Abstract
With the advancement in wireless communications, more and more wireless networks appear, e.g., Mobile Ad Hoc Network (MANET), Wireless Sensor Network (WSN), etc. Shortest path routing is very efficient as it saves time and economically beneficial in terms of cost. One of the most important characteristics in mobile wireless networks is the topology dynamics, that is, the network topology changes over time due to energy conservation or node mobility. 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 will discuss these algorithms on various wireless networks.

Keywords
Mobile Ad Hoc Networks (MANET), Wireless Sensor Network (WSN), Shortest Path (SP), Genetic Algorithms (GAs), Particle Swarm Optimization (PSO)

I. Introduction

A. Wireless Ad hoc network
[1,2] is a decentralized wireless network. The network is ad hoc because it does not rely on a preexisting infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data for other nodes, and so the determination of which nodes forward data is made dynamically based on the network connectivity. A mobile ad hoc network, also called a mobile mesh network, is a self-configuring network of mobile devices connected by wireless links. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. In MANETs, unicasting establishes a multi-hop forwarding path for two nodes beyond the direct wireless communication range. Routing protocols also maintain connectivity when links on these paths break due to effects such as node movement, battery drainage, radio propagation, or wireless interference. In multi-hop networks, routing is one of the most important issues that has significant impact on the network’s performance. So far, there are mainly two types of routing protocols, namely, topological routing and geographic routing. In topological routing, nodes utilize topological information to construct routing tables or search routes directly. In geographic routing, each node knows its own position and makes routing decisions based on the destinations position and its local neighbour’s positions.

B. Shortest Path Problem
The shortest path routing belongs to the topological routing. The shortest path problem 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 SP problem: the Dijkstra’s algorithm, the breadth-first search algorithm and the Bellman-Ford algorithm, etc. All these algorithms have polynomial time complexity. Therefore, they will be effective in fixed infrastructure wireless or wired networks. But, they exhibit unacceptably high computational complexity for real-time communications involving rapidly changing network topologies. [3,4] We have study bioinspired optimization algorithms for SPP like GA, PSO, and ACO etc. Biologically inspired algorithms are a category of algorithms that imitate the way nature performs. Numerous problems can be solved without rigorous mathematical approaches, more fastly reach a solution.

II. Literature Survey
A few research work have been conducted to solve the routing problems using artificial intelligence techniques, e.g., ANNs [5], GAs [6], and PSO [7]. In [5], 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, they will regard it as a new network and restart the algorithms over the new topology. As is well known that the topology changes rapidly in MANETs due to the characteristics of wireless networks, e.g., battery exhaustion, node mobility. Therefore, for the dynamic SP problem in MANETs, these algorithms are not good choices 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 Ad Hoc routing protocol based on HPSO.

III. Model Of Network
Let us consider a ad hoc network operating within a fixed geographical region. We model it by a undirected and connected topology graph $G_0(V_0, E_0)$, where $V_0$ represents the set of wireless nodes and $E_0$ represents the set of communication links connecting two neighboring routers falling into the radio transmission range.

- $G_0(V_0, E_0)$, the initial MANET topology graph.
- $G_i(V_i, E_i)$, the MANET topology graph after the $i$th change.
- $s$, the source node, $r$ is the destination node.
- $P_i(s, r)$, a path from $s$ to $r$ on the graph $G_i$.
- $d_i$, the transmission delay on the communication link $l$.
- $c_l$, the cost on the communication link $l$.
- $\Delta(P_i)$, the total transmission delay on the path $P_i$.
- $C(P_i)$, the total cost of the path $P_i$.

Here we wish to find a delay-bounded least cost loop-free path on the topology graph. In mobile networks, the topology changes from time to time. The objective of our problem is to quickly find the new optimal delay-constrained least cost acyclic path after each topology change. Firstly, a series of paths $\{P_i| i \Delta \{0, 1, ...\}\}$ over a series of graphs $\{G_i| i \epsilon \{0, 1, ...\}\}$ are found which satisfy the delay constraint as shown in (1) and have the least path cost as shown in (2).

\[
\Delta(P_i) = \sum_{l \in P_i(s, r)} d_l \leq \Delta. \tag{1}
\]

\[
C(P_i) = \min_{F \in G_i} \left( \sum_{l \in P_i(s, r)} c_l \right). \tag{2}
\]

IV. Optimization Tech-Niques 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 MANETS.

1. GA for static shortest path problem
2. Modified GA for dynamic shortest path problem

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 pseudocode for EIGA is as follows:

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

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 \( rei \times n \) individuals are iteratively generated by mutating \( E(t-1) \) with a probability \( pim \), 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[14]. 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:

\[
\begin{align*}
\nu_{k+1} &= c_0 \nu_k + c_1 (p_{best_k} - x_k) + c_2 (g_{best_k} - x_k) \\
x_{k+1} &= x_k + \nu_{k+1}
\end{align*}
\]

\( k \) represents the iterative number; \( \nu_k \) is the velocity vector of a particle; \( x_k \) is the current position of a particle; \( p_{best_k} \) is the
position of optimization that a particle has found; gbest<sub>k</sub> is the position of optimization that the whole swarm has found; c<sub>0</sub>, c<sub>1</sub> and c<sub>2</sub> 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 (pbest), make the value of pbest equal to the new value. Otherwise, keep pbest unchangeable. Then choose the optimal one from local extrema of all particles and take it as the global extremum (gbest).

4. Check gbest or the iteration number. If the stop condition is satisfied, output gbest and the optimal value of the target function. Otherwise, return to step 2. 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 (β<sub>j</sub>) and the edge cost is minimum. 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

\[
 j = \min \{wij \beta j | (i, j) \in E \}, \beta j \in [-1.0,1.0]
\]

where j is the next node, wij is the cost of path between this node and the current node, β<sub>j</sub> 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.

![Fig 4: the flow of path decoding procedure with PSO using SPP.](image)

C. **Hybrid particle swarm optimization algorithm**

Since PSO is more accurate and fast than GA, so combination of these two algorithm 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.

- **Design of Coding for Particles**

For the routing problem, the particles in PSO denote the routes and every particle is a set of order numbers of nodes that a route passes. For example, the route (1, 3, 5, 7, 11, 23, 6, 9, 10, 17) is considered as a particle.

- **The Equivalence of c<sub>2</sub> (gbest<sub>k</sub> - x<sub>k</sub>) and c<sub>1</sub> (pbest<sub>k</sub> - x<sub>k</sub>) of velocity update formula**

In the velocity formula of PSO, the subtracting indicates the tendency that a particle is close to the extremum. The moving tendency is similar to the operation of crossing in genetic algorithm (GA). So crossing is considered as the equivalent form of the subtracting items in velocity formula. The item c<sub>2</sub>(gbest<sub>k</sub> - x<sub>k</sub>) indicates that a particle tends to the global extremum, so it is equivalent to the crossing operator of dealing with common items and gbest. Example:

\[
 gbest = (1 3 7 5 10 15 2 4 6 8),
 x = (1 2 5 6 11 13 3 9 10 8).
\]

After crossing, (1 376 11 13 3 9 10 8).

The equivalence above represents that ordinary routes approach toward the global optimal route. The item c<sub>1</sub> (pbest<sub>k</sub> - x<sub>k</sub>) indicates that a particle tends to the...
local extremum, so it is equivalent to the crossing operator between common items and pbest. In order to differentiate it, we replace the 4th and 5th items of x with the corresponding items of pbest. Example:
pbest=(1 376 1113 159 10 8),
x=(12 5 12 1459 1548)
After crossing, (1 2 5 6 11 5 9 15 4 8), which represents that ordinary routes approach toward the local optimal routes.

• The Equivalence of COVk of velocity updated formula
In the velocity formula of PSO, multiplying Co by vk indicates that a particle searches toward new space. This moving tendency is similar to the operation of mutation in genetic algorithm (GA) that have large changeable scales and search toward wider space. So mutation is used as the equivalent form of the multiplying items. Example:
x= (13711613159108)
After mutation, (1 13 3 7 11 6 15 9 10 8).
This equivalence represents that an ordinary route searches toward new space in order to find a new route.

• The Equivalence of The Addition Operation
The addition operation indicates the sum of several processing steps. Through executing orderly the equivalent sub-items in PSO, we can obtain the effect of addition. In the equivalence of the addition the outputs of previous steps are the inputs of latter steps.

\[ x_{k+l} = x_k + v_{k+l} = x_k + C_1 \cdot v_{k+l} + C_2 \cdot (p_{best_k} - x_k) + C_2 \cdot (g_{best_k} - x_k) \]
The computation sequence is from right to left. Thus by means of the equivalence of particle coding, velocity and displacement, the hybrid particle swarm optimization (HPSO) can be used to solve the routing optimization problem. The algorithm steps of HPSO are the same as that of PSO.

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 pim is set to 0.8. In addition, the number of changes is set to be 10. The delay upper bound \( \Delta \) 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.

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 ; neighbour topology = Ring ; \( \alpha_1 = 2.0 ; \alpha_2 \) is chosen to be 2.2, maximum velocity is \( =+1.0 ; \) the success rate is average number of time over maximum number of runs for which shortest path is reached. This algorithm is compared with dijkastara’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] shows 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.

VI. Conclusion

We have lot of algos to solve routing problem in fixed networks but for dynamic networks like MANETS, its very challenging. In recent years, there has been a growing interest in studying bioinspired algos 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 it maintains population diversity. Both these algorithms are combined in [9] and these show better results.

Table 1: parameters used in simulation

<table>
<thead>
<tr>
<th>Name of parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area of simulation</td>
<td>2000m x 1000m</td>
</tr>
<tr>
<td>Simulation time</td>
<td>120 s</td>
</tr>
<tr>
<td>Amount of nodes</td>
<td>120</td>
</tr>
<tr>
<td>Moving velocity of node</td>
<td>5, 10, 15, 20, 25, 30, 35 (mph)</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>300 m</td>
</tr>
<tr>
<td>Movement model</td>
<td>Random Waypoint Model (RWM)</td>
</tr>
</tbody>
</table>

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

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 it maintains population diversity. Both these algorithms are combined in [9] and these show better results.

References


Kulvir Kaur is pursuing her M.Tech in Electronics and Communication Engineering, Punjabi University, Patiala, Punjab, India.

Sonia Goyal is currently working as an Assistent Professor in Department of Electronics and Communication Engineering, Punjabi University, Patiala, Punjab, India.