

## Binary Particle Swarm Optimization Algorithm for Kidney Exchanges Acceleration using Parallel MATLAB

Wael M. F. Abdel-Rehim

Faculty of Computers and Information, Suez University, Egypt

[wael\\_fawaz@hotmail.com](mailto:wael_fawaz@hotmail.com)

---

### Abstract

In this paper, we implement a new method binary Particle Swarm Optimization (PSO) for solving the kidney exchange problem, which will improve the future decisions of kidney exchange programs. Because using a kidney exchange, we can help incompatible patient-donor couples to swap donors to receive a compatible kidney. Kidney paired donation programs provide an innovative approach for increasing the number of available kidneys. Further, we implementing binary particle swarm optimization in parallel with MATLAB with one, two, three and four threads and from the computations point of view, the authors compare the performance to reduce the running time for kidney exchange to match patients as fast as possible to help clinicians. Moreover, implementing binary particle swarm optimization in solving the kidney exchange problem is an effective method. The obtained results indicate that binary PSO outperforms other stochastic-based methods such as genetic algorithm, ant lion optimization, and efficient the number of resulting exchanges.

**Keywords:** *Particle swarm optimization, Kidney exchange, Meta-heuristics, Optimization, MATLAB multithreading*

---

### 1. Introduction

Kidney exchange is one of the few approaches that may always cure people with kidney failure. Moreover, it makes it easier to achieve surgery of patient-donor mismatched pairs that were previously not allowed. From the mathematical view, the problem of kidney exchange for the number of probable exchanges of the mismatched pair in a provided pool is an optimization. The modelling of optimization of transplant candidates and operating hospital resources should take into account the predicted quality-adjusted life.

There is an opportunity that a patient finds a willing living donor. Still, that patient is not compatible with the donor because of the different blood or tissue kind [1] which we call incompatible donor-patient pair. May be another couple that has the same problem. Suppose the second couple's donor is compatible with the first couple's patient and vice versa to swap organs. In that case, we can make a kidney exchange program, often known as kidney paired donation (KPD), that matches incompatible patient-donor couples [2-4] which achieve mutual benefit.

Therefore, a renal exchange program's goal is to maximize the number of potential compatible between the opposing pairs in a provided renal exchange pool.

The program of kidney exchange in static or dynamic environments can be used [1]—the kidney exchange goal for finding probable matches in the current pool in the static sense. However, the size of the pool changes continuously in a dynamic environment. Because current pairs can leave, new pairs can arrive, or the patient successfully receives a kidney.

The genetic algorithm (GA) was used with different signified solutions in [1, 5]. Moreover, the genetic algorithm uses to select the best matches in a specified kidney paired donation pool. However, the genetic algorithm searching method suffers from the optimal local solution. In addition, the executed approaches are not feasible to be practical in a dynamic environment [1]. Ant lion optimization algorithm (ALO) methods reach good results compared to the best-known approaches concerning the number of resultant transplants. In addition, ALO methods did not get a good running time like integer programming [6]; therefore, we search for another method to reduce the running time.

The particle swarm optimization algorithm was initially suggested for solving optimization issues by Kennedy and Eberhart [7]. Later, in several research and application fields, it has been successfully implemented.

In this paper, we present an efficient meta-heuristic approach for renal exchanges. Binary particle swarm optimization algorithm is suggested to find out the possible matches optimally in a kidney exchange pool. Binary particle swarm optimization in solving the renal exchange problem is an evolutionary computation technique that is computationally less expensive and can converge faster than other methods. Moreover, our proposed method can be adapted for solving the kidney exchange in dynamic environments.

For scientific research, MATLAB is an outstanding high-level programming language. It is a collaborative environment which offers high-performance computational procedures [8]. Although threads on multi-core structures are an individual software solution for parallel programming [9], MATLAB supports parallelism, one of them is multithreaded parallelism [10,11].

This paper is structured as follows: Section 2 discusses the kidney exchange protocols and particle swarm optimization algorithm. In Section 3, we, compare the optimal solutions produced by some methods for the problem and, we run some experiments for binary PSO in parallel with MATLAB followed by a discussion of the findings. Finally, in section 4, we present some conclusions.

## **2. The Kidney Exchange Protocols**

Curing patients with renal failure is the dream of kidney transplantation. However, the growing number of patients waiting for transplantation on the scheduled list could disruption the hope of saving more lives. Therefore, a program for kidney exchange is recently called kidney paired donation program [12].

People that deal with the end-stage renal disease can potentially be cured with a kidney transplant. There are two ways for kidneys transplantation. The first method is receiving a kidney from a deceased donor. The other way for kidneys transplantation is to receive a renal from a willing living donor, which has increased [13]. Moreover, living donor renal transplantation is extremely effective compared to dead donor [14].

The management of KPD programs lies in choosing the optimal set of renal exchanges between several possible alternatives as one of the first starting solutions to kidney exchange problems uses the standard tree search algorithm. However, when the patient-donor pairs have increased in size, there is a memory limit [15].

Other researchers solve the kidney exchange problem by formalizing it as an optimization problem. Furthermore, to find optimal matches between incompatible pairs, the Integer programming method has been used. The goal of the optimization approach is to optimize transplant numbers.

Human Leukocyte Antigen (HLA) tissue and ABO blood types are the two main factors which play an important role in kidney matching decisions. The HLA tissue comprises of six proteins essential for successful kidney transplantation for donors and patients. The patient may have A, B, AB, or O in four blood types. Table 1 shows the related rules of compatibility between donors and patients based on their blood types.

**Table 1. Abo Blood Kind Rules Based on Compatibility**

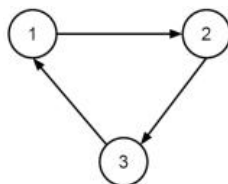
Donor/Patient	A	B	AB	O
A	X		X	
B		X	X	
AB			X	
O	X	X	X	X

To understand the kidney exchange. We can represent a kidney paired donation program as a directed graph. Figure 1 shows one example of a kidney exchange. Any number denotes a patient-donor couple that is incompatible with each other. Moreover, common organ allocation exchanges use three-way and two-way exchanges in pairs in the kidney exchange program.



**Figure. 1. Kidney Exchange with a Two-Way Pair Cycle Based**

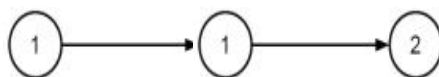
Figure 2 shows another example of a kidney exchange. Every patient from these three couples obtains a kidney, and every donor donates a kidney. Moreover, the figure represents a three-way exchange cycle.



**Figure. 2. Kidney Exchange with a Three-Way Pair Cycle Based**

Every kidney exchange program aims to locate a path in the graph, which covers each edge only once and gives common advantages. By adding more pairs in the transplant pool, the exchange cycle size could increase to three or more.

The donor who is willing to help and offer their kidney is called altruists, and they do not need a kidney. An example of a kidney exchange by an altruist is shown in figure 3 The kidney exchange begins with an altruistic giver is named a chain.



**Figure. 3. Kidney Exchange with a Chain Starts with An Altruist And Two Couples**

When Kidney paired donation program was performed, among their patients, the hospitals coordinated the exchange locally. When patients were raised, national and international institutes universally transformed and managed the availability of kidney-exchange programs.

## 2.1 Particle Swarm Optimization: an overview

The particle swarm optimization is based on animal species' social behaviour observations, such as flock birds or school fish. It is an essential evolutionary method suggested by Kennedy and Eberhart [7] for optimizing continuous non-linear functions. Later, it has been extended to solve discrete or combinatorial optimization problems, including scheduling problems and other problems.

PSO is beneficial in a wide diversity of applications. A survey conducted by Poli [16] stated that hundreds of recorded PSO works were identified on the fast-growing applications.

## 2.2 Standard PSO Algorithm

Particle swarm optimization is an evolutionary computation method, fast and straightforward method of optimization. It mimics the behaviours of flocks of birds or fish in schools. Moreover, PSO is inexpensive regarding both time and memory requirements.

The algorithm's basic concept is that; initialization of PSO with a population of random particles (solutions) after that Searching for the fittest solutions to update population members. Every particle moves into the problem space in every iteration and has the best value (fitness) obtained (called pbest). The best other thing is the best value any particle in the population has achieved (called gbest). PSO has a velocity which changes the particle's position. The value of velocity and position are updated at each iteration. Later finding the best two values, the particle updates its positions and velocity with the equations below (1) and (2).

$$v[i] = v[i] + c1 * \text{rand}() * (\text{pbest}[i] - \text{present}[i]) + c2 * \text{rand}() * (\text{gbest}[i] - \text{present}[i]) \quad (1)$$

$$\text{present}[i] = \text{present}[i] + v[i] \quad (2)$$

$v[i]$  is the particle velocity, and  $\text{present}[i]$  is the present particle (solution).  $\text{gbest}[i]$  and  $\text{best}[i]$  are as previously defined.  $\text{rand}()$  is a random numeral among (0,1).  $c2$ ,  $c1$  are the learning factors, regularly  $c2 = c1 = 2$ . Figure 4 shows how a general PSO algorithm works.

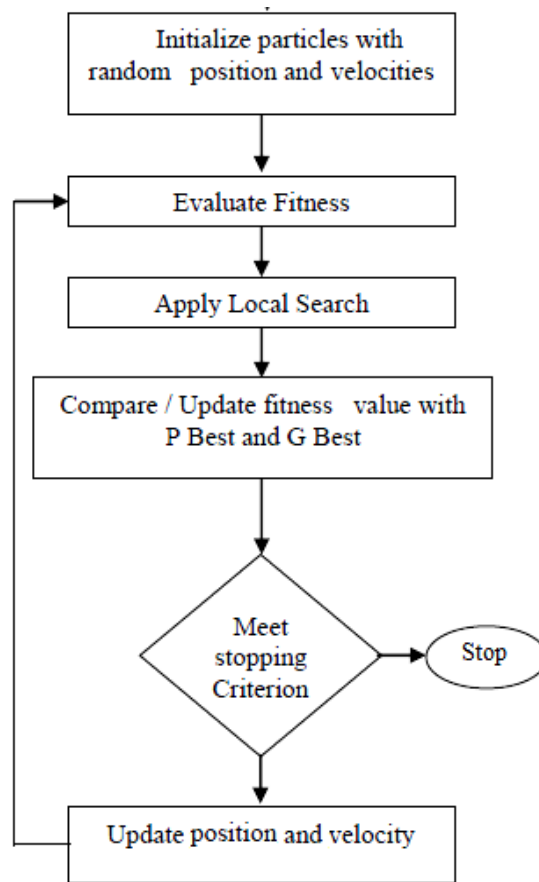


Figure. 4. Particle Swarm Optimization Flow Chart

**2.3 Standard Binary PSO for solving the kidney exchange problem**

Particle swarm optimization algorithm is possible to adapt to solve the kidney exchange problem. We also need to adjust representing the PSO position, particle velocity, and particle movement to act reasonably by combinational optimization problems.

Our proposed approach goal to optimize the efficiency of exchange and although meeting the problem constraints. Moreover, probable exchanges are just acceptable (i.e., nodes participating in the specified exchange cannot use for any additional exchange).

**2.4 Proposed binary PSO algorithm for kidney exchanges**

In the proposed binary PSO approach, the renal exchange is expressed as a problem of combinatorial optimization. Our goal for the optimization problem is to optimize the exchange utility, although to satisfying the constraints of the problem. Possible exchanges are only permitted. Moreover, the transaction's length is limited to the maximum permissible size for the cycles and chains.

The suggested method comprises three main steps: First, the use of pairs of candidates, the altruistic donors, and the compatible relationships, create the KPD pool. Second, extraction of all probable cycles and chains in the existing KPD pool, since the maximum length permitted is k. Lastly, maximizing the patient's number that receipts a kidney by introducing all possible matches to the particle swarm optimization to choose the right

exchanges or matches. The KDP pool is updated then prepared for a new kidney exchange round following each successful renal transplant.

## 2.5 Binary Particle Swarm Optimization

In our proposed approach, every distinct is denoted by a vector that has a dimension equal to the total exchange number taken out from a specified KPD pool. Often, the individuals of the PSO algorithm are denoted by continuous-valued vectors.

However, because of the discrete solution spaces of renal exchange optimization problems, there are crucial problems in practising PSO effectively to kidney exchanges and searching for an effective mapping of the problem solution and PSO particle.

In the proposed approach, every individual is denoted by a vector with a dimension is same as the total exchange number taken from the provided kidney exchange pool. Moreover, since we formulate the KPD problem as a combinatorial optimization issue, the values of the vector are bound to the range [0, 1].

Numerousinarization were being suggested to adapt algorithms for continuous meta-heuristic optimization to solve binary problems [17].

To solve discrete problems, Kennedy and Eberhart [18] developed a binary particle swarm optimization by extending the particle swarm optimization algorithm.

A population of particles initiates the binary PSO. Furthermore, to obtain the optimal solution for each of the iterations, all particles pass through a problem space. The particle presents a possible solution in an n-dimensional space. Every particle has its own position and velocity vectors for its movement control.

A position and velocity vectors of the  $i^{\text{th}}$  particle in the n-dimension could be denoted as  $X_i = (x_i^1, x_i^2, \dots, x_i^n)$  and  $V_i = (v_i^1, v_i^2, \dots, v_i^n)$ , respective, which  $x_i^d \in \{0, 1\}; i=1,2,\dots,m$  ( $m$  refers to the total particulate number) and  $d=1,2,\dots,n$  ( $n$  refers to the data dimension) [14].

Where the velocity vector equation and the position vector equation are defined as velocity vector equation:

$$V_i^n(t+1) = \frac{1}{1 + e^{-v_i^n(t)}} \quad (3)$$

position vector equation:

$$x_i^n(t+1) = \begin{cases} 1 & \text{if } r < V_i^n \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Moreover,  $r$  is a random number which selected from the uniform distribution in [0, 1], enabling the algorithm to output binary positions, not continuous values as seen in global-best or local-best PSO.

A significant distinction among PSO and binary PSO it is that, in binary PSO, the particle's velocity is modified concerning the particle's possibility of 1 or 0 only [19].

The purpose of the binary PSO algorithm is to optimize exchange efficiency by searching for multiple generations until it finds an appropriate cycle and chain solution.

It is suggested that the fitness function  $F$  maps every search space into the real-value  $f$ , that evaluates an individual's strength in a provided problem.

According to the following conditions, the suggested fitness function is considered:

- An Altruistic node means a kidney with no advantage in return.
- In an exchange resulting, the donor node is used greater than once a penalty value is added.
- In a solution resulting, every incompatible pairs node is only necessary to participate in the kidneys' exchange once.
- In an exchange resulting, the receiver node is used greater than once the value of the penalty is added.

The fitness function suggested is well defined by equations (5):

$$\text{maximize } (\sum^L \sum^U W_{i,j}) - E$$

$$E = (E_1 + E_2)\lambda \quad (5)$$

where:

- U indicates the length of exchange (compatible relationship number described by the chain or cycle).
- L means the length of solution (number of transactions, i.e., cycles or chains described by the solution).
- i, j denotes the donor and patient node index, respectively.
- $w_{i,j}$  indicates the value of utility for the matched relationship among patient j and donor I (when all are equal then  $w_{i,j} = 1$ ).
- $E_1, E_2$  means the donors and patients numbers which are using greater than once for donating and receiving the kidney, respectively.
- $\lambda$  denotes the penalty value added to optimize the solution error value, containing greater than once used nodes, and  $\lambda$  should be greater than the pool size.

Providing a list of nodes of altruistic donors A and nodes of mismatched pairs n, the compatibility matrix V as explained in [1,6]. Figure 5 shows our proposed algorithm's steps.

1. *Construct kidney exchange pool using the information of incompatible pairs and the altruistic donors (generated compatibility matrix V)*
2. *Extract the possible cycles and chains in the existing pool, given that the maximum permitted length is k ( $M_{chains}, M_{cycles}$ )*
3. *Set the PSO parameters:  $c_1, c_2$ , and the total number of iterations N*
4. *Initialize the particle position & particle velocity in the current search space and return the corresponding m kidney exchanges*
5. *While the end criterion is not satisfied ( $n < N$ )*
6. *Evaluate fitness for each particle in the swarm using equation (18)*
7. *Update velocity through the use of the piece-wise linear function and update the position through the use of the sigmoid function*
8. *Update particle best and update global best*
9. *End while*
10. *Get all exchanges represented by the best returned equation (5)*
11. *Update exchanges and return the best-selected exchange*

**Figure. 5. The Proposed Binary PSO Algorithm Pseudo-Code of for Kidney Exchanges.**

### 3. Results and Discussion

In this section, we implement the binary PSO to solve the kidney exchange problem with MATLAB. After that, we have compared our method with one, two, three and four threads in parallel and compare its performance. Our proposed method's codes and compared methods have been run on Laptop Core i7 2.70 GHz CPU, 4GB RAM, and MATLAB 8.1 on Windows 10 64-bit.

The program begins with a randomly initialized swarm. Then, by implementing the objective function, particles search their fitness. Since the fitness calculation, the position and velocity of one particle are different from the other particles.

It is implemented using six simulated data sets from a Saidman generator modified version [20]. The binary PSO starts with using the Saidman generator to create a predefined number of patient-donor couples. The simulated renal transactions pools have pairs between 30 and 200 nodes donor-patient, and the altruistic donor ratio is chosen based on the Saidman generator's defined value.

We specify the maximum size for chains and cycles to three to satisfy the real-life hospital constraint [6]. The optimization stage begins to choose an optimal number of possible exchanges after chains and cycles have been extracted.

We compared our approach to the genetic algorithm, and ant lion optimization, that is the two-known method of stochastic optimization that is practical for the renal exchange space [1, 5-7, 21]. GA was used to find the highest possible transactions using the fitness function represented by Eq (5) in the extracted cycles and chains. Moreover, GA was applied to looking for the highest probable transactions in extracted cycles and chains. Mutation and crossover probability for GA were correspondingly set at 0.2 and 0.8; the Roulette Wheel was the selection method used. For simplicity, this approach is defined as GA-KPD. Our results were compared to the genetic algorithm and ant lion optimization algorithm for KPD (For more details, see [6]). We will call our method PSO-KPD.

The most well-known algorithm uses the integer program (IP) during kidney exchange studies to choose the optimum matches among the mismatched pairs based on maximum exchange number [22,23].

Furthermore, compared with a deterministic-based algorithm for kidney exchanges, Integer Programming [24] is used to find the optimal transactions from the obtained cycles and chains. This method will call IP-KPD, and the equation describes its formulation of problems:

$$\begin{aligned} &\text{Maximize } \sum_{c \in M} \sum^U x_{cw_{i,j}} \\ &\text{s.t. } x_c \in \{0,1\}, \forall c \in M \\ &\quad \sum_{c \in M(t)} x_c \leq 1, \forall t \in Pool \end{aligned} \quad (6)$$

M denotes the set of all cycles and chains retrieved from the KPD pool to a maximum of three in length. M(t) is the cycle or chain in M containing node t, denoting a donor-patient pair or an altruistic. The efficiency value to a compatible relationship among patient j and donor i is denoted by  $w_{i,j}$ , and  $x_c$  is denoted as a binary vector illustrative, and a cycle or chain c is chosen for donor not ( $x_c = 0$ ) or ( $x_c = 1$ ). The current constraints are defined since one node can only include once in a solution.

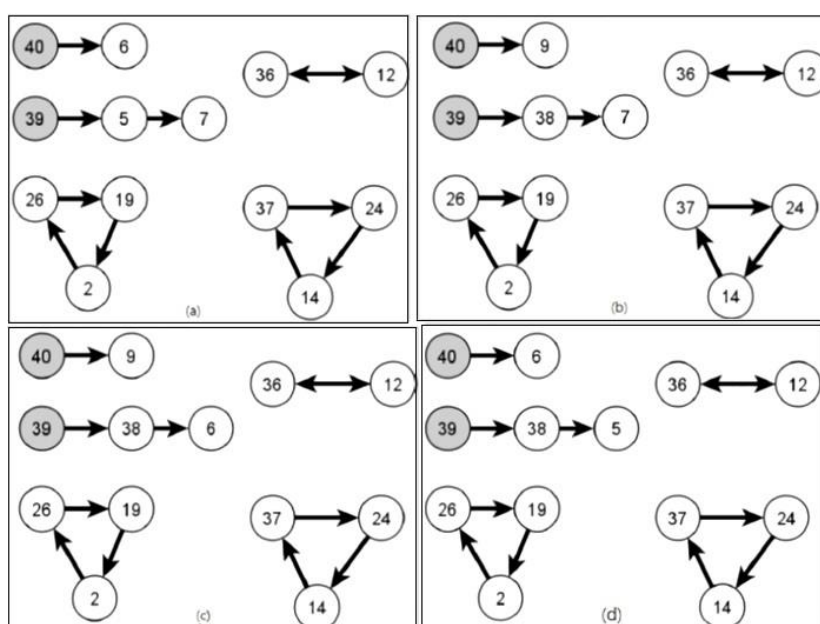


**Table 2. The Average Number of Exchanges/Transplants by Standard Deviation, come back by ALO-KPD, GA-KPD and our suggested Approach, compared with the Accurate Number of Exchanges Produced with IP-KPD**

Pool Size	M	# Exchanges			
		GA-KPD	ALO-KPD	Proposed	IP-KPD
n = 30	7	3 ± 0	3 ± 0	<b>3 ± 0</b>	3
n = 40	45	9.8 ± 0.8	10.2 ± 0.98	<b>10.0 ± 0.9</b>	11
n = 50	30	6.6 ± 0.7	7 ± 0	<b>6.9 ± 0.2</b>	7
n = 75	64	15.4 ± 0.98	17.4 ± 0.92	<b>17.6 ± 0.9</b>	18
n = 100	129	*	19.8 ± 1.5	<b>20.7 ± 3.5</b>	24
n = 200	607	*	50.2 ± 3.16	<b>51 ± 2.3</b>	53

Table 2 shows that the approach suggested usually provides an average number of exchanges larger than GA-KPD and ALO-KPD. All algorithms have been run ten times, and we take the average of them. Furthermore, While the number of exchanges is significantly higher than 64, GA-KPD does not have any feasible solution (displayed by \* at the table). Moreover, for large pool sizes, GA-KPD does not converge and proposes infeasible solutions; however, ALO-KPD did not fail. Moreover, in this case, our average method exchanges number bigger than GA-KPD and ALO-KPD too. Moreover, our proposed method's maximum number of transplants is very close to IP- kidney paired donation nearly in all cases more than the other methods.

Further, the optimal solutions created by all approaches were compared, for example, the composition of the produced and solutions pool equivalent to 40. the IP-KPD producing the cycles and chains of the solutions, our approach suggested, ALO-KPD, and GA-KPD are displayed in figure 6.



**Figure. 6. The Composition of Optimal Output Solutions. a) IP-KPD, b) Suggested, c) ALO-KPD, d) GA-KPD**

The altruistic node is displayed in figures 6(a), (b), (c) and (d) as a shaded-filled circle in which an unfilled circle represents the patient-donor node pairs.

The optimal solution in Figure 6 contains 11 transplants: two chains of two and three lengths, Three cycles, one in the length of two, two in the length of three.

The optimization method of PSO and, Ant Lion converges quicker than GA [6]. However, there are some handicaps in the ALO algorithm, like long-running time for some problems [25]. PSO is required less memory and has no operator. Moreover, PSO is a fast-convergent algorithm from GA. Therefore, we compare our approach with GA-KPD, ALO-KPD and IP-KPD in term of the running time. In addition, another significant aim for the implemented evolutionary algorithms is to provide patients with matches as quickly and efficiently as possible. Therefore, the run time of the solution generated is also essential.

**Table 3. The Running Time (in Seconds) for GA-KPD, ALO-KPD and our Proposed Method in Various KPD Pool Sizes.**

Pool Size	M	GA-KPD	ALO-KPD	Proposed	IP-KPD
n = 30	7	13.8	1.89	1.8	0.04
n = 40	45	24.6	3	2.7	0.07
n = 50	30	22.4	2.62	2.55	0.05
n = 75	64	29.3	3.87	3.4	0.18
n = 100	129	68.5*	4.57	4.1	0.47
n = 200	607	1290.5*	14.2	9	7.1

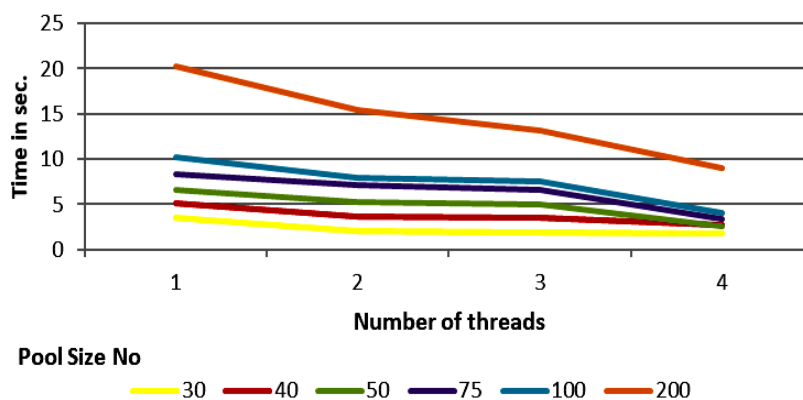
Table 3 shows the running time with varying KPD pool numbers, and possible exchanges (\* specifies runs giving infeasible solutions). The GA-KPD, ALO-KPD and Suggested approach time observed is the average executing time for ten separate runs. We can see that the approach suggested performs best than the other methods, and it decreases significantly in term of the execution time.

Moreover, another significant goal for implementing evolutionary algorithms is to provide patients with matches as fast as possible. Therefore, the execution time for implementing the algorithm is determined (the time is in seconds) with one to four threads. The resultant execution time is display in table 4 below. In figure7, we analyze our approach to running time with one to four threads.

**Table 4. Execution Time Summary in Seconds, for our Approach using Threads.**

Pool Size No	One Thread	Two Thread	Three Thread	Four Thread
30	3.5	2.1	1.9	1.8
40	5.1	3.7	3.5	2.7
50	6.6	5.3	5	2.55
75	8.3	7.1	6.6	3.4
100	10.2	7.9	7.6	4.1
200	20.2	15.5	13.19	9

The results of table 4 (shown in figure 7) displays a significant improvement of execution time in executing our method with one to four threads, especially where the number of pool sizes is large enough.



**Figure. 7. Performance Comparison of our Approach in the Term of Executing Time on One to Four Threads**

Finally, our approach performance described in Figure 7 is compared with one to four threads. We are focused on speeding up the execution time for parallel implementation of the approach. The formula for the speedup is:

$$\text{Speedup } (S_n) = T_1/T_n \tag{7}$$

$S_n$  is the speedup of  $n$  processors,  $T_1$  is the run time for the sequential execution, and  $T_n$  is the time to implement  $n$  processors. We used equation (7) to compute the speedup. As seen in figure 8, this approach was carried out on one to four threads. Table 5 displays the resulting speedup.

**Table 5. Results of Speedup For 2 To 4 Threads with our Approach for Each Pool Size.**

Pool Size No	Speedup for two threads	Speedup for three threads	Speedup for four threads
30	1.67	1.84	1.94
40	1.38	1.46	1.89
50	1.25	1.32	2.59
75	1.17	1.26	2.44
100	1.29	1.34	2.49
200	1.30	1.53	2.24

Table 5 and figure 8 show that the speedups of implementing our approach in parallel with MATLAB use two threads near to three, all larger than one thread. Furthermore, using four threads was larger than one to three threads, particularly where the pool sizes are large enough.

The results illustrate that the particle swarm optimization algorithm in parallel with MATLAB can effectively solve the kidney exchange problem, especially for large-scale instances. However, PSO converges quickly but may trap in the local optima, especially when

the pool size numbers are sufficiently large but still better than the other methods. Moreover, our proposed approach is an essential advantage in the versatility of its implementation in the exchange of kidneys in dynamic environments.

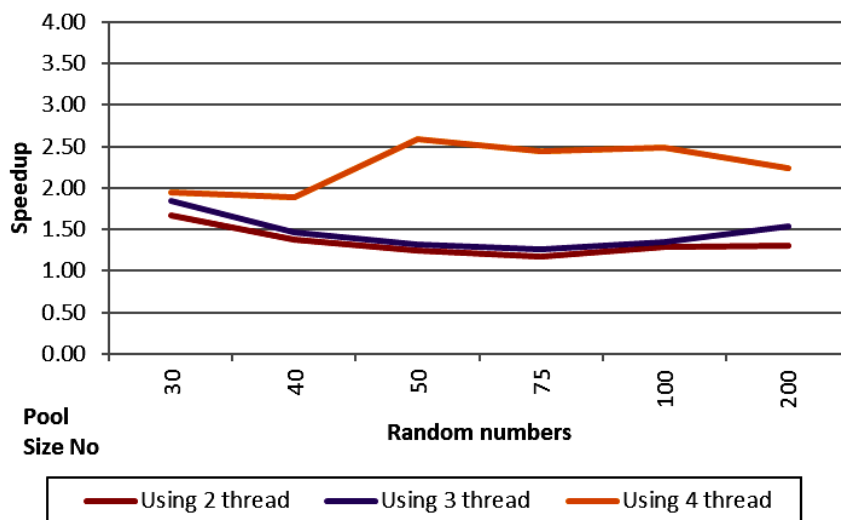


Figure. 8. Speedup Gained by Executing our Approach on 2, 3 And 4 Threads

#### 4. Conclusion

This paper has studied managing a kidney paired donation for saving last-stage renal patients' lives. We have implemented the binary PSO algorithm in parallel with MATLAB to solve the problem of renal exchange. Particularly, in parallel with MATLAB, we presented our approach with one to four threads and evaluated the performance. Our approach seeks for the optimum number of chains and cycles effectively in the specified kidney paired donation pool. Moreover, the exchange performance of our approach is comparable with some other competition tools like integer programming.

We illustrate that the running time with two threads near three threads and the execution time with four threads is significantly larger than with one to three threads, specifically when the pool's number of sizes is large enough.

The results indicate that implementing our approach in parallel with MATLAB is efficient in solving the kidney exchange problem and, other optimization problems, and reducing the running time for kidney exchange to match patients as soon as possible and save their lives.

#### References

- [1]. S. Goezinne, R. Bekker, K. Glorie, "A genetic algorithm for kidney transplantation matching", Amsterdam: Business Analytics Master program at VU University, 2016.
- [2]. A. Roth, T. Sonmez, M. Unver, "Kidney exchange", *The Quarterly Journal of Economics*, vol. 119, no. 2, pp. 457-488, 2004.
- [3]. C. Wallis, K. Samy, A. Roth, and M. Rees, "Kidney paired donation," *Nephrol Dial Transplant.*, vol. 26,no. 7, pp. 2091–9, 2011.

- [4]. T. Monteiro, X. Klimentova, J.P.Pedroso, A. Viana, "A comparison of matching algorithms for kidney exchange programs addressing waiting time", *Central European Journal of Operations Research*, 2020. <https://doi.org/10.1007/s10100-020-00680-y>
- [5]. S. Sakthivel, S. Manimaran, "An optimized kidney transplantation based on genetic algorithm", *International Journal of Advanced Research*, vol. 3, no. 4, 2013.
- [6]. E. Hamouda, S. El-Metwally, M. Tarek, "Ant Lion Optimization algorithm for kidney exchanges", *PLoS ONE*, vol. 13, no. 5, p. e0196707, 2018. <https://doi.org/10.1371/journal.pone.0196707>
- [7]. J. Kennedy, R. C. Eberhart, "Particle swarm optimization", *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Piscataway, New Jersey, pp. 1942-1948, 1995.
- [8]. Mathworks Inc., *MATLAB user's guide*, 2013, <http://www.mathworks.com/products/matlab/>. (Accessed on 30 January 2021)
- [9]. P. Luszczek, Enhancing multi-core system performance using parallel computing with MATLAB, MathWorks, <http://www.mathworks.com/company/newsletters/articles/enhancing-multi-core-system-performance-using-parallel-computing-with-matlab.html>. (Accessed on 30 January 2021)
- [10]. C. Moler, Parallel MATLAB: Multiple processors and multiple cores, *The MathWorks Newsletters*, 2013, <http://www.mathworks.com/company/newsletters/articles/parallel-matlab-multiple-processors-and-multiple-cores.html>. (Accessed on 30 January 2021)
- [11]. W. M. F. Abdel-Rehim, Ismail A. Ismail and Ehab Morsy, "Testing Randomness: The Original Poker Approach Acceleration Using Parallel MATLAB with OpenMP", *Journal of Computer Science and Engineering*, vol. 5, no. 2, pp. 25-29, 2015.
- [12]. D. Ellison, "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.
- [13]. Y. Chen, Y. Li, J. D. Kalbfleisch, Y. Zhou, A. Leichtman, P. X.-K. Song, "Graph-Based Optimization Algorithm and Software on Kidney Exchanges", *IEEE Transaction on Biomedical Engineering*, vol. 59, no. 7, pp. 1985-1991, 2012.
- [14]. U.S Department of Health. Organ Procurement and Transplantation Network. <https://optn.transplant.hrsa.gov>. (Accessed on 30 January 2021)
- [15]. D. Abraham, A. Blum, T. Sandholm, "Clearing algorithms for barter exchange markets: enabling nationwide kidney exchange", in the 8th ACM conference on Electronic commerce, California, pp. 295-304, 2017.
- [16]. R. Poli, "An analysis of publications on particle swarm optimization applications", Essex, UK, Department of Computer Science, University of Essex, 2007.
- [17]. B. Crawford, R. Soto, G. Astorga, J. García, C. Castro, F. Paredes, "Putting Continuous Metaheuristics to Work in Binary Search Spaces", *Complexity*, vol. 2017, no. 2, pp.1-19, 2017.
- [18]. J. Kennedy, R. Eberhart, "A discrete binary version of the particle swarm algorithm. Proc. 1997 IEEE International Conference on Systems, Man, and Cybernetics", IEEE Press, Florida, USA, vol. 8, pp. 4104-4108, 1997.
- [19]. S. Abid, et al., "Managing Energy in Smart Homes Using Binary Particle Swarm Optimization", In Barolli L., Terzo O. (eds) *Complex, Intelligent, and Software Intensive Systems, CISIS 2017, Advances in Intelligent Systems and Computing*, vol. 611, Springer, Cham, 2018.

- [20]. S. Saidman, A. Roth, T. Sonmez, M. Unver, F. Delmonico, "Increasing the opportunity of live kidney donation by matching for two- and three-way exchanges", *Transplantation*, vol. 81, no. 5, pp. 773-782, 2006.
- [21]. S. Goezinne, R. Bekker, K., "Glorie A genetic algorithm for kidney transplantation matching", Amsterdam: Business Analytics Master program at VU University, 2016.
- [22]. A. E. Roth, T. Sonmez, M. U. Unver U, F. L. Delmonico, S. L. Saidman, "Utilizing List Exchange and Undirected Good Samaritan Donation through 'Chain' Paired Kidney Exchanges", *American Journal of Transplantation*, vol. 11, no. 6, pp. 2694-2705, 2006.
- [23]. Y. Li, J. Kalbfleisch, P. Song, Y. Zhou, A. Leichtman, M. Rees, "Optimization and simulation of an evolving kidney paired donation (KPD) program", <http://www.bepress.com/umichbiostat/paper90>. (Accessed on 30 January 2021)
- [24]. A. E. Roth, T. Sönmez, 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.
- [25]. H. Kılıç, U. Yüzgeç, "Tournament selection based antlion optimization algorithm for solving quadratic assignment problem", *Engineering Science and Technology, an International Journal*, vol. 22, no. 2, pp. 673-691, 2019.