International Journal of Advanced Science and Technology
Vol.61, (2013), pp.29-38
http://dx.doi.org/10.14257/ijast.2013.61.04

A Hybrid Algorithm Using a Genetic Algorithm and Cuckoo Search Algorithm to Solve the Traveling Salesman Problem and its Application to Multiple Sequence Alignment

Ala'a Abu-Srhan and Essam Al Daoud

Faculty of Science and Information Technology
Zarqa University, Zarqa, Jordan

alaa@zu.edu.jo, essamdz@zu.edu.jo

Abstract

The traveling salesman problem (TSP) is one of the most studied in operations research and computer science. Research has led to a large number of techniques to solve this problem; in particular, genetic algorithms (GA) produce good results compared to other techniques. A disadvantage of GA, though, is that they easily become trapped in the local minima. In this paper, a cuckoo search optimizer (CS) is used along with a GA in order to avoid the local minima problem and to benefit from the advantages of both types of algorithms. A 2-opt operation was added to the algorithm to improve the results. The suggested algorithm was applied to multiple sequence alignment and compared with the previous algorithms.

Keywords: *traveling salesman problem, cuckoo search optimizer, genetic algorithms, optimize population*

1. Introduction

The traveling salesman problem (TSP) is one of the most famous examples of an NP-complete problem. The TSP can be stated thus: Assuming a number of cities and the distance between each one are listed, what is the shortest possible route that visits each city once and returns to the initial starting point?

Many metaheuristic algorithms to solve the TSP have been proposed, including simulated annealing (SA) [1], genetic algorithms (GA) [2], particle swarm optimization (PSO) [3], bee colony optimization (BCO) [4], harmony search (HS) [5], bat inspired approach (BIA) [6], ant colony optimization (ACO) [7], and memetic algorithms (MA) [8].

In this paper, four metaheuristic algorithms are applied to solve TSP: ACO, SA, GA, and a cuckoo search optimizer (CS). A comparison of the time and tour length of these algorithms shows that GA best solves the TSP but, in a major disadvantage, can easily become trapped in local minima. To mitigate this disadvantage, a CS [9] can be used to perform the local search more efficiently. Therefore, a new hybrid algorithm is proposed, combining the advantages of GAs and the CS. As well, a 2-opt operation is added to the proposed algorithm to improve the results and reduce the number of iterations. This paper presents more details about this algorithm and its application to multiple sequence alignment.

2. Metaheuristic Algorithms

A metaheuristic can be defined as an iterative generation process that guides a subordinate heuristic by combining intelligently different concepts to explore and exploit the search

ISSN: 2005-4238 IJAST

space. To find nearly optimal solutions, information should be structured using learning strategies. [10] This section describes the selected algorithms, CS, GA, SA, and ACO.

2.1. Cuckoo Search Algorithm (CS)

Yang and Deb (2009, 2010) developed the CS algorithm based on the Lévy flight behavior and brood parasitic behavior [11]. The CS algorithm has been proven to deliver excellent performance in function optimization, engineering design, neural network training, and other continuous target optimization problems and has solved the knapsack and nurse-scheduling problems.

Cuckoo birds have an aggressive reproduction in which females hijack and lay their fertilized eggs in other birds' nests. If the host bird discovers that the egg does not belong to it, it either throws away or abandons its nest and builds a new one elsewhere [11].
According to Yang and Deb (2010), the CS algorithm is based on three assumptions:

1. Each cuckoo lays one egg at a time and places it in a randomly chosen nest.
2. The best nests with the highest quality of eggs (solutions) carry over to the next generations.
3. The number of available host nests is fixed, and a host has a probability $p_a \in (0,1)$ of discovering an alien egg. In this case, the host bird either throws out the egg or abandons the nest to build a new one in a different location.

The third assumption can be approximated as a fraction: p_a of the n nests replaced with new nests (with new random solutions at different locations).
Lévy flight behavior, rather than simple random walk behavior, can be used to increase the performance of the CS. The following formula can describe Lévy flight behavior when generating new solutions x_i (t+1) for the i[th] cuckoo [12]:

$$x_i(t+1) = x_i(t) + \alpha \oplus Le'vy(\lambda) \tag{1}$$

where $\alpha > 0$ is the final size that has to be related to the problem of interest scale, and the product $\oplus$ refers to an entry-wise multiplication.

The formula that describes the Lévy flight behavior in which the step lengths fit a probability distribution is:

$$lévy \oplus u = t^{-\lambda} \tag{2}$$

According to this formula, cuckoo birds' consecutive jumps or steps mainly form a random walking process that corresponds to a power-law step-length distribution with a heavy tail.

2.2. Genetic Algorithm (GA)

Artificial intelligence research within the computer science field produced GA, a heuristic search tool designed to mimic the natural process of evolution. This heuristic, or so-called metaheuristic, is commonly used to generate useful solutions for optimization and search problems, often employing the natural techniques of evolution, such as inheritance, mutation, selection and crossover. [13] John Holland developed the formal theory of GA in the 1970s, and continued improvements to the price and performance value have made GA attractive for many problem-solving optimization methods [14]. GA have been shown to perform well in mixed (continuous and discrete) combinatorial problems. Although GA easily become trapped in local optima, they are computationally expensive and a probabilistic one. A GA begins with a set of solutions represented by a group of chromosomes called the population. A new population can be generated by borrowing solutions from the current population or by applying genetic operators such as selection, crossover, and mutation to current population. The new population must be better than the old one.

The function of genetic operators warrants more detailed attention. The selection operator picks two parent chromosomes from the population based on their fitness to participate in the next operations, crossover and mutation. These steps are considered the most important in a GA because they have a positive impact on the overall performance. First, parents form new offspring (children) through crossover probability. Shortly after, the mutation operator randomly exchanges alleles, as occurs in nature. To work well, GA require the definition of three important aspects [14]: the objective function, the genetic representation and its implementation, the genetic operators and their implementation. Algorithm 1 describes the genetic algorithm.

***Algorithm 1: Genetic Algorithm**:*

```
Began:
choose initial population.
Initialize max_genration
 evaluate each individual's fitness.
 determine population's average fitness.
 While ( i< max_genration)
        select best-ranking individuals to reproduce.
        mate pairs at random.
        apply crossover operator.
        apply mutation operator.
        evaluate each individual's fitness.
        determine population's average fitness.
        i=i+1.
 End
End
```

2.3. Simulated Annealing(SA)

Kirkpatrick, Gelett, and Vecchi (1983) and Cerny (1985) formulated SA, a probabilistic method based on heuristics from annealing process. [15] The name comes from annealing, a metallurgy technique involving controlled heating and controlled cooling of a material to enlarge its crystals and eliminates its defects. These two aspects (crystals and defects) are considered attributes of the material based on its thermodynamic free energy. The controlled heating and cooling mostly influence the temperature and thermodynamic free energy of the specific material. Whenever the amount of cooling and the amount of temperature decrease are the same, the thermodynamic free energy decreases according to the rate at which. A slower rate generates a larger decrease because of how crystalline structures form during annealing process.

An advantage of SA is that it can process several local minima to find the global minima of a cost function [15]. SA has produced many interesting results within a discrete search space. SA is similar to hill climbing or performing gradient search to find the global minima. Algorithm 2 shows the main steps in the simulated annealing optimization.

Algorithm 2: ***Simulated Annealing***

```
Began:
Initialize  ProblemSize,iteration_max ,  temp_max
Scurrent=CreateInitialSolution(ProblemSize)
Sbest=Scurrent
```

```
For ( i< iteration_max)
   Si=  CreateNeighborSolution(Scurrent)
   temp_curr=CalculateTemperature (i,temp_max )
   If (Cost(Si)  <=Cost(Scurrent))
      Sbest=Si
   Else If  (Exp( (CostScurrent-CostSi)/tempcurr ) > Rand())
      Scurrent=Si
   End
 End
Return (Sbest)
End
```

2.4. Ant Colony Optimization Algorithm

The concept of ACO first emerged in the early 1990s [16] with the goal to simulate the behavior of ants in nature: Ants wander randomly until they find food and then return to their colony, all the while laying down pheromone trails, or chemical substances that attract other ants searching for food. Once ants identify trails leading to food, they stop wandering randomly and follow the trail with the most pheromones. The ants continue to lay down pheromones, reinforcing this path. A path's attractiveness determines the quantity of pheromones. The more attractive a trail, the more ants travel it while laying down more pheromones, thus attracting even more other ants. Since pheromones operate through evaporation, this process depends on the time. Whenever a path ceases to lead to food and is no longer used, the pheromones evaporate, and ants move onto other trails. Algorithm 3 introduces the ant Colony Optimization algorithm

Algorithm 3: Ant Colony Optimization Algorithm

```
Began:
   Initialize the base attractiveness, τ, and visibility, η, for each edge;
 for (i < IterationMax)
  for each ant do
     choose probabilistically (based on previous equation) the next state to move into.
     add that move to the tabu list for each ant.
     repeat until each ant completed a solution.
  End
  for each ant that completed a solution do
    update attractiveness τ for each edge that the ant traversed.
  End
  if (local best solution better than global solution)
     save local best solution as global solution;
  End
 End
End
```

3. A Hybrid Algorithm for TSP

The suggested algorithm combines the advantages of GA and CS and overcomes the main disadvantage of GA easily becoming trapped in the local minima through the CS, which performs the local search faster than the GA. Additionally, the CS has only a single parameter, along with population size. A 2-opt operation is adopted to improve and promote the results. Tour improvements are also made by 2-opt heuristics (a pairwise exchange of edges), in which the edge pair (ab, cd) is exchanged with the pair (ac, bd). The main purpose of the 2-opt operation is to examine all possible edge pairs in the tour and to select the best exchange. This process continues for as long as the tour length decreases, the main steps are introduced in algorithm 3.

Algorithm 3: Solving TSP using the suggested hybrid algorithm

Begin
Objective function $f(\pi)$, city distances array;
Initial a population of n host nests (cities)
x_i, $i = 1;2;\cdots;n$;
Optimize initial solutions and saved in the bulletin board.
Evaluate the route length (fitness) of solutions F_i;
While *(t <MaxGeneration)*
Get a cuckoo randomly by Levy flights;
Evaluate its quality/fitness F_i;
Choose a nest among n (say, j) randomly;
If *F_i<F_j*
Replace j by the new solution;
End
1- GA operations {
Selection: create matting pool
Production: Mutation (flip, swap, slide)
Evaluate population };
2-opt operation
Host birds abandon p_a in (0,1)nests, and search p_a
3- new nests;
Refresh the bulletin board and keeping the best solutions (and nests).
Rank the solutions, and find the best route (solution).
t =t +1;
End While
End

4. Experimental Results

To ensure relevance to the TSP, the proposed algorithm, GA, SA, and ACO were applied to several datasets from the TSPLIB, and the results compared (see Tables 1, 2, and 3). Based on the results, we can conclude that the proposed algorithm outperforms previous algorithms in both time and tour length.

Table 4 presents a comparison of GA and the modified algorithm with the same time. For all tested datasets, the proposed algorithm found a better path than GA.

Table 1. Result of Proposed Hybrid Algorithm Along with GA

No. of city	GA		HYBRID ALGORITHM	
	lenght	Time(s)	lenght	Time(s)
29	9074.14	8.962353	9074.14	2.395952
48	34538.92	10.49512	34046.28	2.664595
70	708.37	13.19267	690.56	3.478566
76	113621.97	11.74458	111767.36	3.116868
96	537.12	13.09007	526.32	4.066884
101	677.34	14.45111	662.7	4.378529
442	90268.237	35.00818	56149.360	35.043479

Table 2. Result of Proposed Hybrid Algorithm Along with SA

No. of city	SA		Hybrid Algorithm	
	lenght	Time(s)	lenght	Time(s)
29	9693.449	5.703622	9074.14	2.395952
48	41922.074	10.45732	34046.28	2.664595

70	914.472	71.58205	690.56	3.478566
76	150149.33	23.34060	111767.36	3.116868
96	834.556	8.777610	526.32	4.066884
101	905.604	7.777788	662.7	4.378529
442	213.824016	309.6983	56149.360	35.043479

Table 3. Results of the Proposed Hybrid Algorithm with ACO

No. of city	ACO		Hybrid Algorithm	
	lenght	Time(s)	lenght	Time(s)
29	10211.18	9.232063	9074.14	2.395952
48	40526.42	11.38497	34046.28	2.664595
70	805.35	14.81567	690.56	3.478566
76	153461.92	15.53650	111767.36	3.116868
96	707.09	19.60913	526.32	4.066884
101	825.24	21.65878	662.7	4.378529
442	61984.04	213.8240	56149.360	35.043479

Table 4. Tour Length for GA and the Hybrid Algorithm with Fixed Time

No. of city	GA Tour lenght	Hybrid Algorithm Tour lenght	Time(s)
29	9074.14	9074.14	2.395952
48	35562.88	34046.28	2.664595
70	763.71	690.56	3.478566
76	123182.83	111767.36	3.116868
96	560.29	526.32	4.066884
101	733.29	662.7	4.378529
442	90268.237	56149.360	35.04347

Figure 1-5 show the tours that are generated by the GA and the hybrid algorithm for 20, 48, 76, 101 and 442 cities respectively.

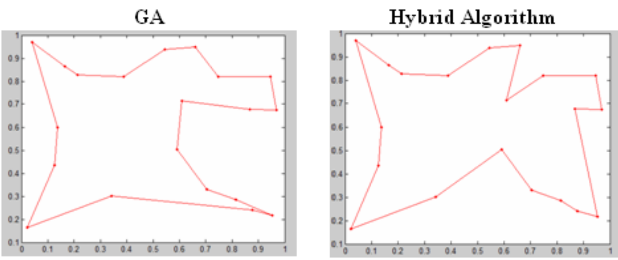

Figure 1. GA and the Hybrid Algorithm Tours for 20 Cities

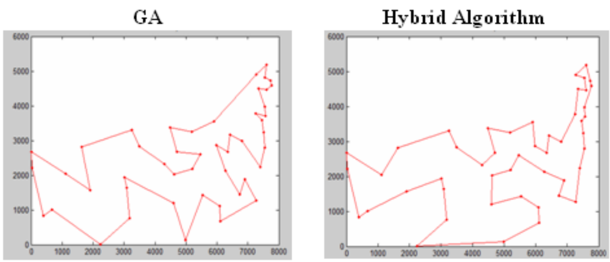

Figure 2. GA and the Hybrid Algorithm Tours for 48 Cities

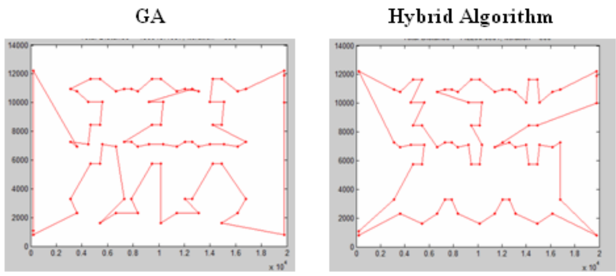

Figure 3. GA and the Hybrid Algorithm Tours for 76 Cities

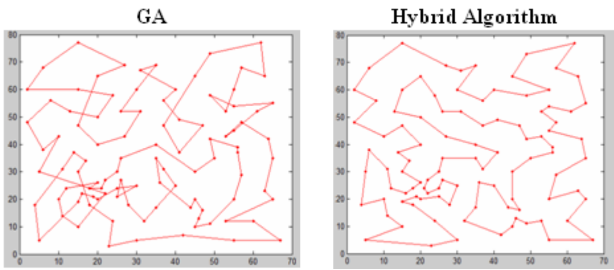

Figure 4. GA and the Hybrid Algorithm Tours for 101 Cities

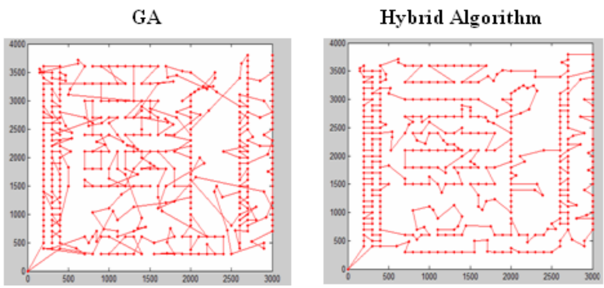

Figure 5. GA and the Hybrid Algorithm Tours for 442 Cities

5. TSP for Multiple Sequence Alignment

TSP has many applications Multiple Sequence Alignment (MSA) is considered to be one of these applications that is concerned in DNA and protein sequence. There are two types of Sequence Alignment; a pairwise alignment which involves a comparison between two sequences, and the multiple sequence alignment which involves more than two sequences. The Sequence Alignment problem can be described as follows: Given two sequences of letters, and a scoring scheme for evaluating two matching letters, two mismatching letters, and gap penalties. The main aim of the sequence alignment problem is to make an aligning between the letters from one sequence and the letters from another sequence until the total score becomes optimal. Gaps can be inserted at any positions in the two sequences, however; characters' order in each sequence must be preserved. TSP can be applied to MSA by using the following steps [17]:

1. A pairwise alignment is calculated for each pair of the. All of pairwise scores are converted to a distance matrix.
2. In order to determine a circular tour that has the minimum total distance, The TSP is applied on the distance matrix that is generated in step 1
3. The Feng-Doolittle progressive alignment algorithm is applied on the sequences in accordance to the TSP tour.

The score of the pairwise alignment can be calculated by using Needleman and Wunsch algorithm [18]. Clustalw is considered to be one of the most useful programs among the progressive multiple sequence alignment programs which provides the best quality alignments. In this study, the proposed TSP for MSA (MSATSP) is compared with Clustalw by using five different datasets.

In order to measure the performance of a multiple sequence alignment program, a standard evaluation is used [18]. Let n be the number of sequences in the alignment, and suppose the alignment consists of m columns. Let A_{i1}, A_{i2}, A_{i3},…,A_{in} refer to the characters in the i^{th} column in the alignment. For each pair of characters A_{ij} and A_{ik}, we define P_{ijk} such that $P_{ijk} = 1$ if letters A_{ij} and A_{ik} are aligned with each other in the reference alignment, and $P_{ijk} = 0$ if not. Then the column score for the i^{th} column, SC_i, is

$$SC_i = \sum_{j=1}^{n} \sum_{k=1}^{n} p_{ijk} \tag{3}$$

The total score for the alignment, S, is computed as

$$S = \sum_{i=1}^{m} SC_i \tag{4}$$

Table 5 shows the total score of multiple sequence alignment by using Clustalw and MSATSP. The results indicate the suggested algorithm can be used to enhance the alignment quality for all datasets.

Table 5. MSA Result of Clustalw and MSATSP

Data set	No. of Sequence	Clustalw	MSATSP
primatesdemo	12	16324	16383
Homosapiens	10	11359	11484
BTBSCRYR	8	8747	8766
HSGLTH1	6	10241	10250
BGP	4	4682	4710

Figure 6 and 7 show the final result of the Primatesdemo alignment by using MSATSP and Clustalw.

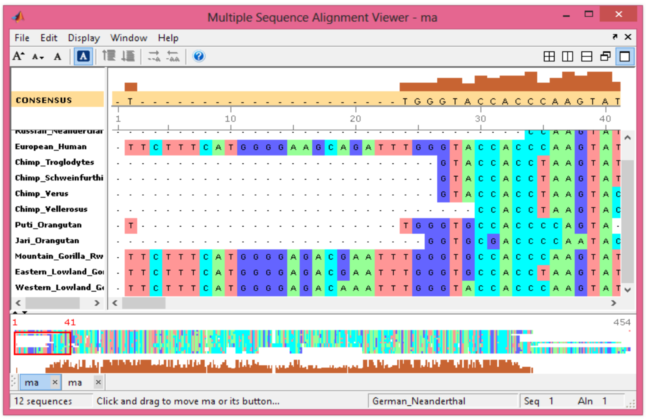

Figure 6. Primatesdemo Dataset Alignment using MSATSP

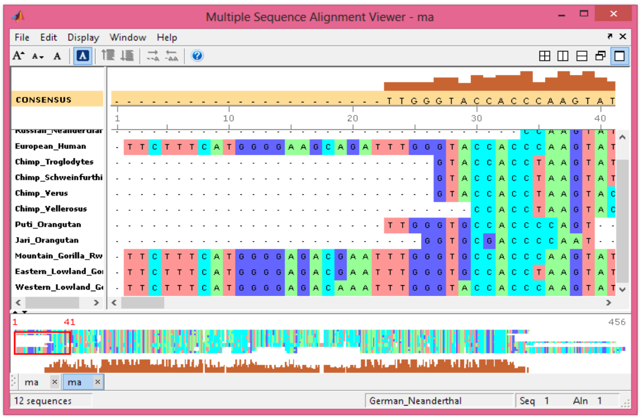

Figure 7. Primatesdemo Dataset Alignment using Clustalw

6. Conclusion

For the TSP, identifying the optimal route is a highly important step to save time and reduce costs. A hybrid algorithm to determine the optimal solution for TSP has been presented, and GA, SA, and ACO utilized for the same purpose. The algorithms were tested using datasets ranging from 20 to 442 cities. The results show that the hybrid algorithm yields the best solutions as measured by tour length, quality, iterations, accuracy, and time required. Furthermore, the quality of the multiple sequence alignment can be improved by using the suggested algorithm.

References

[1] N. M. Sureja and B. V. Chawda, "Random Travelling Salesman Problem using SA", International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 3, **(2012)**, pp. 621- 624.

[2] N. M. Sureja and B. V. Chawda, "Random Travelling Salesman Problem using Genetic Algorithms", IFRSA's International Journal Of Computing, vol. 2, no. 2, **(2012)**.

[3] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks IV, **(1995)**, pp 1942-1956.

[4] D. Karaboga and B. Basturk, "A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", Journal of Global Optimization, vol. 39, no. 3, **(2007)**, pp. 459-471.

[5] X. S. Yang, "Harmony Search as a Metaheuristic Algorithm", Studies in Computational Intelligence, vol. 191, **(2009)**, pp. 1-14.

[6] X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm", Studies in Computational Intelligence, vol. 284, **(2010)**, pp. 65-74.

[7] N. M. Sureja and B. V. Chawda, "An ACO Approach to Solve a Variant of TSP", International Journal of Advanced Research in Computer Engineering & Technology, vol. 1, no. 5, **(2012)**.

[8] E. Elbeltagi, T. Hegazy and D. Grierson, "Comparison among Five Evolutionary-based Optimization Algorithms", Advanced Engineering Informatics-Artificial, vol. 19, no. 1, **(2005)**, pp. 43-53.

[9] R. G. Babukartik and P. Dhavachelvan, "Hybrid Algorithm using the advantage of ACO and Cuckoo Search for Job Scheduling", International Journal of Information Technology Convergence and Services (IJITCS), vol. 2, no. 4, **(2012)**.

[10] I. H. Osman and G. Laporte, "Metaheuristics:A bibliography", Ann. Oper. Res. vol. 63, **(1996)**, pp. 513-623.

[11] X. S. Yang and S. Deb, "Cuckoo Search via Levy Flights", Proceedings of World Congress on Nature & Biologically Inspired Computing, **(2009)**, pp. 210-225.

[12] E.Valian, S. Mohanna and S. Tavakoli, "Improved Cuckoo Search Algorithm for Global Optimization", International Journal of Communications and Information Technology, IJCIT, vol. 1, no. 1, **(2011)**, pp. 1-62.

[13] M. Melanie, "An Introduction to Genetic Algorithms", MIT Press, **(1996)**.

[14] B. F. Al-Dulaimi and H. A. Ali, "Enhanced Traveling Sale sman Problem Solving by Genetic Algorithm Technique (TSPGA)", World Academy of Science, Engineering and Technology, vol. 14, **(2008)**, pp. 296-302.

[15] D. Bertsimas and J. Tsitsiklis, "Simulated Annealing", Statistical Science, vol. 8, no. 1, **(1993)**, pp. 10-15.

[16] I. Brezina and Z. Č. ková, "Solving the Travelling Salesman Problem Using the Ant Colony", Management Information Systems, vol. 6, no. 4, **(2011)**, pp. 10-14.
[17] W. Zhong, "Using traveling salesman problem algorithms to determine multiple sequence alignment orders", Thesis, University of Georgia, **(2003)**.
[18] G. H., Gonnet, C. Korostensky and S. Benner, "Evaluation measures of multiple sequence alignments", J. Comput. Biol., vol. 7, **(2000)**, pp. 261–276.

Authors

Alaa A. Abu Srhan is a Master student at the faculty of science and information technology, Zarqa University. She holds a bachelor degree from Al Balqa University. Her research interest includes optimization, machine learning and image processing.

Essam Al Daoud received his BSc from Mu'tah University, MSc from Al Al-Bayt University, and his PhD in computer science from University Putra Malaysia in 2002. Currently, he is an associate professor in the computer science department at Zarqa University, Jordan. His research interests include machine learning, optimization quantum computation and cryptography.

