

# Reformulation-Linearization Technique Approach for Kidney Exchange Program IT Healthcare Platforms

Junsang Yuh <sup>1,2</sup>, Seokhyun Chung <sup>2</sup> and Taesu Cheong <sup>2,\*</sup>

<sup>1</sup> R&D Center, Begas, Seoul 06179, Korea; yuhjunsang@gmail.com

<sup>2</sup> School of Industrial Management Engineering, Korea University, Seoul 02841, Korea; csh98016@gmail.com

\* Correspondence: tcheong@korea.ac.kr; Tel.: +82-2-3290-3382

Received: 8 July 2017; Accepted: 11 August 2017; Published: 17 August 2017

**Abstract:** Kidney exchange allows a potential living donor whose kidney is incompatible with his intended recipient to donate a kidney to another patient so that the donor's intended recipient can receive a compatible kidney from another donor. These exchanges can include cycles of longer than two donor-patient pairs and chains produced by altruistic donors. Kidney exchange programs (KEPs) can be modeled as a maximum-weight cycle-packing problem in a directed graph. This paper develops a new integer programming model for KEPs by applying the reformulation-linearization technique (RLT) to enhance a lower bound obtained by its linear programming (LP) relaxation. Given the results obtained from the proposed model, the model is expected to be utilized in the integrated KEP IT (Information Technology) healthcare platform to obtain plans for optimized kidney exchanges.

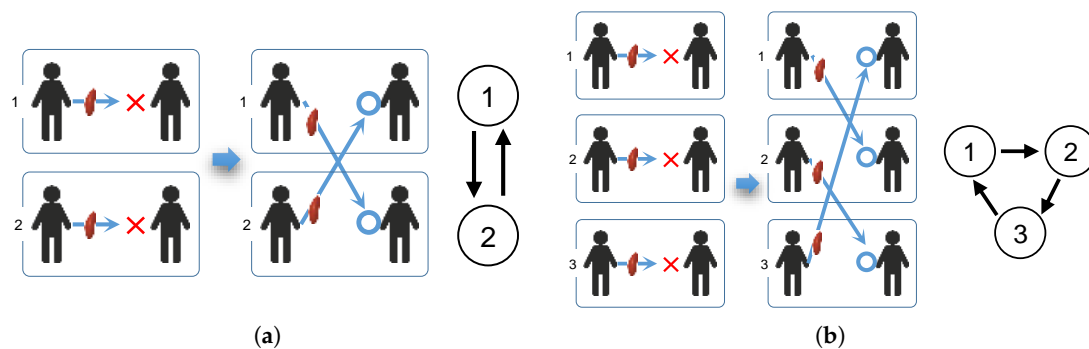
**Keywords:** kidney exchange program; integer programming; reformulation-linearization technique; IT healthcare platform

## 1. Introduction

Kidney transplants are essential for many end-stage renal disease sufferers. However, finding compatible kidneys may be difficult because of blood or tissue incompatibilities between donors and their intended recipients. The issue of incompatibility can be overcome by establishing a kidney exchange program (KEP). The first of these was performed in South Korea [1–3] in 1991 and they have since been widely performed in many countries, including the United States of America [4–9], the United Kingdom [10–12] and Australia [13]. A KEP helps a potential living donor to donate his or her kidney, which is incompatible with the intended recipient, to another recipient so that the donor's intended recipient can also receive a compatible kidney from another donor. The Korea Centers for Disease Control & Prevention (KCDC), the government ministry that maintains a unified administration to avoid disease and to improve public health welfare in South Korea, has recently paid attention to the contribution of the KEP, which increases the number of renal disease sufferers able to receive kidney transplants. The KCDC has pushed ahead with a plan for the development of an integrated IT healthcare platform for the KEP. On the platform, hospitals can register the essential information for verifying the compatibility of their renal disease sufferers. Then, the proposed KEP platform would provide matching plans to maximize the number of possible kidney exchanges in the program. Hence, it is important to establish an efficient optimization model for KEPs that can provide optimized kidney exchange plans within a reasonable time, and, in this paper, we intend to provide a high-performance mathematical programming model easily solvable by commercial optimization solvers (e.g., Cplex, Gurobi) to support the integrated KEP IT healthcare platform of South Korea. We remark that, as in our case, operations research (OR) techniques have been applied and utilized to tackle many important and challenging optimization issues in healthcare (e.g., [14] for an overview of

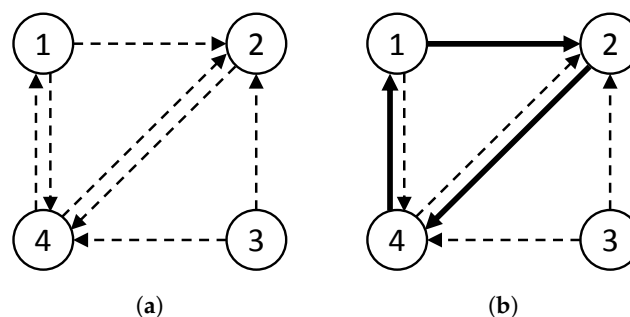
recent research on several OR applications in healthcare, and [15,16] for medical resident scheduling problems and sensor-based patient monitoring problems, respectively).

Figure 1 illustrates two-way and three-way KEPs. Figure 1a illustrates a case consisting of only two pairs. Within each pair, the patient and donor are incompatible. However, the patient in pair 1 is compatible with the donor in pair 2 and the patient in pair 2 is compatible with the donor in pair 1. We note that in precedent times, when exchanges between pairs were not allowed, there was no way to perform the transplants. However, with the KEP, exchanges between the pairs become possible, and the two transplants can be performed. As we will explain in detail below, the KEP can be expressed as a digraph for which a node corresponds to an incompatible donor–recipient pair and a directed arc is introduced if a donor in an initial node is compatible with a recipient in the terminal node of the arc (see the graph in Figure 1a). In practice, three-way exchanges are the most complicated exchanges performed because one of all the operations on an exchange cycle must be performed simultaneously to avoid the risk of one of the donors reneging, and it is technically difficult to simultaneously reserve operation rooms for more than six patients in a hospital. Figure 1b illustrates three-way kidney exchanges.



**Figure 1.** Illustration of (a) two-way and (b) three-way kidney exchange program (KEP).

As we briefly mentioned before, the KEP model can be expressed by a digraph  $G(V, A)$ , called a *KEP graph* in this paper, where  $V$  and  $A$  indicate the set of vertices and arcs of the graph  $G$ , respectively. Figure 2 presents an example of  $G$  and a solution to the model (i.e., possible kidney exchanges on a KEP graph). In Figure 2a, a vertex  $i \in V$  corresponds to a current incompatible pair consisting of a patient  $i$  and a donor  $i$ , and an arc  $(i, j) \in A$  indicates the compatibility between donor  $i$  and patient  $j$ , which means that donor  $i$  can provide his or her kidney to patient  $j$ . Figure 2b demonstrates an example of a feasible solution. The thick arrows represent an exchange cycle among pairs 1, 2, and 4.



**Figure 2.** Kidney exchange program (KEP) graph (a) and its feasible solution (b).

We now discuss the optimization problem for a KEP, given a KEP graph  $G$ . The objective of the optimization of a KEP is generally to maximize the number of possible kidney exchanges in a given pool of candidates, and the KEP can be mathematically modeled as an integer programming problem,

which is the main focus of this paper. Roth et al. [17] proposed an edge formulation and a cycle formulation for KEPs, and Abraham et al. [7] developed a solution algorithm for those formulations. Furthermore, Constantino et al. [18] proposed two compact formulations—that is, an edge-assignment (EA) formulation and an extended edge (EE) formulation. Manlove and O'Malley [19] presented how the cycle formulation by Roth et al. [17] can be extended to find optimal solutions for the KEP in the United Kingdom, which are compatible with U.K. National Living Donor Kidney Sharing Schemes (NLDKSS).

The reformulation-linearization technique (RLT), which we intend to utilize in this paper, is a procedure that can be used to generate a tight linear programming (LP) relaxation for a discrete optimization problem. Various strong formulations on several classes of structured problems are generated by the RLT. Sherali et al. [20] developed a new integer programming model for the traveling salesman problem (TSP) by applying the RLT to the Miller–Tucker–Zemlin (MTZ) formulation. Moreover, they showed that the resulting formulation dominates the lifted-MTZ formulation of Desrochers and Laporte [21]. Sherali et al. [22] applied the RLT to the path-based TSP model using logical restrictions. They proposed a variety of new formulations for the TSP, and exploited several dominance relationships between them. Park et al. [23] dealt with a two-level facility location–allocation problem arising from the access network design of a telecommunication network. They developed a mixed-integer program for the problem and applied the RLT to improve computational effectiveness.

In this paper, we focus on the best-known compact formulations for the KEP in the literature, which are the EA and EE formulations proposed by Constantino et al. [18]. We first induce the EE formulation by applying the partial level 1 RLT to the EA formulation. After that, we further strengthen the formulation via the level 1 RLT and propose a new integer programming model that dominates existing formulations. Computational results are provided to demonstrate the effectiveness of our formulation compared to that of those previous. Therefore, the contribution of our study can be summarized as below.

- We propose a new integer programming model, which is proved to be more compact and effective than previous optimization models through computational experiments. Although our new formulation gives a solution that allows only one more patient to have a kidney transplant than the matching solutions made by existing formulations, this is a meaningful result because a kidney transplant is immediately related to the precious life of a patient. Furthermore, our proposed model would be directly related to the performance of the KEP IT healthcare platform.
- We derive the EE formulation from the EA formulation by applying a systematic approach, the RLT. As presented by Constantino et al. [18], establishing a more compact integer programming formulation requires the exploitation of the special structures inherent to the problem of interest. On the other hand, by using our approach, such a formulation can be derived by a simple and systematic application of the RLT independent of the problem-specific structures. This difference suggests the possibility that our approach can be applied to other real-world optimization problems to obtain more significant formulations.

This paper is organized as follows: Section 2 briefly provides the overview of the RLT for the reference. Section 3 reviews the two representative integer programming models (i.e., the EE and EA formulations) for the KEP in literature. Section 4 presents how the EE formulation can be derived from the EA formulation by applying the RLT and how we could further strengthen the EE formulation. Section 5 presents the computational results. Finally, we conclude the discussion in Section 6.

## 2. Reformulation-Linearization Technique

In this section, we briefly describe the concept of the RLT process in the context of zero-one integer problems. Let us define the feasible region of a zero-one integer problem as

$$X = \{x : \mathbf{A}x \leq b, x \in \mathbb{B}^r\}$$

where  $\mathbf{A}$  is a  $q \times r$  matrix,  $b$  is a  $q$  vector, and  $N = \{1, 2, \dots, r\}$  is the index set of all variables. Without loss of generality, we assume that every component of  $\mathbf{A}$  and  $b$  is an integer. Additionally, the inequalities accommodate the upper bounds of  $x_j \leq 1$  for  $j \in N$ . As usual,  $\mathbb{B}$  and  $\mathbb{R}$  denote the sets of binary and real numbers, respectively.

The RLT process by Sherali and Adams [24] to construct relaxation  $X_d$  at any level  $d \in \{1, \dots, r\}$  in higher dimensional space consists of two steps, as follows:

- Step 1.** (Reformulation) Multiply each constraint defining the feasible region with every product factor of the form  $\prod_{i \in J_1} x_i \prod_{i \in J_2} (1 - x_i)$ , where  $J_1, J_2 \subseteq N$ ,  $J_1 \cap J_2 = \emptyset$ , and  $|J_1 \cup J_2| = d$ , and then apply the identity  $x_i^2 = x_i$  for all  $i \in N$ .
- Step 2.** (Linearization) Linearize the resulting polynomial system by substituting a variable  $\lambda_j$  for every distinct product term  $\prod_{i \in J} x_i$ , where  $J \subseteq N$ .

The results of Sherali and Adams [24] show that, by denoting the projection of  $X_d$  onto the space of the original variables  $x$  by  $X_{P_d} = \{x : (x, \lambda) \in X_d\}$ , we obtain the hierarchy of relaxations

$$X_{P_0} \equiv X_0 \supseteq X_{P_1} \supseteq X_{P_2} \supseteq \dots \supseteq X_{P_r} \equiv \text{conv}(X)$$

where  $X_{P_0} \equiv X_0$  denotes the linear programming relaxation of  $X$ . The results above imply that we can theoretically obtain the convex hull of the feasible solution set  $X$  via the level  $r$  RLT and can hence obtain the optimal solution by solving the linear programming problem over  $X_{P_r}$ .

For a better understanding, we illustrate the RLT process through the following example [25].

**Example 1.** Consider the following polytope:

$$X = \{x \in \mathbb{R}^2 : 2x_1 + 3x_2 \leq 4, x_j \in \{0, 1\}, j = 1, 2\}.$$

The linear programming relaxation of this polytope is given as  $X_0 = \{x \in \mathbb{R}^2 : 2x_1 + 3x_2 \leq 4, 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1\}$ , and the convex hull of  $X$  is given as  $\text{conv}(X) = \{x \in \mathbb{R}^2 : x_1 + x_2 \leq 1, x_1 \geq 0, x_2 \geq 0\}$  (see Figure 3).

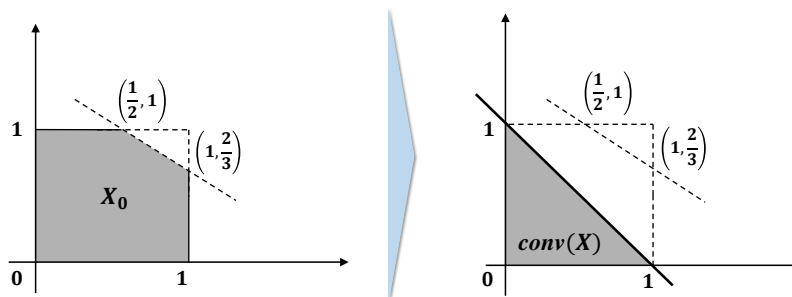


Figure 3. Illustration of polytope  $X_0$  (left) and its convex hull  $\text{conv}(X)$  (right).

Now, we apply the level 2 RLT to  $X_0$  to illustrate how  $\text{conv}(X)$  can be obtained by the RLT process.

**Step 1. (Reformulation)** Multiply each constraint of  $X_0$  by each of the factors  $x_1x_2$ ,  $x_1(1-x_2)$ ,  $(1-x_1)x_2$ , and  $(1-x_1)(1-x_2)$ . This process generates various duplicated or redundant inequalities; thus, we summarize the valid inequalities only, as follows:

$$\begin{array}{llll}
 (2x_1 + 3x_2 \leq 4) \times x_1x_2 & \rightarrow & 2x_1x_2 + 3x_1x_2 \leq 4x_1x_2 \\
 & \hookrightarrow & x_1x_2 \leq 0 \\
 (2x_1 + 3x_2 \leq 4) \times (1-x_1)(1-x_2) & \rightarrow & 0 \leq 4(1-x_1)(1-x_2) \\
 & \hookrightarrow & x_1 + x_2 \leq 1 + x_1x_2 \\
 (x_1 \geq 0) \times x_1x_2 & \rightarrow & x_1x_2 \geq 0 \\
 (x_1 \geq 0) \times x_1(1-x_2) & \rightarrow & x_1 - x_1x_2 \geq 0 \\
 (x_2 \geq 0) \times (1-x_1)x_2 & \rightarrow & x_2 - x_1x_2 \geq 0
 \end{array}$$

**Step 2. (Linearization)** After linearizing the resulting polynomial system by substituting a variable  $z_{12}$  for the product term  $x_1x_2$ , we have

$$\begin{aligned}
 z_{12} &\leq 0 \\
 x_1 + x_2 &\leq 1 + z_{12} \\
 z_{12} &\geq 0 \\
 x_1 - z_{12} &\geq 0 \\
 x_2 - z_{12} &\geq 0
 \end{aligned}$$

By  $z_{12} \leq 0$  and  $z_{12} \geq 0$ , we have  $z_{12} = 0$ . Therefore, the last set of inequalities produces

$$\{x \in \mathbb{R}^2 : x_1 + x_2 \leq 1, x_1 \geq 0, x_2 \geq 0\}$$

which is  $\text{conv}(X)$ .

### 3. Two Representative Models for the KEP: EA and EE Formulations

In this section, we present two compact formulations, the EA and EE formulations, which were proposed by Constantino et al. [18] as reference models. In these formulations, the possible cycles in  $G$  and  $L$  are considered to be defined as a set composed of indices  $\{1, \dots, l_u\}$ , where  $l_u$  indicates the upper bound on the possible number of cycles in  $G$ . The additional parameters and decision variables used in the formulation are given as follows:

#### Parameters

$w_{ij}$ : Weight of edge  $(i, j) \in A$ , representing the level of compatibility between donor  $i$  and recipient  $j$ .  
 $k$ : Maximum cycle length.

#### Decision Variables

$x_{ij}$ : 1 if a kidney is transplanted from donor  $i$  to patient  $j$ , and 0 otherwise.  
 $y_i^l$ : 1 if node  $i$  is included in cycle  $l$ , and 0 otherwise.

Then, the EA formulation is given as follows:

$$\text{EA : Maximize } \sum_{(i,j) \in A} w_{ij} x_{ij} \quad (1a)$$

$$\text{subject to } \sum_{j: (j,i) \in A} x_{ji} = \sum_{j: (i,j) \in A} x_{ij}, \quad \forall i \in V \quad (1b)$$

$$\sum_{j: (i,j) \in A} x_{ij} \leq 1, \quad \forall i \in V \quad (1c)$$

$$\sum_{i \in V} y_i^l \leq k, \quad \forall l \in L \quad (1d)$$

$$\sum_{l \in L} y_i^l = \sum_{j: (i,j) \in A} x_{ij}, \quad \forall i \in V \quad (1e)$$

$$y_i^l + x_{ij} \leq 1 + y_j^l, \quad \forall (i,j) \in A, \forall l \in L \quad (1f)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A \quad (1g)$$

$$y_i^l \in \{0, 1\}, \quad \forall i \in V, l \in L \quad (1h)$$

The objective function of Equation (1a) is to maximize the sum of weights of the transplantations. Because the weight of each edge indicates the appropriateness of the corresponding transplantation, the KEP maximizing the total appropriateness of all transplantations is planned. We note that setting  $w_{ij}$  to the same value for all  $(i,j) \in A$  allows the maximum number of possible kidney exchanges in a given pool of candidates to be obtained. Constraint (1b) indicates that if a donor in pair  $i$  donates his or her kidney to a patient in another pair, then the patient in pair  $i$  must receive a kidney. This constraint further implies that if a donor does not participate in transplantation, the paired patient would not receive a kidney. Constraint (1c) implies that a donor cannot donate more than one kidney. Constraint (1d) ensures that the length of each cycle cannot be greater than  $k$ . Constraint (1e) assigns a pair to a cycle if it takes part in kidney exchanges. We note that because  $\sum_{j: (i,j) \in A} x_{ij} \leq 1$ , a pair has to be assigned to one of the cycles. Constraint (1f) implies that if a donor in pair  $i$  gives his or her kidney to a patient in pair  $j$  and if the pair  $i$  is assigned to a cycle  $l$ , then pair  $j$  also has to be included in cycle  $l$ . Constraints (1g) and (1h) indicate that all the decision variables in the model are binary variables.

We now present the EE formulation. We suppose that there are copies of graph  $G$  and we let the index of each copy be  $l \in L$ , where  $L$  is a set of the copies of  $G$ . Then, the EE formulation requires the additional decision variables  $x_{ij}^l$ , as follows:

### Decision Variable

$x_{ij}^l$ : 1 if a kidney is transplanted from donor  $i$  to patient  $j$  in copy  $l$ , and 0 otherwise.

Then, the EE formulation is given as follows:

$$\text{EE : Maximize } \sum_{l \in L} \sum_{(i,j) \in A} w_{ij} x_{ij}^l \quad (2a)$$

$$\text{Subject to } \sum_{j: (j,i) \in A} x_{ji}^l = \sum_{j: (i,j) \in A} x_{ij}^l, \quad \forall i \in V, \forall l \in L \quad (2b)$$

$$\sum_{l \in L} \sum_{j: (i,j) \in A} x_{ij}^l \leq 1, \quad \forall i \in V \quad (2c)$$

$$\sum_{(i,j) \in A} x_{ij}^l \leq k, \quad \forall l \in L \quad (2d)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A \quad (2e)$$

The objective function of Equation (2a) indicates that the model maximizes the total weight of arcs among all copies of  $G$ . Constraint (2b) ensures that the number of kidneys given by donor  $i$  and received by patient  $i$  should be same at each pair  $i$  for each copy  $l$ . Constraint (2c) indicates that a pair

can donate or receive at most one kidney. We note that the constraint holds for all copies  $l \in L$  and that an arc cannot be selected for more than two copies of  $G$ . Constraint (2d) restricts the length of exchange cycles to less than or equal to  $k$ .

In addition, in order to have further simplified formulations, symmetry elimination constraints were applied, as discussed by Constantino et al. [18]:

$$y_i^l \leq y_l^l, \quad \forall i \in V, l \in L, i > l \quad (3)$$

for the EA formulation, and

$$\sum_{j: (i, j) \in A} x_{ij}^l \leq \sum_{j: (l, j) \in A} x_{lj}^l, \quad \forall i > l$$

for the EE formulation. These inequalities enforce node  $l$  to be in cycle  $l$  and enforce all other nodes in the cycle to have indices larger than  $l$ . For further details, we refer the readers to Constantino et al. [18].

#### 4. RLT Applications

In this section, we will present the application of the RLT to two existing compact formulations (i.e., the EA and EE formulations). We first show that the EE formulation can be derived by applying the RLT to the EA formulation systemically without relying on any problem-specific structures or insights. After that, we derive a new compact formulation that dominates the EE formulation by further applying the RLT to it.

##### 4.1. Derivation of the EE Formulation from the EA Formulation

We now apply a specialized version of the RLT to the EA formulation by following the discussion by Sherali and Adams [24,26]. To contain the size of the resulting relaxation, we apply only a partial first-level version of this approach. Our approach is to first reformulate the EA formulation by generating the additional (nonlinear) implied constraints. By applying a linearization, which substitutes variables in place of each distinct nonlinear term, we can expose the useful relationships between the new and original variables that intend to enforce the nonlinear relationships. Specifically, we perform the following operations:

- **Reformulation Phase.** Construct additional sets of constraints via (R1)–(R5) stated below.

(R1) Similarly to Constraint (1f), construct the following valid inequalities:

$$y_j^l + x_{ij} \leq 1 + y_i^l, \quad \forall (i, j) \in A, \forall l \in L \quad (4)$$

Using Constraints (1f) and (4), construct the valid inequalities

$$x_{ij}(y_i^l + x_{ij} - 1 - y_j^l) \leq 0 \quad \text{for } (i, j) \in A, l \in L$$

and

$$x_{ij}(y_j^l + x_{ij} - 1 - y_i^l) \leq 0 \quad \text{for } (i, j) \in A, l \in L.$$

- (R2) For each  $i \in V$ , multiply the equation  $\sum_{j: (j, i) \in A} x_{ji} = \sum_{j: (i, j) \in A} x_{ij}$  of Constraint (1b) by  $y_i^l \geq 0$  for each  $i \in V, l \in L$ .
- (R3) For each  $i \in V$ , multiply the inequality  $\sum_{j: (i, j) \in A} x_{ij} \leq 1$  of Constraint (1c) by  $y_i^l \geq 0$  for each  $i \in V, l \in L$ .
- (R4) From Constraints (1c) and (1e), we can derive  $\sum_{l \in L} y_i^l \leq 1$  for  $i \in V$ . For each  $i \in V$ , multiply  $\sum_{l \in L} y_i^l \leq 1$  and  $\sum_{l \in L} y_i^l = \sum_{j: (i, j) \in A} x_{ij}$  by  $y_i^{l'}$  for each  $i \in V, l' \in L$ .

(R5) For each  $i \in V$ , multiply the inequality  $\sum_{j: (i,j) \in A} x_{ij} \leq 1$  of Constraint (1c) by  $x_{ij'}$  for each  $(i, j') \in A$ , and for each  $i \in V$ , multiply the equation  $\sum_{l \in L} y_i^l = \sum_{j: (i,j) \in A} x_{ij}$  by  $x_{ij'}$  for each  $(i, j') \in A$ .

- **Linearization Phase.** Linearize the new classes of constraints generated by (R1)–(R5) above by using the substitutions

$$u_{ij}^l = y_i^l x_{ij} \quad \text{and} \quad v_{ij}^l = y_j^l x_{ij} \quad \forall (i, j) \in A, \forall l \in L.$$

Reformulation step (R1) produces  $u_{ij}^l \leq v_{ij}^l$  for  $(i, j) \in A, l \in L$  and  $v_{ij}^l \leq u_{ij}^l$  for  $(i, j) \in A, l \in L$ . Therefore, we have  $u_{ij}^l = v_{ij}^l$  for  $(i, j) \in A, l \in L$  and we only use  $u$  variables for the linearization. Step (R2) produces  $\sum_{j: (j,i) \in A} u_{ji}^l = \sum_{j: (i,j) \in A} u_{ij}^l$  for  $i \in V, l \in L$ . Step (R3) gives  $\sum_{j: (i,j) \in A} u_{ij}^l \leq y_i^l$  for  $i \in V, l \in L$ , and surrogates them for each  $l \in L$ , which results in  $\sum_{l \in L} \sum_{j: (i,j) \in A} u_{ij}^l \leq \sum_{l \in L} y_i^l$  for  $i \in V$ . By Constraints (1c) and (1e), we have  $\sum_{l \in L} \sum_{j: (i,j) \in A} u_{ij}^l \leq 1$  for  $i \in V$ . Reformulation step (R4) generates  $y_i^l + \sum_{l' \neq l} y_i^{l'} y_i^{l'} \leq y_i^{l'}$  for  $i \in V$ , which leads to  $\sum_{l' \neq l} y_i^{l'} y_i^{l'} = 0$ . Therefore,  $\sum_{l \in L} y_i^l = \sum_{j: (i,j) \in A} x_{ij}$  multiplied by  $y_i^{l'}$  for each  $i \in V, l' \in L$  yields  $y_i^l = \sum_{j: (i,j) \in A} u_{ij}^l$  for  $i \in V, l \in L$  ( $l'$  is substituted by  $l$  for convenience). Next, surrogating  $y_i^l = \sum_{j: (i,j) \in A} u_{ij}^l$  for each  $i \in V$  gives  $\sum_{i \in V} y_i^l = \sum_{(i,j) \in A} u_{ij}^l$  for  $l \in L$ . Hence, by Constraint (1d), we have  $\sum_{(i,j) \in A} u_{ij}^l \leq k$  for  $l \in L$ . According to the first statement of step (R5) along with  $x_{ij} \geq 0$  for  $(i, j) \in A$ , we can obtain  $\sum_{j \neq j': (i,j) \in A} x_{ij} x_{ij'} = 0$  for  $i \in V$ . The next statement along with the first result produces  $x_{ij} = \sum_{l \in L} u_{ij}^l$  for  $i \in V$ . Therefore, we can replace the objective function of Equation (1a) with  $\sum_{l \in L} \sum_{(i,j) \in A} w_{ij} u_{ij}^l$ . Consequently, after substituting  $x_{ij}^l$  for  $u_{ij}^l$ , the RLT application constructs the EE formulation. We again emphasize that the aforementioned procedures do not rely on any problem-specific structures or insights, but they derive the same compact formulation, the EE formulation, from the EA formulation.

#### 4.2. Extended Formulation Derived from the EE Formulation

In this subsection, we further strengthen the EE formulation by applying the RLT while maintaining the  $y$  variables. The specific RLT procedure we perform is explained in detail below.

- **Reformulation Phase.** Construct the additional set of constraints via (R6)–(R9) stated below.
  - (R6) For each  $i \in V$ , multiply the inequality  $\sum_{j: (i,j) \in A} x_{ij} \leq 1$  of Constraint (1c) by  $y_i^l \geq 0$  for each  $i \in V, l \in L$ .
  - (R7) For each  $(i, j) \in A$  and  $l \in L$ , multiply the inequality  $y_i^l + x_{ij} \leq 1 + y_j^l$  of Constraint (1f) by  $y_i^l \geq 0$  and  $y_j^l$ . Similarly, for each  $(i, j) \in A$  and  $l \in L$ , multiply the inequality  $y_j^l + x_{ij} \leq 1 + y_i^l$  by  $y_i^l \geq 0$ .
  - (R8) For each  $l \in L$ , multiply the inequality  $\sum_{i \in V} y_i^l \leq k$  of Constraint (1d) by  $y_j^l \geq 0$  for each  $j \in V$ .
  - (R9) For each  $i \in V, l \in L, i > l$ , multiply the inequality  $y_i^l \leq y_l^l$  of the symmetry elimination constraints (3) by  $y_i^l \geq 0, y_l^l \geq 0$ , and  $y_j^l \geq 0$  for each  $j \in V$ .
- **Linearization Phase.** Linearize the new classes of constraints generated by (R6)–(R9) above by using the substitutions

$$x_{ij}^l = y_i^l x_{ij} = y_j^l x_{ij} \quad \forall (i, j) \in A, \forall l \in L \quad \text{and} \quad z_{(ij)}^l = y_i^l y_j^l \quad \forall i, j \in V, \forall l \in L.$$

Reformulation step (R6) produces  $\sum_{j: (i,j) \in A} x_{ij}^l \leq y_i^l$  for  $i \in V, l \in L$ . Step (R7) gives three classes of valid inequalities  $x_{ij}^l \leq z_{(ij)}^l, z_{(ij)}^l + x_{ij}^l \leq 2y_j^l$ , and  $z_{(ij)}^l + x_{ij}^l \leq 2y_i^l$  for  $(i, j) \in A, l \in L$ . Step (R8)

generates  $\sum_{j(\neq i) \in V} z_{ij}^l \leq (k-1)y_i^l$  for  $i \in V, l \in L$ . The last step (R9) produces  $y_i^l \leq z_{il}^l \leq y_i^l$  for  $i \in V, l \in L$ , and  $z_{ij}^l \leq z_{ji}^l$  for  $i, j \in V, l \in L$ .

We summarize the overall RLT process above to derive a new compact formulation from the EE formulation in Figure 4. By augmenting the valid inequalities generated above to some set of constraints from the EE and EA formulations, we derive a new integer programming model for the KEP, which is more compact and tighter than the EE formulation, as follows:

$$\text{RLT : Maximize } \sum_{l \in L} \sum_{(i,j) \in A} w_{ij} x_{ij}^l \quad (5a)$$

Subject to Constraints (2b), (2c), (2d), (2e), (1h)

$$\sum_{j: (i,j) \in A} x_{ij}^l \leq y_i^l, \quad \forall i \in V, l \in L \quad (5b)$$

$$x_{ij}^l \leq z_{ij}^l, \quad \forall (i,j) \in A, l \in L \quad (5c)$$

$$z_{ij}^l + x_{ij}^l \leq 2y_j^l, \quad \forall (i,j) \in A, l \in L \quad (5d)$$

$$z_{ij}^l + x_{ij}^l \leq 2y_i^l, \quad \forall (i,j) \in A, l \in L \quad (5e)$$

$$\sum_{j(\neq i) \in V} z_{ij}^l \leq (k-1)y_i^l, \quad \forall i \in V, l \in L \quad (5f)$$

$$y_i^l \leq z_{il}^l \leq y_i^l, \quad \forall i \in V, l \in L \quad (5g)$$

$$z_{ij}^l \leq z_{ji}^l, \quad \forall i, j \in V, l \in L \quad (5h)$$

$$z_{ij}^l \in \{0, 1\}, \quad \forall i, j \in V, l \in L. \quad (5i)$$

### Reformulation

$$\begin{aligned} & \left( \sum_{j: (i,j) \in A} x_{ij} \leq 1 \right) \times y_i^l \\ & (y_i^l + x_{ij} \leq 1 + y_j^l) \times y_j^l \\ & (y_i^l + x_{ij} \leq 1 + y_j^l) \times y_j^l \\ & (y_j^l + x_{ij} \leq 1 + y_i^l) \times y_i^l \\ & \left( \sum_{i \in V} y_i^l \leq k \right) \times y_i^l \\ & (y_i^l \leq y_i^l) \times y_i^l \\ & (y_i^l \leq y_i^l) \times y_i^l \\ & (y_i^l \leq y_i^l) \times y_i^l \end{aligned}$$

### Linearization

$$\begin{aligned} z_{ij}^l & \equiv y_i^l \times y_j^l \\ z_{ij}^l & = \begin{cases} z_{ij}^l, & i < j \\ z_{ji}^l, & i > j \end{cases} \end{aligned}$$

### Resulting valid inequalities

$$\begin{aligned} & \sum_{j: (i,j) \in A} x_{ij}^l \leq y_i^l, \quad \forall i \in V, l \in L \\ & x_{ij}^l \leq z_{ij}^l, \quad \forall (i,j) \in A, l \in L \\ & z_{ij}^l + x_{ij}^l \leq 2y_j^l, \quad \forall (i,j) \in A, l \in L \\ & z_{ij}^l + x_{ij}^l \leq 2y_i^l, \quad \forall (i,j) \in A, l \in L \\ & \sum_{j(\neq i) \in V} z_{ij}^l \leq (k-1)y_i^l, \quad \forall i \in V, l \in L \\ & y_i^l \leq z_{il}^l \leq y_i^l, \quad \forall i \in V, l \in L \\ & z_{ij}^l \leq z_{ji}^l, \quad \forall i, j \in V, l \in L \end{aligned}$$

**Figure 4.** Illustration of the reformulation-linearization technique (RLT) for deriving new valid inequalities.

## 5. Computational Results

In this section, we present the results of the computational experiments for the KEP. The experiments were performed using an Intel(R) Core(TM) i7-4770 CPU @ 3.40 GHz, and CPLEX 12.4 as a LP/MIP (mixed integer programming) optimization solver. The experiments were designed to compare the EA, EE, and RLT formulations in terms of performance. Problem instances were

constructed as by Constantino et al. [18]. A random generator that created random KEP graphs of chosen certain arc densities was used. To be specific, on each position of the node adjacency matrix of a graph, a probability  $p$  is used to determine whether the element equals 1 or 0. Constantino et al. [18] used values of 0.2, 0.5, and 0.7 for the probability  $p$ . However, it was found that with probabilities of 0.5 and 0.7, almost every instance had an optimal objective  $n$  equal to the number of nodes. Because  $n$  is the trivial upper bound of these KEP problems, no formulation can improve the linear relaxation bound of such problem instances. Thus, because an important goal of this paper is to compare the tightness of KEP formulations,  $p$  was set to 0.1, 0.2, and 0.5 for low-, medium-, and high-density instances, respectively. For each density, we generated the problem instances with the input parameter  $k = 3, 4, 5, 6$ . In addition,  $n$ , the number of pairs, was further set to 10, 20, and 30 for each  $k$ . For each  $n$  and  $k$ , 10 instances were generated and tested. All the cases were executed within a time limit of 3600 s. The formulations were compared in terms of the optimality gap (GAP) and the execution time. The GAP is calculated by  $(Z_{LP} - Z_{IP})/Z_{IP}$ , where  $Z_{IP}$  is the best objective function value of the integer programming model and  $Z_{LP}$  is the optimal objective function value of the corresponding linear programming relaxation. Hence, the lower the GAP obtained, the better the solution (i.e., closer to the optimal solution). On the other hand,  $T_{IP}$  and  $T_{LP}$  are the average elapsed time of the integer programming model and its linear programming relaxation, respectively.

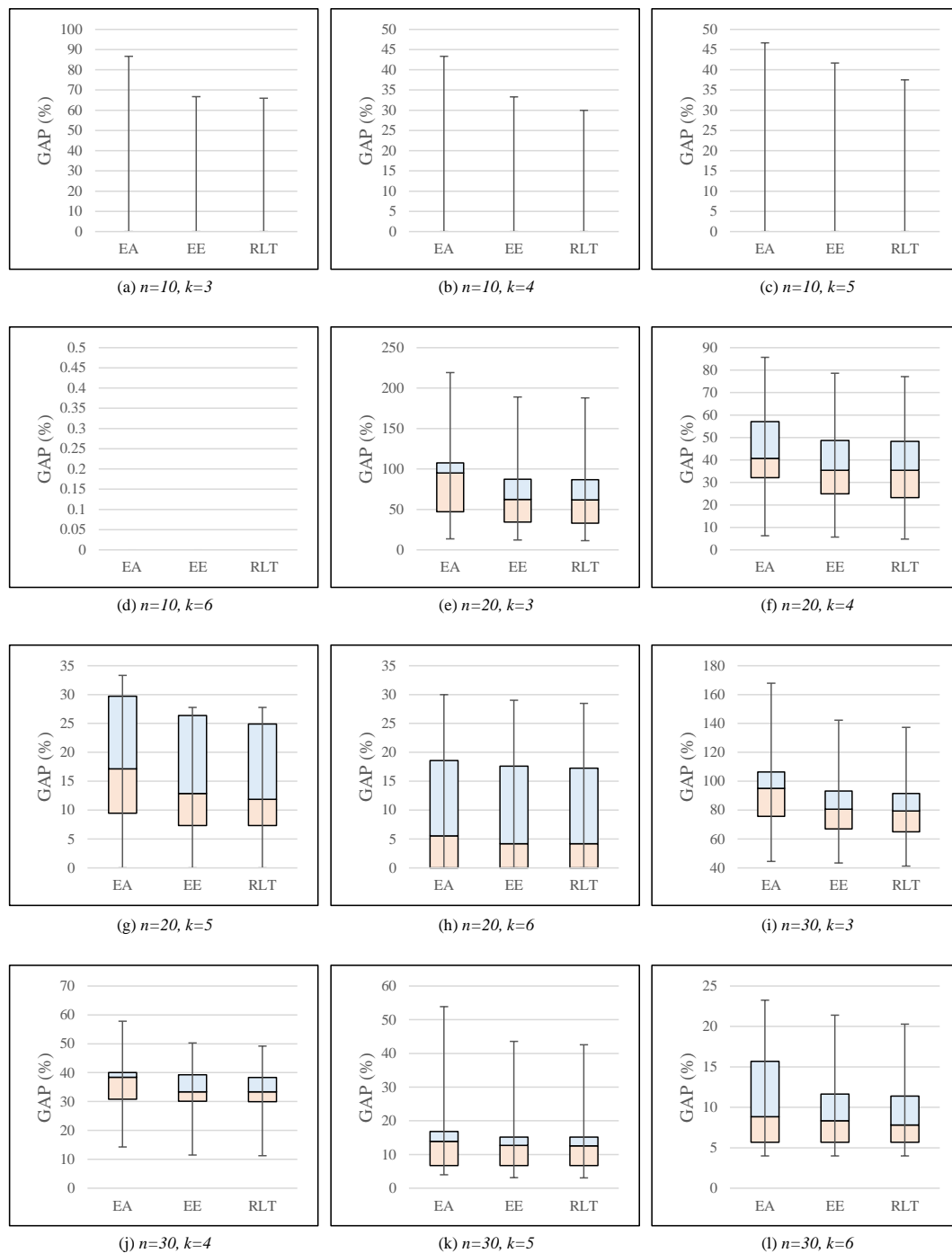
**Table 1.** Computational results for low-density instances. EA: edge-assignment; EE: extended edge; RLT: reformulation-linearization technique; GAP: optimality gap.

Problem Instances		EA			EE			RLT		
		$T_{LP}$	$T_{IP}$	GAP	$T_{LP}$	$T_{IP}$	GAP	$T_{LP}$	$T_{IP}$	GAP
$k = 3$	$n = 10$	0.00	0.00	10.7%	0.00	0.00	8.7%	0.00	0.01	6.7%
	$n = 20$	0.00	0.01	87.5%	0.00	0.00	68.2%	0.02	0.03	67.8%
	$n = 30$	0.02	0.53	98.1%	0.01	0.01	84.2%	0.29	0.11	83.9%
$k = 4$	$n = 10$	0.00	0.00	4.3%	0.00	0.00	3.3%	0.00	0.00	3.0%
	$n = 20$	0.00	0.01	45.1%	0.00	0.00	39.0%	0.00	0.00	38.1%
	$n = 30$	0.02	1.68	37.3%	0.01	0.02	33.2%	0.22	0.05	33.1%
$k = 5$	$n = 10$	0.00	0.00	4.7%	0.00	0.00	4.2%	0.00	0.00	3.8%
	$n = 20$	0.00	0.01	17.9%	0.00	0.01	14.8%	0.02	0.03	14.1%
	$n = 30$	0.02	0.76	18.0%	0.01	0.07	15.9%	0.34	0.10	14.9%
$k = 6$	$n = 10$	0.00	0.00	0.0%	0.00	0.00	0.0%	0.00	0.00	0.0%
	$n = 20$	0.00	0.01	10.5%	0.00	0.01	9.6%	0.02	0.04	9.5%
	$n = 30$	0.01	0.59	10.9%	0.01	0.12	9.6%	0.59	0.27	9.6%

Table 1 summarizes the computational results for low-density instances. The EE formulation found the optimal upper bound for 43 out of 120 low-density instances. Of the remaining 77 instances for which the EE formulation could not find the optimal solution, the upper bounds of 36 instances were improved by our RLT model. Subsequently, the average value of the GAP was improved by our RLT formulation. On the other hand, the improvement in the GAP obtained by the RLT formulation tended to be smaller as  $k$  grew.

Figure 5 presents box plots of the GAPs (%) produced by each formulation from problem instances for each combination of  $(n, k)$ . From the figure, we can observe that our RLT formulation yielded the lowest average GAPs, and that furthermore, the variability of the GAPs produced by our RLT formulation was also lower than that of the best-known compact formulations. Specifically, both the maximum and minimum GAPs, as well as the difference between the lower and upper quartiles, significantly dropped from the EA formulation to the EE formulation, and then further again to the RLT formulation. Moreover, such observations were consistent for every combination of  $(n, k)$ . The results imply that the EA formulation is highly dominated by the EE and RLT formulations, and moreover, that our proposed RLT formulation is more reliable than the EE formulation in terms of

solution quality—that is, our approach produces more reliable and higher quality solutions than the best-known compact formulations, without compromising computational time.



**Figure 5.** Box plots of optimality gaps (GAPs) for low-density instances.

Similarly, Table 2 demonstrates the computational results for medium-density instances. According to the results, the average GAP was smaller than for the low-density instances for all the formulations. Having such a result is intuitive because the medium-density graph had more arcs representing compatibility; thus, feasible solutions could be more easily found than in the low-density cases. In the experiments, the EE formulation found the optimal upper bounds for 83 out of 120 instances. Our RLT formulation improved the upper bounds of 17 out of the remaining

37 instances whose optimal solutions were not obtained by the EE formulation. When it comes to the box plots of the GAPs for the medium-density problem instances, we are able to make similar observations to those for the low-density instances as presented in Figure 6. As mentioned previously, compared to the low-density case, there were more combinations of  $(n, k)$  for which the GAPs of all the problem instances were equal to zero (i.e., (j), (k) and (l) in Figure 6).

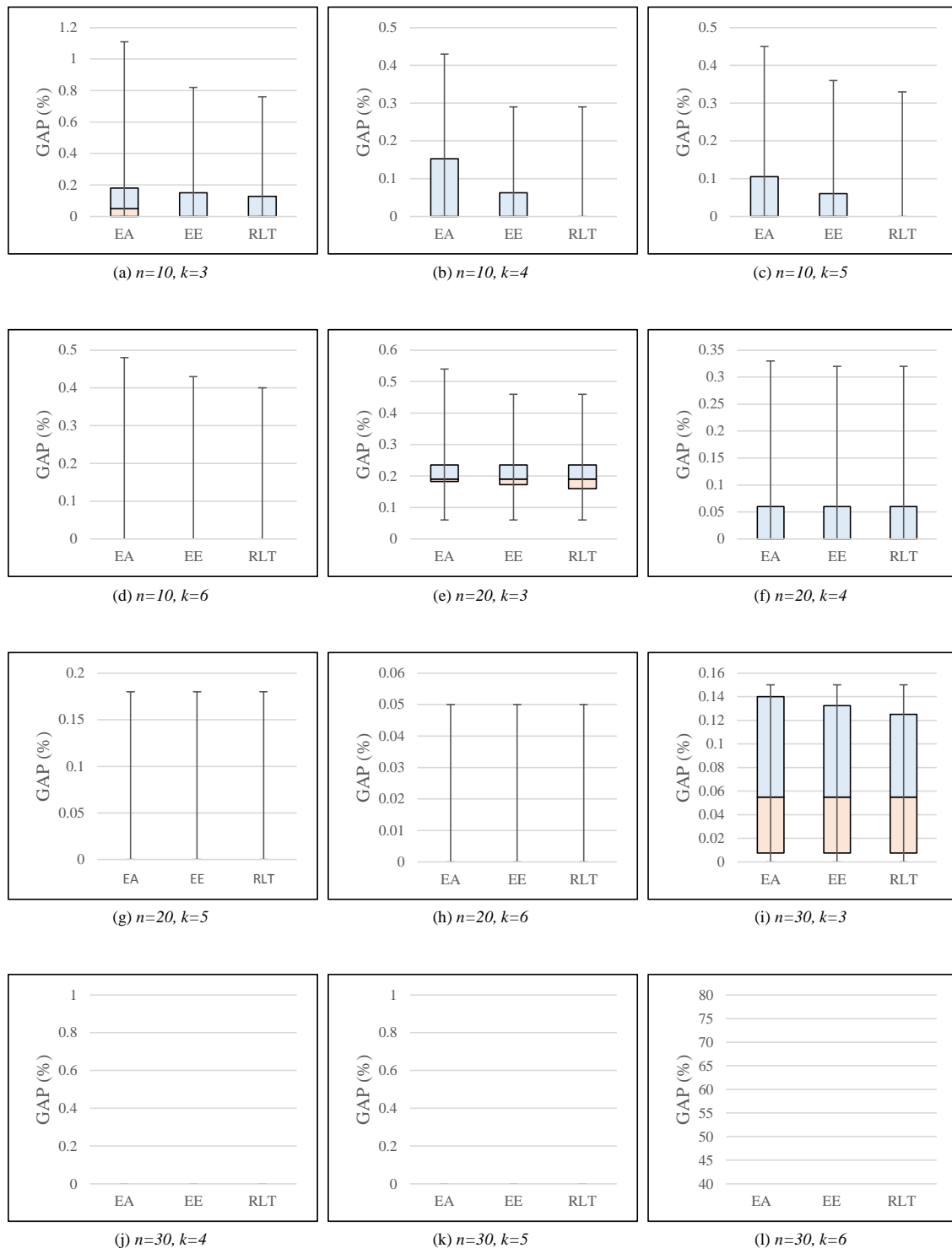


Figure 6. Box plots of optimality gaps (GAPs) for medium-density instances.

**Table 2.** Computational results for medium-density instances.

Problem Instances		EA			EE			RLT		
		$T_{LP}$	$T_{IP}$	GAP	$T_{LP}$	$T_{IP}$	GAP	$T_{LP}$	$T_{IP}$	GAP
$k = 3$	$n = 10$	0.00	0.00	22.0%	0.00	0.00	16.2%	0.00	0.00	16.0%
	$n = 20$	0.01	0.95	22.2%	0.00	0.01	20.8%	0.05	0.07	20.5%
	$n = 30$	0.03	238.11	7.1%	0.03	0.09	7.0%	0.05	0.12	6.9%
$k = 4$	$n = 10$	0.00	0.00	9.7%	0.00	0.02	6.3%	0.00	0.01	5.1%
	$n = 20$	0.01	0.41	6.2%	0.00	0.06	6.0%	0.53	0.41	6.0%
	$n = 30$	0.02	100.95	0.0%	0.03	0.93	0.0%	0.68	1.55	0.0%
$k = 5$	$n = 10$	0.00	0.00	7.8%	0.00	0.00	5.7%	0.00	0.01	3.8%
	$n = 20$	0.01	0.22	2.9%	0.00	0.07	2.9%	0.40	0.58	2.9%
	$n = 30$	0.02	12.55	0.0%	0.02	0.38	0.0%	0.29	0.67	0.0%
$k = 6$	$n = 10$	0.00	0.00	6.3%	0.00	0.00	5.5%	0.00	0.01	4.8%
	$n = 20$	0.01	0.11	1.1%	0.00	0.07	1.1%	0.27	0.76	1.1%
	$n = 30$	0.02	3.91	0.0%	0.03	0.25	0.0%	0.19	0.42	0.0%

Finally, the computational results for the high-density instances are presented in Table 3. We observe that the GAPs of all the formulations were equal to 0 for all the instances; hence, we here compare them in terms of the computational time only. As presented in Table 3, in terms of computational time, the RLT formulation outperformed the EA formulation, but the EE formulation performed slightly better than the RLT formulation. However, both cases were reasonably acceptable for practical-size problems. We remark that high-density instances rarely appear in real-world situations, as a donor is unlikely to find as many compatible patients as in high-density cases in a given pool of candidates.

**Table 3.** Computational results for high-density instances.

Problem Instances		EA			EE			RLT		
		$T_{LP}$	$T_{IP}$	GAP	$T_{LP}$	$T_{IP}$	GAP	$T_{LP}$	$T_{IP}$	GAP
$k = 3$	$n = 10$	0.00	0.02	0.0%	0.00	0.01	0.0%	0.02	0.03	0.0%
	$n = 20$	0.01	1.20	0.0%	0.01	0.14	0.0%	0.96	0.31	0.0%
	$n = 30$	0.03	57.15	0.0%	0.03	0.71	0.0%	0.86	1.87	0.0%
$k = 4$	$n = 10$	0.00	0.03	0.0%	0.00	0.01	0.0%	0.01	0.03	0.0%
	$n = 20$	0.01	0.32	0.0%	0.01	0.09	0.0%	0.68	0.49	0.0%
	$n = 30$	0.03	35.29	0.0%	0.04	0.43	0.0%	0.82	0.99	0.0%
$k = 5$	$n = 10$	0.00	0.01	0.0%	0.00	0.01	0.0%	0.01	0.02	0.0%
	$n = 20$	0.01	0.23	0.0%	0.01	0.08	0.0%	0.65	0.48	0.0%
	$n = 30$	0.03	10.02	0.0%	0.04	0.29	0.0%	0.57	0.88	0.0%
$k = 6$	$n = 10$	0.00	0.01	0.0%	0.00	0.01	0.0%	0.01	0.02	0.0%
	$n = 20$	0.01	0.17	0.0%	0.01	0.07	0.0%	0.51	0.46	0.0%
	$n = 30$	0.02	2.53	0.0%	0.04	0.26	0.0%	1.04	0.82	0.0%

## 6. Conclusions

In this paper, we first show that the EE formulation, until now considered to be the most compact formulation, can be systematically derived from the EA formulation through the RLT process. A previous study developed the EE formulation by exploiting special structures inherent to the problem. However, we show that we can readily apply the RLT procedure to the EA formulation to obtain the same EE formulation. Furthermore, we apply the RLT to the EA and EE formulations and thereby derive a tighter and more compact formulation. Computational results show that the proposed RLT model outperforms the existing EA and EE formulations in terms of solution quality as well as computational time. Specifically, the RLT model provides improved GAPs in several instances

for which the EE formulation could not find the optimal solutions within the time limit—especially for low- and medium-density graphs. For high-density instances, although the RLT model does not significantly reduce the execution time, it can produce the same high-quality solutions as the existing formulations. We believe that, given the results obtained by the proposed model, it could be utilized in the integrated KEP IT healthcare platform to obtain optimized KEP exchange plans.

As extended research, several models derived from that proposed in this paper can be considered. For example, firstly, a problem dealing with fair allocations, such as in terms of donor and patient age and health, among the pairs in the candidate pool may be interesting. Indeed, some problems of fairness in kidney transplantation or kidney exchange are discussed by Bertsimas et al. [27] and by Anderson [15] respectively in literature. Secondly, some limitations of this study include that all data (e.g.,  $w_{ij}$ ) were considered to be certain. When a KEP solution is actually implemented, it may be possible to have several unexpected events, including last-minute incompatibility and unavailability of either donors or recipients [28,29]. To handle the data or event uncertainty in KEPs, Alvelos et al. [29], for example, examined an optimization problem whose objective was to maximize the expected number of transplants in a KEP given that all the arcs and vertices in a KEP graph have equal probabilities of failure, and there are some attempts to apply robust optimization theory [30,31] to a KEP [28]. Thus, we can extend our model to incorporate the aforementioned uncertainties and take the solution approaches, such as stochastic optimization or robust optimization techniques, into account. Finally, proposing an efficient heuristic-based approach for solving highly large-scale problems more generally would be the topic for future research.

**Acknowledgments:** This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(Ministry of Science, ICT & Future Planning) (No. NRF-2015R1C1A1A02036682).

**Author Contributions:** Junsang Yuh developed the mathematical programming models and performed the experiments. Seokhyun Chung contributed to the literature review, experimental part and result analysis. Taesu Cheong guided the whole research, supported the structure of the paper and prepared the manuscript. All authors read and approved the final manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Kwak, J.Y.; Kwon, O.J.; Lee, K.S.; Kang, C.M.; Park, H.Y.; Kim, J.H. Exchange-donor program in renal transplantation: A single-center experience. *Transplant. Proc.* **1999**, *31*, 344–345.
2. Park, K.; Lee, J.H.; Huh, K.H.; Kim, S.I.; Kim, Y.S. Exchange living-donor kidney transplantation: Diminution of donor organ shortage. *Transplant. Proc.* **2004**, *36*, 2949–2951.
3. Huh, K.H.; Kim, M.S.; Ju, M.K.; Chang, H.K.; Ahn, H.J.; Lee, S.H.; Park, K. Exchange living-donor kidney transplantation: Merits and limitations. *Transplantation* **2008**, *86*, 430–435.
4. Ellison, M.D.; McBride, M.A.; Taranto, S.E.; Delmonico, F.L.; Kauffman, H.M. Living kidney donors in need of kidney transplants: A report from the organ procurement and transplantation network. *Transplantation* **2002**, *74*, 1349–1351.
5. Segev, D.L.; Gentry, S.E.; Warren, D.S.; Reeb, B.; Montgomery, R.A. Kidney paired donation and optimizing the use of live donor organs. *JAMA* **2005**, *293*, 1883–1890.
6. Saidman, S.L.; Roth, A.E.; Sönmez, T.; Ünver, M.U.; Delmonico, F.L. Increasing the opportunity of live kidney donation by matching for two- and three-way exchanges. *Transplantation* **2006**, *81*, 773–782.
7. Abraham, D.J.; Blum, A.; Sandholm, T. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In Proceedings of the 8th ACM Conference on Electronic Commerce, San Diego, CA, USA, 11–15 June 2007; pp. 295–304.
8. Veale, J.; Hil, G. National Kidney Registry: 213 transplants in three years. *Clin. Transpl.* **2010**, 333–344. Available online: [http://www.kidneyregistry.org/docs/nkr\\_transplants\\_2010\\_chain\\_chapter.pdf](http://www.kidneyregistry.org/docs/nkr_transplants_2010_chain_chapter.pdf) (accessed on 10 April 2017).
9. Wallis, C.; Samy, K.; Roth, A.; Rees, M. Kidney paired donation. *Nephrol. Dial. Transplant.* **2011**, *26*, 2091–2099, doi:10.1093/ndt/gfr155.

10. Johnson, R.; Collett, D.; Birch, R.; Fuggle, S.; Rudge, C. Kidney donation and transplantation in the UK from 1998 to 2007. *Clin. Transpl.* **2008**, 75–88, doi:10.1201/9781420086423.ch10.
11. Johnson, R.J.; Allen, J.E.; Fuggle, S.V.; Bradley, J.A.; Rudge, C. Early experience of paired living kidney donation in the United Kingdom. *Transplantation* **2008**, 86, 1672–1677.
12. Biro, P.; Manlove, D.F.; Rizzi, R. Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discret. Math. Algorithms Appl.* **2009**, 1, 499–517.
13. Ferrari, P.; Woodroffe, C.; Christiansen, F.T. Paired kidney donations to expand the living donor pool: The Western Australian experience. *Med. J. Australia* **2009**, 190, 700.
14. Rais, A.; Viana, A. Operations Research in Healthcare: A Survey. *Int. Trans. Oper. Res.* **2011**, 18, 1–31.
15. Anderson, R.M. Stochastic Models and Data Driven Simulations for Healthcare Operations. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2014.
16. D'Andreagiovanni, F.; Nardin, A. Towards the Fast and Robust Optimal Design of Wireless Body Area Networks. *Appl. Soft Comput.* **2015**, 37, 971–982.
17. Roth, A.E.; Sönmez, T.; Ünver, M.U. Efficient Kidney Exchange: Coincidence of Wants in Markets with Compatibility-Based Preferences. *Am. Econ. Rev.* **2007**, 97, 828–851.
18. Constantino, M.; Klimentova, X.; Viana, A.; Rais, A. New insights on integer-programming models for the kidney exchange problem. *Eur. J. Oper. Res.* **2013**, 231, 57–68.
19. Manlove, D.F.; O'Malley, G. Paired and Altruistic Kidney Donation in the UK: Algorithms and Experimentation. *ACM J. Exp. Algorithmics* **2015**, 19, doi:10.1145/2670129.
20. Sherali, H.D.; Driscoll, P.J. On Tightening the Relaxations of Miller-Tucker-Zemlin Formulations for Asymmetric Traveling Salesman Problems. *Oper. Res.* **2002**, 50, 656–669.
21. Desrochers, M.; Laporte, G. Improvements and Extensions to the Miller-Tucker-Zemlin Subtour Elimination Constraints. *Oper. Res. Lett.* **1991**, 10, 27–36.
22. Sherali, H.D.; Sarin, S.C.; Tsai, P. A Class of Lifted Path and Flow-Based Formulations for the Asymmetric Traveling Salesman Problem with and without Precedence Constraints. *Discret. Optim.* **2006**, 3, 20–32.
23. Park, G.; Lee, Y.; Han, J. A Two-Level Location-Allocation Problem in Designing Local Access Fiber Optic Networks. *Comput. Oper. Res.* **2014**, 51, 52–63.
24. Sherali, H.D.; Adams, W.P. A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. *Discret. Appl. Math.* **1994**, 52, 83–106.
25. Sherali, H.D.; Lee, Y.; Kim, Y. Partial convexification cuts for 0–1 mixed-integer programs. *Eur. J. Oper. Res.* **2005**, 165, 625–648.
26. Sherali, H.D.; Adams, W.P. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discret. Math.* **1990**, 3, 411–430.
27. Bertsimas, D.; Farias, V.F.; Trichakis, N. Fairness, Efficiency, and Flexibility in Organ Allocation for Kidney Transplantation. *Oper. Res.* **2013**, 61, 73–87.
28. Dickerson, J.P. Robust Dynamic Optimization with Application to Kidney Exchange. In Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems, Paris, France, 5–9 May 2014; pp. 1701–1702.
29. Alvelos, F.; Klimentova, X.; Rais, A. A Compact Formulation for Maximizing the Expected Number of Transplants in Kidney Exchange Programs. *J. Phys. Conf. Ser.* **2015**, 616, 012011.
30. Bertsimas, D.; Sim, M. The Price of Robustness. *Oper. Res.* **2004**, 52, 35–53.
31. Büsing, C.; D'Andreagiovanni, F. New Results about Multi-Band Uncertainty in Robust Optimization. In *Experimental Algorithms—SEA 2012, Lecture Notes in Computer Science*; Klasing, R., Ed.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7276, 63–74.

