, bioinformatics, e-commerce. Blazewicz has a PhD in computer science from the Poznan University of Technology. His publication record includes over 340 papers in many outstanding journals. He is also the author and co-author of over ten monographs. He is an IEEE Fellow.