

RESEARCH ARTICLE

Ant Lion Optimization algorithm for kidney exchanges

Eslam Hamouda^{1*}, Sara El-Metwally¹, Mayada Tarek¹

Computer Science Department, Faculty of Computers and Information, Mansoura University, Mansoura, Dakahlia, Egypt

¹ These authors contributed equally to this work.

* eslam_foad@mans.edu.eg



OPEN ACCESS

Citation: Hamouda E, El-Metwally S, Tarek M (2018) Ant Lion Optimization algorithm for kidney exchanges. PLoS ONE 13(5): e0196707. <https://doi.org/10.1371/journal.pone.0196707>

Editor: Xiangtao Li, Northeast Normal University, CHINA

Received: November 2, 2017

Accepted: April 18, 2018

Published: May 3, 2018

Copyright: © 2018 Hamouda et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: A reference implementation of ALO algorithm for kidney exchanges is available as a free open-source software at the github repository: https://github.com/SaraEl-Metwally/ALO_algorithm_for_Kidney_Exchanges. All other relevant data are within the paper. We also provide the set of scripts and commands to replicate our simulated data sets and our results that are mentioned in the manuscript. The simulated data sets are available in the folder named "Simulation-UNOS" in our github repository. The direct link to this folder is: https://github.com/SaraEl-Metwally/ALO_algorithm_for_Kidney_Exchanges/tree/master/Simulation-UNOS. The

Abstract

The kidney exchange programs bring new insights in the field of organ transplantation. They make the previously not allowed surgery of incompatible patient-donor pairs easier to be performed on a large scale. Mathematically, the kidney exchange is an optimization problem for the number of possible exchanges among the incompatible pairs in a given pool. Also, the optimization modeling should consider the expected quality-adjusted life of transplant candidates and the shortage of computational and operational hospital resources. In this article, we introduce a bio-inspired stochastic-based Ant Lion Optimization, ALO, algorithm to the kidney exchange space to maximize the number of feasible cycles and chains among the pool pairs. Ant Lion Optimizer-based program achieves comparable kidney exchange results to the deterministic-based approaches like integer programming. Also, ALO outperforms other stochastic-based methods such as Genetic Algorithm in terms of the efficient usage of computational resources and the quantity of resulting exchanges. Ant Lion Optimization algorithm can be adopted easily for on-line exchanges and the integration of weights for hard-to-match patients, which will improve the future decisions of kidney exchange programs. A reference implementation for ALO algorithm for kidney exchanges is written in MATLAB and is GPL licensed. It is available as free open-source software from: https://github.com/SaraEl-Metwally/ALO_algorithm_for_Kidney_Exchanges.

Introduction

The number of terminal-stage renal patients has significantly risen around the world. In this final stage of disease, kidneys fail to meet patient's life needs, it may eventually lead to a complete kidney failure. The only available treatment for those patients is a kidney transplant operation [1]. There are two sources for kidneys transplantation. The first way is through deceased donors. However, this way can not satisfy the increasing number of kidney patients, by the end of 2016, there is a waiting list of more than 96,000 patients who need a kidney transplant operation in United States [2]. The other kidney transplantation way is through willing living donors. Recently, the number of kidney transplantation through willing living donors has

simulation scripts along with the read-me file to replicate our results can be found in the folder called Saidman Generator Simulation.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

increased [1]. Transplanting kidneys from living donors have higher rate of success than deceased donors [3].

It is common that a willing donor is ready to donate his kidney, but his blood and tissue types are not compatible with the patient (incompatible donor-patient pair) [4]. Having a set of incompatible pairs, there is a possibility that a donor of a pair i is compatible with the patient of a pair j and vice versa. This case achieves mutual benefits among donor-patient pairs. Therefore, the goal of a kidney exchange program, also called Kidney paired donation (KPD), is to maximize the number of possible matches among the incompatible pairs in a given kidney exchange pool. [3], and hence enhancing the success of kidney transplantation through living donors.

Generally, KPD could be applied in static or dynamic environment [4]. In static environment, KPD attempts to find possible matches that are found in the current pool, i.e., providing local solutions without taking future events in consideration. In dynamic environment, pool size periodically changes, existing pairs can leave because patient passes away or he successfully receives a kidney (cured), or new pairs can arrive. KPD attempts to match as many pairs by taking into account future events. In this case, KPD first searches for a match for hard-to-match pairs (i.e., patients with a rare blood type) and postpones easy-to-match pairs to the next round, which will increase the number of kidney transplantation and save more lives.

Solving a kidney exchange problem efficiently is still an active area of research. A standard tree search algorithm is used as one of the early starting solution to this problem and has a memory space limitation when the size of the patient-donor pairs is increased [5]. Other researchers in this field attempt to solve this problem by formalizing the kidney exchange as an optimization problem. Integer programming approach is used to find the optimal matches among incompatible pairs. The goal of the optimization process is to maximize the number (quantity) or utility (quality) of transplants. In [1, 5–9], Integer programming is used to solve some variants of kidney exchanges with/without chains and make a bound on the pool size or the number of constraints in the matching process. Also, kidney exchange problem can be viewed as a travelling salesperson problem and can be solved using a recursive integer programming formulation [10].

Despite the advantages introduced by the variant techniques that rely on integer programming approach, no stochastic-based algorithm is introduced and used to solve the kidney exchange problem efficiently. This opens a race to solve this problem using different types of heuristics. In [11], authors present a novel model for enhancing matching in the dynamic environment of networked markets. The proposed model has a set of parameters to learn. Inspired by this model, a new trend of kidney exchange algorithms based on stochastic approaches is introduced [4]. This approach employs a learning strategy for optimal solution searching. The methods in [4, 12] used genetic algorithm (GA) with different represented solutions. GA is utilized to search for the best matches in a given KPD pool. The proposed methods achieve reasonable results in terms of the number of resulting transplants compared to the best known methods. However, the computational running time needed to provide these solutions is significantly long, which is impractical to real life situations. Also, due to the large search space, classical GA searching strategy suffers from the local optimal solution, which in turn affects the overall performance. Moreover, the adopted methods are not flexible to be applied in the dynamic environment [4].

In the view of these limitations, this paper presents a novel stochastic based method for kidney exchanges. Ant Lion Optimization (ALO) algorithm is proposed to optimally find the possible matches in a kidney exchange pool. ALO is a recent meta-heuristic algorithm that is computationally less expensive than other methods. It provides an improving search strategy in terms of exploration, exploitation, and convergence [13]. A post-processing stage follows

the Ant lion optimization algorithm is also introduced to enhance the utility of the resulting exchanges. Moreover, the proposed method can be easily adopted in the dynamic kidney exchanges.

The rest of this paper is organized as follows: a background information about the kidney donation programs, and a brief introduction of the general ant lion optimization algorithm are given in Section 1. The proposed ant lion optimization for kidney exchanges is introduced in Section 2. Experimental results are presented in Section 3. Finally, Section 4 concludes this paper with a brief discussion of our produced results.

1 Background

1.1 Kidney paired donations

Kidney transplantation is the only hope for curing patients with terminal-stage renal disease. The increasing number of waiting patients on the scheduled list for transplantation and stagnant rate of kidney donations could break this hope of saving more lives. Some patients have relatives that are willing to donate their kidney but they have incompatible blood or tissue types. These patients constitute donor-patient couples and create a pool of pairs that are able to exchange their kidneys based on their compatibility, allowing what is called recently kidney paired donation program [14].

When this program was first executed, the hospitals arranged the exchange locally among their patients. With the increasing number of patients, the kidney shortage, and the availability of databases and registries, the organization of kidney exchange programs are globally transformed and controlled by national and international centers. The National Kidney Registry (NKR) [15], the Alliance for Paired Donation (APD) [16, 17], and the United Network for Organ Sharing (UNOS) [18] are among the emerging organizations for kidney exchange programs on a large scale in the USA. Other emerging centers are reported from several countries such as UK [19], Canada [20], and Australia [21].

One of the major goals of KPD programs is allowing more kidney exchange to take place while improving the efficiency of kidney matching by searching for a better match in less waiting and dialysis time (i.e., the kidney donation from a living person lasts longer than the kidney from a deceased one and the long waiting list causes that some patients are died and others are becoming too ill for surgery).

The two important factors that play a key role on a kidney matching decisions are ABO blood types and Human Leukocyte Antigen tissue type, HLA. There are four blood types a patient could have, A, B, AB, or O. The matching compatibility rules between the patients and donors according to their blood types are described in Table 1. Donors with blood type O, a general donor, can donate to any patients while patients with type AB, general recipient, can receive a kidney from any donors. HLA tissue type consists of six proteins that are required to match between the donors and patients to successfully transplant a kidney. Moreover, there is a possibility of the positive cross match between the patient and the donor but the patient's

Table 1. Compatibility rules based on ABO blood type.

Donor/Patient	A	B	AB	O
A	✓		✓	
B		✓	✓	
AB			✓	
O	✓	✓	✓	✓

<https://doi.org/10.1371/journal.pone.0196707.t001>

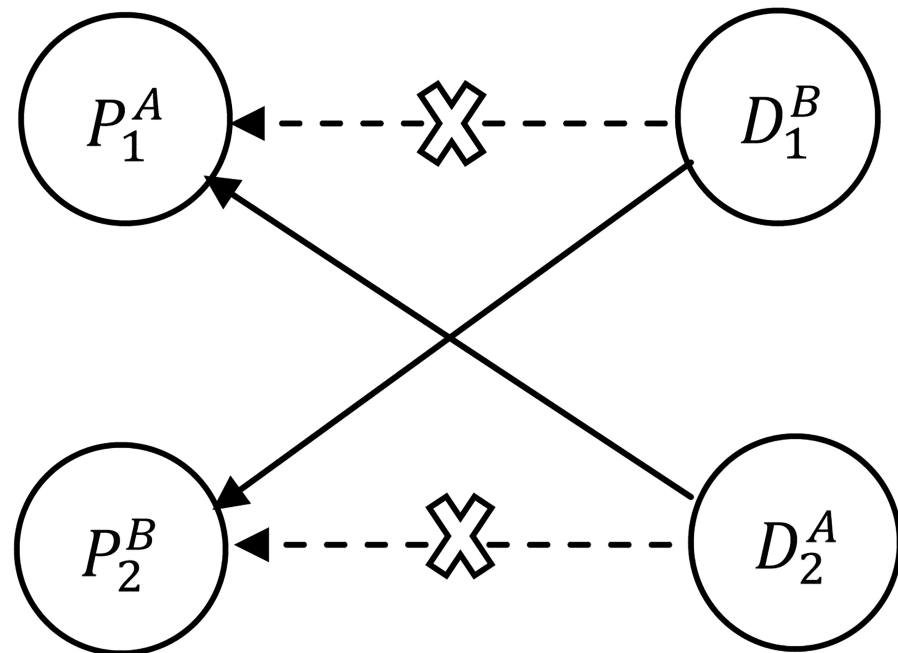


Fig 1. Two-cycle exchange. P_1^A blood type is incompatible with his donor D_1^B , the incompatibility is represented by dotted lines. The two incompatible pairs can swap their kidneys as illustrated by solid lines.

<https://doi.org/10.1371/journal.pone.0196707.g001>

body has antibodies that target the donor’s HLA, which precludes the kidney transplantation [22].

The number of donor-patient pairs in a given pool shapes the complexity of kidney exchange problem. The simple 2-way or 2-cycle exchange is represented by alternating directed cycle of two incompatible donor-patient pairs where the donor of the first incompatible pair gives a kidney to the patient of the second incompatible pair and vice versa. (Fig 1 illustrates 2-way or 2-cycle exchange). The size of exchange cycle can be scaled up to three or more by including more donor-patient pairs in the exchange pool. Also, the pool can be considered as a dynamic i.e., on-line, if the donor-patient pairs are appeared and expired over a time, otherwise it is static or off-line [23, 24].

The long exchange cycle, k -cycle, where $k \geq 3$ allows more options for kidney exchange and consequently better optimization for solving the kidney matching problem. When the number of pairs in an exchange pool is not fixed to a number n , i.e., $k = n$, the problem is considered as an assignment problem with a polynomial solving time [5]. In the reality, the number of pairs in the pool and consequently the length of an exchange cycle should be bounded for the two reasons. First, all operations involved in a cycle must be performed simultaneously to guarantee the commitment of kidney exchange process between the incompatible pairs. This will introduce additional overhead on the hospitals with limited resources. Second, the long exchange cycle might have unpredictable events that cause the cancellation of the exchange process. For example, the patient death or the new incompatibility issues that are appeared between the pairs on the last minute can preclude the exchange. The kidney matching problem with two-cycle can be solved optimally in a polynomial time using Edmonds maximum cardinality matching algorithm while the large scale k -cycle is difficult to solve and considered as NP-complete problem [25].

The kidney exchange problem is formulated as a directed graph with the nodes being the exchange pairs in the pool. The directed edges encode the constrains of the compatibility

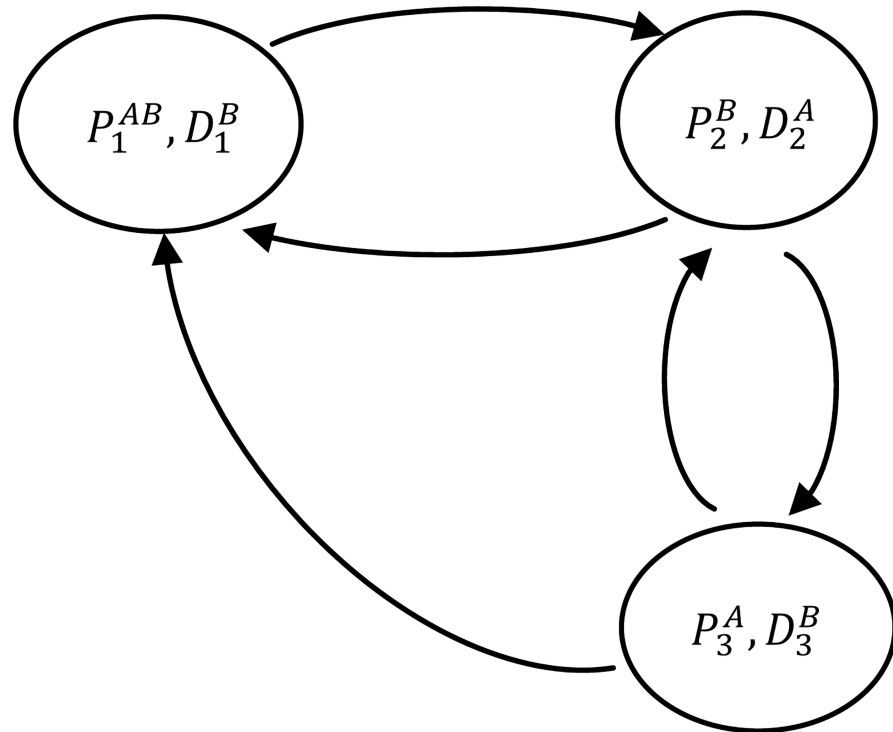


Fig 2. A directed graph representation of three-cycle exchange. The solid arcs represent the potential swaps, i.e. the two pairs (P_1^{AB}, D_1^B) and (P_2^B, D_2^A) can exchange their kidneys.

<https://doi.org/10.1371/journal.pone.0196707.g002>

matches among the pairs (Fig 2 shows a directed graph of three-cycle exchange). The goal of any kidney exchange program is to find a tour in the graph that includes every edge exactly once and provides mutual benefits. This type of tour is called a feasible tour or a feasible cycle.

There are some variants of kidney exchange problem formulation such as initiating the exchange by a donor who is not associated with any patient and is willing to donate a kidney, i.e., altruistic donor. The exchange starts with an altruistic donor is called a chain or non-directed exchange. The altruistic donor donates a kidney to a patient of one pairs in the kidney exchange pool and the associated patient’s donor will exchange a kidney with the first compatible patient on the deceased donors waiting list. Never-Ending Altruistic Donor, NEAD, is a variant of altruistic kidney exchange where the recipient’s donor is not associated with any waiting list but donates a kidney to a compatible patient. The patient’s incompatible donor continues this cascading chain by donating a kidney in the same way and so on (theoretically, this cascading chain never ends) [26] (Fig 3 illustrates an example of NEAD). Another variant of a kidney exchange problem is including of compatible pairs and multiple donors, which will increase the possibility of finding more cycles and chains with many incompatible pairs in the kidney exchange pool [5, 27].

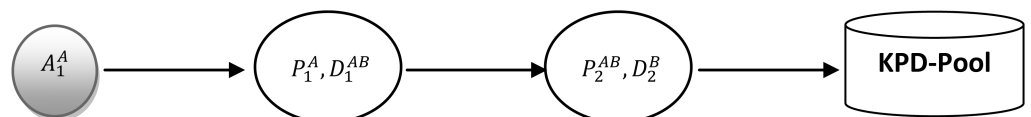


Fig 3. A NEAD exchange graph. A chain starts with an altruistic donor and ends with a bridge donor assigned to the first compatible patient on the next round.

<https://doi.org/10.1371/journal.pone.0196707.g003>

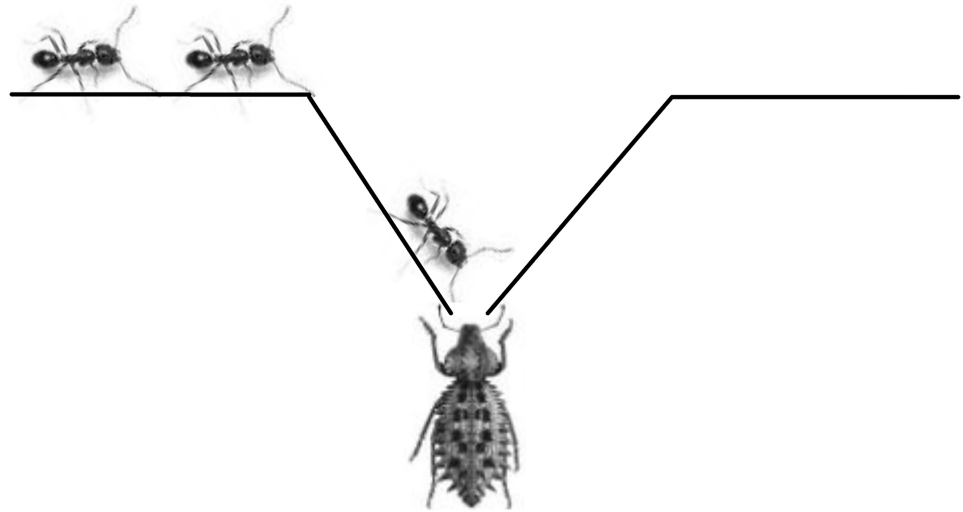


Fig 4. A hunting behavior of Ant lions.

<https://doi.org/10.1371/journal.pone.0196707.g004>

1.2 Ant Lion Optimization algorithm

Inspired by the rich nature of biological phenomena, scientists mathematically introduce novel algorithms for modeling the real world optimization problems [28]. These bio-inspired algorithms adopt stochastic procedures to overcome the limitations of traditional techniques for solving the optimization problems [29, 30]. Recently, a new bio-inspired algorithm, namely, Ant Lion Optimization (ALO) is introduced by modeling the nature chasing behavior of ant lions to catch its prey, usually ants [13].

Ant lions move in circular paths to drill cone-shaped holes in the sand to catch their prey. They hide at the bottom of the holes waiting for hunting insects as illustrated in Fig 4. The bigger holes have higher probabilities for catching a prey and are usually created by a fitter elite ant lion. Once an insect is caught, Ant lions pull it under the soil by throwing the sand towards the outer edge of the hole using its big jaw, which prevents the insect from escaping. Then, ant lions consume the prey, throw the leftovers outside the hole, rebuild the hole, and get ready for the next hunt. The fitness of ant lions and the quality of created holes are improving during each hunt [13].

The ants in ALO algorithm represent the possible random solutions for a given problem in the search space and the ant lions drill holes in the ground to catch and consume ants. The hunting ability of ant lion is encoded in the objective function and is optimized according to the interaction between ants and ant lions. There are some rules to consider when modeling the optimization problem using the nature of hunting behavior of ant lions. The ants in ALO generate random walks in the search space, which are affected by the dimensions of all ants and the ant lions created holes. The cone-shaped holes sizes are proportional to the fitness of each ant lion, i.e. the fitter ant lions can build bigger holes and hence have a higher possibility to catch a prey. The ants can be consumed by any ant lion or the elite one and the range of their random walks is decreased adaptively when the ant lion slides the ants towards the bottom of the hole. Consequently, the consuming ant lions become fitter than ants, take their positions, and rebuild the hole for improving their chance for catching other ants [13, 31–33].

In terms of clinical application of ALO in the kidney exchange space, the ants represent the possible kidney exchanges in a given pool. Maximizing the hunting ability of ant lion mimics increasing the number of matches among the pairs according to the defined optimization

function. ALO optimally explores the search space and lead to better optimized solutions and convergence rates [33].

The ants and hidden ant lions are located in the same search space and their positions during the optimization process are stored in M_{ant} and $M_{antlion}$ matrices.

$$M_{ant} = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,d} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \cdots & A_{n,d} \end{bmatrix} \tag{1}$$

$$M_{antlion} = \begin{bmatrix} AL_{1,1} & AL_{1,2} & \cdots & AL_{1,d} \\ AL_{2,1} & AL_{2,2} & \cdots & AL_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ AL_{n,1} & AL_{n,2} & \cdots & AL_{n,d} \end{bmatrix} \tag{2}$$

The n and d denote the ant's /ant lion's number and their dimension respectively. $A_{i,j}$ and $AL_{i,j}$ represent the value of j^{th} dimension of i^{th} ant/ant lion respectively. The ant's dimension encodes the number of variables in the optimization problem that affect the current problem solution, i.e. current ant. Each row in M_{ant} represents a solution to a given optimization problem according to d decision variables. The ants/ant lions are evaluated using objective function f and their evaluation values are stored in M_{OA} and M_{OAL} matrices [13, 31–33].

$$M_{OA} = \begin{bmatrix} f(A_{1,1}, A_{1,2}, \cdots, A_{1,d}) \\ f(A_{2,1}, A_{2,2}, \cdots, A_{2,d}) \\ \cdots, \cdots, \cdots, \cdots \\ f(A_{n,1}, A_{n,2}, \cdots, A_{n,d}) \end{bmatrix} \tag{3}$$

$$M_{OAL} = \begin{bmatrix} f(AL_{1,1}, AL_{1,2}, \cdots, AL_{1,d}) \\ f(AL_{2,1}, AL_{2,2}, \cdots, AL_{2,d}) \\ \cdots, \cdots, \cdots, \cdots \\ f(AL_{n,1}, AL_{n,2}, \cdots, AL_{n,d}) \end{bmatrix} \tag{4}$$

Modeling any optimization problem using ALO algorithm requires defining the following operations:

- Building ant lion traps: each ant is assumed to be caught by one ant lion. The selection of an ant lion is based on its hunting ability that is encoded in the objective function. The fitter ant lions can build bigger holes and have higher ability to catch a prey. A Roulette wheel method gives high chances for the fitter ant lions to be selected for catching ants.

- Generating ants' random walks: naturally, ants move in random paths searching for their food. Therefore, their movements in the i^{th} dimension of a d -dimensional space, X_i^n , is modelled using random cumulative sum function defined by Eq (5):

$$X_i^n = [0, \text{cumsum}(2r(n_1) - 1); \text{cumsum}(2r(n_2) - 1); \dots \text{cumsum}(2r(n_N) - 1)] \quad (5)$$

where: n denotes the current iteration, N represents the total number of iterations, cumsum returns the cumulative sum, and $r(n)$ is a random function returns value '1' if a randomly generated number is greater than '0.5' and returns '0' otherwise.

X_i^n encodes the changes (movements) that happen to the current solution (ant) by including the i^{th} decision variable (dimension) in the optimization problem.

In order to bound ant's movements inside the search space, values are normalized using a min-max normalization function defined by Eq (6):

$$X_i^n = \frac{(X_i^n - a_i) \times (d_i - c_i^n)}{(b_i^n - a_i)} + c_i \quad (6)$$

where: a_i and b_i denote the minimum and maximum random walk of the i^{th} variable, respectively, and c_i^n and d_i^n represent the minimum and maximum of the i^{th} variable at n^{th} iteration, respectively.

- Chasing ants in the holes: once the ant is trapped in the ant lion's hole, its movement is restricted by the position of the hunting ant lion (antlion_j). This is modelled by Eqs (7), and (8):

$$c_i^n = c^n + \text{antlion}_j^n \quad (7)$$

$$d_i^n = d^n + \text{antlion}_j^n \quad (8)$$

where: c^n and d^n represent the minimum and maximum of all variables at n^{th} iteration, respectively. c_i^n and d_i^n denote the minimum and maximum of all variables for ant_i, respectively.

- Sliding trapped ants: ant lions slide ants towards their created traps to prevent them from escaping. This process is mathematically modeled by iteratively decreasing the lower and upper bounds of ant's random walk, Eqs (9) and (10):

$$c^n = \frac{c^n}{I} \quad (9)$$

$$d^n = \frac{d^n}{I} \quad (10)$$

where: c^n and d^n represent the minimum and the maximum of all variables at n^{th} iteration respectively, and i is a ratio defined by Eq (11):

$$I = 10^w \frac{n}{N} \quad (11)$$

where: w denotes a dynamic parameter used to control the search exploitation level.

- Catching ants and hole re-building: The final step of hunting process is to eat the trapped ants and re-building the previously used hole. Mathematically, this process is modelled by comparing the fitness of consumed ants with their hunting ant lions. If an ant becomes fitter than its corresponding ant lions, it means that it is inside the hole and it is about to be

consumed. Accordingly, Ant lions replace their positions with the consumed ants looking for increasing their chances for the next hunt. This is defined by Eq (12):

$$\text{Antlions}_j^n = \text{Ant}_i^n \quad \text{If } f(\text{Ant}_i^n) > f(\text{Antlions}_j^n) \tag{12}$$

where: Antlions_j^n represents ant lion's position j at n^{th} iteration, and Ant_i^n represents ant's position i at n^{th} iteration, assuming maximization of the objective function.

- Maintaining the best solution: ALO adapts elitism strategy in order to preserve the best obtained solution across different iterations. The position of the fittest ant lion, i.e. elite ant lion, is used to guide the random walk of each ant as represented by Eq (13):

$$\text{Ant}_i^n = \frac{(R_A^n + R_E^n)}{2} \tag{13}$$

where: R_A^n denotes random walk around the selected ant lion, and R_E^n denotes the random walk around the elite ant lion at n^{th} iteration.

2 The proposed ALO algorithm for kidney exchanges

In our proposed method, kidney exchange is formulated as a combinatorial optimization problem. The goal is to maximize the exchange utility while satisfying the problem constraints. Feasible exchanges are only allowed (i.e. nodes involved in a given exchange can not be used in any other exchanges). Also, the length of exchange is restricted to the maximum allowed size for chains and cycles, which satisfies the hospital resources. The proposed methodology utilizes the ant lion optimization algorithm to select the optimal exchanges in a manner similar to wrapper methods in feature selection algorithms. Fig 5 gives a schematic description of the proposed ALO algorithm for kidney exchanges.

As depicted in Fig 5, the proposed method is composed of three main stages: First, constructing the KPD pool using the candidate pairs, the altruistic donors, and the compatible

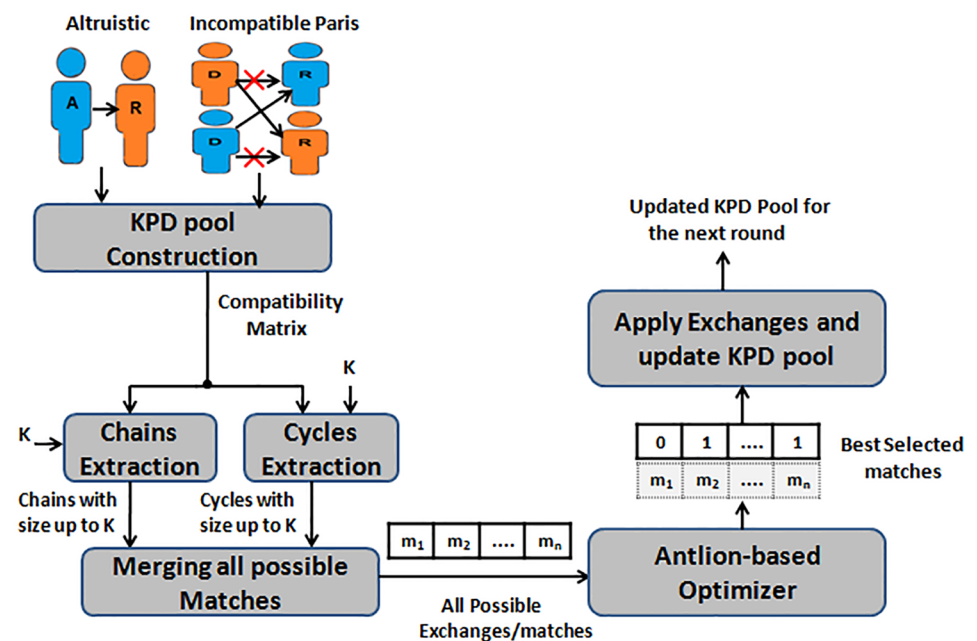


Fig 5. KPD using Binary Ant Lion Optimization algorithm.

<https://doi.org/10.1371/journal.pone.0196707.g005>

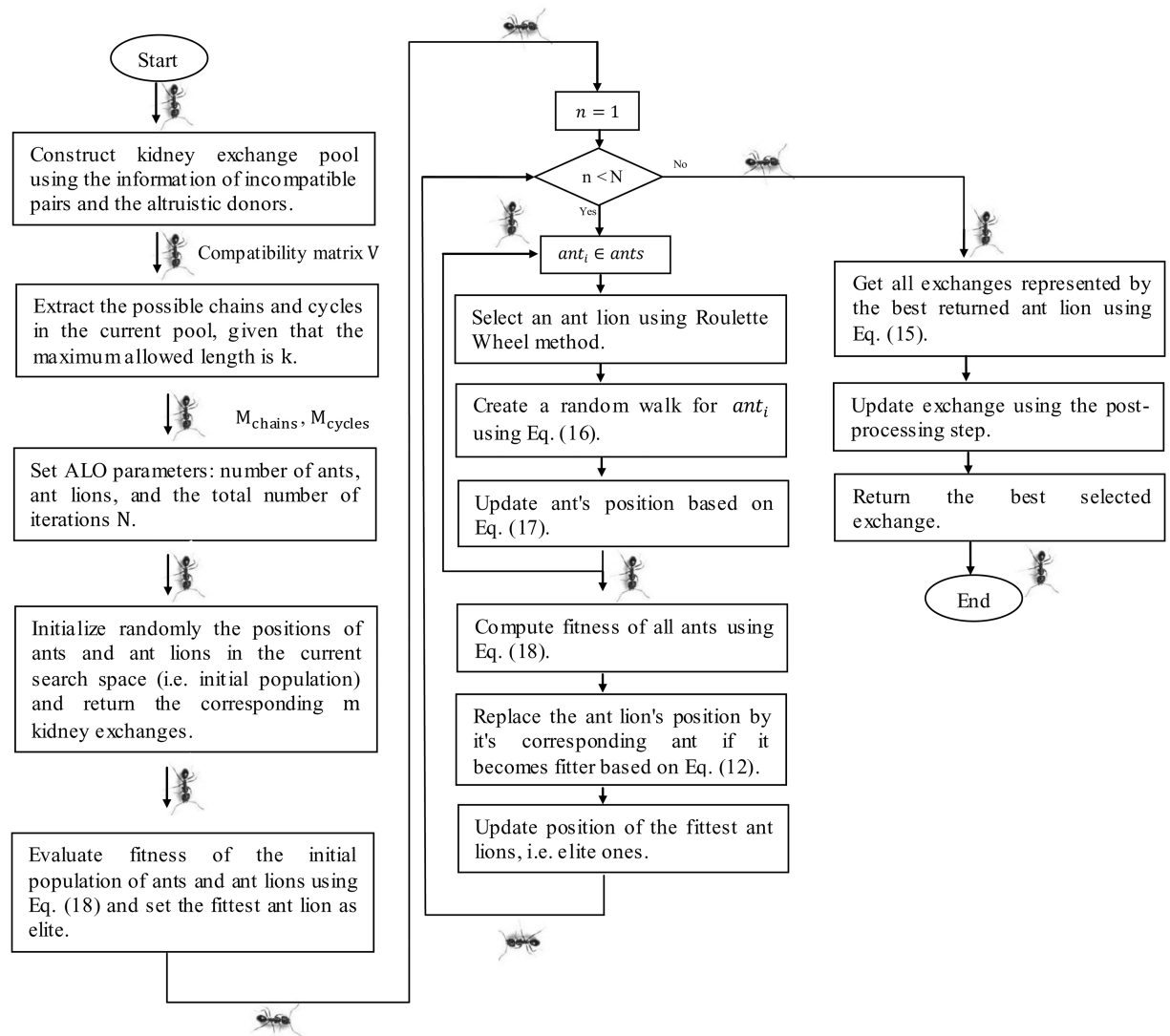


Fig 6. A flow chart of the proposed ALO algorithm for kidney exchanges.

<https://doi.org/10.1371/journal.pone.0196707.g006>

relations between them. Second, extracting all possible chains and cycles in the current KPD pool given that the maximum allowed length is k . Finally, providing all of the possible matches/ exchanges, which maximizes the number of patients receiving a kidney. After each successfully kidney transplants, KPD pool is updated and ready for a new round of kidney exchange. Fig 6 summarizes the steps used in our algorithm while the following sections explain each step in detail.

2.1 KPD pool construction

Giving a list of incompatible pairs nodes n , and altruistic donors nodes A , the compatibility matrix V , is generated, as explained by Algorithm (1). This stage is implemented in every new round or upon the arrival/ departure of new nodes to KPD pool.

Algorithm 1 KPD Pool Construction.

Input :

- The Altruistic donors set $A: \{a_1, \dots, a_u\}$.

- A set of incompatible pairs $N: \{n_1, \dots, n_z\}$ where n_i : denotes the donor/ patient pair (D_i^A, P_i^B) .
 - The compatible relations set $C: \{c_1, \dots, c_1\}$ where $c_i: (n_i, n_j, w_{i,j})$ indicates that donor of node n_i is compatible with patient of node n_j with utility value $w_{i,j}$.
- Output:** Compatibility matrix, a square matrix indicates the compatibility relations in the current pool.
- 1: Initialize $v(i, j) = 0, \forall \{i, j\}$, where: $i, j \leq u + z$.
 - 2: **for each** $c_i \in C$ **do**
 - 3: $x \leftarrow n_i$.
 - 4: $y \leftarrow n_j$.
 - 5: $v(x, y) = w_{i,j}$.
 - 6: **end for**
 - 7: Return compatibility matrix V .

2.2 Chains and cycles extraction

Chains and cycles can have different lengths. However, it is not recommended to have a large size of cycles and/or chains. Practically, exchanges are executed simultaneously to guarantee the commitment of kidney exchange process between the incompatible pairs. Therefore, chains' and cycles' length are restricted to the available hospital resources. In our proposed methodology, there is a dynamically selected parameter k denotes the maximum allowable length for chains and cycles. Algorithms (2) and (3) present the process of chains and cycles extraction respectively. The chains and cycles can be extracted sequentially or in parallel.

Algorithm 2 Chains Extraction.

Input:

- Compatibility matrix V , if all $w_{i,j} = 1, v_{i,j} \in \{0, 1\}$.
- Maximum allowed length for chains, k .

Output: All possible chains up to the maximum length.

- 1: $M_{chains} \leftarrow \Phi$.
- 2: Find the altruistic nodes in current pool, A .
- 3: **for each** $n_i \in A$ **do**
- 4: $t = 1$.
- 5: **while** $t < k$ **do**
- 6: Generate all possible compatible sequence of nodes starting from root node n_i to nodes at depth = t .
- 7: Save all resulting paths in X , A path is the sequence of nodes: $\{n_i, \dots, n_t\}$.
- 8: For each $x_i \in X$,
- 9: $M_{chains} \leftarrow \{M_{chains} \cup x_i\}$.
- 10: $t = t + 1$.
- 11: **end while**
- 12: **end for**
- 13: Return all possible chains M_{chains} .

As stated in [4], one of the limitations for stochastic-based optimization methods for kidney exchanges is the large computational running time needed for producing the optimal feasible exchanges. Extra processing is needed during evolution for restricting exchanges to the only feasible solutions. In order to decrease the overall running time, we suggested the restriction of search space to the initial extracted chains and cycles from KPD pool. The chains/cycles extraction is performed only once in our algorithm as a pre-processing stage, which reduces the overall algorithm running time.

Algorithm 3 Cycles Extraction.

Input:

- Compatibility matrix V , where if all $w_{i,j} = 1, v_{i,j} \in \{0, 1\}$.
- Maximum allowed length for cycles, k .

Output: All possible cycles up to the maximum length.

```

1:  $M_{cycles} \leftarrow \Phi$ .
2:  $t = 2$ .
3: while  $t \leq k$  do
4:   for each  $n_i \in N$ : incompatible pair nodes in the current pool do
5:     Generate all possible compatible sequence of nodes starting
       from root node  $n_i$  to nodes at depth =  $t$ .
6:     Save all resulting paths in  $X$ , A path is the sequence of nodes:
        $\{n_i, \dots, n_t\}$ .
7:     For each  $x_i \in X$ ,
8:       if  $n_i = n_t$  then
9:         if  $x_i \notin M_{cycles}$  then
10:           $M_{cycles} \leftarrow \{M_{cycles} \cup x_i\}$ 
11:        end if
12:      end if
13:    end for
14:     $t = t + 1$ .
15: end while
16: Return all possible cycles  $M_{cycles}$ .

```

2.3 Binary Ant Lion Optimization algorithm

In our proposed method, each individual is represented by a vector with a dimension equals to the total number of exchanges extracted from the given KPD pool. Usually individuals of a classical Ant Lion Optimization algorithm are represented by a continuous valued vectors. Since our representation of KPD problem is formulated as a combinatorial optimization problem, vector values are restricted to the binary range [0, 1].

Many binarization methods have been proposed to adapt continuous meta-heuristic optimization algorithms for solving binary problems [34]. Great Value Priority (GVP), is a binarization method used to map the individual's representation from a continuous space to the binary one. GVP generates a permutation sequence P based on sorting the original individual's vector and assigning to P the maximum values in order. The process is repeated for all dimensions of all individuals and P is binarized using Eq (14):

$$Y_i = \begin{cases} 1 & P_i \geq P_{i+1} \\ 0 & \text{Otherwise} \end{cases} \quad (14)$$

The mapping process of GVP reflects a priority order relation, which is suitable for our formulation of KPD problem. Inspired by this priority binary mapping, we initialized the positions of ants and ant lions using the previous mapping procedures. The ALO with a binary vector representation is called Binary Ant Lion Optimization algorithm. The solution represents the best selected matches/exchanges in the complete set of M exchanges computed by Eq (15):

$$m_i = \begin{cases} M(i) & Y_i = 1 \\ \Phi & Y_i = 0 \end{cases} \quad (15)$$

where M represents the complete set of pool's exchanges produced by the pre-processing stage.

In order to maintain the binary representation of ALO's individual, we replaced the random walk function (Eq (5)) by a mutation operation applied to the given ant lion (Eq (16))

$$\text{Antlion}(i) = \neg \text{Antlion}(i) \quad (16)$$

where Antlion represents the positions vector for the input ant lion, and i is a randomly selected dimension of the ant lion. Each ant updates its position based on a crossover operation computed by Eq (17):

$$\text{Ant}^d = \begin{cases} \text{RE}^d & r^d < 0.5 \\ \text{RA}^d & \text{Otherwise} \end{cases} \quad (17)$$

where RE and RA represent the mutated positions for the elite ant lions and a randomly selected ant lion, respectively. Ant^d represents the new ant position at dimension d , r is a vector contains randomly generated numbers drawn from a uniform distribution in the range $[0, 1]$.

The goal of the utilized Binary Ant Lion Optimization algorithm is to maximize the kidney exchange’s utility. Fitness function F is proposed to map each individual in the search space into a real-valued f , which is used to evaluate the strength of an individual in a given problem. The proposed fitness function is designed according to the following criteria:

- The Altruistic node donates a kidney without taking any benefits in return.
- Every incompatible pairs node in the resulting solution must be involved in the kidney exchange only once:
 - If the node’s donor is used more than once in the resulting exchanges, a penalty value is added.
 - If the node’s receiver is used more than once in the resulting exchanges, a penalty value is added.

The proposed fitness function is defined by Eq (18):

$$\text{maximize } \left(\sum^L \sum^U w_{i,j} \right) - E \quad (18)$$

$$E = (E_1 + E_2)\lambda \quad (19)$$

where:

- L denotes the solution length (number of exchanges i.e., chains or cycles represented by the solution);
- U denotes the exchange length (number of compatible relations represented by the cycle or chain);
- i denotes the node’s donor index;
- j denotes the node’s patient index;
- $w_{i,j}$ denotes a utility value for the compatible relation between donor i and patient j ($w_{i,j} = 1$, if all are equal).
- E_1 denotes the number of donors used more than once for donating a kidney;
- E_2 denotes the number of patients used more than once for receiving a kidney;
- λ a penalty value added to maximize error value for solutions, which contain nodes used more than once, $\lambda \gg$ pool size.

To illustrate the impact of the proposed fitness function, Fig 7 represents an example of exchanges extracted from a KPD pool. The exchanges contain two chains, one cycle of length

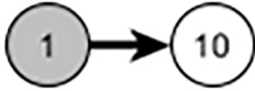


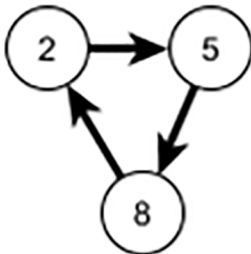
Extracted Exchanges	Antlions_1	Antlions_2
	1	0
	1	1
	0	0
	1	1

Fig 7. An example of a problem contains four exchanges.

<https://doi.org/10.1371/journal.pone.0196707.g007>

2, and one cycle of length 3. The number of ALO dimensions equals four (two chains and two cycles). Fitness with a negative value represents an infeasible exchange. To prevent ALO from producing the infeasible exchanges, a penalty value λ is added to maximize the error value E. As depicted in Fig 7, given antlion₁ represented by the vector [1101], node (1), is used twice for donating a kidney, therefore, the fitness value of antlions₁, $f = -44$, when $\lambda = 50$. If the antlion represents a feasible exchange, as given by antlion₂ which is represented by the vector [0101], the error value E equals zero and the fitness value indicates the number of transplants, $f = 5$, which denotes the global optimum solution.

Algorithm (4) presents steps of the proposed Binary Ant Lion optimization algorithm for kidney exchanges. Upon completion of Ant Lion evolution, the resulting exchanges are recommended for kidney transplantations and the KPD pool is updated for preparing the next round of exchanges.

Algorithm 4 Ant Lion Optimization algorithm for kidney Exchanges.

Input: Total number of: ants, antlions, and number of iterations N.

Output: Best selected matches/exchanges.

- 1: For each round, construct KPD pool using Algorithm (1).
- 2: Extract all possible chains, M_{chains} using Algorithm (2).
- 3: Extract all possible cycles, M_{cycles} using Algorithm (3).
- 4: $M \leftarrow \{M_{chains} \cup M_{cycles}\}$.
- 5: For each $ant_i \in Ants$, $Length(ant_i) \leftarrow |M|$.
- 6: For each $antlion_i \in Antlions$, $Length(antlion_i) \leftarrow |M|$.

```

7: Initialize populations of ants' and antlions' positions within the
   given search domain.
8: Return the exchanges  $m$ , represented by ants' and antlions' positions
   where:  $\{m \subseteq M\}$ .
9: Evaluate exchanges using the fitness function defined by Eq (18)
   and set the corresponding ants' and antlions' fitness values.
10: Find the best antlion, i.e. elite.
11: for  $n = 1, 2, \dots, N$  do
12:   For each  $ant_i \in Ants$ 
13:   Select an antlion using Roulette Wheel selection method.
14:   Create a random walk for  $ant_i$  based on Eq (16) and update it's
       position by Eq (17).
15:   Calculate the fitness of all ants.
16:   Replace an antlion with it's corresponding ant based on Eq (12).
17:   Update the elite if an antlion becomes fitter than the current
       elite.
18: end for
19: Get Exch, all exchanges  $m_i \in M$ , represented by the best returned
   ant lion based on Eq (15).
20: Update Exch, using the post-processing process defined in
   Algorithm (5).
21: Return Exch, the best selected exchanges.

```

Some runs of Ant Lion Optimization algorithm fail to produce the optimal solution and are trapped in a local optimal region. Therefore, a post-processing stage is introduced to maximize the overall exchanges' utility produced by ALO. Algorithm (5) presents the post-processing step applied to the resulting exchanges.

Algorithm 5 Candidate Exchanges Post-processing.

Input: All possible exchanges M , and candidate exchanges Exch.
Output: Updated exchanges, Exch.

```

1: For each  $m_i \in M$ ,
2: if  $m_i \notin Exch$  then
3:   if  $Nodes(Exch) \cap Nodes(m_i) = \Phi$  then
4:      $Exch \leftarrow \{Exch \cup m_i\}$ 
5:   end if
6: end if
7: For each  $m_i \in M$ 
8: For each  $m_j \in Exch$ ,
9: if  $Utility(m_i) > Utility(m_j)$  then
10:    $TmpExch \leftarrow \{Exch - m_j\}$ 
11:   if  $Nodes(TmpExch) \cap Nodes(m_i) = \Phi$  then
12:      $Exch \leftarrow \{TmpExch \cup m_i\}$ 
13:   end if
14: end if
15: Return updated exchanges, Exch.

```

In Algorithm (5), $Nodes(x)$ denote nodes represented by exchanges x , and $Utility(x)$ denotes the utility value represented by exchanges x , (if all $w_{i,j} = 1$, the utility value expresses the number of resulting transplants). The post-processing step searches for extra exchanges, which are not included in the current exchanges. Also, it can maximize the exchanges' utility or replace the existing exchanges with better ones without violating the solution's criteria. Giving a KPD pool with M exchanges, a brute force search requires the maximum number of trials equals $2^{|M|}$, in order to optimally select the best combination of exchanges $m \in M$, while satisfying the problem constrains. The post-processing step that follows the Ant Lion Optimization algorithm requires only the maximum number of trials equals $|M| + |M| * |Exch|$, where Exch, represents the candidates' exchanges produced by ALO.

The computational running time for all our experiments is explained in detail in the following section.

3 Experimental results and discussion

3.1 Data sets and experimental settings

We evaluated the performance of the proposed method using six simulated datasets generated from an updated version of Saidman generator [35]. The simulated kidney exchange data has a distribution that approximately mimics the UNOS pool as of April 15, 2013 and is computed based on the Cumulative Match Report performed by the data analysis group in this study [36]. The simulated kidney exchange pools have a number of donor-patient pairs varying from 30 to 200 nodes and the percentage of altruistic donors is chosen according to the value reported by Saidman generator.

For each experimental run, the compatibility matching rules encoded in the simulated datasets differ according to the random generation of a kidney pool. However, it is likely that the larger pool size results in existing more compatible pairs. Table 2 illustrates the compatibility information with respect to various pool sizes. To satisfy the real life hospitals constraints, we set the maximum length for cycles and chains to three, $k = 3$. The total number of extracted chains and cycles is also shown in Table 2.

As shown in Table 2: for the KPD pool that contains 200 nodes, there are 1705 compatible relations (arcs) between nodes. Also, it contains 53 chains of length two and 363 chains of length three, while the number of cycles of lengths two and three are 21 and 170 respectively. On the other hand, KPD pool of size 30 contains 56 compatible relations. Also, the pool contains a single chain of length two and 3 chains of length three and the number of cycles of lengths two and three equals 2 and 1 respectively. It is noted that when the pool size increases, the total number of compatible relations increases. As a result, the total number of extracted chains and cycles consequently increases. The extracted chains and cycles together compose all of the available exchanges in KPD pool.

Once chains and cycles are extracted, the optimization step is initiated to select the optimal number of feasible exchanges, which in turn maximizes the KPD utility. We compared our proposed method to Genetic Algorithm(GA), which is the only known stochastic-based optimization algorithm applied to the kidney exchange space [4, 12]. GA was utilized to search for the maximum number of feasible exchanges in the extracted chains and cycles using the fitness function defined by Eq (18). The parameters setting for the applied optimization algorithms is listed in Table 3. Crossover and mutation probabilities for GA were set to 0.8 and 0.2 respectively and the used selection method was Roulette Wheel. For simplicity, this method is mentioned as GA-KPD.

Table 2. Compatibility information with respect to the pool size, n.

	Compatibility Information				
	# Arcs	# Chain(2)	# Chain(3)	# Cycle(2)	# Cycle(3)
n = 30	56	1	3	2	1
n = 40	108	13	26	3	3
n = 50	123	3	11	7	9
n = 75	238	15	38	3	8
n = 100	482	15	79	10	25
n = 200	1705	53	363	21	170

<https://doi.org/10.1371/journal.pone.0196707.t002>

Table 3. Parameter settings for GA-KPD and the proposed method.

Parameters	GA-KPD	Proposed
Max number of iterations (n)	100	200
Population size	800	200
Individual Dimension	# number of cycles + # number of chains	
Search Domain	[0 1]	

<https://doi.org/10.1371/journal.pone.0196707.t003>

To compare the performance of the applied optimization methods, Ten empirical experiments were performed for the same KPD pool. Each experiment is initiated with a randomly generated population and the average results are reported. Fig 8 shows the convergence curves of the proposed method compared to GA-KPD. For each experiment, the fitness value of the elite individual is recorded per iteration.

As illustrated in Fig 8, the adopted Ant Lion Optimization algorithm in our proposed method has the fastest convergence capability towards the optimal solution. The proposed method needs less than 90 generations to converge. On the other hand, GA-KPD could not converge to the optimal solution for the maximum number of generations equals 100.

Also, we compared our results with a deterministic-based algorithm for kidney exchanges, Integer Programming (IP), which is utilized for finding the optimal number of exchanges in the extracted chains and cycles. We will call this method IP-KPD and its problem formulation is defined by the following equation:

$$\begin{aligned}
 &\text{maximize} \quad \sum_{c \in M} \sum_{i,j}^U x_c w_{i,j} \\
 &\text{s.t.} \quad x_c \in \{0, 1\}, \forall c \in M \\
 &\quad \quad \sum_{c \in M(t)} x_c \leq 1, \forall t \in \text{Pool}
 \end{aligned} \tag{20}$$

where M is the set of all extracted chains and cycles from KPD pool up to the maximum length

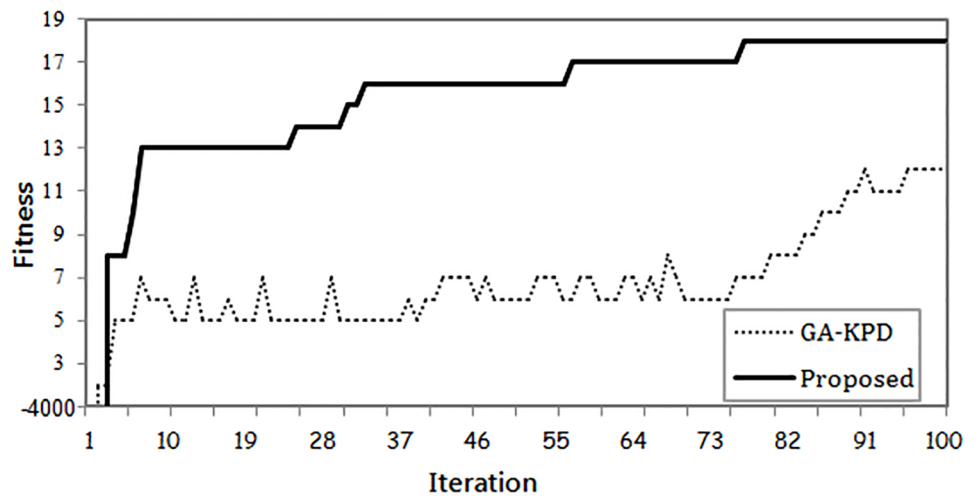


Fig 8. Elite individual's fitness per iteration.

<https://doi.org/10.1371/journal.pone.0196707.g008>

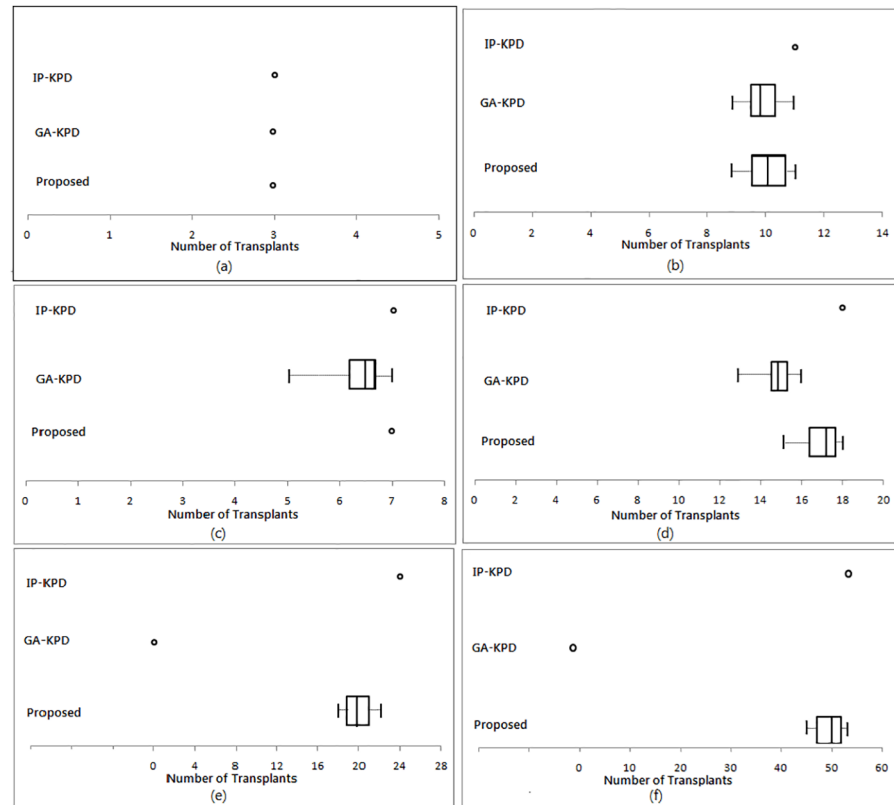


Fig 9. Transplants returned by all applied methods. (a) Pool size = 30, (b) Pool size = 40, (c) Pool size = 50, (d) Pool size = 75, (e) Pool size = 100, (f) Pool size = 200.

<https://doi.org/10.1371/journal.pone.0196707.g009>

equals three. $M(t)$ is a chain or cycle in M that contains node t , which represents an altruistic or a donor-patient pair. w_{ij} denotes a utility value for the compatible relation between donor i and patient j , and x_c is a binary vector representing whether the chain or cycle c is selected for transplant ($x_c = 1$) or not ($x_c = 0$). The existing constraints are stated as no node can be used more than one in a solution.

To compare the performance of all methods, another set of experiments is executed. All experiments are tested for different pool sizes. Experiments for GA-KPD and the proposed method are performed using the parameters specified in Table 3. Also, all experiments are repeated for ten times to account for possible statistical variations. Fig 9 shows experiments results in terms of the number of exchanges (transplants) produced by all applied methods for KPD pool sizes equal 30, 40, 50, 75, 100, and 200. The best, average, and worst number of resulting transplants are illustrated in the graph.

We can observe that when the pool size was 30, all applied methods gave the same number of transplants. When pool size is greater than 50, GA-KPD fails to converge and produce infeasible solutions, i.e. the number of resulting transplants are considered zeros. Also, we can observe that the proposed method performs better than GA-KPD. The searching strategy of ALO is more efficient than GA due to the enhanced exploration, exploitation, and convergence capabilities of ALO over GA. Moreover, it is clear that the proposed method succeeded to provide a very close performance to IP-KPD in terms of the number of resulting transplants. Table 4 shows the average number of transplants/ exchanges with the standard deviation,

Table 4. Number of transplants produced by methods(GA-KPD, Proposed, IP-KPD) for various KPD pool sizes. M denotes total number of extracted exchanges from KPD pool.

Pool Size	M	# Transplants		
		GA-KPD	Proposed	IP-KPD
n = 30	7	3 ± 0	3 ± 0	3
n = 40	45	9.8 ± 0.8	10.2 ± 0.98	11
n = 50	30	6.6 ± 0.7	7 ± 0	7
n = 75	64	15.4 ± 0.98	17.4 ± 0.92	18
n = 100	129	*	19.8 ± 1.5	24
n = 200	607	*	50.2 ± 3.16	53

<https://doi.org/10.1371/journal.pone.0196707.t004>

returned by GA-KPD and the proposed method, compared to the exact number of exchanges returned by IP-KPD.

From Table 4, we can observe that, the number of extracted exchanges (chains/ cycles) does not imply anything about the number of resulting transplants. For example, when the pool size equals 100, the total number of exchanges was 129 and the optimal number of transplants produced by IP-KPD was 24. Also, we can see that our proposed method generally provides an average number of transplants greater than GA-KPD. When the number of exchanges was greater than 64, GA-KPD fails to provide any feasible solution (indicated in the table by “*”). Additionally, the maximum number of transplants produced by our proposed method is very close to IP-KPD in almost all cases.

Moreover, we compared the optimal solutions generated by all methods, Fig 10 depicts an example for the composition of a generated solution for the pool size equaling 40. The chains and cycles of the solutions produced by IP-KPD, proposed method, and GA-KPD are shown

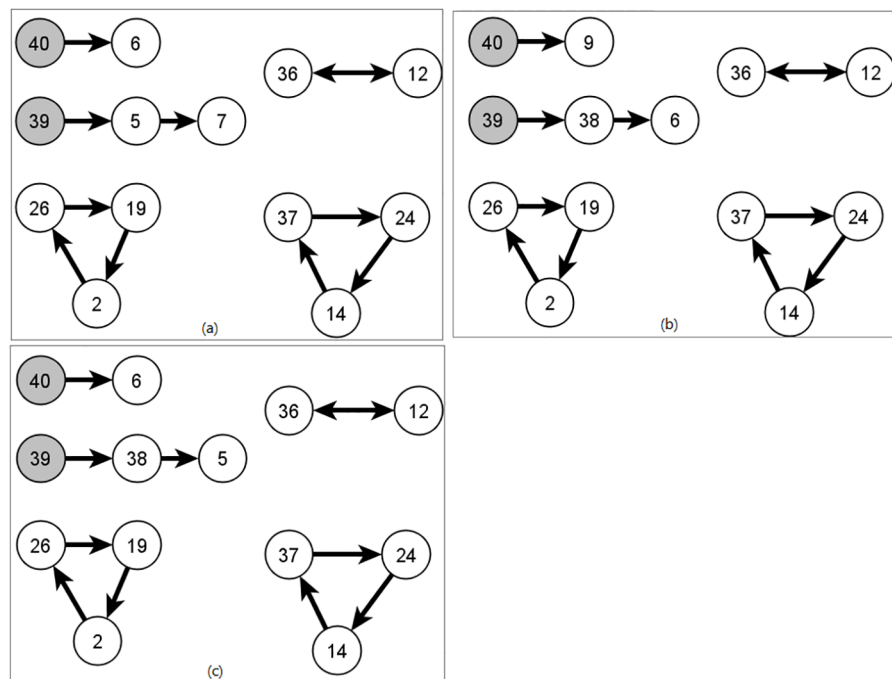


Fig 10. The composition of a resulting optimal solution. (a) IP-KPD, (b) Proposed, (c) GA-KPD.

<https://doi.org/10.1371/journal.pone.0196707.g010>

Table 5. Elapsed running time (in seconds) for all methods in various KPD pool sizes. * indicates runs giving infeasible solutions.

Pool Size	GA-KPD	Proposed	IP-KPD
n = 30	13.8	1.89	0.04
n = 40	24.6	3	0.07
n = 50	22.4	2.62	0.05
n = 75	29.3	3.87	0.18
n = 100	68.5*	4.57	0.47
n = 200	1290.5*	14.2	7.1

<https://doi.org/10.1371/journal.pone.0196707.t005>

in Fig 10(a), 10(b) and 10(c), respectively. In all figures, the altruistic node is visualized as a gray-filled circle where the non-filled circles indicate the nodes of patient-donor pairs.

The compatibility relations varies among the pool pairs and consequently the compared methods can generate different solutions with the same number of transplants. In Fig 10, the optimal solution produces 11 transplants and contains two chains with lengths two and three, and three cycles one of length two and the others of length three.

The computational running time of the produced solution is also important because patients can not wait too long for a kidney transplantation. Therefore, another important goal for our applied optimization algorithm is providing matches to patients as soon as possible. Table 5 lists the computational running time for all applied methods. Recall that the time reported for GA-KPD and our proposed method is the average running time for ten different runs. Matlab is used for implementing all of the utilized methods, all experiments ran on Windows 8 machine with Core i7 2.7-GHz Intel processors and 4-GB RAM.

It is remarkable to mention that the Ant Lion optimization algorithm converges faster than GA, which needs larger population size and additional computational steps accordingly. Despite this, GA fails to converge for large pool sizes and gives infeasible solutions. Moreover, our proposed method for kidney exchanges overcomes the computational complexity of the one proposed in [4]. In our KPD problem formulation, the process of extracting all cycles and chains from a kidney exchange pool is executed only once, which reduces the overall computational running time.

Moreover, In order to study the effect of the introduced post-processing step on the kidney exchange results, we compared our results with/without including it. Table 6 lists the max, average, and min numbers of returned transplants for each case.

For a pool size less than 100 nodes, we can observe that the post-processing step has no effect on the final solutions (i.e. ALO produces the global optimal solutions). On the other hand, for large KPD pool sizes, some ALO runs trapped in the local optimal regions of the search space. Consequently, the post-processing step plays a key role to improve the non optimal solutions given by ALO for large pool sizes.

Table 6. Number of transplants (with) and without the proposed post-processing step for various KPD pool sizes.

Pool Size	Max	Average	Min
n = 30	(3)3	(3)3	(3)3
n = 40	(11)11	(10.2)10.2	(9)9
n = 50	(7)7	(7)7	(7)7
n = 75	(18)18	(17.4)17.4	(15)15
n = 100	(22)21	(19.8)19.4	(18)18
n = 200	(53)49	(50.2)45	(45)41

<https://doi.org/10.1371/journal.pone.0196707.t006>

The most important advantage of our proposed method is the flexibility of its application in the dynamic kidney exchange environments. Recall that the first term (subgoal) of the proposed fitness function defined in Eq (18), denotes the number of given transplants (if $w_{i,j}$ all equals 1). By assigning different utility values for each compatible relation $w_{i,j}$, our method can be adopted easily to handle on-line exchanges. For example, a high utility value can prioritize the hard-to-match patients among other pool pairs and increasing their possibility for selection by the optimization algorithm. Additionally, other factors such as patient's registration time, patient's departure time, etc. could be easily included in the ALO algorithm fitness function in order to take decisions based on the multi-criteria optimized solutions for kidney exchanges.

4 Conclusion

Motivated by the increasing number of terminal stage renal patients who have the only option of kidney transplantation for saving their lives, kidney paired donation (KPD) programs are recently introduced for increasing the number of kidney exchanges on a large scale. The dynamic nature of KPD problem and the large size of its candidates' pool complicate the process of searching for the optimal number of possible exchanges. In this paper, we formulate the KPD as a combinatorial optimization problem using stochastic-based Ant Lion Optimization algorithm. The proposed novel method searches efficiently for the optimal number of exchange cycles and chains in a given KPD pool considering the constraints imposed by the limited computational and operational hospital resources. The stochastic-based ALO algorithm opens the future of taking fast and customized-based decisions for kidney exchanges considering the dynamic nature of the problem and its multi-criteria optimized solutions. Our proposed method has comparable exchange results to other competing tools including deterministic IP, and outperforms stochastic GA in terms of its speed and the quantity of resulted optimal solutions. One of our future insights is updating the current implementation of ALO algorithm-based program to include multi-criteria objective function and make it more informative and usable for patients and hospitals. Ant Lion Optimization algorithm-based program will bring more hope for caring patients with terminal-stage renal diseases and adjusting the expectancy of their life quality by taking the optimal decisions at the right time.

Author Contributions

Conceptualization: Eslam Hamouda, Sara El-Metwally, Mayada Tarek.

Data curation: Eslam Hamouda, Sara El-Metwally, Mayada Tarek.

Formal analysis: Eslam Hamouda, Sara El-Metwally, Mayada Tarek.

Methodology: Eslam Hamouda, Mayada Tarek.

Project administration: Eslam Hamouda, Sara El-Metwally, Mayada Tarek.

Software: Eslam Hamouda, Sara El-Metwally, Mayada Tarek.

Supervision: Eslam Hamouda, Sara El-Metwally, Mayada Tarek.

Validation: Eslam Hamouda, Sara El-Metwally, Mayada Tarek.

Visualization: Eslam Hamouda, Sara El-Metwally, Mayada Tarek.

Writing – original draft: Eslam Hamouda, Sara El-Metwally, Mayada Tarek.

Writing – review & editing: Eslam Hamouda, Sara El-Metwally, Mayada Tarek.

References

1. Chen C, Yijiang L, John D, Kalbfleisch D, Zhou Y, Leichtman A, et al. Graph-Based Optimization Algorithm and Software on Kidney Exchanges. *IEEE Transaction on Biomedical Engineering*. 2012; 59(7):1985–1991. <https://doi.org/10.1109/TBME.2012.2195663>
2. U.S department of Health. Organ Procurement and Transplantation Network. <https://optn.transplant.hrsa.gov>. (Accessed on May 2017)
3. Roth A, Sonmez T, Unver M. Kidney exchange. *The quarterly Journal of Economics*. 2004; 119(2):457–488. <https://doi.org/10.1162/0033553041382157>
4. Goezinne S, Bekker R, Glorie K. A genetic algorithm for kidney transplantation matching. Amsterdam: Business Analytics Master program at VU University; 2016.
5. Abraham D, Blum A, Sandholm T. Clearing algorithms for barter exchange markets: enabling nationwide kidney exchange. In: the 8th ACM conference on Electronic commerce. California; 2007. p. 295–304.
6. Alvin E, Roth E, Sonmez T, Unver U. A kidney exchange clearinghouse in New England. *American Economic Review*. 2005; 95(2):376–380. <https://doi.org/10.1257/000282805774669989>
7. Alvin E, Roth E, Sonmez T, Unver U, Francis L, Delmonico L, et al. Utilizing List Exchange and Undirected Good Samaritan Donation through ‘Chain’ Paired Kidney Exchanges. *American Journal of Transplantation*. 2006; 11(6):2694–2705.
8. Kalbfleisch D, Song P, Yijiang L. Optimization and simulation of kidney paired donation programs. University of Michigan; 2012.
9. Roth A, Sonmez T, Unver M. Efficient kidney exchange: Concidence of wants in a market with compatibility-based preferences. *American Economic Review*. 2007; 97(3):828–851. <https://doi.org/10.1257/aer.97.3.828> PMID: 29135211
10. Anderson R, et al. Finding long chains in kidney exchange using the traveling salesman problem. *Proceedings of the National Academy of Sciences*. 2015; 112(3):663–668. <https://doi.org/10.1073/pnas.1421853112>
11. Akbarpour M, Shengwu L, Gharan S. Dynamic matching market design. In: the 15 ACM conference on Economics and Computation. California; 2014.
12. Sakthivel S, Manimaran S. An optimized kidney transplantation based on genetic algorithm. *International Journal of Advanced Research*. 2013; 3(4).
13. Mirjalili S. The Ant Lion Optimizer. *Advances in Engineering Software*. 2015; 83(2015):80–98. <https://doi.org/10.1016/j.advengsoft.2015.01.010>
14. Ellison D. A Systematic Review of Kidney Paired Donation: Applying Lessons From Historic and Contemporary Case Studies to Improve the US Model. Wharton School, University of Pennsylvania; 2014.
15. Leiser D, et al. Living donor kidney paired donation transplantation: experience as a founding member center of the National Kidney Registry. *Clinical Transplantation*. 2012; 26(3):E213–E222. <https://doi.org/10.1111/j.1399-0012.2012.01606.x> PMID: 22872872
16. Rees M, Bargnesi D, Samy K, Reece L. Altruistic donation through the alliance for paired donation. *Clinical Transplantation*. 2009; 2009:235–46.
17. Ashlagi, Gilchrist D, Roth A, Rees M. Nonsimultaneous chains and dominos in kidney- paired donation-revisited. *American Journal of Transplants*. 2011; 11(5):984–994. <https://doi.org/10.1111/j.1600-6143.2011.03481.x>
18. The United Network for Organ Sharing. <https://www.unos.org>.
19. Johnson R, et al. Early experience of paired living kidney donation in the United kingdom. *Transplantation*. 2008; 86(12):1672–7. <https://doi.org/10.1097/TP.0b013e3181901a3d> PMID: 19104403
20. Malik S, Cole E. Foundations and principles of the Canadian living donor paired exchange program. *Canadian Journal of Kidney Health and Disease*. 2014;2014: 1:6. <https://doi.org/10.1186/2054-3581-1-6> PMID: 25780601
21. Ferrari P, Woodroffe C, Christiansen F. Paired kidney donations to expand the living donor pool: the Western Australian experience. *The Medical Journal of Australia*. 2009; 19(12):700–703.
22. Ferrari P, et al. Kidney paired donation: principles, protocols and programs. *Nephrol Dial Transplant*. 2015; 30(8):1276–1285. <https://doi.org/10.1093/ndt/gfu309> PMID: 25294848
23. Constantino M, et al. New insights on integer-programming models for the kidney exchange problem. *European Journal of Operational Research*. 2013; 231(1):45–63. <https://doi.org/10.1016/j.ejor.2013.05.025>
24. Segev D, et al. Kidney paired donation and optimizing the use of live donor organs. *Journal of the American Medical Association*. 2005; 239(26):1883–1890. <https://doi.org/10.1001/jama.293.15.1883>

25. Abraham D. Matching Markets: Design and Analysis. School of Computer Science, Carnegie Mellon University; 2009.
26. Gentry S, et al. The Roles of Dominos and Nonsimultaneous Chains in Kidney Paired Donation. *American Journal of Transplantation*. 2009; 9(6):1330–1336. <https://doi.org/10.1111/j.1600-6143.2009.02622.x> PMID: 19656136
27. Gentry S, et al. Expanding Kidney Paired Donation Through Participation by Compatible Pairs. *American Journal of Transplantation*. 2007; 7(10):2361–2370. <https://doi.org/10.1111/j.1600-6143.2007.01935.x> PMID: 17845570
28. Rai D, Tyagi K. Bio-Inspired Optimization Techniques: A Critical Comparative Study. *ACM SIGSOFT Software Engineering Notes*. 2013; 38(4):1–7. <https://doi.org/10.1145/2492248.2492271>
29. Blum C, J JP, Raidl R, Roli A. Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*. 2011; 11(6):4135–4151. <https://doi.org/10.1016/j.asoc.2011.02.032>
30. Boussa I, Lepagnot J, Siarry P. A survey on optimization metaheuristics. *Information Sciences*. 2013; 237(2013):82–117. <https://doi.org/10.1016/j.ins.2013.02.041>
31. Zawbaa H, Emary E, PARV B. Feature selection based on antlion optimization algorithm. In: the IEEE Third World Conference on Complex Systems (WCCS). Morocco; 2015.
32. Yogarajan G, Revathi T. A Discrete Ant Lion Optimization (DALO) Algorithm for Solving Data Gathering Tour Problem in Wireless Sensor Networks. *Middle-East Journal of Scientific Research*. 2016; 24(10):3113–3120.
33. Zawbaa H, Emary E, Grosan C. Feature Selection via Chaotic Antlion Optimization. *PLoS One*. 2016; 11(3). <https://doi.org/10.1371/journal.pone.0150652>
34. Crawford B, et al. Putting Continuous Metaheuristics to Work in Binary Search Spaces. *Complexity*. 2017; 2017:1–19. <https://doi.org/10.1155/2017/8404231>
35. Saidman S, et al. Increasing the opportunity of live kidney donation by matching for two- and three-way exchanges. *Transplantation*. 2006; 81(5):773–782. <https://doi.org/10.1097/01.tp.0000195775.77081.25> PMID: 16534482
36. Dickerson J, Procaccia A, Sandholm T. Failure-aware kidney exchange. In: the 14 ACM conference on Electronic commerce. Pennsylvania; 2013. p. 323–340.