An Improved Discrete PSO with GA Operators for Efficient QoS-Multicast Routing

Rehab F. Abdel-Kader
Electrical Engineering Department, Faculty of Engineering - Port-Said
Port-Said University, Port Fouad 42523, Port-Said, Egypt
E-mail: rehabfarouk@eng.psu.edu.eg

Abstract

QoS multicast routing is a non-linear combinatorial optimization problem that arises in many multimedia applications. Providing QoS support is crucial to guarantee effective transportation of multimedia service in multicast communication. Computing the band-width-delay constrained least cost multicast routing tree is an NP-complete problem. In this paper, a novel heuristic QoS multicast routing algorithm with bandwidth and delay constraints is proposed. The algorithm applies the discrete particle swarm optimization (PSO) algorithm to optimally search the solution space for the optimal multicast tree which satisfies the QoS requirement. New PSO operators have been introduced to modify the original PSO velocity and position update rules to adapt to the discrete solution space of the multicast routing problem. A new adjustable PSO-GA hybrid multicast routing algorithm which combines PSO with genetic operators was proposed. The proposed hybrid technique combines the strengths of PSO and GA to realize the balance between natural selection and good knowledge sharing to provide robust and efficient search of the solution space. Two driving parameters are utilized in the adjustable hybrid model to optimize the performance of the PSO-GA hybrid by giving preference to either PSO or GA. Simulation results show that with the correct combination of GA and PSO the hybrid algorithm outperforms both the standard PSO and GA models. The flexibility in the choice of parameters in the hybrid algorithm improves the ability of the evolutionary operators to generate strong-developing individuals that can achieve faster convergence and avoids premature convergence to local optima.

Keywords: *Multicast Routing; Particle Swarm Optimization; QoS*

1. Introduction

With the rapid development of Internet, mobile networks and high-performance networking technology, multicast routing has become a very important research issue in the areas of networks and distributed systems. Currently multicast-based applications have pervasive presence and influence in wide area networks. This increased the demand for multicasting algorithms that can efficiently mange network resources and satisfy QoS requirements [1-4]. The provision of quality-of-service (QoS) guarantees is of utmost importance as the internet expands many new real-time communication services such as multimedia teleconference, audio/video broadcasting, and remote education. These services usually require the transmission of information from one source node to a large number of destination nodes according to a multicast distribution tree [1]. Their analysis is classified as multiple destinations routing (MDR). The main objective of the MDR problem is to construct the optimal multicast tree in the distributed network that determines the best routing for the delivery of a message from the source node to multiple destination nodes while optimizing a certain performance criteria and meeting all QoS requirements. Recently, with the high demand of fast and better quality of services, a number of rigid QoS criteria, such as bandwidth, delay, jitter, and packet loss rate, have been considered.

This QoS multicast routing (QMR) problem has drawn wide spread attention from researchers who have been using different methods to solve the problem using conventional algorithms, such as exhaustive search routing and greedy routing. Typical approaches include (1) applying Dijkstra algorithm to find the shortest path, (2) seeking the minimum network cost using Steiner tree routing algorithm, and (3) finding multicast trees that the paths between source node and the destination nodes are connected and their cost is minimized [4]. An extensive review of the QMR problem can be found in [2, 3].

The multicast routing problem is known to be NP-Complete [1] and for a large scale network with high real time response, it is expensive or even infeasible to find the optimal multicast trees. Thus, most previous researchers have focused on developing heuristic algorithms that take polynomial time and produce near optimal results for solving the multicasting routing problem [4 -19]. The advent of evolutionary computation has inspired new resources for optimization problem solving. Many evolutionary algorithms, such as genetic algorithm (GA) [5-9], simulated annealing (SA) [10], and ant colony optimization (ACO) [4, 11-13], have been proposed for solving the QMR problem. However GA, SA, and ACO have practical limitations in real-time multicast routing. The GA climbing capacity is weak and premature easily. Both the efficiency of the SA algorithm and the quality of the solution depends on procedures that are sensitive to the influence of random annealing sequence. The ACO algorithm has many parameters and cannot guarantee convergence to the global optimal.

Until now, limited papers have discussed the application of particle swarm optimization (PSO) algorithm to solve the QMR problem. Liu et al. [14] and Wang et al. [15] proposed a PSO based algorithm to solve the QMR problem in by means of serial path selection to realize the optimization of a multicast tree. The multicast tree can obtain a feasible solution by exchanging paths in the vector. Experimental results indicate that the proposed algorithm could converge to the optimal or a near-optimal solution with lower computational cost. Another algorithm was described by Sun et al. [16] based on the quantum-behaved PSO (QPSO). It was inspired by quantum mechanics and seemed to be a promising optimization problem solver. The proposed method converts the QoS multicast routing problem into an integer-programming problem and then solves the problem by QPSO. Additionally, combining PSO with other optimization techniques to deal with QoS routing was also proposed in the literature. In [17] Xi-Hong et al. proposed an ACO-PSO algorithm to solve the QMR problem. The solution generated by ACO is regulated by position update strategy of PSO, which extends the search scope efficiently and avoids premature convergence to local optima. The simulation results demonstrate its superiority to other algorithms such as the GA and the ACO. Li et al. [18] described a new evolutionary scheme for the optimization of multicast QoS routing based on the hybrid of GA and PSO, called HGAPSO. In HGAPSO, the upper-half of the best-performing individuals in a population are regarded as elites. Instead of being reproduced exactly in the next generation, these elites are enhanced first. The group constituted by the elites is regarded as a swarm, and each elite corresponds to a particle within it. Wang et al. [19] proposed a new method for tree-based optimization. The algorithm optimizes the multicast tree directly, unlike the conventional solutions to finding paths and integrating them to generate a multicast tree. The algorithm combines PSO with tree based optimization to the solution to control the optimization orientation of the tree shape.

The aim of the paper is to develop an efficient heuristic algorithm that can solve the QMR problem. The proposed algorithm utilizes PSO algorithm that has emerged as a new heuristic that can efficiently solve large-scale optimization problems. This study differs from existing literature in the following aspects: First, in this study various QoS measures are considered such as cost, bandwidth, delay and jitter. The proposed model treats these constraints separately, and can be extended to add more constraints. Second, new discrete PSO operators have been presented to modify the original PSO velocity and position update rules to the discrete solution space in the multicast routing problem. Third, a new adjustable PSO-GA hybrid multicast routing algorithm which combines PSO with genetic operators was proposed. The performance of the adjustable hybrid model is optimized by two driving parameters that give preference to either PSO or GA. The proposed hybrid algorithm can overcome the disadvantages of both PSO and genetic algorithm, and can achieve better QoS performance.

The paper is organized as follows: The PSO algorithm is presented in Section 2.The assumptions and definitions of the multicast routing problem are described in Section 3. The improved PSO algorithm and a demonstration of its realization for solving to multicast routing are explained in Section 4. The Hybrid PSO-GA algorithm is described in Section 5. Simulation results are given in Section 6. Finally, we summarize the paper with some concluding remarks in Section 7.

2. Particle Swarm Optimization

PSO proposed by Dr. Eberhart and Dr. Kennedy in 1995 is a computational paradigm based on the idea of collaborative behavior and swarming in biological populations inspired by the social behavior of bird flocking or fish schooling [20-23]. The algorithm, which is based on a metaphor of social interaction, searches a space by adjusting the trajectories of individual vectors, called “particles” as they are conceptualized as moving points in multidimensional space. The individual particles are drawn stochastically toward the position of their own previous best performance and the best global performance among its neighbors. PSO algorithm is simple, easy to implement, robust to control parameters, and computationally efficient compared to other heuristic optimization techniques. The original PSO has been applied to a learning problem of neural networks and function optimization problems, and efficiency of the method has been confirmed.

When PSO is used to solve an optimization problem, a swarm of particles, is used to explore the solution space for an optimum solution. Each particle represents a candidate solution and is identified with specific coordinates in the D-dimensional search space. The position of the i^{th} particle is represented as $X_i = (x_{i1}, x_{i2},\ldots\ldots, x_{iD})$. The velocity of a particle is denoted as $V_i = (v_{i1}, v_{i2}, \ldots\ldots\ldots, v_{iD})$. The fitness function is evaluated for each particle in the swarm and is compared to the fitness of the best previous result for that particle and to the fitness of the best particle among all particles in the swarm. After finding the two best values, the particles evolve by updating their velocities and positions according to the following equations:

$$V_i^{t+1} = \omega * V_i^t + c_1 * rand_1 * (p_{i\_best} - X_i^t) + c_2 * rand_2 * (g_{best} - X_i^t) \quad (1)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2)$$

Where $i =(1, 2,\ldots\ldots N)$ and N is the size of the swarm; p_{i_best} is the particle best reached solution and g_{best} is the global best solution in the swarm. c_1 and c_2 are cognitive and social parameters that are bounded between 0 and 2. $rand_1$ and $rand_2$ are two random numbers, with uniform distribution $U[0,1]$. $-V_{max} \le V_i^{t+1} \le V_{max}$ (V_{max} is the maximum velocity). The inertia weight ω, is a factor used to control the balance of the search algorithm between exploration and exploitation. The recursive steps will go on until we reach the termination condition.

3. Multicast Routing Problem Formulation

In this paper we mainly focus on the band-width-delay constrained least cost multicast routing problem [4-7, 10, 15, 17, 19]. The communication network can be modeled as an undirected weighted graph *G (V, E),* where *V* is the set of all nodes representing routers or switches, *E* is the set of all edges representing physical or logical connection between nodes. Each link $(x, y) \in E$ in *G* has three weights *(b(x,y), d(x,y), c(x,y))* which correspond to the available bandwidth, the delay and the cost of the link respectively. We assume that $s \in V$ represents the source node and $M \subseteq \{V-\{s\}\}$ represents a set of multicast destination nodes, then *s* and *M* construct a multicast tree *T*(*s*, *M*), the following relationship exits in multicast tree *T*(*s*, *M*):

The total delay of the path *P(s, d)* in *T* is simply the sum of the delay of all links along *P(s, d)* where $v \in M$, i.e.

$$Delay\ P(s, d) = \sum_{(i, j) \in P(s, d)} d(i, j) \quad (3)$$

The total delay of the tree T(s, M) is defined as the maximum value of the delay on the paths from the source node to each destination node:

$$Delay\ T(s, M) = \max(Delay\ P(s, d)), \forall v \in M \quad (4)$$

The jitter of the tree is defined as the average difference of the delay on the paths from the source node to each destination node:

$$Jitter(T(s, M)= \sqrt{\sum_{v \in M} (Delay\ P(s, d) - delay\_avg\ )^2\ )} \quad (5)$$

where *delay_avg* refers to the average value of the delay on the paths from the source node to each destination node. The bottleneck bandwidth of the tree is defined as the minimum value of the bandwidth of all the links in *T:*

$$Bandwidth\ T(s, M) = \min\{ b(i,j) \mid (i,j) \in T(s, M) \} \quad (6)$$

The total cost of the tree *T(s, M)* is defined as the sum of the costs of all links in that tree and can be given by:

$$Cost\ T(s, M) = \sum_{(i,j) \in T(s,M)} c(i,j) \quad (7)$$

The QoS multicast routing problem with delay and bandwidth constrained can be described as follows: Given network graph *G*, a source node *s*, and a multi destination multicast member set *M*, the delay, the jitter delay and bandwidth constraints D_{max}, D_J, and B_{min}. The problem is to find the multicast tree $T(s, M) \subseteq G$ spanning the source node *s* and the set of destination nodes $v \in M$ that *minimizes* the cost function *Cost T(s, M)* subject to the following conditions:

1. The multicast tree is a spanning tree of G
2. The root of the multicast tree is node s
3. All the destination nodes are in the multicast tree
4. All the leave nodes in the multicast tree belong to M
5. The tree must satisfy QoS constraints such as:

Delay $T(s, M) \leq D_{max}$ End-to end delay requirement (8)
Jitter($T(s, M) \leq D_J$ Jitter delay constraint (9)
Bandwidth $T(s, M) \geq B_{min}$ Bandwidth provisioning for guaranteed QoS (10)
Equations (8) and (9) guarantee the delay requirements of QoS, in which D_{max} and D_J are the maximum permitted delay of real time services. Relation (10) guarantees the bandwidth of communication traffic, in which B_{min} is the minimum required bandwidth for all applications.

4. Proposed Multicast Routing Algorithm

In the network graph, $G = (V, E)$, there are |V| x (|V|-1) possible source-destination pairs. A source-destination pair can be connected by a set of links, which is called a “route”. There are usually many possible routes between any source destination pair. For example, consider the network shown in Figure 1; the possible routes between s and v_6 include s - v_2- v_6, s - v_1- v_3- v_6, …… and so on.

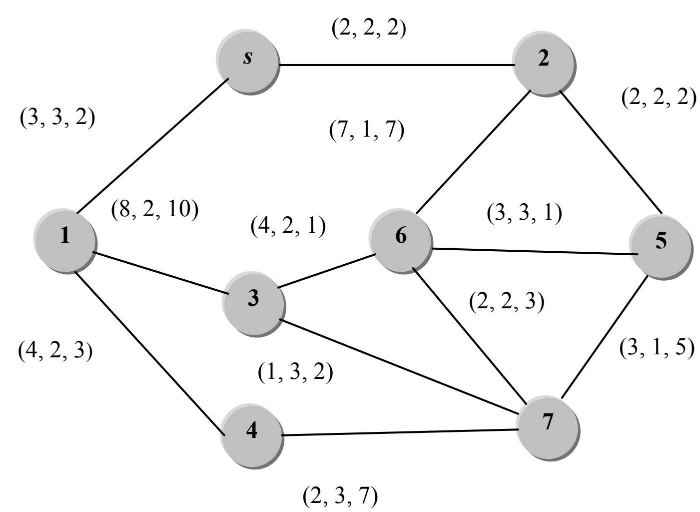

Figure 1. Topology of a Multicast Network.

Our simple PSO-based multicast routing algorithm assumes that a routing table has been constructed for each source-destination pair. Table 1 shows the routing table for the source-destination pair (s, v_6). When the network size is large, the number of possible routes between a source destination pair becomes huge. Therefore the number of candidate routes must be limited to a reasonable amount R which is a parameter of the algorithm. All possible routes for a source destination pair are sorted and indexed according to the path cost, or delay (length) such that the shorter paths are assigned smaller route numbers. Only the R shortest routes will be listed in the routing table. Preferring routes with smaller delay are preferred so as to take path delay into consideration in finding multicast trees. All links with bandwidths less than the minimum required bandwidth threshold B_{min} will be removed before the path search process to compress the search space.

Table 1. Example of a Routing Table for s→ v6

Routing table for $s \rightarrow v_6$		
Route number	**Route list**	**Route delay**
0	s- v_2- v_6	3
1	s- v_1- v_3- v_6	7
2	s- v_2- v_5- v_6	7
i		
R-1		

4.1. Particle Encoding

For a given source node s and a set of multicast destination nodes $M = \{v_1, v_2, \ldots\ldots, v_k\}$ a particle/individual P = $(A_1, A_2, \ldots., A_i, \ldots., A_k)$ is a string of integers $\in\{0,1,\ldots..R\text{-}1\}$ with length k. Each element (gene) g_i in the particle represents a possible route between nodes s and v_i. The relation between particle, path gene, and path routing table is shown in Figure 2.

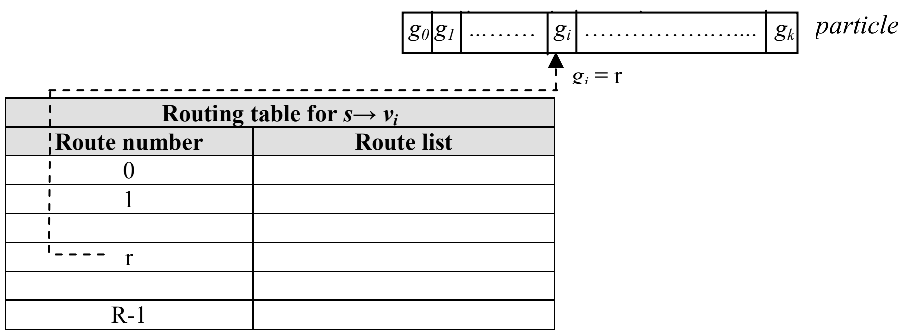

Figure 2. Particle Encoding.

This encoding schema denoted as *path-oriented encoding* was first introduced in [8] for point-to point routing communication problems. Each individual represents a candidate solution for the multicast routing problem as it guarantees a path from the source node to every destination. However the individual does not necessary represent a multicast tree as it might violate light splitting constraints. This problem will be handled by preventing cycles to appear in the individual by assigning them to bad fitness values and thus they will be removed from the population. The major advantage of using this coding method is that given an individual it is easy to identify the links in the multicast tree. Another advantage is that path delay or any other performance criteria can be used in the selection of the proper route from the routing table.

4.2. Fitness Function

PSO maintains a swarm of individuals that are evaluated in each generation. In the QoS multicast routing problem the evolution is driven by a fitness function that evaluates the quality of the evolved particles based on their ability to minimize the cost of multicast tree $T(s, M)$. In addition, QoS constraints should be considered except the bandwidth constraint as if the link does not meet bandwidth constraint, it will be automatically eliminated from network. The fitness function of the algorithm is defined as follows:

$f(T(s, M)) = Cost(T(s, M)) + \eta_1 \min\{D - delay(T(s, M)), 0\} + \eta_2 \min\{D_J - Jitter(T(s, M)), 0\}$ (11)

Where η_1 and η_2 are punishment coefficients, the value of the coefficient decides the punishment extent. If $a \geq 0$, then min (a, 0) = a; else min (a, 0) = 0.

4.3. Simple PSO Multicasting Algorithm

The first use of PSO in discrete optimization was for solving the traveling salesman problem (TSP) [21]. The main issue was to modify the position and velocity vector equations (1) and (2) in the original PSO algorithm to span the discrete search domain. These equations can only be used in continuous optimization problems. In the multi-cast route problem the position vector of the particles is coded as an integer sequence rather than a real number vector. The traditional addition and subtraction operators in equation (1) can not be applied in this problem. The proposed algorithm modifies the original velocity and position update equations to adapt to the multicast routing problem domain by introducing new operators as follows:

adjust(i, n) operation

The particle P = $(A_1, A_2, \ldots, A_i, \ldots, A_k)$ represents the possible routes from the source node to each of the destination nodes. After applying the *adjust*(*i*, *n*) operator, A_i changes as shown in equation (12):

$$A_i = \begin{cases} A_i + n & \text{if } A_i + n \leq R \\ (A_i + n)\%R & \text{if } A_i + n > R \end{cases} \quad (12)$$

Where R is the size of the routing table.

Subtraction between two positions

Assume that $X = (x_1, x_2, \ldots, x_i, \ldots, x_k)$ and $Y = (y_1, y_2, \ldots, y_i, \ldots, y_k)$. The velocity V is defined as the result of subtraction between two position $V = X - Y$. For example, if X = (1, 2, 4, 5) and Y = (1, 3, 4, 5), then V=$X-Y$ = *adjust* (2, 1).

Addition between two velocities

Assume to velocity V_1 and velocity V_2, we can denote V_1 add $V2$ by $V = V_1 \oplus V_2$. For example, if V_1 =*adjust*(3, 4) and V_2 =*adjust* (2, 1), $V = V_1 \oplus V_2$ = {*adjust* (3, 4), *adjust* (2, 1)}.

A *change* sequence *CS* is made up of one or more *adjust* operators; i.e.:

$$CS = (adjust_1(), adjust_2(), \ldots, adjust_n()) \quad (13)$$

The order of the *change* operator in *CS* is important as the *adjust* operators in the change sequence act on the solution in order. This can be described by the following formula:

$$Y = X + CS = X + (adjust_1(), adjust_2(), \ldots, adjust_n()) = (((X, adjust_1()), adjust_2()), ., adjust_n()) \quad (14)$$

Addition between position and velocity

Assume to position X and velocity V, a new position vector Y is obtained by imposing the velocity vector to the old position vector. For example, if X = (1, 2, 4, 5) and V = *adjust* (2, 3), then $Y = X + V$ = (1, 5, 4, 5).

According to equation (2) a new position vector is obtained by imposing the velocity vector to the old position vector. The velocity vector can be defined as a change sequence *CS* acting on the position vector. The velocity vector evolves according to the formula given in equation (1). The $(p_{i\_best} - X_i(t))$ component in equation (1) means the basic change sequence CS_1 that should act on $X_i(t)$ to get to p_{i_best}, $CS_1 = p_{i\_best} - X_i(t)$. We can adjust the nodes in $X_i(t)$ according to p_{i_best} from left to right to get CS_1. For example, consider $X_i(t)$ = [2, 3, 3, 2, 5] and p_{i_best} = [1, 3, 4, 2, 4] and a routing table of size 5. The first adjust operator to operate on $X_i(t)$ is *adjust*(1, 4); i.e. $X_i'(t) = X_i(t) + adjust(1, 4)$ = [1, 3, 3, 1, 5]. Similarly, the second adjust operator is *adjust*(3, 1), and $X_i''(t)$ = [1, 3, 4, 1, 5]. The third adjust operator to operate on $X_i''(t)$ is *adjust*(5, 4) producing $X_i'''(t)$ where $X_i'''(t) = p_{i\_best}$= [1, 3, 4, 2, 4]. Finally we get the basic adjust sequence $CS_1 = p_{i\_best} - X_i(t) = (adjust(1, 4), adjust(3, 1), adjust(5, 4))$.

The same rule is applied to the $(g_{best} - X_i(t))$ component in the velocity equation. The new velocity vector consists of three *CS*s: the old velocity vector $V_i(t)$, $(p_{i\_best} - X_i(t))$ and $(g_{best} - X_i(t))$. The three *change* sequences can be merged into a new equivalent *change* sequence. Suppose, CS_1, CS_2, CS_3 and act on one solution P in this particular order, CS_1 first, CS_2 second, and CS_3 third to get a new solution P'. This is equivalent to a new *change* sequence CS described as follows:

$$CS = CS_1 \oplus CS_2 \oplus CS_3 \tag{15}$$

Assuming that the inertia weight factor $w=1$, the new position and velocity evolution equations in the discrete domain can be rewritten as follows:

$$V_i(t+1)=V_i(t) \oplus c_1 * r_1 * (p_{i\_best} - X_i(t)) \oplus c_2 * r_2 * (g_{best} - X_i(t)) \tag{16}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{17}$$

5. Hybrid PSO-GA Multicast Routing Algorithm

GA and PSO are both population based algorithms that have proven to be successful in solving very difficult optimization problems [20]. However, both models have strengths and weaknesses. The PSO algorithm is conceptually simple and can be implemented in a few lines of code. PSOs also have memory, whereas in a GA if an individual is not selected the information contained by that individual is lost. In PSO the collaborative group interactions enhance the search for an optimal solution, whereas GAs have trouble finding an exact solution and are best at reaching a global region. However, without a selection operator PSOs may waste resources on a poor individual that is stuck in a poor region of the search space. Comparisons between GA and PSOs have been performed by both Eberhart [22] and Angeline et al. [23] and both studies suggested that a hybrid of the standard GA and PSO models would lead to a very effective search strategy.

The standard PSO algorithm may not be flexible enough for practical applications particularly when the problem to be tackled is complicated, conflicting and multitasking. Means for modifying the PSO structure, fitness function, and PSO operators are sought in order to meet the design requirements. In this paper we propose a hybrid PSO-GA algorithm combining the strengths of PSO and GA to enhance the search process in the QoS multicast problem. The hybrid algorithm combines the standard velocity and position update rules of PSOs with the ideas of selection, crossover and mutation from GAs. The population update concept can be easily understood thinking that a part of the individuals are the same of the previous generation but moved on the solution space by PSO. The remaining individuals are substituted by new generated ones by means of GA operators. This kind of updating results in a more natural evolution, where individuals not only improve their scores for natural selection of the fitness, or for good-knowledge sharing, but for both of them at the same time.

The main objective of the proposed algorithm is to design an adjustable technique that makes it possible to optimize the performance of the PSO-GA hybrid. Two driving parameters are added in the hybrid algorithm to give preference to either PSO or GA. The PSO velocity vector is multiplied by an influence term $\lambda \in [0: 1.0]$. When this term is set to 0 the PSO has no effect on the population, when set to 1 the PSO runs as the standard PSO. For intermediate values the PSO functions normally, but the size of the steps taken by the particles is reduced. The GAs selection operator has a *replacement* term $\psi \in [0:1.0]$ which determines how many individuals in the population get replaced and crossed over in the current generation. When the $\psi = 0$ no individuals/particles are selected for crossover or mutation and the GA has no effect on the population. When the $\psi = 1$ the entire population is replaced in the generation. First, the hybrid algorithm performs the standard velocity and position update rules, with the *influence* term. The top *(population size *(1 −ψ))* individuals, based on fitness, are copied into the new population. Selection, crossover and mutation then occur on the appropriate number of individuals determined by the *replacement* term to fill the remainder of the population. The flowchart of the proposed PSO-GA algorithm is shown in Figure 3.

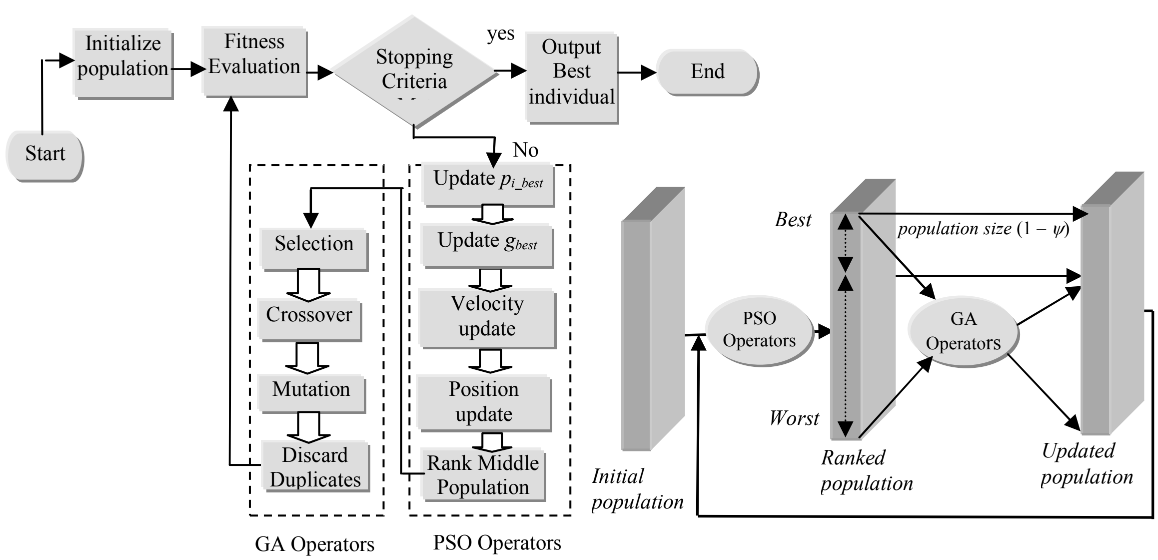

Figure 3. Flow-chart of the PSO-GA Hybrid Algorithm.

In later evolution stage of PSO algorithm, the convergence speed becomes significantly slower. At the same time, after the algorithm converges to a certain precision, it can not optimize anymore. In order to maintain the algorithm diversity, improve the search performance and avoid PSO algorithm plunged into local optimum, we propose to join the crossover and mutation operator together. The crossover and mutation operator are described as follows:

5.1. Crossover Operator

Randomly select two particles from the population according to the crossover probability P_c. The crossover operator used in this algorithm is two point crossover (*TPC*). The *TPC* randomly select two integers i_1, i_2 ($i_1 < i_2$) are generated in the interval *[1, K]*. These integers are used as the crossover sites. The two generated offspring's are evaluated based on their fitness.

5.2. Mutation Operator

Single routing path mutation (*SRPM*): first, a single destination node v_i *(i* $\in$ *1, 2,……,k)* is selected randomly. Then the *SRPM* changes the value of path gene g_i corresponding to v_i to a random integer which is selected randomly from *[1, … ,R]*. After performing *SRPM* the new individual is evaluated and it may replace the original individual if it has a better fitness value.

Multiple routing paths mutation (*MRPM*): first, select an integer *z* between 1 and *K* which correspond to the number of genes that will be mutated. Then, *MRPM* changes the value of path genes g_i's corresponding to the destination nodes selected to random integers selected randomly from *[1, …,R]*. After performing the MRPM, the new individuals are evaluated and each individual may replace its original individual if it has a better fitness value.

5.3. Discard Duplicate Particles

In the particle population, there may exist two or more duplicate particles. A large number of duplicate particles in the population will reduce the searching ability. Once this situation occurs, the duplicated particles must be discarded. In our algorithm, they are replaced by new randomly generated particles.

6. Experimental Results

In the proposed algorithm, each gene in the particle encoding represents a possible route from the source node to one of the destinations. Hence, the routing table for each source-destination pair needs to be constructed first. Obviously, the number of possible routes between two nodes heavily depends on the network topology. If the network is densely connected or the size of the network is large, the number of possible routes of a source-destination pair becomes huge. Hence, it is impossible to list all the possible routings in the routing table. In the simulation, we set the size of routing table to 64. An algorithm was designed to automatically generate the shortest 64 routes for each destination.

The WAXMAN [24] model was adopted in the experiments to generate different scale random network topologies. A WAXMAN network topology with 40 nodes is shown in Figure 4. The random graphs are generated with an average degree 4, which have the appearance roughly resembling that of geographical maps of major nodes in the internet.

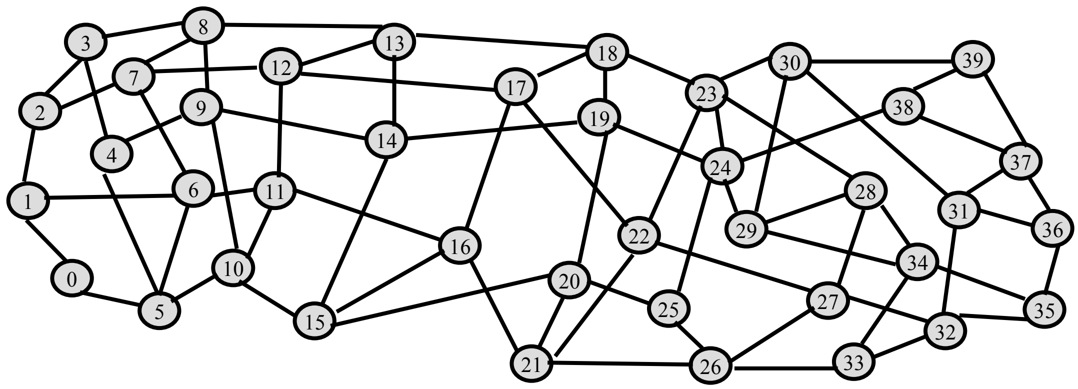

Figure 4. The WAXMAN Network Topology Model with 40 Nodes.

The bandwidth and delay of each link are uniformly distributed in the range [40, 80] and [0, 30] respectively. The cost of each link is uniformly distributed in the range [5, 10]. The proportion of the Multicast member nodes is between 10% and 20% of the total network nodes. All tests are performed with a population size of 100 and the average solution is obtained by running the program 50 times. Furthermore, to consider the influence of selecting source and destination nodes on the algorithm, different source and destination nodes in the same simulation topology were selected in the various runs of the same test. We performed experiments differentiating scales of random networks and the QoS constraints as presented in Table 2.

Table 2. Setting of the Upper Bound of Delay and Jitter on Different Network Scales.

Node number	Delay upper bound	Jitter upper bound
≤ 40	120	60
40-80	150	62
80-120	190	65
120-160	240	66
160-200	300	68

To compare the capability of the hybrid model, four implementations of the hybrid algorithm, using different *influence* and *replacement* parameters were examined as follows:

- PSO - used an *influence* term of 1 throughout the entire test. *Replacement* was set to 0, so the GA has no effect on the population. This implementation is the standard implementation of PSO.
- GA - used an *influence* term of 0, so the PSO has no effect on the population. *Replacemen*t was set to 1 throughout, which is effectively a normal generational GA.
- Hybrid-1 - increases the *influence* term linearly from 0 to 1, based on the current generation. The *replacement* term is reduced linearly from 1 to 0, based on the current generation. This test was designed with the expectation that the best results would be with the GA performing an initial global search with the PSO finishing as a local search.
- Hybrid-2 - reduces the *influence* term from 1 to 0 and increases the *replacement* term from 0 to 1. This test was designed with the expectation that the PSO performing an initial global search with the GA finishing as a local search would produce poor results.

Figure 5 represents the convergence time of the proposed QoS multicast routing algorithm with different topology scales for the PSO, GA, Hybrid 1, and Hybrid 2 implementations.

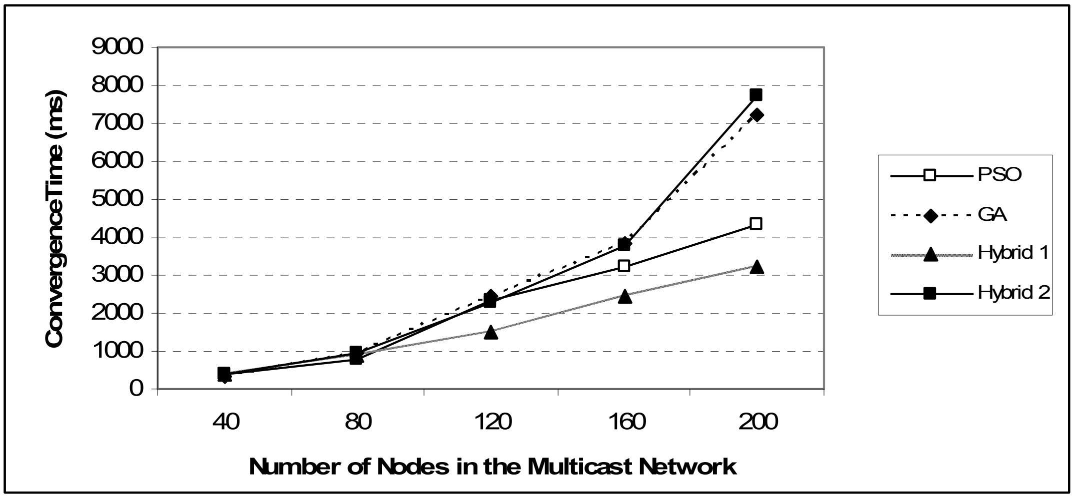

Figure 5. Comparison of Convergence Time (ms) in Different Topology Scales.

Results clearly show that the convergence time of each algorithm increases with the increase of the topology scale. The speed of time consumption of the PSO, GA and Hybrid 2 algorithms grows faster than the Hybrid 1 algorithm when there are more than 80 nodes in the network. The more nodes there are, the more the discrepancy is apparent. Figure 5 shows that when the topology has 200 nodes, the Hybrid 1 algorithm can yield a 25% - 62% reduction in the convergence time compared to the other implementations. This indicates that the multicast routing using the Hybrid 1 parameter setting provides a significant improvement for obtaining a global optimum or a near global optimum solution quickly.

The average tree costs of the four implementations in different topology scales are shown in Figure 6. The proportion of the Multicast member nodes is between 10% and 20% of the total network nodes The Figure indicates that the PSO and the Hybrid 1algorithm can find better solutions than the GA and Hybrid 2 algorithms. The Hybrid 1 algorithms outperforms the PSO algorithms for larger scale topologies with nodes >80. The Hybrid 1 performs well for different source and destination nodes, the cost can converge to a good result, indicating that the algorithm is feasible and effective. It not only converges fast, but also can escape from local optimum and effectively search the global optimum.

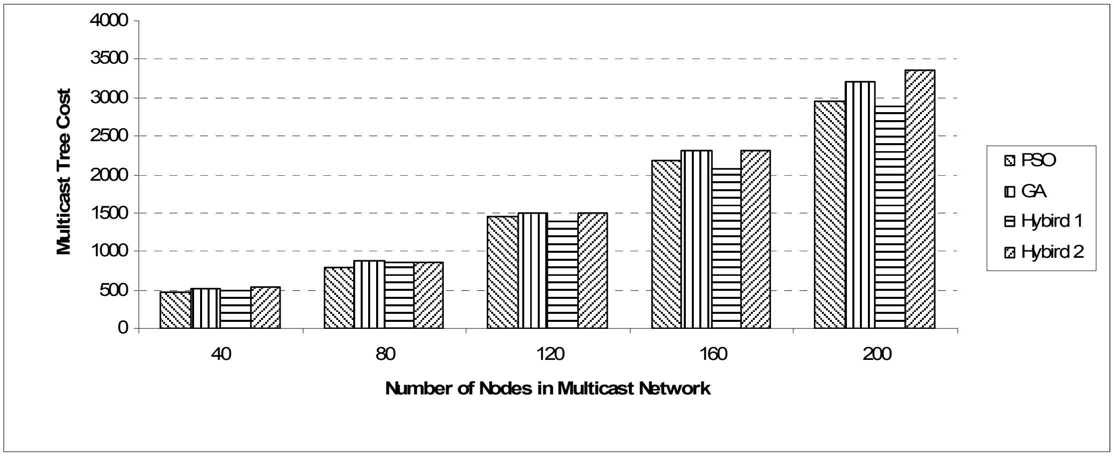

Figure 6. Comparison of the Average Tree Cost in Different Topology Scales.

By experiment, it was found that not only the topology scale but also the number of destination nodes within the topology that affects the algorithm's performance. To test this effect, we chose the multicast group with ratios ranging from 5% to 75% in the case of a topology with 40 nodes. The average tree cost of the four algorithms tested is shown in Figure 7. All algorithms have approximately similar performance when the ratio of multicast destination nodes is below 15%. However, as the ratio increases, the performance of the Hybrid 1 is superior to the other three algorithms and can achieve up to 20% improvement in the average tree cost. This indicates that when the Hybrid 1algorithm converges to a local optimum, it can escape premature convergence and effectively search for the global optimum.

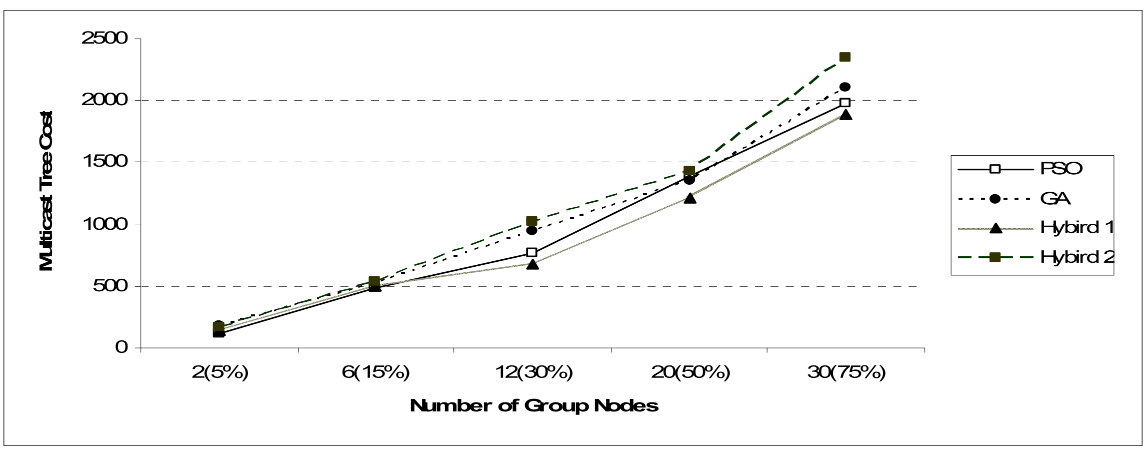

Figure 7. Comparison of Algorithms with Different Ratios of Group Member Nodes.

7. Conclusions

Multicast routing problem arises in many multimedia communication applications. The provision of quality-of-service (QoS) guarantees is of utmost importance as the internet expands many new real-time communication services. Computing the band-width-delay constrained least-cost multicast routing tree is a NP-complete problem. In this paper, a novel multicast routing algorithm based on an improved discrete PSO algorithm is proposed. The algorithm utilizes the discrete PSO algorithm to optimally search the solution space for the optimal multicast tree which satisfies the QoS requirement. New PSO operators have been proposed to modify the original PSO velocity and position update rules to the discrete solution space in the multicast routing problems. Furthermore, a new PSO-GA hybrid multicast routing algorithm which combines PSO with genetic operators was proposed. The proposed hybrid technique combines the strengths of PSO and GA to realize the balance between natural selection and good knowledge sharing to provide robust and efficient search of the solution space. Two driving parameters are utilized in the PSO-GA hybrid to give preference to either PSO or GA Simulation results show that the proposed hybrid algorithm can overcome the disadvantages of PSO and genetic algorithm, and achieve better QoS performance. The flexibility in the choice of parameters in the hybrid algorithm improves the ability of the evolutionary operators to generate strong-developing individuals that can achieve faster convergence and avoids premature convergence to local optima. To verify the performance of the proposed algorithms, simulations were carried out with different sizes of multicast groups on diverse topology networks. Experimental results show that the algorithm is feasible and effective. It not only converges fast, but also can escape from local optimum and effectively search for the global optimum. Therefore, the proposed PSO-GA hybrid algorithm for multicast routing is an effective solution to the QoS multicast routing problem.

References

[1] Z. Wang and J. Crowcroft, "Quality of Service for Supporting Multimedia Application," IEEE Journal on Selected Areas in Communication, Vol. 14, pp. 1228–1234, 1996.

[2] L. H. Sahasrabuddhe and B. Mukherjee, "Multicast Routing Algorithms and Protocols: A Tutorial," IEEE Network, pp. 90-102, 2000.

[3] B. Wang and J. C. Hou, "Multicast Routing and Its QoS Extension: Problems, Algorithms, and Protocols," IEEE Network, pp. 22-36, 2000.

[4] C.-H. Chu, J. Gu, X. Hou, and Q. Gu, "A Heuristic Ant Algorithm for Solving QoS Multicast Routing Problem," In Proceedings of the 2002 World on Congress on Computational Intelligence, Vol. 2, pp. 1630-1635, 2002.

[5] R.-H. Hwang, W.-Y. Do, and S.-C. Yang, "Multicast Routing Based on Genetic Algorithms," Journal of Information Science and Engineering, pp. 885-901, 2000.

[6] X. Li, Q. Ning, Z. Jun-Ya, and L. Yang- Qian, "An Improved GA for QoS Multicast Routing Algorithm," In Proceedings of 2008 Chinese Control and Decision Conference, pp. 393-396, 2008.

[7] P. Yang and B. Huang, "GA Based QoS Multicast Routing Algorithm in Mobile Ad Hoc Network," 2008 International Seminar on Future BioMedical Information Engineering, pp.247-250, 2008.

[8] N. Shimamoto, A. Hiramatu, and K. Yamasaki, "A Dynamic Routing Control based on a Genetic Algorithm," In Proceedings of 1993 IEEE International Conference on Neural Network, pp. 1123-1128, 1993.

[9] Z. Wang, B. Shi, "Solution to QoS Multicast Routing Problem based on Heuristic Genetic Algorithm", Journal of Computer, China Computer Federation, pp. 55-61, 2001.

[10] N. Lin, T. Yang, and L.-X. Song, "A New QoS Multicast Routing Algorithm for MPLS-TE," In Proceedings of 2010 International Conference on Measuring Technology and Mechatronics Automation, pp. 192-195, 2010.

[11] Q. Ling and C. Ling, "A Multicast Routing Algorithm based on Ant Colony System," Information and Control Journal, Vol. 35, No. 5, pp. 546-550, 2006.

[12] D. Pinto and B. Barán, "Solving Multi-objective Multicast Routing Problem with a new Ant Colony Optimization Approach," In Proceedings of the 3rd international IFIP/ACM Latin American conference on Networking, pp. 11-19, 2005.

[13] L. Carrillo, J.-L. Marzo, L. Fabrega, P. Vila, and C. Guadall, “Ant Colony Behaviour as Routing Mechanism to Provide Quality of Service,” Lecture Notes in Computer Science, Vol. 3172, Springer, Berlin, pp. 306-323, 2004.

[14] J. Liu, J. Sun, and W. Xu, “QoS Multicast Routing based on Particle Swarm Optimization,” Lecture Notes in Computer Science, Vol. 4224, Springer, Berlin, pp. 936–943, 2006.

[15] Z. Wang, X. Sun, and D. Zhang, “A PSO-Based Multicast Routing Algorithm,” In Proceedings of Third International Conference on Natural Computation (ICNC), pp. 664 – 667, 2007.

[16] J. Sun, J. Liu, and W. Xu, “ QPSO-based QoS Multicast Routing Algorithm”, Lecture Notes in Computer Science Vol. 4247, Springer, Berlin, pp. 261–268, 2006.

[17] C. Xi-Hong, L. Shao-Wei, G. Jiao, and L. Qiang, “Study on QoS Multicast Routing Based on ACO-PSO Algorithm, ” In Proceedings of 2010 International Conference on Intelligent Computation Technology and Automation, pp. 534-537, 2010.

[18] C. Li, C. Cao, Y. Li, and Y. Yu, “Hybrid of Genetic Algorithm and Particle Swarm Optimization for Multicast QoS Routing, ” In Proceedings of IEEE International Conference on Control and Automation, pp. 2355–2359, 2007.

[19] H. Wang , X. Meng, S. Li, and H. Xu , “A Tree-Based Particle Swarm Optimization for Multicast Routing, ” Computer Networks, Vol. 54, pp. 2775–2786, 2010.

[20] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” IEEE International Conf. on Neural Networks, Vol. 4, pp. 1942-1948, 1995.

[21] K.-P. Wang, L. Huang, C.-g. Zhou, and W. Pang, “Particle Swarm Optimization for Traveling Salesman Problem,” In Proceedings of the Second International Conference on Machine Learning and Cybernetics. Vol. 5, pp. 1583-1585, 2003.

[22] P. J. Angeline, “Evolutionary Optimization versus Particle Swarm Optimization: Philosophy and Performance Differences,” Evolutionary Programming VII, Lecture Notes in Computer Science, Vol. 1447, pp. 601-611, 1998.

[23] R. C. Eberhart, and Y. Shi, “Comparison between Genetic Algorithms and Particle Swarm Optimization,” Evolutionary Programming VII, Lecture Notes in Computer Science, Vol. 1447, pp. 611-616, 1998.

[24] B. M. Waxman, “Routing of Multipoint Connections,” IEEE Journal on selected Areas in Communications, Vol. 6, No. 9, pp. 1617–1622, 1988.