International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2015): 6.391

Network Route Optimization Using Particle Swarm Intelligence Algorithm

Lebeta Belachew Abdissa[1], Professor Zheng Xiao Yan[2]

[1]Tianjin University of Technology and Education, School of Information Technology and Engineering,
1310 # Dagunan road, Hexi District, Tianjin 300222, PRC, China
esa_abdi[at]yahoo.com

[2]Tianjin University of Technology and Education, School of Information Technology and Engineering,
1310 # Dagunan road, Hexi District, Tianjin 300222, PRC, China
Zhengxycn[at]163.com

Abstract: ***This paper presents a hybrid algorithm based on particle swarm optimization (PSO) intelligence algorithm and a Tabu search meta-heuristics algorithm for efficient network route optimization. This hybrid search process combines particle swarm optimization (PSO) for iteratively computing a population of better solutions and Tabu search method for diversifying the local search scheme to solve this problem. A priority based indirect encoding and decoding scheme based on heuristics has been used for representing the shortest path problem parameters as a particle in PSO. Tabu search based meta-heuristics have been integrated in order to enhance the overall search efficiency. Specifically, an iteration of the proposed hybrid algorithm consists of a standard PSO iteration and Tabu search based algorithm applied to each improved particle for local search, where the neighborhood of each such particle is explored with two neighborhood generating operations on particles in order to escape possible local minima and to diversify the search. Simulation results in several networks with random topologies are used to illustrate the efficiency of the proposed hybrid algorithm for the optimal route computation. The simulation result reveals that the proposed algorithm outperforms than the comparison algorithms used on result analysis.***

Keywords: Swarm Intelligence, Particle swarm optimization, Tabu search algorithm, Network routing optimization problem.

1. Introduction

Routing is one of the most important issues that have a significant impact on the performance the multi-hop networks, such as the Internet and the Mobile Ad-hoc Networks [2-4]. The best routing algorithm should endeavor to find an optimum path for packet transmission within a precise time, so as to assure the Quality of Service (QoS) [2-4]. Route optimization is a core concept of Internet traffic engineering, which encompasses all methodologies capable of providing Quality of Service (QoS) in IP networks [1].

Due to a wide spectrum of its applications [6-8], ranging from routing in communication networks to robot motion planning, scheduling, sequence alignment in molecular biology, and length-limited Huffman coding shortest path computation is one of the most significant problems in a graph theory. Besides, the shortest-path problem also has many variations such as the minimum weight problem, the quickest path problem.

The shortest path problem has been scrutinized by numerous researchers. With the advancement of a communication, Computer Science, and transportation system, more divergence of the shortest path algorithm has appeared. Some of these include TSP (Travel Sales Man Problem), K-shortest paths, CSP (Constrained Shortest path problem, Multi-objective shortest path problem, the network flow problem to name a few of them. Most of these problems are NP-hard problems. Hence, several search algorithms are investigated for the shortest path problems.

A large number of algorithms for solving shortest path problems and applications of the shortest path problems were reviewed by Deo and Pang [9]. From the most commonly known non-evolutionary algorithms, the Dijkstra's, Bellman ford and bread-first search algorithms were the popular ones. Since these algorithms can resolve the shortest path problem in polynomial time, they will be valuable in fixed infrastructure wireless or wired networks.

Conversely, they demonstrate the unacceptably high computational complexity of real-time communications involving rapidly changing network topologies [3-4]. For example, in communication networks like IP, ATM, and Optical networks, there is a need to find a path with minimum cost while maintaining a bound on delay to support a QoS application. This problem is known to be an NP-hard problem [10]. Multiple edge weights and weight limits may be defined, and the general problem is called the constrained shortest-path problem.

In another case, it is required to find the shortest path such that cost or delay paths are minimized, and quality and bandwidth are to be minimized. These types of the shortest path are referred to as multi-criteria or multi-objective shortest path which are also NP-hard problems [10]. If we consider in case of mobile Ad-hoc networks, since all the nodes considerately maintain network connectivity without the help of any fixed infrastructure networks, dynamic changes in the network topology are to be expected.

An optimal shortest path has to be computed within a very microsecond in order to support time-constrained services such as voice, video, and teleconferencing [3-5]. The above pointed out algorithms do not assure this (real-time) requirement.

Some form of shortest path computation was employed by routing algorithms in the current packet-switching networks at the network layer [3- 5]. Particularly, the network links are weighted, the weights reflecting the link transmission capacity, the congestion of networks and an anticipated transmission status, such as the queuing delay head-of-line (HOL) packet or the link failures.

The shortest path problem can be formulated as one of finding a minimum cost path that contains the designated source and destination nodes of the network. In other words, the shortest path routing problem involves a classical combinatorial optimization problem arising in many designs and planning contexts [2], [11]. Hence, modern optimization schemes like Artificial Neural Networks [2-4] and other evolutionary algorithms guarantee a superior solution to such complicated problems, and they have been used in various practical applications.

2. Review of Shortest Path Algorithms

It was investigated in [3-5] that artificial neural networks (ANN) can solve the shortest path problem using their parallel and distributed architectures to provide a fast solution. Nevertheless, the complexity of the hardware with increasing number of network nodes in this algorithm was pointed out as its shortcoming, that means, ANN hardware size cannot accommodate networks of arbitrary size because of its physical limitation; correspondingly, the trustworthiness of the solution decreases, and those ANN algorithms for Shortest path problem were also less flexible to topological changes in the network graph [12].

Further ANN for the shortest path problem does not consider suboptimal paths and the quality of the solution (computed path) returned by the ANNs is constrained by their inherent characteristics. The other alternative algorithms developed to solve the shortest path problems where the evolutionary algorithms (e.g. Genetic algorithm) and meta-heuristic algorithms (e.g. Tabu Search, Simulated Annealing).

It was investigated to apply the Genetic algorithm to the shortest path problem [11], multicast routing problem [13], ATM bandwidth allocation problem [14], capacity and flow assignment problem [15], and dynamic routing problem [16]. It was reported that all of these problems can be formulated as some sort of combinatorial problem.

Among the recently reported genetic algorithms [33], [41] for shortest path problem was indicated a promising result and show an improved performance compared to those of ANN approaches and prevail over the limitations of ANN explained above. The achievement of the evolutionary programming methods promptly persuades analytical studies to use other powerful evolutionary algorithms or other similar intelligence algorithms to solve shortest path problem. In this paper, we have used a PSO swarm intelligence algorithm and tabu search scheme for the network route optimization.

The intention of this paper is to study the performance and efficiency of PSO for network route optimization problem in finding an optimal route path. In this regard, this paper reports the use of PSO scheme to solve the shortest-path problem, where a priority based indirect encoding/decoding system is used to represent the particle (position). Furthermore, in order to diversify the local search, a Tabu search [39-40] meta-heuristic algorithm has been integrated into the main PSO.

The main idea behind the use of the Tabu Search is to apply a diversification scheme to search another local optimum solution from which has been acquired previously from the PSO iteration as a better solution. It was noted that [19] PSO algorithm is effective in obtaining a population of the local optimal solution in which one of those solutions might be the required global solution. Hence, hybridization of both algorithms makes the use of the best features of both of PSO and Tabu search algorithm in the network route optimization problem.

The target result of the proposed algorithm has been tested in simulation experiment on various random network topologies and compared with a PSO (standard PSO and PSO with velocity re-initialization) algorithm and benchmark algorithm (Dijkstra's) used for the study in terms of quality of solution (route optimality) and convergence rate. The comparison of the analyzed result shows the effectiveness of the proposed hybrid PSO approach over the other comparison algorithms used in result analysis.

The rest of the paper is organized as follows. In section 3, a brief description of PSO and Tabu Search algorithm is presented. In section 4, the local search process using the Tabu search and hybrid PSO and Tabu search is explained. The simulation result analysis is discussed in section 5 and the last section concludes the paper.

3. Particle Swarm Optimization

Particle swarm optimization (PSO) is an evolutionary computation technique motivated by the simulation of the social behavior of the flock of birds and fishes school. PSO was developed by Kennedy and Eberhart (Kennedy and Eberhart 1995; Eberhart, Simpson, and Dobbins 1996). PSO is similar to a genetic algorithm (GA) in that the system is initialized with a population of random solutions. Conversely, it is unlike a genetic algorithm, in that each potential solution is also assigned a randomized velocity, and the potential solutions, called *particles*, are then "flown" through the problem space. The algorithm, which is based on a metaphor of social interaction, searches a space by adjusting the trajectories of individual vectors, "particles" as they are conceptualized as moving points in multidimensional space.

The individual particles are drawn stochastically toward the positions of their own previous best performance and the best previous performance of their neighbors. Each individual in PSO flies in the search space with a velocity which is dynamically adjusted according to its own flying experience and its companions' flying experience. Each individual is treated as a volume-less particle (a point) in the D-dimensional search space. The ith particle is represented as Xi = (xi1, xi2… XiD). The best previous position (the position giving the best fitness value) of the ith particle is recorded and represented as Pi = (Pi1, Pi2… PiD).

The index of the best particle among all the particles in the population is represented by the symbol g. The rate of the position change (velocity) for particle *i* is represented as Vi = (vi1, vi2, viD). The search for an optimal position (solution) is performed by updating the particle velocity, hence positions, in each iteration/generation in a specific manner as follows.

In every iteration, the fitness of each particle is determined by some fitness measure and the velocity of each particle is updated by keeping track of the two "best" positions, that is , the first one is the best position (solution) a particle has traversed so far, called pBest , and the other "best" value is the position that any neighbor of the particle has traversed so far, called neighborhood, best (nBest).when a particle takes the whole population of its neighborhood, the neighborhood best becomes the global best and accordingly called gBest. The particles are updated according to the following equations:

$$Vid=Vid+\varphi 1*rand()*(Pid-Xid)+\varphi 2*Rand()*(Pgb-Xid), \quad i=1,2..,Ns,d=1,2,\dots,D \qquad (3.1)$$

$$Xid=Xid+Vid \qquad (3.2)$$

Where φ1 and φ2 are two positive constants and called acceleration coefficients that are used to control the influence of *pBest* and *nBest* on the search process. In all initial studies of the PSO, both φ1& φ2 are taken to be [23], the Ns, is the total number of particles in the swarm and Rand (), rand () are two random functions in the range of [0, 1]. D is the dimension of the search space.

When the convergence criterion met, a particle whose best position is stored in *Xbest* and best fitness *fbest* is taken as the near optimal solution to the problem. In most of the cases, particularly for particles that are far from their global best, the velocities quickly reach large values, hence, the velocity clamping is used to control the change in velocity. Accordingly, to prevent the explosion a parameter *Vmax* is defined and prevents the velocity from exceeding it on each dimension d for individual *i*.

Unlike in genetic algorithms, evolutionary programming, and evolution strategies, in PSO, the selection operation is not performed [27, 30]. All particles in PSO are kept as members of the population through the course of the run (a run is defined as the total number of generations of the evolutionary algorithms prior to termination) [27]. It is the velocity of the particle which is updated according to its own previous best position and the previous best position of its companions as per equation (3.1). The particles fly with updated velocities.

PSO is the only evolutionary algorithm that does not implement survival of the fittest [20]. By considering equation (3.1b) as similar to mutation operation, the PSO algorithm is similar to the evolutionary programming algorithm since neither of the algorithms performs a crossover operation. In evolutionary programming, each individual is mutated by adding a random function (the most commonly used random function is either a Gaussian or Cauchy function) [23, 24], whereas, in PSO, each particle (an individual) is updated according to its own flying experience and the group's flying experience.

In other words, at each generation, each particle in PSO can only fly in a limited number of directions which are expected to be good areas to fly toward according to the group's experience; while in evolutionary programming each individual has the possibility to "fly" in any direction.

That is to say, PSO performs a kind of "mutation" operation with "conscience" [25]. PSO has more chance to 'fly" into better solution areas more quickly when the "conscience" provides helpful information. In PSO, a parameter called inertia weight is brought in for balancing the global and local search and equation (3.1) is changed to

$$Vid = w*Vid+\varphi 1*rand () *(Pid - Xid) + \varphi 2* Rand () * (Pgb - Xid), \quad i = 1,2..,Ns,d=1,2,\dots,D \qquad (3.1a)$$

$$Xid = Xid+Vid \qquad (3.2b)$$

Where w is the inertia weight [33]. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search. By linearly decreasing the inertia weight from a relatively large value to a small value through the course of the PSO run, the PSO tends to have more global search ability at the beginning of the run while having more local search ability near the end of the run.

As from [26], Clerc's generalized constriction model suggests that the inertia weighted particle swarm will eventually converge when $0 < \varphi <= 2\alpha + 2 - \varepsilon$ and $0 <= \alpha < 1$. Using these convergence criteria, it is possible to parameterize the inertia – weighted particle swarm so that no **Vmax** is necessary for convergence, though it may still turn out to be useful as a problem – solving heuristic.

Eberhart and Shi usually implement the inertia weight so that it decreases over time, typically from approximately 0.9 to 0.4, with $\varphi = 4.0$.

The common PSO is either global version or local version of PSO. In global version, all other particles influence the velocity of a particle, while in the local version of PSO, a selected number of the neighbor particles affect the particle's velocity.

In [27], PSO is tested with regular – shaped neighborhoods, such as global version, local version, pyramid structure, ring structure, and von Neumann topology. The neighborhood topology of the particle swarm has a significant effect on its ability to find an optimal solution. For enhanced performance the following adjustment was made on the basic PSO algorithm:

Constriction Factor Method (CFM):

The idea behind the use of construction factor is to prevent the velocity explosion, thus the velocity clamping is not required. The Clerc's [26], simplest constriction coefficient, called "Type 1", requires the application of coefficient to both terms of the velocity formula. This CFM is described as:

Vid = χ[Vid+φ1*rand () *(Pid - Xid) + φ1* Rand () * (Pgb - Xid)] , i = 1, 2..,Ns,d=1,2,…,D (3.1c)

Xid = Xid+Vid (3.2c)

The simplest formula to compute the constriction coefficient is

$$\chi = \frac{2k}{(2 - \varphi - \sqrt{\varphi 2 + 4\varphi}\ )} \quad (3.3)$$

As described in [37] the variable k can range in [0, 1]; a value of 1.0 works fine, as does a value of φ = 4.1. Thus, if φ=4.1 and k = 1, X=0.73, simultaneously damping the previous velocity term and the random φ. The “type 1” constriction coefficient is not defined for φ<=4.0. As φ increases above 4.0, X gets smaller. However, it was reported in [28] that the best performance can be obtained by using both CFM and velocity clamping. In this paper, a PSO with construction factor method is used for the network route optimization problem.

Velocity re-initialization*:* PSO has a problem of premature convergence to a local minimum. It does not continue to improve the quality of the solutions after certain numbers of iterations have passed [29]. Because of this, the swarm becomes stagnated after a certain number of iterations and may end up with the solution far from optimality. Gregarious PSO [22] avoids premature convergence of the swarm; the particles are reinitialized with a random velocity when stuck at local-minimum. Dissipative PSO [31] reinitialize the particle positions at each iteration with a small probability. This additional perturbation is carried out with different probabilities based on time-dependent energy [32].

4. Shortest path computation By PSO algorithm and Particle Encoding

The shortest path problem is defined as follows. A directed graph G = (N, A), where N is a set of N nodes (vertices), and A is a set of its links (arcs or edges) [2-5].There is a cost C_{ij} associated with each link (i, j). The costs are specified by the cost matrix C = [C_{ij}], where C_{ij} denotes a cost of transmitting a packet on the link (i, j). Source and destination nodes are denoted by S and D, respectively. Each link has the link connection indicator denoted by I_{ij}, which plays the role of mapping, providing information on whether the link from node i to node j is included in a routing path or not. It can be defined as follows:

$$I_{ij} = \begin{cases} 1, \text{ if the link from the node i to node j Exists} \\ 0, \text{ otherwise} \end{cases} \quad (4.1)$$

It is clear that the diagonal elements of I_{ij} must be zero. Using the above definition, the shortest path routing problem can be formulated as a combinatorial optimization problem minimizing the objective function (4.2) as follows:

Minimize

$$\sum_{i=S}^{D} \sum_{\substack{j=S \\ j \neq i}}^{D} Cij.Iij \quad (4.2)$$

Subject to

$$\sum_{\substack{j=S \\ j \neq i}}^{D} Iij - \sum_{\substack{j=S \\ j \neq i}}^{D} Lji = \begin{cases} 1, & if\ i = S \\ -1, & if\ i = D \\ 0, & otherwise \end{cases}$$

And

$$\sum_{\substack{j=S \\ j \neq i}}^{D} Iij \begin{cases} \leq 1, if\ i \neq D \\ = 0, if\ i = D \end{cases}$$

And

$I_{ij} \in \{0, 1\}$, for all i. (4.3)

The constraint (4.3) ensures that the computed result is indeed a path (without loops) between a source and a destination.

4.1 Particle Encoding

How to encode a path in a network into a particle in PSO or chromosome in GA is the thorniest problem in applying PSO or (GA) to compute an optimal path in route optimization. This encoding, in turn, affects the effectiveness of the search process. The proposed path encoding algorithm for PSO basically depend on indirect priority based encoding. The direct encoding scheme is not preferred since a random sequence of nodes is definitely not a good choice for path construction and arithmetic operations for updating the particles position and velocity.

In the proposed system, the particle encoding is based on node priorities and decoding is based on path growth procedure taking into account the node priorities. The particles include a vector of node priority values (particle length = number of nodes). To construct a path from a particle initial node (source node) to the designated destination node, the edges are appended to the path consecutively.

Let $\mathbf{N_{max}}$ be the maximum number of nodes in the network topology and ROUTE (i) be a partial path analogous to position / priority vector of a particle under construction which contains i+1 nodes with a terminal node N (i = 0, related to the partial path with the source node, {So}) and at each step, the next node (node Sj) is to be selected from the set of adjacent nodes Adj (S0) having direct links with the current node such that its corresponding priority (PSj) is maximum. Suppose that X is a dynamic priority vector, which initially contains the priority values (position vector of the particle) indicated to be P. This node priority can take negative or positive real numbers.

Each time the next node is appended to the ROUTE (i) undergrowth, its equivalent position in X is given a value of large negative number (-N∞). The terminal node N_1 is considered to be the source node ID (So) and $\mathbf{N_{max}}$ is assigned to the destination node ID. If the number of iterations exceeds $\mathbf{N_{max}}$ (the maximum number of nodes), it would mean either a valid path has not been found due to loops or the path does not terminate at the destination node in N_{max} steps. In that case, the objective function evaluation of the corresponding particle is made to return a very low value as a penalty. Path construction often leads to the formation of loops; hence to avoid this, the selected nodes are assigned a very large negative value for their priority. In order to avoid backtracking, a heuristic operator ***M*** is used. In this paper, the value of M is considered as a constant value 4.

4.2 Fitness Function Evaluation

The quality of a particle (solution) is measured by a fitness function. Here, the fitness function is noticeable as the goal is to find the minimal cost path. Thus, the fitness of i*th* particle is defined as:

$$f_i = \frac{1}{\sum_{j=1}^{Ni-1} Cyz}, \quad y = PPi(j), \quad z = PPi(j+1) \quad (4.4)$$

Where PPi is the set of sequential node IDs for the ith particle, Ni equal to the number of nodes that constitute the path represented by ith particle, and Cyz is the cost of the link connecting node y and node z. Accordingly, the fitness function takes maximum value when the shortest path is obtained. If the path represented by a particle happens to be an invalid path, then its fitness is assigned a penalty value so that the particle's attributes will not be considered by others for future search.

4.3. Local search diversification using Tabu Search

Evolutionary optimization algorithms are very effective and commanding global optimization method for tackling large-scale optimization problems with many local optima; nevertheless, they utilize high execution times and are generally deprived in terms of convergence performance. On the other side, local search algorithms can converge in a few iterations but lack global perception. The combinations of global and local search schemes, offer the benefit of both optimization procedures while counterbalances their drawbacks [37]. From this point of view, the PSO algorithm does not suitably handle the relationship between exploitation (local search) and exploration (global search), so it usually converges to a local minimum quickly.

Consequently, it was suggested to hybridize PSO with some traditional and evolutionary optimization algorithms in order to take the advantages of both Methods and compensate the limitation of each other, this type of PSO is called hybridized PSO. Tabu Search algorithm, which was first proposed by Fred Glover [39-40], is based on using the mechanisms that are enthused by the human memory. The hybridization of PSO and Tabu Search algorithms has been shown to be faster and more promising in solving most problems. Based on this, in this paper, we have proposed a hybridized PSO – Tabu Search algorithm, PSO scheme for global search and meta-heuristic Tabu search algorithm to diversify the local search for the network route optimization problem.

Two neighborhood solution generating move operations was applied to generate a neighborhood solution for the current solution or initial solution transferred from PSO and one repair function has been incorporated to verify the feasibility of solution in the local search diversification scheme. The enthused Tabu search algorithm for local search diversification is described as the under mentioned procedures 1) Accepting initial solution from PSO algorithm 2) applying neighborhood solution generating operations and selecting the next solution (a neighborhood solution with minimum cost) 3) comparing the fitness value best next neighborhood solution with an initial solution and better solution found so far. 4) Repeating the second step until the termination criterion is met.

The algorithm transfers the particle that has experienced an enhancement in PSO to the Tabu Search method. The Tabu Search scheme will take this particle as an initial solution for a further search around it. If the Tabu search is able to find a solution better than the original particle, then the particle will be updated and returned. Also, this new solution is compared with the best solution found so far by that particle; if it is better, then it will also be updated for reflecting the new found solution back on the swarm.

The termination criteria for the local search diversification is the Kmax (if the current iteration is greater than the maximum Tabu search iteration) or if there is no better solution obtained in comparison with the better solution obtained so far.

5. Computer Simulation Results and Discussion

The proposed hybrid algorithm for network route optimization is appraised on networks with fixed and random with varying network topologies generated using Waxman model [46] in which nodes are generated randomly on two dimensional planes of size 100x100, and there is a link between two nodes u and v with probability $P(u, v) = \alpha \cdot e^{-d(u, v) / (\beta L)}$, where $0 < \alpha, \beta \leq 1$, d(u, v) is a Euclidean distance between u and v, and L is the maximum distance between any two nodes, using a Python language on an Intel ® Core ™ i7-5500U CPU @ 2.4 GHz and RAM 6.0 GB. The edge costs of the networks are randomly generated in the interval [5, 500].The results of the proposed algorithm are also compared with other versions of PSO (Standard PSO with CFM and PSO with velocity re-initialization) and Dijkstra's algorithm [2] that is used as a benchmark for comparison purpose. In all simulation tests, the optimal solution obtained using proposed PSO reveals the superiority of the hybrid PSO algorithm.

5.1 Parameter Selection for the simulation of the algorithm

The parameters of the algorithms that are used in the experiment are discussed as follows:

1) *Population size*: In summary, an enhanced search performance is acquired in any evolutionary search algorithms using reasonably large population size. Conversely, as the number of population size increased, it increases the cost of fitness function evaluations. In [43], it is stated that a population size of 30 is a reasonably good choice; it is small enough to be efficient, yet large enough to produce reliable results.
2) *Particle initialization*: the particle position (node priorities) and velocity are initialized with random real numbers in the range [-1.0, 1.0]. The maximum velocity is set as ±1.0.

3) *Neighborhood Topology*: For PSO a Ring neighborhood topology [42] is used to avoid premature convergence and allow each particle to be connected to its neighbors.
4) Constriction Factor: as reported in [46] A reasonable compromise for the cognitive and social component values appear to be $\varphi_1 = 2.8$ and $\varphi 2 = 1.3$, respectively. In, [26], it is shown that the CFM has linear convergence for $\varphi > 4$. The experimental result from [46] shows the value $\varphi = 4.1$ are better in the general case. Therefore, from equation (3.3), $\chi = 0.73$.
5) Tabu search MAX iteration K_{max}=50/100/200

5.2. Performance Analysis Based on Random Network Size

In this simulation test, the route optimality and the convergence speed of the algorithms were studied. The route optimality or quality of the solution is defined as the percentage of time that the algorithm finds a global optimum shortest path). The route failure ratio is the contrary of route optimality i.e. it is asymptotically the probability that the computed route is not optimal because it is a relative frequency of route failure. For this experiment, a total of 25 random network topologies with random link cost are examined.

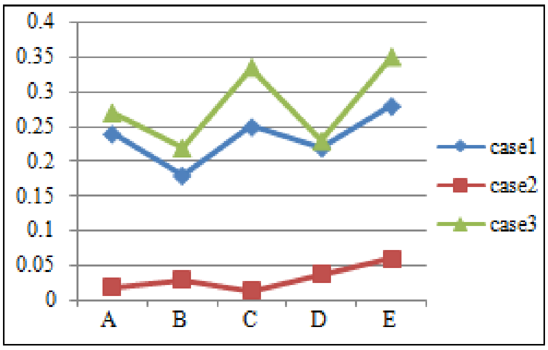
From the randomly generated networks, the records of five networks were considered as specified in Table 1 to illustrate the result analysis

Table 1: list of random networks used in simulation test

Problem	Number of nodes	Number of links /edges
A	50	60
B	70	221
C	80	255
D	90	227
E	100	255

Based on aforementioned problems route optimality and average fitness function evaluation result, the performance analysis of the following algorithms was assessed in terms of their route optimality or route quality and convergence characteristics.
Case 1. PSO with CFM and velocity Re-initialization
Case 2. PSO with CFM and Tabu Search (proposed)
Case 3. PSO with CFM only
First the route optimality (quality of solution) was examined as depicted in fig 5.1.

Route failure ratio



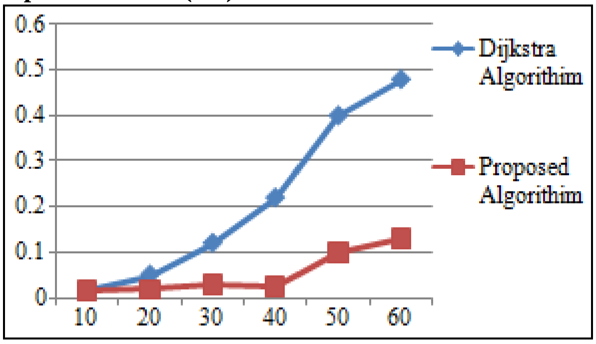
Problems

Figure 5.1 Comparison of quality of solution for each case.

The quality of the solution of the PSO algorithms is compared in Fig 5.1. From the figure, we can see that the quality of the solution of the proposed hybrid PSO (case 2) is much higher than the other comparison algorithms. For example, on problem C, the proposed algorithm shows a better quality than case 1 and case with a probability of < 0.24 and < 0.33 respectively.

Next, the convergence speed of the proposed algorithm is contrasted with Dijkstra's considering their execution time in seconds and a variation in the number of nodes.

Computation time (sec)



Number of nodes

Figure 5.2 Convergence speed comparison between proposed PSO and Dijkstra's algorithm

From figure 5.2, we have examined that as the number of nodes in the network topology varies, the computation time of the proposed PSO algorithm shows a very small variation, whereas, the computation time considerably increased with the increased number of nodes in Dijkstra's algorithm. The average execution time of Dijkstra's algorithm is 0.213secs and that of the proposed algorithm is 0.0535 seconds.

6. Conclusions and Future Work

This paper presented PSO and a Tabu search hybrid algorithm for solving the shortest path problem in networks. A priority based indirect encoding and decoding procedures are applied in order to represent the particles in PSO, hence a computation and updating of a particle's position and velocity value was easily accomplished. Simulation studies reveal that the proposed hybrid algorithm generates a better quality of solutions compared with the other comparison algorithms. Furthermore, as the size of network significantly increases, the proposed hybrid algorithm shows a minor variation in route optimality value and convergence rate compared with the Dijkstra's algorithm and the other variants of the PSO algorithm used in the result analysis. Hence, hybridization of an evolutionary algorithm for global search and meta-heuristic schemes for local search diversification in order to solve complex combinatorial optimization problems can create a flexibility to exploit the merits of both algorithms and diminish their limitations. It is also believed that the combination of both of the algorithms is promising to solve other combinatorial problems. Consequently, we have planned to apply the hybridization of PSO and Tabu search algorithm to optimize other variant network route problems and related optimization problems.

References

[1] D.Awduche, J.Malcolm, J. Agogbua, M. O'dell, J.Mcmanus, "Requirements for traffic engineering over MPLS, "IETF RFC 2702, sept 1999
[2] W. Stalling, High-Speed Networks: TCP/IP and ATM Design Principles. Englewood Cliffs, NJ: Prentice-Hall, 1998.
[3] M. K. Ali and F. Kamoun, "Neural networks for shortest path computation and routing in computer networks," IEEE Trans. Neural Networks, vol. 4, pp. 941–954, Nov. 1993.
[4] D. C. Park and S. E. Choi, "A neural network based multi-destination routing algorithm for communication network," in Proc. Joint Conf. Neural Networks, 1998, pp. 1673–1678
[5] C. W. Ahn, R. S. Ramakrishna, C. G. Kang, and I. C. Choi, "Shortest path routing algorithm using hopfield neural network," Electron. Lett., vol. 37, no. 19, pp. 1176–1178, Sept. 2001.
[6] F.B. Zahn and C.E. Noon, "Shortest path algorithms: an evolution using real road networks, " Transportation Science , vol.32, no. 1, pp,65-73,1998.
[7] J.Moy, "Open shortest path first version 2. RFQ 1583," Internet Engineering Task Force, 1994 http://www.ietf.org/.
[8] G.Desaulniers and F.Soumis, "An efficient algorithm to find a shortest path for a car-like robot," IEEE transactions on Robotics and Automation, Vol. 11, no .6 pp. 819-828, 1995.
[9] N.Deo and C.Y.Pang, "shortest-path algorithms: taxonomy and annotation," Networks, vol.14, no. 2 pp. 275-323, 1984.
[10] M.R. Garey and D.S Johnson, Computer and Incredibility. A guide to the theory of NP-completeness, W.H. Freeman, San Francisco, Calif, USA, 1979.
[11] Y. Leung, G. Li, and Z. B. Xu, "A genetic algorithm for the multiple destination routing problems," IEEE Trans. Evol. Comput., vol. 2, pp. 150–161, Nov. 1998.
[12] F.Araujo, B.Ribeiro, and L.Rodrigues, "A neural network for shortest path computation," IEEE Transaction on Neural Networks, vol.12 no.5 pp.1067-1073,2001.
[13] Z. Xiawei, C. Changjia, and Z. Gang, "A genetic algorithm for multicasting routing problem," in Proc. Int. Conf. Communication Technology (WCC-ICCT 2000), 2000, pp. 1248–1253.
[14] H. Pan and I. Y. Wang, "The bandwidth allocation of ATM through genetic algorithm," in Proc. IEEE GLOBECOM'91, 1991, pp. 125–129.
[15] M. E. Mostafa and S. M. A. Eid, "A genetic algorithm for joint optimization of capacity and flow assignment in packet switched networks," in Proc. 17th National Radio Science Conf., 2000, pp. C5-1–C5-6.
[16] N. Shimamoto, A. Hiramatsu, and K. Yamasaki, "A dynamic routing control based on a genetic algorithm," in Proc. IEEE Int. Conf. Neural Networks, 1993, pp. 1123–1128.
[17] C. Blum and D. Merkle, editors. Swarm Intelligence: Introduction and Applications. Natural Computing. Springer Verlag, Berlin, Germany, 2008.
[18] S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards. Artificial intelligence: a modern approach, volume 74. Prentice Hall Englewood Cliffs, NJ, 1995.
[19] J.kennedy and R. C. Eberhart , "particle swarm optimization ,'' in the proceeding of IEEE international conference on neural network , vol. 4, pp. 1942-1948, perth, western Australia, November – December 1995.
[20] R. C. Eberhart and Y.shi, "comparison between genetic algorithms and particle swarm optimization," in the proceeding of the 7th International Conference on evolutionary Programming, pp.611-616, Springer, San Diego, Calif, USA, March 1998.
[21] X.Hu, Y.shi , and R.C.Eberhart , "Recent advances in particle swarm ," in the proceeding of the congress on Evolutionary computation (CEC'04), vol.2, pp.90-97, Portland , Ore ,USA , June , 2004.
[22] Angeline, P.J. (1998), Using selection to improve particle swarm optimization. IEEE International conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, 1998.
[23] Fogel, D., Beyer H. a note on the empirical evolution of intermediate recombination. Evolutionary computation, vol .3, no.4.
[24] Yao, X., Liu, Y. (1996). Fast evolutionary programming .The Fifth annual conference on Evolutionary programming.
[25] Shi, Y. H., Eberhart, R.C (1998). Parameter selection in particle swarm optimization. 1998 Annual Conference on Evolutionary Programming, San Diego, March 1998.
[26] Swarm Intelligence, First Edition by James kennedy and Russel C.Eberhart with Yuhui shi, page. 338-339.
[27] J.kennedy and R.Mendes, "population structure and particle swarm performance," in the proceeding of the congress on Evolutionary Computation (CEC'02), vol. 2, pp. 1671-1676, Honolulu, Hawaii, USA, July 1999.
[28] R.C. Eberhart and Y.Shi , " Comparing inertia weights and constriction factors in particle swarm optimization , in the proceeding of the congress on Evolutionary Computation (CEC'00), vol. 1, pp. 84-88, La Jolla , Calif , USA, July 2000.
[29] Angeline , "Evolutionary optimization versus particle swarm optimization : philosophy and performance difference ," in the proceeding of the 7th international Conference on Evolutionary programming , pp.601-610, San Diego , Calif , USA, March 1998.
[30] P. Srinivas and R. Battati , "The gregarious particle swarm optimizer (G-PSo)," in the proceeding of the 8th Annual Conference Genetic and Evolutionary Computation (GECCO '06), pp. 67-74, Seattle, Wash, USA, July 2006.
[31] X.-F, W.-J. Zang , and Z.-L. Yang, "Dissipative particle swarm optimization," in the proceeding of the congress on Evolutionary Computation (CEC'02), vol. 2, pp. 1456-1461, Honolulu, Hawaii, USA, July 2002.
[32] M.Iqlab, A.A.Freitas, and C.G. Johnson, "Varying the topology and probability of re-initialization in particle swarm optimization," in the proceeding of 7th International Conference on Artificial Evolution, Lille, France, October 2005.
[33] M.Gen, R.Cheng , and D.Wang, "Genetic algorithms for solving shortest path problems," in the proceeding of the IEEE International Conference on Evolutionary

Computation , pp.401-406, Indianapolis, Ind , USA , April 1997.

[34] A. salman , I.Ahmad , and S.Al-madani, "Particle swarm optimization for task management problem, " Microprocessor and Microsystems, Vol, no .8 , pp.363-371,2002.

[35] L.Cagnina, S.Esquivel, and R.Gallard , "Particle Swarm optimization for sequencing problem : a case study , " in proceeding of the IEEE conference on Evolutionary Computation (CEC'04), vol. 1, pp.536-541, Portlan, Or, USA, June 2004

[36] X.Hu, R.C. Eberhart, and Y.Shi, " Swarm intelligence for permutation optimization: a case study of n-queens problem , " in proceeding of the IEEE swarm Intelligence Symposium (SIS'03), pp.243-246, Indianapolis, Ind, USA, April 2003.

[37] V.kelner, F.Capitanescue , O.Leonard, and L.Wehenkel, " A hybrid optimization techniques coupling evolutionary and local search algorithms, " in Proceedings of the 3rd International conference On advanced Computational methods in Engineering (ACOMEN '05), Ghent, Belgium, May-June 2005.

[38] Yu dong Zhang, ShuihuaWang, and GenlinJi, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," Mathematical Problems in Engineering Volume 2015, Article ID 931256, 38 pages http://dx.doi.org/10.1155/2015/93125.

[39] Glover, F.(1998) . Tabu search, Part I .ORsimulated Anneaaling Journal on computing , I, 190 - 206

[40] Glover, F.(1998) . Tabu search , Part II .ORsimulated Anneaaling Journal on computing , 2,4-32

[41] Chang Wook Ahn, R.S.Raakirshna,"a genetic Algorithm for Shortest path Routing Problem and the sizing of populations", IEEE transaction on Evolutionary computation, vol. 6, No.6. December 2002

[42] J.Kennedy, " Small worlds and mega –minds: effects of neighborhood topology on particle swarm optimization , " in proceeding of the Congress on Evolutionary Computation (CEC '99), vol. 3,pp.1931– 1938, Washington,DC, USA, July 1999.

[43] Y.shi and R.C. Eberhart, "Empirical study of particle swarm optimization, " in proceeding of the Congress on Evolutionary Computation (CEC' 99), vol. 3, pp. 1945 – 1950, Washington , DC, USA, July 1999.

[44] Kulturel-Konak, S., Norman, A. E., & Coit, D. W. (2003).Efficiently solving the redundancy allocation problem using tabu search. IIE Transactions, 35, 515–526.

[45] B.M. Waxman, "Routing of multipoint connections," journal of selected Areas in communications, vol. 6, no. 9, pp. 1617 – 1622, 1998.

[46] Anthony Carlisle, Gerry Dozier, an Off-The-Shelf PSO

Author Profile

Lebeta Belachew Abdissa Received a B.Sc. degree in Computer Science from Jimma University, Ethiopia in 2008. Now, he is a Postgraduate student with a specialization in Applied Computer Technology at Tianjin University of Technology and Education, China (2016-2018). His research interest area is Computer Networking and Security, and intelligent computing.

Zheng Xiaoy Received the B.Sc. degree in Computer Science and Application from Tianjin Polytechnic University, M.Sc. degree in Computer Application from Tianjin Polytechnic University and doctorial degree in Computer Science and Technology from Tianjin University. Her research area is Intelligent Algorithm, Distribution Computing and Data Mining.