informs
https://pubsonline.informs.org/journal/ited

INFORMS TRANSACTIONS ON EDUCATION

Vol. 00, No. 0, Xxxxx 0000, pp. 000–000

Submitted to *INFORMS Transactions on Education*

ISSN 1532-0545, EISSN 1532-0545

Odd-Length Exchanges in ABO-Only Kidney Exchange: A Feasibility Puzzle for the Classroom

Gabriel Teodoro
Department of Industrial, Manufacturing & Systems Engineering, Texas Tech University, TX, US, gteodoro@ttu.edu

Hamidreza Validi
Department of Industrial, Manufacturing & Systems Engineering, Texas Tech University, TX, US, hvalidi@ttu.edu

Abstract. Most classroom puzzles in Operations Research emphasize optimizing an objective; here we instead pose a *feasibility* puzzle rooted in kidney exchange programs (KEPs). We ask whether odd-length cycles—three-way exchanges as the canonical case—can arise when every donor–recipient pair is internally incompatible and compatibility is defined by ABO blood type alone. We formalize the puzzle as a mixed integer feasibility model that encodes assignment, incompatibility, and donor-to-next-recipient implication constraints. Simple variable fixings collapse the model to a reduced formulation whose structure reveals a bipartition of the cycle-capable nodes (A-to-B and B-to-A), thereby precluding any odd cycle and certifying infeasibility for the three-way case; the same logic extends to all odd lengths ≥ 5. We provide a concise Python+Gurobi implementation and an undergraduate classroom activity that uses the model to contrast feasibility reasoning with optimization thinking, connect blood-compatibility rules to mathematical constraints, and practice modular indexing on cyclic structures. The puzzle thus serves both as a correctness certificate for the ABO-only setting and as a compact teaching vehicle linking graph structure and integer programming.

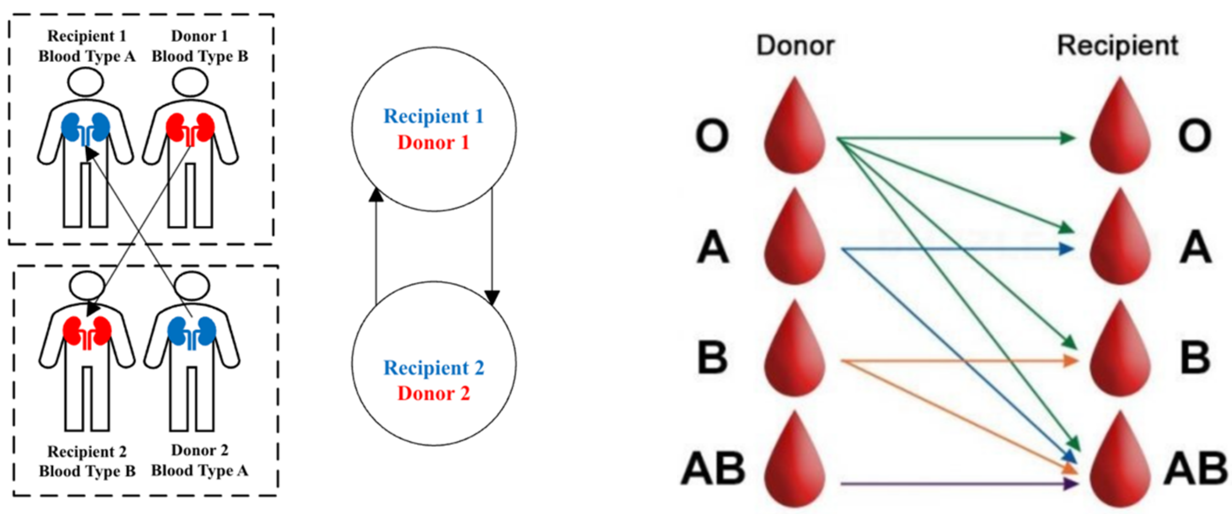
Key words: Kidney transplantation, Kidney exchange program, Puzzle, Mixed integer programming, Operations research, Management science

1. Introduction

Chronic kidney disease (CKD) imposes a substantial global mortality burden—about 1.5 million deaths in 2021—while end-stage kidney failure continues to face severe transplant shortages (Deng et al. 2025). In the United States, demand far exceeds supply: roughly 90,000 people are on the kidney waitlist at any time, yet only about 27,000–28,000 transplants were performed in 2023 (National Institute of Diabetes and Digestive and Kidney Diseases 2023). Median waiting times commonly range from 3 to 5 years, depending on blood type, geography, and health status (American

Kidney Fund 2023). These facts motivate educational activities that help students understand how modeling choices influence the feasibility of life-changing exchanges.

Kidney exchange programs (KEPs)[1] enable donor–recipient pairs to swap donors so that each patient receives a compatible organ. KEPs resolve incompatibilities by *re-matching* pairs through cycles and chains so that each patient receives a compatible organ while each willing donor helps another patient (Roth et al. 2004, 2005, Rees et al. 2009). In introductory descriptions, it is natural to depict both two-way and three-way exchanges, and many models allow cycles of length $k \geq 2$. Figure 1 shows a standard two-way exchange alongside a blood-type compatibility chart that underpins many textbook discussions.



(a) A 2-way exchange between incompatible pairs

(b) Blood compatibility for kidney transplantation

Figure 1 **A 2-way exchange between incompatible pairs with blood types A and B and a blood type compatibility diagram for living kidney transplants.**

The literature frequently illustrates three-way exchanges as a natural extension of two-way swaps, and some expository figures depict such cycles among incompatible pairs (e.g., Delorme et al. 2025, Fig. 1). Our contribution is not to debate those illustrations in general, but to clarify what happens *under an ABO-only compatibility assumption with internally incompatible pairs*, a simplification well suited for a first modeling exercise.

This paper uses the KEP setting to craft a short, self-contained *feasibility* puzzle for undergraduate operations research courses. Whereas many classroom problems emphasize optimizing an objective,

[1] Also referred to as kidney paired donation (KPD) programs.

our activity asks a crisp structural question: *Can odd-length cycles—three-way exchanges as the canonical case—occur when every pair is internally incompatible and compatibility is defined solely by ABO blood type?* The question is pedagogically appealing because it requires students to translate domain rules into binary assignment and implication constraints, reason carefully about what those constraints permit, and distinguish between "could be optimal" and "is even possible."

We make three practical contributions for instructors and students: (i) we formulate the puzzle as a mixed integer feasibility model that encodes assignment, incompatibility, and donor-to-next-recipient implication constraints; (ii) we show that simple variable fixings immediately reduce the model and make the feasibility answer transparent, including a direct certificate for (in)feasibility in the three-way case and a template for larger odd cycles; and (iii) we provide a concise Python+Gurobi implementation together with a classroom activity that highlights feasibility reasoning, modular (cyclic) indexing, and the connection between domain rules and binary constraints.

The rest of the paper is organized as follows. Section 2 presents the puzzle statement and visual aid used in class. Section 3 develops the base feasibility formulation and the resulting variable fixings, followed by a reduced model. Section 4 summarizes the Python+Gurobi implementation used to generate certificates. Section 5 briefly reports on our classroom experience and survey instrument. Section 6 concludes the paper.

2. The Puzzle

This section formalizes the classroom activity used in our course. Students are given three donor–recipient pairs that are *internally blood-incompatible*; in the diagram (Figure 2), each dashed box encloses a donor and their intended recipient, and blank labels indicate unknown blood types. A separate blood-compatibility chart for living kidney transplantation appears in Figure 1b. The goal is to determine whether a three-way exchange can be achieved by assigning ABO types to all blanks while respecting the following rules.

1. **Internal incompatibility:** within each dashed box, the donor's blood type must be incompatible with their paired recipient's blood type.
2. **One-to-one donation:** every donor must donate to exactly one recipient.
3. **One-to-one reception:** every recipient must receive exactly one kidney.

Compatibility on cross-pair arcs must be judged using the chart in Figure 1b. We restrict attention to ABO-only rules.

Core question. Is there an assignment of blood types $\{A, B, O, AB\}$ to all blanks in Figure 2 that enables the directed cycle

$$d_1 \to r_2, \quad d_2 \to r_3, \quad d_3 \to r_1,$$

with each arc blood-compatible and each donor remaining incompatible with their own paired recipient?

Student task. Students model the puzzle as a *feasibility* problem in mixed-integer programming:

- introduce binary assignment variables for each donor and recipient over $\{A, B, O, AB\}$,
- enforce the three rule families above (assignment, internal incompatibility, and cycle-compatibility implications using Figure 1b),
- run the model in Python using Gurobi and report either (i) a feasible ABO assignment for Figure 2 or (ii) a clear certificate of infeasibility.

Deliverables. Students submit (i) a brief statement of the formulation with variables and constraints labeled by role; (ii) code and a concise solver log showing "feasible" or "infeasible"; and (iii) a short plain-language explanation tying the outcome to specific constraints.

Optional extension. Students are asked to derive a reduced model with fewer variables/constraints by fixing variables implied by the rules (e.g., types that can never appear under internal incompatibility).

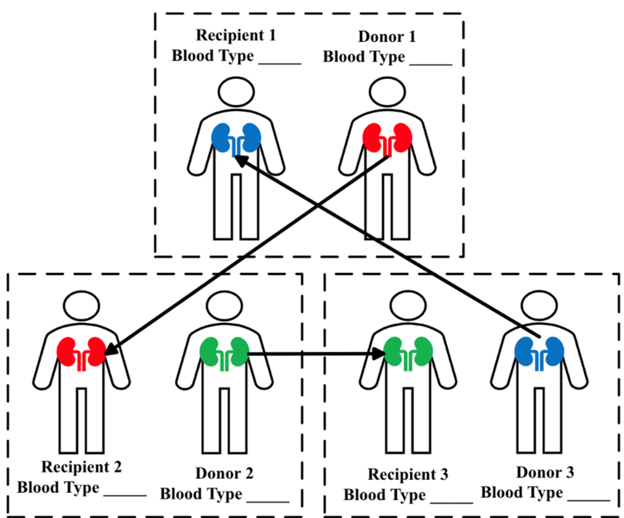


Figure 2 **The 3-way exchange puzzle. The color scheme encodes compatibility: similar colors denote blood-type compatibility and dissimilar colors denote incompatibility.**

The next section provides a clean feasibility formulation that we use in class, followed by simple variable fixings that streamline the model and make the outcome transparent.

3. Formulations and Fixings

This section translates the puzzle into a short mixed-integer *feasibility* model and then shows how a few immediate fixings simplify the formulation. Our goal is didactic: make each constraint family traceable to a rule in the puzzle and expose where "impossibility" comes from through algebra rather than solver magic. Throughout, we preserve the notation from Figure 2 and use modular indexing to express the three-cycle succinctly.

3.1. Base formulation

Let $P := \{(r_1, d_1), (r_2, d_2), (r_3, d_3)\}$ be the set of three recipient-donor pairs illustrated in Figure 2. For every pair index $i \in \{1, 2, 3\}$, we define binary decision variables $x_{r_i,b}$ and $y_{d_i,b}$ that shows if the recipient and donor in pair i are assigned to blood type $b \in \{\mathrm{A, B, O, AB}\}$, respectively. The model enforces three ingredients that mirror the puzzle rules:

1. **Assignment (one type per person).** Each recipient and each donor receives exactly one blood-type label; see constraints (1a)–(1b).
2. **Internal incompatibility (within each dashed box).** The donor's type is *not* compatible with the paired recipient's type. Constraints (1c)–(1f) and (1g)–(1j) encode this rule in both directions so that either choice (fixing a donor or fixing a recipient) propagates correctly.
3. **Cycle implications (across boxes).** If pair i donates to the next recipient $j = 1 + (i \bmod 3)$, then the donor type selected for d_i must be compatible with the recipient type selected for r_j, and conversely the chosen type for r_j must be supported by some compatible type for d_i; see (1k)–(1n) and (1o)–(1r).

Now, we have the following feasibility integer program for the puzzle. We first provide constraints that ensure any donor and recipient is assigned to exactly one blood type. So, for every pair $i \in \{1, 2, 3\}$, we have

$$x_{r_i,\mathrm{A}} + x_{r_i,\mathrm{B}} + x_{r_i,\mathrm{AB}} + x_{r_i,\mathrm{O}} = 1 \tag{1a}$$

$$y_{d_i,\mathrm{A}} + y_{d_i,\mathrm{B}} + y_{d_i,\mathrm{AB}} + y_{d_i,\mathrm{O}} = 1. \tag{1b}$$

Here, constraints (1a) imply that every recipient at any incompatible KEP pair is assigned to exactly one blood type. Similarly, constraints (1b) imply that every donor at any incompatible KEP pair is assigned to exactly one blood type.

The next set of constraints ensures that the blood types of donor and recipient are incompatible for each donor-recipient pair. So, for every pair $i \in \{1,2,3\}$, we have

$$y_{d_i,\mathrm{O}} \leq 1 - (x_{r_i,\mathrm{O}} + x_{r_i,\mathrm{A}} + x_{r_i,\mathrm{B}} + x_{r_i,\mathrm{AB}}) \tag{1c}$$

$$y_{d_i,\mathrm{A}} \leq 1 - (x_{r_i,\mathrm{A}} + x_{r_i,\mathrm{AB}}) \tag{1d}$$

$$y_{d_i,\mathrm{B}} \leq 1 - (x_{r_i,\mathrm{B}} + x_{r_i,\mathrm{AB}}) \tag{1e}$$

$$y_{d_i,\mathrm{AB}} \leq 1 - x_{r_i,\mathrm{AB}} \tag{1f}$$

For every incompatible KEP pair $i \in \{1,2,3\}$,

- constraints (1c) imply that if the donor has a blood type of O, then its corresponding recipient can have neither of blood types O, A, B, and AB;
- constraints (1d) imply that if the donor has a blood type of A, then its corresponding recipient can have neither of blood types A, and AB;
- constraints (1e) imply that if the donor has a blood type of B, then its corresponding recipient can have neither of blood types B, and AB; and
- constraints (1f) imply that if the donor has a blood type of AB, then its corresponding recipient can not have blood type of AB.

We also add another set of conflict constraints as follows:

$$x_{r_i,\mathrm{O}} \leq 1 - y_{d_i,\mathrm{O}} \tag{1g}$$

$$x_{r_i,\mathrm{A}} \leq 1 - (y_{d_i,\mathrm{O}} + y_{d_i,\mathrm{A}}) \tag{1h}$$

$$x_{r_i,\mathrm{B}} \leq 1 - (y_{d_i,\mathrm{O}} + y_{d_i,\mathrm{B}}) \tag{1i}$$

$$x_{r_i,\mathrm{AB}} \leq 1 - (y_{d_i,\mathrm{O}} + y_{d_i,\mathrm{A}} + y_{d_i,\mathrm{B}} + y_{d_i,\mathrm{AB}}). \tag{1j}$$

For every incompatible KEP pair $i \in \{1,2,3\}$,

- constraints (1g) imply that if the recipient has a blood type of O, then its corresponding donor can not have a blood type of O;
- constraints (1h) imply that if the recipient has a blood type of A, then its corresponding donor can have neither of blood types O and A;

- constraints (1i) imply that if the recipient has a blood type of B, then its corresponding donor can have neither of blood types O and B; and
- constraints (1j) imply that if the recipient has a blood type of AB, then its corresponding donor can have neither of blood types O, A, B, and AB.

We then impose the implication constraints that dictate that if a donor in pair i has a specific blood type, then the recipient in the next consecutive pair must have a compatible blood type. So, for every pair $i \in \{1,2,3\}$ and consecutive pair $j = 1 + (i \bmod 3)$, we have

$$y_{d_i,\mathrm{O}} \leq x_{r_j,\mathrm{A}} + x_{r_j,\mathrm{B}} + x_{r_j,\mathrm{O}} + x_{r_j,\mathrm{AB}} \tag{1k}$$

$$y_{d_i,\mathrm{A}} \leq x_{r_j,\mathrm{A}} + x_{r_j,\mathrm{AB}} \tag{1l}$$

$$y_{d_i,\mathrm{B}} \leq x_{r_j,\mathrm{B}} + x_{r_j,\mathrm{AB}} \tag{1m}$$

$$y_{d_i,\mathrm{AB}} \leq x_{r_j,\mathrm{AB}}. \tag{1n}$$

For every incompatible KEP pair $i \in \{1,2,3\}$ and its consecutive pair $j = 1 + (i \bmod 3)$,

- constraints (1k) imply that if the donor of pair i has the blood type of O, then the recipient in the consecutive pair j needs to have the blood type of either A, B, O or AB;
- constraints (1l) imply that if the donor of pair i has the blood type of A, then the recipient in the consecutive pair j needs to have the blood type of either A or AB;
- constraints (1m) imply that if the donor of pair i has the blood type of B, then the recipient in the consecutive pair j needs to have the blood type of either B or AB; and
- constraints (1n) imply that if the donor of pair i has the blood type of AB, then the recipient in the consecutive pair j needs to have the blood type of AB.

We also have the following set of implication constraints.

$$x_{r_j,\mathrm{O}} \leq y_{d_i,\mathrm{O}} \tag{1o}$$

$$x_{r_j,\mathrm{A}} \leq y_{d_i,\mathrm{O}} + y_{d_i,\mathrm{A}} \tag{1p}$$

$$x_{r_j,\mathrm{B}} \leq y_{d_i,\mathrm{O}} + y_{d_i,\mathrm{B}} \tag{1q}$$

$$x_{r_j,\mathrm{AB}} \leq y_{d_i,\mathrm{O}} + y_{d_i,\mathrm{A}} + y_{d_i,\mathrm{B}} + y_{d_i,\mathrm{AB}}. \tag{1r}$$

For every incompatible KEP pair $i \in \{1,2,3\}$ and its consecutive pair $j = 1 + (i \bmod 3)$,

- constraints (1o) imply that if the recipient of pair j has the blood type of O, then the donor in pair i needs to have the blood type O as well;
- constraints (1p) imply that if the recipient of pair j has the blood type of A, then the donor in pair i needs to have the blood type of either A or O;
- constraints (1q) imply that if the recipient of pair j has the blood type of B, then the donor in pair i needs to have the blood type of either B or O; and
- constraints (1r) imply that if the recipient of pair j has the blood type of AB, then the donor in pair i needs to have the blood type of either A, B, AB, or O.

We finally have the following constraints to clarify that all decision variables are binary. So, for every pair $i \in \{1, 2, 3\}$, we have

$$x_{r_i,\mathrm{A}}, x_{r_i,\mathrm{B}}, x_{r_i,\mathrm{AB}}, x_{r_i,\mathrm{O}} \in \{0, 1\}, \tag{1s}$$

$$y_{r_i,\mathrm{A}}, y_{r_i,\mathrm{B}}, y_{r_i,\mathrm{AB}}, y_{r_i,\mathrm{O}} \in \{0, 1\}. \tag{1t}$$

We note that formulation (1) can be easily generalized for any odd cycle with sizes greater than or equal to five.

3.2. Variable fixings

Before solving, several variables can be fixed by simple algebra using (1). These fixings shrink the model and highlight why certain assignments can never occur.

Fixing $y_{d_i,\mathrm{O}}$ to zero. For every pair $i \in \{1, 2, 3\}$, we have $y_{d_i,\mathrm{O}} = 0$ because

$$\begin{aligned} 0 \le y_{d_i,\mathrm{O}} &\le 1 - (x_{r_i,\mathrm{O}} + x_{r_i,\mathrm{A}} + x_{r_i,\mathrm{B}} + x_{r_i,\mathrm{AB}}) \\ &\le 1 - 1 = 0. \end{aligned}$$

Here, the first inequality holds by constraints (1t). The second inequality holds by constraints (1c). The last inequality holds by constraints (1a).

Fixing $x_{r_i,\mathrm{AB}}$ to zero. For every pair $i \in \{1, 2, 3\}$, we also have $x_{r_i,\mathrm{AB}} = 0$ because

$$\begin{aligned} 0 \le x_{r_i,\mathrm{AB}} &\le 1 - (y_{d_i,\mathrm{O}} + y_{d_i,\mathrm{A}} + y_{d_i,\mathrm{B}} + y_{d_i,\mathrm{AB}}) \\ &\le 1 - 1 = 0. \end{aligned}$$

Here, the first inequality holds by constraints (1s). The second inequality holds by constraints (1j). The last inequality holds by constraints (1b).

Fixing $y_{d_i,\text{AB}}$ to zero. Because $x_{r_j,\text{AB}}$ is zero for every pair $j \in \{1,2,3\}$, we have $y_{d_i,\text{AB}} = 0$ for any pair $i \neq j$ donating to pair j. In other words, we have

$$0 \leq y_{d_i,\text{AB}} \leq x_{r_j,\text{AB}} = 0.$$

Here, the first inequality holds by constraints (1t). The second inequality holds by constraints (1n). The equality holds because $x_{r_j,\text{AB}}$ is zero.

Fixing $x_{r_j,\text{O}}$ to zero. Because $y_{d_i,\text{O}}$ is zero for every pair $i \in \{1,2,3\}$, we have $x_{r_j,\text{O}} = 0$ for any pair $j \neq i$ receiving a kidney from pair i. In other words, we have

$$0 \leq x_{r_j,\text{O}} \leq y_{d_i,\text{O}} = 0.$$

These immediate fixings are useful in class because they show how the rules—without any solver—already narrow the feasible labels to the only types that can matter downstream.

3.3. Reduced formulation

Applying the fixings above collapses the model to a two-type world for each person (A or B). Keeping the same indices and labels, the reduced system below enforces: (i) one-of-two assignment ((2a)–(2b)); (ii) within-box “not-equal” constraints expressed as linear inequalities ((2c)–(2f)); (iii) cycle implications carried only by matching A→A and B→B ((2g)–(2j)); (iv) binary domains.

$$x_{r_i,\mathrm{A}} + x_{r_i,\mathrm{B}} = 1 \qquad \forall i \in \{1,2,3\} \tag{2a}$$
$$y_{d_i,\mathrm{A}} + y_{d_i,\mathrm{B}} = 1 \qquad \forall i \in \{1,2,3\} \tag{2b}$$
$$y_{d_i,\mathrm{A}} \le 1 - x_{r_i,\mathrm{A}} \qquad \forall i \in \{1,2,3\} \tag{2c}$$
$$y_{d_i,\mathrm{B}} \le 1 - x_{r_i,\mathrm{B}} \qquad \forall i \in \{1,2,3\} \tag{2d}$$
$$x_{r_i,\mathrm{A}} \le 1 - y_{d_i,\mathrm{A}} \qquad \forall i \in \{1,2,3\} \tag{2e}$$
$$x_{r_i,\mathrm{B}} \le 1 - y_{d_i,\mathrm{B}} \qquad \forall i \in \{1,2,3\} \tag{2f}$$
$$y_{d_i,\mathrm{A}} \le x_{r_j,\mathrm{A}} \qquad j = 1 + (i \bmod 3),\ \forall i \in \{1,2,3\} \tag{2g}$$
$$y_{d_i,\mathrm{B}} \le x_{r_j,\mathrm{B}} \qquad j = 1 + (i \bmod 3),\ \forall i \in \{1,2,3\} \tag{2h}$$
$$x_{r_j,\mathrm{A}} \le y_{d_i,\mathrm{A}} \qquad j = 1 + (i \bmod 3),\ \forall i \in \{1,2,3\} \tag{2i}$$
$$x_{r_j,\mathrm{B}} \le y_{d_i,\mathrm{B}} \qquad j = 1 + (i \bmod 3),\ \forall i \in \{1,2,3\} \tag{2j}$$
$$x_{r_i,\mathrm{A}}, x_{r_i,\mathrm{B}} \in \{0,1\} \qquad \forall i \in \{1,2,3\} \tag{2k}$$
$$y_{r_i,\mathrm{A}}, y_{r_i,\mathrm{B}} \in \{0,1\} \qquad \forall i \in \{1,2,3\}. \tag{2l}$$

Note that (2e)–(2f) replicate (2c)–(2d) and may be viewed as redundant. Combining the implication pairs yields a more compact version in which (3e)–(3f) equate donor choices in pair i with recipient labels in the next pair $j = 1 + (i \bmod 3)$:

$$x_{r_i,\mathrm{A}} + x_{r_i,\mathrm{B}} = 1 \qquad \forall i \in \{1,2,3\} \tag{3a}$$
$$y_{d_i,\mathrm{A}} + y_{d_i,\mathrm{B}} = 1 \qquad \forall i \in \{1,2,3\} \tag{3b}$$
$$y_{d_i,\mathrm{A}} \le 1 - x_{r_i,\mathrm{A}} \qquad \forall i \in \{1,2,3\} \tag{3c}$$
$$y_{d_i,\mathrm{B}} \le 1 - x_{r_i,\mathrm{B}} \qquad \forall i \in \{1,2,3\} \tag{3d}$$
$$y_{d_i,\mathrm{A}} = x_{r_j,\mathrm{A}} \qquad j = 1 + (i \bmod 3),\ \forall i \in \{1,2,3\} \tag{3e}$$
$$y_{d_i,\mathrm{B}} = x_{r_j,\mathrm{B}} \qquad j = 1 + (i \bmod 3),\ \forall i \in \{1,2,3\} \tag{3f}$$
$$x_{r_i,\mathrm{A}}, x_{r_i,\mathrm{B}} \in \{0,1\} \qquad \forall i \in \{1,2,3\} \tag{3g}$$
$$y_{r_i,\mathrm{A}}, y_{r_i,\mathrm{B}} \in \{0,1\} \qquad \forall i \in \{1,2,3\}. \tag{3h}$$

4. Solution

We implement the feasibility model in Python using the Gurobi Optimizer (`gurobipy`) and solve either for a valid ABO assignment that realizes the 3-cycle or for a certificate of infeasibility (Gurobi Optimization, LLC 2025). The code mirrors the formulation in Section 3: it constructs binary assignment variables for each donor and recipient over $\{\mathrm{A}, \mathrm{B}, \mathrm{O}, \mathrm{AB}\}$, adds the assignment constraints, encodes internal (within-pair) incompatibility, and enforces donor-to-next-recipient implications using modular indexing $j = 1 + (i \bmod k)$ to express the directed cycle succinctly. For transparency in class, each block of constraints is created by a short function whose name matches the role used in the paper.

For classroom use, the model solves essentially instantly on a standard laptop. We encourage students to echo the counts of variables and constraints before and after the elementary fixings from Section 3 to visualize how presolve-style reasoning shrinks the model, and to save a concise solver log showing the final status (FEASIBLE/INFEASIBLE) and, when relevant, IIS statistics. A minimal, self-contained package (Python script, short `README`, and example notebook) is available at `https://github.com/gabrieltepin/odd_cycle_puzzle` so that instructors can drop the activity into a single lab or homework without additional infrastructure.

5. Classroom Experience

We implemented the activity in IE 3311: Deterministic Operations Research (Fall 2025) in the Industrial, Manufacturing, and Systems Engineering Department at Texas Tech University. Forty undergraduate students completed the puzzle. The in-class workflow was simple and deliberate: students first read the KEP context and the three rules, then translated those rules into a compact mixed integer feasibility model, and finally tested their formulations in Python with Gurobi. Working individually, they were asked to label each constraint by role (assignment, incompatibility, implication) and to explain—in two or three sentences—why their model either produced a valid assignment or certified infeasibility. This emphasis on labeled constraints and short plain-language explanations helped keep the focus on feasibility reasoning rather than solver "tuning."

To capture perceptions and learning outcomes, we administered a short, anonymous survey immediately after the submission deadline. The instrument consisted of Likert-scale statements on a 5-point scale (Strongly Disagree–Strongly Agree) (Likert 1932) targeting (i) conceptual understanding of feasibility versus optimization, (ii) the translation of medical rules into binary constraints, (iii) confidence with modular indexing on cycles, and (iv) engagement and course relevance. The exact prompts used in class appear in Table **??**. Students were also invited to add brief comments on what aspects of the modeling or coding were most challenging and which scaffolds (e.g., variable naming templates) were most helpful.

Instructors reported that the activity fits comfortably within a single class meeting, including a short debrief. A practical pacing guide is 10–12 minutes for context and rules, 30–35 minutes for modeling and coding, and 5–8 minutes for a wrap-up that contrasts feasibility certificates with objective-driven models and highlights how simple variable fixings (Section 3) clarify the outcome. Keeping the code modular—one short function per constraint family—made it straightforward for students to trace each line of LaTeX to a handful of implementation lines and to diagnose infeasibility using Gurobi's output. The survey responses (and informal feedback) suggest that the puzzle is an effective way to connect domain rules, modeling, and computational verification in a compact, undergraduate-friendly format.

Table 1 Post-activity Likert-scale survey (1 = Strongly Disagree, 5 = Strongly Agree).

Statement	1	2	3	4	5
The activity improved my skill in translating real-world puzzles into optimization models.					
I can explain how blood compatibility rules translate into mathematical constraints.					
Implementing the model in Python+Gurobi clarified how a mathematical formulation can be programmed.					
I could generalize the 3-cycle formulation to larger odd cycles with minimal guidance.					
I felt more engaged with the course material by participating in this activity.					
The activity increased my interest in OR applications in healthcare (e.g., kidney exchange).					
The instructions, time allocation, and materials were appropriate for the course level.					
Similar puzzle-based activities should be used in future IE 3311 offerings.					
The activity improved my problem-solving and analytical thinking skills.					

Response scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree.

6. Conclusion

This paper posed a kidney-exchange question as a feasibility puzzle: can odd-length cycles occur when each pair is internally incompatible and compatibility is defined solely by ABO blood type? By expressing the rules as a mixed integer model—and then applying simple variable fixings—we obtained a reduced formulation that makes the answer transparent: under these assumptions, a three-way exchange cannot be realized, and the same reasoning rules out all longer odd-length cycles as well. The model therefore serves as a short, verifiable certificate of infeasibility for the odd-cycle case in the ABO-only setting. Pedagogically, the activity centers on feasibility thinking rather than objective-driven optimization. Students translate clinical rules into binary assignment and implication constraints, practice modular (cyclic) indexing, and see how presolve-style fixings simplify a formulation and expose what is (and is not) structurally possible. The accompanying Python+Gurobi script makes these ideas concrete and reproducible in a short lab or homework.

References

American Kidney Fund. Kidney transplant waiting list. `https://www.kidneyfund.org/kidney-donation-and-transplant/transplant-waiting-list`, 2023. Accessed: September 1, 2025.

Maxence Delorme, Wendy Liu, and David Manlove. Mathematical models for the kidney exchange problem with reserve arcs, 2025. Preprint.

Ling Deng, Shujin Guo, Yuping Liu, Yaojia Zhou, Youren Liu, Xiaoxia Zheng, Xijie Yu, and Ping Shuai. Global, regional, and national burden of chronic kidney disease and its underlying etiologies from 1990 to 2021: a systematic analysis for the Global Burden of Disease Study 2021. *BMC Public Health*, 25:636, 2025. doi: 10.1186/s12889-025-21851-z. URL `https://bmcpublichealth.biomedcentral.com/articles/10.1186/s12889-025-21851-z`.

Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*, 2025. URL `https://www.gurobi.com`.

Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):1–55, 1932.

National Institute of Diabetes and Digestive and Kidney Diseases. Kidney disease statistics for the united states. `https://www.niddk.nih.gov/health-information/health-statistics/kidney-disease`, 2023. Accessed: September 1, 2025.

Michael A. Rees, John E. Köpke, Ron Pelletier, et al. A nonsimultaneous, extended, altruistic-donor chain. *The New England Journal of Medicine*, 360(11):1096–1101, 2009. doi: 10.1056/NEJMoa0803645.

Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Kidney exchange. *The Quarterly Journal of Economics*, 119(2): 457–488, 2004. doi: 10.1162/0033553041382157.

Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Pairwise kidney exchange. *Journal of Economic Theory*, 125(2): 151–188, 2005. doi: 10.1016/j.jet.2005.04.004.