A Proposed Genetic Algorithm Approach for the Kidney Exchange Problem

Diana Dababneh
School of Engineering
University of South Australia
South Australia, Australia
dabdy003@mymail.unisa.edu.au

Yousef Amer
School of Engineering
University of South Australia
South Australia, Australia
Yousef.Amer@unisa.edu.au

Linh Thi Truc Doan
Department of Industrial Management,
Can Tho University, Vietnam
School of Engineering
University of South Australia
South Australia, Australia
linh.doan@mymail.unisa.edu.au

Dung Thi My Tran
Department of Industrial Management
Can Tho University
Can Tho City, Vietnam
ttmdung@ctu.edu.vn

***Abstract*—Approximately 10-15% of the population worldwide is affected by Chronic Kidney Diseases (CKD). The most severe form of CKD is an end-stage renal disease (ESRD) and the treatment for ESRD is either by dialysis or kidney transplantation. Around 30% of patients with ESRD have a willing living donor in time of transplant, but their donors are incompatible due to either blood group incompatibility or human leucocyte antigen sensitization of the recipient against the donor. Kidney Exchange Program (KEP) is a policy that aims to solve this issue by matching incompatible pairs of donors and recipients with other incompatible pairs, thus increasing the chance of both pairs of receiving a kidney. Most existing research applied the exact method to solve the KEP models, but this method has some drawbacks. This research aims to propose a Genetic Algorithms (GA) approach in order to maximize the potential number of transplants in KEP. The proposed method counts and extracts all the cycles and chains prior to starting the algorithm. This step will significantly decrease the computing time needed to run the algorithm, which is one of the drawbacks of using GA. The result showed that solving the KEP by GA approach has the potential of achieving optimal results with 88.8% matching efficiency.**

Keywords— Kidney Exchange Program, Genetic Algorithms, the number of transplants.

I. Introduction

Around 10-15% of the population worldwide is affected by Chronic Kidney Diseases (CKD)[1, 2]. CKD results in reducing life expectancy for patients and impairing their quality of life. It also has significant cost implications, costing the Australian health system 4.1 billion dollars in 2012 alone [3]. The most severe form of CKD is an end-stage renal disease (ESRD), which if left untreated, can be fatal. The treatment for ESRD is either by dialysis or kidney transplantation. Around 30% of patients with ESRD have a willing living donor in time of transplant [4]; however, their donors are incompatible due to either blood group incompatibility or human leucocyte antigen (HLA) sensitization of the recipient against the donor [4-6].

Kidney Exchange Program (KEP) is a policy that aims for increasing the number of kidney transplants performed at a certain point in time by matching incompatible pairs of donors and recipients with other incompatible pairs so that the recipient of one pair receives a kidney from a matching donor in another pair, and so forth [7]. The KEP received positive adaptation around the world, including the USA, Netherlands, and Australia [4]. Currently, there have been several KEP models developed to optimise the potential number of transplants, while tackling different variants and challenges in each. Based on the literature review, most of the studies used exact methods and Integer Programming (IP) formulations for optimisation of KEP models [8-10]. Some metaheuristic methods have been developed, but in comparison to IP formulations, the research in this area is still considered at its infancy.

In this paper, a methodology of applying genetic algorithms to the KEP problem will be proposed. The genetic algorithm methodology discussed in this research aims to decrease the computing time by extracting the cycles and chains initially, followed by generating solutions that are a combination of the extracted cycles and chains to maximise the number of matched pairs.

The rest of this paper is organized as follows. A literature review regarding the overview of KEP and current research on KEP models is presented in Section II. Genetic Algorithms (GA) approach is described in Section III whereas Section IV investigates an illustrative example to validate the proposed model. Section V discusses the results of the model. Lastly, conclusions and some future directions are devoted in Section VI.

II. Literature Review

This section aims to cover an overview of KEP and current research on KEP models.

A. An overview of Kidney Exchange Program

Kidney Exchange Program (KEP) was first developed in 1986 [11]. A KEP consists of incompatible pairs, where each pair

978-1-7281-0525-3/19/$31.00©2019 IEEE

consists of a patient and an incompatible donor. In this case, a swap happens between the donor-recipient pairs, where the donor of one pair donates their kidney to a compatible recipient of another pair, and the recipient in that pair receives a kidney from the donor of the other group. Depending on the number of pairs, Kidney Exchange Programs can be two-way (two pairs), three-way (three pairs), or more. Fig. 1 illustrates a two-way kidney exchange.

The KEP model can be presented by a digraph G (V, A), where V indicates the set of vertices and A represents the arcs or edges connecting the vertices. Each vertex represents a pair of incompatible donor and recipient (vi, vj), while A contains all the arcs connecting the compatible pairs.

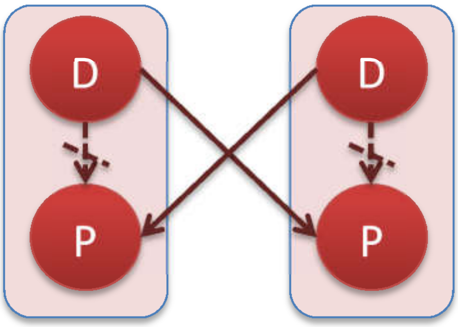

Fig. 1. Two-way Kidney Exchange [1]

Edges (e) are the arcs connected between the pairs if the donor of one pair is compatible with the recipient of another pair. Each arc (edge) is assigned a weight based on the compatibility of the donor and patient.

KEP allows for an increase in the number of living-donor kidney transplants, which has many benefits including prolonging patients' lives and allowing them an active life free from dialysis. This is done by having a pool of donor-recipient pairs (per country or geographical regional areas) and then matching compatible pairs through mathematical models to maximize the number of potential kidney transplants [4, 11].

B. *Existing research on KEP models*

Several optimisation approaches have been developed to solve the KEP problem including exact and heuristic algorithms which are presented in the following sections.

1) *Exact algorithms for KEP models*

Roth et al. [12] proposed two formulations to solve KEP and Abraham et al. [13] presented the exact methods to solve them. The two methods presented are based on integer programming (IP) formulations. They are referred to as "edge" and "cycle" formulations and they are the most common methods for KEP optimisation methods with extensive research work and models that expand on them. The challenges in these formulations are that the cycle formulation and the edge formulation have an exponential number of variables and constraints, respectively with the increase of registered pairs. This significantly increases the computational time of solving the models.

Constantino et al. [11] then presented compact formulations (extended edge (EE) and edge assignment (EA)) based on the edge and cycle models that address the issue of the exponential rise in variables and constraints. This was through bounding both the number of variables and constraints by a polynomial in the size of the problem. Hence, their model is advantageous compared with the previously mentioned models. They have also outlined problem variants in their research that are crucial to KEP and modified the model accordingly to reflect them. The aforementioned problem variants are the inclusion of altruistic donors, the non-simultaneous extended altruistic donor (NEAD) and the inclusion of compatible pairs.

Altruistic donors are those who do not belong to any pair, which leads to Non-Directed Exchanges (ND). When the altruistic donor donates his kidney, the recipient's donor goes to the next pair, and they keep rolling through the pairs till the last donor donates their kidney on the deceased donor waiting list. This results in what is described as the domino paired donation chain (DPDC) [14]. Non-simultaneous extended altruistic donor (NEAD) chain is the non-conventional case that allows non-simultaneous transplants. This chain may continue indefinitely unlike the non-directed exchanges, where numbers of pairs are limited due to the simultaneous transplant requirement [15].

The two new compact formulations that were presented by Constantino et al. [11] are bounded by a number of variants and constraints through a polynomial of the size of the problem. These two formulations are the previously mentioned EA and EE formulations. The result of the study included that the direct cycle formulation and the EE formulation perform better than the edge formulation and EA formulation. Moreover, with numbers of pairs (up to 1000) and with smaller values of k (3 or 4 in some cases), the cycle formulation performs very well (k is the maximum number of pairs in a simultaneous exchange). However, when k has larger values, the compact EE formulation proved to be more efficient. This can be considered in the case of multi-country kidney exchange programs, as usually the number of pairs is significantly less than 1000 for individual countries. For example, the Australian Kidney Exchange in Australia, as per the bi-annual reports, the average number of pairs in each round (every three months) in 2014 and 2015 was 49 and 52, respectively [16].

The study of Glorie [17] considered smart barter-exchange markets to match the supply and demand in KEP, given that the method does not only consider pairwise exchanges, but also chain and cycle exchanges. Several innovative models and techniques for matching algorithms were presented, which are tailored to include multiple objective criteria, side constraints as well as a limit on the allowed number of simultaneous exchanges. The research presented contributes towards the optimisation of the KEP model, and also covers several topics such as transplantation across the blood type barrier and multi-centre coordination of unspecified living kidney donation, and where it should be used in DPDC and NEAD.

Manlove and O'Malley [18] studied the kidney-paired donation (KPD) algorithms and presented computational results relative to the UK's definition of optimality. It is based on expanding the direct cycle formulation presented by Roth et al. [12] to incorporate the UK's National Living Donor Kidney Sharing Schemes (NLDKSS) optimality criteria for kidney exchange selection. The criteria are as follows: a set of exchanges is optimal if: the number of effective 2-cycles is maximised; the exchange has the maximum size subject to the first criterion; the number of 3-cycles is minimized subject to the first two criteria; the number of back-arcs in the 3-cycles is maximized in the exchange and the overall weight of cycles is maximised given that all of the previous criteria apply. The aim of the criteria is to ensure maximising the number of transplants while still ensuring that if a 3-cycle exchange fails, embedded 2-cycles can proceed with the transplants.

Alvelos et al. [8] proposed a new integer-programming model based on the cycle formulation that maximizes the "expected" number of transplants in the case of potential failure that results from either a pair dropping out from the exchange, or failure due to incompatibilities that arise when more tests were requested prior to the transplant. However, this method was limited only to cycles of length up to 4, and it is not always the case that there is an equal failure probability for the two types of failures mentioned. Computational results of the proposed algorithm were studied and calculated using the programming solver – CPLEX.

2) Genetic Algorithm for solving KEP

Genetic algorithm (GA) has been used to provide solutions to optimisation problems through a trade-off between local randomised search pathways and global exploration of solutions [19, 20]. Sakthivel & Manimaran [21] investigated the application of GA to the kidney exchange problem and provided a comparison with the graph-based optimisation method. Their findings included that the GA approach is faster than the graph-based method and that it might have a higher match yield. The research also discussed how the utilization of the paired donors-patients and altruistic donors is effective in increasing the quality and quantity of kidney transplants. However, the details of the methodology in applying the GA to the kidney exchange problem was not outlined and discussed.

Goezinne, Bekker & Glorie [22] investigated the application of GA in the static situation, where a solution is generated for couples that are already in the program at a certain point in time. A dynamic situation is when the situation considers the potential changes in the future such as new pairs entering or exiting the pool, or failures of arcs after the match occurs. The study considered three different methods depending on keeping infeasible solutions in the algorithm or not, with the goal of reaching an optimal feasible solution. In their study, GA was compared with the exact methods, and their method generated an optimal solution in the case of having less than or equal to 40 incompatible pairs. However, their run time was high when comparing it with the exact methods because in each generation, cycles and chains were counted and the feasibility of the solutions was assessed. A suggestion for future research was to have the step of counting cycles and chains initially prior to starting the algorithm, and formulate the solution based on cycles and chains instead.

Hamouda & El-Metwally [23] studied and applied the stochastic-based Ant Lion Optimization (ALO) to solve the kidney exchange problem. ALO is a metaheuristic algorithm that is inspired by the behaviour of ant lions to catch their preys or ants [24]. Their study showed that the metaheuristic approach can achieve results similar or very close to the IP formulation. The ALO method has the potential to consider other factors, such as hard-to-match patients, which can improve the outcomes of the match. They have also developed software using MATLAB for the ALO algorithm to simulate and match donors and patients, including altruistic donors in the space.

III. Genetic Algorithm Methodology

Genetic algorithm (GA) is a very popular form of Evolutionary Algorithms (EAs). Evolutionary Algorithms are metaheuristic optimisation algorithms that fall under the neo-Darweinian paradigm [25].

Typically, GA consists of the following steps: initialize population, evaluate fitness, select parents, recombination (cross-over) and mutation (due to error or external environmental factors) to produce the next generation (as seen in Fig. 2). The steps of selecting parents, cross-over and mutation are the genetic operators of the GA method. The process is repeated to evolve the population of solutions to better solutions and find an optimal solution.

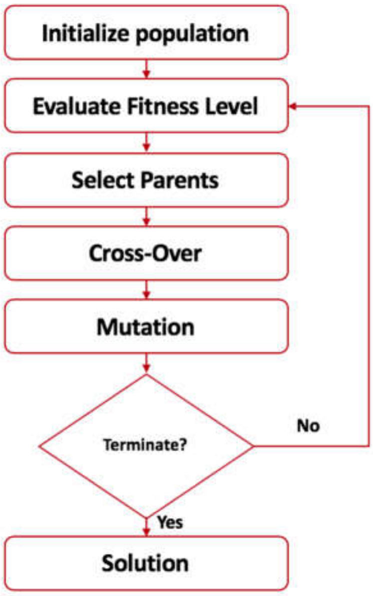

Fig. 2. Genetic Algorithm Methodology [20]

1) Initialize population: This step includes generating random initial solutions to a problem. These initial solutions form the initial "population". Each solution is represented by a chromosome, and the population consists of "n" chromosomes (n: number of solutions). Each individual chromosome (solution) is denoted by x_i [25].

The population is represented by $P(t) = x_1 + x_2 + \cdots + x_n$, where n is the size of the population.

A chromosome is a coded solution that consists of Genes, and each Gene consists of Alleles. Genes are usually encoded as binary strings but can also be encoded as numbers or letters depending on the problem. An Allele is the smallest unit of information in the gene.

2) Evaluate Fitness Level: In biology, fitness is the ability of an individual to survive and reproduce [25]. In the case of GA, it refers to the value of the objective function given the solution found. Namely, this is described as $f(\boldsymbol{x_i})$ for each x_i, where $f(\boldsymbol{x})$ is the objective function and x_i is the ith-solution. $f(\boldsymbol{x_i})$ is defined based on the problem that we need to solve. For example, it can be a minimization or a maximization formula.

3) Select Parents: Selection of parents for the regeneration of the next population is determined by two facets, as follows:

- Fitness of the solution in the solution space, which is $f(\boldsymbol{x_i})$ for each x_i.
- The probability of the selection of each parent, where solutions with higher fitness values are more likely to be selected as parents. That is solutions with higher $f(\boldsymbol{x_i})$ (or lower in case of a minimization function) are more likely to be chosen as parents.

The selection of the parents can be performed in a deterministic or a stochastic matter, or a mix of both [25]. An example of a stochastic selection method is the roulette-wheel selection [26] while an example of a mixed selection method is the ranking selection [27] and tournament selection [20].

4) Cross-Over : Once the parent solutions are selected, a cross-over or recombination occurs to produce the next offspring (new generation of solutions). This is done by exchanging parts of the parents' chromosomes. The new generation is different from either of the parents, although they have inherited parent traits

5) Mutation: Mutation in the new generation allows to keep a degree of variation to allow for a better chance at reaching the optimal solution. It occurs by selecting a random position in a chromosome, and then altering it by either changing it or by replacing it with another gene or information. It can occur on one parent to generate new offspring.

6) Insert the newly generated population: Once the cross-over and mutation steps are completed, a new generation of solutions that is different from the parent generation is produced. This new generation has inherited some aspects from the parents who have high fitness, while still have their own characteristics through cross-over of genetics and a degree of mutation to introduce change in the new offspring.

7) Repeat: Steps 3-6 (Evaluate Fitness Level, Selection, Cross-Over, and Mutation) are repeated for a number of generations until the termination criteria are met. Each new offspring will theoretically have a higher fitness level from the previous generation in order to optimise the objective function.

8) Terminate: Once the termination criteria are met, the GA can terminate and the last generation will be the best solution found to the problem, which is potentially the optimal solution to the problem. The termination criteria can be a maximum number of generations, the fitness level staying constant for a pre-determined number of generations, or any other selected criteria depending on the problem.

IV. An Illustrative Example

This section aims to demonstrate the practicability of the proposed GA method for solving the kidney exchange problem. In Section IV.1, the problem formulation including parameters and KEP solution chromosome characteristics is illustrated. Section IV.2 presents KEP using GA.

1) Problem Formulation

The input parameters needed in this model:

n: number of pairs of donor-patients
a: number of altruistic donors
k: maximum allowable cycle and chain length
A: compatibility matrix

In this illustration, there is an assumption of n = 8 pairs, a = 1 altruistic donor, k = 3.

The compatibility matrix A for this example is shown in Table I. The weight of the compatibility here is assumed to be 0 or 1. An edge is connected from node i to node j if the donor in pair i is compatible with the patient in node j. The weight in that case is 1. Otherwise, the weight is zero. It can be noted that all the weights from the nodes 1-8 to node 9 are zeroes, due to the fact that an altruistic donor only donates but does not receive a kidney in return.

In Section II, KEP can be represented through a Digraph G (V, E). The digraph for the example was constructed in MATLAB and shown in Fig. 3. Nodes 1-8 are the pairs of patients-donors while node 9 is the altruistic donor.

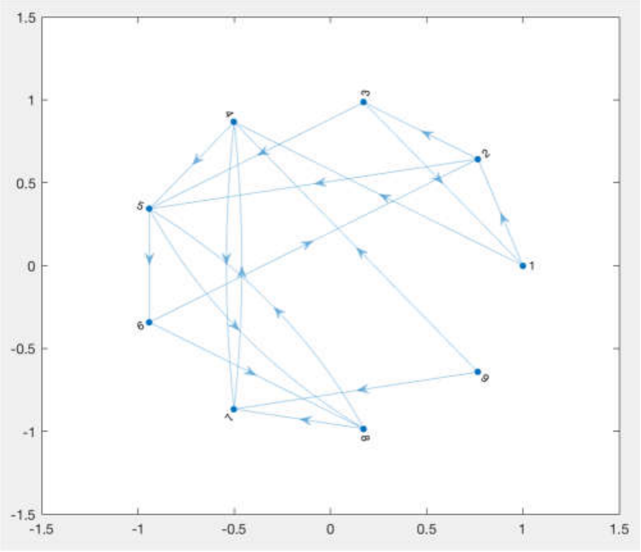

Fig. 3. Digraph for example problem

Find all cycles and chains of maximum length k

In the example above, we need to find out all the cycles and chains such that k ≤ 3.

a) Cycles of length 2 (k = 2)

- Nodes (4,7): 4→7, 7→4
- Nodes (5,8): 5→8, 8→5

b) Cycles of length 3 (k = 3)

- Nodes (1,2,3): 1→2, 2→3, 3→1
- Nodes (2,5,6): 2→5, 5→6, 6→2
- Nodes (5,6,8): 5→6, 6→8, 8→5

c) Chains of length 2 (k = 2)

Chains are started by altruistic donors. In this example, node 9 is an altruistic donor, and 3 chains of length 2 are found.

- Nodes (9,7,4): 9→7, 7→4

- Nodes (9,4,7): 9→4, 4→7
- Nodes (9,4,5): 9→4, 4→5

d) Chains of length 3 (k = 3)

- Nodes (9,4,5,6): 9→4, 4→5, 5→6
- Nodes (9,4,5,8): 9→4, 4→5, 5→8

KEP – Chromosome

The suggested chromosome in this project for the GA for KEP is as follows:

Gene 1: Cycles of length k=2
Allele 1: 4→7, 7→4
Allele 2: 5→8, 8→5

Gene 2: Cycles of length k=3
Allele 3: 1→2, 2→3, 3→1
Allele 4: 2→5, 5→6, 6→2
Allele 5: 5→6, 6→8, 8→5

Gene 3: Chains of length k=2
Allele 6: 9→7, 7→4
Allele 7: 9→4, 4→7
Allele 8: 9→4, 4→5

Gene 4: Chains of length k=3
Allele 9: 9→4, 4→5, 5→6
Allele 10: 9→4, 4→5, 5→8

TABLE I. COMPATIBILITY MATRIX

Node	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9 (atruistic)*
1	0	1	0	1	0	0	0	0	0
2	0	0	1	0	1	0	0	0	0
3	1	0	0	0	1	0	0	0	0
4	0	0	0	0	1	0	1	0	0
5	0	0	0	0	0	1	0	1	0
6	0	1	0	0	0	0	0	1	0
7	0	0	0	1	0	0	0	0	0
8	0	0	0	0	1	0	1	0	0
9	0	0	0	1	0	0	1	0	0

The binary variables indicate whether a cycle/chain was chosen for the solution or not, with 1 indicating that it was chosen, and 0 indicating that it was not chosen.

Run the Genetic Algorithm

After importing the data for the problem, finding all possible cycles and chains, and encoding the solutions chromosomes, the genetic algorithm is run to find the optimal solution for the problem. In the following section, the genetic operators will be discussed through the example.

TABLE II. POPULATION GENERATED USING MATLAB

Chromosome	*Encoding* Cycles and Chains	*Decoding* Nodes in each cycle and chain chosen	*Feasible* (YES/NO)
X1	[1,0,0,0,1,0,0,0,0,0]	(4,7), (5,6,8)	Yes
X2	[1,1,1,1,1,1,1,0,0,0]	(4,7), (5,8), (1,2,3), (2,5,6), (5,6,8), (9,7,4)	No
X3	[1,1,0,0,1,0,0,0,1,1]	(4,7), (5,8), (5,6,8), (9,4,5,6), (9,4,5,8)	No
X4	[1,0,1,0,0,0,0,0,0,0]	(4,7), (1,2,3)	Yes
X5	[1,0,1,1,0,1,0,0,0,0]	(4,7), (1,2,3), (2,5,6), (9,7,4)	No
X6	[1,0,0,0,1,1,0,0,0,0]	(4,7), (5,6,8), (9,7,4)	No
X7	[0,0,0,1,1,1,1,0,1,1]	(2,5,6), (5,6,8), (9,7,4), (9,4,5,6), (9,4,5,8)	No
X8	[1,1,0,1,1,0,0,0,1,0]	(4,7), (5,8), (2,5,6), (5,6,8), (9,4,5,6)	No
X9	[0,1,0,0,0,0,1,0,0,0]	(5,8), (9,4,7)	Yes
X10	[0,1,0,1,0,0,1,0,1,1]	(5,8), (2,5,6), (9,4,5,6), (9,4,5,8)	No

2) KEP – GA Application

Initialize population

The initial population consists of a pre-defined number of chromosomes, each representing a potential solution to the problem. In each chromosome, there is a 0.5 probability that a cycle or chain is equal to 1, and a 0.5 chance that it equals 0.Using MATLAB, an example of 10 randomly generated chromosomes were obtained as seen in Table II.

It can be noted that some solutions generated are infeasible due to the fact that some nodes are selected in more than one cycle. For example, in X3, node 4 is selected in 3 cycles, node 5 in 4 cycles, node 6 in 2 cycles, and node 8 in 3 cycles.

Evaluate Fitness Level

For each solution (Xi) in the population, the fitness is evaluated to measure how good the solution is. In the KEP problem, the aim is to find how many transplants this solution achieves.

$$f(X) = \sum_{c\in C} w_c - P$$

Where:

$f(X)$: *Fitness value of solution X*

w_c: *weight ofcycle or chain c in chromosome X*

P: *penalty value for infeasible solutions*

TABLE III. FITNESS VALUE FOR EACH CHROMOSOME

Chromosome	*Encoding* Cycles and Chains	*Decoding* Nodes in each cycle and chain chosen	$\boldsymbol{f(X)}$
X1	[1,0,0,0,1,0,0,0,0,0]	(4,7), (5,6,8)	5
X2	[1,1,1,1,1,1,1,0,0,0]	(4,7), (5,8), (1,2,3), (2,5,6), (5,6,8), (9,7,4)	15-(2+2+3+2+2+2) =2
X3	[1,1,0,0,1,0,0,0,1,1]	(4,7), (5,8), (5,6,8), (9,4,5,6), (9,4,5,8)	13-(3+4+3+2+2) =-1
X4	[1,0,1,0,0,0,0,0,0,0]	(4,7), (1,2,3)	5
X5	[1,0,1,1,0,1,0,0,0,0]	(4,7), (1,2,3), (2,5,6), (9,7,4)	10-(2+2+2) = 4
X6	[1,0,0,0,1,1,0,0,0,0]	(4,7), (5,6,8), (9,7,4)	7-(2+2) = 3
X7	[0,0,0,1,1,1,1,0,1,1]	(2,5,6), (5,6,8), (9,7,4), (9,4,5,6), (9,4,5,8)	13-(4+3+2+3+3) =-2
X8	[1,1,0,1,1,0,0,0,1,0]	(4,7), (5,8), (2,5,6), (5,6,8), (9,4,5,6)	13-(2+4+2+3) =2
X9	[0,1,0,0,0,0,1,0,0,0]	(5,8), (9,4,7)	5
X10	[0,1,0,1,0,0,1,0,1,1]	(5,8), (2,5,6), (9,4,5,6), (9,4,5,8)	11-(4+2+2+2+2) =-1

The fitness function (as seen in Table III) calculates the weighted cycles and chains in each solution to find the number of transplants. In each cycle of k=2, the weight is 2. In each cycle of k=3, the weight is 3. In each chain of k=2, the weight is 2, and for each chain of k=3, the weight is 3.

A penalty P is applied to decrease the fitness value of the infeasible solutions so that they are unlikely to be chosen as parents for the next generation. The penalty P should be high to ensure that infeasible solutions are not favoured for selection as parents for the next generation. In this methodology, $P = \sum r_{\acute{n}} \times \acute{n}$

Where:

$\acute{n}$: *nodes that are used more than once*
$r_{\acute{n}}$: *the number of times* $\acute{n}$ *was repeated*

Select Parents

In the proposed algorithm, the selection of parents will be based on the roulette-wheel. The successful probability of a parent being selected is higher when they have a higher fitness value.

$$P_i = \frac{f(x_i)}{\sum_{i=1}^{Np} f(x_i)}, i = 1,2,\ldots,N$$

A run is carried out by spinning the roulette wheel, where individuals with higher fitness ranking have a higher chance of being selected.

Cross-Over

After selecting the parents, a cross-over at one random point will occur to generate the next generation. A combination of two vectors (x_1 andx_2) will generate two offspring as follows:

$$\acute{x}_1 = \lambda x_1 + (1-\lambda)x_2$$

$$\acute{x}_2 = \lambda x_2 + (1-\lambda)x_1$$

where $\acute{x}_1$ and $\acute{x}_2$ are the new offspring, and $0 < \lambda < 1$ [28].

In this demonstration, the cross-over occurs between Allele 3 and 4 for X1 and X4, and the offspring is X11 and X12 (as seen in Table IV).

TABLE IV. CROSS-OVER AND NEW OFFSPRING

Chromosome	*Parents*	*Decoding*	$f(X)$
X1	[1,0,0,0,1,0,0,0,0,0]	(4,7), (5,6,8)	5
X4	[1,0,1,0,0,0,0,0,0,0]	(4,7), (1,2,3)	5
	Offspring		
X11-Gen2	[1,0,0,0,0,0,0,0,0,0]	(4,7)	2
X12-Gen2	[1,0,1,0,1,0,0,0,0,0}	(4,7), (1,2,3), (5,6,8)	8

TABLE V. MUTATION OF CHROMOSOMES

Parent	*Encoding*	*After mutation*	$\boldsymbol{f(X)}$
X9	[0,1,**0**,0,0,0,1,0,0,0]	(5,8), (9,4,7)	5
	After mutation		
X13-Gen2	[0,1,**1**,0,0,0,1,0,0,0]	(5,8), (1,2,3), (9,4,7)	8

Mutation

The mutation method for this algorithm is the uniform bit-flip. It changes each allele of the chromosome (switches the 0 to 1, or vice versa) with a probability of $p = \frac{1}{L}$. In this case, the chromosome length is 10 because it consists of 10 alleles, and thus $L = 10$. Accordingly, there is a probability of 0.1 that an allele in the chromosome would change its binary value. The results are shown in Table V.

Insert the newly generated population and repeat

The new generation is inserted into the population by the fitness-based replacement method. The new population is $\lambda + \mu$, where μ is the number of parents and λ is the number of children [29]. The new offspring will replace the previous chromosomes with lower fitness. The new population is shown in Table VI.

The algorithm then repeats the steps of evaluating the new population, selecting parents, cross-over, mutation and inserting a new population.

TABLE VI. POPULATION - GENERATION 2

Chromosome	*Encoding* Cycles and Chains	*Decoding* Nodes in each cycle and chain chosen	$\boldsymbol{f(X)}$
X1	[1,0,0,0,1,0,0,0,0,0]	(4,7), (5,6,8)	5
X2	[1,1,1,1,1,1,1,0,0,0]	(4,7), (5,8), (1,2,3), (2,5,6), (5,6,8), (9,7,4)	2
X4	[1,0,1,0,0,0,0,0,0,0]	(4,7), (1,2,3)	5
X5	[1,0,1,1,0,1,0,0,0,0]	(4,7), (1,2,3), (2,5,6), (9,7,4)	4
X6	[1,0,0,0,1,1,0,0,0,0]	(4,7), (5,6,8), (9,7,4)	3
X8	[1,1,0,1,1,0,0,0,1,0]	(4,7), (5,8),(2,5,6), (5,6,8), (9,4,5,6)	2
X9	[0,1,0,0,0,0,1,0,0,0]	(5,8), (9,4,7)	5
X11-Gen2	[1,0,0,0,0,0,0,0,0,0]	(4,7)	2
X12-Gen2	[1,0,1,0,1,0,0,0,0,0}	(4,7), (1,2,3), (5,6,8)	8
X13-Gen2	[0,1,1,0,0,0,1,0,0,0]	(5,8), (1,2,3), (9,4,7)	8

Terminate

The algorithm terminates after a pre-defined number of generations, or after the solution stays constant for a number of generations.

V. RESULT

There were 8 pairs of patients-donors and 1 altruistic donor which were used in the illustrative example. The solution chromosome was encoded by a binary vector that represents all the potential cycles and chains, with the value 1 representing a cycle/chain that is chosen, and 0 representing a non-chosen cycle/chain. The step of initializing population has shown that the solutions obtained are not all feasible, thus a penalty was added to the fitness function to un-favour those infeasible solutions and favour feasible solutions in the parent selection step.

Using the proposed approach, the best solutions found were X12-Gen2 and X13-Gen 2. These solutions allow for 8 out of 9 nodes to be matched. The matching efficiency is 88.8% (8/9), which is the optimal solution for this problem. Even though it is a manual demonstration of how to encode the KEP problem and optimise the problem, the method of applying genetic algorithms to solve the KEP proved to have the potential of achieving optimal results.

In comparison to the GA approach developed by [22], the proposed method counts and extracts all the cycles and chains prior to starting the algorithm. This step will significantly decrease the computing time (18%) needed to run the algorithm, which is one of the drawbacks of using GA. In the previous studies, the solution chromosome consisted of a binary vector representing all the edges between the nodes, which can be a huge binary vector in large data sets.

VI. Conclusions and Future Works

A genetic algorithm was developed to solve the KEP which can try to maximize the potential number of transplants. In addition, this approach proved to have the potential of achieving optimal results. Compared to the previous studies, the proposed approach counts and extracts all the possible cycles and chains before applying the algorithm. This leads to the remarkable reduction of the runtime.

A further study would consider multi-objective functions for KEP including the maximum of the number of transplants in the case of arc and node failures and the minimum of risks to the matching methods. In addition, studying the computing time needed to extract the cycles and chains and the runtime of the algorithms and comparing it with the traditional methods would be a potential direction.

Acknowledgment

The authors would like to thank University of South Australia for the support.

References

[1] A. Levin *et al.*, "Global kidney health 2017 and beyond: a roadmap for closing gaps in care, research, and policy," *The Lancet,* 2017.

[2] K. Lundwall, S. H. Jacobson, G. Jörneskog, and J. Spaak, "Treating endothelial dysfunction with vitamin D in chronic kidney disease: a meta-analysis," *BMC nephrology,* vol. 19, no. 1, p. 247, 2018.

[3] G. B. d. Silva Junior, J. G. R. d. Oliveira, M. R. B. d. Oliveira, L. J. E. d. S. Vieira, and E. R. Dias, "Global costs attributed to chronic kidney disease: a systematic review," *Revista da Associação Médica Brasileira,* vol. 64, no. 12, pp. 1108-1116, 2018.

[4] R. D. Allen, H. C. Pleass, C. Woodroffe, P. A. Clayton, and P. Ferrari, "Challenges of kidney paired donation transplants involving multiple donor and recipient surgeons across Australia," *ANZ journal of surgery,* vol. 88, no. 3, pp. 167-171, 2018.

[5] H. Kim, O. Kwon, and C. Kang, "The Utilization and Advantages of an Exchange Donor Program in Living Donor Renal Transplantation: A Single-Center Experience," in *Transplantation proceedings*, 2012, vol. 44, no. 1: Elsevier, pp. 14-16.

[6] M. Karpinski, G. Knoll, A. Cohn, R. Yang, A. Garg, and L. Storsley, "The impact of accepting living kidney donors with mild hypertension or proteinuria on transplantation rates," *American journal of kidney diseases,* vol. 47, no. 2, pp. 317-323, 2006.

[7] D. Chisca, M. Lombardi, M. Milano, and B. O'Sullivan, "From offline to online kidney exchange optimization," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2018: IEEE, pp. 587-591.

[8] F. Alvelos, X. Klimentova, A. Rais, and A. Viana, "Maximizing expected number of transplants in kidney exchange programs," *Electronic Notes in Discrete Mathematics,* vol. 52, pp. 269-276, 2016.

[9] F. Alvelos, X. Klimentova, and A. Viana, "Maximizing the expected number of transplants in kidney exchange programs with branch-and-price," *Annals of Operations Research,* vol. 272, no. 1-2, pp. 429-444, 2019.

[10] V. Bartier, B. Smeulders, Y. Crama, and F. C. Spieksma, "Recourse in Kidney Exchange Programs," 2019.

[11] M. Constantino, X. Klimentova, A. Viana, and A. Rais, "New insights on integer-programming models for the kidney exchange problem," *European Journal of Operational Research,* vol. 231, no. 1, pp. 57-68, 2013.

[12] A. E. Roth, T. Sönmez, and M. U. Ünver, "Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences," *American Economic Review,* vol. 97, no. 3, pp. 828-851, 2007.

[13] D. J. Abraham, A. Blum, and T. Sandholm, "Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges," in *Proceedings of the 8th ACM conference on Electronic commerce*, 2007: ACM, pp. 295-304.

[14] D. L. Segev, S. E. Gentry, D. S. Warren, B. Reeb, and R. A. Montgomery, "Kidney paired donation and optimizing the use of live donor organs," *Jama,* vol. 293, no. 15, pp. 1883-1890, 2005.

[15] I. Ashlagi, D. S. Gilchrist, A. E. Roth, and M. A. Rees, "Nonsimultaneous chains and dominos in kidney ‐ paired donation—Revisited," *American Journal of transplantation,* vol. 11, no. 5, pp. 984-994, 2011.

[16] M. P. Sypek *et al.*, "Optimizing outcomes in pediatric renal transplantation through the Australian paired kidney exchange program," *American Journal of Transplantation,* vol. 17, no. 2, pp. 534-541, 2017.

[17] K. Glorie, *Clearing barter exchange markets: Kidney exchange and beyond* (no. EPS-2014-329-LIS). 2014.

[18] D. F. Manlove and G. O'Malley, "Paired and altruistic kidney donation in the UK: Algorithms and experimentation," in *International Symposium on Experimental Algorithms*, 2012: Springer, pp. 271-282.

[19] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, "Metaheuristic algorithms in modeling and optimization," in *Metaheuristic applications in structures and infrastructures*: Elsevier, 2013, pp. 1-24.

[20] S. Kumar, S. Jain, and H. Sharma, "Genetic Algorithms," *Advances in Swarm Intelligence for Optimizing Problems in Computer Science,* pp. 27-52, 2018.

[21] S. Sakthivel and S. Manimaran, "An optimized kidney transplantation based on genetic algorithm," *International Journal of Advanced Research,* vol. 3, no. 4, 2013.

[22] S. Goezinne, R. Bekker, and K. Glorie, "A genetic algorithm for kidney transplantation matching," *Amsterdam: Business Analytics Master program at VU University,* 2016.

[23] E. Hamouda, S. El-Metwally, and M. Tarek, "Ant Lion Optimization algorithm for kidney exchanges," *PloS one,* vol. 13, no. 5, p. e0196707, 2018.

[24] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software,* vol. 83, pp. 80-98, 2015.

[25] K.-L. Du and M. Swamy, "Search and optimization by metaheuristics," *Techniques and Algorithms Inspired by Nature; Birkhauser: Basel, Switzerland,* 2016.

[26] J. Holland, "Adaptation in artificial and natural systems," *Ann Arbor: The University of Michigan Press,* 1975.

[27] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of genetic algorithms*, vol. 1: Elsevier, 1991, pp. 69-93.

[28] H.-P. Schwefel, *Numerical optimization of computer models*. John Wiley & Sons, Inc., 1981.

[29] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. Springer, 2003.