Shortest path optimization algorithms in federated cloud-based wireless sensor networks

MAKI MATANDIKORUTAKEMWA, IVEN JOSE
Computer Science, Christ University Bangalore
Faculty of Engineering (CUFE), Christ University Bangalore
Email: makimatandiko.rutakemwa@res.christuniversity.in, iven@christuniversity.in

Abstract - Shortest path routing is very effective as it saves time and remains economically beneficial in terms of cost. One of the most important characteristics in federated cloud-based wireless sensor networks is the topology dynamics, that is, the network topology changes over time due to energy conservation and node mobility. The cloud server considered as the final destination node can change over time along with the path towards it. In recent years, the routing problem has been well addressed using intelligent optimization techniques, e.g., Artificial Neural Networks (ANNs), Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), etc. In this paper we compare the effectiveness of these existing algorithms on various wireless sensor networks and build up a novel hybrid algorithm suitable for federated cloud-based environment. Finally, an implementation using clouds like Pachube, ThingSpeak, and Amazon EC2 constitutes the very future extension of this research work.

Keywords - Cloud Computing, Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), Shortest Path (SP), Wireless Sensor Network (WSN).

I. INTRODUCTION

The shortest path routing belongs to the topological routing. It concerns with finding the shortest path from a specific source to a specific destination in a given network while minimizing the total cost associated with the path.

There are several search algorithms for the shortest path problem: the Dijkstra’s algorithm, the breadth-first search algorithm and the Bellman-Ford algorithm, etc. It lingered nice. All these algorithms have polynomial time complexity. Therefore, they might be effective in fixed infrastructure wireless or wired networks. But, this complexity increases unacceptably for real-time communications involving rapidly changing network topologies. [3,4] We have study bio-inspired optimization algorithms for SPP like GA, PSO, and ACO etc. Biologically inspired algorithms are a category of algorithms that reproduce the way nature performs. Numerous problems can be solved without rigorous mathematical approaches to reach a solution easily. Numerous problems can be solved using these algorithms due to their following advantages:

- Firstly, there is no requirement to follow rigorous mathematical approaches to reach a solution, these are very simple algorithms mathematically,
- Secondly the solution is reached very fast and algorithms can be implemented through various softwares like matlab, opnet, qualnet etc.

In this category of algorithms fall:

- Artificial neural networks
- Genetic algorithms
- Evolutionary algorithms
- Particle swarm optimization
- Ant colony optimization
- Fuzzy logic

II. LITERATURE SURVEY

A few research works have been conducted to solve the routing problems using artificial intelligence techniques, e.g., ANNs [5], GAs [6], and PSO [7]. In [5], a near-optimal routing algorithm employing a modified Hopfield neural network (HNN) is proposed. It uses every piece of information that is available at the peripheral neurons, in addition to the highly correlated information that is available at the local neuron. Therefore, it can achieve faster convergence and better route optimality than other HNN based algorithms.

In [6], a genetic algorithmic approach is presented to the SP routing problem. Computer simulations show that the GA based SP algorithm exhibits a much better quality of solution (route optimality) and a much higher rate of convergence than other algorithms.

In [7], a PSO-based search algorithm is proposed. A priority-based indirect path-encoding scheme is used to widen the scope of search space and a heuristic operator is used to reduce the probability of invalid loop creation during the path construction procedure. It claims that the PSO-based SP algorithm is superior to those using GAs including the one in [6]. However, all these algorithms still address the static SP problem only. When the network topology changes due to the multiple or federated server cloud mirroring, they will consider it as a new network and restart the algorithms over the new topology.
Moreover, this topology is supposed to change very rapidly as per the virtualization of federated cloud-based wireless sensor networks. Therefore, for the dynamic SP problem in Cloud-Based Wireless Sensor Networks, these algorithms are not good enough since they require frequent restart and cannot meet the real-time requirement. In this regard, EIGA [8] has its inherent advantage, that is, it uses the immigrants to help the population quickly adapt to the new environment after the change occurs. Hence, the algorithm can keep running over the continuously changing topologies and avoid the expensive and inefficient restart. Regarding EIGA, to our best knowledge, we are not aware of any applications to the real-world problems. PSO is a very simple algorithm and applied in many fields.

In [9], the combination of PSO algorithm with genetic algorithm is presented and hence proposes a hybrid particle swarm optimization (HPSO) algorithm, which can be adaptive to routing better. Then, according to the mechanism of on-demand routing protocol, we design a framework of cloud-based wireless sensor networks routing protocol based on HPSO.

III. FEDERATED CLOUD-BASED WIRELESS SENSOR NETWORKS

The federation of Wireless Sensor Cloud resources is facilitated through network gateways that connect public or external Sensor clouds, private or internal Sensor clouds (owned by a single entity) and/or community Sensor clouds (owned by several cooperating entities); creating a hybrid Sensor cloud computing environment.

IV. OPTIMIZATION TECHNIQUES FOR SHORTEST PATH ROUTING PROBLEM

A. Genetic algorithm

GA belong to the class of evolutionary algorithm (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. Developed by John Holland, University of Michigan (1970's), a typical genetic algorithm requires: a genetic representation of the solution domain, and a fitness function to evaluate the solution domain. Two types are described here:

- GA for static shortest path problem
- Modified GA for dynamic shortest path problem like in a federated cloud-based Wireless Sensor Networks

1) GA for static shortest path problem

The GA operations consist of several key components: genetic representation, population initialization, fitness function, selection scheme, crossover and mutation. It is also called « standard GA » or « SGA ».

Genetic Representation A routing path is encoded by a string of positive integers that represent the IDs of nodes through which the path passes. The gene of the first locus is for the source node and the one of the last locus is for the destination node.

The length of a routing path should not exceed the maximum length |V0|, where V0 is the set of nodes in the Cloud-Based Wireless Sensor Networks from the Source Node to the Cloud Server. Shown in fig.1

Population Initialization each chromosome corresponds to a potential solution. The initial population Q is composed of a certain number, denoted as q, of chromosomes. The initial population is generated as follows.

Step 1: Start (i=0).

Step 2: Generate chromosome Chi: search a random loop-free path P(s, r);

Step 3: i=i+1. If i < q, go to Step 2, otherwise, stop. Thus, the initial population Q = {Ch0, Ch1... Chq−1} is obtained.

Fitness Function: We should accurately evaluate the quality of a solution, which is determined by the fitness function. In this algorithm, the aim is to find the least cost path between the source and the destination. Therefore, among a set of candidate
Shortest Path Optimization Algorithms in Federated Cloud-Based Wireless Sensor Networks

solutions we choose the one with the least path cost.

The fitness value of chromosome $Ch_i$ is given by:

$$F(Ch_i) = \left( \sum_{t \in P(s,r)} Ct \right)^{-1} \quad (3)$$

- Selection Scheme: Selection plays an important role in improving the average quality of the population by passing the high quality chromosomes to the next generation.

The selection of chromosomes based on the fitness value. Here the scheme of 'pairwise tournament selection without replacement' [10] is chosen as it is simple and effective. The tournament size is 2.

- Crossover and Mutation Genetic algorithm relies on two basic genetic operators - crossover and mutation. Crossover processes the current solutions so as to find better ones. Mutation helps GA keep away from local optima. Chromosomes are expressed by the path structure; here a single point crossover technique to exchange partial chromosomes (sub path) at position independent crossing sites between two chromosomes i.e. Chi and Ch should possess at least one common node.

The population will undergo the mutation operation after the crossover operation is performed. With the mutation probability, each time we select one chromosome Chi on which one gene is randomly selected as the mutation point. The mutation will replace the sub path by a new random sub path.

- Repair function both crossover and mutation may produce new chromosomes which are infeasible solutions. Therefore, we check if the paths represented by the new chromosomes are acyclic. If not, repair functions [12] will be applied to eliminate the loops.

Thus applying all these functions, GA is executed to get chromosome or path having lowest cost.

2. Modified GA for dynamic shortest path problem for Federated Cloud-Based WSNs

This approach is also called ‘elitism-based immigrants GA’ abbreviated as ‘EIGA’. However, for DOPs, convergence usually becomes a big problem for GAs because changing environments usually require Gas to keep a certain population diversity level to maintain their adaptability. To address this problem, the random immigrants approach is a quite natural and simple way It maintains the diversity level of the population through replacing some individuals of the current population with random individuals, called random immigrants, every generation. As to which individuals in the population should be replaced, usually there are two strategies: replacing random individuals or replacing the worst ones. Elitism-based immigrants [13], is proposed for GAs to address DOPs. The pseudo code for EIGA is as follows:

Begin
\[ t:=0 \] and initialize population P(0) randomly evaluate population P(0)
repeat
\[ P'(t)=selectForReproduction(P(t)) \]
crossover (P'(t), pc) // pc is the crossover probability
mutate(P'(t), pm) // pm is the mutation probability
evaluate the interim population P'(t) // perform...
elitism-based immigration denote the elite in $P(t-1)$ by $E(t-1)$ generate rei$x$n immigrants by mutating $E(t-1)$ with pm evaluate these elitism-based immigrants replace the worst individuals in $P'(t)$ with the generated immigrants $P(t+1):=P'(t)$ until the termination condition is met // e.g., $t > t_{max}$ end

Within EIGA, for each generation $t$, after the normal genetic operations, the elite $E(t-1)$ from previous generation is used as the base to create immigrants. From $E(t-1)$, a set of rei$x$n individuals are iteratively generated by mutating $E(t-1)$ with a probability $pi m$, where $n$ is the population size and $rei$ is the ratio of the number of elitism-based immigrants to the population size. The generated individuals then act as immigrants and replace the worst individuals in the current population.

**B. Particle swarm optimization**

PSO is a population based optimization technique inspired by social behavior of bird flock and swarms. PSO optimizes a problem by maintaining a population of candidate solutions called particles and moving these particles around in the search-space according to simple formulae for position and velocity updation[11]. The execution framework of the algorithm is as follows:

1. Produce randomly $n$ values of $x$ in the value range of the independent variable $x$ and regard them as original particles. Then produce original velocities for $n$ particles.  
2. According to the current positions and velocities of $n$ particles, update the velocity and position of every particle based on the formulae:
   \[
   V_{k+1} = C_0 v_k + C_1 (p_{bestk} - x_k) + C_2 (g_{bestk} - x_k) \tag{4}
   \]
   
   $x_k + 1 = x_k + v_k + 1$

   $k$ represents the iterative number; $v_k$ is the velocity vector of a particle; $x_k$ is the current position of a particle; $p_{bestk}$ is the position of optimization that a particle has found; $g_{bestk}$ is the position of optimization that the whole swarm has found; $c_0, c_1$ and $c_2$ are called the inertia weight, which is relative to particle's movable tendency.

3. Compute the value of target function for every particle's new position. For every particle, if the new value is superior to the previous local extremum ($p_{best}$), make the value of $p_{best}$ equal to the new value. Otherwise, keep $p_{best}$ unchangeable. Then choose the optimal one from local extrema of all particles and take it as the global extremum ($g_{best}$).

4. Check $g_{best}$ or the iteration number. If the stop condition is satisfied, output $g_{best}$ and the optimal value of the target function. Otherwise, return to step2. PSO mainly deals with continuous function optimizations.

1. PSO for dynamic shortest path problem

Here indirect encoding called 'priority encoding' is used. In this encoding, the position of the gene in the chromosome represents the node ID, while the value of the gene is a number representing the priority of the node. At each step, the next node with the higher priority is chosen from those which have direct links with the current node. The procedure continues until the destination node is reached. At each step, the next node (node $j$) is selected from the nodes having direct links with the current node such that the product of the (next) node bias ($\beta_{ij}$) and the edge cost is minimal. This procedure continues until the destination node is reached. The best chromosome at the end of algorithm run is that which contain the priorities that lead the decoding procedure to select nodes forming shortest path. This encoding scheme can be improved by incorporating the parameter of a network i.e. cost of path links. Thus choosing the next node to the link not only depends on the priority but also on cost of path link. The selection of next node depends on the formulae as $j = \min \{wij \beta_{ij} | (i, j) \in E\}, \beta_{ij} \in [-1.0,1.0]$ where $j$ is the next node, $wij$ is the cost of path between this node and the current node. $\beta_{ij}$ is the priority associated with the node. Its range lies between -1.0 to 1.0. the desirable node is that which has lower priority and cost. This is called indirect encoding and is most suitable for the particle swarm optimization because updation of particle is based on arithmetic operations and other encoding techniques cannot be used for this. The algorithm is based on cost priority decoding for path growth procedure i.e. explained in the flow chart given below. The path is initialized by source node 1, and count is set to 0. As count increases, the cost priority multiplication is measured for each node directly linked to source node. The node having the minimum value is selected and then this is considered as current node $i$ and now again by repeating above steps, we find the next node $j$ to this current node $i$ and thus go on until we reach the destination.

---

Shortest Path Optimization Algorithms in Federated Cloud-Based Wireless Sensor Networks

44
C. Hybrid particle swarm optimization algorithm

Since PSO is more accurate and fast than GA, so combination of these two algorithms proved to be the most efficient technique in [16]. The particle swarm optimization algorithm is applicable to continuous optimizations. For the problem of route selection, the different routes are independent, so the routing problem is a discrete optimization. So we cannot directly do arithmetic operations on routes using PSO. Facing the problems, we propose a hybrid particle swarm optimization (HPSO). The core of this algorithm is an equivalent form of velocity and displacement formulae combining the thought of genetic algorithm.

V. EXPERIMENTAL RESULT AND COMPARISON OF ALGORITHMS

A. Genetic algorithm

Implementation of EIGA, the SGA, and the Restart GA for the dynamic SP problem has been done in [8]. The initial network topology consist of a square region with the area of 200×200. Then 100 nodes are generated and the position (x, y) of each node is randomly specified within the square area. If the distance between two nodes falls into the radio transmission range D, a link will be added to connect them. Finally, we check if the generated topology is connected. If not, the above process is repeated until a connected topology is generated. In the experiments, D is given a reasonable value 50. All the algorithms start from the initial network topology. Then after a certain number (saying, R) of generations, a certain number (saying, M) of nodes are scheduled to sleep or wake up depending on their current status. Therefore, the network topology is changed accordingly. R and M determine the change frequency and severity, respectively. The larger the value of R, the slower the changes.

The larger the value of M, the more severe the changes. The value of M is set to be 2. In all the experiments, the mutation probability is set to 0.1. For the elitism based immigrants scheme, rei is set to 0.2 and pm is set to 0.8. In addition, the number of changes is set to be 10. The delay upper bound Δ is set to be 2 times of the minimum end-to-end delay. The value of R is changed from 5 to 10, respectively to see the impact of change frequency on the performance. Fig. 5 shows a rapidly changing environment. In all these settings, we can see that both EIGA and SGA experience more significant changes in subfig (b) than in subfig (a). The reason is that when more nodes are rescheduled, the changes to the initial network topology become more drastic. Also EIGA bring more diversity to the population in EIGA and therefore enhance its search capability than SGA. However, the Restart GA exhibits the worst performance because it does not exploit any useful information in the old environment and that the frequent restart sacrifices its evolving capability.

![Fig 5: Comparison of EIGA, SGA, and Restart GA when Z is 20 and R is 5 : (a) generation 0 to 24; (b) generation 25 to 54.](image)
B. Particle swarm optimization algorithm for shortest path problem

Random topologies of networks are generated in [7] with edge or link cost of range [1, 1000]. The other PSO parameters are chosen as: Population size = 50, maximum number of iterations are 100; neighbor topology = Ring; \(\Pi_l = 2.0; \Pi_2\) is chosen to be 2.2, maximum velocity is \(\pm 1.0\); the success rate is average number of time over maximum number of runs for which shortest path is reached. This algorithm is compared with dijkastra’s algorithm in this research. The tested rate is very high for this algorithm. But this algorithm is suitable for static shortest path problems but for dynamic we have to do further over it. HPSO algorithm is best suited for it which is explained next.

C. Hybrid particle swarm optimization algorithm

The experimental results shown in [9] show the accuracy of this system. The various parameters for experiment is as shown in table 1. Under different moving velocity, the residence time is fixed on 30s to make simulations. The experimental result is shown in Fig. 6. From Fig.6, we see that in most cases the average end to end (ETE) delay of HPSO is lower than that of AODV at the same node moving velocity.

CONCLUSION

Routing problems can be solved by lots of algorithms in fixed networks. But when it comes to dynamic networks like federated cloud-based wireless sensor networks, it is very challenging. In recent years, there has been a growing interest in studying bio inspired algorithms to solve shortest path problem. So it has good scope in future. PSO is a very simple algorithm and applied in many fields. So it has high efficiency but GA also has good features like as it maintains population diversity. Both these algorithms are combined have been combined and these show better results applicable in the federated cloud-based environment. An implementation in real scenario is considered as the very future opportunity of the present research work.

Table 1: parameters used in simulation

<table>
<thead>
<tr>
<th>Name of Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area of Simulation</td>
<td>2000m x 2000m</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>260s</td>
</tr>
<tr>
<td>Amount of Nodes</td>
<td>140</td>
</tr>
<tr>
<td>Moving velocity of nodes</td>
<td>5, 10, 15, 20, 25, 30, 35 (m/s)</td>
</tr>
<tr>
<td>Time of residence for nodes</td>
<td>5, 10, 20, 40, 60, 80, 100 (s)</td>
</tr>
<tr>
<td>Maximum communication distance</td>
<td>300m</td>
</tr>
<tr>
<td>Moveable Model</td>
<td>Random Waypoint Model(RWM)</td>
</tr>
</tbody>
</table>

Fig. 6: End to End delay for residence time 30 sec.

REFERENCES