

Research Article

A Rough Penalty Genetic Algorithm for Multicast Routing in Mobile Ad Hoc Networks

Chih-Hao Lin and Chia-Chun Chuang

Department of Information Management, Chung Yuan Christian University, Zhongli City 32023, Taiwan

Correspondence should be addressed to Chih-Hao Lin; linch@cycu.edu.tw

Received 26 April 2013; Accepted 13 July 2013

Academic Editor: Anyi Chen

Copyright © 2013 C.-H. Lin and C.-C. Chuang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multicast routing is an effective way to transmit messages to multiple hosts in a network. However, it is vulnerable to intermittent connectivity property in mobile ad hoc network (MANET) especially for multimedia applications, which have some quality of service (QoS) requirements. The goal of QoS provisioning is to well organize network resources to satisfy the QoS requirement and achieve good network delivery services. However, there remains a challenge to provide QoS solutions and maintain end-to-end QoS with user mobility. In this paper, a novel penalty adjustment method based on the rough set theory is proposed to deal with path-delay constraints for multicast routing problems in MANETs. We formulate the problem as a constrained optimization problem, where the objective function is to minimize the total cost of the multicast tree subject to QoS constraints. The RPGA is evaluated on three multicast scenarios and compared with two state-of-the-art methods in terms of cost, success rate, and time complexity. The performance analyses show that this approach is a self-adaptive method for penalty adjustment. Remarkably, the method can address a variety of constrained multicast routing problems even though the initial routes do not satisfy all QoS requirements.

1. Introduction

Multicasting is a service method in which a source node can deliver copies of messages to multiple recipients at different locations in a communication network. Multicasting techniques play a critical role in many applications such as video conference, internet games, and web-based learning. In this paper, multicast routing problems mainly focus on finding a minimum Steiner tree and satisfying quality-of-service (QoS) requirements. Unfortunately, the problem of finding a Steiner tree is known to be a NP-complete problem [1], even if links have unit costs.

The multicast tree in mobile ad hoc networks (MANETs) is vulnerable to intermittent connectivity property during the transmission period [2]. Due to the nodal mobility and the dynamic topology of mobile networks, a service provider should find a cost-effective tree for its multicast customers in real time and assure certain QoS requirements [3]. Without the support of communication infrastructure, the challenge of the dynamic routing on a changing topology is how to decide a multicast tree as soon as possible. To enhance the effectiveness and efficiency, this work uses

a routing representation scheme to encode the multicast tree. Therefore, this paper models the multicast routing problem with QoS constraints to support multimedia transmission in MANETs [4, 5].

The battery limitation of a mobile node is a critical constraint while developing multicast routing protocols. Genetic algorithm (GA) presents a potential solution for the multiconstrained multicast routing problem [6]. Traditionally, penalty-function methods are the most popular constraint-handling techniques. According to the degree of constraint violation, penalty coefficient should be determined carefully [7]. The static-penalty (SP) method applies a static coefficient for each constraint and then adjusts penalty coefficient manually [8]. To adjust penalty coefficient automatically, the dynamic-penalty (DP) method combines the generation number and a scaling constant to adjust coefficient automatically [9]. Furthermore, the adaptive-penalties (AP) method tries to avoid infeasible solutions by adjusting the penalty coefficient according to the convergent situation [10].

Different from several related researches [2, 4, 5], this paper synthesizes the rough set theory (RST) and penalized techniques as a rough penalty genetic algorithm (RPGA).

The emphasis of the proposed RPGA uses a rough-penalty (RP) method to releases/enforces some penalties on inefficient/efficient constraints during evolution. To facilitate the effectiveness of multiple penalties, the RPGA incorporates with a therapeutic crossover to enlarge the genetic diversity in a population and guide to find the optimal solution. The performance of the RPGA is evaluated by three kinds of constrained multicast routing scenarios. Experimental results show that the proposed RPGA not only can find near-optimal solutions but also can obtain robust feasible results for QoS-based multicast routing problems.

The rest of paper is organized as follows. Section 2 models the multicast routing problems with QoS constraints in MANETs. In Section 3, the operations of the proposed RPGA are described in detail. Section 4 introduces the proposed RP method. Section 5 reports the experimental results, algorithm analyses, and performance comparisons for three test networks. Finally, the paper is summarized in Section 6.

2. Problem Description and Formulation

2.1. Network Modeling. At a certain time period in a MANET, we assume that the service provider knows (1) traffic load from node i to node j ($i \neq j$), (2) QoS requirements/constraints (e.g., delay constraint), (3) bandwidth available of each link, (4) link cost for the traffic to pass through each link, and (5) time delay for the traffic to pass through each OD pair. We consider the multicast routing problem with bandwidth and delay constraints. The communication network is modeled as a connected weighted, directed graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is a finite set of network nodes and $E = \{e_1, e_2, \dots, e_m\}$ is the set of network links. The link $e = (u, v) \in E$ connects node $u \in V$ to node $v \in V$ with positive cost function (i.e., $\text{cost}(e) : E \rightarrow R^+$), available bandwidth (i.e., $\text{BW}(e) : E \rightarrow R^+$), and delay function (i.e., $\text{delay}(e) : E \rightarrow R^+$). The number of nodes and links (i.e., the cardinalities of V and E) is n and m , respectively. For each multicast session, messages are routed from a source node s to a set of multicast destination group $D = \{d_1, d_2, \dots, d_k\} \subseteq V$. A multicast tree $T(s, D) = (V_T, E_T)$ represents a solution to the multicast routing problem, where $V_T \subseteq V$ and $E_T \subseteq E$. Thus, this tree $T(s, D)$ is a subgraph of G with root s and a set of nodes D . Let $P_T(s, d)$ represent a path in the tree, $T(s, D)$, from source node s to a destination node $d \in D - \{s\}$. We have the following definitions.

Definition 1. The cost of multicast tree $T(s, D)$ is the sum of the links' cost in the tree. The link cost may represent its monetary cost or resource utilization:

$$\text{cost}(T) = \sum_{e \in E_T} \text{cost}(e). \quad (1)$$

Definition 2. The bottleneck bandwidth of path $P_T(s, d)$ is the minimum value of links' bandwidth along the path, which represents the residual bandwidth of a communication path:

$$\text{BW}(P_T(s, d)) = \min(\text{BW}(e) \mid \forall e \in P_T(s, d)), \quad (2)$$

$$\forall P_T(s, d) \subset T, \quad \forall d \in D.$$

Definition 3. The delay of path $P_T(s, d)$ is the sum of links' delay along the path from s to d . The link delay may include its nodal processing, queuing, transmission, and propagation delays:

$$\text{delay}(P_T(s, d)) = \sum_{e \in P_T(s, d)} \text{delay}(e), \quad (3)$$

$$\forall P_T(s, d) \subset T, \quad \forall d \in D.$$

Definition 4. The delay of multicast tree T is the maximum value of paths' delay in the tree, that is,

$$\text{delay}(T)$$

$$= \max(\text{delay}(P_T(s, d)) \mid \forall P_T(s, d) \subset T, \forall d \in D). \quad (4)$$

Figure 1 depicts an example of a network graph, link parameters, and a multicast tree. Parameters along links are triple (cost, delay, bandwidth). In Figure 1, the source node is Node 1 (i.e., $s = 1$). The destination nodes are nodes 3, 7, and 8 (i.e., $D = \{3, 7, 8\}$). The Steiner tree $T(s, D)$ consists of three paths $P_T(1, 3) = \{(1, 4), (4, 5), (5, 3)\}$, $P_T(1, 7) = \{(1, 4), (4, 5), (5, 7)\}$, and $P_T(1, 8) = \{(1, 4), (4, 5), (5, 8)\}$. The total cost of the Steiner tree can be calculated by (1), that is, $\text{cost}(T) = 20$. By using (2), we can calculate the bottleneck bandwidth of each path along the Steiner tree, that is, $\text{BW}(P_T(1, 3)) = 8$, $\text{BW}(P_T(1, 7)) = 8$, and $\text{BW}(P_T(1, 8)) = 7$. We can also apply (3) to calculate $\text{delay}(P_T(1, 3)) = 9$, $\text{delay}(P_T(1, 7)) = 8$, and $\text{delay}(P_T(1, 8)) = 10$. Finally, we can derive the total delay of the Steiner tree $\text{delay}(T) = 9$ by (4).

2.2. Problem Definition. The optimal multicast tree T depends on the operator objectives, such as network cost, transmission delay, or target QoS [11]. Therefore, solving the multicast routing problem is equivalent to find the optimal distribution tree on the basis of a certain cost function under a given set of constraints. In this paper, the multicast routing in MANETs can be modeled as a combinatorial optimization problem in the following:

$$\begin{aligned} & \text{minimize} \quad \text{cost}(T(s, D)) \\ & \text{subject to:} \quad \text{delay}(P_T(s, d)) \leq \Delta_d \quad \forall d \in D, \quad (5) \\ & \quad \quad \quad \text{BW}(P_T(s, d)) \geq B_d \quad \forall d \in D. \end{aligned}$$

The objective function is to minimize the total cost of the multicast tree. The path-delay constraint enforces that the total delay of each OD pair must be smaller than or equal to its delay bound Δ_d . The minimum bandwidth requirement is denoted as B_d . Therefore, the multicast routing problem is to determine a multicast tree connecting the source node to every destination node such that the cost of this tree is minimum, while the path-delay and bottleneck-bandwidth from the source node to any destination node satisfy the prescribed QoS requirements.

2.3. Routing Table. Since there are so many candidate paths between two nodes in the network graph $G = (V, E)$,

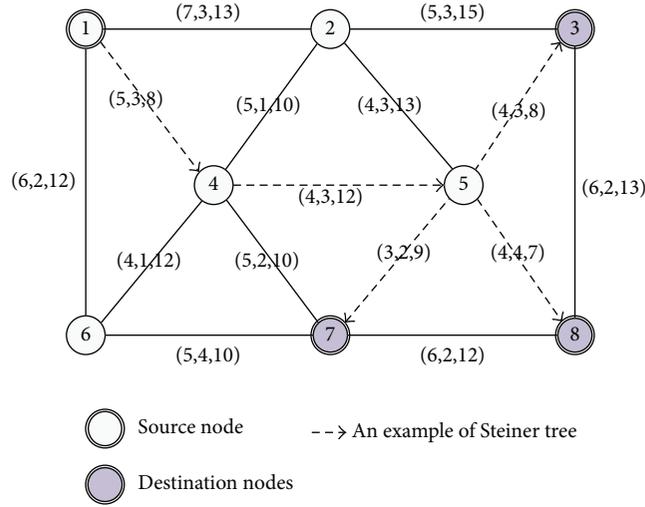


FIGURE 1: An example of a network graph, link parameters, and a Steiner tree.

TABLE 1: An example of routing table for OD pair (1, 7) in Figure 1.

Routing table of path from node 1 to node 7				
Route no.	Route path	Cost of the path	Delay along the path	Bottleneck bandwidth
0	1-4-7	10	5	8
1	1-6-7	11	6	10
2	1-4-5-7	12	8	8
⋮	⋮	⋮	⋮	⋮
$R - 1$	1-2-3-8-7	24	10	12

traditional GAs may consume considerable computational effort in searching infeasible solutions because genetic operations do not always preserve feasibility. Therefore, to reduce the search space, this work uses the K shortest path routing algorithm to precalculate the first R shortest paths and record in a routing table. For the network topology in Figure 1, an example of its routing table for OD pair (1,7) will look like Table 1, which includes the first R shortest paths from node 1 to node 7 with the route path, total cost, aggregate delay, and bottleneck bandwidth.

3. RPGA for Multicasting Routing Problem

The proposed RPGA adopts a RP method to enhance the searching abilities of original GAs for handling constrained multicasting routing problems. To enhance the exploration ability, the RPGA adopts the RST to enlarge the genetic diversity by releasing inefficient constraints and also enforcing efficient ones when the generation number is odd. The flow chart of RPGA (in Figure 2) consists of several genetic operations, such as initialization, selection, crossover, mutation, replacement, and RP method. To enhance the exploitation ability, the proposed RPGA applies the therapeutic crossover to improve the convergence rate during the evolution.

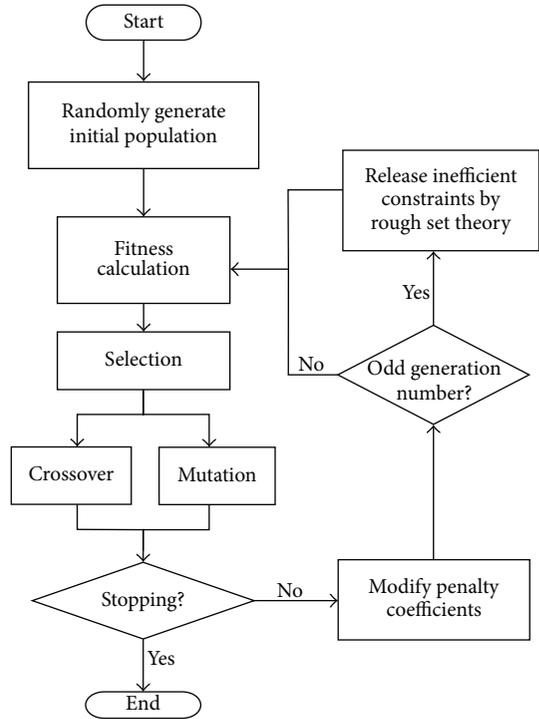


FIGURE 2: Flow chart of the RPGA.

3.1. Encoding and Initialization. In this paper, the encoding method is based on a routing representation for multicast trees. The RPGA maintains a population of chromosomes, which represent a candidate set of Steiner trees for the multicast routing problem. Given a source node s and a set of destination nodes $D = \{d_1, d_2, \dots, d_k\}$, a chromosome can be represented by a string of integers with length k . The chromosome is denoted as $\vec{X} = (x_1, x_2, \dots, x_k)$, where x_i is an integer in interval $[0, R - 1]$ to represent a route number for the OD pair from s to u_i in the routing table. In the

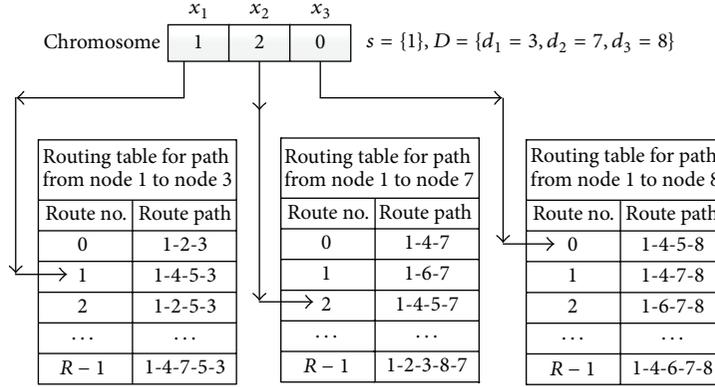


FIGURE 3: Representation of chromosome, genes, and routing table.

example of Figure 1, the second OD pair (i.e., source node 1 to destination node 7) is routed along Path 1-4-5-7. The route number of this path in its routing table (in Table 1) is 2. Thus, we should assign the route number as the value of the second gene, that is, $x_2 = 2$. Therefore, the relationship between chromosome, gene, and routing table for the example in Figure 1 can be illustrated as Figure 3. The RPGA starts with a random population within the gene value interval $[0, R - 1]$, no matter whether these multicast trees satisfy the QoS constraints or not. The emphasis is that the RPGA can find the global optimum for constrained problems even though the initial population is infeasible.

3.2. Fitness Function with Self-Adaptive Penalty Adjustment. The fitness value of each chromosome represents the quality of the corresponding multicast tree (i.e., $\vec{X} = T(s, D)$). However, the penalty adjustment for QoS constraints is difficult to adapt suitably. In this paper, we mix the aspects of the Joines and Houck's DP method [9] with the RST to find a Steiner tree that satisfies the QoS constraints and minimizes total routing cost as well.

The proposed RP method adjusts penalty terms according to both violation magnitude and evolution time. To solve constrained optimization problems effectively, each individual in generation t is evaluated using an expanded objective function (6):

$$\psi(\vec{X}) = \text{cost}(\vec{X}) + \sum_{k=1}^m \left((C \times t)^{\pi(k,t)} \times \max(0, \Phi_k(\vec{X}))^2 \right), \quad (6)$$

where C is a "severity" factor, m is the total number of constraints, and Φ_k is the violation magnitude of constraint k . This fitness function combines a *coefficient* ($C \times t$) with an *exponent* $\pi(k, t)$ to increase penalty pressure over time. For constraint k in generation t , the exponent $\pi(k, t)$ is a representative penalty multiplier that is initially assigned as 2. And then, the penalty multiplier is tuned iteratively according to the discernible mask $\vec{\mu}$ and the representative attribute value γ_k of superior class X_{good} . The exponent $\pi(k, t)$ is

defined as

$$\pi(k, t) = \begin{cases} \pi(k, t-1) \times \gamma_k, & \text{if } \mu_k = 1, \\ \pi(k, t-1), & \text{if } \mu_k = 0, \end{cases} \quad \forall k = 1, \dots, m; \quad \forall t = 1, \dots, \text{MaxGeneration}, \quad (7)$$

$$\pi(k, 0) = 2 \quad \forall k = 1, \dots, m.$$

Remarkably, the discernible mask $\vec{\mu}$ can be used to enable $\pi(k, t)$ by differentiating significant characteristics between classes X_{good} and X_{bad} . If the k th constraint is discernible (i.e., $\mu_k = 1$), the exponent $\pi(k, t)$ is adjusted by the representative attribute value (γ_k); otherwise, the exponent retains the same value as in the previous generation. All these RP coefficients will be introduced in Section 4.

3.3. Selection Operation. A selection operation uses fitness to determine the solution quality and to select high-quality chromosomes for the recombination operation [12]. The RPGA employs a stochastic universal selection to create selection pressure towards the global optimal solution. The measurement of a chromosome's fitness is its value of the expanded objective function in (6).

3.4. Crossover Operation. The crossover operation represents the mixing of genetic material from two selected parents to produce one child chromosome. The RPGA proposes a therapeutic crossover that incorporates a gene-therapy method with a conventional crossover scheme to enhance the exploitation ability and speed up the convergence rate [13].

Each time the selection operation chooses two crossover parents from the population. The therapeutic crossover gives each gene locus an equal chance of being a crossover point (i.e., belongs to a therapeutic genome $i \in G_c$ where $G_c \in \{1, 2, \dots, k\}$). The proposed gene-therapy method evaluates the merit of two selected genomes by comparing the changes in the chromosome fitness before and after interchanging the genomes with the other mating chromosome [13].

```

user is Number of OD pair;
Node is all nodes in the network;
Parents is the chromosomes which be selected;
nKids is the numbers of chromosomes create by crossover;
r1, r2 is the chromosomes which are crossover;
coeC is the number of gene will be crossover;
thisScore is the multi path score;
thisGeneScore is the path score in the multi path;
xoverPoint is genes in the chromosome which be select to crossover;
lengthXP is length of xoverPoint;

Begin
for i = 1:nKids
  Pick chromosomes r1 and r2 from parents;
  coeC is decrease by generations from 1 to 0.2;
  Randomly select xoverPoint which probability small than coeC;

  if thisScore(r1) ≤ thisScore(r2)
    xoverKids(i) = thisPopulation(r1);
    for j = 1 to lengthXP
      if thisGeneScore(r2(j)) < thisGeneScore(r1(j))
        Record path r1(j) as a;
        Record path r2(j) as b;
        if the route paths in a and b have the same intermediate node (m)
          Interchange the sub-routes (s, m) or (m, d) in a and b;
        else
          xoverKids(i, j) = r2(j);
        end if
      end if
    end for
  else
    xoverKids(i) = thisPopulation(r2);
    for j = 1 to lengthXP
      if thisGeneScore(r1(j)) < thisGeneScore(r2(j))
        Record path r1(j) as b;
        Record path r2(j) as a;
        if the route paths in a and b have the same intermediate node (m)
          Interchange the sub-routes (s, m) or (m, d) in a and b;
        else
          xoverKids(i, j) = r1(j);
        end if
      end if
    end for
  end if
end for
End

```

PSEUDOCODE 1: Pseudocode of the crossover operation for routing problem.

According to their relative merit, these two genomes combine to generate a new genome for their offspring. Therefore, offspring inherit more genetic material from the superior genome than the inferior one. We depict the pseudocode of the therapeutic crossover in Pseudocode 1.

3.5. Mutation Operation. A mutation operation used in GAs can increase population diversity to enhance its exploration ability [13]. This work uses the bit-flip mutation with a fixed small probability p_m . According to this mutation probability

p_m , the mutation operation randomly selects a subset of genes and chooses new paths from its routing table. Thus, the route numbers of these new paths in its routing table replace the original values of selected genes. The resulting chromosome is a new multicast tree and can increase population diversity.

3.6. Replacement and Termination. The proposed RPGA adopts a replacement-with-elitism method to prevent best solutions from being lost through a selection process. A successive population is produced from three sources:

(1) the replacement-with-elitism method selects the best 10% chromosomes to join the new population; (2) the crossover operation recombines 80% of child chromosomes; and (3) the mutation operation constructs other child chromosomes for the next generation. The RPGA will stop when it reaches the predefined maximum iterations.

4. RP Method for Constraint Handling

To address the multicast routing problem with QoS constraints, the challenge is how to optimize the objective function value against its constraint violations. Traditional GAs are ineffective in searching feasible solutions because genetic operations do not always preserve feasibility [14]. Therefore, penalty-function methods are the most popular constraint-handling techniques for constrained optimization [15].

The novel RP method has been proposed in our previous work for numerical constrained problems [13]. This proposed RPGA inspired by the Pawlak's RST [16, 17] has been proved better than several existing algorithms for solving a variety of numerical optimization problems [13]. The RP method can automatically adjust penalty coefficients during the evolution. Furthermore, the method does not depend on extra functional analyses for its solution space. Therefore, this study aims to effectively extend the original RPGA as a new constraint-handling technique to address the multicast routing problem with QoS constraints in MANETs. During the genetic evolution, the proposed RP method uses the attribute reduction concept to find appropriate penalizing strategies and release some inefficient constraints.

4.1. Flow of RP Method. The pseudocode of the RP method is depicted in Pseudocode 2. The proposed RP method not only penalizes constraint violations to exploit feasible space but also releases ineffective constraints to explore infeasible space. Therefore, the RP method is a self-adaptive approach that can measure infeasibility and can adjust each penalty coefficient automatically.

4.2. Rough Penalty Classification. This work uses information granulation as a key function for implementing a divide-and-conquer strategy. Elementary information granules are indiscernibility classes of constraint violations. The information system is an information table of attribute values containing rows labeled by objects and columns labeled by attributes [18].

Remark 5. A partition granularity (ρ) is defined for classifying the magnitude of constraint violations. The design principle is that solution quality increases as its constraint penalty moves closer to zero. Therefore, this study uses a smaller range in near-zero regions than in other regions.

4.3. Rough Decision System. A decision system is an IS with the form $DT = (U, A \cup \{d\}, D)$ in which each individual is treated as an object of a nonempty finite set U . Attribute set $A = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$ is a nonempty finite set of conditional attributes, where each *penalty multiplier* (α_k) corresponds to a conditional attribute. The supervised knowledge is expressed

by a decision attribute (denoted by $d \notin A$). An information function D maps each object to a decision attribute, that is, $D : U \rightarrow V_d$ for $d \notin A$, $V_d = \{0, 1\}$. For a minimization problem, the information function is designed as follows:

$$D(\vec{x}_j) = d_j = \begin{cases} 1, & \text{if } f(\vec{x}_j) < f_{\text{average}}, \\ 0, & \text{if } f(\vec{x}_j) \geq f_{\text{average}}, \end{cases} \quad (8)$$

where

$$f_{\text{average}} = \frac{1}{P} \sum_{j=1}^P f(\vec{x}_j). \quad (9)$$

Remark 6. Because the penalty multiplier should be adjusted according to the region of its constraint violation, this work enlarges the penalty multiplier when its violation level increases. In the illustration in Figure 4, the penalty multiplier will be assigned as $(\alpha)^{-2}$, $(\alpha)^{-1}$, $(\alpha)^0$, $(\alpha)^{+1}$, $(\alpha)^{+2}$, and $(\alpha)^{+3}$ for constraint regions 1, 2, 3, 4, 5, and 6, where α denotes the coefficient and the partition granularity (ρ) is six.

4.4. Significant Penalty and Attribute Reduction. Based on the concept of attribute reduction, attributes may not be equally important, and some of them can be eliminated from a decision table without degrading information quality. Attribute reduction can be generalized by introducing attribute evaluation, which can express the merit of each attribute in the information table [19].

Remark 7. Decision attribute d in DT determines a partition $\text{CLASS}(d)$ of object set U , where $\text{CLASS}(d)$ is the object classification with respect to decision attribute d . The minimal subset of penalized constraints is applied to distinguish above-average individuals (i.e., their decision attributes are "1") and below-average ones (i.e., those values are "0"). The representative value of each relevant attribute is assigned as the attribute value with the maximum cardinality in the same class.

5. Computational Experiments

5.1. Test Platform and Parameter Setting. In this paper, the proposed RPGA is evaluated by solving multicast routing problems in MANETs. We use the well-known network generation tool [20] to create an asynchronous network based on the Waxman's techniques [21] and depicted in Figure 5. The network illustrates a random graph in which 40 nodes are connected, and each node has average of four connections to other nodes (i.e., average degree of a node is 4). Each link has its own cost, delay, and bandwidth information. In order to reduce the complexity of the graph representation, Figure 5 only shows the cost/delay information along one direction link (from a smaller ID node to a larger ID one). For example, traffic along the link from node 20 to node 34 will spend 54 units cost and delay 14 msec. In all cases, the maximum number of iterations is 40, the population size is 20, and the number of elite individuals is 2. For each test problem, 30 independent runs with different seeds are performed using the MATLAB environment.

```

conN is mean constraint numbers
popnum is population numbers
IT is information table, which is (conN + 1) * popnum matrix
gp is good population which decision variable = 1 from IT
bp is bad population which decision variable = 0 from IT
gck is good characteristic from gp
bck is bad characteristic from bp
t is the number of current generation
πk(t) is a RP exponent for the t generation
λk is a RP coefficient
C is constraints weight
τ is next generation penalty coefficient
Begin
    Create information table IT;
    //Divide IT to two group gp and bp
    For i = 1 to popnum
        IF decision variable = 1 Pick IT(i) to good population gp;
        Else Pick IT(i) to bad population bp;
    End if
End for
//Find characteristic from gp and bp
For k = 1 to conN
    gck = mod all gpk;
    bck = mod all bpk;
End for
//Find RP coefficient μ
For k = 1 to conN
    If gck = bck λk = gck;
    Else λk = 1;
    End if
End for
//Modify RP penalty exponent π(k, t)
For k = 1 to conN
    πk(t) = πk(t - 1) * λk;
    τk = (C × t) ∧ πk(t), for all k = 1 to conN;
End for
End

```

PSEUDOCODE 2: Pseudocode of the RP method.

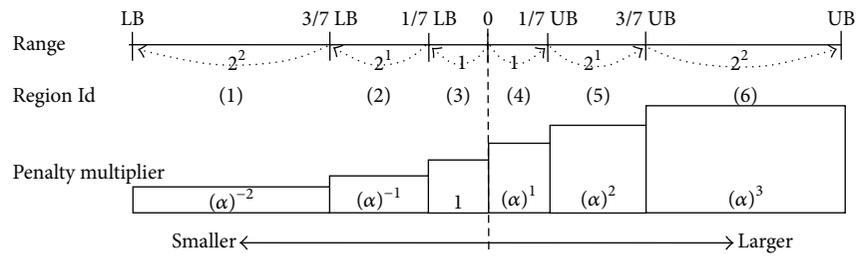


FIGURE 4: Penalty multiplier classification.

5.2. Performance Metrics. In this paper, the performance metrics of solution algorithms consist of (1) the total cost of the obtained multicast tree; (2) the success rate with respect to the QoS constraints; and (3) the required CPU time for computing the multicast routing problems. The success rate (θ_{req}) concerns about the percentage of feasible routes with respect to the QoS requirements. We can define the success

rate as follows [22]:

$$\theta_{req} = \frac{N_{ack}}{N_{req}}, \quad (10)$$

where N_{ack} is the number of OD pairs that satisfy all QoS constraints and N_{req} is the total number of OD pairs in this multicast group.

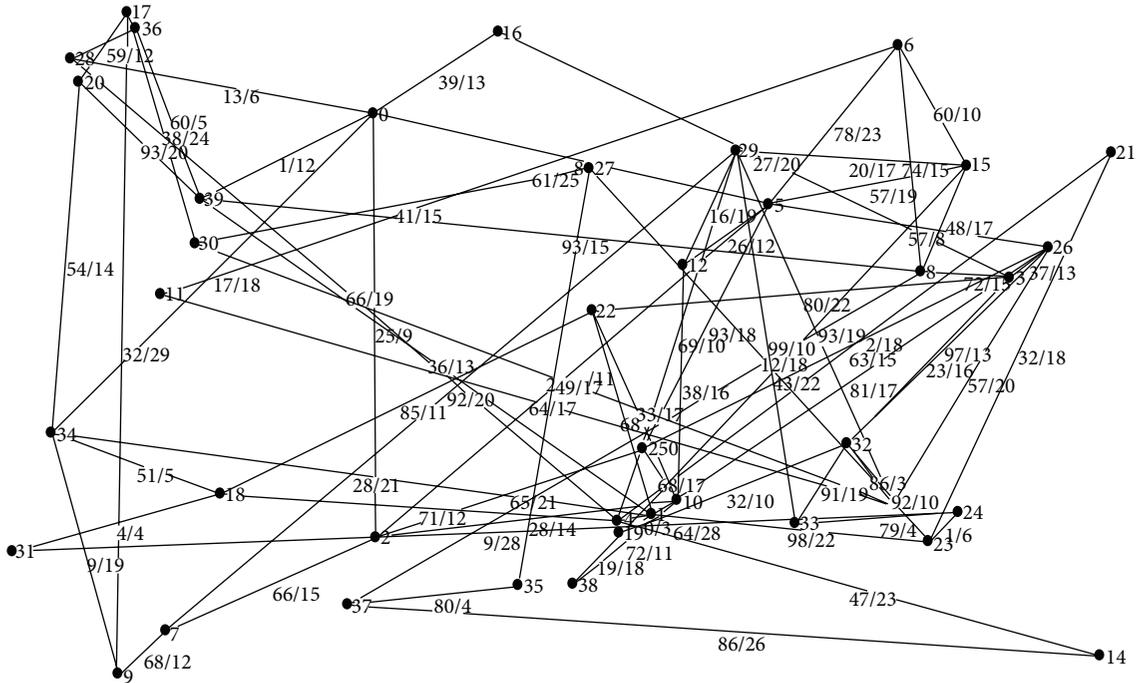


FIGURE 5: A randomly generated network with 40 nodes and average degree four.

TABLE 2: The results obtained by different partition granularities (ρ) when $\alpha = 1.01$.

Partition (ρ)	Cost	Success rate	CPU time
4	1032.9	0.966666667	1.089747617
6	1006.4	0.98245614	1.075849966
8	1021.5	0.970175439	1.085799014

5.3. Algorithm Analyses for Different Parameter Settings. It is well known that the performance of GAs significantly depends on the configuration of its operating parameters. To investigate the impact of various parameter settings in the RP method, this study experiments on two parameters: the partition granularity (ρ) and the penalty coefficient (α). The test network (in Figure 5) has 40 nodes with average degree of 4. We randomly select 20 nodes (50% of total nodes) as the test multicast group, that is, one source node and 19 destination nodes. All OD pairs have the same delay bound for 60 msec.

Firstly, this study conducts 30 runs to find the appropriate partition granularity (ρ) for this problem. When the penalty coefficient is fixed ($\alpha = 1.01$), the experimental results for different partition granularities changing from $\rho = 4$, $\rho = 6$, to $\rho = 8$ are shown in Table 2. We normalize these results relative to that of $\rho = 6$ and show the percentage comparison in Figure 6. Obviously, the RPGA with 6 partitions can achieve better results than other partition settings with respect to all the three performance metrics.

Secondly, when the partition granularity is given ($\rho = 6$), the experiments on different penalty coefficients ($\alpha = 1.0001, 1.001, 1.01, 1.1, \text{ and } 10$) are tested for 30 runs. The experimental

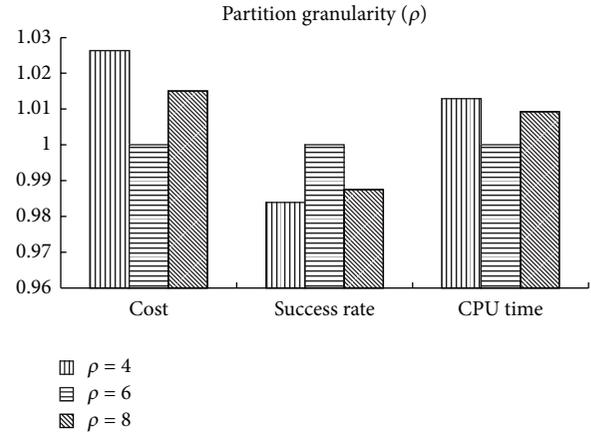


FIGURE 6: Comparison with different partition granularities (in percentage relative to $\rho = 6$).

TABLE 3: The results obtained by different penalty coefficients (α) when $\rho = 6$.

Coefficient (α)	Cost	Success rate	CPU time
10	1022.833333	0.989473684	1.084000564
1.1	1031.966667	0.978947368	1.068427877
1.01	1006.4	0.98245614	1.075849966
1.001	1031.7	0.975438596	1.065008048
1.0001	1022.333333	0.973684211	1.059503391

results on the average of cost, success rate and computing time are depicted in Table 3. The normalized percentages relative

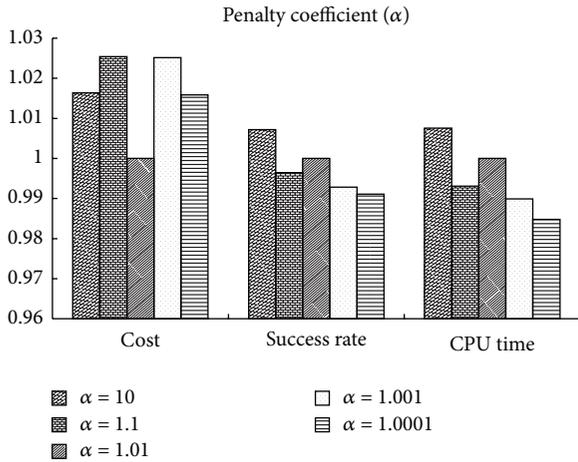


FIGURE 7: Comparison with different penalty coefficients (in percentage relative to $\alpha = 1.01$).

to $\alpha = 1.01$ are shown in Figure 7. Noticeably, the RPGA with $\alpha = 1.01$ can achieve better results than other settings on all the cost, success rate, and computing time metrics.

5.4. Comparison with Other Existing Methods. The performance of the proposed RPGA is compared with two well-known penalty methods, which are the Wang's penalty (WP) method [22] and the DP method [9] for three kinds of multicast scenario. In the first test scenario, the test network has 40 nodes with average degree of 4, which is the same as Figure 5. The multicast group includes 20 OD pairs. All OD pairs have the same delay bound for 60 msec. We execute each method for 30 independent runs and report the experimental results in Table 4.

For the mean cost in Table 4, the RPGA can find the best result in average 30 runs. Furthermore, the standard deviation of the routing cost obtained by the RPGA is smaller than that by the DP and similar to that of the WP. That is, the RPGA can reliably find the minimum-cost Steiner tree. Compared with the success rate, the RPGA is the best method with respect to the mean and standard deviation of the success rate, even though the WP has a high probability of converging on infeasible solutions. That is, the proposed RPGA can succeed in finding feasible and minimum-cost solutions. From the CPU-time metric, the comparison can help us to realize how much time complexity is needed to compute the RP method because the RPGA is enhanced from the DP method. In Table 4, the RPGA only spends a little computing effort (about extra 6.7% computing time) to obtain better results than the DP method.

For comparison, all the results are normalized as percentages relative to the results of the RPGA (in Figure 8). Obviously, the performance comparison shows that the RPGA can find the minimum-cost multicast tree effectively (with higher success rate) and efficiently (with lower computing effort) than the other two methods.

The evolution curves of all three methods on the total cost and path delay are shown in Figures 9(a) and 9(b), respectively. In Figure 9(a), we can observe that both the

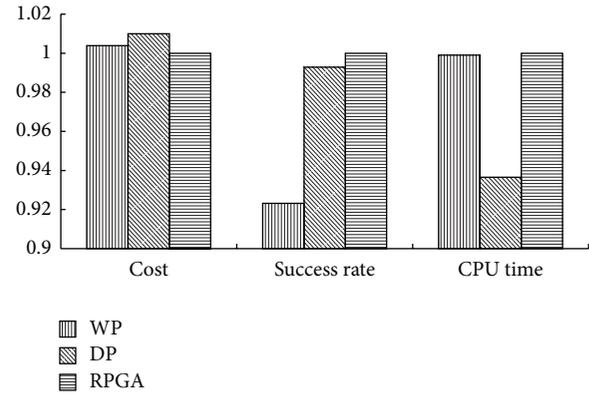


FIGURE 8: Comparison with different penalty methods (in percentage relative to the RPGA).

DP and RPGA methods rapidly converge on the low-cost results after 8 generations; however, the WP method takes 13 generations to slowly converge on a high-cost one. From the path delay aspect in Figure 9(b), all the three methods can satisfy the delay bound. The RPGA, DP, and WP methods take 2, 2, and 4 generations to achieve feasible solutions, respectively. Therefore, the proposed RPGA has a similar convergent effect with the DP. Both the RPGA and DP method outperform the WP with respect to the effectiveness and efficiency abilities.

In the second test scenario, we randomly generate 8 test networks with the numbers of nodes from 10 to 80 to mimic the stress test for these three methods. In those tests, all delay constraint bounds are 60 msec and the multicast group size is 50% of network nodes. When nodes number increases, the network overhead increases obviously and end-to-end delay increases at the same time. The experimental results depicted in Figure 10 are normalized as percentages relative to the result of the RPGA for comparison. Compared with Figures 10(a) and 10(b), we can observe that RPGA can find the minimum cost of feasible multicast tree, even though the success rates of the DP in the 60-node network and the WP in almost all networks are less than 90%. The WP is the worst method in these three methods on the success rate. However, the computing effort of the RPGA is higher than the other two methods in Figure 10(c). We can also find that the more number of network nodes in the test scenarios, the closer computing time needed for all three penalty methods.

In the third test scenario, we change the delay-bound requirements from 20 msec to 90 msec in a test network, which has 40 nodes and its multicast group size is 20. The comparisons between the success rates and the delay bounds are shown in Figure 11. The success rate of the RPGA is similar to that of the DP and is better than that of the WP, especially, when the delay bounds are lower than 60 msec.

6. Conclusions and Future Works

The proposed RP method cooperates with GAs for dealing with QoS-based multicast routing problems. The principle of the RP method is that the RPGA releases/enforces some

TABLE 4: The results of different penalty methods.

Penalty methods	Cost (mean)	Cost (st. dev.)	Success rate (mean)	Success rate (st. dev.)	CPU time
WP	1010.366667	82.23745406	0.907017544	0.077799684	1.074814229
DP	1016.5	101.1191683	0.975438596	0.033095275	1.00755095
RPGA	1006.4	83.13910238	0.98245614	0.028772225	1.075849966

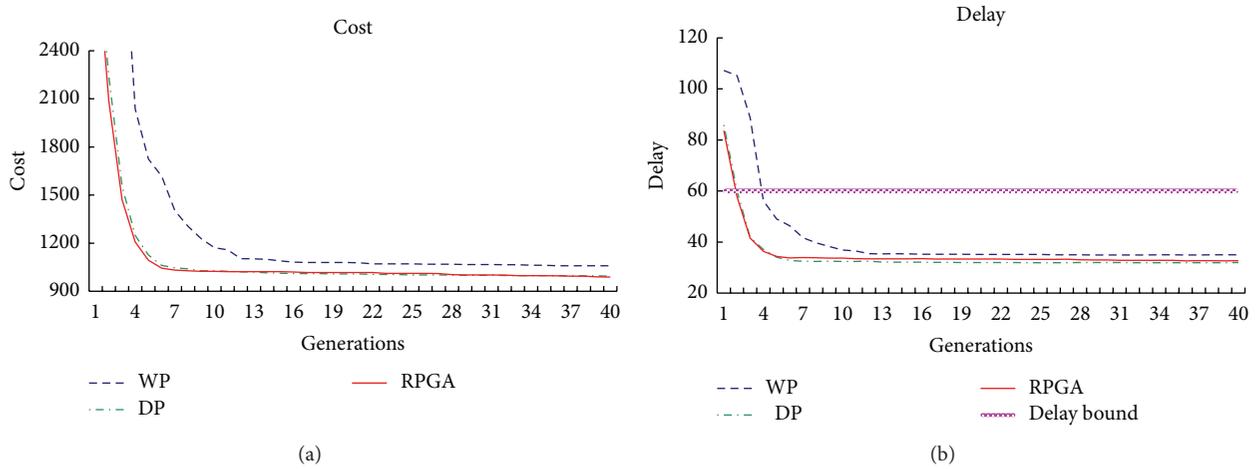


FIGURE 9: Evolution curves of three methods on the performance metrics: (a) cost and (b) delay.

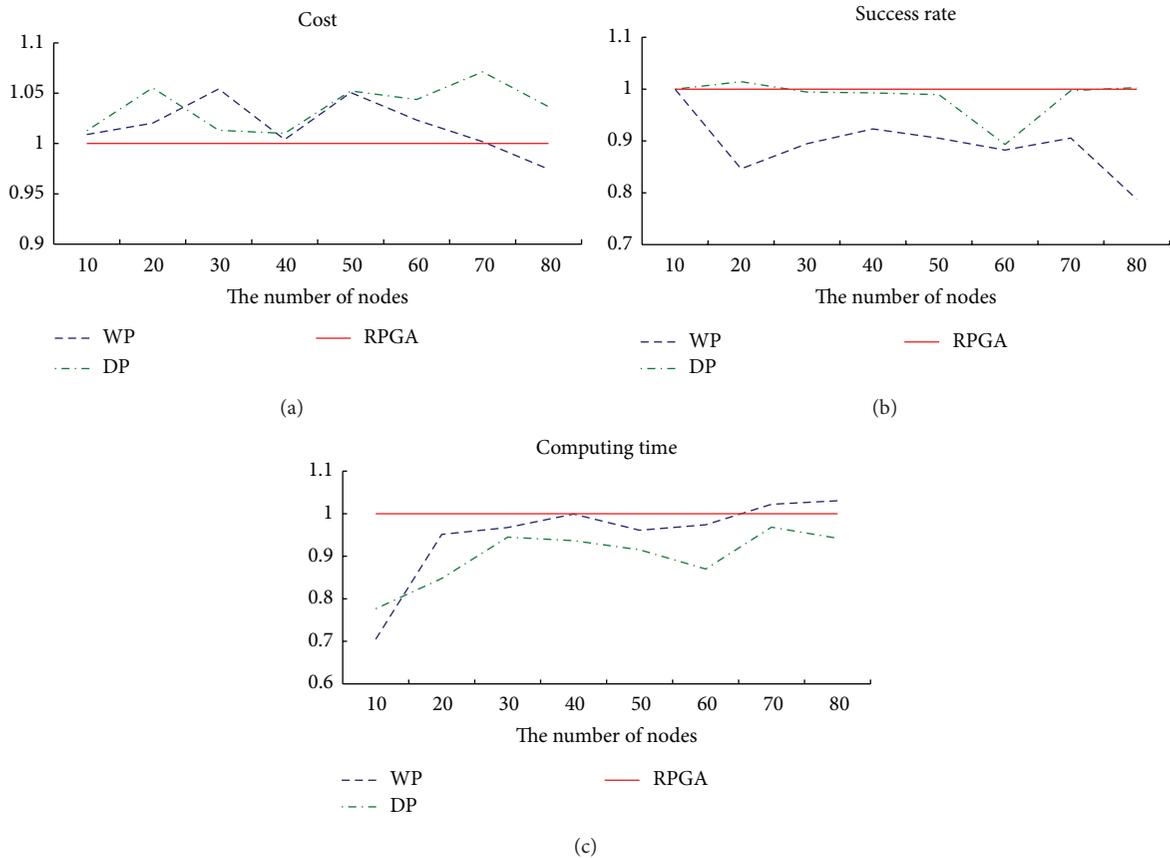


FIGURE 10: Comparison with three methods for different numbers of network nodes in percentage relative to the RPGA on (a) cost, (b) success rate, and (c) CPU time.

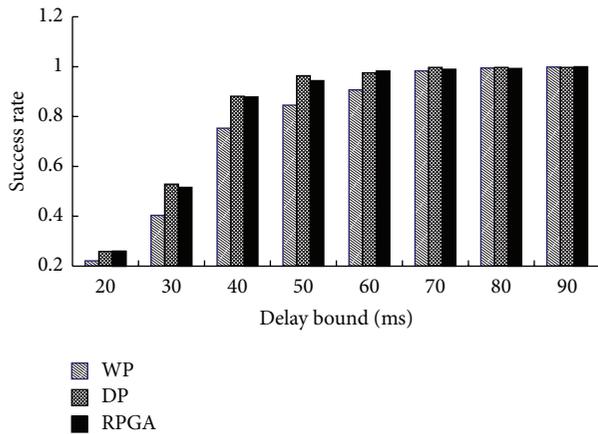


FIGURE 11: Comparison with three methods on the success rates for different delay bounds.

penalties on inefficient/efficient constraints during evolution. Importantly, this approach can find the optimum or near-optimal solutions even though the initial population includes infeasible solutions. The performance of the proposed algorithm was measured using three kinds of test scenarios and compared with two state-of-the-art methods. Experimental results indicate that the proposed RPGA can find near-optimal solutions and outperforms two existing methods for constrained multicast routing problems. The proposed algorithm is also robust in obtaining feasible solutions of all the test functions even though the WP method has smaller success rates for some difficult problems. In conclusion, the performance assessment also demonstrates that the proposed RP method has a remarkable capability to balance the objective function and the constraint violations as an effective and efficient method for solving a variety of QoS-based multicast routing problems.

We have observed that the computing effort of the RPGA is higher than that of other two penalty methods in small-scale networks. In the future, we will study the scalability of the proposed RPGA in finding multicast routes for dynamic MANETs with large-scale dimension. Since MANETs allow ubiquitous service access without any fixed infrastructure, developing a distributed algorithm for high mobility environments is also our future work.

References

- [1] M. R. Gareg and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*, W.J. Freeman, New York, NY, USA, 1979.
- [2] C.-H. Liu, T.-C. Chiang, and Y.-M. Huang, "A near-optimal multicast scheme for mobile ad hoc networks using a hybrid genetic algorithm," in *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, vol. 1, pp. 465–470, April 2006.
- [3] S. Gangwar, S. Pal, and K. Kumar, "Mobile ad hoc networks: a comparative study of QoS routing protocols," *International Journal of Computer Science & Engineering Technology*, vol. 2, no. 1, pp. 771–775, 2012.
- [4] M. Hamdan and M. E. El-Hawary, "A Novel Genetic Algorithm Searching Approach for Dynamic Constrained Multicast Routing," in *Proceedings of the Canadian Conference on Electrical and Computer Engineering: Toward a Caring and Humane Technology (CCECE '03)*, pp. 1127–1130, May 2003.
- [5] H. T. Tran and R. J. Harris, "Solving QoS multicast routing with genetic algorithms," in *Proceedings of the International Conference on Information Communications and Signal Processing*, vol. 3, pp. 1944–1948, 2003.
- [6] S. Sumathy and E. Sri Harsha, "Survey of genetic based approach for multicast routing in MANET," *International Journal of Engineering and Technology*, vol. 4, no. 6, pp. 474–485, 2013.
- [7] M. Gen and R. Cheng, "Survey of penalty techniques in genetic algorithms," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '96)*, pp. 804–809, May 1996.
- [8] A. Homaifar, C. X. Qi, and S. H. Lai, "Constrained optimization via genetic algorithms," *Simulation*, vol. 62, no. 4, pp. 242–254, 1994.
- [9] J. A. Joines and C. R. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs," in *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, vol. 2, pp. 579–584, June 1994.
- [10] A. B. Hadj-Alouane and J. C. Bean, "A genetic algorithm for the multiple-choice integer program," *Operations Research*, vol. 45, no. 1, pp. 92–101, 1997.
- [11] A. T. Haghghat, K. Faez, M. Dehghan, A. Mowlaei, and Y. Ghahremani, "GA-based heuristic algorithms for bandwidth-delay-constrained least-cost multicast routing," *Computer Communications*, vol. 27, no. 1, pp. 111–127, 2004.
- [12] M. Zhang, W. Luo, and X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," *Information Sciences*, vol. 178, no. 15, pp. 3043–3074, 2008.
- [13] C. H. Lin, "A rough penalty genetic algorithm for constrained optimization," *Information Science*, vol. 241, pp. 119–137, 2013.
- [14] D. Powell and M. M. Skolnick, "Using genetic algorithms in engineering design optimization with non-linear constraints," in *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 424–431, 1993.
- [15] Z. Michalewicz, "A survey of constraint handling techniques in evolutionary computation methods," in *Proceedings of the 4th Annual Conference on Evolutionary Programming*, pp. 135–155, 1995.
- [16] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.
- [17] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Information Sciences*, vol. 177, no. 1, pp. 3–27, 2007.
- [18] Z. Pawlak and A. Skowron, "Rough sets and Boolean reasoning," *Information Sciences*, vol. 177, no. 1, pp. 41–73, 2007.
- [19] S. O. Kimbrough, G. J. Koehler, M. Lu, and D. H. Wood, "On a Feasible-Infeasible Two-Population (FI-2Pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch," *European Journal of Operational Research*, vol. 190, no. 2, pp. 310–327, 2008.
- [20] H. Salama, "The multicast routing simulator," The Real-Time Communication Project, Version 2, 1997, <http://rtcomm.csc.ncsu.edu/index.htm>.
- [21] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, 1988.
- [22] Z. Wang, B. Shi, and E. Zhao, "Bandwidth-delay-constrained least-cost multicast routing based on heuristic genetic algorithm," *Computer Communications*, vol. 24, no. 7-8, pp. 685–692, 2001.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

