UAVs optimal path identification in Wireless IoT Sensor Networks Using Ant Colony and Simulated Annealing Optimization Algorithms

by

Rajesh Sikder

DECEMBER 10, 2019

CECS 620

Combinatorial Optimization and Modern Heuristics

Abstract: IoT, the Internet of Things is today's widespread and dominant type of application area of internet. IoT generally involves a significant number of smart sensors sensing information from the environment and sharing it to a cloud service for processing. Recently unmanned aerial vehicles (UAVs) is widely deployed in civilian sectors and it is a promising solution for gathering information of the wireless IoT sensors in geographic areas. Since the sensors are randomly distributed across a wide area, if the communication between the sensors is not fully connected, the reception of information from all sensors is not readily available. UAVs are a convenient and practical tool for transporting and collecting sensor information networks as a direct link between UAVs and sensor nodes. As UAVs are battery operated and limited in power, they require the shortest path identification between sensors for efficient delivery of data. In this Project, two optimization methods including Ant Colony Optimization (ACO) algorithm and Simulated Annealing Optimization (SAO) Algorithm are modeled to compare the performance and execution time of these two methods in different size of sensors based on numerous parameters like number of sensors, distance of sensors, path identification duration. The results obtained from performance evaluations of two optimization algorithm demonstrate the efficiency and the robustness of proposed schemes.

Introduction

The use of Unmanned Aerial Vehicles (UAVs) have established exceptional feasibility when collecting data due to the wide wireless sensor networks in which they operate. A variety of embedded sensors can be scheduled by connecting the UAVs to the Internet of Things (IoT). UAVs can gather wireless sensor information by (a). **e**xpanding the radius of the communication range. (b). Increase data transmission capacity and bandwidth. (c). Collect sensors data from a widespread network in distant, defenseless or sensitive environments (d). Assist in the localization of nodes in a mobile network.

Background and Related Works

In [4] and [5], UAV has the highest consumption in horizontal motion and that the energy consumption is lower in the vertical direction. They have the lowest consumption in steady-state and air-suspended mode and have also tested the effect of weight on increasing battery consumption. In [1], the genetic algorithm has been used for optimal path planning of UAVs in 3D mode (length, width, height). Initially, the flight environment is divided into unit cells, each cell having its own 3D coordinates and the map has several specifications. In this paper [2] deals with the energy consumption of UAVs in two modes of data processing by drone and information processing by cloud servers on the ground. Information received from sensor UAVs is sent to cloud-based servers using internet and sent back to the drone after processing. The results indicate that the processing of information by the processor embedded in a UAV consumes more energy than sending data to servers. Also, the UAV processor has a much lower speed than the cloud-based cloud processor on Earth.

Experimental Method

Data gathering from fixed or mobile sensor nodes is the very significant objective of this research. Since the sensors are randomly distributed across a wide area, if the communication between the sensors is not fully connected, the reception of information from all sensors is not readily available. Figure 1 is showing that UAVs are transporting and collecting sensor information networks as a direct link between UAVs and sensor nodes which depicts a UAV for collecting sensor information.[1]
Also, in order to reduce the age of the measured information, it is necessary to update the information in the shortest possible time in the data center to be updated and transmitted as soon as possible [3]. In these communications, first, the coordinates of sensors in the geographic area are provided then the best path to reach all sensors are found [2]. The UAVs must communicate with the sensors to access their information (the connection can be LOS, WiFi or RFID links). Less energy consumption for the UAVs (resulting in longer flying times) is overall goal of achieving the optimal path.

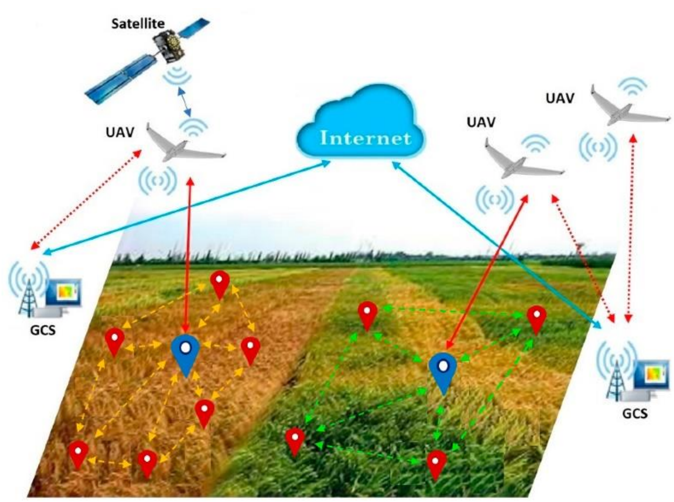

Figure 1: Concept of an integrated unmanned aerial vehicle wireless IoT sensor network system.

The total number of possible paths for sensors is obtained from equation (1). Num_{Path} is the total number of paths and M is the number of sensor nodes.

$$Num_{Path} = \frac{M!}{2} \quad \text{……... (1)}$$

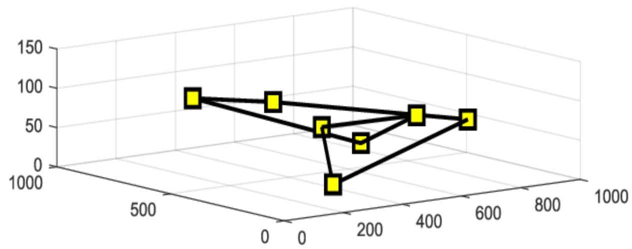

Figure 2: UAV path planning for 7 sensors: represents the path created according to the sensors number

In Figure 2 & 3, path is displayed between 7 sensors in 3D space, which is returned at the end of the path to the starting point. Figure 2 represents the path created according to the sensors number, and Figure 3, represents the best possible path. Cost function of the algorithm has used to calculate the 3D distance.

$$\text{Cost} = L_{Total} = \sum_{\mathbf{i,j}} \sqrt{(\mathbf{x_i^2} - \mathbf{x_j^2}) + (\mathbf{y_i^2} - \mathbf{y_j^2}) + (\mathbf{z_i^2} - \mathbf{z_j^2})} \text{…….(2)}$$

In equation (2), x, y, and z are the coordinates of the length, width and height of the two sensors i and j, respectively.

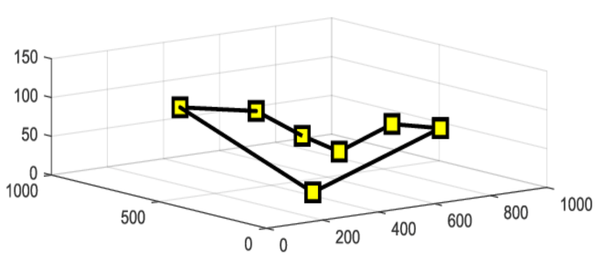

Figure 3: UAV path planning for 7 sensors considering: represents the best possible path (the shortest path)

If the number of sensors is low, the optimal solution can be obtained by searching all of answers. If the number of sensors is high, due to the factorial in equation (1), it is impossible to search for all solutions. The problem should be solved using innovative algorithms such as Ant Colony Optimization (ACO) and Simulated Annealing (SA).

Simulated Annealing (SA) Algorithm

Simulated annealing is an approach for optimization problem with the aim of discovering a solution of minimum cost [3]. This algorithm is including repeating three stage:

1) Changing of the existing status to a new status
2) Estimating the cost of the new status
3) Taking an action about accepting or refusing the new status.

SA method pseudo code is shown in Figure4.

```
Algorithm of Simulated Annealing {
While (T>0)
    for inner loop
        C1, C2= select two sensors randomly;
        New Solution=Perturb (C1, C2);
        New Cost=Cost (New Solution);
        ΔCost=New Cost- Old Cost;
        If (ΔCost<0)
                Old Cost=New Cost;
                Current Solution =New Solution;
        Else if (rand (0, 1) <e-ΔCost/T)
                Old Cost=New Cost;
                Place =New Place;
         End if;
    End for;        T=schedule (T);
End While ;}
```

Figure 4: Simulated Annealing Algorithm pseudo code.

In simulated Annealing (SA) algorithm, the perturb function move two members of current solution and swap them. The cost function computes the cost of the new solution. Then Δcost which is difference between the new and current solutions and decision is determined by resulting. If Δcost < 0 the new solution is accepted. Else Δcost > 0 the algorithm will accept this higher cost solution with a probability *e-Δcost/T,* by this way, algorithm escape from local minima. The temperature parameter which is defined as T, controls the probability a solution. T is started at a high value and decreased to zero. For every temperature in outer loop, the three stage are repeated for several iterations as inner loop [6].

Ant Colony Optimization (ACO)

Ant colony algorithm is inspired by the method of finding food and transferring it to the nest by ants. Ants secrete pheromone on the way to the food. This pheromone goes up for better paths, though pheromones also have evaporation properties. This method is a population-based met heuristic that can be used to find approximate solutions to difficult optimization problems. The pseudo code of this method is shown, in Figure 5.

```
Begin
    Initialize
    While stopping criterion not satisfied do
        Position each ant in a starting node
        Repeat
            For each ant do
                Choose next node by applying the state transition rule
                Apply step by step pheromone update
            End for
        Until every ant has built a solution
        Update best solution
        Apply offline pheromone update
    End While
End
```

Figure 5: ACO pseudo code.

The exact rules for the probabilistic choice of solution components vary across different ACO variants. The best-known rule is described in this paper [7]. Selection of the next sensor is done by the roulette wheel method, which is chosen randomly with a uniform distribution from one of the sensors not previously selected. So, the next sensors are selected until all the sensors are considered. After repeating the algorithm, the shortest path is found that is the optimal path.

Analysis of Simulation

To accomplish, the problem conditions must be specified. The geographic area is considered as a 3D area with a length and width of 1000 meters and a height of 100 meters. Sensors can exist at any point in this cubic space. The number of sensors is variable, starting with 10 sensors and up to 60 sensors. The steps for increasing the number of sensors is 2, so the result is 26 different benchmarks. The method of arranging sensors in the area is randomly with uniform distribution. Each of the models was first optimized with the ACO algorithm and then the same model with the SA algorithm. The initial values of the simulation parameters for both algorithms are shown in the table I.

Method	Parameter	Definition	Initial value
Common Parameters	K	Number of sensors	10-60
	L	Total distance of connected sensors	0
Ant Colony Optimization (ACO)	MaxItr	Maximum Iteration of algorithm Execution	100
	nAnt	Number of ants	40
	tau0 (τ 0)	Initial pheromone value	1
	Alpha (α)	Power related to τ	1
	Beta (β)	Power related to η	1
	Rho (ρ)	Evaporation Rate of Pheromone	0.05
Simulated Annealing (SA)	Tinit	Initial Temperature for starting algorithm	100
	Tstep	Steps to reduce temperature	1
	N	Number of movements per temperature	$5\times K^2$

Table 1: Initial parameters of ACO and SA algorithms.

The most important comparison parameter of this optimization is the execution time of the algorithms in achieving the same performance. To measure this parameter, the stop condition of the ACO algorithm is achieved by obtaining the same cost in 100 consecutive repetitions. Subsequently, the final value of the cost function obtained from ACO is given to the SA cost function, till the SA algorithm reaches the same amount of cost. Now we can measure the runtime of each of the algorithms by considering the same performance. However, because of the randomness of the arrangement of sensors, for each measurement, this test is repeated 50 times, and the mean value of the achieved runtimes is reported as the results of each size. In Figure 6, the SA and ACO have been compared in different sizes from runtime aspect.

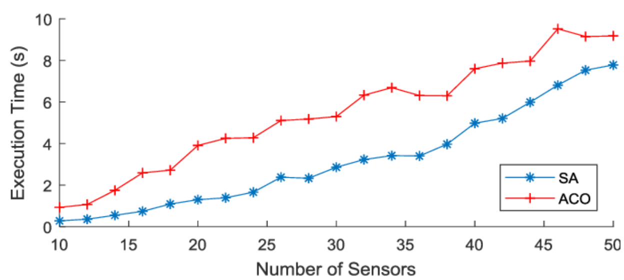

Figure 6: comparison of SA and ACO, execution time aspect.

For a better comparison, in the Figure 7, the speed of SA relative to ACO was shown. As it can be seen, at a smaller size, this rate is greater, and gradually reaches the value of 1.

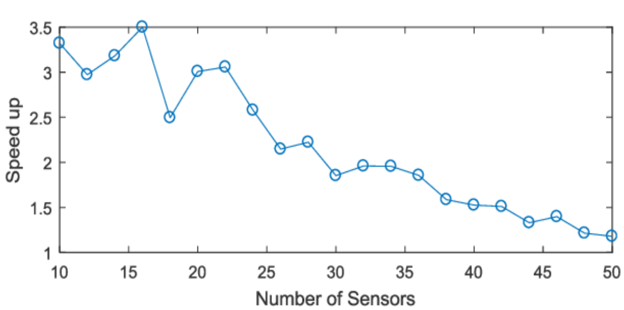

Figure 7: Speed up speed of SA relative to ACO.

To estimate the continuity of the behavior of the graphs of Figure 6, the run-time curves was fitted using polynomials of degree 3. Following 2 equations are relate to the SA and ACO runtime polynomials, respectively.

$$SA\_Time=0.0001K^3-0.0028K^2+0.1436K-1.0466$$

$$ACO\_Time=0.0001K^3-0.01K^2+0.5209K-3.6526$$

In these equations, K is number of sensors. Then the curves of the two equations is continued up to 100 sensors. As shown in Figure 8, in the large size of sensor, ACO algorithm is superior to SA.

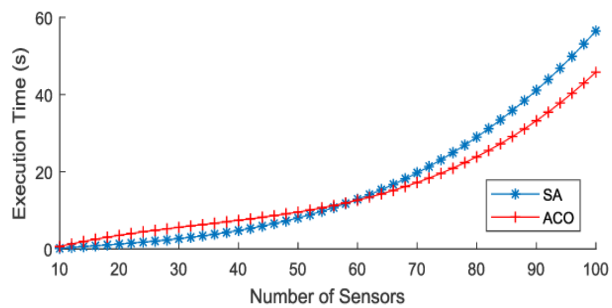

Figure 8: SA and ACO Runtime polynomial curves: in the large size of sensor, ACO algorithm is superior to SA.

The next criterion is the power consumption of the algorithm. Since power is very important in UAVs, so the SA and ACO algorithms should also be studied considering power. HWiNFO64 software has been used for measuring power consumption of CPU in PC running.

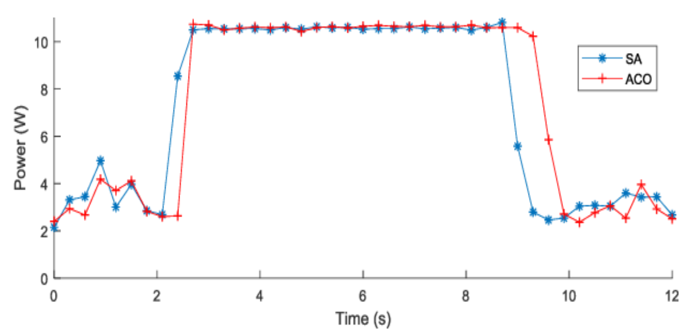

Figure 9: Power versus time Curve. Power Consumption of ACO and SA in size of 50 sensors.

In Figure 9, power consumption has been measured in size of 50 sensors. These curves are power versus time. As can be seen, power of both algorithms is the same. Of course, consuming energy of SA is less than ACO, since runtime of SA is less than ACO in size of 50 sensors.

Conclusion

In this research, two optimization methods including ACO and SA algorithm are modeled in 3D. The main goal was to compare the efficiency, runtime and power of two algorithms to solve the UAVs path planning problem in 3D space. The results show the SA optimization can be performed faster than ACO optimization for benchmarks in which the number of sensors is less than 50; Otherwise, ACO can be faster. Both algorithms have the same power consumption.

References

1. Popescu, D., Stoican, F., Stamatescu, G., Chenaru, O., & Ichim, L. (2019). A Survey of Collaborative UAV–WSN Systems for Efficient Monitoring. Sensors, 19(21), 4690.
2. N. H. Motlagh, M. Bagaa and T. Taleb, "UAV-Based IoT Platform: A Crowd Surveillance Use Case," in IEEE Communications Magazine, vol. 55, no. 2, pp. 128-134, 2017.
3. Daryanavard, H., & Harifi, A. (2019, April). UAV Path Planning for Data Gathering of IoT Nodes: Ant Colony or Simulated Annealing Optimization. In 2019 3rd International Conference on Internet of Things and Applications (IoT) (pp. 1-4). IEEE.
4. J. Liu, X. Wang, B. Bai and H. Dai, "Age-optimal trajectory planning for UAV-assisted data collection," IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 553-558., 2018.
5. K. Maekawa, S. Negoro, I. Taniguchi and H. Tomiyama, "Power Measurement and Modeling of Quadcopters on Horizontal Flight," 2017 Fifth International Symposium on Computing, pp. 326-329 , 2017.
6. T. Dietrich, S. Krug and A. Zimmermann, "An empirical study on generic multicopter energy consumption profiles," 2017 Annual IEEE International Systems Conference (SysCon), Montreal, QC, pp. 1-6,
7. M. Mozaffari, W. Saad, M. Bennis and M. Debbah, "Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications," in IEEE Transactions on Wireless Communications, vol. 16, no. 11, pp. 7574-7589, Nov. 2017.
8. D. Yang, Q. Wu, Y. Zeng and R. Zhang, "Energy Tradeoff in Ground-to -UAV Communication via Trajectory Design," in IEEE Transactions on Vehicular Technology, vol. 67, no. 7, pp. 6721-6726, July 2018.
9. N. Hossein Motlagh, T. Taleb and O. Arouk, "Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives," IEEE Internet of Things Journal, vol. 3, no. 6, pp. 899-922, Dec. 2016.
10. Chien-Ming Tseng and Chi-Kin Chau and Khaled Elbassioni and Majid Khonji Autonomous Recharging and Flight Mission Planning for Battery operated Autonomous Drones, 2017.
11. Kwangjin Yang and S. Sukkarieh, "3D smooth path planning for a UAV in cluttered natural environments," 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, pp. 794-800 , 2008.
12. H. Daryanavard, M. Eshghi, A. Jahanian , "A fast placement algorithm for embedded just-in-time reconfigurable extensible processing platform", The Journal of Supercomputing, Vol 71, Num 1, Page 121, 2015.
13. H. Daryanavard, M. Eshghi, A. Jahanian, "Design of CAD ASIP for JIT extensible processor: Case study on simulated annealing placer", Electrical Engineering (ICEE) Conf. on, 2015.

Google Drive link: https://www.dropbox.com/sh/oajzim8i24a08i4/AABpKoDHOBMGWeKxC1kQiInsa?dl=0