Check for
updates

Romeo and Juliet Meeting in Forest Like Regions

Neeldhara Misra[1] · **Manas Mulpuri[1]** · **Prafullkumar Tale[2]** ·
Gaurav Viramgami[1]

Abstract

The game of rendezvous with adversaries is a game on a graph played by two players:
Facilitator and *Divider*. Facilitator has two agents and Divider has a team of $k \geq 1$
agents. While the initial positions of Facilitator's agents are fixed, Divider gets to select
the initial positions of his agents. Then, they take turns to move their agents to adjacent
vertices (or stay put) with Facilitator's goal to bring both her agents at same vertex and
Divider's goal to prevent it. The computational question of interest is to determine if
Facilitator has a winning strategy against Divider with k agents. Fomin, Golovach, and
Thilikos [WG, 2021] introduced this game and proved that it is PSPACE-hard and co-
W[2]-hard parameterized by the number of agents. This hardness naturally motivates
the structural parameterization of the problem. The authors proved that it admits an
FPT algorithm when parameterized by the modular width and the number of allowed
rounds. However, they left open the complexity of the problem from the perspective
of other structural parameters. In particular, they explicitly asked whether the problem
admits an FPT or XP-algorithm with respect to the treewidth of the input graph. We
answer this question in the negative and show that RENDEZVOUS is co-NP-hard even
for graphs of constant treewidth. Further, we show that the problem is co-W[1]-hard

✉ Prafullkumar Tale
    prafullkumar@iiserpune.ac.in
    https://pptale.github.io/

    Neeldhara Misra
    neeldhara.m@iitgn.ac.in
    https://www.neeldhara.com

    Manas Mulpuri
    mulpuri.m@iitgn.ac.in

    Gaurav Viramgami
    viramgami.g@iitgn.ac.in

[1] Indian Institute of Technology, Gandhinagar, India

[2] Indian Institute of Science Education and Research, Bhopal, India

🖄 Springer

when parameterized by the feedback vertex set number and the number of agents, and is unlikely to admit a polynomial kernel when parameterized by the vertex cover number and the number of agents. Complementing these hardness results, we show that the RENDEZVOUS is FPT when parameterized by both the vertex cover number and the solution size. Finally, for graphs of treewidth at most two and girds, we show that the problem can be solved in polynomial time.

Keywords Games on graphs · Dynamic separators · W[1]-hardness · Structural parametersization · Treewidth

1 Introduction

The game of rendezvous with adversaries on a graph—RENDEZVOUS—is a natural dynamic version of the problem of finding a vertex cut between two vertices s and t introduced by Fomin, Golovach, and Thilikos [4]. The game is played on a finite undirected connected graph G by two players: *Facilitator* and *Divider*. Facilitator has two agents Romeo and Juliet that are initially placed in designated vertices s and t of G. Divider, on the other hand, has a team of $k \geq 1$ agents $D_1, \ldots, D_k$ that are initially placed in some vertices of $V(G)\backslash\{s, t\}$ chosen by him. We note that a single vertex can accommodate multiple agents of Divider.

Then the players make their moves in turn, starting with Facilitator. At every move, each player moves some of his/her agents to adjacent vertices or keeps them in their old positions. No agent can be moved to a vertex that is currently occupied by adversary's agents. Both players have complete information about G and the positions of all the agents. Facilitator aims to ensure that Romeo and Juliet meet; that is, they are in the same vertex. The task of Divider is to prevent the rendezvous of Romeo and Juliet by maintaining $D_1, \ldots, D_k$ in positions that block the possibility to meet. Facilitator wins if Romeo and Juliet meet, and Divider wins if they succeed in preventing the meeting of Romeo and Juliet forever. This setup naturally leads to the following computational question.

RENDEZVOUS

Input: A graph G with two given vertices s and t, and a positive integer k.
Question: Can Facilitator win on G starting from s and t against Divider with k agents?

We will often refer to k, the number of agents employed by Divider to keep Romeo and Juliet separated, as the "solution size" for this problem.

Known Results

Fomin, Golovach, and Thilikos [4] initiated an extensive study of the computational complexity of RENDEZVOUS. They concluded that the problem is PSPACE-hard and co-W[2]-hardwhen parameterized by the number of Divider's agents, while also demonstrating an $|V(G)|^{\mathcal{O}(k)}$ algorithm based on backtracking stages over the game

arena. They also show that the problem admits polynomial time algorithms on chordal graphs and P_5-free graphs. A related problem considered is RENDEZVOUS IN TIME, which asks if Facilitator can force a win in at most τ steps. It turns out that REN-DEZVOUS IN TIME is co-NP-complete even for $\tau = 2$ and is FPT when parameterized by τ and the neighborhood diversity of the graph. The latter is an ILP-based approach and uses the fact that INTEGER LINEAR PROGRAMMING FEASIBILITY is FPT in the number of variables. We refer readers to [4], and references within, for more related problems.

The smallest number of agents that Divider needs to use to win on a graph G is called the "dynamic" separation number of G. We denote this number by $d_G(s, t)$. Note that if s and t are adjacent or $s = t$, then $d_G(s, t) := +\infty$. The "static" separation number between s and t, the original positions of Facilitator's agents, is simply the smallest size of a (s, t)-vertex cut, i.e, a subset of vertices whose removal disconnects s and t. We use $\lambda_G(s, t)$ to denote the minimum size of a vertex (s, t)-separator in G. It is clear that $d_G(s, t) \leq \lambda_G(s, t)$, since positioning $\lambda_G(s, t)$ many guards on the vertices of a (s, t)-vertex allows Divider to win the game right away. It turns out that $d_G(s, t) = 1$ if and only if $\lambda_G(s, t) = 1$. However, there are examples of graphs where $d_G(s, t)$ is arbitrarily smaller than $\lambda_G(s, t)$ [4]. The results in [4] for chordal graphs and P_5-free graphs are based on the fact that in these graphs, it turns out that $d_G(s, t) = \lambda_G(s, t)$.

Our Contributions

Given that the problem is hard when parameterized the solution size, often regarded the "standard" parameter, a natural approach is to turn to structural parameters of the input graph. One of the most popular structural parameters in the context of graphs is treewidth, which is a measure of how "tree-like" a graph is. XP and FPT algorithms parameterized by treewidth are natural generalizations of tractability on trees. Indeed, RENDEZVOUS is easy to solve on trees because $\lambda_G(s, t) = 1$ for any distinct s and t when G is a tree and $st \notin E(G)$. The complexity of RENDEZVOUS parameterized by treewidth, however, is wide open—in particular it is not even known if the problem is in XP parameterized by treewidth.

Interestingly, it was pointed out in [4] that if the initial positions s and t are not is the same bag of a tree decomposition of width w, then the upper bound for the dynamic separation number by $\lambda_G(s, t)$ together with the XP algorithm for the standard parameter can be employed to solve the problem in $n^{O(w)}$ time. Thus, the question that was left open by Fomin, Golovach, and Thilikos was if the problem can be solved in the same time if s and t are in the same bag. Our first contribution is to answer this question in the negative by showing that RENDEZVOUS is in fact co-NP-hard even for graphs of *constant* treewidth. In fact, we show more:

Theorem 1 RENDEZVOUS *is* co-NP-hard *even when restricted to:*

- *graphs whose feedback vertex set number is at most 14, or*
- *graphs whose pathwidth is at most 16.*

In particular, RENDEZVOUS *is* para-co-NP-hard *parameterized by treewidth.*

We obtain this hardness by a non-trivial reduction from the 3- DIMENSIONAL MATCHING problem. In the backdrop of this somewhat surprising result, we are motivated to pursue the question of the complexity of RENDEZVOUS for larger parameters.

It turns out that even augmenting the feedback vertex set number or the pathwidth with the solution size is not enough. Specifically, we show that the problem is unlikely to admit an FPT-algorithm even when parameterized by these combined parameters.

Theorem 2 RENDEZVOUS *is* co-W[1]-hard *when parameterized by:*

- *the feedback vertex set number and the solution size, or*
- *the pathwidth and the solution size.*

This result is shown by a parameter preserving reduction from the (MONOTONE) NAE- INTEGER- 3- SAT problem, which was shown to be W[1]-hard when parameterized by the number of variables by Bringmann et al. [1]. Note that with this, we have a reasonably complete understanding of RENDEZVOUS in the combined parameter. Indeed, recall that the problem is co-W[1]-hard and XP parameterized by the solution size alone, and para-co-NP-hard parameterized by the feedback vertex set number alone as shown above (Fig. 6).

Given the above hardness, we consider RENDEZVOUS parameterized by the vertex cover number, a larger parameter compared to both the feedback vertex set number and pathwidth. The status of RENDEVOUS with respect to the vertex cover number parameterization was also left open in [4]. We see that the problem admits a natural exponential kernel in this parameter when combined with the solution size, and is hence FPT in the combined parameter; however this kernel cannot be improved to a polynomial kernel under standard complexity-theoretic assumptions.

Theorem 3 RENDEZVOUS *is FPT when parameterized by the vertex cover number of the input graph and the solution size. Moreover, the problem does not admit a polynomial kernel when parameterized by the vertex cover number and the solution size unless* $NP \subseteq$ *co-NP/poly.*

We briefly describe the intuition for the exponential kernel with respect to the vertex cover number. Suppose the graph G has a vertex cover X, where $|X| \leq \ell$, and, one may assume, without loss of generality, that $s, t \in X$. Further, for any $Y \subseteq X$, let I_Y denote the set of all vertices in $G \setminus X$ whose neighborhood in X is exactly Y. It is not hard to see that if $|I_Y| > k$, then one might as well "curtail" the set to $k + 1$ vertices without changing the instance. This leads to an exponential kernel in the combined parameter. It is also true that k is bounded, without loss of generality, by ℓ and the size of the common neighbhorhood of s and t to begin with; however, it is unclear if k can always be bounded by some function of the vertex cover size alone. The kernelization lower bound follows from observing the structure of the reduced instance in the reduction used in [4] to prove that the problem is co-W[2]-hard when parameterized by the solution size.

Finally, we present polynomial time algorithms on two restricted cases.

Theorem 4 RENDEZVOUS *admits a polynomial-time algorithm when input graph is a grid or has treewidth at most two.*

Recall that the polynomial time algorithm on the classes of trees, chordal graphs, and P_5-free graphs is obtained by proving that the size of a dynamic separator is same

as that of a separator. In case of grids, we present a winning strategy for Divider for any non-trivial instances. This makes the class of grid graphs a unique graph class in which the problem admits polynomial time algorithm even when dynamic separators can be smaller than separators.

Organization of the Paper After presenting technical preliminaries in Sect. 2, we first describe the proof of Theorem 1 in Sect. 3, along with a separate discussion focused on the intuition for the proof. We present Theorem 2 in Sect. 4. The proof of Theorem 3 can be found in Sect. 5. The polynomial time results are presented in Sect. 6.

2 Preliminaries

For a positive integer q, we denote the set $\{1, 2, \ldots, q\}$ by $[q]$. We use $\mathbb{N}$ to denote the collection of all non-negative integers.

Graph Theory
We use standard graph-theoretic notation, and we refer the reader to [3] for any undefined notation. For an undirected graph G, sets $V(G)$ and $E(G)$ denote its set of vertices and edges, respectively. We denote an edge with two endpoints u, v as uv. Unless otherwise specified, we use n to denote the number of vertices in the input graph G of the problem under consideration. Two vertices u, v in $V(G)$ are *adjacent* if there is an edge uv in G. The *open neighborhood* of a vertex v, denoted by $N_G(v)$, is the set of vertices adjacent to v. The *closed neighborhood* of a vertex v, denoted by $N_G[v]$, is the set $N_G(v) \cup \{v\}$. We say that a vertex u is a *pendant vertex* if $|N_G(v)| = 1$. The *degree* of a vertex v, denoted by $\deg_G(v)$, is equal to the number of vertices in the open neighbourhood of v, i.e., $\deg_G(v) = |N_G(v)|$. We omit the subscript in the notation for neighborhood if the graph under consideration is clear.

For a subset S of $V(G)$, we define $N[S] = \bigcup_{v \in S} N[v]$ and $N(S) = N[S] \backslash S$. For a subset F of edges, we denote by $V(F)$ the collection of endpoints of edges in F. For a subset S of $V(G)$ (resp. a subset F of $E(G)$), we denote the graph obtained by deleting S (resp. deleting F) from G by $G - S$ (resp. by $G - F$). We denote the subgraph of G induced on the set S by $G[S]$.

A graph is *connected* if there is a path between every pair of distinct vertices. A subset $S \subseteq V(G)$ is said to be a *connected set* if $G[S]$ is connected.

A *simple path*, denoted by $P[u, v, d]$, is a non-empty graph G of the form $V(G) = \{u, x_1, \ldots, x_d, v\}$, and $E(G) = \{ux_1, x_1x_2, \ldots, x_{d-1}x_d, x_dv\}$, where $u, v,$ and all x_i's are distinct. The vertices $\{x_1, x_2, \ldots, x_d\}$ are the *internal vertices* of $P[u, v, d]$, and the vertices $\{x_i : \deg(x_i) > 2, i \in [d]\}$, i.e., internal vertices whose degree *in the graph* is strictly greater than 2 are the *branching points* of $P[u, v, d]$. Let $P[u, v, d_1]$ be the path on the vertices $\{u, x_1, \ldots, x_{d_1}, v\}$ and let $P[v, w, d_2]$ be the path on the vertices $\{v, y_1, \ldots, y_{d_2}, w\}$ (with edges as described above). We then use $P[u, v, d_1] \circ P[v, w, d_2]$ to denote the path on $\{u, x_1, \ldots, x_{d_1}, v, y_1, \ldots, y_{d_2}, w\}$, which has $d_1 + d_2 + 1$ internal vertices. While this notation for paths would be slightly non-standard, it turns out to be a convenient choice for describing the structure of the graphs in our proofs.

A set of vertices Y is said to be an *independent set* if no two vertices in Y are adjacent. For a graph G, a set $X \subseteq V(G)$ is said to be a *vertex cover* if $V(G) \backslash X$ is an independent set. A set of vertices Y is said to be a *clique* if any two vertices in Y are adjacent. A vertex cover X is a *minimum vertex cover* if for any other vertex cover Y of G, we have $|X| \leq |Y|$. We denote by $\mathtt{vc}(G)$ the size of a minimum vertex cover of a graph G. For a graph G, a set $X \subseteq V(G)$ is said to be a *feedback vertex set* if $V(G) \setminus X$ is does not contain a cycle. We denote by $\mathtt{fvs}(G)$ the size of a minimum feedback vertex set of a graph G.

A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$, where T is a tree whose every node t is assigned a vertex subset $X_t \subseteq V(G)$, called a *bag*, such that the following conditions hold:

- $\bigcup_{t \in V(T)} X_t = V(G)$. In other words, every vertex of G is in at least one bag.
- For every $uv \in E(G)$, there exists a node t of T such that bag X_t contains both u and v.
- For every $u \in V(G)$, the set $T_u = \{t \in V(T) : u \in X_t\}$, i.e., the set of nodes whose corresponding bags contains u, induces a connected subtree of T.

The *width* of a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ is $max_{t \in V(T)} |X_t| - 1$. The *treewidth* of a graph G, denoted by $\mathtt{tw}(G)$, is the minimum possible width of a tree decomposition of G.

A *path decomposition* of a graph G is a sequence $\mathcal{P} = (X_1, X_2, \ldots, X_r)$ of *bags*, where $X_i \subseteq V(G)$ for each $i \in [r]$, such that the following conditions hold:

- $\bigcup_{i=1}^{r} X_i = V(G)$. In other words, every vertex of G is in at least one bag.
- For every $uv \in E(G)$, there exists $\ell \in [r]$ such that the bag X_ℓ contains both u and v.
- For every $u \in V(G)$, if $u \in X_i \cap X_k$ for some $i \leq k$, then $u \in X_j$ also for each j such that $i \leq j \leq k$. In other words, the indices of the bags containing u form an interval in $[r]$.

The *width* of a path decomposition $(X_1, X_2, \ldots, X_r)$ is $max_{1 \leq i \leq r} |X_i| - 1$. The *pathwidth* of a graph G, denoted by $\mathtt{pw}(G)$, is the minimum possible width of a path decomposition of G.

A $M \times N$ *grid* is the graph G of the form $V(G) = \{(i, j) : i \in [M], j \in [N]\}$, and $E(G) = \{(i, j)(i', j') : |i - i'| + |j - j'| = 1, i, i' \in [M], j, j' \in [N]\}$.

Let X and Y be multisets of vertices of a graph G (i.e., X and Y can contain several copies of the same vertex). We say that X and Y of the same size are *adjacent* if there is a bijective mapping $\alpha : X \to Y$ such that for $x \in X$, either $x = \alpha(x)$ or x and $\alpha(x)$ are adjacent in G.

Parameterized Complexity

An instance of a parameterized problem Π consists of an input I, which is an input of the non-parameterized version of the problem, and an integer k, which is called the *parameter*. A problem Π is said to be *fixed-parameter tractable*, or FPT, if given an instance (I, k) of Π, we can decide whether (I, k) is a YES-instance of Π in time $f(k) \cdot |I|^{\mathcal{O}(1)}$. Here, $f : \mathbb{N} \mapsto \mathbb{N}$ is some computable function depending only on k. Parameterized complexity theory provides tools to rule out the existence of FPT algorithms under plausible complexity-theoretic assumptions. For this, a hierarchy of

parameterized complexity classes $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \cdots \subseteq \mathsf{XP}$ was introduced, and it was conjectured that the inclusions are proper. A parameterized problem is in co-$\mathsf{W}[i]$, $i \in \mathbb{N}$, if its complement is in $\mathsf{W}[i]$, where the complement of a parameterized problem is the parameterized problem resulting from reversing the YES and NO answers. If any co-$\mathsf{W}[i]$-complete problem is fixed-parameter tractable, then co-$\mathsf{W}[i]$=FPT=co-FPT=$\mathsf{W}[i]$ follows, which causes the Exponential Time Hypothesis to fail. Hence co-$\mathsf{W}[i]$-completeness provides strong theoretical evidence that a problem is not fixed-parameter tractable. The most common way to show that it is unlikely that a parameterized problem admits an FPT algorithm is to show that it is $\mathsf{W}[1]$ or $\mathsf{W}[2]$-hard. It is possible to use reductions analogous to the polynomial-time reductions employed in classical complexity. Here, the concept of $\mathsf{W}[1]$-hardness replaces the one of NP-hardness, and we need not only to construct an equivalent instance FPT-time, but also to ensure that the size of the parameter in the new instance depends only on the size of the parameter in the original instance. These types of reductions are called *parameter preserving reductions*. For a detailed introduction to parameterized complexity and related terminologies, we refer the reader to the book by Cygan et al. [2].

A *reduction rule* is a polynomial-time algorithm that takes as input an instance of a problem and outputs another, usually reduced, instance. A reduction rule said to be *applicable* on an instance if the output instance and input instance are different. A reduction rule is *safe* if the input instance is a YES-instance if and only if the output instance is a YES-instance.

A *kernelization* of a parameterized problem Π_1 is a polynomial algorithm that maps each instance (I_1, k_1) of Π_1 to an instance (I_2, k_2) of Π_2 such that (1) (I_1, k_1) is a YES-instance of Π_1 if and only if (I_2, k_2) is a YES-instance of Π_2, and (2) the size of I_2 is bounded by $g(k)$ for a computable function $g(\cdot)$. If $g(\cdot)$ is a polynomial function, then we call it a *polynomial kernel*. It is known that a decidable problem is FPT if and only if it admits a kernel (See, for example, [2, Lemma 2.2]).

Rendezvous Games with Adversaries

Recall that the game is played on a connected graph G, and s and t are initial positions of the agents of Facilitator. Let also k be the number of agents of Divider.

Notice that a placement of the agents of Facilitator is defined by a multiset of two vertices, as R and J can occupy the same vertex. We denote by $\mathcal{F}_G$ the family of all multisets of two vertices. Similarly, a placement of k agents of Divider is defined by a multiset of k vertices, because several agents (of the same player) can occupy the same vertex. Let $\mathcal{D}_G^k$ be the family of all multisets of k vertices. We say that $F \in \mathcal{F}_G$ and $D \in \mathcal{D}_G^k$ are *compatible* if $F \cap D = \phi$. Notice that the number of pairs of compatible $F \in \mathcal{F}_G$ and $D \in \mathcal{D}_G^k$ is $n\binom{n+k-2}{k} + \binom{n}{2}\binom{n+k-3}{k}$. We denote by

$$\mathcal{P}_G^k = \{(F, D) | F \in \mathcal{F}_G, D \in \mathcal{D}_G^k \text{ s.t. } F \text{ and } D \text{ are compatible}\}$$

the set of *positions* in the game.

Formally, a *strategy* of Facilitator for RENDEZVOUS is a function $f : \mathcal{P}_G^k \to \mathcal{F}_G$ that maps $(F, D) \in \mathcal{P}_G^k$ to $F' \in \mathcal{F}_G$ such that F and F' are adjacent and F' is compatible with D. In words, given a position (F, D), Facilitator moves R and J

from F to F' if this is her turn to move. Similarly, a *strategy* of Divider is a function $d : \mathcal{P}_G^k \to \mathcal{D}_G^k$ that maps $(F, D) \in \mathcal{P}_G^k$ to $D' \in \mathcal{F}_G^k$ such that D and D' are adjacent and D' is compatible with F, that is, Divider moves his agents from D to D' if this is his turn to move. To accommodate the initial placement, we extend the definition of d for the pair $(\{s, t\}, \phi)$ and let $d(\{s, t\}, \phi) = D'$, where $D' \in \mathcal{D}_G^k$ is compatible with $\{s, t\}$.

Another variant of the game is when the number of moves of the players is at most some parameter τ. Then Facilitator wins if R and J meet within the first τ moves, and Divider wins otherwise. Thus the problem is:

RENDEZVOUS IN TIME

Input: A graph G with two given vertices s and t, and positive integers k and τ.

Question: Can Facilitator win on G starting from s and t in at most τ steps against Divider with k agents?

Notice that, in the above problem, τ is part of the input. When τ is a fixed constant, this generates a family of problems, one for each different value of τ referred to as τ-RENDEZVOUS IN TIME problem. The definitions of strategies for RENDEZVOUS IN TIME are more complicated, because the decisions of the players also depend on the number of the current step. A *strategy* of Facilitator for RENDEZVOUS is a family of functions $f_i : \mathcal{P}_G^k \to \mathcal{F}_G$ for $i \in \{1, \dots, \tau\}$ such that f_i maps $(F, D) \in \mathcal{P}_G^k$ to $F' \in \mathcal{F}_G$, where F and F' are adjacent and F' is compatible with D. Facilitator uses f_i for the move in the i-th step of the game. A *strategy* of Divider is a family of functions $d_i : \mathcal{P}_G^k \to \mathcal{D}_G^k$ for $i \in \{0, \dots, \tau - 1\}$ such that for $i \in \{1, \dots, \tau - 1\}$, d_i maps $(F, D) \in \mathcal{P}_G^k$ to $D' \in \mathcal{F}_G^k$, where D and D' are adjacent and D' is compatible with F, and d_0 maps $(\{s, t\}, \phi)$ to $D' \in \mathcal{D}_G^k$ compatible with $\{s, t\}$ (slightly abusing notation we do not define d_0 for the elements of $\mathcal{P}_G^k$).

3 para-co-NP-Hardness Parameterized by FVS and Pathwidth

In this section, we prove that RENDEZVOUS is para-co-NP-hard when parameterized by the feedback vertex set number and the pathwidth of the input graph. To do that, we present a parameter preserving reduction from the 3- DIMENSIONAL MATCHING problem, which is known to be NP-hard [6, SP 1]. For notational convenience, we work with the following definition of the problem. An input consists of a universe $\mathcal{U} = \{\alpha, \beta, \gamma\} \times [n]$, a family $\mathcal{F} = \{A_1, A_2, \dots, A_m\}$ of subsets of $\mathcal{U}$ such that for every $j \in [m]$, set $A_j = \{(\alpha, a_1), (\beta, b_1), (\gamma, c_1)\}$ for some $a_1, b_1, c_1 \in [n]$. The goal is to find a subset $\mathcal{F}' \subseteq \mathcal{F}$ that covers $\mathcal{U}$ (and contains exactly n sets).

Reduction

The reduction takes as input an instance $(\mathcal{U}, \mathcal{F})$ of 3- DIMENSIONAL MATCHING and returns an instance (G, s, t, k) of RENDEZVOUS. It defines $M = n^2 + m^2$ where
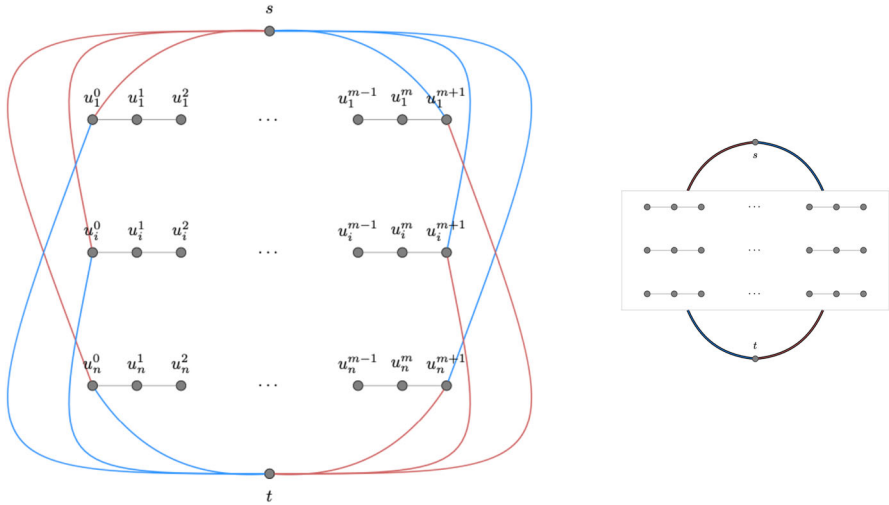
Fig. 1 (Left) The base gadget except the guard vertices g_1 and g_2 (which are not shown for clarity). Each red and blue path has m internal vertices. (Right) Schematic representation (Color figure online)

$n = |\mathcal{U}|/3$ and $m = |\mathcal{F}|$. We construct the graph G as follows: (c.f. Figs. 1, 2, 3 and 4).

The Base Gadget It starts by adding special vertices s and t and two more vertices g_1 and g_2, and makes them common neighbours of s and t. Recall that we use $P[u, v, d]$ to denote a simple path from u to v that contains d many internal vertices.

- For every $i \in [n]$, it adds the following simple paths[1]:

  - $P[u_i^0, u_i^{m+1}, m]$,
  - $P[s, u_i^0, m]$, $P[s, u_i^{m+1}, m]$, $P[t, u_i^0, m]$, and $P[t, u_i^{m+1}, m]$.

See Fig. 1 for an illustration.

Encoding Elements The reduction constructs a symmetric graph to encode elements in $\mathcal{U}$ and has 'left-side' and 'right-side.' It starts by adding vertices $\{\alpha^\ell, \beta^\ell, \gamma^\ell\}$ and $\{\alpha^r, \beta^r, \gamma^r\}$.

- For every $i \in [n]$, it adds six vertices in $\{\alpha_i^\ell, \beta_i^\ell, \gamma_i^\ell\} \cup \{\alpha_i^r, \beta_i^r, \gamma_i^r\}$, and the following simple paths:

  - $P[\alpha^\ell, \alpha_i^\ell, M^2 - M \cdot i]$, $P[\beta^\ell, \beta_i^\ell, M^2 - M \cdot i]$, $P[\gamma^\ell, \gamma_i^\ell, M^2 - M \cdot i]$,
  - $P[\alpha^r, \alpha_i^r, M^2 + M \cdot i]$, $P[\beta^r, \beta_i^r, M^2 + M \cdot i]$, and $P[\gamma^r, \gamma_i^r, M^2 + M \cdot i]$.

  Note that the number of internal vertices in paths from α^ℓ to α_i^ℓ and from α^r to α_i^r, and similar such pairs, are different and depend on i.

- For every $i \in [n]$, it adds six vertices $\{x_i^\ell, y_i^\ell, z_i^\ell\} \cup \{x_i^r, y_i^r, z_i^r\}$, and the following simple paths:

---

[1] We use i as well as a_1, b_1, c_1 as running variables in set $[n]$. We reserve later types of variables for the integer part of elements in sets $\mathcal{F}$.
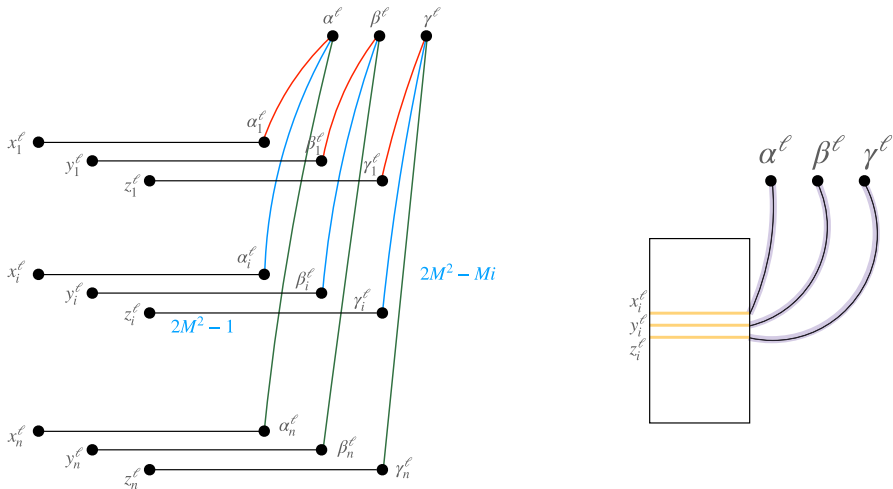
Fig. 2 (Left) The left side of the gadget is added to encode elements in $\mathcal{U}$. The number of internal vertices in each red, blue, and green path depends on i. The number of internal vertices in each yellow shaded path is $2M^2 - 1$. (Right) Schematic representation of the gadget (Color figure online)

- $P[x_i^\ell, \alpha_i^\ell, 2M^2 - 1]$, $P[y_i^\ell, \beta_i^\ell, 2M^2 - 1]$, $P[z_i^\ell, \gamma_i^\ell, 2M^2 - 1]$,
- $P[x_i^r, \alpha_i^r, 2M^2 - 1]$, $P[y_i^r, \beta_i^r, 2M^2 - 1]$, and $P[z_i^r, \gamma_i^r, 2M^2 - 1]$.

See Fig. 2 for an illustration.

Encoding Sets The reduction adds simple paths to encode sets. Consider set A_j for some $j \in [m]$. Suppose the internal vertices of $P[u_i^0, u_i^{m+1}, m]$ are denoted by u_i^j for every $j \in [m]$, and u_i^0 is adjacent with u_i^1 and u_i^{m+1} is adjacent with u_i^m. All the vertices in the jth 'column' corresponds to set A_j. This, however, is not an encoding of set A_j as it does not provide any information about its elements. By the definition of the problem, set A_j has an element of the form (α, a_1). To encode this element, it adds $2n$ many paths connecting j^{th} column to α^ℓ and to α^r. The number of internal vertices in these paths depends on a_1. We encode the remaining two elements in A_j similarly. We formalise this construction as follows:

- For every $j \in [m]$, suppose $A_j = \{(\alpha, a_1), (\beta, b_1), (\gamma, c_1)\}$. Then, for every $i \in [n]$, the reduction adds the following six simple paths:

  - $P[\alpha^\ell, u_i^j, M^2 + M \cdot a_1]$, $P[\beta^\ell, u_i^j, M^2 + M \cdot b_1]$, $P[\gamma^\ell, u_i^j, M^2 + M \cdot c_1]$,
  - $P[\alpha^r, u_i^j, M^2 - M \cdot a_1]$, $P[\beta^r, u_i^j, M^2 - M \cdot b_1]$, and $P[\gamma^r, u_i^j, M^2 - M \cdot c_1]$.

See Fig. 3 for an illustration.

Critical Vertices and Connecting Paths In the last phase of the reduction, it adds critical vertices and connect them to $\{s, t\}$, and also to x-type, y-type, and z-type ends of paths added while encoding elements in $\mathcal{U}$.

- For the special vertex s, it adds critical vertices, say s_α^ℓ, s_β^ℓ and s_γ^ℓ, on the left side.

  - It adds $P[s, s_\alpha^\ell, 2M^2 + 1]$, $P[s, s_\beta^\ell, 2M^2 + 1]$, and $P[s, s_\gamma^\ell, 2M^2 + 1]$.
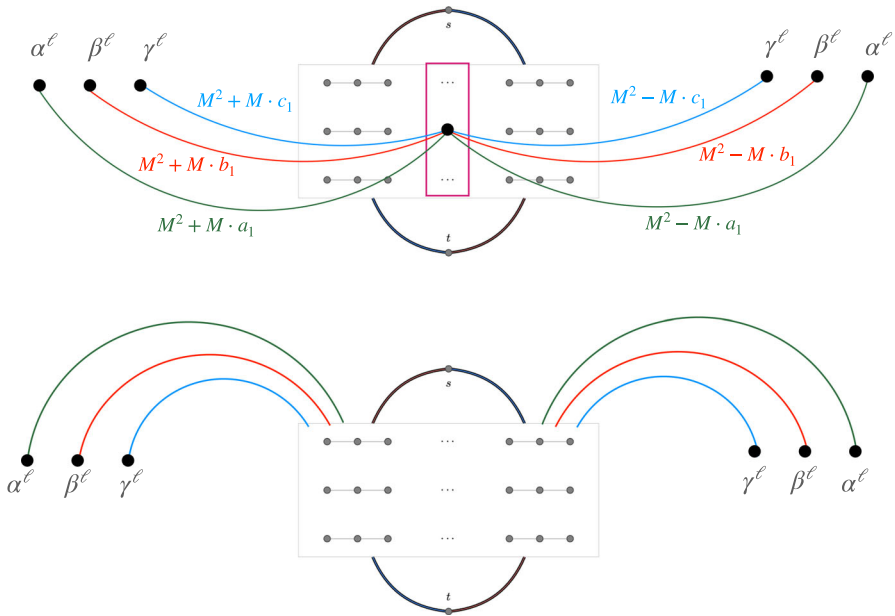
Fig. 3 (Top) Vertices added to encode sets in $\mathcal{F}$. The number of internal vertices in the paths depend on elements in A_j and are denoted next to it. (Bottom) Schematic representation of the gadget used in subsequent figures (Color figure online)

 – For every $i \in [n]$, it adds $P[s_\alpha^\ell, x_i^\ell, 2\,M^2]$, $P[s_\beta^\ell, y_i^\ell, 2\,M^2]$, and $P[s_\gamma^\ell, z_i^\ell, 2M^2]$.

It adds the other critical vertices and paths symmetrically. We present them for the sake of completeness.

For the special vertex s, it adds critical vertices, say s_α^r, s_β^r, and s_γ^r, on the right side.

 – It adds $P[s, s_\alpha^r, 2\,M^2 + 1]$, $P[s, s_\beta^r, 2\,M^2 + 1]$, and $P[s, s_\gamma^r, 2\,M^2 + 1]$.
 – For every $i \in [n]$, it adds $P[s_\alpha^r, x_i^r, 2\,M^2]$, $P[s_\beta^r, y_i^r, 2\,M^2]$, and $P[s_\gamma^r, z_i^r, 2\,M^2]$.

For the special vertex t, it adds critical vertices, say $t_\alpha^\ell, t_\beta^\ell, t_\gamma^\ell$, on the left side.

 – It adds $P[t, t_\alpha^\ell, 2\,M^2 + 1]$, $P[t, t_\beta^\ell, 2\,M^2 + 1]$, and $P[t + t_\gamma^\ell, 2\,M^2 + 1]$.
 – For every $i \in [n]$, it adds $P[t_\alpha^\ell, x_i^\ell, 2\,M^2]$, $P[t_\beta^\ell, y_i^\ell, 2\,M^2]$, $P[t_\gamma^\ell, z_i^\ell, 2M^2]$,

For the special vertex t, it adds critical vertices, say t_α^r, t_β^r, and t_γ^r, on the right side.

 – It adds $P[t, t_\alpha^r, 2\,M^2 + 1]$, $P[t, t_\beta^r, 2\,M^2 + 1]$, and $P[t, t_\gamma^r, 2\,M^2 + 1]$.
 – For every $i \in [n]$, it adds $P[t_\alpha^r, x_i^r, 2\,M^2]$, $P[t_\beta^r, y_i^r, 2\,M^2]$, and $P[t_\gamma^r, z_i^r, 2\,M^2]$.

This completes the construction of the graph G. See Fig. 6 for the overview of the constructed graph. The reduction sets $k = n + 2$ and returns (G, s, t, k) as the reduced instance of RENDEZVOUS.
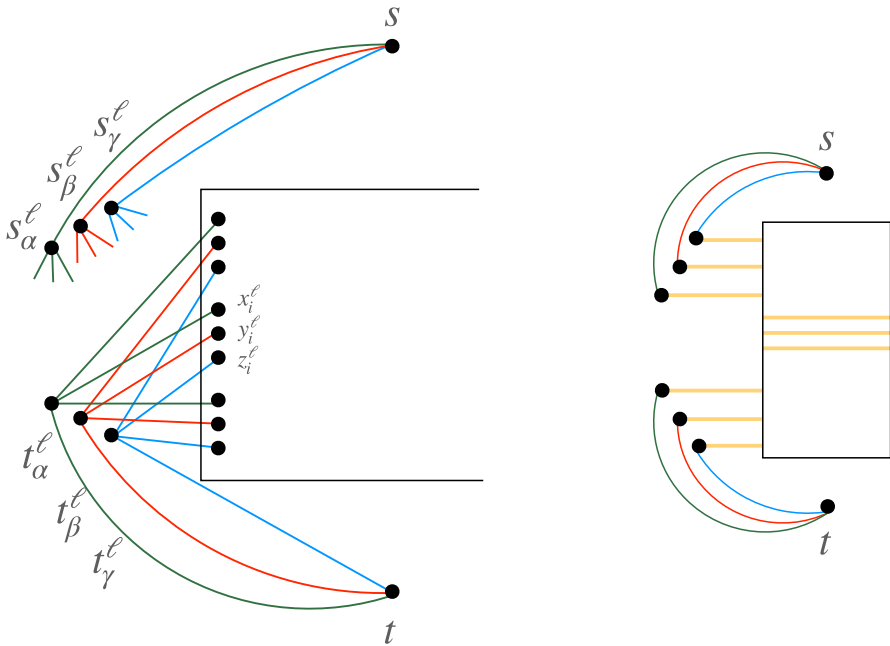
Fig. 4 (Left) Critical vertices added by the reduction. The number of internal vertices in the paths are fixed ($2M^2 + 1$ or $2M^2$). (Right) Schematic representation of the gadget (Color figure online)

Intuition for the Correctness We present an intuition for the correctness of the reduction in the reverse direction. In other words, we state how the initial positions of the Divider's agent correspond to sets and the elements they can cover. We start with determining the possible initial positions.

As g_1, g_2 are common neighbors of s and t, Divider needs to put two agents on g_1 and g_2. For the remaining n agents, consider the paths $P[s, u_i^0, m]$ and $P[u_i^0, t, m]$ or $P[s, u_i^{m+1}, m]$ and $P[u_i^{m+1}, t, m]$ for every $i \in [n]$. Facilitator can move both Romeo and Juliet to u_i^0 or u_i^{m+1} in m steps. Hence, Divider needs to place remaining n agents at the positions that are at distance at most m simultaneously from u_i^0 and u_i^{m+1}. We ensure that he needs to place an agent on an internal vertex of $P[u_i^0, u_i^{m+1}, m]$ for every $i \in [n]$. This will correspond to selecting a set in $\mathcal{F}$ in a solution. Formally, an agent at u_i^j for some $j \in [m]$ corresponds to selecting A_j in the cover of $\mathcal{U}$. As there are n 'rows', this will correspond to selecting n (different) sets from $\mathcal{F}$. Hence, the initial position of the Divider's agent will correspond to a collection of sets in $\mathcal{F}$.

Suppose for every $i \in [n]$, vertices in $\{x_i^\ell, x_i^r\}$ correspond to element $(\alpha, i) \in \mathcal{U}$. Similarly, vertices in $\{y_i^\ell, y_i^r\}$ correspond to (β, i), and vertices in $\{z_i^\ell, z_i^r\}$ correspond to (γ, i). We say (α, i) is covered if Divider can prevent Facilitator from moving both Romeo and Juliet at x_i^ℓ as well as at x_i^r.

From the Facilitator's preservative, she has $6n$ possible meeting points of the above form. She can make these choices in two phases. In the first phase, she can decide to move Romeo towards one of the six vertices in $\{s_\alpha^\ell, s_\beta^\ell, s_\gamma^\ell\} \cup \{s_\alpha^r, s_\beta^r, s_\gamma^r\}$. To win, she
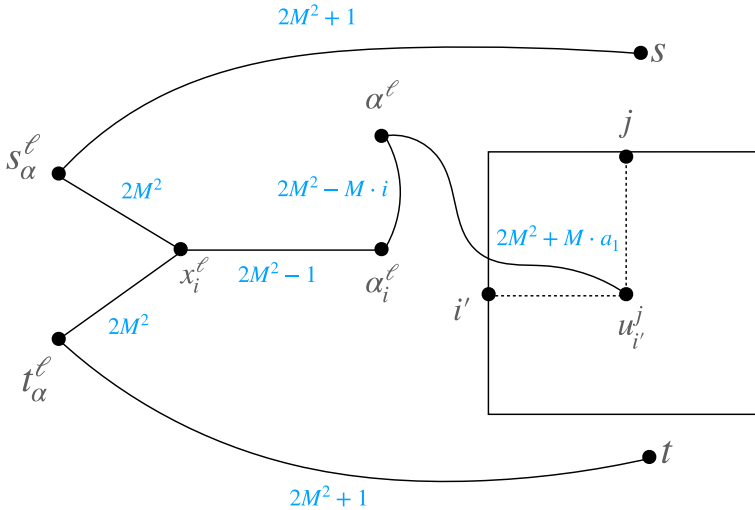
Fig. 5 Vertices mentioned while presenting the intuition. The vertices near the paths indicate the number of internal vertices. Set A_j contains (α, a_1)

will have to move Juliet towards the corresponding vertices with respect to t. Suppose she moves Romeo towards s_α^ℓ and Juliet towards t_α^ℓ, i.e., Romeo along $P[s, s_\alpha^\ell, 2M^2 + 1]$ and Juliet along $P[t, t_\alpha^\ell, 2M^2 + 1]$. She can move Romeo at s_α^ℓ and Juliet at t_α^ℓ in $2M^2 + 2$ steps. At this point, she can make one of the n choices and decide to move both Romeo and Juliet towards x_i^ℓ for some $i \in [n]$, i.e., Romeo along $P[s_\alpha^\ell, x_i^\ell, 2M^2]$ and Juliet along $P[t_\alpha^\ell, x_i^\ell, 2M^2]$ for some $i \in [n]$. See Fig. 5 for relevant vertices.

From the Divider's perspective, he can see the first choice made by Facilitator. However, he has no information about her second choice until next $2M + 2$ steps, i.e., until she moves Romeo at s_α^ℓ and Juliet at t_α^ℓ. Note that Facilitator can move Romeo from s_α^ℓ to x_i^ℓ and Juliet from t_α^ℓ to x_i^ℓ in $2M^2 + 1$ steps. Divider can move an agent from α_i^ℓ to x_i^ℓ in $2M^2$ steps. Considering the initial positions of agents, he needs to ensure that one of his agents is present on α_i^ℓ for *every* $i \in [n]$ in $2M^2 + 2$ steps. For every $i \in [n]$, he needs to place an agent at $u_{i'}^j$ for some $j \in [m]$ and $i' \in [n]$ that he can move it to α_i^ℓ in $2M^2 + 2$ steps. We remark that i may not be equal to i'.

The only feasible way to do so is by moving the agent from $u_{i'}^j$ to α^ℓ and then move it from α^ℓ to α_i^ℓ. Suppose $(\alpha, a_1) \in A_j$ for some $a_1 \in [n]$. Recall that the number of internal vertices of path from $u_{i'}^j$ to α^ℓ is $M^2 + M \cdot a_1$ where as that of the path from α^ℓ to α_i^ℓ is $M^2 - M \cdot i$. Formally, the number of internal vertices in the path $P[u_{i'}^j, \alpha^\ell, M^2 + M \cdot a_1] \circ P[\alpha^\ell, \alpha_i^\ell, M^2 - M \cdot i]$ is $(M^2 + M \cdot a_1) + 1 + (M^2 - M \cdot i) = 2M^2 + 1 + M^2 \cdot (a_1 - i)$. This implies that Divider can move an agent from $u_{i'}^j$ to α_i^ℓ in $2M^2 + 2 + M^2 \cdot (a_1 - i)$ steps. Hence, for *every* $i \in [n]$, Divider should place an agent at $u_{i'}^j$ for some $j \in [m]$ and $i' \in [n]$ such that for $(\alpha, a_1) \in A_j$, we have $a_1 \le i$. Using identical arguments and considering the number of internal vertices on
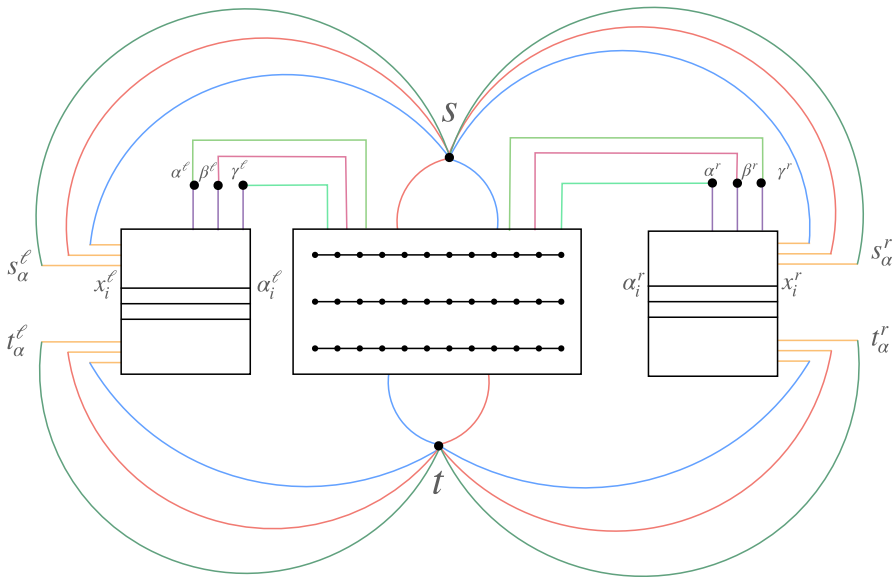
Fig. 6 Overview of the reduction in Sect. 3

the right side, we prove that for *every* $i \in [n]$, he needs to place an agent at $u_{i''}^{j'}$ for some $j' \in [m]$ and $i'' \in [n]$ such that for $(\alpha, a_2) \in A_j$, we have $a_2 \geq i$. Combining these two arguments, Divider needs to place an agent at $u_{i'}^{j}$ such that for $(\alpha, a_1) \in A_j$, we have $a_1 = i$. Moving the agent from this position will prevent Facilitator from moving both Romeo and Juliet at x_i^{ℓ} and x_i^{r}. This corresponds to selecting a set from $\mathcal{F}$ to cover the element $(\alpha, i) \in \mathcal{U}$. This concludes the intuition for the correctness of the reduction.

We now present arguments formalizing the ideas described above.

Lemma 5 *If $(\mathcal{U}, \mathcal{F})$ is a* YES-*instance of* 3- DIMENSIONAL MATCHING, *then $(G, s, t, n+2)$ is a* NO-*instance of* RENDEZVOUS.

Proof We show that if $(\mathcal{U}, \mathcal{F})$ is a YES-instance of 3- DIMENSIONAL MATCHING, then Divider with $n + 2$ agents can win in Rendezvous Game with Adversaries. Recall that $\mathcal{U} = \{\alpha, \beta, \gamma\} \times [n]$, and $\mathcal{F} = \{A_1, A_2, \ldots, A_m\}$ such that $A_j = \{(\alpha, a_1), (\beta, b_1), (\gamma, c_1)\}$ for some $a_1, b_1, c_1 \in [n]$. Let $\mathcal{F}' = \{A_{j_1}, A_{j_2}, \ldots, A_{j_n}\} \subseteq \mathcal{F}$ be the solution for 3- DIMENSIONAL MATCHING such that $j_1 < j_2 < \cdots < j_n$.

Since $\mathcal{F}'$ covers every element of $\mathcal{U}$, each element of $\mathcal{U}$ appears in exactly one of the set in $\mathcal{F}'$.

We describe a winning strategy for Divider with the agents $D_1, D_2, \ldots, D_{n+2}$. Initially, he puts D_i in the vertex $u_i^{j_i}$, for every $i \in [n]$, and D_{n+1} and D_{n+2} in g_1 and g_2 respectively. He does not move agents $D_1, \ldots, D_{n+2}$, until Facilitator moves Romeo or Juliet from s or t, respectively. Suppose Facilitator moves Romeo from s (she may or may not move Juliet from t). By the construction, she can move Romeo either on the paths $P[s, u_i^0, m], P[s, u_i^{m+1}, m]$ for some $i \in [n]$ or on one of the following

paths: $P[s, s_\alpha^\ell, 2\,M^2+1]$, $P[s, s_\beta^\ell, 2\,M^2+1]$, $P[s, s_\gamma^\ell, 2\,M^2+1]$, $P[s, s_\alpha^r, 2\,M^2+1]$, $P[s, s_\beta^r, 2\,M^2+1]$, or $P[s, s_\gamma^r, 2\,M^2+1]$.

Suppose Facilitator moves Romeo from s to a vertex on the path $P[s, u_i^0, m]$ for some $i \in [n]$. Divider moves D_{n+1} from g_1 to s and then towards u_i^0 as she moves Romeo towards u_i^0. He also moves D_i to u_i^0 in at most j_i steps along the path $P[u_i^0, u_i^{m+1}, m]$. Facilitator needs at least $m+1$ steps to move both Romeo and Juliet in u_i^0 starting from s and t respectively. As $j_i \le m$, Divider can move D_i to u_i^0 before Facilitator can move both Romeo and Juliet to u_i^0. Hence, he can block Romeo by D_i and D_{n+1} on the path $P[s, u_i^0, m]$.

Divider keeps moving D_i and D_{n+1} towards Romeo's position and in at most $j_i + m - 1$ steps Facilitator can not move Romeo. This implies Divider wins by keeping Romeo in its current position with its neighbors occupied by D_i and D_{n+1}. The argument also follows when Facilitator moves Romeo from s to a vertex on the path $P[s, u_i^{m+1}, m]$ for some $i \in [n]$ since Divider can move D_i to u_i^{m+1} in at most $m - j_i + 1 \ (\le m)$ steps.

Suppose Facilitator moves Romeo from s to a vertex on the path $P[s, s_\alpha^\ell, 2M^2+1]$. In this case, the remaining strategy for Divider is guided by the α-type elements in the universe and in sets. Recall that each element of $\mathcal{U}$ appears in exactly one of the set in $\mathcal{F}'$. This implies that for every $a_1 \in [n]$, there is a unique set in $\mathcal{F}'$ that contains element (α, a_1). We define function $\psi_\alpha : [n] \mapsto [n]$ with respect to α. More formally, $\psi_\alpha(a_1) = i$ if (α, a_1) is contained in A_{j_i} in $\mathcal{F}'$. By the construction, Divider can move agent D_i from $u_i^{j_i}$ to $\alpha_{a_1}^\ell$ in at most $(M^2 - M \cdot a_1) + 1 + (M^2 + M \cdot a_1) + 1$ steps through the path $P[u_i^{j_i}, \alpha^\ell, M^2 - M \cdot a_1] \circ P[\alpha^\ell, \alpha_{a_1}^\ell, M^2 + M \cdot a_1]$. Hence, Divider can move D_i to $\alpha_{a_1}^\ell$ in at most $2M^2 + 2$ steps. Hence, for every $a_1 \in [n]$, Divider can move the agent at $u_i^{j_i}$ to $\alpha_{a_1}^\ell$ in $2M^2 + 2$ steps where $i = \psi_\alpha(a_1)$. For notational convenience, we re-write the previous statement while changing the running variable from a_1 to i. Divider can move agents $D_1, D_2, \ldots, D_n$ in $2M^2 + 2$ steps such that for every $i \in [n]$, one of its agent is present in α_i^ℓ. As in the previous case, he can move D_{n+1} from g_1 to s and then keep moving towards s_α^ℓ as Facilitator moves Romeo towards s_α^ℓ. He can move D_{n+2} in a similar manner with respect to Juliet.

After the first move, Facilitator can not move back Romeo and Juliet towards s, t respectively, because of the agents D_{n+1} and D_{n+2}. However, she will need at least $2M^2 + 2$ steps to move Romeo to s_α^ℓ and Juliet to t_α^ℓ starting from s and t. If she moves Romeo to s_α^ℓ and Juliet to t_α^ℓ, then she can only move Romeo and Juliet towards x_i^ℓ for some $i \in [n]$. However, she will need at least $2M^2 + 1$ steps to move them along the path $P[s_\alpha^\ell, x_i^\ell, 2\,M^2]$ and $P[t_\alpha^\ell, x_i^\ell, 2\,M^2]$, respectively.

Divider can move his agent from α_r^ℓ to x_i^ℓ in at most $2M^2 \ (< 2M^2 + 1)$ steps along the path $P[\alpha_i^\ell, x_i^\ell, 2M^2 - 1]$ for every $i \in [n]$. Hence, he can place his agent at x_i^ℓ before Facilitator can move Romeo and Juliet to that place. He can keep moving D_{n+1} towards the position of Romeo and similarly D_{n+2} towards the position of Juliet. Hence, he can block Romeo by the agents at x_i^ℓ for every $i \in [n]$ and D_{n+1} either on the path $P[s, s_\alpha^\ell, 2M^2+1]$ if Facilitator never moves Romeo at s_α^ℓ or otherwise on the path $P[s_\alpha^\ell, x_i^\ell, 2\,M^2]$ for some $i \in [n]$.

This implies if Facilitator moves Romeo from s to a vertex on the path $P[s, s_\alpha^\ell, 2M^2 + 1]$, then Divider has a winning strategy. It is easy to see that the similar arguments follows if Facilitator moves Romeo from s to a vertex on the path $P[s, s_\beta^\ell, 2M^2 + 1]$, $P[s, s_\gamma^\ell, 2M^2 + 1]$, $P[s, s_\alpha^r, 2M^2 + 1]$, $P[s, s_\beta^r, 2M^2 + 1]$, or $P[s, s_\gamma^r, 2M^2 + 1]$. Hence, if $(\mathcal{U}, \mathcal{F})$ is a YES-instance of 3- DIMENSIONAL MATCH-ING, then Divider with $n + 2$ agents can win in Rendezvous Game with Adversaries, i.e., $(G, s, t, n + 2)$ is a NO-instance of RENDEZVOUS. $\qquad\square$

Lemma 6 *If $(\mathcal{U}, \mathcal{F})$ is a NO-instance of* 3- DIMENSIONAL MATCHING, *then $(G, s, t, n + 2)$ is a* YES-*instance of* RENDEZVOUS.

Proof We show that if $(\mathcal{U}, \mathcal{F})$ is a NO-instance of 3- DIMENSIONAL MATCHING, then Facilitator has a winning strategy in at most $4M^2 + 3$ steps against Divider with $n + 2$ agents.

We first consider two simple cases where Facilitator has an easy winning strategy. First, consider the case when Divider does not place his agents at g_1 or g_2. Then, she can move Romeo and Juliet there and win in one step. Second, consider the case when there is $i \in [n]$ such that none of Divider's agents is within distance m from u_i^0 or from u_i^{m+1}. In the first sub-case, she can move Romeo and Juliet to u_i^0 in $m + 1$ steps through the paths $P[s, u_i^0, m]$ and $P[t, u_i^0, m]$, respectively, and win. Similarly, in the second sub-case she can move Romeo and Juliet to u_i^{m+1} in $m + 1$ steps through the paths $P[s, u_i^{m+1}, m]$ and $P[t, u_i^{m+1}, m]$, respectively, and win.

In what follows, we suppose that Divider places D_{n+1} at g_1 and D_{n+2} at g_2. More-over, for every $i \in [n]$, there is a Divider's agent within distance m from u_i^0 and within distance m from u_i^{m+1}. By the construction, the choice of M, and the fact that Divider can not place an agent at s or t, a single Divider's agent cannot be within distance m from both u_i^0 and u_j^0, or from u_i^{m+1} and u_j^{m+1}, or from u_i^0 and u_j^{m+1}, for $i \neq j \in [n]$. As Divider has n remaining agents, for every $i \in [n]$, there must be an agent, say D_i, within distance m from both u_i^0 and u_i^{m+1}. This is possible only when for every $i \in [n]$, D_i is on one of the internal vertices of the path $P[u_i^0, u_i^{m+1}, m]$ or on one of the paths joining u_i^j to vertex in $\{\alpha^\ell, \beta^\ell, \gamma^\ell\} \cup \{\alpha^r, \beta^r, \gamma^r\}$, for some $j \in [m]$. Suppose $\phi : [n] \mapsto [m]$ is the mapping corresponding to the initial position of the Divider's agents. Formally, for every $i \in [n]$, Divider places agent D_i either on the internal vertex $u_i^{\phi(i)}$ or on the path joining $u_i^{\phi(i)}$ to vertex in $\{\alpha^\ell, \beta^\ell, \gamma^\ell\} \cup \{\alpha^r, \beta^r, \gamma^r\}$. While defining the mapping, the first condition is prioritized.

We now define the Facilitator's strategy. Considering $\mathcal{U} = \{\alpha, \beta, \gamma\} \times [n]$ as the universe, she constructs a collection $\mathcal{F}$ of subsets of $\mathcal{U}$ such that for every $j \in [m]$, set $A_j = \{(\alpha, a_1), (\beta, b_1), (\gamma, c_1)\}$ for some $a_1, b_1, c_1 \in [n]$. Alternately, she reverse-engineers the process used by the reductions to encode sets. She also constructs a subset $\mathcal{F}'$ of $\mathcal{F}$ by considering the initial positions of agents $D_1, D_2, \ldots, D_n$. Formally, she includes $A_{\phi(i)}$ in $\mathcal{F}'$ for every $i \in [n]$, i.e. $\mathcal{F}' = \{A_{\phi(1)}, A_{\phi(2)}, \ldots, A_{\phi(n)}\}$. Note that $\mathcal{F}'$ contains exactly n elements.[2]

---

[2] It is tempting to imagine that if (α, a_1) is not covered by $\mathcal{F}'$ than Facilitator can move Romeo and Juliet at $x_{a_1}^\ell$ or at $x_{a_1}^r$. However, we argue that Facilitator can move Romeo and Juliet at $x_{a_2}^\ell$ for some $a_2 \leq a_1$ or at $x_{a_3}^r$ for some $a_3 \geq a_1$.

For every $i \in [n]$, define $N(\alpha, \leq i)$ as the number of sets $A_{\phi(i')}$ in $\mathcal{F}'$ such that the α-element $(\alpha, a_1) \in A_{\phi(i')}$, $a_1 \leq i$. Also, define $N(\alpha, \geq i)$ as the number of sets $A_{\phi(i')}$ in $\mathcal{F}'$ such that for the α-element $(\alpha, a_1) \in A_{\phi(i')}$, $a_1 \geq i$. It similarly defines and computes $N(\beta, \leq i)$, $N(\beta, \geq i)$, $N(\gamma, \leq i)$, and $N(\gamma, \geq i)$. It then determines whether the following statements are `True`.

1. For all $i \in [n]$, $N(\alpha, \leq i) \leq i$ and $N(\alpha, \geq i) \leq n + 1 - i$.
2. For all $i \in [n]$, $N(\beta, \leq i) \leq i$ and $N(\beta, \geq i) \leq n + 1 - i$.
3. For all $i \in [n]$, $N(\gamma, \leq i) \leq i$ and $N(\gamma, \geq i) \leq n + 1 - i$.

Facilitator has to make two critical choices. Her first critical choice is at the first step where she has to decide about moving Romeo towards $\{s_\alpha^\ell, s_\alpha^r\}$, $\{s_\beta^\ell, s_\beta^r\}$, or $\{s_\gamma^\ell, s_\gamma^r\}$. This choice depends on which of the above statement is *false*. If the first statement is false, she narrows down her choice of moving Romeo either to s_α^ℓ or s_β^ℓ. Suppose the first statement is false because of the first inequality for some i. In that case, she moves Romeo towards the right side, i.e., towards s_α^r; otherwise, she moves Romeo towards the left side, i.e., towards s_α^ℓ. She moves Juliet towards the corresponding vertex with respect to t, i.e., towards t_α^r and t_α^ℓ in the first and the second case, respectively.

To explain her second choice, suppose, without loss of generality, that the first statement is false because of its first inequality. She then moves Romeo from s to s_α^r and Juliet from t to t_α^r in $2M^2 + 2$ steps. For her second choice, she finds $i \in [n]$ such that Divider's agent has not been across α_i^ℓ since the game started. She then moves Romeo and Juliet at x_i^ℓ in $2M^2 + 1$ additional moves and wins the game. She uses a similar strategy in other cases. We remark that the initial positions of Divider's agents are 'close' to the base gadget. For Divider to move his agent from their initial positions to say vertices like α_i^ℓ or α_i^r, he needs to move them via α^ℓ or α^r, respectively.

To argue that this is indeed a winning strategy for Facilitator, we first argue that for any initial positions of the Divider's agents, at least one of the three statements above is false. Suppose $a_1 \in [n]$ is the integer such that (α, a_1) does not appear in any sets in $\mathcal{F}'$. This implies for every $a_1' \in [a_1]$, element (α, a_1') appears in at least one set in $\mathcal{F}'$. Suppose every (α, a_1') appears in exactly one set in $\mathcal{F}'$. As $\mathcal{F}'$ contains n sets, each set contains an α-element, and there $(a_1 - 1)$ sets that contains α-element (α, i) such that $i < a_1$, we can conclude the following. There are $n - (a_1 - 1)$ sets that contains α-element (α, i) such that $i \geq a_1 + 1$. This implies that the second inequality in the first statement is false for $i = a_1 + 1$. Consider the case when there is $a_1' \in [a_1]$ such that (α, a_1') appears in at least two sets in $\mathcal{F}'$. Suppose a_1' is the smallest such integer. As $a_1' < a_1$, there are at least $a_1' + 1$ many sets in $\mathcal{F}'$ that contains α-element (α, i) such that $i \leq a_1'$. Hence, in either case, the first statement is false. Conversely, if the first statement is `True`, then (α, a_1) is present in at least one set in $\mathcal{F}'$ for every $a_1 \in [n]$. This implies if none of the three statements is false, every element in $\mathcal{U}$ appears in some set in $\mathcal{F}'$. This, however, contradicts the fact that $(\mathcal{U}, \mathcal{F}')$ is a No-instance. Hence, for any initial positions of Divider's agents, at least one of the three sentences is false.

This allows Facilitator to make her first choice. It remains to argue that there exists $i \in [n]$ with desired properties for her second choice. Towards that, we first identify the conditions in which Divider can move the agent $D_{i'}$ to α_i^ℓ in $2M^2 + 2$ steps for two indices $i, i' \in [n]$. Suppose Divider initially places the agent $D_{i'}$ at distance $p_{i'}$ from the vertex $u_{i'}^{\phi(i')}$. As $D_{i'}$ must be within distance m from both $u_{i'}^0$ and $u_{i'}^{m+1}$, we can

conclude that $p_{i'}$ is at most $m/2$. Note that $p_{i'}$ can be zero. Suppose $(\alpha, a_1) \in A_{\phi(i')}$ for some $a_1 \in [n]$. Recall that the number of internal vertices of path from $u_{i'}^{\phi(i')}$ to α^ℓ is $M^2 + M \cdot a_1$ where as that of the path from α^ℓ to α_i^ℓ is $M^2 - M \cdot i$. Hence, Divider can move the agent $D_{i'}$ to α_i^ℓ in $2 M^2 + 2 + M \cdot (a_1 - i) - p_{i'}$ steps if $D_{i'}$ is on the path $P[u_{i'}^{\phi(i')}, \alpha_\ell, M^2 + M \cdot a_1]$, and in $2 M^2 + 2 + M \cdot (a_1 - i) + p_{i'}$ steps otherwise. Hence, $D_{i'}$ can only reach to α_i^ℓ within $2M^2 + 2$ steps if $a_1 \leq i$, for any $i \in [n]$, since $p_{i'} \ll M$. Using similar arguments, $D_{i'}$ can only reach to α_i^r within $2M^2 + 2$ steps if $a_1 \geq i$, for any $i \in [n]$. Note that if $a_1 \leq i$ then Divider can not move $D_{i'}$ to α_{i+1}^r within $2M^2 + 2$ steps as $2 M^2 + 2 + M \cdot (i + 1 - a_1) \pm p_{i'} \geq 2 M^2 + 2 + M - m/2 > 2 M^2 + 2$. Moreover, as the number of internal points between the paths from α^r to α_i^r is $M^2 + M \cdot i$, if Divider can not move $D_{i'}$ to α_{i+1}^r within $2 M^2 + 2$ then it can not move it to $\alpha_{i°+1}^r$ for any $i° \geq i$.

We now argue about the second critical choice of Facilitator. Suppose the facilitator moves Romeo from s to s_α^r and moves Juliet from t to t_α^r according to the strategy. This implies that the first inequality in the first statement is false for some $i \in [n]$, i.e. $N(\alpha, \leq i) = q > i$. By the definition of $N(\alpha, \leq i)$, there are q sets in $\mathcal{F}'$ that contains α-element (α, a_1) such that $a_1 \leq i$. As discussed in the previous paragraphs, if $(\alpha, a_1) \in A_{\phi(i')}$ for some $i' \in [n]$, then Divider can not move $D_{i'}$ from its initial position to α_{i+1}^r in $2M^2 + 2$ steps. This statement is `True` for $q > i$ many agents. Consider the set $\{\alpha_{i+1}^r, \alpha_{i+2}^r, \ldots, \alpha_n^r\}$ of $n - i$ vertices. Divider can move at most $n - q$ ($< n - i$) agents to these vertices in $2M^2 + 2$ steps. Hence, there exists $i° \in [n] \setminus [i]$ such that none of the Divider's agent has reached $\alpha_{i°}^r$.

Suppose for some $i \in [n]$, none of the Divider's agents can reach α_i^r or any vertex on path $P[x_i^r, \alpha_i^r, 2 M^2 - 1]$ after $2M^2 + 2$ steps from the start. Then, every Divider's agent will be at distance at least $2M^2 + 1$ from x_i^r after $2M^2 + 2$ steps from start. Note that Facilitator can move Romeo from s to s_α^r and Juliet from t to t_α^r in $2M^2 + 2$ steps. She can then move Romeo and Juliet to x_i^r in $2M^2 + 1$ steps along the path $P[s_\alpha^r, x_i^r, 2 M^2]$ and $P[t_\alpha^r, x_i^r, 2 M^2]$, respectively. Since Facilitator takes the first turn, she can move Romeo and Juliet to x_i^r before the Divider's agents and win in $4M^2 + 3$ steps.

A similar argument follows when the second inequality of the first statement is false, then Romeo and Juliet can meet at x_i^ℓ for some $i \in [n]$. Similarly, when second or third statement is false, then also Romeo and Juliet will be able to meet at $\{y_i^\ell, y_i^r\}$ or $\{z_i^\ell, z_i^r\}$ for some $i \in [n]$, respectively. As mentioned earlier, Facilitator will decide the 'left' or 'right' vertex based on which inequality of the statement is false.

This implies that if $(\mathcal{U}, \mathcal{F})$ is a No-instance of 3- DIMENSIONAL MATCHING, then Facilitator wins in at most $4M^2 + 3$ steps against Divider with $n + 2$ agents, i.e., $(G, s, t, n + 2)$ is a YES-instance of RENDEZVOUS. □

We now argue that the pathwidth of the reduced instance is at most 16. Consider set $S := \{s, t\} \cup \{s_\alpha^\ell, s_\beta^\ell, s_\gamma^\ell\} \cup \{t_\alpha^\ell, t_\beta^\ell, t_\gamma^\ell\} \cup \{\alpha^\ell, \beta^\ell, \gamma^\ell\} \cup \{\alpha^r, \beta^r, \gamma^r\}$ in G. It is easy to verify that $G - S$ is a forest, i.e., the feedback vertex set number of G is at most 14. Moreover, every connected component of $G - S$ is either a path or a subdivided caterpillar. The paths correspond to the paths added while encoding elements in $\mathcal{U}$

or while adding the critical paths. The subdivided caterpillars correspond to the base gadgets, and the path added while encoding sets in $\mathcal{F}'$. Note that the *spine* of the caterpillar is the path $P[u_i^0, u_i^{m+1}, m]$ for some $i \in [n]$ added as a part of base gadget. This implies that the pathwidth of the resulting graph is at most 16.

To summarize, Lemmas 5, 6, and the fact that the reduction can be completed in time polynomial in the size of the input imply Theorem 1 which we restate here.

Theorem 1 RENDEZVOUS *is* co-NP-hard *even when restricted to:*

- *graphs whose feedback vertex set number is at most 14, or*
- *graphs whose pathwidth is at most 16.*

In particular, RENDEZVOUS *is* para-co-NP-hard *parameterized by treewidth.*

4 co-W[1]-Hardness Parameterized by FVS, Pathwidth, and the Solution Size

In this section, we prove Theorem 2 that states RENDEZVOUS is co-W[1]-hard when parameterized by the feedback vertex set number or pathwidth and the solution size. To do that, we present a parameter preserving reduction from the (MONOTONE) NAE-INTEGER- 3- SAT problem. For notational convenience, we work with the following definition of the problem. An input consists of variables $\mathcal{X} = \{x_1, \ldots, x_n\}$ that each take a value in the domain $\mathcal{D} = \{1, \ldots, d^\star\}$ and clauses $\mathcal{C} = \{C_1, \ldots, C_m\}$ of the form

$$\text{NAE}\left(x_{i_1} \le d_1, x_{i_2} \le d_2, x_{i_3} \le d_3\right),$$

where $d_1, d_2, d_3 \in [d^\star]$. Such a clause is satisfied if not all three inequalities are True and not all are False (i.e., they are "not all equal"). The goal is to find an assignment of the variables that satisfies all given clauses. Bringmann et. al. [1] proved that (MONOTONE) NAE- INTEGER- 3- SAT is W[1]-hard when parameterized by the number of variables.

Reduction
The reduction takes as input an instance $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ of (MONOTONE) NAE-INTEGER- 3- SAT and returns an instance (G, s, t, k) of RENDEZVOUS. We construct the graph G as follows: (See Fig. 7 for the overview of the constructed graph.)

The Variable Gadget Recall that we use $P[u, v, d]$ to denote a simple path from u to v that contains d many internal vertices. For every $i \in [n]$, it adds a simple path $P[u_i^0, u_i^{d^\star+1}, d^\star]$. Suppose the internal vertices of $P[u_i^0, u_i^{d^\star+1}, d^\star]$ are denoted by u_i^d for every $d \in [d^\star]$, and u_i^0 is adjacent with u_i^1 and $u_i^{d^\star+1}$ is adjacent with $u_i^{d^\star}$.

The Clause Gadget For every $j \in [m]$, the reduction adds two vertices c_j^ℓ and c_j^r. Suppose $C_j = \text{NAE}\left(x_{i_1} \le d_1, x_{i_2} \le d_2, x_{i_3} \le d_3\right)$ for some $j \in [m]$. To encode the inequality $x_{i_1} \le d_1$, the reduction adds simple paths $P[c_j^\ell, u_{i_1}^0, 2d^\star - d_1]$ and $P[c_j^r, u_{i_1}^{d^\star+1}, d^\star + d_1]$. It encodes the other two inequalities similarly. We highlight
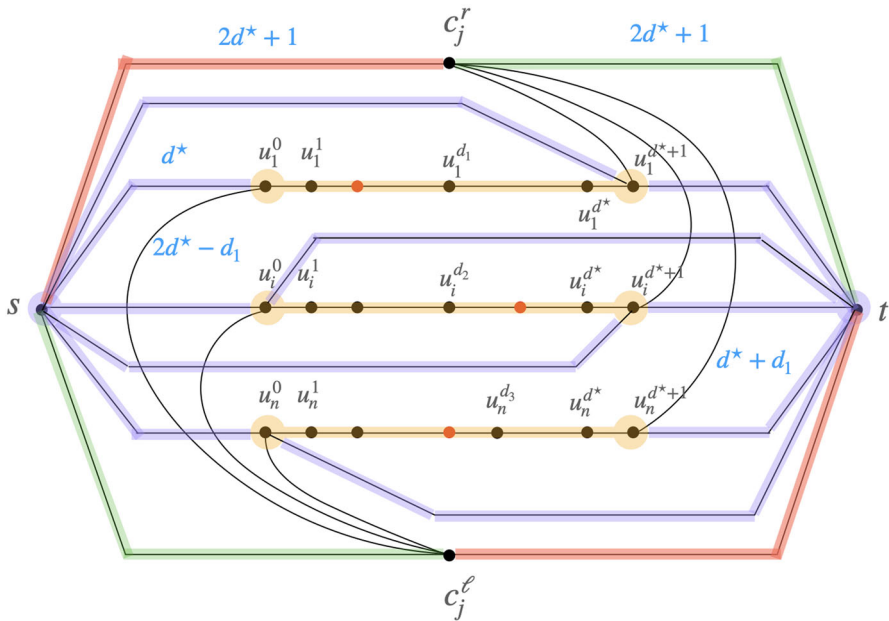
Fig. 7 The reduction adds a yellow shaded path for each variable. Each yellow, purple, or blue shaded path has $d^\star$ many internal vertices. The green and red shaded paths have $2d^\star + 1$ many internal vertices. The number of internal vertices in the remaining path in the figure depends on constants in the clause they are encoding. Note that vertices g_1, g_2 and paths $P[s, u_n^{d^\star+1}, d^\star]$, $P[t, u_1^0, d^\star]$ are not shown in the figure for clarity. The red vertices denote the positions of the agents (Color figure online)

that the number of internal vertices in these simple paths depends on the constant in the inequalities they encode.

Critical Vertices and Connecting Paths The reduction adds special vertices s and t and two more vertices g_1 and g_2, and makes them common neighbours of s and t.

- For every $i \in [n]$, it adds the following simple paths:
  - $P[s, u_i^0, d^\star]$, $P[s, u_i^{d^\star+1}, d^\star]$,
  - $P[t, u_i^0, d^\star]$, $P[t, u_i^{d^\star+1}, d^\star]$.
- For every $j \in [m]$, it adds the following simple paths:
  - $P[s, c_j^\ell, 2d^\star + 1]$, $P[s, c_j^r, 2d^\star + 1]$,
  - $P[t, c_j^\ell, 2d^\star + 1]$, $P[t, c_j^r, 2d^\star + 1]$.

This completes the construction of the graph G. The reduction sets $k = n + 2$ and returns (G, s, t, k) as the reduced instance of RENDEZVOUS.

Intuition for the Correctness

We present an intuition for the correctness of the reduction. Recall that we use $P[u, v, d_1] \circ P[v, w, d_2]$ to denote the unique path from u to w that contains v. Consider the paths $P[s, u_i^0, d^\star] \circ P[u_i^0, t, d^\star]$ and $P[s, u_i^{d^\star+1}, d^\star] \circ P[u_i^{d^\star+1}, t, d^\star]$

for every $i \in [n]$ and paths $P[s, c_j^\ell, 2d^\star +1] \circ P[c_j^\ell, t, 2d^\star +1]$ and $P[s, c_j^r, 2d^\star +1] \circ P[c_j^r, t, 2d^\star +1]$ for every $j \in [m]$. As we will see, the only way Facilitator can win in Rendezvous Games with Adversaries is by moving Romeo and Juliet along with one of these $2n + 2m$ paths. As g_1, g_2 are common neighbors of s and t, Divider needs to put two of the $k = n + 2$ agents on g_1 and g_2. Suppose he puts the remaining n agents at some internal vertices of the paths added while encoding variables. He places the agents such that each path contains one of them. For example, the red vertices in Fig. 7 corresponds to the positions of the agents on the paths added while encoding variables $x_1, x_2,$ and x_3.

Suppose Divider places an agent at an internal vertex, say u_1^d, of $P[u_1^0, u_1^{d^\star +1}, d^\star]$. Facilitator can move Romeo and Juliet to either u_1^0 or $u_1^{d^\star +1}$ in $d^\star + 1$ steps. The length of the path from u_1^0 to u_1^d is d and the length of the path from $u_1^{d^\star +1}$ to u_1^d is $d^\star - d + 1$.

- Divider can move the agent from u_1^d to u_1^0 in at most $d^\star$ steps as $d \leq d^\star$, and
- Divider can move the agent from u_1^d to $u_1^{d^\star +1}$ in at most $d^\star$ steps as $d^\star - d + 1 \leq d^\star$.

Recall that the simple path $P[c_j^\ell, u_1^0, 2d^\star - d_1]$, as the notation suggests, has $2d^\star - d_1$ internal vertices. Hence, the path $P[c_j^\ell, u_1^0, 2d^\star - d_1] \circ P[u_1^0, u_1^d, d - 1]$ has $(2d^\star - d_1) + 1 + (d - 1)$ many internal vertices. Hence, the length of path from c_j^ℓ to u_1^d is $2d^\star + 1 + d - d_1$.

- Divider can move the agent from u_1^d to c_j^ℓ in at most $2d^\star + 1$ steps only if $d \leq d_1$.

Consider symmetric arguments for c_j^r. The simple path $P[c_j^r, u_1^{d^\star +1}, d^\star + d_1]$ has $d^\star + d_1$ many internal vertices. Hence, the path $P[c_j^r, u_1^{d^\star +1}, d^\star + d_1] \circ P[u_1^{d^\star +1}, u_1^d, d^\star - d]$ has $(d^\star + d_1) + 1 + (d^\star - d)$ many internal vertices. Hence, the length of the path from u_1^d to c_j^r is $2d^\star + 2 + d_1 - d$.

- Divider can move the agent from u_1^d to c_j^ℓ in at most $2d^\star + 1$ steps only if $d > d_1$.

Suppose there is a clause $C_j \in \mathcal{C}$ such that $C_j = \mathrm{NAE}(x_1 \leq d_1, x_2 \leq d_2, x_3 \leq d_3)$. Consider the two vertices c_j^ℓ and c_j^r added while encoding C_j. Note that Facilitator can move Romeo and Juliet to either c_j^ℓ or c_j^r in $2d^\star + 2$ steps. Moreover, apart from s and t, the only branching points in paths $P[s, c_j^\ell, 2d^\star + 1] \circ P[c_j^\ell, t, 2d^\star + 1]$ and $P[s, c_j^r, 2d^\star + 1] \circ P[c_j^r, t, 2d^\star + 1]$ are c_j^ℓ and c_j^r, respectively. Hence, Divider needs to place an agent that he can move to c_j^ℓ in at most $2d^\star + 1$ steps. Similarly, he needs to place an agent that he can move to c_j^r in at most $2d^\star + 1$ steps. As we will see, Divider can only move the agents stationed at the paths corresponding to variables $x_1, x_2,$ or x_3 to c_j^ℓ or c_j^r in at most $2d^\star + 1$ steps. Hence, he needs to place agents at the interior vertices, say $u_1^{c_1}, u_2^{c_2}, u_3^{c_3}$, of $P[u_1^0, u_1^{d^\star +1}, d^\star]$, $P[u_2^0, u_2^{d^\star +1}, d^\star]$ and $P[u_3^0, u_3^{d^\star +1}, d^\star]$, respectively, such that

- at least one of the inequalities in $\{c_1 \leq d_1; c_2 \leq d_2; c_3 \leq d_3\}$ is `True`, and
- *simultaneously* at least one of the inequalities in $\{c_1 > d_1; c_2 > d_2; c_3 > d_3\}$ is `True`.

This position of agents corresponds to the value of variables x_1, x_2, x_3 in $[d^\star]$ that satisfy the clause $C_j = \text{NAE}(x_1 \leq d_1, x_2 \leq d_2, x_3 \leq d_3)$. In the following two lemmas, we formalize these intuitions.

Lemma 7 *If* $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ *is a* YES-*instance of* (MONOTONE) NAE- INTEGER- 3- SAT, *then* $(G, s, t, n+2)$ *is a* NO-*instance of* RENDEZVOUS.

Proof We show that if $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ is a YES-instance of (MONOTONE) NAE- INTEGER- 3- SAT, then Divider with $n+2$ agents can win in Rendezvous Game with Adversaries. Recall that $n = |\mathcal{X}|$, and $m = |\mathcal{C}|$. Suppose $\psi : \mathcal{X} \to [d^\star]$ be a satisfying assignment, and $\psi(x_i) = d_i$ for every $i \in [n]$.

We describe a winning strategy for Dividers with the agents $D_1, D_2, \ldots, D_{n+2}$. Initially, he puts D_i in the vertex $u_i^{d_i}$, for every $i \in [n]$, and D_{n+1} and D_{n+2} in g_1 and g_2 respectively. He does not move agents $D_1, \ldots, D_{n+2}$, until Facilitator moves Romeo or Juliet from s or t, respectively. Suppose without loss of generality Facilitator first moves Romeo from s (she may or may not move Juliet from t). By the construction, she can move Romeo either on the paths $P[s, u_i^0, d^\star]$, $P[s, u_i^{d^\star+1}, d^\star]$ for some $i \in [n]$ or on the paths $P[s, c_j^\ell, 2d^\star + 1]$, $P[s, c_j^r, 2d^\star + 1]$ for some $j \in [m]$.

Suppose Facilitator moves Romeo from s to a vertex on the path $P[s, u_i^0, d^\star]$ for some $i \in [n]$. Divider moves D_{n+1} from g_1 to s and then towards u_i^0 as she moves Romeo towards u_i^0. He also moves D_i to u_i^0 in at most $\psi(x_i)$ steps along the path $P[u_i^0, u_i^{d^\star+1}, d^\star]$. Facilitator needs at least $d^\star + 1$ steps to move both Romeo and Juliet in u_i^0 starting from s and t respectively. As $\psi(x_i) \leq d^\star$, Divider can move D_i to u_i^0 before Facilitator can move both Romeo and Juliet to u_i^0. Hence, he can block Romeo by D_i and D_{n+1} on the path $P[s, u_i^0, d^\star]$. Divider keeps moving D_i and D_{n+1} towards Romeo's position and in at most $\psi(x_i) + d^\star - 1$ steps Facilitator can not move Romeo. This implies Divider wins by keeping Romeo in its current position with its neighbors occupied by D_i and D_{n+1}. The argument also follows when Facilitator moves Romeo from s to a vertex on the path $P[s, u_i^{d^\star+1}, d^\star]$ for some $i \in [n]$ since Divider can move D_i to $u_i^{d^\star+1}$ in at most $d^\star - \psi(x_i) + 1$ ($\leq d^\star$) steps.

Suppose Facilitator moves Romeo from s to a vertex on the path $P[s, c_j^\ell, 2d^\star+1]$ for some $j \in [m]$. Let $C_j = \text{NAE}(x_{i_1} \leq d_1, x_{i_2} \leq d_2, x_{i_3} \leq d_3)$. Since ψ is a satisfying assignment, it sets the values of variables such that at least one of the inequalities will be True and at least one of the inequalities will be False. We assume without loss of generality that $\psi(x_{i_1}) \leq d_1$ and $\psi(x_{i_2}) > d_2$. Divider moves D_{i_1} to c_j^ℓ in at most $2d^\star - d_1 + 1 + \psi(x_{i_1})$ steps through the path $P[c_j^\ell, u_{i_1}^0, 2d^\star - d_1] \circ P[u_{i_1}^0, u_{i_1}^{d^\star+1}, d^\star]$. As in the previous case, he can move D_{n+1} from g_1 to s and then keep moving towards c_j^ℓ as Facilitator moves Romeo towards c_j^ℓ. He can move D_{n+2} in a similar manner with respect to Juliet.

Facilitator can move both Romeo and Juliet to c_j^ℓ in at least $2d^\star + 2$ steps starting from s and t respectively. Divider can move D_{i_1} to c_j^ℓ before Romeo and Juliet as $2d^\star - d_1 + 1 + \psi(x_{i_1}) \leq 2d^\star + 1$. Hence, Romeo is blocked by D_{i_1} and D_{n+1} on the path $P[s, c_j^\ell, 2d^\star + 1]$ and Juliet cannot reach Romeo. Divider keeps moving D_{i_1} and D_{n+1} towards Romeo and in at most $4d^\star - d_1 + 1 + \psi(x_{i_1})$ steps Romeo cannot move.

This implies Divider wins. The argument also follows when Facilitator moves Romeo from s to a vertex on the path $P[s, c_j^r, 2d^\star + 1]$ for some $j \in [m]$ since Divider can move D_{i_2} to c_j^r in at most $2d^\star + 2 + d_2 - \psi(x_{i_2})$ $(< 2d^\star + 2)$ steps.

This implies that if $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ is a Yes-instance of (Monotone) NAE- Integer-3- Sat, then Divider with $n+2$ agents can win in Rendezvous Game with Adversaries, i.e., $(G, s, t, n + 2)$ is a No-instance of Rendezvous. □

Lemma 8 *If* $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ *is a* No*-instance of* (Monotone) NAE- Integer- 3- Sat*, then* $(G, s, t, n + 2)$ *is a* Yes*-instance of* Rendezvous*.*

Proof We show that if $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ is a No-instance of (Monotone) NAE- Integer-3- Sat, then Facilitator wins in at most $2d^\star + 2$ steps against Divider with $n + 2$ agents.

We first consider two simple cases where Facilitator has an easy winning strategy. First, consider the case when Divider does not place his agents at g_1 or g_2. Then, she can move Romeo and Juliet there and win in one step. Second, consider the case when there is $i \in [n]$ such that none of Divider's agents is within distance $d^\star$ from u_i^0 or from $u_i^{d^\star+1}$. In the first sub-case, she can move Romeo and Juliet to u_i^0 in $d^\star + 1$ steps through the paths $P[s, u_i^0, d^\star]$ and $P[t, u_i^0, d^\star]$, respectively, and win. Similarly, in the second sub-case she can move Romeo and Juliet to $u_i^{d^\star+1}$ in $d^\star + 1$ steps through the paths $P[s, u_i^{d^\star+1}, d^\star]$ and $P[t, u_i^{d^\star+1}, d^\star]$, respectively, and win.

In the remaining proof, we suppose that Divider places D_{n+1} at g_1 and D_{n+2} at g_2. Moreover, for every $i \in [n]$, there is a Divider's agent within distance $d^\star$ from u_i^0 and within distance $d^\star$ from $u_i^{d^\star+1}$. Suppose from now that for every $i \in [n]$, there exists a Divider's agent within distance $d^\star$ from u_i^0 and within distance $d^\star$ from $u_i^{d^\star+1}$. By the construction and the fact that Divider can not place an agent at s or t, a single Divider's agent cannot be within distance $d^\star$ from both u_i^0 and u_j^0, or $u_i^{d^\star+1}$ and $u_j^{d^\star+1}$, or u_i^0 and $u_j^{d^\star+1}$, for $i \neq j \in [n]$. As Divider has n remaining agents, for every $i \in [n]$, there must be an agent, say D_i, within distance $d^\star$ from both u_i^0 and $u_i^{d^\star+1}$. This is possible only when for every $i \in [n]$, D_i is on one of the internal vertices of the path $P[u_i^0, u_i^{d^\star+1}, d^\star]$. Suppose $\phi : [n] \to [d^\star]$ is the mapping corresponding to the initial position of the Divider's agents. Formally, for every $i \in [n]$, Divider places agent D_i on $u_i^{\phi(i)}$. For every $i \in [n]$, the initial position of D_i also represents a possible assignment of variable x_i in $(\mathcal{X}, \mathcal{D}, \mathcal{C})$.

We now define the Facilitator's strategy. Considering $\mathcal{X} = \{x_1, \ldots, x_n\}$ as the variables that each take a value in the domain $\mathcal{D} = \{1, \ldots, d^\star\}$, she constructs a collection $\mathcal{C}$ of clauses such that for every $j \in [m]$, clause $C_j = \text{NAE}(x_{i_1} \leq d_1, x_{i_2} \leq d_2, x_{i_3} \leq d_3)$, where $x_{i_1}, x_{i_2}, x_{i_3} \in \mathcal{X}$ for some $d_1, d_2, d_3 \in [d^\star]$. Alternately, she reverse-engineers the process used by the reductions to encode clauses. She also constructs an assignment $\psi : \mathcal{X} \to \mathcal{D} = [d^\star]$ by considering the initial positions of agents $D_1, D_2, \ldots, D_n$. Formally, $\psi(x_i) = \phi(i)$ for every $i \in [n]$. It then determines whether the following statements are True.

1. For some clause $C_j = \text{NAE}(x_{i_1} \leq d_1, x_{i_2} \leq d_2, x_{i_3} \leq d_3)$, all of the inequalities in $\{\psi(x_{i_1}) \leq d_1; \psi(x_{i_2}) \leq d_2; \psi(x_{i_3}) \leq d_3\}$ are True, where $j \in [m]$.

2. For some clause $C_j = \text{NAE}(x_{i_1} \leq d_1, x_{i_2} \leq d_2, x_{i_3} \leq d_3)$, all of the inequalities in $\{\psi(x_{i_1}) \leq d_1; \psi(x_{i_2}) \leq d_2; \psi(x_{i_3}) \leq d_3\}$ are `False`, where $j \in [m]$.

Facilitator has to make a critical choice in the first step where she has to decide about moving Romeo towards $c_1^\ell, \ldots, c_m^\ell, c_1^r, \ldots,$ or c_m^r. This choice depends on which of the above statement is `True` and for which clause it is `True`. If the first statement is `True` for the clause $C_j \in \mathcal{C}$, then she moves Romeo and Juliet towards c_j^r. Similarly, if the second statement is `True` for the clause $C_j \in \mathcal{C}$, then she moves Romeo and Juliet towards c_j^ℓ.

To argue that this is indeed a winning strategy for Facilitator, we first argue that for any initial positions of Divider's agents, at least one of the two statements above is `True`. Assume the above two statements are `False` for all $j \in [m]$, which implies in all the clauses $C_j \in \mathcal{C}$, not all three inequalities are `True` and not all are `False`. Hence, all the clauses are satisfied by the assignment ψ. This, however, contradicts the fact that $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ is a No-instance. Hence, for any initial positions of Divider's agents, at least one of the two sentences is `True`.

This allows Facilitator to make her choice. It remains to argue that Romeo and Juliet can meet at the vertex c_j^ℓ or c_j^r which Facilitator has chosen. Suppose, one of the statements is `True` for the clause C_j. For notational convenience, suppose $C_j = \text{NAE}(x_1 \leq d_1, x_2 \leq d_2, x_3 \leq d_3)$.

Suppose $\psi(x_1) \leq d_1, \psi(x_2) \leq d_2, \psi(x_3) \leq d_3$ (i.e. First statement is `True`). Then, as mentioned in the Facilitator's strategy, her choice will be to move Romeo and Juliet towards c_j^r. For $i \in \{1, 2, 3\}$, Divider needs at least $d^\star - \psi(x_i) + 1 + d^\star + d_i + 1 \geq 2d^\star + 2$ steps to move D_i from $u_i^{\psi(x_i)}$ to c_j^r via the shortest path $P[u_i^{\psi(x_i)}, u_i^{d^\star+1}, d^\star - \psi(x_i)] \circ P[u_i^{d^\star+1}, c_j^r, d^\star + d_i]$. Note that, by the construction, the Divider's agents that are at distance less than or equal to $2d^\star + 2$ from c_j^r are D_1, D_2 and D_3, only. Facilitator can move Romeo and Juliet to c_j^r in $2d^\star + 2$ steps through the paths $P[s, c_j^r, 2d^\star + 1]$ and $P[t, c_j^r, 2d^\star + 1]$, respectively. Since Facilitator takes the first turn, she can move Romeo and Juliet to c_j^r before Divider's agents. Hence, Facilitator wins in $2d^\star + 2$ steps.

Suppose $\psi(x_1) > d_1, \psi(x_2) > d_2, \psi(x_3) > d_3$ (i.e. Second statement is `True`). Then, as mentioned in the Facilitator's strategy, her choice will be to move Romeo and Juliet towards c_j^ℓ. For $i \in \{1, 2, 3\}$, Divider needs at least $\psi(x_i) - 1 + 1 + 2d^\star - d_i + 1 > 2d^\star + 1$ steps to move D_i from $u_i^{\psi(x_i)}$ to c_j^ℓ via the shortest path $P[u_i^{\psi(x_i)}, u_i^0, \psi(x_i) - 1] \circ P[u_i^0, c_j^\ell, 2d^\star - d_i]$. Once again, by the construction, the Divider's agents that are at distance less than or equal to $2d^\star + 2$ from c_j^ℓ are D_1, D_2 and D_3. Facilitator moves Romeo and Juliet to c_j^ℓ in $2d^\star + 2$ steps through the paths $P[s, c_j^\ell, 2d^\star + 1]$ and $P[t, c_j^\ell, 2d^\star + 1]$ respectively. Since Facilitator takes the first turn, Romeo and Juliet is moved to c_j^ℓ before Divider agents and Facilitator wins in $2d^\star + 2$ steps.

This implies that if $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ is a No-instance of (Monotone) NAE-Integer-3-Sat, then Facilitator wins in at most $2d^\star + 2$ steps against Divider with $n + 2$ agents, i.e., $(G, s, t, n + 2)$ is a Yes-instance of Rendezvous. □

By the construction, the number of agents is upper bounded by the number of variables in (MONOTONE) NAE- INTEGER- 3- SAT plus two. Consider the set $S := \bigcup_{i \in [n]} \{u_i^0, u_i^{d^\star+1}\} \cup \{s, t\}$ of $2n + 2$ vertices in G. It is easy to verify that $G - S$ is a collection of paths (corresponding to variable gadgets) and subdivided stars (centered at the vertices added while encoding the clauses). It is easy to verify that the pathwidth of a subdivided star is at most two. Hence, the feedback vertex set number and the pathwidth of the resulting graph are bounded by the linear function in the number of variables. Lemmas 7, 8 and the fact that the reduction can be completed in time polynomial in the input size imply Theorem 2 which we restate here.

Theorem 2 RENDEZVOUS *is co-W[1]-hardwhen parameterized by:*

- *the feedback vertex set number and the solution size, or*
- *the pathwidth and the solution size.*

5 Parameterizing by Vertex Cover

In this section we focus on Theorem 3:

Theorem 3 RENDEZVOUS *is FPT when parameterized by the vertex cover number of the input graph and the solution size. Moreover, the problem does not admit a polynomial kernel when parameterized by the vertex cover number and the solution size unless NP $\subseteq$ co-NP/poly.*

Throughout this section, we assume that a vertex cover X of size $\text{vc}(G)$ is given as a part of the input. We first discuss the FPT result.

Reduction Rule 5.1 *Consider an instance (G, X, s, t, k) of* RENDEZVOUS. *If $s = t$, $st \in E(G)$, $|N(s) \cap N(t)| > k$, then return a trivial* YES-*instance.*

For the rest of this discussion, we will assume that any instance (G, s, t, k) of RENDEZOUS under consideration does *not* satisfy the premise of Reduction Rule 5.1, i.e, we assume that we are not dealing with trivial YES instances. Also, since the vertices s and t can always be added to the vertex cover and this only increases the parameter by two, we assume for simplicity—and without loss of generality—that $s, t \in X$.

We now introduce some notation. For a subset $Y \subseteq X$, let $I_Y \subseteq G \setminus X$ denote the set of vertices in $G \setminus X$ whose neighborhood is exactly Y. Note that $\{I_Y\}_{Y \subseteq X}$ is a partition of $G \setminus X$ into at most $2^{\text{vc}(G)}$ many parts. For a vertex $v \in G \setminus X$, we use $\mathcal{E}_{G,X}(v)$ to denote the part that v belongs to, in other words, $\mathcal{E}_{G,X}(v) = I_{N(v)}$. We now apply the following reduction rule.

Reduction Rule 5.2 *Consider an instance (G, X, s, t, k) of* RENDEZVOUS. *Repeat the following for each $v \in G \setminus X$. If $|\mathcal{E}_{G,X}(v)| > k + 1$, then choose any subset of exactly $k + 1$ vertices from $\mathcal{E}_{G,X}(v)$ and delete the rest of the vertices from $\mathcal{E}_{G,X}(v)$.*

Lemma 9 *Reduction Rule 5.2 is safe.*

Proof Let (G, X, s, t, k) denote the input instance, and let $v \in G \backslash X$ be arbitrary but fixed. Further, let (H, X, s, t, k) denote the instance obtained by applying Reduction Rule 5.2 with respect to v. If $|\mathcal{E}(v)| \leq k + 1$ in G then $G = H$ and there is nothing to prove. Otherwise, let $Q_v \subseteq \mathcal{E}_{G,X}(v)$ denote the set of vertices deleted by the application of the reduction rule with respect to v. Note that $H = G \backslash Q_v$. Also observe that $|\mathcal{E}_{H,X}(v)| = k + 1$.

To begin with, suppose the Facilitator has a winning strategy in G. Observe that the Facilitator can employ the same strategy in H as well, except when the strategy involves moving to a vertex $u \in Q_v$. However, since $|\mathcal{E}_{H,X}(v)| = k + 1$, we have that there is at least one vertex w in $H \setminus X$ that has the same neighborhood as u and is not occupied by an agent of the Divider, since the Divider has only k agents at their disposal. The strategy, at this point, would remain valid if we were to replace u with w. If the strategy involved using two distinct vertices from Q_v in the same step, then note that we can modify the strategy and have the Faciliator's agents meet immediately at the vertex w.

On the other hand, if the Facilitator had a winning strategy in H, then it is easy to check that the Facilitator can win in G by mimicking the strategy directly. Another way to see this is the following. Suppose that the Divider had a winning strategy in G. Then observe that in any step, without loss of generality, if the Divider's agents occupy some vertices of $\mathcal{E}_{G,X}(v)$, we can replace this configuration with all of these agents on a single vertex of $\mathcal{E}_{G,X}(v)$ outside Q_v. Thus any winning strategy for the divider in G can be adapted to a valid winning strategy in H. This concludes the argument for the equivlance of the two instances. $\square$

Lemma 10 RENDEZVOUS *is FPT when parameterized by the vertex cover number and the solution size.*

Proof Observe that repeated applications of Reduction Rule 5.2 ensures that $|V(G)| = |X| + |G \backslash X| \leq \text{vc}(G) + 2^{\text{vc}(G)} \cdot (k + 1)$. Thus we have an exponential kernel in $\text{vc}(G)$, and the claim follows.

Now, we establish the lower bound claimed in Theorem 3 by showing the following.

Lemma 11 RENDEZVOUS *does not admit a polynomial kernel when parameterized by the vertex cover number and the solution size unless* NP⊆ *co-NP/poly.*

The proof is based on observing that the instance in the reduction used in [4]—to prove that problem is co-W[2]-hard when parameterized by the solution size—has bounded vertex cover number. In particular, the reduction is from SET COVER, which does not admit a poylnomial kernel parameterized by the size of the universe unless NP⊆ co-NP/poly [5]. We reproduce the construction here for completeness (Fig. 8).

Proof Recall that an instance of SET COVER consists of a universe U of size n, a family over U of size m, and a budget of k; and the question is if there exists a collection of at most k sets from the given family whose union is U. Let $(U, \mathcal{S}, k)$ be an instance of SET COVER. Let $U = \{u_1, \ldots, u_n\}$ and $\mathcal{S} = \{S_1, \ldots, S_m\}$.

- Construct a set of n vertices $U = \{u_1, \ldots, u_n\}$ corresponding to the universe.
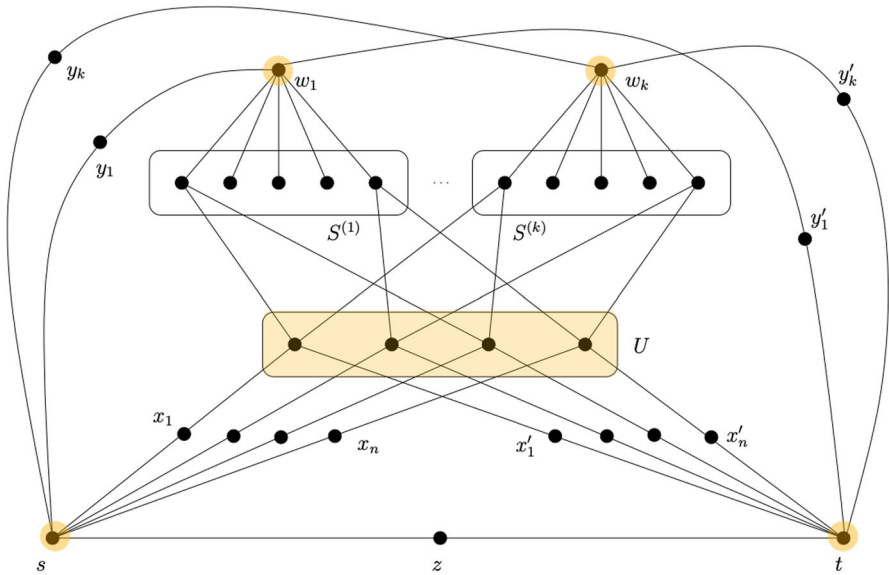
Fig. 8 A schematic of the reduction with the vertex cover vertices highlighted

- For every $i \in \{1, \ldots, k\}$, construct a set of m vertices $S^{(i)} = \left\{s_1^{(i)}, \ldots, s_m^{(i)}\right\}$; each $S^{(i)}$ corresponds to a copy of $\mathcal{S}$.
- For every $i \in \{1, \ldots, k\}, h \in \{1, \ldots, m\}$ and $h \in \{1, \ldots, n\}$, make $s_j^{(i)}$ and u_h adjacent if the element of the universe u_h is in $S_j \in \mathcal{S}$.
- For every $i \in \{1, \ldots, k\}$, construct a vertex w_i and make it adjacent to $s_1^{(i)}, \ldots, s_m^{(i)}$.
- Construct two vertices s and t.
- For every $h \in \{1, \ldots, n\}$, join s and u_h by a path sx_hu_h and joint u_h and t by a path $u_hx_h't$.
- For every $i \in \{1, \ldots, k\}$, join s and w_i by a path sy_iw_i and join w_i and t by a path $w_iy_i't$.
- Construct a vertex z and make it adjacent to s and t.

It is shown in [4] that $(U, \mathcal{S}, k)$ is a yes-instance of SET COVER if and only if Divider with $k + 1$ agents can win in the RENDEZVOUS game. It is straightforward to check that $U \cup \{s, t\} \cup \{w_i \mid i \in [k]\}$ is a vertex cover for the reduced instance of size at most $m + k + 2$. The claim follows from the hardness of obtaining a polynomial kernel for SET COVER parameterized by $m + k$, since the equivalence of the instances is already known. □

6 Some Polynomial Cases

In this section, we focus on a few tractable scenarios.

Theorem 4 RENDEZVOUS *admits a polynomial-time algorithm when input graph is a grid or has treewidth at most two.*

As the ideas used to prove the theorem are different for these two classes of graphs, we present two lemma separate lemmas. As mentioned in the Introduction, all the known polynomial-time algorithms are obtained by proving that the size of a dynamic separator, denoted by $d_G(s, t)$, is same as that of a (static) separator, denoted by $\lambda_G(s, t)$. In the next lemma, we present a winning strategy for Divider for any non-trivial instances when the input graph is a grid. This makes the class of grid graphs a unique graph class in which the problem admits polynomial time algorithm even when dynamic separators can be smaller than separators.

Lemma 12 *For a grid graph G and two non-adjacent vertices $s, t \in V(G)$, $d_G(s, t) = 2$.*

Proof Consider an instance (G, s, t, k) of RENDEZVOUS where G is a $M \times N$ undirected grid. Without loss of generality, we assume that s and t are non-adjacent in G. It is known [4, Theorem 2] that $d_G(s, t) = 1$ if and only if $\lambda_G(s, t) = 1$. Since in grid graphs any two vertices are part of at least one cycle, $\lambda_G(s, t) \geq 2$. Hence, for any non-adjacent pair of vertices s and t, $d_G(s, t) \geq 2$. Therefore, it is sufficient to show that $d_G(s, t) \leq 2$. We prove that Divider with 2 agents has a winning strategy on G against Facilitator starting from s and t.

We respresent vertex v of G that is in i^{th} row and j^{th} column as (i, j), where $i \in [M]$ is the row number of vertex v and $j \in [N]$ is the column number of vertex v. Let s be (s_x, s_y) and t be (t_x, t_y), where $s_x, t_x \in [M]$, and $s_y, t_y \in [N]$. For Facilitator to win, she must make the difference between the row number as well as column number of the vertices having Romeo and Juliet equal to 0. Since (s_x, s_y) and (t_x, t_y) are two different and non-adjacent vertices either $|s_x - t_x| > 0$ or $|s_y - t_y| > 0$. We assume without loss of generality $|s_x - t_x| > 0$ and $s_x < t_x$; in other words, s and t are on different rows and s is "below" t in the grid.

We describe a winning strategy for Divider with the agents D_1 and D_2. Intuitively, the agent D_1 starts off to the "top" of s and the agent D_2 starts off at a location to the "bottom" of t. Their goal will be to maintain the initial separation between s and t by not allowing the agent on s to advance upwards or the agent on t to advance downwards. They do this by "tracking" the agent movements and mimicking them whenever there is a shift to an adjacent column, and staying put if the agents are moving along the same column, in which case they are drifting further apart.

In particular, to begin with, Divider puts D_1 in the vertex $(s_x + 1, s_y)$ and D_2 in the vertex $(t_x - 1, t_y)$ (since $s_x < t_x, s_x < M$ and $t_x > 1$). Then the following strategy is used. The agents D_1 and D_2 are keeping their positions until Facilitator moves Romeo or Juliet from (s_x, s_y) or (t_x, t_y), respectively. Whenever Facilitator moves Romeo, the agent D_1 replicates her move and similarly, whenever Facilitator moves Juliet, the agent D_2 replicates her move. Facilitator can move Romeo to either $(s_x - 1, s_y)$ (if $s_x > 1$) or $(s_x, s_y - 1)$ (if $s_y > 1$ and D_2 is not on this vertex) or $(s_x, s_y + 1)$ (if $s_y < N$ and D_2 is not on this vertex). The vertex $(s_x + 1, s_y)$ is occupied by D_1. Let the new position of Romeo be (s'_x, s'_y), where $s'_x \in [M]$ and $s'_y \in [N]$. Divider moves the agent D_1 to (s_x, s_y) or $(s_x + 1, s_y - 1)$ or $(s_x + 1, s_y + 1)$ corresponding to the above mentioned three possible moves of the Facilitator for Romeo. Similarly, Facilitator can move Juliet to either $(t_x + 1, t_y)$ (if $t_x < M$) or $(t_x, t_y - 1)$ (if $t_y > 1$ and D_1 is not on this vertex) or $(t_x, t_y + 1)$ (if $t_y < N$ and D_1 is not on this vertex). The

vertex $(t_x - 1, t_y)$ is occupied by D_2. Let the new position of Juliet be (t'_x, t'_y), where $t'_x \in [M]$ and $t'_y \in [N]$. Divider moves the agent D_2 to (t_x, t_y) or $(t_x - 1, t_y - 1)$ or $(t_x - 1, t_y + 1)$ corresponding to the above mentioned three possible moves of the Facilitator for Juliet. Observe that, the difference of the row number of Juliet and Romeo does not decrease after any of the possible moves, i.e. $t'_x - s'_x \geq t_x - s_x$. Divider follows the same strategy after every move of Facilitator for Romeo and Juliet, and the strategy ensures that the difference of the row number of Juliet and Romeo does not decrease after any possible move of the Facilitator. Hence, Divider prevents Romeo and Juliet from meeting by ensuring that the difference of their row number does not decrease after any number of moves. This implies Divider wins. The argument also follows when $|s_y - t_y| > 0$ since Divider can prevent Romeo and Juliet from meeting by ensuring that the difference of their column number does not decrease after any number of moves.

We conclude that Divider with 2 agents has a winning strategy on G against Facilitator starting from s and t, which implies $d_G \leq 2$. Since $d_G \leq 2$ as well as $d_G \geq 2$, $d_G = 2$. This implies that for a grid graph G and two non-adjacent vertices $s, t \in V(G)$, $d_G(s, t) = 2$.   □

Lemma 13 *If G is a connected graph of treewidth at most* 2, *then for every $s, t \in V(G)$, $d_G(s, t) = \lambda_G(s, t)$.*

Proof We first consider the simple case when $d_G(s, t) = 1$. As stated in [4, Theorem 2], $d_G(s, t) = 1$ if and only if $\lambda_G(s, t) = 1$, and hence in this case, $d_G(s, t) = \lambda_G(s, t) = 1$ which proves the claim. In the remaining proof, we consider the case when $d_G(s, t) \geq 2$ which also implies $\lambda_G(s, t) \geq 2$.

Consider an optimal tree decomposition of G. In this decomposition, if s and t are in different bags, then $\lambda_G(s, t) \leq 2$. This implies $\lambda_G(s, t) = 2$, and as $d_G(s, t) \leq d_G(s, t)$, we have $d_G(s, t) = 2$. In this case also, we get the desired result. Now, consider the case when s and t are in the same bag in the tree decomposition. Consider the graph G' by adding an edge (s, t) to graph G. Since, both of these vertices are in the same bag in an optimal tree decomposition, we can conclude that treewidth of G' is also at most 2. Now, consider the connected components $C_1, C_2, \ldots, C_q$ of $G' - \{s, t\}$ (which is same as connected components of $G - \{s, t\}$). Any (s, t)-paths in G is completely contained in one of these connected components.

Consider a connected component C_i that contains one of (s, t)-paths in G, apart from its endpoints. We argue that there exists a vertex $x_i \in V(C_i)$ such that every (s, t)-paths in G whose all internal vertices are in C_i, also contains x_i. Assume, for the sake of contradiction, that it is not the case, and there exists two (s, t)-paths in G, say P_1 and P_2, whose all internal vertices are in C_i. As C_i is a connected component, there exists a vertex y_1, part of P_1 and a vertex y_2, part of P_2, such that there is a (y_1, y_2)-path all the internal vertices are in $V(C_i) \setminus (V(P_1) \cup V(P_2))$. This implies that the following five paths in G (and hence also in G') only meet at the endpoints: (s, y_1)-path, (y_1, t)-path (both along P_1), (s, y_2)-path, (y_2, t)-path (both along P_2), and (y_1, y_2)-path. As edge (s, t) is present in G', this implies that G' contains K_4 as a contraction-minor. This, however, contradicts the fact that the treewidth of G' is at most 2. Hence, our assumption is wrong and exists a vertex $x_i \in V(C_i)$ such that every (s, t)-paths in G whose all internal vertices are in C_i, also contains x_i.

The arguments in the previous paragraph implies that $\lambda_G(s, t)$ is the number of connected components of $G - \{s, t\}$ that contains a (s, t)-path. Note that Divider has to put one agent in each such component, as otherwise, Facilitator wins by moving Romeo and Juliet towards each other along the shortest path that goes through the component that does contain the agent. Hence, $d_G(s, t) \leq \lambda_G(s, t)$. But, by definitions, $d_G(s, t) \geq \lambda_G(s, t)$. This implies the lemma. □

7 Conclusion

In this work, we studied the game of rendezvous with adversaries on a graph introduced by Fomin, Golovach, and Thilikos [4]. The game is a natural dynamic version of the problem of finding a vertex cut between two vertices s and t. Given that the problem is W[2]-hard when parameterized by the natural parameter, i.e. the solution size, we continued studying structural parameters of the input graph. We proved, to our surprise, that the problem is co-NP-hard even when restricted to graphs whose feedback vertex set number is at most 14, or pathwidth is at most 16. In particular, we proved RENDEZVOUS is para-co-NP-hard parameterized by treewidth, thereby answering an open question by Fomin et al. [4]. It turns out that even augmenting the feedback vertex set number or the pathwidth with the solution size is not enough. Specifically, we proved that RENDEZVOUS is co-W[1]-hard when parameterized by the feedback vertex set number and the solution size, or the pathwidth and the solution size. Towards the positive side, we proved that the problem admits a natural exponential kernel when parameterized by the vertex cover number and the solution size, however this kernel cannot be improved to a polynomial kernel under standard complexity-theoretic assumptions. Finally, we presented polynomial time algorithms on two restricted cases and proved that RENDEZVOUS can be solved in polynomial time on the classes of treewidth at most two graphs and grids.

While we addressed the structural parameterized by arguably the most well studied parameters, it remains interesting to study the parameterized complexity for other structural parameters. Amongst these, we highlight the following question: *Is* REN-DEZVOUS W[1]-hard *when parameterized by the vertex cover number (only)?* We tend to believe it is indeed the case. To the best of our knowledge, the problems that are W[1]-hard when parameterized by the vertex cover number, like LIST COLORING, WEIGHTED (k, r)-CENTER, etc., have additional input arguments like lists or weights. We believe that the dynamic nature of the RENDEZVOUS problem might make it an exception to the above known trend.

Declarations

Competing interests The authors declare no competing interests.

References

1. Bringmann, K., Hermelin, D., Mnich, M., Van Leeuwen, E.J.: Parameterized complexity dichotomy for steiner multicut. J. Comput. Syst. Sci. **82**(6), 1020–1043 (2016). https://doi.org/10.1016/j.jcss.2016.03.003
2. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015). https://doi.org/10.1007/978-3-319-21275-3
3. Diestel, R.: Graph Theory, 4th Edition, vol. 173 of Graduate Texts in Mathematics. Springer (2012)
4. Fomin, F.V., Golovach, P.A., Thilikos, D.M.: Can Romeo and Juliet meet? Or rendezvous games with adversaries on graphs. Inf. Comput. **293**, 105049 (2023). https://doi.org/10.1016/j.ic.2023.105049
5. Fomin, F.V., Lokshtanov, D., Saurabh, S., Zehavi, M.: Kernelization: Theory of Parameterized Preprocessing. Cambridge University Press (2019). https://doi.org/10.1017/9781107415157
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)