# An ETH-Tight FPT Algorithm for Rejection-Proof Set Packing with Applications to Kidney Exchange

**Bart M. P. Jansen**
Eindhoven University of Technology, The Netherlands

**Jeroen S. K. Lamme**
Eindhoven University of Technology, The Netherlands

**Ruben F. A. Verhaegh**
Eindhoven University of Technology, The Netherlands

## Abstract

We study the parameterized complexity of a recently introduced multi-agent variant of the Kidney Exchange problem. Given a directed graph $G$ and integers $d$ and $k$, the standard problem asks whether $G$ contains a packing of vertex-disjoint cycles, each of length $\leq d$, covering at least $k$ vertices in total. In the multi-agent setting we consider, the vertex set is partitioned over several agents who reject a cycle packing as solution if it can be modified into an alternative packing that covers more of their own vertices. A cycle packing is called *rejection-proof* if no agent rejects it and the problem asks whether such a packing exists that covers at least $k$ vertices.

We exploit the sunflower lemma on a set packing formulation of the problem to give a kernel for this $\Sigma_2^P$-complete problem that is polynomial in $k$ for all constant values of $d$. We also provide a $2^{\mathcal{O}(k \log k)} + n^{\mathcal{O}(1)}$ algorithm based on it and show that this FPT algorithm is asymptotically optimal under the ETH. Further, we generalize the problem by including an additional positive integer $c$ in the input that naturally captures how much agents can modify a given cycle packing to reject it. For every constant $c$, the resulting problem simplifies from being $\Sigma_2^P$-complete to NP-complete. The super-exponential lower bound already holds for $c = 2$, though. We present an ad-hoc single-exponential algorithm for $c = 1$. These results reveal an interesting discrepancy between the classical and parameterized complexity of the problem and give a good view of what makes it hard.

**2012 ACM Subject Classification** Theory of computation → Parameterized complexity and exact algorithms; Applied computing → Health informatics; Theory of computation → Algorithmic game theory; Mathematics of computing → Combinatorial algorithms

**Keywords and phrases** Parameterized complexity, Multi-agent kidney exchange, Kernelization, Set packing

**Digital Object Identifier** 10.4230/LIPIcs.IPEC.2025.9

**Related Version** *Full Version*: `https://arxiv.org/abs/2509.11965`

**Funding** *Bart M. P. Jansen*: Supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003.
*Jeroen S. K. Lamme*: Supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003.

## 1 Introduction

The goal of this paper is to analyze the parameterized complexity of algorithmic problems that arise from a recently introduced multi-party variant of kidney exchange problems. Before describing our results, we first present some relevant context.

For patients with severely reduced kidney function, transplantation is often the preferred treatment [7]. As it is possible to live with only one healthy kidney, live kidney donation is employed to reduce waiting times for sparsely available donor organs. In this setting, a willing donor offers one of their kidneys for transplantation to a patient. The patient and donor must be medically compatible for a successful transplantation. When the patient and donor are not compatible, they can opt to participate in a kidney exchange program (KEP).

Such programs aim to link multiple patient-donor pairs so that a donor from one pair is compatible with a patient from another pair. Their goal is to group sets of patient-donor pairs, such that within each group, a cyclic order of compatibility exists. Performing this cycle of transplants means that donors only undergo surgery if their corresponding patient receives a kidney in that exchange. Individual programs each have their own additional restrictions [5]. It is common to restrict the length of transplantation cycles to some constant $d$, due to practical considerations which include doing the associated surgeries simultaneously.

We study these KEPs from an algorithmic perspective. Given a set of participating patient-donor pairs and the medical compatibilities between them, the algorithmic task becomes to find a combination of transplantation cycles that maximizes the total number of patients receiving a healthy kidney. We can model each patient-donor pair as a vertex and the compatibility of the donor of a pair $a$ to the patient of a pair $b$ as a directed edge from $a$ to $b$. We call a cycle of length at most $d$ a *$d$-cycle* and use this notation to obtain the following algorithmic graph problem that describes KEPs in one of their most basic forms.

Kidney Exchange with $d$-cycles (KE-$d$)
**Input:** A directed graph $G$ and an integer $k$.
**Task:** Determine whether there is a set $\mathcal{C}$ of pairwise vertex-disjoint $d$-cycles such that $\mathcal{C}$ covers at least $k$ vertices.

This problem has already been studied from the perspective of parameterized algorithms in the literature, with fixed-parameter tractable (FPT) algorithms w.r.t. the input parameter $k$ and the number of neighborhood equivalence classes as examples [16, 20].

**Multi-agent KEPs.** Extensions of the standard kidney exchange problem have also been studied. In this work, we consider one such extension which we now motivate and introduce. It is known that the fraction of patients to receive a healthy kidney can increase significantly with an increase in the total number of participating pairs [2]. This motivates collaborations between multiple programs, from now on referred to as agents. Such collaborations, however, can be hindered by a mismatch between the individual interests of agents, as these need not align with the collective goal of helping as many patients as possible overall. In particular, a socially optimal solution – in which as many patients as possible are treated overall – is not guaranteed to help as many people from each individual agent as that agent could have achieved on their own. This may stop agents from participating in collaborative programs or, when they do participate, move them to withhold information from the collective.

Several ways of incorporating fairness into multi-agent KEPs have been studied [2, 3, 4, 8, 9, 15, 19]. Recently, Blom, Smeulders, and Spieksma [6] also introduced a framework to capture fairness via a game-theoretical concept of *rejections*. They translate this into an algorithmic task that asks for a solution that is not rejected by any agent. We study the resulting algorithmic task. To introduce it, we first define the concept of rejections.

Recall the original KE-$d$ problem on directed graphs. To encode multiple agents into the input, we partition the vertex set of the input graph over the agents. In the obtained graph, we refer to a cycle that only contains vertices from a single agent $i$ as an *$i$-internal* cycle. Like in the original KE-$d$ problem, a solution consists of a packing $\mathcal{C}$ of pairwise vertex-disjoint $d$-cycles. A given such packing $\mathcal{C}$ will be rejected by an agent under the conditions listed in the definition below. See also Figure 1 for an example, adapted from a figure by Blom et al.

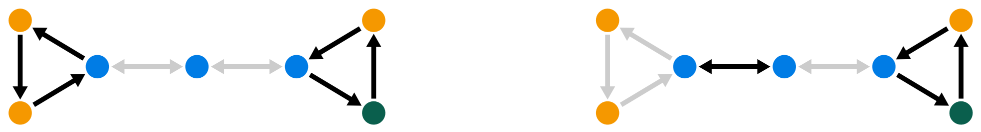


**Figure 1** A socially optimal packing of cycles is shown on the left and covers six vertices. However, the blue agent rejects this solution as they could modify it to cover more blue vertices as shown on the right. They can do this by rejecting the left-most 3-cycle and including an internal 2-cycle.

▶ **Definition 1.1.** *Let $G$ be a graph, where $V(G) = V_1 \cup \ldots \cup V_p$ is partitioned over $p$ different agents. Let $d$ be an upper bound on the length of allowed transplantation cycles. Let $\mathcal{C}$ be a set of pairwise vertex-disjoint $d$-cycles in $G$. If there is an agent $i \in [p]$ for which there exist*

- *a subset $\mathcal{C}_{\text{rej}} \subseteq \mathcal{C}$, and*
- *a set $\mathcal{C}_{\text{int}}$ of $i$-internal $d$-cycles, such that $\mathcal{C}' := (\mathcal{C} \setminus \mathcal{C}_{\text{rej}}) \cup \mathcal{C}_{\text{int}}$ is a set of cycles that are still pairwise vertex-disjoint while $\mathcal{C}'$ covers more vertices from $V_i$ than $\mathcal{C}$ does,*

*then, agent $i$ is said to* reject *the packing $\mathcal{C}$. Equivalently, we may say that agent $i$ specifically rejects the cycles in $\mathcal{C}_{\text{rej}}$ and we refer to $\mathcal{C}'$ as an* alternative packing *by which agent $i$ rejects.*

This definition captures that the rejecting agent $i$ could modify a given cycle packing $\mathcal{C}$ into another packing $\mathcal{C}'$ that benefits more of their own patients by only making changes pertaining to patients within their control. We call a packing that is not rejected by any agent *rejection-proof.* Then, a guarantee to always find such a solution also serves as a guarantee for agents that participation in the joint program is never detrimental to their personal interests. This led Blom et al. [6] to introduce the following problem, whose parameterized complexity we investigate in this work.

| | |
|---|---|
| REJECTION-PROOF KIDNEY EXCHANGE WITH $d$-CYCLES (RPKE-$d$) | |
| **Input:** | A directed graph $G$ where $V(G) = V_1 \cup \ldots \cup V_p$ is partitioned among $p$ agents, and an integer $k$. |
| **Task:** | Determine whether there is a *rejection-proof* set $\mathcal{C}$ of pairwise vertex-disjoint $d$-cycles such that $\mathcal{C}$ covers at least $k$ vertices. |

For ease of notation, we refer to any packing of pairwise vertex-disjoint $d$-cycles that covers at least $k$ vertices in an RPKE-$d$ instance as a *candidate solution.* Hence, a packing of $d$-cycles forms a solution to an instance if it is both a candidate solution and rejection-proof.

The above problem takes a leap in complexity compared to the single-agent problem KE-$d$. Whereas KE-$d$ is known to be NP-complete for all finite $d \geq 3$ [1], Blom et al. have shown RPKE-$d$ to be $\Sigma_2^P$-complete for $d \geq 3$, even when there are only two participating agents [6, Theorem 1]. In fact, it can be shown that checking whether a given agent rejects a given candidate solution is already NP-hard since this is at least as hard as checking whether a given packing is optimal in the NP-hard setting of only one agent. Partly motivated by this observation, we also consider a more restricted definition of rejections.

For some integer $c \geq 0$, we say that an agent *$c$-rejects* a packing $\mathcal{C}$ if the rejection criterion from Definition 1.1 can be satisfied with $|\mathcal{C}_{\text{rej}}| \leq c$. We call a packing that no agent $c$-rejects *$c$-rejection-proof.* Now, for each pair of positive integers $c$ and $d$ we define the problem $c$-REJECTION-PROOF KIDNEY EXCHANGE WITH $d$-CYCLES ($c$-RPKE-$d$) as the variant of the above RPKE-$d$ problem in which we aim to determine whether the input contains a candidate solution that is $c$-rejection-proof.

The integer $c$ in this definition indicates some level of complexity that a rejection can have. Additionally, it provides a way to interpolate between the problems KE-$d$ ($c = 0$) and RPKE-$d$ ($c = \infty$). More importantly, for any constant $c$, we observe that the task of determining whether a given agent $c$-rejects a given packing $\mathcal{C}$ of $d$-cycles is polynomial-time solvable when $d$ is also constant. This is witnessed by the fact that, if an agent $c$-rejects the packing $\mathcal{C}$, an alternative packing can always be obtained by rejecting a constant number of cycles and including a constant number of internal cycles. It follows that $c$-RPKE-$d$ is contained in NP when $c$ and $d$ are constants, as opposed to the $\Sigma_2^P$-complete RPKE-$d$ problem that does not limit the number of cycles that agents are allowed to reject.

**Our contribution.** We study the complexity of RPKE-$d$ and $c$-RPKE-$d$ parameterized by $k$, the number of vertices to be covered, and we present both positive and negative results. We prove our positive results on related set packing problems that generalize the two graph problems. This allows us to use some of the tools developed for such problems. Prior to our work, the only positive result known for RPKE-$d$ was a polynomial-time algorithm for $d = 2$ [8].

We give an algorithm that, for constant values of $d$, solves RPKE-$d$ in $2^{\mathcal{O}(k \log k)} + n^{\mathcal{O}(1)}$ time on $n$-vertex graphs. It consists of two parts, the first of which is a kernelization algorithm. It takes an arbitrary instance as input and outputs an equivalent instance of size $\mathcal{O}\left((k^{d+1})^d\right)$ in polynomial time, via a careful agent-aware application of the sunflower lemma [12] to eliminate transplantation cycles. In set packing problems, removing sets during preprocessing typically comes at the risk of removing sets that are required to make an optimal solution, thereby possibly transforming YES-instances into NO-instances. The sunflower lemma provides a tool to circumvent this. Our rejection-proof problem settings introduce an additional type of behavior that needs care: transplantation cycles that do not belong to optimal solutions may still play a role in rejections. Removing the wrong cycles may cause an agent not to reject a certain candidate solution that it otherwise would have. We present a two-step reduction approach guaranteed to preserve the answer. The second part of the algorithm consists of brute-force on the reduced instance.

We give a reduction from Subgraph Isomorphism to show that, under the Exponential Time Hypothesis (ETH), the algorithm is asymptotically optimal in terms of $n$ and $k$. We even show the stronger bound (independent of $k$) that a $2^{o(n \log n)}$ time algorithm for RPKE-$d$ or $c$-RPKE-$d$ would violate the ETH for any $c \geq 2$ and $d \geq 3$. We exploit the mechanism of rejections to facilitate our reduction from the Subgraph Isomorphism problem, for which such a superexponential ETH-based lower bound is known [10]. This gives the rejection-proof kidney exchange problems a similar complexity status to other problems in the literature, such as Metric Dimension and Geodetic Set parameterized by vertex cover number, for which a brute-force algorithm on a polynomial kernel is also ETH-optimal [14].

To complement the ETH-hardness of $c$-RPKE-$d$ with $c \geq 2$, we give a $3^n \cdot n^{\mathcal{O}(1)}$ time algorithm for the problem when $c = 1$. This allows us to see surprising jumps in the algorithmic complexity of $c$-RPKE-$d$ for varying values of $c$. While no big difference in complexity is obtained between $c = 1$ and $c = 0$ (in which case the problem can be solved in $2^n \cdot n^{\mathcal{O}(1)}$ time with a standard application of dynamic programming over subsets [10, Chapter 6]), our results show that the problem does become computationally harder from a parameterized perspective when $c = 2$ (assuming the ETH). We also see that no more jumps in parameterized complexity appear for larger values of $c$. The surprising aspect here is that this one discrete jump in the parameterized complexity of $c$-RPKE-$d$ does not coincide with the jump in the classical complexity of the problem between being contained in NP vs. not being contained in NP: we have seen that, for constant $d \geq 3$, the $c$-RPKE-$d$ problem is contained in NP for *every* constant value of $c$, while the RPKE-$d$ problem is $\Sigma_2^P$-complete.

Another jump in complexity is observed when looking at the number of vertices that are *not* covered by a solution. When asked to find a solution that covers all vertices in the input, the problem again collapses to the NP-complete packing problem KE-$d$. However, our ETH-hardness proof shows that finding a solution that covers all but one vertex is essentially as difficult as the general problem in terms of running time, assuming the ETH. Combined, our results yield a good understanding of what makes rejection-proof kidney exchange hard.

**Organization.** The remainder of this work is organized as follows. We start with some preliminaries in Section 2. Next, we give a polynomial kernel and an FPT algorithm for a set packing generalization of RPKE-$d$ in Section 3, and we present a matching ETH-lower bound in Section 4. In Section 5, we give a single-exponential algorithm under 1-rejections and we conclude in Section 6. The proofs of statements marked (★) are deferred to the full version of this paper.

## 2 Preliminaries

All graphs considered are directed, unless stated otherwise. We use standard notation for graphs and concepts from parameterized complexity; see the textbook by Cygan et al. [11]. We write $\langle v_1, \ldots, v_\ell \rangle$ to denote the directed cycle with edge set $\{(v_1, v_2), \ldots, (v_{\ell-1}, v_\ell), (v_\ell, v_1)\}$. We use the term *packing* to refer to a collection of sets that satisfies certain pairwise disjointness conditions, such as collections of vertex-disjoint cycles in a graph or collections of pairwise-disjoint subsets of a universe. Since the disjointness conditions differ depending on the context, we always specify which condition holds for the packing at hand.

A set system is a pair $(U, \mathcal{S})$ where $U$ is a universe of elements and $\mathcal{S} \subseteq 2^U$ is a collection of subsets of this universe. For a subset $U' \subseteq U$, we use the notation $\mathcal{S}[U']$ to denote the collection of sets from $\mathcal{S}$ that contain only elements of $U'$. A *hitting set* of $\mathcal{S}$ is a subset of $U$ that intersects each set $S \in \mathcal{S}$. We use the following folklore lemma.

▶ **Lemma 2.1** (★). *For a set system $(U, \mathcal{S})$, it can be checked in $2^{|U|} \cdot (|U| + |\mathcal{S}|)^{\mathcal{O}(1)}$ time whether $\mathcal{S}$ contains a collection of pairwise disjoint sets that covers all elements of $U$.*

Next, we present the sunflower lemma by Erdős and Rado [12] and a related definition of *sunflowers* in set systems. In a set system $(U, \mathcal{S})$, a sunflower $\mathcal{F}$ with $z$ petals and *core* $Y$ is a collection of sets $S_1, \ldots, S_z \in \mathcal{S}$ such that $S_i \cap S_j = Y$ for all $i \neq j$. We refer to a set $S_i \setminus Y$ as the *petal* corresponding to set $S_i$ and we require all petals to be non-empty. We state a variant of it as proven in the textbook by Flum and Grohe [13, page 212].

▶ **Lemma 2.2** (Sunflower lemma). *Let $(U, \mathcal{S})$ be a set system, such that each set in $\mathcal{S}$ has size at most $d$. If $|\mathcal{S}| > d \cdot d!(z-1)^d$, then the set system contains a sunflower with $z$ petals and such a sunflower can be computed in time polynomial in $|U|$, $|\mathcal{S}|$, and $z$.*

The following lemma follows from the definitions of sunflowers and hitting sets.

▶ **Lemma 2.3** (★). *Let $(U, \mathcal{S})$ be a set system with sets of size $\leq d$ for which there exists a hitting set of size $\leq h$, for some integers $d, h$. Let $\mathcal{X} \subseteq \mathcal{S}$ be a collection of disjoint sets and let $\mathcal{F} \subseteq \mathcal{S}$ be a sunflower of size $d(h-1)+2$ that contains one of the sets $X \in \mathcal{X}$. Then, there is a set $X' \in \mathcal{F} \setminus \{X\}$ such that $(\mathcal{X} \setminus X) \cup \{X'\}$ is a collection of pairwise disjoint sets.*

When analyzing the running time of an algorithm whose input is a graph, we denote the number of vertices and edges in the input graph by $n$ and $m$, respectively. For algorithms operating on a set system, we use $n$ to denote the number of elements in the universe and $m$ for the total number of sets in the system.

**Rejection-proof set packing.** As mentioned, we present our positive results for problems that generalize the kidney exchange problems defined in the introduction. Intuitively, one may think of translating an RPKE-$d$ instance on a graph $G$ into this generalized setting by constructing a set system $(U, \mathcal{S})$ with $U = V(G)$ and including a set in $\mathcal{S}$ for every vertex set that forms a $d$-cycle in $G$. Related definitions follow analogously, but we briefly reiterate and name them below. To represent $p$ different agents in a set system $(U, \mathcal{S})$, the universe $U = U_1 \cup \ldots \cup U_p$ is partitioned into $p$ sets. A set $S \in \mathcal{S}$ is called *$i$-internal* if $S \subseteq U_i$.

▶ **Definition 2.4.** *Let $(U, \mathcal{S})$ be a set system such that the universe $U = U_1 \cup \ldots \cup U_p$ is partitioned over $p$ different agents. Let $\mathcal{X} \subseteq \mathcal{S}$ be a packing of pairwise disjoint sets. Suppose there is an agent $i \in [p]$ for which there exist:*

- *a subset $\mathcal{X}_{\text{rej}} \subseteq \mathcal{X}$, and*
- *a set $\mathcal{X}_{\text{int}}$ of $i$-internal sets such that $\mathcal{X}' := (\mathcal{X} \setminus \mathcal{X}_{\text{rej}}) \cup \mathcal{X}_{\text{int}}$ is a collection of sets that are still pairwise disjoint, while $\mathcal{X}'$ covers more elements from $U_i$ than $\mathcal{X}$ does,*

*then, agent $i$ is said to* reject *the packing $\mathcal{X}$. Equivalently, we may say that agent $i$ specifically rejects the sets in $\mathcal{X}_{\text{rej}}$ and we refer to $\mathcal{X}'$ as an alternative packing by which agent $i$ rejects.*

We call a packing that is not rejected by any agent *rejection-proof*, and we use this notation to introduce the following generalized problem statement.

REJECTION-PROOF $d$-SET PACKING (RP-$d$-SP)

| | |
|---|---|
| **Input:** | A universe $U = U_1 \cup \ldots \cup U_p$ that is partitioned over $p$ agents, a collection $\mathcal{S} \subseteq 2^U$ of sets of size $\leq d$, and an integer $k$. |
| **Task:** | Determine whether there is a rejection-proof collection $\mathcal{X} \subseteq \mathcal{S}$ of disjoint sets that covers at least $k$ elements from $U$. |

Like in the graph setting, we call any packing of pairwise disjoint sets that cover at least $k$ vertices a *candidate solution*. Further, for some integer $c \geq 0$, we say that an agent *$c$-rejects* a given packing $\mathcal{X}$ if the rejection criterion from Definition 2.4 can be satisfied with $|\mathcal{X}_{\text{rej}}| \leq c$. A packing that no agent $c$-rejects is called *$c$-rejection-proof*. We define $c$-REJECTION-PROOF $d$-SET PACKING ($c$-RP-$d$-SP) as a problem with the same input as RP-$d$-SP, but in which the goal is to determine whether the input contains a $c$-rejection-proof candidate solution.

**Exponential Time Hypothesis (ETH).** In Section 4, we derive a hardness result that is conditional on the well-known ETH. This is a technical conjecture implying that the satisfiability of Boolean 3-CNF formulas cannot be checked in subexponential time [17]. Specifically, we use a known hardness result for the SUBGRAPH ISOMORPHISM problem conditional on the ETH. This problem takes two undirected graphs $G$ and $H$ as input and asks whether $H$ is isomorphic to a subgraph of $G$. More formally, the problem asks whether there is an injective function $\varphi : V(H) \to V(G)$ such that $\{u, v\} \in E(H)$ implies $\{\varphi(u), \varphi(v)\} \in E(G)$. Such a function is called a subgraph isomorphism from $H$ to $G$. The following conditional lower bound on the computational complexity of this problem is known.

▶ **Lemma 2.5** ([10]). *Assuming the ETH, no algorithm exists that determines in $2^{o(n_G \log n_G)}$ time whether an undirected graph $H$ is isomorphic to a subgraph of $G$, where $n_G = |V(G)|$.*

## 3 A kernelization and FPT algorithm for rejection-proof packings

In this section, we present a polynomial kernel and a derived FPT algorithm for RP-$d$-SP, which translates directly to RPKE-$d$ as explained in Section 2. We start with two simple statements that are useful in proving the correctness of our kernel.

▶ **Lemma 3.1.** *Let $I_1 = (U_1, \dots, U_p, \mathcal{S}, k)$ be an RP-d-SP instance, let $\mathcal{X} \subseteq \mathcal{S}$ be a solution for it and let $X \in \mathcal{S}$. If $X \notin \mathcal{X}$, then $\mathcal{S}$ is a solution for the instance $I_2 = (U_1, \dots, U_p, \mathcal{S} \setminus \{X\}, k)$.*

**Proof.** Clearly, $\mathcal{X}$ is also a candidate solution in $I_2$. Now, suppose for contradiction that some agent $i$ rejects $\mathcal{X}$ by proposing an alternative packing $(\mathcal{X} \setminus \mathcal{X}_{\text{rej}}) \cup \mathcal{X}_{\text{int}}$. This alternative packing can also be constructed in $I_1$, contradicting that $\mathcal{X}$ is rejection-proof in $I_1$. ◀

▶ **Observation 3.2.** *Let $S_i = \bigcup_{S \in \mathcal{S}[U_i]} S$ denote the elements which are contained in $i$-internal sets for some agent $i \in [p]$. If for collections $\mathcal{X}$ and $\mathcal{X}'$ it holds that $\{X \in \mathcal{X} \mid X \cap S_i \neq \emptyset\} = \{X' \in \mathcal{X}' \mid X' \cap S_i \neq \emptyset\}$, then $i$ rejects $\mathcal{X}$ if and only if $i$ rejects $\mathcal{X}'$.*

The above observation captures that an agent $i$ cannot gain anything by rejecting a set that is not intersected by an $i$-internal set. Thus, we can assume they will not reject such sets. $\mathcal{X}$ and $\mathcal{X}'$ become identical after removing these sets, so the observation is correct.

Now, we present two reduction rules that act on an input instance $(U_1, \dots, U_p, \mathcal{S}, k)$ of RP-$d$-SP and prove that they are safe. A reduction rule is *safe* if its application preserves the YES/NO answer to the decision problem. The first rule aims to reduce the number of internal sets.

▶ **Reduction rule 1.** *Let $\mathcal{F} \subseteq \mathcal{S}[U_i]$, for some $i \in [p]$, be a sunflower of size $d(k \cdot d - 1) + 2$. Remove a set $F \in \mathcal{F}$ of minimum size.*

▶ **Lemma 3.3.** *Reduction rule 1 is safe if $\mathcal{S}$ has a hitting set of size $\leq k \cdot d$.*

**Proof.** Let $I = (U_1, \dots, U_p, \mathcal{S}, k)$ be an instance of RP-$d$-SP where there exists some $i \in [p]$ and sunflower $\mathcal{F} \subseteq \mathcal{S}[U_i]$ of size $d(k \cdot d - 1) + 2$ with core $Y$. Let $S$ be the set of minimum size in $\mathcal{F}$ that is removed by Reduction rule 1 to create the instance $I' = (U_1, \dots, U_p, \mathcal{S} \setminus \{S\}, k)$. We will now argue that $I$ is a YES-instance if and only if $I'$ is a YES-instance.

**(⇒).** Assume $I$ has a solution $\mathcal{X}$. We claim there is a solution for $I$ that does not use $S$. If $S \notin \mathcal{X}$, our claim holds. Suppose $S \in \mathcal{X}$. Invoking Lemma 2.3 with $h = k \cdot d$ yields that a $S' \in \mathcal{F}$ exists such that the sets in $\mathcal{X}' = (\mathcal{X} \setminus \{S\}) \cup \{S'\}$ are pairwise disjoint. Since $S$ is a minimum-size set in $\mathcal{F}$, it follows that $|S'| \geq |S|$, so the collection $\mathcal{X}'$ covers at least $k$ elements. To show that $\mathcal{X}'$ is rejection-proof, suppose for contradiction that some agent $j$ rejects $\mathcal{X}'$ by proposing the alternate packing $(\mathcal{X}' \setminus \mathcal{X}'_{\text{rej}}) \cup \mathcal{X}'_{\text{int}}$. We distinguish three cases:

- $j \neq i$. Since $S \in \mathcal{S}[U_i]$, each set $J \in \mathcal{S}$ with $J \cap U_j \neq \emptyset$ is contained in $\mathcal{X}'$ if and only if it is contained in $\mathcal{X}$. From Observation 3.2 it follows that agent $j$ would reject $\mathcal{X}$.
- $j = i$ and $S' \in \mathcal{X}'_{\text{rej}}$. Let $\mathcal{X}_{\text{rej}} = (\mathcal{X}'_{\text{rej}} \setminus \{S'\}) \cup \{S\}$ and note that $\mathcal{X}' \setminus \mathcal{X}'_{\text{rej}} = \mathcal{X} \setminus \mathcal{X}_{\text{rej}}$. By the definition of rejection $(\mathcal{X} \setminus \mathcal{X}_{\text{rej}}) \cup \mathcal{X}'_{\text{int}}$ is a collection of disjoint sets. Both $S$ and $S'$ are $j$-internal and $|S| \leq |S'|$, hence $\mathcal{X}$ covers at most as many elements of $U_j$ as $\mathcal{X}'$ does. By definition of rejection, $(\mathcal{X}' \setminus \mathcal{X}'_{\text{rej}}) \cup \mathcal{X}'_{\text{int}} = (\mathcal{X} \setminus \mathcal{X}_{\text{rej}}) \cup \mathcal{X}'_{\text{int}}$ covers more elements from $U_j$ than $\mathcal{X}'$ does, thus also more than $\mathcal{X}$ covers. This would mean that $\mathcal{X}$ would be rejected by agent $j$ by rejecting the sets $\mathcal{X}_{\text{rej}} \subseteq \mathcal{X}$ and replacing these with the sets $\mathcal{X}'_{\text{int}}$.
- $j = i$ and $S' \notin \mathcal{X}'_{\text{rej}}$. Let $\mathcal{X}_{\text{rej}} = \mathcal{X}'_{\text{rej}} \cup \{S\}$ and $\mathcal{X}_{\text{int}} = \mathcal{X}'_{\text{int}} \cup \{S'\}$. Observe that $\mathcal{X}_{\text{int}} \subseteq \mathcal{S}[U_j]$ and $(\mathcal{X} \setminus \mathcal{X}_{\text{rej}}) \cup \mathcal{X}_{\text{int}} = (\mathcal{X}' \setminus \mathcal{X}'_{\text{rej}}) \cup \mathcal{X}'_{\text{int}}$. Hence agent $j$ would reject $\mathcal{X}$.

As all cases contradict the assumption that $\mathcal{X}$ is rejection-proof, $\mathcal{X}'$ is a solution for $I$ that does not contain $S$. Now Lemma 3.1 implies that $I'$ has a solution and is a YES-instance.

**(⇐).** Assume $I'$ has a solution $\mathcal{X}'$. We claim that $\mathcal{X}'$ is also a solution for the instance $I$. It suffices to show that no rejection can take place where $S \in \mathcal{X}_{\text{int}}$, as the other requirements for $\mathcal{X}'$ to be a solution for $I$ follow directly from $\mathcal{X}'$ being a solution for $I'$.

Assume, for sake of contradiction, that agent $i$ rejects $\mathcal{X}'$ by rejecting $\mathcal{X}_{\text{rej}}$ and replacing these sets with $\mathcal{X}_{\text{int}}$, where $S \in \mathcal{X}_{\text{int}}$. Because $(\mathcal{X}' \setminus \mathcal{X}_{\text{rej}}) \cap \mathcal{X}_{\text{int}}$ is a collection of disjoint sets and $S \in \mathcal{F}$ it follows from Lemma 2.3 with $h = k \cdot d$ that there exists some set $S' \in \mathcal{F}$ such that $(((\mathcal{X}' \setminus \mathcal{X}_{\text{rej}}) \cap \mathcal{X}_{\text{int}}) \setminus \{S\}) \cup \{S'\}$ is a collection of disjoint sets. Note that $|S'| \geq |S|$ and $S, S' \in \mathcal{S}[U_i]$. Hence agent $i$ can also reject $\mathcal{X}'$ as a solution for $I$ by rejecting $\mathcal{X}_{\text{rej}}$ and replacing these sets with $(\mathcal{X}_{\text{int}} \setminus \{S\}) \cup \{S'\}$. Similarly, agent $i$ can reject $\mathcal{X}'$ as a solution for $I'$; a contradiction. Hence $\mathcal{X}'$ is a solution for $I$, which means $I$ is a YES-instance. ◀

Observe that in the forward direction of the proof we increase the number of sets that are rejected by some agent $j$. This is possible because an agent can reject an unbounded number of sets in RPKE-$d$. The proof as written does not work in the setting of $c$-RPKE-$d$. Our second reduction rule aims to reduce the number of sets that are not internal.

▶ **Reduction rule 2.** *Let $\mathcal{F} \subseteq \mathcal{S}$ be a sunflower with core $Y$ of size $d(k \cdot d - 1) + 2$ such that:*

- *No petal $F \setminus Y$, for $F \in \mathcal{F}$, is intersected by an internal set.*
- *For each agent $i$ that has an internal set that intersects $Y$, for all $F, F' \in \mathcal{F}$:*

$$|F \cap U_i| = |F' \cap U_i|.$$

*Remove a set $F \in \mathcal{F}$ of minimum size.*

▶ **Lemma 3.4 (★).** *Reduction rule 2 is safe if $\mathcal{S}$ has a hitting set of size $\leq k \cdot d$.*

The proof of the lemma follows the same structure as the proof of Lemma 3.3. The restrictions placed on the sunflower used in Reduction rule 2 together with Lemma 3.1 can be used to prove one direction of the proof. The other direction follows from the fact that any removed set is not internal and can thus not be used as a replacement set in a rejection.

To bound the kernel size we define $f_d(k) = d \cdot d!(d(k \cdot d - 1) + 1)^d$ and $g_d(k) = d \cdot d! \left(d^{d-1} \cdot (d(k \cdot d - 1) + 1) + d \cdot k \cdot f_d(k)\right)^d$. Note that $g_d(k) = \mathcal{O}\left(\left(k^{d+1}\right)^d\right)$ for fixed $d$.

▶ **Lemma 3.5 (★).** *If for an instance $I = (U_1, \ldots, U_p, \mathcal{S}, k)$ of RP-d-SP it holds that $|\mathcal{S}| \geq g_d(k)$, then one of the reduction rules can be applied to $I$ in polynomial time, or it can be concluded in polynomial time that $I$ is a YES-instance.*

**Proof sketch.** It can easily be shown that an instance in which at least $k$ agents have an internal set is a YES-instance. Else, if the instance contains at least $k \cdot f_d(k)$ internal sets, we can use Lemma 2.2 to show that Reduction rule 1 can be applied in polynomial time. Otherwise, the number of internal sets is bounded, and we show that Reduction rule 2 can be applied. In that case, the number of petals in a sunflower that are intersected by internal sets is also bounded and the core of a sunflower is intersected by internal sets from at most $d$ agents. Since $|\mathcal{S}| \geq g_d(k)$, Lemma 2.2 yields a sufficiently large sunflower such that some of its sets have the same number of elements from each agent with an internal set intersecting the core. Some subset of this is a sunflower meeting the prerequisites from Reduction rule 2 and this can be found in polynomial time. ◀

Now, the above insights can be used to give the kernel. Given an RP-$d$-SP instance $I$, it starts by greedily computing an inclusion-wise maximal packing $\mathcal{X}$ of disjoint sets. In the full version, we show that $I$ is a YES-instance if $|\mathcal{X}| > k$. Otherwise, the union of all sets in $\mathcal{X}$ forms a hitting set of size $\leq k \cdot d$, meaning that Reduction rules 1 and 2 are safe to apply. Lemma 3.5 shows that at least one of them is applicable if $|\mathcal{X}| \geq g_d(k)$ and we repeatedly apply one until an instance is obtained with at most $g_d(k)$ sets. This proves the following.

▶ **Theorem 3.6** (★). *There is a polynomial-time algorithm that takes an RP-d-SP instance as input and either correctly determines that this is a YES-instance or returns an equivalent RP-d-SP instance with at most $g_d(k)$ sets and a hitting set of size at most $k \cdot d$. Hence, for constant d, RP-d-SP admits a kernel of size $g_d(k) = \mathcal{O}\left(\left(k^{d+1}\right)^d\right)$.*

This kernel can be used as a subroutine to solve RP-$d$-SP. Since it outputs an equivalent instance with a bounded hitting set, the number of collections of pairwise disjoint sets is also bounded. This limits the number of candidate solutions and alternative packings that a brute-force algorithm needs to consider, and it allows us to prove the following statement.

▶ **Theorem 3.7** (★). *RP-d-SP is solvable in $2^{\mathcal{O}(k \log k)} + (n+m)^{\mathcal{O}(1)}$ time for each constant d.*

## 4 A tight ETH lower bound for rejection-proof kidney exchange

We prove that the algorithm presented in Theorem 3.7 is asymptotically optimal to solve RP-$d$-SP under the ETH for any $d \geq 3$. More strongly, we show that no faster algorithm exists for the problem RPKE-3 under the ETH, even when the input contains only two agents. To see that this is a stronger result, note that when $d \geq 3$ the RP-$d$-SP problem strictly generalizes RPKE-3. The construction also yields a lower bound for $c$-RPKE-3 for every $c \geq 2$. We use the ETH lower bound on Subgraph Isomorphism from Lemma 2.5 as starting point.

▶ **Lemma 4.1.** *Assuming the ETH, there is no integer $c \geq 2$ such that c-RPKE-3 is solvable in $2^{o(n \log n)}$ time, even when the input contains only two agents. The same lower bound applies to RPKE-3 with two agents.*

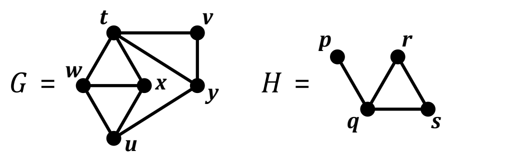
**Proof.** Let $c \geq 2$ be arbitrary. We prove the statement by a reduction from Subgraph Isomorphism. The resulting instance can serve as an input to both RPKE-3 and $c$-RPKE-3. First, we give the reduction after which we prove it is correct for both problems. A visual example of the reduction can be seen in Figure 2.

Let $(G, H)$ be an input for the Subgraph Isomorphism problem consisting of two graphs and let $n_G := |V(G)|$ and $n_H := |V(H)|$. We assume that $n_H \leq n_G$, as $(G, H)$ would trivially be a NO-instance otherwise. Now, we give our reduction by showing how to modify the input $(G, H)$ to an input $(G', V_1', V_2', k)$ with two agents for RPKE-3 and $c$-RPKE-3.

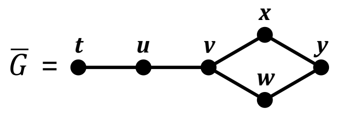
We start by defining the vertex set of $G'$. To define the complete vertex set, it is useful to think of it as six separate parts: five sets $W_1, W_2, W_3, W_4$, and $W_5$, and a separate vertex $z$. First, for every vertex $u \in V(H)$, we add a vertex $u_1$ to $W_1$ and a vertex $u_2$ to $W_2$. Next, for every vertex $x \in V(G)$, we add a vertex $x_3$ to $W_3$. Then, for every integer $i \in [n_G - n_H]$ we add a vertex $i_4$ to $W_4$ and a vertex $i_5$ to $W_5$. Finally, we add the additional vertex $z$ to the graph. The vertex set is partitioned over the two agents by defining $V_1' := W_1 \cup W_3 \cup W_5 \cup \{z\}$ and $V_2' := W_2 \cup W_4$. Remark that $|W_1 \cup W_4| = |W_2 \cup W_5| = |W_3| = n_G$, so that $|V(G')| = 3n_G + 1$.

Next, we define the edge set of $G'$. Although $G$ and $H$ are undirected graphs, recall that $G'$, being part of a kidney exchange instance, is a directed graph.

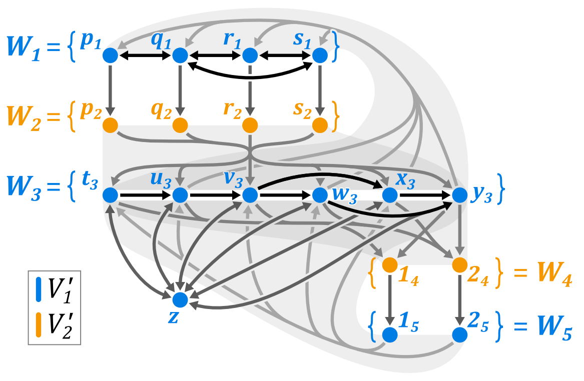
1. For every $\{u, v\} \in E(H)$, we add $(u_1, v_1)$ and $(v_1, u_1)$ to $E(G')$.
2. For every $u \in V(H)$, we add $(u_1, u_2)$ to $E(G')$.
3. For every $u \in V(H)$ and every $x \in V(G)$, we add $(u_2, x_3)$ and $(x_3, u_1)$ to $E(G')$.
4. For every $i \in [n_G - n_H]$, we add $(i_4, i_5)$ to $E(G')$.
5. For every $x \in V(G)$ and every $i \in [n_G - n_H]$, we add $(x_3, i_4)$ and $(i_5, x_3)$ to $E(G')$.
6. For every $x \in V(G)$, we add $(x_3, z)$ and $(z, x_3)$ to $E(G')$.
7. We fix an arbitrary ordering $\prec$ on $V(G)$. Then, for every $x, y \in V(G)$ such that $\{x, y\} \notin E(G)$ and $x \prec y$, we add $(x_3, y_3)$ to $E(G')$.



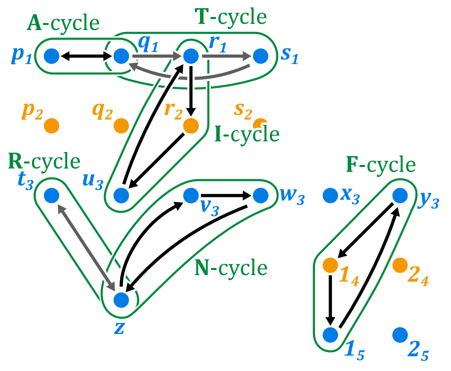
**(a)** An instance $(G, H)$ of Subgraph Isomorphism as example input for the reduction.



**(b)** The complement of $G$, which is implicitly used to construct $G'$.



**(c)** Resulting graph $G'$ obtained by running the reduction on the example input shown in Figure 2a.



**(d)** A simplified representation of $G'$ indicating each of the different classes of cycles.

**Figure 2** The reduction applied to an example input. For the sake of visualization, not all edges in the graph $G'$ are shown. Four (partially overlapping) areas shaded in gray indicate that all vertices of $W_2$ and $W_5$ each have outgoing edges to all vertices in $W_3$ and that all vertices of $W_3$ have outgoing edges to all vertices of $W_1$ and $W_4$. Vertex colors are used to indicate the partition of the vertex set over the two agents with blue vertices belonging to $V'_1$ and orange vertices to $V'_2$.

Finally, we define $k = 3 \cdot n_G$. Note that this is one less than the total number of vertices in $G'$, denoted as $n := |V(G')| = 3n_G + 1$. This concludes the description of the reduction.

We distinguish six different classes of 3-cycles that live in $G'$. Below, we describe for each of these classes what their intuitive role is in the reduction and we indicate each class of cycles with a letter to refer to them more easily later in the proof. See also Figure 2d.

- For every edge $\{u, v\} \in E(H)$, the graph $G'$ contains the cycle $\langle u_1, v_1 \rangle$. We call such cycles A-cycles for encoding the **a**djacencies in $H$.
- For every triangle in $H$ on vertices $u$, $v$, and $w$, the graph $G'$ contains both orientations of the directed cycle on vertices $u_1$, $v_1$ and $w_1$. We call such cycles T-cycles due to their correspondence to **t**riangles in the graph $H$. These cycles originate merely as a byproduct of constructing the A-cycles and serve no explicit purpose in our reduction.
- For every $u \in V(H)$ and every $x \in V(G)$, the graph $G'$ contains the cycle $\langle u_1, u_2, x_3 \rangle$. We call such cycles I-cycles, as they are used to encode subgraph **i**somorphisms from $H$ to $G$: including a cycle $\langle u_1, u_2, x_3 \rangle$ will correspond to mapping $u \in V(H)$ to $x \in V(G)$ in a subgraph isomorphism. See also Figure 3 (left image) for an example.
- For every $x \in V(G)$ and every $i \in [n_G - n_H]$, the graph $G'$ contains the cycle $\langle x_3, i_4, i_5 \rangle$. We call such cycles F-cycles as they are used in the construction of optimal solutions as **f**iller cycles to cover vertices in $W_3$ that are not already covered by I-cycles.
- For every $x, y \in V(G)$ with $\{x, y\} \notin E(G)$, the graph $G'$ contains the cycle $\langle x_3, y_3, z \rangle$ (or $\langle y_3, x_3, z \rangle$ depending on whether $x \prec y$). We call such cycles N-cycles for encoding the **n**on-adjacencies in $G$.

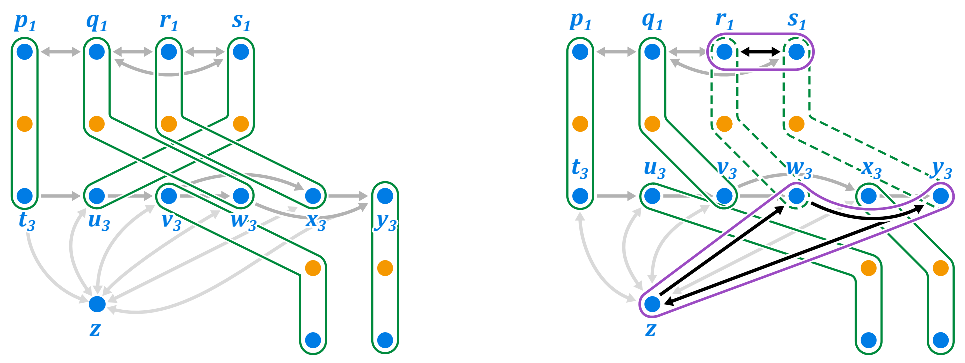


**Figure 3** Both figures depict the same graph $G'$ as in Figure 2, each with a different packing of cycles shown in green. For readability, only some edges are shown. The cycle packing on the left encodes the subgraph isomorphism from $H$ to $G$ that maps $p$ to $t$, $q$ to $w$, $r$ to $x$, and $s$ to $u$. The green packing on the right encodes a function that maps $r$ to $w$ and $s$ to $y$. This is not a subgraph isomorphism because $\{r, s\} \in E(H)$ and $\{w, y\} \notin E(G)$. Hence, agent 1 can reject the green packing by proposing an alternative one in which the dashed cycles are replaced by the purple cycles.

- Finally, for every $x \in V(G)$, the graph $G'$ contains the cycle $\langle x_3, z\rangle$. Much like T-cycles, these cycles serve no explicit purpose in our reduction and they originate merely as byproduct of constructing the N-cycles. We call them R-cycles, since optimal rejection-proof solutions typically do not include any of them, making these cycles mostly **r**edundant.

Before proving the correctness of our reduction, observe that $G'$ does not contain any 3-cycles other than those contained in the six classes described above. In particular, note that $G'[W_3]$ is a DAG as the obvious extension of $\prec$ to $W_3$ forms a topological ordering. Finally, cycles whose length is greater than 3 will not be relevant in our proof, since RPKE-3 and $c$-RPKE-3 only deal with 3-cycles by definition.

Most of the remainder of this proof is used to show the correctness of the reduction. We do this by proving that $H$ is isomorphic to a subgraph of $G$ if and only if $G'$ has a ($c$-)rejection-proof packing of 3-cycles that covers at least $k$ vertices. Below, we prove the two directions of this bi-implication separately.

▷ Claim 4.2. If $G'$ contains a 2-rejection-proof packing of 3-cycles that covers at least $k = 3 \cdot n_G$ vertices (e.g. because this packing is even (completely) rejection-proof), then $H$ is isomorphic to a subgraph of $G$.

Proof. Let $\mathcal{C}$ be a 2-rejection-proof packing of 3-cycles in $G'$ that covers at least $k = 3 \cdot n_G$ vertices. Before showing how to construct from $\mathcal{C}$ a subgraph isomorphism from $H$ to $G$, we derive some structural properties of $\mathcal{C}$. We start by observing that at most one vertex from $G'$ is not covered by $\mathcal{C}$. We use this property to show that $\mathcal{C}$ must contain $n_H$ I-cycles.

First, we argue that the number of I-cycles in $\mathcal{C}$ must be more than $n_H - 2$. Since vertices from $W_2$ only appear in I-cycles and each I-cycle contains exactly one of these vertices, at least two vertices from $W_2$ would be uncovered by $\mathcal{C}$ otherwise. This would contradict the assumption that $\mathcal{C}$ leaves at most one vertex uncovered.

Next, we argue that the number of I-cycles in $\mathcal{C}$ cannot be exactly $n_H - 1$. In that case exactly one vertex $u_1$ from $W_1$ and exactly one vertex $v_2$ from $W_2$ would not be covered by an I-cycle in $\mathcal{C}$. If $v_2$ is not covered by an I-cycle in $\mathcal{C}$, this means that it is not covered at all by $\mathcal{C}$. Furthermore, $u_1$ only appears in I-cycles, A-cycles, and T-cycles. By assumption, it is not covered by an I-cycle in $\mathcal{C}$. Moreover, it cannot be covered by an A-cycle or T-cycle in $\mathcal{C}$ either: such cycles contain at least two vertices from $W_1$, but $u_1$ is the only vertex in $W_1$ that is not covered by an I-cycle in $\mathcal{C}$. As such, $\mathcal{C}$ does not contain A-cycles or T-cycles since these would not be vertex-disjoint from all I-cycles in $\mathcal{C}$. This means that neither $v_2$ nor $u_1$ is covered by $\mathcal{C}$, contradicting that $\mathcal{C}$ leaves at most one vertex uncovered.

By arguing that the number of I-cycles in $\mathcal{C}$ must be strictly more than $n_H - 2$ but not exactly $n_H - 1$, we have shown that the number of I-cycles in $\mathcal{C}$ is exactly $n_H$. This means that all vertices from $W_1$ and $W_2$ are covered by I-cycles in $\mathcal{C}$.

Next, we see that $\mathcal{C}$ must include $n_G - n_H$ F-cycles. If it were to contain less F-cycles, at least one vertex from $W_4$ and $W_5$ each would not be covered by F-cycles. Since vertices from these two sets appear only in F-cycles, this would mean that they are not covered by $\mathcal{C}$ at all. In turn, this would contradict the observation that leaves $\mathcal{C}$ at most one vertex uncovered.

By including $n_H$ I-cycles and $(n_G - n_H)$ F-cycles, all vertices of $G'$ except $z$ are covered. Since every cycle contains a vertex different from $z$, the packing $\mathcal{C}$ does not contain any other cycles. Knowing the structure of $\mathcal{C}$ allows us to construct a subgraph isomorphism $\varphi : V(H) \to V(G)$.

Let $u \in V(H)$ be arbitrary, and let $C_u \in \mathcal{C}$ be the unique I-cycle from $\mathcal{C}$ that contains $u_1$. Since every vertex from $W_1$ is contained in exactly one I-cycle in $\mathcal{C}$, this is well-defined. Let $x_3 \in W_3$ be the unique vertex from $W_3$ in $C_u$. Then, we define $\varphi(u) = x$. To complete the construction of $\varphi$, we do the same for every $u \in V(H)$.

This construction yields a function $\varphi$ that maps every vertex in $H$ to a unique vertex in $G$. To show that $\varphi$ is in particular a subgraph isomorphism, it remains to show that for every $\{u, v\} \in E(H)$ it holds that $\{\varphi(u), \varphi(v)\} \in E(G)$. Suppose for contradiction that there is some $\{u, v\} \in E(H)$ for which this is not the case, i.e. $\{\varphi(u), \varphi(v)\} \notin E(G)$. We show that agent 1 would then reject $\mathcal{C}$. The right image in Figure 3 shows an example of this situation.

Let $x = \varphi(u)$ and $y = \varphi(v)$. Assume w.l.o.g. that $x \prec y$ in the arbitrary ordering fixed onto $V(G)$ in step 7 of the construction of $E(G')$. Then the following cycles exist:

- Because $\{u, v\} \in E(H)$, the A-cycle $\langle u_1, v_1 \rangle$ is added to $G'$ in construction step 1.
- Because $\{x, y\} \notin E(G)$, the N-cycle $\langle x_3, y_3, z \rangle$ is added to $G'$ in steps 6 and 7.

Now, agent 1 can reject the solution by proposing an alternative packing to $\mathcal{C}$ in which $\langle u_1, u_2, x_3 \rangle$ and $\langle v_1, v_2, y_3 \rangle$ are removed and the cycles $\langle u_1, v_1 \rangle$ and $\langle x_3, y_3, z \rangle$ are added.

This modification to $\mathcal{C}$ yields a packing of cycles that are still pairwise disjoint and which cover all vertices of $V_1'$. Moreover, the two cycles added by agent 1 are indeed 1-internal cycles in the graph, which means that agent 1 will indeed reject $\mathcal{C}$ whenever there is a $\{u, v\} \in E(H)$ such that $\{\varphi(u), \varphi(v)\} \notin E(G)$. This contradicts the assumption that $\mathcal{C}$ is 2-rejection-proof, so we conclude that $\varphi$ is a valid subgraph isomorphism from $H$ to $G$. ◁

▷ Claim 4.3 (★). If $H$ is isomorphic to a subgraph of $G$, then $G'$ contains a rejection-proof packing of 3-cycles (which is thus also $c$-rejection-proof) that covers $k = 3 \cdot n_G$ vertices.

Proof sketch. Let $\varphi : V(H) \to V(G)$ be a subgraph isomorphism from $H$ to $G$. To encode this isomorphism as a packing of 3-cycles, we include the I-cycle $\langle u_1, u_2, x_3 \rangle$ for every $u \in V(H)$ and $x \in V(G)$ such that $x = \varphi(u)$. We extend this to a packing $\mathcal{C}$ that covers $k$ vertices by adding arbitrary but disjoint F-cycles that cover the $n_G - n_H$ vertices from $W_3$ that are not contained in any of the added I-cycles. It remains to show that this packing is rejection-proof.

As $\mathcal{C}$ leaves only the vertex $z$ uncovered, which belongs to $V_1'$, only agent 1 can reject and must do so by constructing an alternative packing that covers all vertices of $V_1'$. In particular, they must propose to add some internal cycle $C_z$ that includes $z$. By construction of $G'$, this cycle is either an R-cycle or an N-cycle and we can show with a case distinction that neither of those two options extends to a valid rejection for agent 1. ◁

Finally, we argue that Claims 4.2 and 4.3 combine to prove the original lemma statement. Together, they prove the correctness of the reduction from Subgraph Isomorphism to both RPKE-3 and $c$-RPKE-3. Moreover, the reduction can be executed in polynomial time. Hence, if an algorithm were to exist that solves either RPKE-3 or $c$-RPKE-3 in $2^{o(n \log n)}$ time, this algorithm could be used to determine in $2^{o(n \log n)} = 2^{o(n_G \log n_G)}$ time whether $H$ is isomorphic to a subgraph of $G$. By Lemma 2.5, this is not possible assuming the ETH. ◀

For completeness, we remark without proof that the construction above can be modified to prove that for any $d \geq 3$ and $c \geq 3$, an algorithm that solves $c$-RPKE-$d$ in time $2^{o(n \log n)}$ would violate the ETH.

## 5 A single-exponential algorithm for 1-rejections

In the previous section, we have shown that under the ETH it is impossible to solve RPKE-$d$ (or its restricted case 2-RPKE-$d$) in time $2^{o(n \log n)}$. This raises the question whether a single-exponential running time is possible for 1-RPKE-$d$. We answer this question positively by providing a $3^n \cdot (n+m)^{\mathcal{O}(1)}$ algorithm for the generalization 1-RP-$d$-SP.

▶ **Theorem 5.1 (★).** *For each fixed d, the* 1*-RP-d-SP problem can be solved in* $3^n \cdot (n+m)^{\mathcal{O}(1)}$ *time.*

**Proof sketch.** The main idea for solving an instance $I = (U_1, \ldots, U_p, \mathcal{S}, k)$ of 1-RP-$d$-SP is as follows. Let $U := \bigcup_{i \in [p]} U_i$ be the complete universe. The algorithm iterates over all subsets $U' \subseteq U$ of size at least $k$, each of which could be the set of elements covered by a solution to $I$. It then tries to find a solution covering exactly $U'$. The main insight now is the following: whether or not a set $A \in \mathcal{S}[U']$ leads to a 1-rejection by agent $i$ in a solution that covers exactly $U'$, is uniquely determined by the combination of $U'$ and $A$. It is independent of the identities of the other sets used in the solution. Agent $i$ performs a 1-rejection for $A$ if and only if there is a collection of pairwise disjoint $i$-internal sets avoiding $U' \setminus A$ that covers more elements from $U_i$ than $A$ does. For each choice of $U'$ we can therefore compute a collection $\mathcal{S}'$ of sets that are feasible to be used in a 1-rejection-proof solution covering exactly $U'$. Testing whether there is a 1-rejection-proof solution covering exactly $U'$ then turns into an instance of Exact Set Cover over a universe of size $|U'| \leq |U|$, which can be solved efficiently via Lemma 2.1. The exact running time follows from some small optimizations and the binomial theorem. ◀

## 6 Conclusion and discussion

In this paper we started the parameterized complexity analysis of rejection-proof kidney exchange, which was recently introduced by Blom et al. [6] due to practical applications. Our FPT algorithm shows that the complexity of the problem is governed by the size of the solution, rather than the size of the instance. For the single-agent version of the kidney exchange problem, its parameterized complexity has also been considered with respect to structural parameterizations such as treewidth. We leave such investigations of rejection-proof kidney exchange to further work. Looking beyond the kidney exchange (i.e., $d$-cycle packing) problem studied in this paper, one could consider rejection-proof versions of other packing problems such as Chordless Cycle Packing [21] and Odd Cycle Packing [18].

## References

1 David J. Abraham, Avrim Blum, and Tuomas Sandholm. Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM conference on Electronic commerce*, EC07. ACM, June 2007. doi:10.1145/1250910.1250954.

2 Nikhil Agarwal, Itai Ashlagi, Eduardo Azevedo, Clayton R. Featherstone, and Ömer Karaduman. Market failure in kidney exchange. *American Economic Review*, 109(11):4026–4070, November 2019. doi:10.1257/aer.20180771.

3 Itai Ashlagi, Felix Fischer, Ian A. Kash, and Ariel D. Procaccia. Mix and match: A strategyproof mechanism for multi-hospital kidney exchange. *Games and Economic Behavior*, 91:284–296, May 2015. doi:10.1016/j.geb.2013.05.008.

4 Peter Biro, Marton Gyetvai, Xenia Klimentova, Joao Pedro Pedroso, William Pettersson, and Ana Viana. Compensation scheme with shapley value for multi-country kidney exchange programmes. In *ECMS 2020 Proceedings edited by Mike Steglich, Christian Mueller, Gaby Neumann, Mathias Walther*, pages 129–136. ECMS, June 2020. doi:10.7148/2020-0129.

5 Péter Biró, Joris van de Klundert, David Manlove, William Pettersson, Tommy Andersson, Lisa Burnapp, Pavel Chromy, Pablo Delgado, Piotr Dworczak, Bernadette Haase, Aline Hemke, Rachel Johnson, Xenia Klimentova, Dirk Kuypers, Alessandro Nanni Costa, Bart Smeulders, Frits Spieksma, María O. Valentín, and Ana Viana. Modelling and optimisation in european kidney exchange programmes. *European Journal of Operational Research*, 291(2):447–456, June 2021. doi:10.1016/j.ejor.2019.09.006.

6 Danny Blom, Bart Smeulders, and Frits Spieksma. Rejection-proof mechanisms for multi-agent kidney exchange. *Games and Economic Behavior*, 143:25–50, January 2024. doi:10.1016/j.geb.2023.10.015.

7 Patrizia Burra and Manuela De Bona. Quality of life following organ transplantation. *Transplant International*, 20(5):397–409, May 2007. doi:10.1111/j.1432-2277.2006.00440.x.

8 Margarida Carvalho and Andrea Lodi. A theoretical and computational equilibria analysis of a multi-player kidney exchange program. *European Journal of Operational Research*, 305(1):373–385, February 2023. doi:10.1016/j.ejor.2022.05.027.

9 Margarida Carvalho, Andrea Lodi, João Pedro Pedroso, and Ana Viana. Nash equilibria in the two-player kidney exchange game. *Mathematical Programming*, 161(1–2):389–417, April 2016. doi:10.1007/s10107-016-1013-7.

10 Marek Cygan, Fedor V. Fomin, Alexander Golovnev, Alexander S. Kulikov, Ivan Mihajlin, Jakub Pachocki, and Arkadiusz Socala. Tight bounds for graph homomorphism and subgraph isomorphism. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1643–1649. SIAM, 2016. doi:10.1137/1.9781611974331.CH112.

11 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

12 P. Erdös and R. Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, s1-35(1):85–90, January 1960. doi:10.1112/jlms/s1-35.1.85.

13 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer Berlin, Heidelberg, 2006. doi:10.1007/3-540-29953-X.

14 Florent Foucaud, Esther Galby, Liana Khazaliya, Shaohua Li, Fionn Mc Inerney, Roohani Sharma, and Prafullkumar Tale. Metric dimension and geodetic set parameterized by vertex cover. In *42nd International Symposium on Theoretical Aspects of Computer Science, STACS 2025, March 4-7, 2025, Jena, Germany*, volume 327 of *LIPIcs*, pages 33:1–33:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICS.STACS.2025.33.

15 Chen Hajaj, John Dickerson, Avinatan Hassidim, Tuomas Sandholm, and David Sarne. Strategy-proof and efficient kidney exchange using a credit mechanism. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), February 2015. doi:10.1609/aaai.v29i1.9322.

**16** Úrsula Hébert-Johnson, Daniel Lokshtanov, Chinmay Sonar, and Vaishali Surianarayanan. Parameterized complexity of kidney exchange revisited. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 76–84. ijcai.org, 2024. `doi:10.24963/ijcai.2024/9`.

**17** Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. `doi:10.1006/JCSS.2000.1727`.

**18** Ken-ichi Kawarabayashi and Bruce A. Reed. Odd cycle packing. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 695–704. ACM, 2010. `doi:10.1145/1806689.1806785`.

**19** Xenia Klimentova, Ana Viana, João Pedro Pedroso, and Nicolau Santos. Fairness models for multi-agent kidney exchange programmes. *Omega*, 102:102333, July 2021. `doi:10.1016/j.omega.2020.102333`.

**20** Arnab Maiti and Palash Dey. Parameterized algorithms for kidney exchange. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 405–411. ijcai.org, 2022. `doi:10.24963/IJCAI.2022/58`.

**21** Dániel Marx. Chordless cycle packing is fixed-parameter tractable. In *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 71:1–71:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPICS.ESA.2020.71`.