© The British Computer Society 2014. All rights reserved.
For Permissions, please email: journals.permissions@oup.com
Advance Access publication on 5 November 2014
doi:10.1093/comjnl/bxu116

Covering Pairs in Directed Acyclic Graphs[†]

Niko Beerenwinkel[1], Stefano Beretta[2], Paola Bonizzoni[3],
Riccardo Dondi[4] and Yuri Pirola[3,*]

[1]*Department of Biosystems Science and Engineering, ETH Zurich, Basel, Switzerland*
[2]*Istituto di Tecnologie Biomediche, Consiglio Nazionale delle Ricerche, Segrate, Italy*
[3]*Dip. di Informatica Sistemistica e Comunicazione, Univ. degli Studi di Milano-Bicocca, Milan, Italy*
[4]*Dip. di Scienze Umane e Sociali, Univ. degli Studi di Bergamo, Bergamo, Italy*
**Corresponding author: pirola@disco.unimib.it*

The Minimum Path Cover (MinPC) problem on directed acyclic graphs (DAGs) is a classical problem in graph theory that provides a clear and simple mathematical formulation for several applications in computational biology. In this paper, we study the computational complexity of three constrained variants of MinPC motivated by the recent introduction of Next-Generation Sequencing technologies. The first variant (MinRPC), given a DAG and a set of pairs of vertices, asks for a minimum-cardinality set of (not necessarily disjoint) paths such that both vertices of each pair belong to the same path. For this problem, we establish a sharp tractability borderline depending on the 'overlapping degree' of the instance, a natural parameter in some applications of the problem. The second variant we consider (MinPCRP), given a DAG and a set of pairs of vertices, asks for a minimum-cardinality set of (not necessarily disjoint) paths 'covering' all the vertices of the graph and such that both vertices of each pair belong to the same path. For this problem, we show that, while it is NP-hard to compute if there exists a solution consisting of at most three paths, it is possible to decide in polynomial time whether a solution consisting of at most two paths exists. The third variant (MaxRPSP), given a DAG and a set of pairs of vertices, asks for a single path containing the maximum number of the given pairs of vertices. We show that MaxRPSP is W[1]-hard when parameterized by the number of covered pairs and we give a fixed-parameter algorithm when the parameter is the maximum overlapping degree.

Keywords: minimum path cover; sequence reconstruction; paired-end reads; computational complexity

Received 19 March 2014; revised 29 August 2014
Handling editor: Iain Stewart

1. INTRODUCTION

The *Minimum Path Cover* (MinPC) problem is a well-known problem in graph theory. Given a *directed acyclic graph* (DAG), MinPC asks for a minimum-cardinality set Π of paths such that each vertex of G belongs to at least one path of Π. The problem can be solved in polynomial time with an algorithm based on a proof of the well-known Dilworth's theorem for partially ordered sets, which allows to relate the size of a minimum path cover to that of a maximum matching in a bipartite graph obtained from the input DAG [2].

The MinPC problem has important applications in several fields ranging from bioinformatics [3–5] to software testing [6, 7]. In particular, in bioinformatics, mainly thanks to the advent of the *Next-Generation Sequencing* techniques, the MinPC problem is widely applied to the reconstruction of nucleotide sequences starting from a large set of their short fragments (called *short reads*) [3, 4]. More precisely, each fragment is represented by a single vertex and two vertices are connected if the alignments of the corresponding reads on the genomic sequence overlap. In [4], the paths on such a graph represent putative RNA transcripts and a minimum-cardinality set of paths 'covering' all the vertices represents a set of protein isoforms which likely originated the observed reads. Instead, in [3], the paths of the graph represent the genomes of putative viral quasispecies and a minimum-cardinality set of paths covering the whole graph represents the likely structure

[†] A preliminary version of this paper appeared in [1].

of a viral population. In these applications, the aim is that of reconstructing ‘complete’ (hence, as long as possible) sequences that possibly share some substrings. Hence, it is often assumed that paths start from a source vertex, end at a sink vertex and possibly share some vertices. In the rest of the paper, we will implicitly make this assumption.

Recently, the availability of new kinds of data have motivated the definition of new constrained variants of graph problems in different fields, such as, for example, in the context of social network analysis [8, 9].

Reconstructing sequences via Minimum Path Cover is particularly effective on relatively small regions of the sequences to reconstruct, since on long regions there is not enough information for establishing if two distant fragments were originated from the same sequence. In order to solve (or lessen the impact of) this issue, a particular kind of reads, called *paired-end reads*, could considerably help. In fact, paired-end reads are pairs of reads obtained from the same sequence at a fixed distance, typically larger than the fragments length. Hence, in a ‘valid’ path cover, for each paired-end read, there must exist at least a path which contains the two vertices representing the associated reads.

However, the most widely used methods for sequence reconstruction in bioinformatics [4, 10] do not take fully advantage of the constraints imposed by paired-end reads during the reconstruction process and they only use them to validate (or discard) the reconstructed sequences.

Recent approaches are trying to incorporate the new additional constraints carried by paired-end or longer sequences into the problem formulation, in order to exploit it for the reconstruction of the sequences [5, 11, 12]. For example, in CLASS [11], complete sequences are first exhaustively enumerated and then a smallest set of them satisfying all the constraints derived from paired-end or long reads is selected using a greedy set-cover approximation algorithm. Clearly, this method is both computationally intensive (since it requires a nearly exhaustive enumeration of the transcripts) and also approximate (since it employs the approximation algorithm for set cover). As a consequence, both its applicability to real large datasets and the accuracy of its results are limited. BRANCH [5] overcomes these limits by considering only constraints derived from long reads (or previously found transcripts), which are modeled as subpath constraints (i.e. subpaths that must be contained in some path of the solution). For this scenario, the authors present a polynomial-time algorithm that reduces the constrained path cover problem to MinPC by ‘contracting’ the subpath constraints in single vertices. However, by not considering paired-end reads, the accuracy of BRANCH degrades with the length of the reconstructed sequences (unless there are many long subpath constraints, obviously). Recently, Rizzi *et al.* [12] extended the reduction used by BRANCH in order to correctly handle also the cases where a constraint is a subpath of another constraint. Moreover, they also begin a preliminary study of the computational complexity of the MinPC problem when constraints derived from paired-end reads are introduced.

In this paper, we present a systematic study of the computational complexity of some variants of the MinPC problem where constraints deriving from paired-end reads are introduced. Similar constrained variants have been also studied in the past by Ntafos and Hakimi in the context of software testing [7]. More precisely, in that context, each procedure to be tested is modeled by a graph where vertices correspond to single instructions and two vertices are connected if the corresponding instructions are executed sequentially. The test of the procedure should check each instruction at least once, hence a minimum path cover of the graph represents a minimum set of execution flows that allows one to test all the instructions. Moreover, since there are pairs of vertices that a feasible solution must include, Ntafos and Hakimi proposed and formalized the concept of *required pairs*. In particular, one of the problems they introduced is the *Minimum Required Pairs Cover* (MinRPC) problem where, given a DAG and a set of required pairs, the goal is to compute a minimum set of paths *covering* all the required pairs, i.e. a minimum set of paths such that, for each required pair, at least a path contains both vertices of the pair.

It is easy to see that the concept of required pairs introduced by Ntafos and Hakimi correctly models the constraints deriving from paired-end reads in the sequence reconstruction problems we presented before. However, note that MinRPC asks for a solution that covers only the required pairs, while, in the sequence reconstruction problems, we are interested in solutions that also cover all the vertices. For this reason, we consider a variant of the MinPC problem, called *Minimum Path Cover with Required Pairs* (MinPCRP), that, given a DAG and a set of required pairs, asks for a minimum set of paths covering all the vertices and all the required pairs. Clearly, MinPCRP is closely related to MinRPC. In fact, as we show in Section 2, the same reduction used in [7] to prove the NP-hardness of MinRPC can be applied to our problem, leading to its intractability.

In this paper, we continue the analysis of [7] by studying the complexity of path covering problems with required pairs. More precisely, we study how the complexity of these problems is influenced by two parameters relevant for the sequence reconstruction applications in bioinformatics: (i) the minimum number of paths covering all the vertices and all the required pairs and (ii) the maximum *overlapping degree* (defined later). In the bioinformatics applications we discussed, the first parameter—the number of covering paths—is often small, thus an algorithm exponential in the size of the solution could be of interest. The second parameter we consider in this paper, the maximum overlapping degree, can be informally defined as follows. Two required pairs overlap when there exists a path that connects the vertices of the pairs, and the path cannot be split in two disjoint subpaths that separately connect the vertices of the two pairs. Then, the overlapping degree of a required pair is the number of required pairs that overlap with it. In the sequence reconstruction applications, as the distance between two paired-end reads is fixed, the maximum overlapping degree is small compared with the number of vertices, hence it is a natural parameter for investigating the computational complexity of the problem.

First, we investigate how the computational complexity of MinRPC is influenced by the maximum overlapping degree. We show that MinRPC is APX-hard (hence also NP-hard) when the maximum overlapping degree is bounded by 1, while it is polynomial time solvable when the maximum overlapping degree is 0. Note that MinPCRP is already NP-hard if the maximum overlapping degree is 0. In fact, this can be easily obtained by modifying the reduction presented in [7] to hold also for restricted instances of MinPCRP with no overlapping required pairs.

Then, we investigate how the computational complexity of MinPCRP is influenced by the number of paths that compose a solution. We prove that it is NP-complete to decide if there exists a solution of MinPCRP consisting of at most three paths (via a reduction from the 3-Coloring problem). We complement this result by giving a polynomial-time algorithm for computing a solution with at most two paths, thus establishing a sharp tractability borderline for MinPCRP when parameterized by the size of the solution. These results significantly improve the hardness result that Ntafos and Hakimi [7] presented for MinRPC (and that holds also for MinPCRP), where the solution contains a number of paths which is polynomial in the size of the input. Some of these results have been independently obtained by Rizzi *et al.* [12].

A natural heuristic approach for solving MinPCRP is the one which computes a solution by iteratively adding a path that covers a maximum set of required pairs not yet covered by a path of the solution. This approach leads to a natural combinatorial problem, the *Maximum Required Pairs with Single Path* (MaxRPSP) problem, that, given a DAG and a set of required pairs, asks for a path that covers the maximum number of required pairs. We investigate the complexity of MaxRPSP and we show that it is not only NP-hard, but also W[1]-hard when the parameter is the number of covered required pairs. This result shows that it is unlikely that the problem is fixed-parameter tractable when parameterized by the number of required pairs covered by a single path. We refer the reader to [13, 14] for an in-depth presentation of the theory of fixed-parameter complexity. We consider also the MaxRPSP problem parameterized by the maximum overlapping degree and, differently from MinPCRP, we give a fixed-parameter algorithm for this case. This positive result shows a gap between the complexity of MaxRPSP and the complexity of MinPCRP when parameterized by the maximum overlapping degree.

The rest of the paper is organized as follows. First, in Section 2 we give some preliminary notions and we introduce the formal definitions of the problems we are interested in. In Section 3, we investigate how the computational complexity of MinRPC is influenced by the maximum 'overlapping degree', while in Section 4, we investigate the computational complexity of MinPCRP when the solution consists of a constant number of paths, and in Section 5, we investigate the computational complexity of MaxRPSP. We conclude in Section 6 by presenting some final remarks and some open problems.

2. PRELIMINARIES

In this section, we introduce the basic notions used in the rest of the paper and we formally define the three combinatorial problems we are interested in.

We denote an *undirected graph* as $G = (V, E)$, where V is the set of vertices and E is the set of (undirected) edges, and a *directed graph* (or *digraph*) as $D = (N, A)$, where N is the set of vertices and A is the set of (directed) arcs. We denote an edge of $G = (V, E)$ as $\{v, u\} \in E$, where $v, u \in V$. Moreover, we denote an arc of $D = (N, A)$ as $(v, u) \in A$, where $v, u \in N$.

Given a directed graph $D = (N, A)$, a *path* π from vertex v to vertex u, denoted as vu-path, is a sequence of vertices $\langle v_1, \ldots, v_n \rangle$ such that $(v_i, v_{i+1}) \in A$, $v = v_1$ and $u = v_n$. We say that a vertex v *belongs to* a path $\pi = \langle v_1, \ldots, v_n \rangle$, denoted as $v \in \pi$, if $v = v_i$ for some $1 \leq i \leq n$. Given a path $\pi = \langle v_1, \ldots, v_n \rangle$, we say that a path $\pi' = \langle v_i, v_{i+1}, \ldots, v_{j-1}, v_j \rangle$, with $1 \leq i \leq j \leq n$, is a subpath of π. Given a set $N' \subseteq N$ of vertices, a path π *covers* N' if every vertex of N' belongs to π.

In the paper, we consider a set R of pairs of vertices in N. We denote each pair as $[v_i, v_j]$, to avoid ambiguity with the notations of edges and arcs.

Now, we are able to define the combinatorial problems we are interested in.

Problem 1. *Minimum Required Pairs Cover* (MinRPC)
Input: a DAG $D = (N, A)$, a source $s \in N$, a sink $t \in N$ and a set $R = \{[v_x, v_y] \mid v_x, v_y \in N, v_x \neq v_y\}$ of required pairs.
Output: a minimum cardinality set $\Pi = \{\pi_1, \ldots, \pi_n\}$ of directed st-paths such that every required pair $[v_x, v_y] \in R$ belongs to at least one st-path $\pi_i \in \Pi$, i.e. v_x, v_y belongs to π_i.

Problem 2. *Minimum Path Cover with Required Pairs* (MinPCRP)
Input: a DAG $D = (N, A)$, a source $s \in N$, a sink $t \in N$ and a set $R = \{[v_x, v_y] \mid v_x, v_y \in N, v_x \neq v_y\}$ of required pairs.
Output: a minimum cardinality set $\Pi = \{\pi_1, \ldots, \pi_n\}$ of directed st-paths such that every vertex $v \in N$ belongs to at least one st-path $\pi_i \in \Pi$ and every required pair $[v_x, v_y] \in R$ belongs to at least one st-path $\pi_i \in \Pi$, i.e. v_x, v_y belongs to π_i.

Problem 3. *Maximum Required Pairs with Single Path* (MaxRPSP)
Input: a DAG $D = (N, A)$, a source $s \in N$, a sink $t \in N$ and a set $R = \{[v_x, v_y] \mid v_x, v_y \in N, v_x \neq v_y\}$ of required pairs.
Output: an st-path π that covers a set $R' = \{[v_x, v_y] \mid v_x, v_y \in \pi\} \subseteq R$ of maximum cardinality.

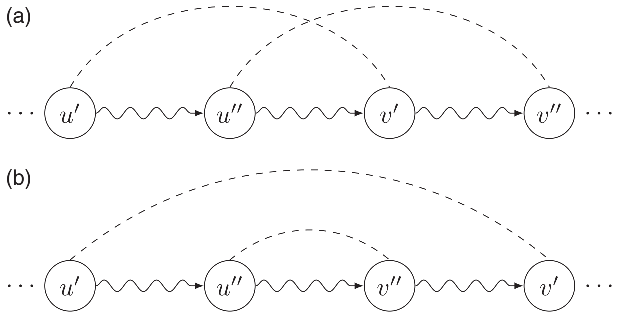

FIGURE 1. Examples of two overlapping required pairs $[u', v']$ and $[u'', v'']$. In (**a**) the required pairs are *alternated*, while in (**b**) $[u'', v'']$ is *nested* in $[u', v']$.

For simplicity, in the rest of the paper, we assume that the source s and the sink t of the DAG are given (otherwise, it is easy to find them). Moreover, we also assume that all the paths (unless otherwise specified) start from s and end at t. Two required pairs $[u', v']$ and $[u'', v'']$ in R *overlap* if there exists a path π in D such that the four vertices appear in π in one of the following orders (assuming that the vertex u' appears before u'' in π), where v' and u'' are two distinct vertices of G (see Fig. 1):

(i) $\langle u', u'', v', v''\rangle$ (the two required pairs are said to be *alternated*);
(ii) $\langle u', u'', v'', v'\rangle$ (the required pair $[u'', v'']$ is said to be *nested* in $[u', v']$).

Note that, according to this definition, the required pairs $[x, y]$ and $[y, z]$ do not overlap.

Moreover, we define the *overlapping degree* of a required pair $[u', v'] \in R$ as the number of required pairs in R that overlap with $[u', v']$.

Given a DAG D and a set R of required pairs, the *compatibility relation* $\mathcal{C} \subseteq R^2$ on the set R is defined as follows. A pair of required pairs $([u', v'], [u'', v''])$ belongs to $\mathcal{C}$ if there exists a path π that covers both $[u', v']$ and $[u'', v'']$ and v' appears strictly before v'' in π or $[u', v']$ is nested in $[u'', v'']$. Note that if there exists a path that covers two required pairs r', r'', then either (r', r'') or (r'', r') is in $\mathcal{C}$. This definition of compatibility among required pairs and the one proposed by [12] are closely related. However, note that, according to our definition, the compatibility relation is irreflexive (a required pair is not compatible with itself), while according to theirs is symmetric. Given a subset R' of R, we denote the restriction of $\mathcal{C}$ to the elements in R' as $\mathcal{C}(R')$. We say that a subset $C \subseteq R$ of required pairs is a *chain* if $\mathcal{C}(C)$ is a strict total order (i.e. for each $r', r'' \in C$ with $r' \neq r''$, at least one of (r', r'') and (r'', r') belongs to $\mathcal{C}(C)$). Finally, we recall that a binary relation is a *strict partial order* if it is *irreflexive* and *transitive*. Please note that, in general, the compatibility relation $\mathcal{C}$ is not transitive (hence, it is not a strict partial order).

The compatibility relation can be also considered as the arc set of a digraph (that we call the *compatibility digraph*) having as vertex set R. Such a digraph is clearly acyclic. We call *compatibility graph* the undirected graph obtained by discarding the edge orientation of the compatibility digraph. In the following, for simplicity, we will interchangeably consider the compatibility relation as a binary relation or as the associated digraph.

Hardness of MinRPC and MinPCRP. As mentioned in Section 1, one of the problems we are interested in, namely MinRPC, was initially defined in the context of program testing [7] and its NP-hardness was proved. From this result, we can immediately derive the NP-hardness of MinPCRP. Indeed, MinRPC can be easily reduced to MinPCRP by ensuring that each vertex of the graph D (input of MinRPC) belongs to at least one required pair. Otherwise, if this condition does not hold for some vertex v, we can modify the graph D by contracting v (that is, removing v and adding an edge (u, z) to A, for each $u, z \in N$ such that (u, v), $(v, z) \in A$). This implies that, since in the resulting instance of MinRPC all the vertices belong to some required pair, a feasible solution of that problem covers every vertex of the graph. Then, a solution of MinRPC is also a solution of MinPCRP, which implies that MinPCRP is NP-hard.

Both MinRPC and MinPCRP on *directed* graphs (not necessarily acyclic) are as hard as MinRPC and MinPCRP, respectively, on DAGs. In fact, since each strongly connected component can be covered with a single path, we can replace each of them with a single vertex, obtaining a DAG and without changing the size of the solution. Finally, MinRPC and MinPCRP on general graphs (with the additional requirement that the covering paths are simple) are as hard as the Hamiltonian path problem, which is NP-complete [15, probl. GT39].

3. A SHARP TRACTABILITY BORDERLINE FOR MINRPC

In this section, we consider the tractability of MinRPC when the maximum overlapping degree of the instance is bounded. We show in Section 3.1 that MinRPC is APX-hard (hence also NP-hard) when the maximum overlapping degree is bounded by 1, while in Section 3.2 we show that MinRPC admits a polynomial time algorithm when the maximum overlapping degree is 0.

3.1. APX-hardness of MinRPC when the maximum overlapping degree is bounded by 1

We will show the APX-hardness of MinRPC when the maximum overlapping degree is bounded by 1 by giving an L-reduction (for details on L-reductions we refer the reader to [16]) from the *Minimum Vertex Cover* problem on Cubic graphs (MinVCC). We recall that a graph is cubic when each vertex is adjacent to exactly three other vertices. Given an undirected cubic graph $G = (V, E)$, the MinVCC problem asks for a minimum cardinality set $V' \subseteq V$ such that for each edge $\{v_i, v_j\} \in E$, $v_i \in V'$ or $v_j \in V'$.

We start by showing how to transform (in polynomial time) an instance $G = (V, E)$ of MinVCC into an instance $\langle D = (N, A), R\rangle$ of MinRPC such that its maximum overlapping degree is bounded by 1.

First, we define the vertex set N:

$$\begin{aligned} N = \{&v_{i,j,q}, v'_{i,j,q} \mid \{v_i, v_j\} \in E, 1 \leq q \leq 4\} \\ &\cup \{v^{i,j}_{i,q} \mid \{v_i, v_j\} \in E, 1 \leq q \leq 2\} \cup \{s, t\} \end{aligned}$$

We define the arc set A by means of a set of paths (see Fig. 2). Note that the paths may share some arcs.

For each edge $\{v_i, v_j\} \in E$, we define two (disjoint) paths $\pi_{i,j}$ and $\pi'_{i,j}$ connecting the sequence of vertices $\langle s, v_{i,j,1}, \ldots, v_{i,j,4}, t\rangle$ and $\langle s, v'_{i,j,1}, \ldots, v'_{i,j,4}, t\rangle$, respectively.

For each vertex $v_i \in V$, we define four paths in D. Let $\{v_i, v_j\}, \{v_i, v_h\}, \{v_i, v_k\} \in E$, with $j < h < k$, be the three edges of G incident to v_i. We define a path:

$$\pi_i = \langle s, v^{i,j}_{i,1}, v^{i,j}_{i,2}, v^{i,h}_{i,1}, v^{i,h}_{i,2}, v^{i,k}_{i,1}, v^{i,k}_{i,2}, t\rangle$$

Moreover, we define three paths, called *additional paths*:

(i) $\pi^i_{i,j} = \langle s, v'_{i,j,1}, v^{i,j}_{i,1}, v^{i,j}_{i,2}, v_{i,j,4}, t\rangle$;
(ii) $\pi^i_{i,h} = \langle s, v'_{i,h,1}, v^{i,h}_{i,1}, v^{i,h}_{i,2}, v_{i,h,4}, t\rangle$;
(iii) $\pi^i_{i,k} = \langle s, v'_{i,k,1}, v^{i,k}_{i,1}, v^{i,k}_{i,2}, v_{i,k,4}, t\rangle$.

The paths defined above will be used later (see Lemmas 3.2 and 3.3) to construct a solution of MinRPC over instance $\langle D = (N, A), R\rangle$.

Finally, we define the set R of required pairs as follows:

$$\begin{aligned} R = \{&[v'_{i,j,1}, v_{i,j,4}], [v_{i,j,2}, v_{i,j,3}], [v'_{i,j,2}, v'_{i,j,3}] \mid \\ &\{v_i, v_j\} \in E\} \cup \{[v^{i,j}_{i,1}, v^{i,j}_{i,2}] \mid v_i \in V, \{v_i, v_j\} \in E\} \end{aligned}$$

Figure 2 represents an extract of a directed subgraph of D associated with an undirected subgraph (constructed from a vertex $v_i \in V$) of G.

It is easy to see that, given a graph G, the corresponding instance $\langle D, R\rangle$ can be constructed in polynomial time. Next, we show that $\langle D, R\rangle$ has maximum overlapping degree 1.

Lemma 3.1. *Instance $\langle D, R\rangle$ has a maximum overlapping degree* 1.

Proof. Note that the only overlapping required pairs in R are, fixed an edge $\{v_i, v_j\} \in E$, $[v^{i,j}_{i,1}, v^{i,j}_{i,2}]$ and $[v'_{i,j,1}, v_{i,j,4}]$. Hence, the maximum overlapping degree in D is 1. □

Now, we are able to prove the two main results of the reduction.

Lemma 3.2. *Given an undirected cubic graph $G = (V, E)$ and a vertex cover $V' \subseteq V$ of G, we can compute in polynomial time a feasible solution Π of the associated instance $\langle D = (N, A), R\rangle$ of MinRPC such that $|\Pi| = 3|E| + |V'|$.*

Proof. Consider a vertex cover V' for $G = (V, E)$. In the following, we compute (in polynomial time) a set Π of $3|E| + |V'|$ paths on D that covers all the required pairs in R. Set Π is constructed as follows.

(1) For each vertex $v_i \notin V'$, add to Π the three additional paths $\pi^i_{i,j}$, $\pi^i_{i,h}$ and $\pi^i_{i,k}$, where v_j, v_h, v_k are the three vertices adjacent to v_i.
(2) For each vertex $v_i \in V'$, add to Π the path π_i.
(3) For each edge $\{v_i, v_j\} \in E$, add to Π the two paths $\pi_{i,j}$ and $\pi'_{i,j}$.
(4) For each edge $\{v_i, v_j\} \in E$ such that $v_i, v_j \in V'$ and $i < j$, add to Π the path $\pi^i_{i,j}$.

It is easy to see that Π covers each required pair in R. Indeed, each required pair $[v_{i,j,2}, v_{i,j,3}]$, $[v'_{i,j,2}, v'_{i,j,3}]$ is covered in step (3). Each pair $[v^{i,j}_{i,1}, v^{i,j}_{i,2}]$ is covered in step (2) if $v_i \in V'$ or in step (1) if $v_i \notin V'$. Each required pair $[v'_{i,j,1}, v_{i,j,4}] \in R$ is covered by a path added in step (1) if $v_i \notin V'$ or $v_j \notin V'$, while it is covered by a path added in step (4) if $v_i \in V'$ and $v_j \in V'$. Note that, since V' is a vertex cover, at least one of v_i, v_j belongs to V'.

For each edge $\{v_i, v_j\} \in E$, steps (1) and (4) add exactly one path containing the vertices $v'_{i,j,1}$ and $v_{i,j,4}$. Hence, they add $|E|$ paths to Π. Step (2) adds $|V'|$ paths, while step (3) adds $2|E|$ paths. As a consequence, we have that Π is a feasible solution of MinRPC and that $|\Pi| = 3|E| + |V'|$. □

Lemma 3.3. *Let $G = (V, E)$ be an undirected cubic graph and let $\langle D = (N, A), R\rangle$ be the associated instance of MinRPC. Then, given a set of paths Π of $D = (N, A)$ which is a solution of MinRPC where $|\Pi| = 3|E| + d$, we can compute in polynomial time a vertex cover $V' \subseteq V$ for G such that $|V'| \leq d$.*

Proof. Note that by construction some required pairs in R, namely $[v_{i,j,2}, v_{i,j,3}]$, $[v'_{i,j,2}, v'_{i,j,3}]$ can only be covered by the paths $\pi_{i,j}$ and $\pi'_{i,j}$, respectively. *A fortiori*, all these paths must belong to Π. Moreover, note that by construction each required pair $[v'_{i,j,1}, v_{i,j,4}]$ can be covered by at most two paths, namely $\pi^i_{i,j}$ and $\pi^j_{i,j}$. Hence, at least one of $\pi^i_{i,j}$ and $\pi^j_{i,j}$ must be in Π. Now, we show that we can restrict ourselves to solutions where exactly one of $\pi^i_{i,j}$ and $\pi^j_{i,j}$ is in Π. If both $\pi^i_{i,j}$ and $\pi^j_{i,j}$ are in Π, then we can compute in polynomial time a

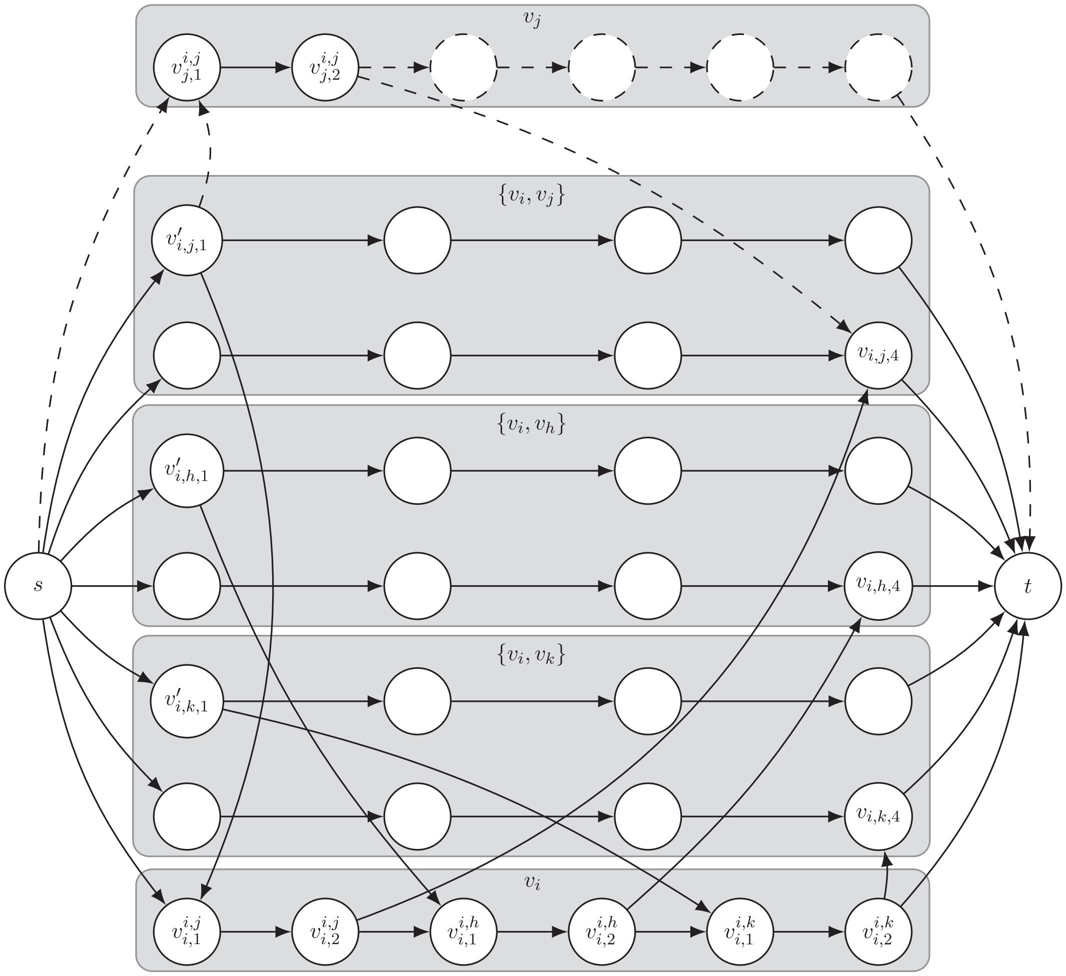

FIGURE 2. Example of the directed (acyclic) subgraph of $D = (N, A)$ associated with a subgraph of a cubic graph $G = (V, E)$ w.r.t. vertex $v_i \in V$. Gray boxes highlight the pairs of paths representing the arcs incident to v_i and the paths representing the two vertices v_i (bottom) and v_j (top). The required pairs (not shown) are $[v'_{i,j,1}, v_{i,j,4}]$, $[v_{i,j,2}, v_{i,j,3}]$, $[v'_{i,j,2}, v'_{i,j,3}]$, $[v'_{i,h,1}, v_{i,h,4}]$, $[v_{i,h,2}, v_{i,h,3}]$, $[v'_{i,h,2}, v'_{i,h,3}]$, $[v'_{i,k,1}, v_{i,k,4}]$, $[v_{i,k,2}, v_{i,k,3}]$, $[v'_{i,k,2}, v'_{i,k,3}]$, $[v^{i,j}_{i,1}, v^{i,j}_{i,2}]$ and $[v^{i,j}_{j,1}, v^{i,j}_{j,2}]$.

solution Π' such that at most one of $\pi^i_{i,j}, \pi^j_{i,j}$ belongs to Π' and such that $|\Pi'| \leq |\Pi|$ as follows. Set Π' is computed by replacing one of $\pi^i_{i,j}, \pi^j_{i,j}$, respectively, with the path π_i, π_j, respectively. We assume w.l.o.g. that $\pi^i_{i,j}$ is replaced with π_i. Clearly, $|\Pi'| \leq |\Pi|$. Moreover, Π covers each required pair in R, hence the same property holds for Π', since the required pair $[v'_{i,j,1}, v_{i,j,4}]$ is covered by the path $\pi^j_{i,j}$ and the required pair $[v^{i,j}_{i,1}, v^{i,j}_{i,2}]$ is covered by π_i.

Hence, in what follows, we assume that Π, for each $\{v_i, v_j\} \in E$, contains exactly one of $\pi^i_{i,j}, \pi^j_{i,j}$. Note that the set Π contains the $2|E|$ paths $\pi_{i,j}, \pi'_{i,j}$, $|E|$ paths $\pi^j_{i,j}$, and d paths π_i.

Define the vertex cover $V' \subseteq V$ as $\{v_i \mid \pi_i \in \Pi\}$. By construction $|V'| \leq d$. We claim that V' is a vertex cover of G. Suppose, on the contrary, that V' is not a vertex cover of G. Then, there exists an edge $\{v_i, v_j\} \in E$ such that neither v_i nor v_j are in V'. It follows that neither π_i nor π_j are in Π. By hypothesis, the set Π covers the required pair $[v^{i,j}_{i,1}, v^{i,j}_{i,2}]$, hence Π must contain both the paths $\pi^i_{i,j}$ and $\pi^j_{i,j}$ which are the only paths different from π_i and π_j that cover the pair $[v^{i,j}_{i,1}, v^{i,j}_{i,2}]$. Since we assumed that Π does not contain both $\pi^i_{i,j}$ and $\pi^j_{i,j}$, it follows that one of the required pair $[v^{i,j}_{i,1}, v^{i,j}_{i,2}]$, $[v^{i,j}_{j,1}, v^{i,j}_{j,2}]$ is not covered by Π. This fact contradicts the hypothesis that Π is a solution of MinRPC, hence V' is a vertex cover of G and the lemma holds. □

Theorem 3.1. *MinRPC is APX-hard even when the input instance has maximum overlapping degree bounded by* 1.

Proof. First, note that by Lemma 3.1 the maximum overlapping degree of $\langle D, R\rangle$ is 1. Since in a cubic graph $|E| = \frac{3}{2}|V|$ and the cardinality of a vertex cover V' is at least $\frac{1}{2}|V|$, it follows by Lemmas 3.2 and 3.3 that we have designed an L-reduction. Since MinVCC is APX-hard [17], it follows that MinRPC is APX-hard even when the input instance has maximum overlapping degree bounded by 1. □

3.2. A polynomial time algorithm for MinRPC without overlapping pairs

We will show that MinRPC can be solved in polynomial time when the instance does not contain overlapping required pairs (i.e. when the maximum overlapping degree is 0). We obtain this result by first proving that, whenever the compatibility relation of the required pairs is a strict partial order, we can compute in polynomial time a minimum-cardinality set of paths which covers all the required pairs. Then, the result follows from the fact that the compatibility relation of a set of non-overlapping required pairs is always a strict partial order.

Let $\langle D, R\rangle$ be an instance of MinRPC and $\mathcal{C}$ be the compatibility relation on R. The basic idea on which the polynomial-time algorithm is built is that a chain C of $\mathcal{C}$ corresponds to a path π_C in D that covers all the required pairs in C, as proved in the following lemma.

Lemma 3.4. *Let $\langle D, R\rangle$ be an instance of MinRPC. Then, there exits a path π_C in D covering a subset $C \subseteq R$ of required pairs if and only if C is a chain of the compatibility relation $\mathcal{C}$ on R.*

Proof. The existence of π_C implies the existence in the compatibility digraph of an arc between each (unordered) pair of required pairs of C. The orientation of each arc is given by the order of the vertices on π_C (according to the definition of the compatibility relation), but, since the compatibility digraph is acyclic, we have that $\mathcal{C}(C)$ is a total order (and C is a chain).

Let C be a chain of $\mathcal{C}$. Since C is a chain, given two vertices v', v'' which belong to two different required pairs of C, there exists a path between v' and v'' (either from v' to v'' or from v'' to v'). Consider the nodes $\langle v_{i_1}, v_{i_2}, \ldots, v_{i_l}\rangle$ that appear in some required pair of C, sorted according to a topological order of D. Connect them to build a path π_C from s (the source of D) to t (the sink of D). Such a path exists since, by the previous observation, there exists a path between each pair of vertices $v_{i_j}, v_{i_{j+1}}$. By construction, π_C covers all the pairs in C. □

In particular, note that the previous proof gives a polynomial-time algorithm for computing a path π_C which covers a subset C of required pairs forming a chain of the relation $\mathcal{C}$. Moreover, the previous result shows that MinRPC can be optimally solved in polynomial time whenever $\mathcal{C}$ is a total order (since in that case there exists a unique chain containing all the required pairs). This result can be generalized for computing an optimal solution in polynomial time whenever $\mathcal{C}$ is a strict partial order, as shown in the following theorem.

Theorem 3.2. *Let $\langle D, R\rangle$ be an instance of MinRPC and $\mathcal{C}$ be the compatibility relation of R. If $\mathcal{C}$ is a strict partial order, then a minimum-cardinality set Π of paths of D covering all the required pairs in R can be computed in polynomial time.*

Proof. Since $\mathcal{C}$ is a strict partial order, we can compute in polynomial time a minimum-cardinality set C of k chains $\{C_1, \ldots, C_k\}$ covering the partially ordered set (poset) $\langle R, \mathcal{C}\rangle$ using the classical MinPC algorithm on DAGs [2]. By Lemma 3.4, we can then compute in polynomial time a set Π of k paths of D associated with the chains $C_1, \ldots, C_k$. By construction, these paths cover all the required pairs in R. The set Π has minimum-cardinality because, otherwise, by Lemma 3.4 there would exist another set of $k' < k$ chains covering the poset $\langle R, \mathcal{C}\rangle$, contradicting the minimum-cardinality of C. □

Since the compatibility relation of a set of pairwise non-overlapping required pairs is a strict partial order, then we have the following corollary.

Corollary 3.1. *An instance $\langle D, R\rangle$ of MinRPC with maximum overlapping degree equal to* 0 *can be solved in polynomial time.*

Proof. We claim that the compatibility relation $\mathcal{C}$ of R is a strict partial order when the maximum overlapping degree is 0. The result then follows from Theorem 3.2. By definition, the compatibility relation is irreflexive, thus we only have to show that $\mathcal{C}$ is transitive. Let $r_i = [v_i^1, v_i^2]$, $r_j = [v_j^1, v_j^2]$, $r_k = [v_k^1, v_k^2]$ be three required pairs such that $\{(r_i, r_j), (r_j, r_k)\} \subseteq \mathcal{C}$. We have to prove that (r_i, r_k) belongs to $\mathcal{C}$. Since $(r_i, r_j) \in \mathcal{C}$ and since r_i and r_j do not overlap, there exists a path $\pi_{i,j}$ connecting v_i^1 to v_j^2 and covering both r_i and r_j. Similarly, there exists a path $\pi_{j,k}$ connecting v_j^2 to v_k^2 and covering r_k. The concatenation of $\pi_{i,j}$ and $\pi_{j,k}$ is a path which covers r_i and r_k. Moreover, v_i^2 is strictly before v_k^2 on this path, thus $(r_i, r_k) \in \mathcal{C}$. □

4. A SHARP TRACTABILITY BORDERLINE FOR MINPCRP

In this section, we investigate the computational complexity of MinPCRP and we give a sharp tractability borderline for k-PCRP, the restriction of MinPCRP where we ask whether there exist k paths that cover all the vertices of the graph and all the set of required pairs. First, we show that 3-PCRP is NP-complete (Section 4.1). This result implies that k-PCRP

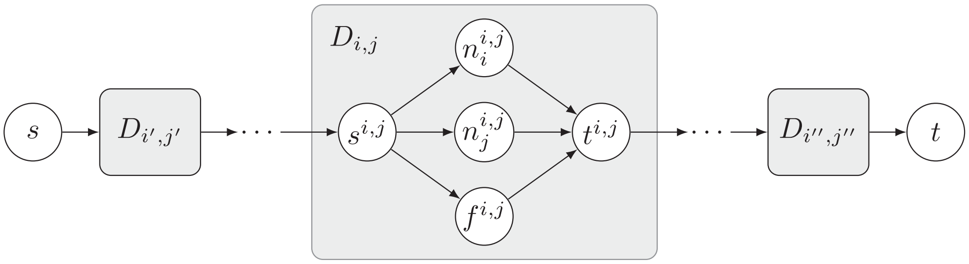

FIGURE 3. Example of graph $D = (N, A)$ associated with graph $G = (V, E)$, in which grey boxes represent subgraphs $D_{i',j'}, \ldots, D_{i'',j''}$. The central grey box shows the subgraph $D_{i,j} = (N_{i,j}, A_{i,j})$ associated with the edge $\{v_i, v_j\} \in E$.

does not belong to the class XP,[1] so it is probably hopeless to look for an algorithm having complexity $O(n^k)$, and hence for a fixed-parameter algorithm in k. The same results hold also for k-RPC, the restriction of MinRPC where we ask whether there exist k paths that cover all the required pairs. We complement this result by giving a polynomial time algorithm for 2-PCRP (Section 4.2), thus defining a sharp borderline between tractable and intractable instances of MinPCRP.

4.1. Hardness of 3-PCRP

In this section, we show that 3-PCRP is NP-complete. We prove this result via a reduction from the well-known 3-Coloring (3C) problem which, given an undirected (connected) graph $G = (V, E)$, asks for a coloring $c : V \rightarrow \{c_1, c_2, c_3\}$ of the vertices of G with exactly three colors, such that, for every $\{v_i, v_j\} \in E$, we have $c(v_i) \neq c(v_j)$.

Starting from an undirected graph $G = (V, E)$ (instance of 3C), we construct a corresponding instance $\langle D = (N, A), R\rangle$ of 3-PCRP as follows. For each edge $\{v_i, v_j\} \in E$ with $i < j$, we define a graph $D_{i,j} = (N_{i,j}, A_{i,j})$. The vertex set $N_{i,j}$ is $\{s^{i,j}, n_i^{i,j}, n_j^{i,j}, f^{i,j}, t^{i,j}\}$. The set $A_{i,j}$ of arcs connecting the vertices of $N_{i,j}$ is (see central grey box of Fig. 3):

$$A_{i,j} = \{(s^{i,j}, n_i^{i,j}), (s^{i,j}, n_j^{i,j}), (s^{i,j}, f^{i,j}), \\ (n_i^{i,j}, t^{i,j}), (n_j^{i,j}, t^{i,j}), (f^{i,j}, t^{i,j})\}$$

The whole graph $D = (N, A)$ is constructed by concatenating the graphs $D_{i,j}$ (for all $1 \leq i < j \leq n$) according to the lexicographic order of their indices i, j. The sink $t^{i,j}$ of each graph $D_{i,j}$ is connected to the source of the graph which immediately follows $D_{i,j}$. A distinguished vertex s is connected to the source of the first subgraph $D_{i',j'}$, while the sink of the last subgraph $D_{i'',j''}$ is connected to a second distinguished vertex t. Figure 3 depicts such a construction.

The set R of required pairs is defined as $\bigcup_{1\leq i\leq n} R_i$, where $R_i = \{[n_i^{i,j}, n_i^{i,h}] \mid \{v_i, v_j\}, \{v_i, v_h\} \in E\}$.

The following lemmas prove the correctness of the reduction.

LEMMA 4.1. *Let $G = (V, E)$ be an undirected (connected) graph and let $\langle D = (N, A), R\rangle$ be the corresponding instance of 3-PCRP. Then, given a 3-coloring of G we can compute in polynomial time three paths of D that cover all its vertices and every required pair in R.*

Proof. Consider a 3-coloring of G and let $\{V_1, V_2, V_3\}$ be the tri-partition of V induced by the 3-coloring. We show how to compute in polynomial time three paths π_1, π_2, π_3 that cover all the vertices of D and every required pair in R. For each $v_i \in V_c$, path π_c passes through vertices $n_i^{i,j}$ of subgraphs $D_{i,j}$ for every $v_j \in V$ such that $\{v_i, v_j\} \in E$, while for each subgraph $D_{p,q}$ such that $v_p, v_q \notin V_c$, π_c passes through vertices $f^{p,q}$. Note that each π_c is well-defined, since there does not exist a pair of vertices $n_i^{i,j}$, $n_j^{i,j}$ associated with the same color c (otherwise $\{V_1, V_2, V_3\}$ is not a 3-coloring of G).

We show that π_1, π_2, π_3 cover every vertex of N. Note that for each $\{v_i, v_j\} \in E$, since v_i and v_j have different colors, by construction one of the paths π_1, π_2, π_3 passes through $n_i^{i,j}$ (say π_{c_1}), while another one passes through $n_j^{i,j}$ (say π_{c_2}). As a consequence, by construction we have that π_{c_3} passes through $f^{i,j}$. The only vertices that might be not covered are $s^{i,j}$ and $t^{i,j}$, for $\{v_i, v_j\} \in E$. However, these vertices are articulation points, hence all the three paths necessarily pass through them.

Now, we show that every required pair in R_i is covered. By construction, the vertices $n_i^{i,j}$ of D associated with the same vertex v_i of G belong to the same path π_c, where c is the color of v_i. Therefore, all the required pairs in each R_i are covered by one of the three paths. □

LEMMA 4.2. *Let $G = (V, E)$ be an undirected graph and let $\langle D = (N, A), R\rangle$ be the corresponding instance of 3-PCRP. Then, given three paths in D that cover all its vertices and every required pair in R, we can compute in polynomial time a 3-coloring of G.*

Proof. Consider three paths π_1, π_2, π_3 of D that cover all the vertices of D and every required pair in R, and we show how

[1] We recall that the class XP contains those problems that, given a parameter k, can be solved in time $O(n^{f(k)})$.

to compute in polynomial time a corresponding 3-coloring of the graph G.

First, we prove a property of the three paths π_1, π_2, π_3. We show that, given a vertex $v_i \in V$, there exists at least one path among π_1, π_2, π_3 that covers all the required pairs in R_i. Consider a vertex $v_i \in V$. Since G is connected, it follows that there exists at least one vertex adjacent to v_i, w.l.o.g. v_j, such that $\{v_i, v_j\} \in E$. Now, consider the subgraph $D_{i,j}$. By construction, a solution of MinPCRP must contain three different paths, each one passing through one of the vertices $n_i^{i,j}$, $n_j^{i,j}$, $f^{i,j}$. Now, assume that path π_1 passes through $n_i^{i,j}$. Obviously, π_2 and π_3 cannot pass through $n_i^{i,j}$. As a consequence, since π_1 is the only path that covers $n_i^{i,j}$ and since R_i contains a pair $[n_i^{i,j}, n_i^{i,h}]$ for each $h \neq j$ such that $\{v_i, v_h\} \in E$, it follows that all the vertices $n_i^{i,h}$ with $\{v_i, v_h\} \in E$ must belong to π_1. It follows that, given a vertex $v_i \in V$, there exists one path in $\{\pi_1, \pi_2, \pi_3\}$ that covers all the required pairs in R_i.

Now, we define the 3-coloring of G, where $C = \{c_1, c_2, c_3\}$ is the set of colors. If a vertex $n_i^{i,j}$ is covered by a path π_x, $1 \leq x \leq 3$, then we assign the color c_x to vertex v_i. The coloring is well-defined since, as noted above, a single path covers all the vertices $n_i^{i,j}$ of D associated with the same vertex v_i of G. The coloring is also feasible, that is $c(v_i) \neq c(v_j)$ when $\{v_i, v_j\} \in E$, since, by construction, vertices $n_i^{i,j}$ and $n_j^{i,j}$ are covered by different paths (hence $c(v_i) \neq c(v_j)$). □

Since 3-PCRP is clearly in NP, the following result is a consequence of the previous lemmas and of the NP-hardness of 3C [15].

THEOREM 4.1. *3-PCRP is NP-complete.*

Proof. The NP-hardness of 3-PCRP follows directly from Lemmas 4.1 and 4.2 and from the NP-completeness of 3C [15]. 3-PCRP is in NP, since, given three paths π_1, π_2, π_3, we can verify in polynomial time that π_1, π_2, π_3 cover all the vertices of D and that every required pair in R is covered by some path in $\{\pi_1, \pi_2, \pi_3\}$. □

The reduction can be easily modified in order to show that also 3-RPC, that is the restriction of MinRPC where we ask whether there exist k paths that cover all the required pairs, is NP-complete for any $k > 2$.

COROLLARY 4.1. *3-RPC is NP-complete.*

Proof. We obtain this result by modifying the reduction from 3C to 3-PCRP presented above. Let $G = (V, E)$ be the undirected graph given as input to 3C and let $\langle D = (N, A), R\rangle$ be the corresponding instance of 3-PCRP. First, we can assume that G does not contain vertices with degree 1 (i.e. vertices with only one edge incident to them). Otherwise, these vertices can be removed from G since they can be always easily colored with a color different to that of their single adjacent vertex. As a consequence, we can assume that all the vertices $n_i^{i,j}$ and $n_j^{i,j}$ of N belong to some required pair of R. Now, construct the instance of 3-RPC $\langle D = (N, A), \hat{R}\rangle$, where $\hat{R} := R \cup \{[s^{i,j}, f^{i,j}] \mid \{v_i, v_j\} \in E\}$. We claim that there exist three paths covering all the vertices in N and all the required pairs in R if and only if there exist three paths covering all the required pairs in $\hat{R}$. Note that a set of paths covering all the vertices of D also covers the required pairs in $\hat{R} \setminus R$. Hence, if there exist 3 paths covering all the vertices in N and all the required pairs in R, then the same paths cover all the required pairs in $\hat{R}$. Conversely, if there exist three paths covering all the required pairs of $\hat{R}$, then the same paths cover all the required pairs of R. Moreover, since all the vertices $n_i^{i,j}$, $n_j^{i,j}$ and $f^{i,j}$ of N belong to some required pair of $\hat{R}$ and since vertices $s^{i,j}$ and $t^{i,j}$ are articulation points, we have that these paths cover also all the vertices of N.

Finally, we also have that 3-RPC is clearly in NP. As a consequence, by Lemmas 4.1 and 4.2 and by the NP-completeness of 3C [15], we have that 3-RPC is NP-hard. □

4.2. A polynomial time algorithm for 2-PCRP

In this section, we give a polynomial time algorithm for computing a solution of 2-PCRP. Note that 1-PCRP can be easily solved in polynomial time, as there exists a solution of 1-PCRP if and only if the reachability relation of the vertices of the input graph is a total order.

The algorithm for solving 2-PCRP is based on a polynomial-time reduction to the 2-Clique Partition problem, which, given an undirected graph $G = (V, E)$, asks whether there exists a partition of V in two sets V_1, V_2 both inducing a clique in G. Computing the existence of a 2-Clique Partition over a graph G is equivalent to computing if there exists a 2-Coloring of the complement graph G' (hence deciding if G' is bipartite), which is well known to be solvable in polynomial time [15, probl. GT15]. To perform this reduction we assume that given $\langle D = (N, A), R\rangle$, instance of 2-PCRP, every vertex of the graph D belongs to at least one required pair in R. Otherwise, we add to R the required pairs $[s, v_i]$ for all $v_i \in N$ that do not belong to any required pair. Therefore, a solution that covers all the required pairs in R covers also all the vertices, hence it is a feasible solution of 2-PCRP. Moreover, note that this transformation does not affect the solution of 2-PCRP, since all the paths start from s and cover all the nodes of the graph, including the additional required pairs.

The algorithm computes, if exists, a solution for an instance $\langle D = (N, A), R\rangle$ of 2-PCRP by computing a 2-Clique Partition of the compatibility graph of R. We recall that the compatibility graph is the graph obtained from the compatibility digraph discarding the edge orientation. Given the compatibility relation $\mathcal{C}$, we denote as $\hat{\mathcal{C}}$ the set of edges of the compatibility graph (i.e. the set $\{\{r', r''\} \mid (r', r'') \in \mathcal{C}\}$). Since the computation of the compatibility graph and of a 2-Clique Partition can be performed in polynomial time, the algorithm solves 2-PCRP in polynomial time.

The algorithm is based on the following property.

Lemma 4.3. *Given an instance $\langle D = (N, A), R\rangle$ of 2-PCRP and the compatibility graph $G = (R, \hat{\mathcal{C}})$ of R, then there exists a path π that covers a set R' of required pairs if and only if R' is a clique of G.*

Proof. If there exists a path π which covers all the required pairs in R', then, by definition of the compatibility relation, R' is clearly a clique of G.

We claim that, if R' is a clique of G, then $\mathcal{C}(R')$ is a total order. First, note that, for each pair $r', r'' \in R'$, we have that either (r', r'') or (r'', r') is in $\mathcal{C}(R')$. Moreover, since the compatibility digraph is acyclic, then also $(R', \mathcal{C}(R'))$ is acyclic. As a consequence, $\mathcal{C}(R')$ is transitive and, by the irreflexivity of $\mathcal{C}$, we conclude that $\mathcal{C}(R')$ is a total order. By Lemma 3.4, since R' is a chain of $\mathcal{C}$, there exists a path π covering the required pairs in R'. □

From Lemma 4.3, it follows that, in order to compute the existence of a solution of 2-PCRP over the instance $\langle D = (N, A), R\rangle$ (in which every vertex of D belongs to at least one required pair in R), we have to compute if there exists a 2-Clique Partition of the corresponding graph G. Since the 2-Clique Partition problem can be solved in polynomial-time [15, probl. GT15], we can conclude that 2-PCRP can be decided in polynomial time.

5. PARAMETERIZED COMPLEXITY OF MAXRPSP

In this section, we consider the parameterized complexity of MaxRPSP. We show that, although MaxRPSP is W[1]-hard (hence unlikely fixed-parameter tractable) when parameterized by the number of required pairs covered by a single path (Section 5.1), the problem becomes fixed-parameter tractable if the maximum overlapping degree is the parameter (Section 5.2).

5.1. W[1]-hardness of MaxRPSP parameterized by the optimum

In this section, we investigate the parameterized complexity of MaxRPSP when parameterized by the size of the solution, that is the maximum number of required pairs covered by a single path, and we prove that the problem is W[1]-hard (note that this result implies the NP-hardness of MaxRPSP). This result shows that it is unlikely that the problem is fixed-parameter tractable, when parameterized by the number of required pairs covered by a single path. For details on the theory of fixed-parameter complexity, we refer the reader to [13, 14].

We prove this result via a parameterized reduction from the h-Clique problem to the decision version of MaxRPSP (k-RPSP), parameterized by the sizes of the respective solutions. Given an undirected graph $G = (V, E)$ and an integer h, h-Clique asks to decide if there exists a clique $C \subseteq V$ of size h. On the other hand, given a DAG D, a set R of required pairs, and an integer k, the k-RPSP problem consists of deciding if there exists a path in D that 'covers' k required pairs. We recall that h-Clique is known to be W[1]-hard [18].

First, we start by showing how to construct an instance of k-RPSP starting from an instance of h-Clique. Given an (undirected) graph $G = (V, E)$ with n vertices $v_1, \ldots, v_n$, we construct the associated DAG $D = (N, A)$ as follows. The set N of vertices is defined as:

$$N = \{v_i^z \mid v_i \in V, 1 \leq z \leq h\} \cup \{s, t\}$$

Informally, N consists of two distinguished vertices s, t and of h copies $v_i^1, \ldots, v_i^h$ of every vertex v_i of G.

The set of arcs A is defined as:

$$\begin{aligned} A = \{&(v_i^z, v_j^{z+1}) \mid \{v_i, v_j\} \in E, 1 \leq z \leq h-1\} \\ &\cup \{(s, v_i^1), (v_i^h, t) \mid v_i \in V\} \end{aligned}$$

Informally, we connect every two consecutive copies associated with vertices that are adjacent in G, the source vertex s to all the vertices v_i^1, with $1 \leq i \leq n$, and all the vertices v_i^h, with $1 \leq i \leq n$, to the sink vertex t.

The set R of required pairs is defined as:

$$R = \{[v_i^x, v_j^y] \mid \{v_i, v_j\} \in E, 1 \leq x < y \leq h\}$$

Informally, for each edge $\{v_i, v_j\}$ of G there is a required pair $[v_i^x, v_j^y]$, $1 \leq x < y \leq h$, between every two different copies associated with v_i, v_j.

By construction, the vertices in N (except for s and t) are partitioned into h *independent sets* $I_z = \{v_i^z \mid 1 \leq i \leq n\}$, with $1 \leq z \leq h$, each one containing a copy of every vertex of V. Moreover, the arcs of A only connect two vertices of consecutive subsets I_z and I_{z+1}, with $1 \leq z \leq h-1$. Figure 4 presents an example of directed graph D associated with an undirected graph G.

Now, we are able to prove the main properties of the reduction.

Lemma 5.1. *Let $G = (V, E)$ be an undirected graph and $\langle D = (N, A), R\rangle$ be the associated instance of k-RPSP. Then*: (1) *starting from an h-clique in G we can compute in polynomial time a path π in D that covers $\binom{h}{2}$ required pairs of R*; (2) *starting from a path π in D that covers $\binom{h}{2}$ required pairs we can compute in polynomial time an h-clique in G.*

Proof. (1) Starting from an h-clique C in G we show how to compute a path π in D that covers $\binom{h}{2}$ required pairs of R.

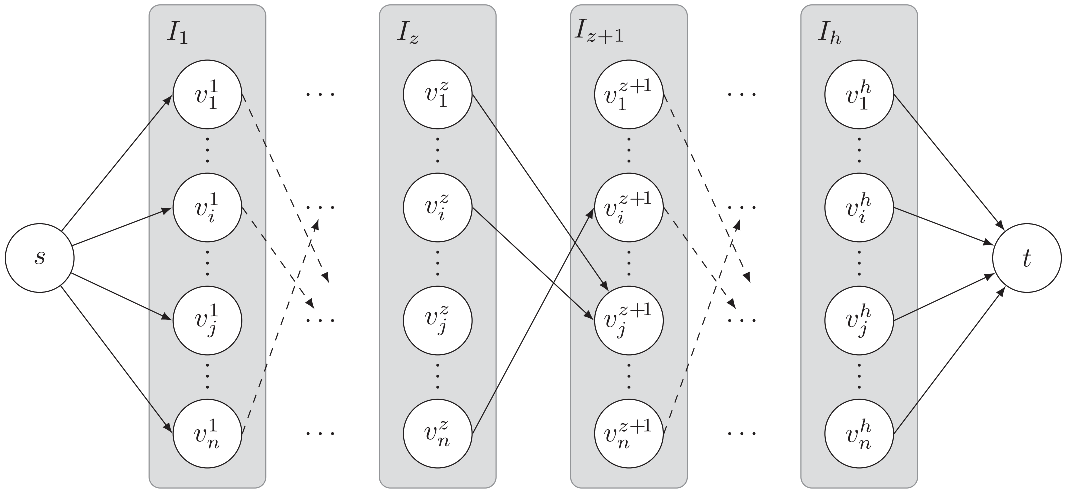

FIGURE 4. Example of DAG $D = (N, A)$ associated with an instance $G = (V, E)$ of the h-Clique problem. Each gray box highlights an independent set I_z composed of one copy of the vertices in V. Edges (v_1^z, v_j^{z+1}), (v_i^z, v_j^{z+1}) and (v_n^z, v_i^{z+1}) are some of the directed edges in A associated with edges $\{v_1, v_j\}, \{v_i, v_j\}, \{v_i, v_n\} \in E$.

Let $C = \{v_{i_1}, \ldots, v_{i_h}\}$ be a clique of G and let $\langle v_{i_1}, \ldots, v_{i_h}\rangle$ be an arbitrary ordering of C. Let $\pi_C = \langle s, v_{i_1}^1, \ldots, v_{i_h}^h, t\rangle$ be a sequence of vertices obtained by selecting the vertex $v_{i_z}^z$ for each independent set I_z, with $1 \le z \le h$ (in addition to vertices s and t). Since C is a clique of G, by construction of D, every pair of vertices $(v_{i_z}^z, v_{i_{z+1}}^{z+1})$ is connected by an arc, hence π_C is a path of D. Moreover, the path π_C covers exactly $\binom{h}{2}$ required pairs of R because, by construction of R, there exists a pair between every two copies of vertices which are adjacent in G. More precisely, since the clique C has all the possible edges among its h vertices, the number of required pairs covered by the path π_C is $\binom{h}{2}$.

(2) Let π be a path in D that covers a set $R' \subseteq R$ of $\binom{h}{2}$ required pairs, then we show how to compute in polynomial time an h-clique C in G. Note that, by construction of D, the path π must contain exactly one vertex v_i^z, $1 \le i \le n$ and $1 \le z \le h$, for each independent set I_z of D. By construction of set R, each vertex v_i^z of π appears in at most $h-1$ required pairs of R'. Hence, the total number of required pairs covered by the path π, which contains exactly h inner vertices v_i^z, is at most $h(h-1)/2 = \binom{h}{2}$. Let C be the set $\{v_i \mid v_i^z \in \pi \setminus \{s, t\}\}$. We claim that C is an h-clique. First, we prove that C contains h vertices. Suppose to the contrary that C has less than h vertices. Then, there exist two vertices $v_{i'}^x$ and $v_{i''}^y$ of π that correspond to the same vertex v_i of C, that is $i' = i'' = i$. Since $[v_i^x, v_i^y] \notin R$, it follows that each v_i^x, v_i^y appears in at most $h-2$ required pairs of R'. As a consequence, the total number of required pairs covered by the path π is strictly less than $\binom{h}{2}$, violating the initial hypothesis that π covers $\binom{h}{2}$ required pairs of R. Hence C contains h vertices. As all the internal vertices of π (i.e. all its vertices but s and t) represent distinct vertices of G, then all the required pairs covered by π represent distinct edges of G. The only undirected graph with h vertices and $\binom{h}{2}$ edges is the complete graph, hence C is an h-clique of G. □

The W[1]-hardness of k-RPSP easily follows from Lemma 5.1 and from the W[1]-hardness of h-Clique when parameterized by h [18].

Theorem 5.1. *k-RPSP is W[1]-hard when parameterized by the number of required pairs covered by a path.*

Proof. The result follows from Lemma 5.1 and from the W[1]-hardness of h-Clique when parameterized by h [18]. □

5.2. An FPT algorithm for MaxRPSP parameterized by the maximum overlapping degree

In this section, we propose a fixed-parameter algorithm (FPT) for the MaxRPSP problem, where the parameter is the maximum overlapping degree. For the rest of the section, let $\langle D = (N, A), R\rangle$ be an instance of the MaxRPSP problem.

For ensuring its correctness, the algorithm will consider the required pairs in R in an order resulting from the topological ordering of the compatibility digraph of R. For ease of exposition, given such a fixed order, we represent the ith required pair of the ordering as $[v_i^1, v_i^2]$ and, whenever no confusion arises, we will refer to that required pair as *i-pair*.

The parameterized algorithm is based on dynamic programming. In fact, we can decompose a path π, starting at s, ending at a vertex v and covering k required pairs, into two subpaths: the first one—π_1—starts at s, ends at a vertex v', and covers k_1 required pairs, while the other one—π_2—starts at v', ends at v and covers the remaining $k_2 = k - k_1$ required pairs (using vertices of π_1, possibly). The key point to define the recurrence is that, for each required pair r, it suffices to keep track of the set of required pairs overlapping r and covered by the path. To this aim, for each required pair $[v_i^1, v_i^2]$, we define the set $OP([v_i^1, v_i^2])$ as the set of vertices v such that v belongs to a required pair that overlaps $[v_i^1, v_i^2]$ and such that v_i^2 is reachable from v. By a slightly abuse of the notation, we consider that $OP([v_i^1, v_i^2])$ always contains vertex v_i^1.

The recurrence relies on the following observation. Let π be a path covering a set $R' \subseteq R$ of required pairs and let $N(R')$ be the set of vertices belonging to the required pairs in R'. Consider two overlapping required pairs $[v_i^1, v_i^2]$ and $[v_j^1, v_j^2]$ in R', with $j < i$. Then, either $[v_j^1, v_j^2]$ is nested in $[v_i^1, v_i^2]$ (hence the fact that π covers the pair $[v_j^1, v_j^2]$ can be checked by the recurrence looking only at the required pairs that overlap with $[v_i^1, v_i^2]$) or pairs $[v_i^1, v_i^2]$ and $[v_j^1, v_j^2]$ are alternated. In the latter case, since $[v_i^1, v_i^2]$ is in R', we only have to consider the vertices in the set $N(R') \cap OP([v_i^1, v_i^2]) \cap OP([v_j^1, v_j^2])$. Moreover, let p_i be the number of required pairs that overlap the required pair $[v_i^1, v_i^2]$. Then $|OP([v_i^1, v_i^2])|$ is at most $2p_i$. Hence, the cardinality of set $N(R') \cap OP([v_i^1, v_i^2]) \cap OP([v_j^1, v_j^2])$ is bounded by $2\max(p_i, p_j)$. Furthermore, given two sets S and S' of vertices such that $S \subseteq OP([v_i^1, v_i^2])$ and $S' \subseteq OP([v_j^1, v_j^2])$, we say that S is in *agreement* with S' if $S \cap (OP([v_i^1, v_i^2]) \cap OP([v_j^1, v_j^2])) = S' \cap (OP([v_i^1, v_i^2]) \cap OP([v_j^1, v_j^2]))$. Informally, when S and S' are in agreement, they must contain the same subset of vertices of $OP([v_i^1, v_i^2]) \cap OP([v_j^1, v_j^2])$.

Let $P([v_i^1, v_i^2], S)$ denote the maximum number of required pairs covered by a path π ending at vertex v_i^2 and such that the set $S \subseteq OP([v_i^1, v_i^2])$ is covered by π. In the following, we present the recurrence to compute $P([v_i^1, v_i^2], S)$. For ease of exposition, we only focus on vertices that appear as second vertices of the required pairs. In fact, paths that do not end at such vertices are not able to cover new required pairs. Furthermore, for simplicity, we consider the source s as the second vertex of a fictitious required pair (with index 0) $[\perp, s]$ which does not overlap any other required pair. Such a fictitious required pair does not contribute to the total number of required pairs covered by the path.

The recurrence is:

$$P([v_i^1, v_i^2], S) = \max\{P([v_j^1, v_j^2], S') + |Ov([v_i^1, v_i^2], S, S')|\} \tag{1}$$

for each $[v_j^1, v_j^2]$ and S' such that:

(i) $[v_j^1, v_j^2]$ not nested in $[v_i^1, v_i^2]$ and $j < i$;
(ii) S' in agreement with S;
(iii) there exists a path from v_j^2 to v_i^2 covering all vertices in $S \setminus S'$

and where $Ov([v_i^1, v_i^2], S, S') = \{[v_h^1, v_h^2] \mid [v_h^1, v_h^2]$ is nested in $[v_i^1, v_i^2] \wedge v_h^1 \in S \wedge v_h^2 \in S \setminus S'\}$. Note that each required pair is assumed to be nested in itself.

The base case of the recurrence is $P([\perp, s], \emptyset) = 0$.

The correctness of the recurrence derives from the following two lemmas.

Lemma 5.2. *If $P([v_i^1, v_i^2], S) = k$, then there exists a path π in D ending at v_i^2, such that every vertex in S belongs to π and the number of required pairs covered by π is k.*

Proof. We prove the lemma by induction on the index i. It is easy to see that the base case holds. Assume that the lemma holds for index values less than i, we prove that the lemma holds for i. Let $P([v_i^1, v_i^2], S) = k$. By Equation (1), there exists a vertex v_j^2 with $j < i$, such that $P([v_j^1, v_j^2], S') = k_1$ for some set S' in agreement with S. Assume that $|Ov([v_i^1, v_i^2], S, S')| = k_2$, with $k_1 + k_2 = k$. By induction hypothesis, since $P([v_j^1, v_j^2], S') = k_1$, there exists a path π' ending at v_j^2, covering every vertex in S', and such that π' covers k_1 required pairs. Furthermore, the k_2 covered required pairs have at least one vertex in $S \setminus S'$, hence the vertices of such required pairs belong to a path π'' which starts at v_j^2 and ends at v_i^2 (path π'' exists by hypothesis). But then, the path obtained by the concatenation of π' and π'' covers $k_1 + k_2$ required pairs. □

Lemma 5.3. *Let π be a path in D ending at v_i^2 and covering k required pairs. Let S be the set of all the vertices belonging to required pairs covered by π and overlapping $[v_i^1, v_i^2]$. Then $P([v_i^1, v_i^2], S) \geq k$.*

Proof. We prove the lemma by induction on the index i. It is easy to see that the base case holds. Assume that the lemma holds for index values less than i, we prove that the lemma holds for i. Let π be a path, ending at v_i^2, that covers k required pairs and let S be the set of vertices that belong to the required pairs covered by π and overlapping $[v_i^1, v_i^2]$. We claim that $P([v_i^1, v_i^2], S) \geq k$. Consider the rightmost vertex v_j^2 of π such that v_j^2 belongs to a required pair covered by π and not nested in the i-pair. Decompose path π into two parts: one—π'—from s to v_j^2, and the other one—π''—from v_j^2 to v_i^2. Let S' be the set of vertices that belong to the required pairs covered by π and overlapping $[v_j^1, v_j^2]$. Let k' be the number of required pairs covered by π' and k'' be the number of the remaining required pairs covered by π (that is, $k = k' + k''$). First, note that $k'' = |Ov([v_i^1, v_i^2], S, S')|$. By induction hypothesis $P([v_j^1, v_j^2], S') = k_1$ for some $k_1 \geq k'$. Moreover, by construction, S' is in agreement with S and the subpath of π from v_j^2 to v_i^2 covers all the vertices in $S \setminus S'$. As a consequence, by Equation (1), $P([v_i^1, v_i^2], S)$ is at least $k_1 + k'' \geq k' + k'' = k$, which concludes the proof. □

Let p be the maximum number of overlapping required pairs in $\langle D, R\rangle$ (that is, $p = \max_i\{p_i\}$). Then, the cardinality of the subsets S of each entry $P([v_i^1, v_i^2], S)$ is bounded by $2p$. As a consequence, for computing each entry $P([v_i^1, v_i^2], S)$ there must be considered at most 2^{2p} subsets S' and at most $|R|$ required pairs $[v_j^1, v_j^2]$. Assume that the DAG D has been pre-processed in order to query the reachability of two vertices in constant time (for example, by computing the transitive closure of its adjacency matrix) and that each set $OP([v_i^1, v_i^2])$ has been pre-computed and represented as a sorted list, where each element of the list is a vertex, and elements are sorted according to a topological ordering of the DAG. Clearly, such a pre-processing step can be performed in polynomial time. For each entry and for each choice of S' and $[v_j^1, v_j^2]$, the first condition of the dynamic programming recurrence (Equation (1)) can be checked in time $O(1)$, the second condition can be checked in time $O(p)$, and the third condition can be checked in time $O(p|R|)$ (the existence of the path can be checked in time $O(p)$ by checking that consecutive elements of $S \setminus S'$ in a topological order of D are reachable, while the cardinality of $Ov([v_i^1, v_i^2], S, S')$ can be computed in time $O(p|R|)$ by enumerating all the $|R|$ required pairs and checking in time $O(p)$ if they belong to $Ov([v_i^1, v_i^2], S, S')$). Then, each entry $P([v_i^1, v_i^2], S)$ requires time $O(p2^{2p}|R|)$ to be computed, and, since there exist $O(2^{2p}|R|)$ entries, the recurrence can be computed in time $O(p4^{2p}|R|^2)$. From Lemmas 5.2 and 5.3, it follows that an optimal solution for MaxRPSP can be obtained by looking for the maximum of the values $P([v_i^1, v_i^2], S)$. Hence, the overall time complexity of the algorithm is bounded by $O(p4^{2p}|R|^2)$ (plus a polynomial pre-processing time.

6. CONCLUSIONS

In this paper, we studied three constrained variants of the well-known MinPC problem on DAGs, namely Minimum Required Pair Cover (MinRPC), Minimum Path Cover with Required Pairs (MinPCRP), and Maximum Required Pairs with a Single Path (MaxRPSP). These problems are motivated by relevant applications in software testing and in bioinformatics. More precisely, we complemented the computational complexity results by Ntafos and Hakimi [7] on MinRPC by identifying a sharp tractability borderline for this problem depending on the maximum overlapping degree of the instance. Furthermore, we extended the analysis on MinPCRP by establishing a second tractability borderline depending on the size of the solution (i.e. the number of paths). Some of these results have been independently obtained by Rizzi *et al.* [12]. Finally, we showed that, albeit MaxRPSP is W[1]-hard when parameterized by the optimum, there exists a fixed-parameter algorithm for the problem when the parameter is the maximum overlapping degree.

Our results do not rule out the existence of constant-factor approximation algorithms for the problems we proposed. For this reason, and since these algorithms would have a significant impact on the bioinformatics applications motivating these problems, an analysis in that direction could be of great interest.

FUNDING

This work was supported by Università degli Studi di Milano-Bicocca (Fondo di ateneo 2011 ‘Metodi algoritmici per l’analisi di strutture combinatorie in bioinformatica’ to S.B., P.B. and Y.P.); Ministero dell’Istruzione, dell’Università e della Ricerca (PRIN 2010-2011 ‘Automi e Linguaggi Formali: Aspetti Matematici e Applicativi’ code H41J12000190001 to P.B., R.D. and Y.P., ‘Flagship InterOmics’ code PB05 to S.B., ‘HIRMA’ code RBAP11YS7K to S.B.); and European Union Seventh Framework Programme (‘MIMOmics’ code 305280 to S.B.).

REFERENCES

[1] Beerenwinkel, N., Beretta, S., Bonizzoni, P., Dondi, R. and Pirola, Y. (2014) Covering Pairs in Directed Acyclic Graphs. *Proc. LATA 2014, Madrid, Spain*, March 10–14, pp. 126–137. Springer, Switzerland.

[2] Fulkerson, D.R. (1956) Note on Dilworth’s decomposition theorem for partially ordered sets. *Proc. Amer. Math. Soc.*, **7**, 701–702.

[3] Eriksson, N., Pachter, L., Mitsuya, Y., Rhee, S.-Y., Wang, C., Gharizadeh, B., Ronaghi, M., Shafer, R.W. and Beerenwinkel, N. (2008) Viral population estimation using pyrosequencing. *PLoS Comput. Biol.*, **4**, e1000074.

[4] Trapnell, C., Williams, B.A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M.J., Salzberg, S.L., Wold, B.J. and Pachter, L. (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.*, **28**, 516–520.

[5] Bao, E., Jiang, T. and Girke, T. (2013) BRANCH: boosting RNA-Seq assemblies with partial or related genomic sequences. *Bioinformatics*, **29**, 1250–1259.

[6] Gabow, H., Maheshwari, S. and Osterweil, L. (1976) On two problems in the generation of program test paths. *IEEE Trans. Softw. Eng.*, **SE-2**, 227–231.

[7] Ntafos, S. and Hakimi, S. (1979) On path cover problems in digraphs and applications to program testing. *IEEE Trans. Softw. Eng.*, **SE-5**, 520–529.

[8] Wu, B.Y. (2012) On the maximum disjoint paths problem on edge-colored graphs. *Discrete Optim.*, **9**, 50–57.

[9] Bonizzoni, P., Dondi, R. and Pirola, Y. (2013) Maximum disjoint paths on edge-colored graphs: approximability and tractability. *Algorithms*, **6**, 1–11.

[10] Guttman, M. *et al.* (2010) Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. *Nat. Biotechnol.*, **28**, 503–510.

[11] Song, L. and Florea, L. (2013) CLASS: constrained transcript assembly of RNA-seq reads. *BMC Bioinf.*, **14**, 1–8.

[12] Rizzi, R., Tomescu, A. and Mäkinen, V. (2014) On the complexity of minimum path cover with subpath constraints for multi-assembly. *BMC Bioinf.*, **15**, S5.

[13] Downey, R. and Fellows, M. (1999) *Parameterized Complexity*. Springer, New York.

[14] Niedermeier, R. (2006) *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, Oxford.

[15] Garey, M. and Johnson, D. (1979) *Computer and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco.

[16] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A. and Protasi, M. (1999) *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*. Springer, Berlin.

[17] Alimonti, P. and Kann, V. (2000) Some APX-completeness results for cubic graphs. *Theor. Comput. Sci.*, **237**, 123–134.

[18] Downey, R.G. and Fellows, M.R. (1995) Fixed-parameter tractability and completeness II: on completeness for $W[1]$. *Theor. Comput. Sci.*, **141**, 109–131.