



Ticket-based QoS routing optimization using genetic algorithm for WSN applications in smart grid

Uthman Baroudi¹ · Manaf Bin-Yahya²  · Meshaan Alshammari¹ · Umair Yaqoub¹

Received: 7 November 2017 / Accepted: 5 June 2018 / Published online: 14 June 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

Wireless sensor network (WSN) information network in Smart Grid is envisioned to handle diversified traffic such as real-time sensitive data and non-real-time traffic. Therefore, QoS routing protocol in smart grid network is essential. Ticket-based routing (TBR) protocol is a promising protocol because it can select routes based on several desired metrics, for example route cost and delay. However, the original TBR suffers the need for transmitting a huge number of tickets to probe the sensor network and discover the path cost and delay. Genetic algorithm can be used to minimize the number of tickets as well as discovery messages overhead. In this work, we implement genetic algorithm (GA-TBR) at the source sensor node to collect the state information inside the WSN environment of Smart Grid and hence optimize the selection of routes to ensure the required QoS. Extensive simulation experiments have been conducted to investigate the performance of GA-TBR. The simulation results have shown that with few tickets, the proposed algorithm is able to select routes with minimum possible delay and shows 28% improvement compared to ad hoc on demand distance vector routing (AODV) protocol.

Keywords Smart Grid · WSN · QoS Routing · Genetic Algorithm · Ticket-Based Routing · Route Discovery Optimization

1 Introduction

SMART Grid has arisen as a new conception of next-generation electricity grid for better efficiency and reliability through automated control and modern communications technologies. Smart Grid aims at enhancing the existing electric power grid infrastructure by building an overlay communication and computing network that will increase the capacity and flexibility of the power grid (Yoldas et al. 2017). One of the main features of Smart Grid is that supply and demand will be automatically controlled via two-way communication among different nodes.

Smart Grid is expected to replace the conventional power grids in the coming years. In some parts of the world, they already form a prominent part of the power supply. Thus, different domains and elements are defined and specified in the new power system. The main domains are generation, transmission, distribution, and customer domains (Gungor et al. 2011). Figure 1 depicts an overall architecture of smart grid; multiple sensors and actuators are distributed overall the smart grid. Moreover, these domains and elements can talk with each other in a large communication system to achieve the requirements of Smart Grid such as efficiency, reliability, flexibility, and demand response. Furthermore, Smart Grid attempts to benefit from the development of Advanced Metering Infrastructure (AMI) as a smart meter system, Wireless Sensor Network (WSN) as a monitoring system and renewable energy systems (Tuballa and Abundo 2016). AMI's are smart meters that are more capable than the conventional energy consumption counters. These smart meters are installed at the electricity supplier as well as the location of the various consumers. Therefore, smart grid poses several challenges to manage energy consumption and related costs in the side of smart homes such as profiling (Gentile et al. 2016a, b), secure billing and financial

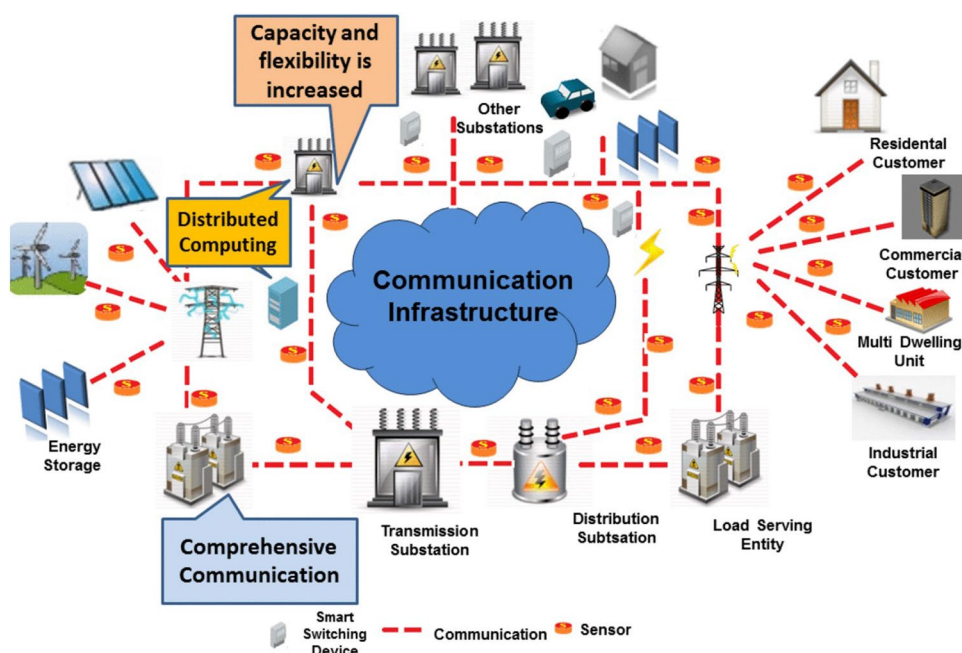
This study was funded by the deanship of scientific research at King Fahd University of Petroleum and Minerals, under grant# RG1319-1.

✉ Uthman Baroudi
ubaroudi@kfupm.edu.sa

¹ Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

² Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada

Fig. 1 Smart grid architecture
(Gungor et al. 2011)



process (Yang et al. 2013), and online social networks effects (Kamilaris et al. 2012).

The effect of a failure of one node or zone in any of Smart Grid domains can comprise the single failure segment, other segments, or the entire system (Rekik et al. 2017). In the current power systems, monitoring systems engage in limited assessments do not provide any automated handling of monitoring data where monitoring devices do not interconnect with each other and provide a finite collection of data. The sensing system can monitor electrical and non-electrical parameters. For example, the monitored parameters for transmission line are weather conditions and wire state to determine the safety level of transmission operation. Furthermore, adding more electrical and mechanical parameters can support overloading control by determining the dynamic line rating (DLR). Also, providing electrical parameters such as current and voltage can help for decisions in real time that improves the power system efficiency (Mazur et al. 2017).

Several communications technologies do exist for data exchange among monitoring and control components through wired and/or wireless media. Wired communication is the most media used in the existing power systems such as Ethernet and fiber optic (Fateh et al. 2013). Wireless networks provide a promising solution with the development of WSN due to inexpensive infrastructure and ease of deployment for unreachable and isolated areas. ZigBee (Fadel et al. 2015) is one of these wireless technologies that provide low cost and ease of setup and implementation with addition of low power usage.

Summing up, WSN can provide vital monitoring data that is used in Smart Grid assessment besides automated decisions. In addition, deploying WSN guarantees reliable

and flexible solutions for providing sensing and control over Smart Grid system. WSN features support low setup, maintenance, and upgrading costs. Additionally, WSN functioning has the advantage over the traditional systems in case of self-organization, coverage, and online responses. The WSN monitoring system can be established across all Smart Grid domains. For instance, WSN can monitor power quality, renewable energy farms, and distributed generation at the generation domain. In addition, detecting any failure in transmission and distribution domains can improve Smart Grid stability. This failure can occur as a power outage or in overhead or underground transmission lines for several reasons such as overheating, natural disasters, animals, theft, etc (Fadel et al. 2015).

On the other hand, several challenges raised from utilizing WSN in Smart Grid such as security, quality of service (QoS) provisioning and transmission overhead. WSN system encounters many security threats due to its open environment (Chhaya et al. 2017). While solutions to overcome its vulnerabilities are not feasible due to its limited resources. Different solutions are proposed to solve security issues of WSN in Smart Grid with specific design requirements. Moreover, reliable communication with minimum latency and adequate bandwidth (Ancillotti et al. 2013) is required to deliver real-time information inside Smart Grid and it should provide QoS depending on the type of data transmitted through the WSN network. QoS routing protocol in WSN-Smart Grid networks must select paths depending on several QoS metrics to achieve certain requirements. However, obtaining best paths to meet QoS constraints requires many control messages, which will increase the background traffic. Hence, delay and congestion in data network will

grow as routing overhead increases. For this reason, route discovery background traffic must be reduced.

Ticket-based routing (TBR) (Xiao et al. 2002) is a well-known method for finding optimal routes in ad-hoc networks. It is applied by broadcasting multiple probes from the source to a destination via multiple routes to find the best path. However, the number of sensor nodes is expected to increase considerably and link states keep changing due to mobility and sporadic nature of the connection. Therefore, TBR becomes a highly complicated, cost-ineffective process (Levis et al. 2009). As the amount of information transferred is expected to increase significantly in the near future with services like video surveillance that require high QoS, a more efficient approach is required. Therefore, it is very important to design a robust efficient routing algorithm that is capable of providing high QoS to Smart Grid networks traffic.

Recently, route discovery optimization in ad hoc networks environment is given a significant attention using metaheuristic algorithms such as genetic algorithm. There are numerous research works that use genetic algorithm in order to shrink routing overhead in data networks as (Yen et al. 2011; Zafar and Soni 2014; Cheng and Yang 2010b). The main motivation for such direction is the great potential of metaheuristic approaches to find near-optimal solution in reasonable running time. This feature is very desirable in industrial applications. In this work, we follow the same path and develop a genetic algorithm to optimize route discovery overhead using TBR protocol. At the source node, the genetic algorithm is applied to find optimal paths that satisfy the QoS requirement. In order to discover the optimal paths, paths from route replies are used as initial population and the fitness function depends on the required QoS metrics. In this paper, we use delay as QoS metric just to show an example. However, using TBR protocol, any quantitative QoS metric which can be accumulated could be used as a fitness function such as throughput and delay jitter. The main contributions of this work can be summarized in the following points:

- Proposing an enhanced GA-TBR algorithm that can select routes according to a predefined set of QoS of requirements with minimal probing tickets and computational complexity.
- Exploiting the routing cache in every sensor node to construct a connectivity matrix that will be employed to validate the generated offsprings. This feature is very essential part of our algorithm that reduces the overhead messages as well as the required number of tickets.
- Performing an in-depth analysis of certain features of genetic algorithm such as validity checking and the fitness function.

- We have compared our proposed approach GA-TBR with AODV which is adopted by IEEE 802.11s and GA-TBR shows 28% improvement on average.

In addition, the proposed algorithm can lend itself to other applications such as social networking based smart grid where each social group can constitute one node in the whole network graph (Huang et al. 2015). For example, users can exploit their social network to facilitate sending grid outage alerts notification and visualization, repair work schedules as well as brand building (Moreno-Munoz et al. 2016).

This paper structure is organized as follows. Section 2 summarizes the existing work about QoS protocols in Smart Grid and researches that use genetic algorithm for routing optimization. Section 3 starts by describing the TBR protocol in 3.1 along with a detailed example in 3.2. The proposed genetic based algorithm is described in details in section 3.3. Section 4 presents the simulation models and discusses the results. Then, analysis of our proposed genetic based approach and the performance evaluation of TBR along with the genetic algorithm are provided in Sect. 5. Finally, Sect. 6 concludes the paper and provides suggestions for the future work.

2 Related works

Saputro et al. 2012 have provided a survey of the existing routing protocols in the Smart Grid communications infrastructure. They extensively studied and analyzed the advantages and disadvantages of the proposed protocols with respect to different purposed areas. They argued that there is not enough research on QoS routing in the Smart Grid networks. In addition, different QoS requirements of WSN applications in smart grids must be addressed such as reliability and latency (Gungor et al. 2013). Nevertheless, there are several surveys studied the impact of WSN deployment in Smart Grids (Rekik et al. 2017; Fadel et al. 2015; Wang et al. 2015; Usman and Shami 2013). Current researches show that the development of WSN pushes towards addressing monitoring and control challenges in the new power systems and provide an immense potential to satisfy its communication requirements. In the rest of this section, we provide a brief literature review about existing WSN applications in Smart Grid networks and its challenges. Also, the QoS communication issues and possible generic algorithms based solution are discussed for such environments.

Fateh et al. (2013) proposed a hybrid wireless structure for the transmission line monitoring. The hierarchical architecture aims to reduce the setup and functioning costs without dimming the communication requirements such as delay and bandwidth. However, using cellular nodes to relay data to the control centers raise the cost of this network design. However,

there is a tradeoff between network cost and transmission delay (Li et al. 2016). Therefore, Li et al. (2016) studied the number of cellular modules and their positions in order to achieve low network cost associated with low transmission delay.

Kovendan and Sridharan (2016) proposed a WSN implementation in a Smart Grid model that provides a monitoring of transmission and generation side. Kurt et al. (2017) proposed a novel method to calculate the optimal packet size of WSN for Smart Grid applications. This technique tries to specify real-time channel characterization in order to obtain packet size optimization measurements for WSN in Smart Grid. He et al. (2017) studied security mechanisms for WSN applications of Smart Grid. Authors analyze cyber security risks that such environments are vulnerable to by providing security assessment and vulnerabilities classification. While in (Yan et al. 2017), a watermarking-based security model is proposed to attain security requirements with low operational cost.

Vallejo et al. (2012) described how a QoS broker can be used to enhance the QoS of smart grid communication by providing QoS management in a centralized and standardized manner to meet their strict requirements. QoS broker devices can update certain parameters of layer 3 and layer 2 networks to improve the effectiveness of end to end QoS through Smart Grid communication. Li and Zhang (2010) proposed optimized multi-constrained routing (OMCR) protocol which is a greedy algorithm implemented for secure QoS routing protocol and satisfy real-time system requirement that can handle the impact of communication metrics. Two QoS parameters namely delay and outage probability are used in this multi-constrained QoS routing protocol. The authors assumed that a home appliance can communicate with the control center by sending a QoS requirement and then the control center assigns one or more routes for the home appliance to guarantee the QoS requirement.

On the other hand, the genetic algorithm can be used in ad-hoc network protocols for optimizing the route discovery process (Sara and Sridharan 2014). A multi-cast tree construction for QoS routing mechanism for mobile ad hoc network is presented in (Lu and Zhu 2013) to moderate end-to-end delay and total energy cost using genetic algorithm. A genetic algorithm is applied on delay constrained source based mechanism to reduce route selection power consumption as well as delay. Zaballos et al. (2013) proposed a new QoS protocol based on TBR that uses genetic algorithm for minimizing route discovery traffic. Multi-paths are probed at the same time and randomly forwarded across the network to return paths that meet QoS constraints. The maximum initial population is the number of tickets issues to establish a connection. After the mutation operation, the node sends a probe with a single ticket to validate the mutated routes causing an increase in message overhead and longer delay. In contrast, genetic algorithms have been applied

to improve routing performance but not in the context of QoS (Ahn and Ramakrishna 2002). However, an efficient method to check the validity has still not been specified. Furthermore, the methods described to discover crossovers and mutations are ineffective because they require longer convergence time. Furthermore, genetic algorithm is used to solve dynamic shortest path routing problem in mobile ad hoc network (Yang et al. 2010), wireless mesh network (Jiang et al. 2010). Immigrants and memory schemes are proposed to enhance genetic algorithm for dynamic QoS routing problem (Cheng and Yang 2010a). Jiang et al. (2010) proposed QoS routing optimization problem solution for wireless mesh networks using hybrid genetic algorithm and ant colony optimization algorithm.

3 Genetic ticket-based routing protocol

In order to have a better understanding of our proposed approach, we start first with introducing the main concepts of TBR along with a detailed example. Then, we describe the genetic ticket-based routing protocol (GA-TBR).

3.1 Ticket-Based Routing Protocol

Ticket-based routing (TBR) (Chen and Nahrstedt 1998) is an on-demand routing protocol such as dynamic source Routing (DSR) and ad hoc on demand distance vector routing (AODV). On-demand routing protocol does not keep updated route tables with the most recent route topology. Enhanced ticket based routing (ETBR) protocol (Xiao et al. 2002) improved the searching ability of discovery method in original TBR. However, the number of tickets used to find a route is still the same. In other words, ETBR causes the same control message overhead as original TBR protocol.

In TBR, when a sensor wants to send data, it should discover a route based on certain metrics like cost and delay. This sensor starts sending probes with a certain number of tickets to all neighbor sensor nodes. In contrast to DSR, intermediate sensor nodes do not flood these probes to its neighbors; each sensor distributes tickets depending on a distribution method defined by many parameters such as the number of tickets received and historical probes information records. In addition, each sensor node adds its address to (source to destination) path and accumulates the path cost. When a probe reaches the destination or a sensor node that have a route to the destination on its route cache, this node sends a route reply (similar to PREP in DSR) containing the source to destination path with its cumulative cost and delay. Moreover, each sensor node has a route cache that stores routes, which have been discovered before by sending probes or learned from route discovery messages of others as intermediate nodes. The sensor node instead of originating a

new request to discover route or forwarding others requests, it can use a route from its cache. Using historical information probes records can eliminate redundant paths to reach the destination in addition to eliminating path cycles. In this case, intermediate nodes pass only one probe message from the same sender node with the same sequence number of the route request.

Generally, once a sensor generates a probe, it can split it into a number of probes and then distribute them to its neighbor nodes. Furthermore, every probe can be divided into more probes as long as there is enough quantity of tickets on this probe. Each probe split and forwarded until it reaches the destination unless it gets dropped by an intermediate node due to QoS constraints or optimizing reasons. Each probe reaches the destination can obtain possible routes. Then, the destination node can handle only routes which satisfy certain QoS requirements. For example, only paths with an accumulated delay less than certain delay threshold are considered as feasible paths. Since probing process will increase messages load inside the network and consume a lot of bandwidth, further optimization is needed. We propose to use genetic algorithm (described in Section 3.3) at the source node and minimize the number of tickets. In this work, we assume that every sensor has its own neighbor list obtained by link level protocol. Furthermore, any changes in topology such as the occurrence of a new sensor/actuator or losing connection with a sensor node should be updated within a finite time.

3.2 Detailed TBR example

Now, we will provide a detailed example of TBR route discovery. Let assume a network of 8 sensors shown in Fig. 2 where sensor 0 wants to discover a route to sensor 7. First, sensor 0 initiates a probe with a certain finite number of tickets ($N = 4$). $P_{n,\{path\}}$ is a general notation for a probe in this paper, where P denotes Probe, n denotes the number of

tickets handled by this probe, and $\{path\}$ denotes the accumulated path. Sensor 1 is the only neighbor of the source sensor. Hence, sensor 0 forwards the probe $P_{4,\{0\}}$ to sensor 1 with all number of tickets. After receiving $P_{4,\{0\}}$, this probe is split into three new probes and forwarded to its neighbors. When a probe reaches sensor 4, this sensor stops forwarding probes as a result of having a path to sensor 7 in its route cache thus a reply $R_{1,\{0,1,4,3,6,7\}}$ is generated and sent back to sensor 0. At sensor 5, $P_{1,\{0,1,3\}}$ is received before $P_{1,\{0,1,2\}}$, therefore, the probe which arrives later will be discarded by this sensor to minimize probing overhead and eliminate infinite cycles probing. Later, $P_{1,\{0,1,3,5\}}$ probe arrives the destination sensor within delay constraint; sensor 7 sends a route reply $R_{2,\{0,1,3,5,7\}}$ that contains the accumulated cost and delay to the source sensor. Finally, two feasible routes are discovered from source and destination $\{0, 1, 3, 5, 7\}$ and $\{0, 1, 4, 3, 6, 7\}$. Now, $\{0, 1, 3, 5, 7\}$ and $\{0, 1, 4, 3, 6, 7\}$ are used by GA as initial population. Let's assume that gene 3 is randomly selected as a crossing point. A new path of $\{0, 1, 3, 6, 7\}$ results from the crossover operator and no validity check is required because the crossover is applied to a common gene. This new resulted path can be an optimal route from source to destination.

However, the number of probes and their tickets can still be minimized and hence the overall discovery messages overhead. In other words, an optimal route can be discovered using genetic algorithm even with reduced number of tickets (N).

3.3 The proposed algorithm

Genetic algorithm (Goldberg 1989) is a search algorithm for optimization, which is used to find a near-optimal solution through a search space using operations inspired by natural genetics. Genetic algorithm improves the quality of its initial population and produces a new population of high-quality outcomes where each outcome can solve the problem and the outcome with the highest factor value is the optimal solution. This algorithm has been proven theoretically and empirically to be effective in complex spaces by providing efficient searching mechanism (Dengiz et al. 1997; Krasnogor and Smith 2005; Liu et al. 2018). Nevertheless, it is not guaranteed that the produced offsprings are always optimal routes; but it is acceptable given the reduced computational cost.

In our proposed method, the genetic algorithm is used to optimize the route discovery using TBR. At the source sensor, the genetic algorithm is applied to find optimal paths that satisfy the QoS requirement. We achieve this goal by using paths constructed from probe tickets' replies as initial population. Each possible path from a probe ticket reply represents a chromosome, which consists of a number of genes. Then, the fitness function is set to evaluate each chromosome from the population.

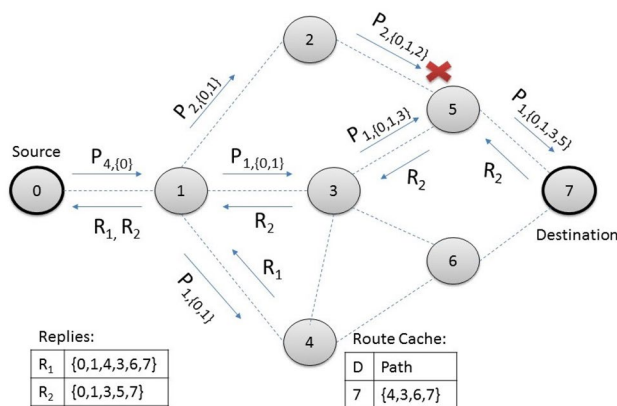


Fig. 2 Example of TBR route discovery in 8-sensor nodes network

3.3.1 Initial population generation algorithm

In this initial stage, original TBR is employed in order to probe and collect information about network connectivity and other credentials such as delay, bandwidth, congestion, etc. Algorithm 1 depicts the pseudo code for this stage. More details are provided in the simulation setup section.

using crossover operator by partially swapping their genes around a common gene unless swapping happens around randomly, it then picks crossing points. When the crossover happens around a common gene, no validity checking is required.

- Mutation: in this operation, mutation operator mutates the new offspring's by randomly altering the value of

Algorithm 1 Initial Population Generation Algorithm

Result: Connectivity Matrix and set of paths

Initialization: Given the i node with j out branches, N tickets need to be sent

```

while Total tickets < N do
  For  $n = 1 : j$  branches
    send a random  $k$  probes  $[0, N]$ 
    For each probe ticket
      initialize a counter for each QoS requirement
      if intermediate sensor then
        For each probe ticket
          update the traversed path
          update counter for each QoS requirement
          if QoS requirement Not satisfied then
            | drop probe
          else
            | Send to the next node towards the destination
          end
        end
      else
      end
    end
  if Destination node then
    | Send a unicast message to source node
  else
    | It is a source node
    | Compile all data
  end
end
end

```

3.3.2 Genetic algorithm

At each generation, the following three operations of the algorithm are performed iteratively:

- Selection: chromosomes from the current population are selected according to their level of fitness value while others are discarded.
- Crossover: in this operation, a new set of chromosomes are generated from two chromosomes in the population

randomly selected gene and then it is added to the current population. Then, the fitness function is computed again for these chromosomes.

Algorithm 2 depicts the pseudo code for this stage. More details are provided in the simulation setup section.

Algorithm 2 Genetic Ticket-Based Routing Algorithm

```

Result: Set of best chromosomes
Initialization: Initial Population
while fitness value is satisfied do
  Selection process: a set of "good" chromosomes is picked
  Crossover process: pick a pair of two chromosomes
  if random selection then
    swap genes around a common gene
    generate a new offsprings
    check validity
  else
    swap genes around a common gene
    generate a new offsprings
  end
  Mutation process: pick randomly a highly connected node to be a mutation point
  if part of a current chromosome then
    choose another node
  else
    generate a new offsprings
  end
end

```

The best resultant chromosomes (paths) are returned by the genetic algorithm as a final outcome, when the optimization condition is achieved. Crossover and mutation operators produce a new set of approximations and breed them together exactly like what happens on genetic process in nature. Furthermore, the fitness function is determined based on the factors that concern our problem. In this work, the cost and delay as QoS requirements specify the fitness function of the genetic algorithm.

In our example shown in Fig. 2, the source sensor received two route replies {0, 1, 3, 5, 7} and {0, 1, 4, 3, 6, 7}. In this case, these two routes can be used as initial population of genetic algorithm integrated in source sensor to find the optimal route that satisfies QoS requirements. Each path represents a chromosome; each sensor in the chromosome represents a gene. Hence, the number of genes in the chromosome determine its size. For example, path {0, 1, 3, 5, 7} is a chromosome consist of 5 genes.

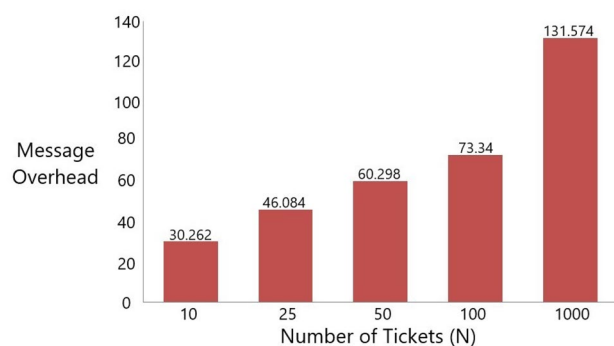
Table 1 Simulation parameters

Parameter	Value
Simulation period	1000 s
Topology dimension	150 m × 150 m
Number of nodes	100
MAC type	802.15.4
Protocol	TBR
Coverage range	35 m

4 Simulation setup

4.1 TBR simulation setup

We use Network Simulator-2 (NS-2) (McCanne et al. 1997) as our platform to simulate TBR as described in (ELMO

**Fig. 3** Number of probing messages per connection request (messages overhead) versus number of tickets (N)**Table 2** Number of probing messages per connection (message overhead)

Number of tickets	Number of probing messages	Number of probing messages per connection
10	15131	30.262
25	23042	46.084
50	30149	60.298
100	36670	73.34
1000	65787	131.574



Fig. 4 The simulation methodology; generating initial population using NS-2 and then it is fed to Matlab to execute the genetic algorithm and produce the best possible paths

	0	1	2	3	4	5	6	7
Node 0	0	1	0	0	0	0	0	0
Node 1	1	0	1	1	1	0	0	0
Node 2	0	1	0	0	0	1	0	0
Node 3	0	1	0	0	1	1	1	0
Node 4	0	1	0	1	0	0	1	0
Node 5	0	0	1	1	0	0	0	1
Node 6	0	0	0	1	1	0	0	1
Node 7	0	0	0	0	0	1	1	0

Fig. 5 Connectivity matrix of 8 nodes network example shown in Fig. 2; '0' entry means no connectivity, '1' means entry means connectivity exists

2010). In the following, we describe some features of our implementation of TBR in NS-2:

- When the source needs to start probing or an intermediate sensor has to forward a certain number of tickets (N) to its neighbors' sensors, the number of tickets that will be sent is determined randomly between $[0, N]$ using uniform distribution routine.
- When a probe is received by an intermediate sensor, the following actions are triggered. First, the accumulative delay and cost are calculated. Second, the probe information is stored a cache and it will be used later as a historical probing record. Finally, current sensor ID is added to TBR header (i.e. traversed path) of the newly generated probes delivered to its neighbors.
- When a probe is received by the destination sensor, a unicast route reply is sent back to the source using the same path after checking the QoS constraint as mentioned before.
- Intermediate sensors will drop a probe in the following cases:

- First, when a probe returns back to the sensor that originates this probe.
- Second, when a probe passed again the same sensor.
- Third, when another probe from the same source with the same request ID (sequence number) has already passed this sensor.
- Finally, if the accumulative delay exceeds the constraint delay or any other QoS constraints.

The simulations are accomplished using NS-2 version 2.34 installed in Ubuntu environment. The network model used in our simulation is randomly created. In $150m \times 150m$ area, one hundred sensor nodes are placed with zero mobility. The transmission range is set to $35m$ for each sensor node. The delay of each link is modeled as a uniform random variable in the range of $[0, 1]$ time unit. Table 1 summarizes the simulation parameters. This scenario is very close to the expected scenario in Smart Grid ad hoc networks environment where short-range wireless smart meters are deployed in a home-area-network.

We run the simulation 5 times for each number of tickets (i.e. $N = 10, 25, 50, 100$, and 1000). In each simulation run, the source sensor node is fixed and it transmits routing requests to 99 randomly selected destinations. Thus, we will have for each case (different ticket number experiment) 495 samples, which are statistically sufficient to achieve 95% confidence in our collected initial population.

The average path cost and success ratio of TBR have been studied in (Xiao et al. 2002). TBR has a lower ratio of all routes cost to the number of routes (average path cost) than flooding and shortest path algorithms. Moreover, the ratio of paths established to the total of route requests (success ratio) was close to the flooding algorithm (Chen and Nahrstedt 1998).

In order to demonstrate that our proposed approach has better searching capability in addition to lower network load, we define message overhead metric for the routing discovery method. Message overhead metric is defined as the average number of probing messages required per connection request. Any probe passes a link from one sensor node to any neighboring sensor node is considered as one probe message. Therefore, when a probe passes k hops, it is recorded as k messages. Figure 3 depicts the message overhead for TBR with different number of tickets. Clearly, we can observe the larger the number of tickets, the larger is message overhead which also shown in Table 2.

The values of $N = 10, 25$ and 50 cases are promising, but we should not miss the success ratio of these cases. However, for $N = 25$ and 50 cases, it can save 27.3% and 17.8% of message overhead, respectively and their success ratios with respect to $N = 100$ case are about 71% and 91%, respectively.

4.2 GA-TBR simulation setup

In this work, MATLAB is used along with NS-2 as our simulation platforms to test the impact of genetic algorithm so that the performance of the algorithm can be analyzed deeply at each stage. The initial population of route replies is exported from NS-2 as csv files. Figure 4 illustrates the simulation steps used in this work.

Moreover, the connectivity, cost and delay information of network model is exported from NS-2 in a matrix representation. Figure 5 shows the corresponding connectivity matrix of the network described in Fig. 2. Where '0' value indicates that there is no connection between these two sensor nodes. In other words, each sensor node that does not exist within the coverage range of the other sensor node has weight of '0'. However, '1' value represents a connectivity between sensor nodes. In a similar manner, other matrices (i.e. delay, cost) are represented with appropriate weights between sensor nodes.

We have implemented five features in our genetic algorithm: fitness function, validity checking, crossover and

mutation discoveries in addition to a defined source and destination for each particular run of the algorithm.

- Fitness Function: The fitness function depends on QoS routing metrics and the feasible route population is selected accordingly. It will be sorted in a decreasing or non-decreasing order based on our requirement. Combination of more than one QoS metric can be achieved by assigning different weights for each metric.
- Validity Checking: We will exploit the information learned and cached in each sensor node to check the validity of the new routes instead of sending unicast messages which will cause a huge background traffic. This step will minimize the message overhead, alleviates the network from extra load and reduces the network congestion.
- Crossover: Two chromosomes will be selected for crossover; if there is a common sensor node, it will be the crossover point otherwise, it will be chosen randomly. A common sensor node will ensure validity, which will save algorithm running time. For example, {0, 1, 3, 5, 7} and {0, 1, 4, 3, 6, 7} are routes discovered by TBR and GA uses them as an initial population. Let's assume that sensor node 3 is randomly selected as a crossing point. A new path of {0, 1, 3, 6, 7} results from the crossover operator and no validity check is required because the crossover is applied to a common gene. This new resulted path can be an optimal route from source to destination.
- Mutation: The algorithm will identify sensor nodes with high connections and one of them will be chosen randomly to be the mutation point. In order to enhance the validity, the new sensor node should not be part of the current chromosome.
- Source and Destination: Source and destination information is added to check the validity. There is a possibility that we might get a chromosome that does not have the source sensor node, destination sensor node nor both. In this case, we assign its fitness function to infinity to avoid selecting them as candidate solutions, when in reality they are not.

```

8 nodes:
initial population:
1,2,7,8          cost=3
1,2,4,5,8       cost=4
final result:
1,2,7,8          cost=3
1,2,4,5,8       cost=4
1,2,4,8         cost=Inf  "not valid route"
1,2,3,5,8       cost=Inf  "not valid route"
=====
16 nodes:
initial population:
1,2,6,8,16       cost=4
1,2,3,7,14,15,16 cost=6
final result:
1,2,6,8,16       cost=4
1,2,3,7,14,15,16 cost=6
1,2,13,8,16     cost=Inf  "not valid route"
1,2,6,7,14,15,16 cost=Inf  "not valid route"
=====
    
```

Fig. 6 Offspring evaluation after running the enhanced algorithm for fitness and validity

Table 3 Statistical analysis for three crossover techniques

	Random selection with cut vertex cross over	Sequential selection with cut vertex cross over	Sequential selection w/o cut vertex cross over
Runs	100	100	100
Mean	0.909	0.747	0.770
STDEV	0.0933	0.0815	0.0838
Level of significance	0.05	0.05	0.05
Confidence level (95%)	0.182	0.159	0.163
Lower bound	0.727	0.588	0.607
Upper bound	1.091	0.905	0.934

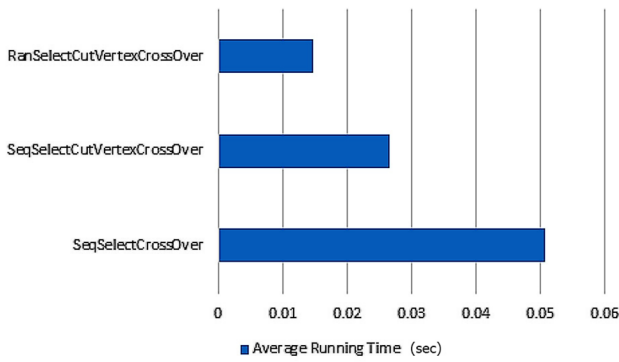


Fig. 7 Comparison of average running time for different crossover techniques with different selection methods performed on whole population

It is well known that the number of rows in the population matrix represents the number of candidate solutions (routes). On the other hand, the relationship between transmission range and the population matrix is the number of columns in the population matrix. In other words, the maximum transmission range determines the chromosome length. Furthermore, using the whole range is impractical for this type of optimization, so the route can be terminated after reaching the destination sensor node using a special termination character (-1 in our case) to fill the remaining cells. This is useful in managing the maximum transmission range by restricting the number of columns in the population matrix.

One of the main advantages of genetic algorithms is randomness; this implies that having a random initial population will be an advantage. However, this is not the case in path optimization problems as we are looking for a valid path and not any path. TBR can provide valid paths so the genetic algorithm can manipulate them and enhance the validity. Even if a random initial population has a large size, it does not have the same impact on the algorithm as a small

Fig. 8 100 runs show the behavior of three crossover techniques with respect to time

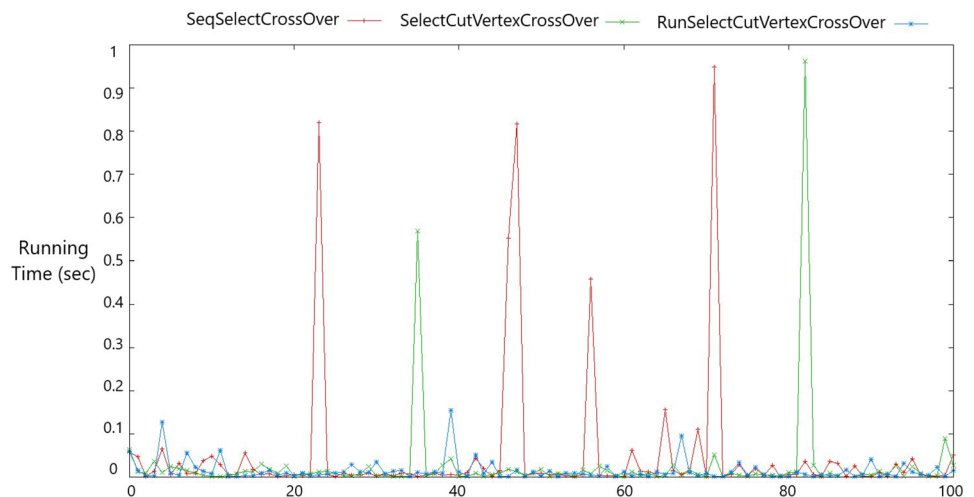


Table 4 Statistical analysis for mutation performed on different population percentages using random selection crossover with cut vertices

	10%	20%	80%	90%	100%
Count	100	100	100	100	100
Mean	0.0808	0.0460	0.0268	0.0216	0.0157
STDEV	0.0116	0.0082	0.0087	0.0081	0.0034
Confidence Coefficient	0.95	0.95	0.95	0.95	0.95
Level of Significance	0.05	0.05	0.05	0.05	0.05
Confidence Level (95%)	0.0227	0.0161	0.0170	0.0158	0.0067
Lower bound	0.0581	0.0298	0.0098	0.0058	0.0090
Upper bound	0.1035	0.0621	0.0437	0.0374	0.0225

initial population with valid routes. Moreover, the algorithm needs a space (Population size) to add off springs; in this case, the source sensor node is allowed to duplicate the initial population so it has valid routes with double size and the random search will take care of duplicates. Therefore, a valid route most likely will lead to another valid route. In contrary, an invalid route most likely will lead to another invalid route or it will take much time to be validated by the algorithm. Figure 6 presents our GA implementation by applying a cost of infinity to sensor nodes where a connection does not exist. To illustrate, when the algorithm is run, there is no feasible path between sensor nodes 1, 2, 4, and 8.

5 Performance analysis

Having introduced above the main features of our proposed approach, we shall first analyze the genetic algorithm alone and then we evaluate the performance of TBR along with the genetic algorithm.

Table 5 Run time (in seconds) results of cost and delay as QoS metrics with various numbers of tickets (N) of TBR

Number of tickets (N)	Cost			Delay			Overall		
	MEAN	MAX	MIN	MEAN	MAX	MIN	MEAN	MAX	MIN
10	0.035025	0.09311	0.000759	0.025254	0.090229	0.000622	0.03014	0.09311	0.000622
25	0.033249	0.09434	0.000762	0.02052	0.070107	0.000548	0.026885	0.09434	0.000548
50	0.03069	0.087502	0.000597	0.019295	0.099336	0.000421	0.024993	0.099336	0.000421
100	0.029454	0.092047	0.000617	0.018653	0.0996	0.000435	0.024053	0.0996	0.000435
1000	0.027749	0.09891	0.000459	0.015425	0.099041	0.000419	0.021587	0.099041	0.000419
Overall	0.029563	0.09891	0.000459	0.017785	0.0996	0.000419	0.023674	0.0996	0.000419

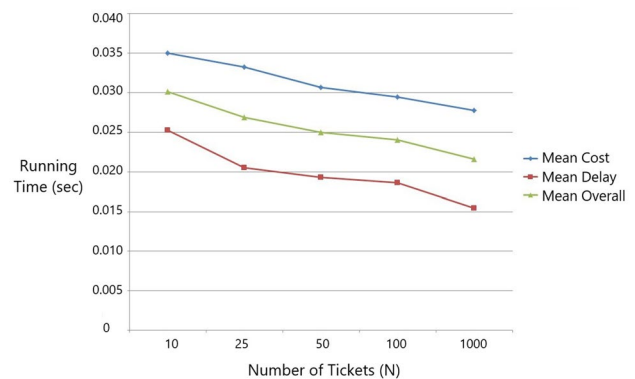
Table 6 Run time (in seconds) results comparison with various number of paths in initial population of genetic algorithm

Initial Population	Cost			Delay			Overall		
	MEAN	MAX	MIN	MEAN	MAX	MIN	MAX	MIN	
2 or 3	0.028696	0.09891	0.000459	0.016201	0.099041	0.000419	0.022448	0.099041	0.000419
4 or more	0.02164	0.080992	0.000751	0.01476	0.084966	0.000564	0.0182	0.084966	0.000564

5.1 Genetic algorithm analysis

It is important to perform the crossover and mutation probability analysis and compare the performance among different possible approaches for route optimization. As mentioned before, the crossover is one of the main components of genetic algorithms where pairs of chromosomes will be selected and a crossing point will be set by the algorithm. Then, the pairs of offspring will be produced. Selecting the mid of the chromosomes as crossover points will most likely produce invalid routes and it increases the overhead on the algorithm due to invalid routes. Hence, it is highly necessary to choose the crossover point carefully. Looking for cut vertices in the selected chromosomes will enhance the validity of the produced offspring, which can save the running time. The crossover phase will look for the common sensor node (cut vertex), otherwise, it will be the mid of the chromosome.

The selection phase has been implemented by two methods namely, sequential and random selection. In the sequential method, the pairs will be selected as they appear in the population. On the other hand, the random selection will choose a chromosome randomly and cross it over with the following chromosome. Furthermore, in order to investigate the impact of cut-vertex sensor nodes, we have tested the following methods: *Sequential Selection Crossover without Cut Vertices*, *Sequential Selection Crossover with Cut Vertices* and *Random Selection Crossover with Cut Vertices*. Table 3 summarizes the comparison between the crossovers using cut vertices and using mid of the chromosome. Figure 7 shows the average running time of different crossover techniques with different selection methods performed on the whole population. The random selection with cross-over

**Fig. 9** Run time results (y-axis) of cost, delay and overall (both cost and delay) as QoS metrics with various numbers of tickets (x-axis) of TBR

shows the shortest running time. The stability of the algorithm with respect to time is a critical factor for genetic algorithms to hit the optimal solution; Fig. 8 shows the behavior of three crossover techniques with respect to time with 100 runs. Again, we can observe that the random selection with crossover has the least fluctuations, while the sequential selection without crossover has shown the highest incident of fluctuations. These fluctuations affect the stability of the genetic algorithm. Therefore, we pick the random selection with cross over technique for testing the mutation process.

The algorithm will identify the connected sensor nodes and one of them will be chosen randomly to be the mutation point. The mutation point connections will be imported from the cost matrix; the connections will be shuffled and

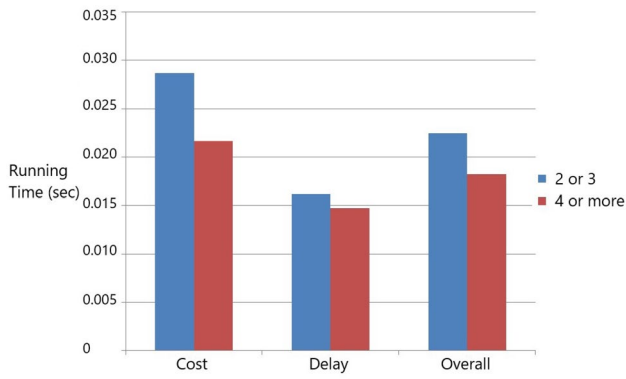


Fig. 10 Run time results (y-axis) comparison with various number of paths in initial population of genetic algorithm

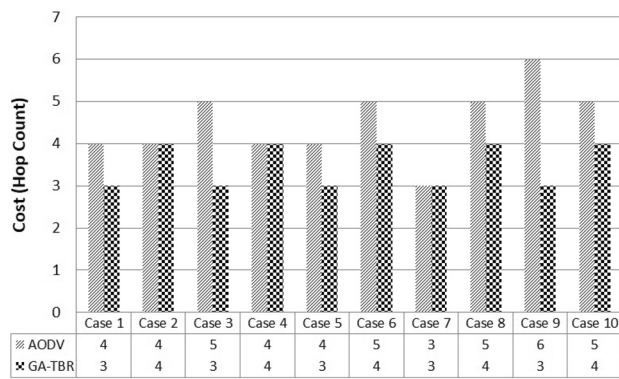


Fig. 11 Cost metric comparison between AODV and GA-TBR protocol using a random topology of 100 nodes with fixed source and 10 random destinations

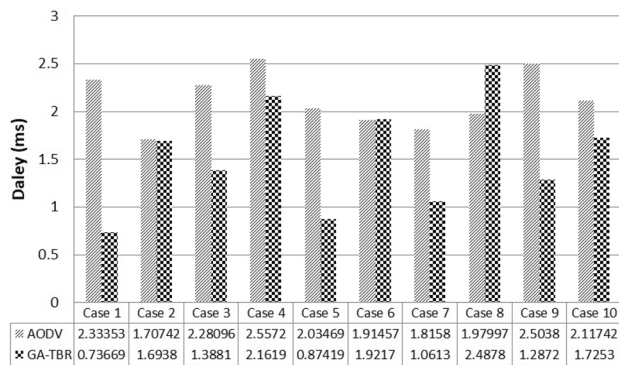


Fig. 12 Delay metric comparison between AODV and GA-TBR protocol using a random topology of 100 nodes with fixed source and 10 random destinations

one of them is randomly chosen. To enhance the validity, the new sensor node should not be part of the current chromosome and the algorithm should not choose the destination sensor node as a mutation point. In Table 4, we illustrate

the comparison between different percentages of mutation candidate chromosomes; we have tested them with Random Selection Crossover with Cut Vertices.

As explained above, source and destination are added to check the validity. There is a possibility that the algorithm produces a chromosome that does not contain the source sensor node, destination sensor node or both. Moreover, checking whether the source and destination sensor nodes within the chromosome can be very helpful in case of parallelism. Assume an intermediate sensor node “A” tries to figure out the optimal route to sensor node “D”; in order to apply parallelism, sensor node “A” will call its neighbor sensor nodes “B” and “C” to search for a route from “A” to “D”; in this case, sensor node “A” has to pass current source and destination information as parameters to sensor nodes “B” and “C”. Finally, “A” will collect the results (i.e. from B, C) along with its own result and evaluate them to choose the best route.

5.2 Genetic algorithm with TBR analysis

Having established and studied the performance of the proposed genetic algorithm (GA), we are ready to study the integration of TBR with GA. As mentioned before and shown in Fig. 4; In order to evaluate the genetic algorithm overhead, we have used NS-2 to simulate TBR that produces matrices of connectivity, cost and delay as well as discovered paths for each connection request considering different number of tickets. These matrices of network model information and discovered paths are saved as csv files. In MATLAB, we use these csv files as inputs for running the genetic algorithm code.

In Tables 5 and 6, we present minimum, maximum and average execution time of genetic algorithm in Matlab for paths and topology generated in NS-2. These results show that the additional processing overhead of genetic algorithm used in our proposed method to discover the best feasible route is very minimal. Cost and delay are used as QoS metric individually to obtain feasible routes. As shown in Fig. 9, the computation time increases with the decrease of number of tickets where the mean execution time of $N = 1000$ is about (0.0296s - Cost, 0.0178s - Delay) and that for $N = 25$ is about (0.03325s - Cost, 0.0205s - Delay). Delay metric has lower minimum and average execution time due to delay aware behavior of TBR routing to find low cost/delay paths. Furthermore, increasing number of tickets to 1000 surely will decrease the overall delay, yet, it is not effective because of the huge overhead traffic that will overwhelm the whole network and cause congestion.

Table 6 presents the execution time which is needed based on initial population size. The mean time required to determine the feasible path from 2 or 3 discovered paths is about (0.0287s—cost, 0.0162s—delay). While, lower computation

time is needed for larger initial population where the mean time is about (0.0216s—cost, 0.0148s—delay) as shown in Fig. 10.

5.3 Comparison of genetic algorithm with AODV

Using the same topology composed of 100 nodes, we have tested the performance of AODV using ns-2 simulator. Ten cases have been examined where we have fixed the source node and randomly chosen ten destination nodes. Figures 11 and 12 depict the number of hop count and total delay for each case, respectively. We can observe that GA-TBR outperforms AODV in all cases except case#8; it proves the notion that have been discussed above that GA does not always provide the best answer. However, in all other cases GA-TBR shows outstanding performance with 28% improvement on average. In addition, in some cases such as case#1 GA-TBR shows 68% improvement.

6 Conclusion

Smart Grid communications network is composed of a numerous number of sensors and actuators. This network will carry diversified types of traffic, each has its own QoS requirements/constraints. Routing selection plays a key role in satisfying these constraints. In this work, we proposed a meta-heuristic technique for routing discovery of QoS aware TBR protocol in a WSN-Smart Grid environment. Genetic algorithm operations are used to explore new feasible routes without sending an extra number of tickets. We validated mutated route using information stored in sensor nodes caches. Therefore, there are no extra routing messages overhead required for validating new routes (off springs) due to unicast traffic; this is a huge advantage of our approach compared to existing works. On the other hand, the results of genetic algorithm running time demonstrated that our proposed method has minimum execution overhead. In addition, GA-TBR shows 28% average improvement compared to AODV and in some cases it shows 68% improvement.

The proposed enhancement on TBR protocol was tested on a network model that has fixed sensor nodes. It is an interesting and challenging problem to test our approach on mobile and dynamic environment. In addition, in this work, only the source sensor nodes applied the genetic algorithm, what if intermediate sensor nodes or even the destination sensor nodes would participate in this process? On the other hand, one of the limitations of this work is that we did not evaluate resource consumption of our proposed approach such as power and memory. So further evaluation is needed to assess our approach due to the usage of heuristic algorithm which requires more processing and memory in the participate node.

Furthermore, although social network could help in promoting the vision of smart grid, incorporating this type of open networks would lead to misinformation attack problem that could cause great damage to the smart grid (Pan et al. 2017). As a future direction, we will investigate the performance of GA-TBR under such attack.

References

- Ahn CW, Ramakrishna RS (2002) A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Trans Evolut Comput* 6(6):566–579
- Ancillotti E, Bruno R, Conti M (2013) The role of communication systems in smart grids: architectures, technical solutions and research challenges. *Comput Commun* 36(17):1665–1697
- Chen S, Nahrstedt K (1998) Distributed qos routing with imprecise state information. In: *Computer Communications and Networks, 1998. Proceedings. 7th International Conference on IEEE*, pp 614–621
- Cheng H, Yang S (2010a) Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks. *Eng Appl Artif Intell* 23(5):806–819
- Cheng H, Yang S (2010b) Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile ad hoc networks. In: *Applications of Evolutionary Computation*, Springer, pp 562–571
- Chhaya L, Sharma P, Bhagwatikar G, Kumar A (2017) Wireless sensor network based smart grid communications: cyber attacks, intrusion detection system and topology control. *Electronics* 6(1):5
- Dengiz B, Altıparmak F, Smith AE (1997) Local search genetic algorithm for optimal design of reliable networks. *IEEE Trans Evolut Comput* 1(3):179–188
- ELMO (2010) Ns2: How to add new routing protocol—causal nexus. <http://elmurod.net/en/index.php/archives/157>
- Fadel E, Gungor VC, Nassef L, Akkari N, Malik MA, Almasri S, Akyildiz IF (2015) A survey on wireless sensor networks for smart grid. *Comput Commun* 71:22–33
- Fateh B, Govindarasu M, Ajarapu V (2013) Wireless network design for transmission line monitoring in smart grid. *IEEE Trans Smart Grid* 4(2):1076–1086
- Gentile U, Marrone S, Mazzocca N, Nardone R (2016a) Cost-energy modelling and profiling of smart domestic grids. *Int J Grid Util Comput* 7(4):257–271
- Gentile U, Marrone S, Mazzocca N, Nardone R (2016b) *μgrimoire*: A tool for smart micro grids modelling and energy profiling. *The Open Cybernetics & Systemics Journal* 10(1)
- Goldberg DE (1989) *Genetic algorithms in search, optimization, and machine learning*, 1989. Reading: Addison-Wesley
- Gungor VC, Sahin D, Kocak T, Ergut S, Buccella C, Cecati C, Hancke GP (2011) Smart grid technologies: communication technologies and standards. *IEEE Trans Ind Inf* 7(4):529–539
- Gungor VC, Sahin D, Kocak T, Ergut S, Buccella C, Cecati C, Hancke GP (2013) A survey on smart grid potential applications and communication requirements. *IEEE Trans Ind Inf* 9(1):28–42
- He D, Chan S, Guizani M (2017) Cyber security analysis and protection of wireless sensor networks for smart grid monitoring. *IEEE Wireless Communications*
- Huang Q, Li X, Zhao J, Wu D, Li XY (2015) Social networking reduces peak power consumption in smart grid. *IEEE Trans Smart Grid* 6(3):1403–1413
- Jiang H, Zheng L, Liu Y, Zhang M (2010) Multi-constrained qos routing optimization of wireless mesh network based on hybrid

- genetic algorithm. In: *Intelligent Computing and Integrated Systems (ICISS)*, 2010 International Conference on, IEEE, pp 862–865
- Kamilaris A, Taliadoros G, Pitsillides A, Papadiomidous D (2012) The practice of online social networking of the physical world. *Int J Space-Based Situat Comput* 2(4):240–252
- Kovendan A, Sridharan D (2016) Wireless sensor networks based control strategies for the enhancement of reliability in smart grids. *Circuits Syst* 7(09):2499
- Krasnogor N, Smith J (2005) A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *IEEE Trans Evolut Comput* 9(5):474–488
- Kurt S, Yildiz HU, Yigit M, Tavli B, Gungor VC (2017) Packet size optimization in wireless sensor networks for smart grid applications. *IEEE Trans Ind Electron* 64(3):2392–2401
- Levis P, Tavakoli A, Dawson-Haggerty S (2009) Overview of existing routing protocols for low power and lossy networks. IETF Draft Report
- Li H, Zhang W (2010) Qos routing in smart grid. In: *Global Telecommunications Conference (GLOBECOM 2010)*, 2010 IEEE, pp 1–6
- Li M, Chi XB, Jia XC, Zhang JL (2016) Wsn-based efficient monitoring for overhead transmission line in smart grid. In: *Control Conference (CCC)*, 2016 35th Chinese, IEEE, pp 8485–8489
- Liu W, Gong M, Wang S, Ma L (2018) A two-level learning strategy based memetic algorithm for enhancing community robustness of networks. *Inf Sci* 422:290–304
- Lu T, Zhu J (2013) Genetic algorithm for energy-efficient qos multicast routing. *IEEE Commun Lett* 17(1):31–34
- Mazur K, Wydra M, Ksiezopolski B (2017) Secure and time-aware communication of wireless sensors monitoring overhead transmission lines. *Sensors* 17(7):1610
- McCanne S, Floyd S, Fall K, Varadhan K, et al (1997) Network simulator ns-2
- Moreno-Munoz A, Bellido-Outeirino F, Siano P, Gomez-Nieto M (2016) Mobile social media for smart grids customer engagement: Emerging trends and challenges. *Renewable and Sustainable Energy Reviews* 53:1611–1616, <https://doi.org/10.1016/j.rser.2015.09.077>, <http://www.sciencedirect.com/science/article/pii/S1364032115010473>
- Pan T, Mishra S, Nguyen LN, Lee G, Kang J, Seo J, Thai MT (2017) Threat from being social: Vulnerability analysis of social network coupled smart grid. *IEEE Access* 5:16,774–16,783
- Rekik S, Baccour N, Jmaiel M, Drira K (2017) Wireless sensor network based smart grid communications: Challenges, protocol optimizations, and validation platforms. *Wireless Personal Communications* pp 1–23
- Saputro N, Akkaya K, Uludag S (2012) A survey of routing protocols for smart grid communications. *Comput Netw* 56(11):2742–2771
- Sara GS, Sridharan D (2014) Routing in mobile wireless sensor network: a survey. *Telecommun Syst* 57(1):51–79
- Tuballa ML, Abundo ML (2016) A review of the development of smart grid technologies. *Renew Sustain Energy Rev* 59:710–725
- Usman A, Shami SH (2013) Evolution of communication technologies for smart grid applications. *Renew Sustain Energy Rev* 19:191–199
- Vallejo A, Zaballos A, Selga JM, Dalmau J (2012) Next-generation qos control architectures for distribution smart grid communication networks. *Commun Magazine, IEEE* 50(5):128–134
- Wang P, Hou H, He X, Wang C, Xu T, Li Y (2015) Survey on application of wireless sensor network in smart grid. *Procedia Comput Sci* 52:1212–1217
- Xiao L, Wang J, Nahrstedt K (2002) The enhanced ticket-based routing algorithm. In: *Communications, 2002. ICC 2002. IEEE International Conference on, IEEE*, vol 4, pp 2222–2226
- Yan X, Zhang L, Wu Y, Luo Y, Zhang X (2017) Secure smart grid communications and information integration based on digital watermarking in wireless sensor networks. *Enterprise Inf Syst* 11(2):223–249
- Yang J, Cheng R, Liu W, Xiao Y, Zhang F (2013) Cryptanalysis and improvement of smart prepayment meter protocol in standard q/gdw 365. *Int J Grid Utility Comput* 4(1):40–46
- Yang S, Cheng H, Wang F (2010) Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks. *IEEE Trans Syst, Man, Cybern, Part C (Appl Rev)* 40(1):52–63
- Yen YS, Chao HC, Chang RS, Vasilakos A (2011) Flooding-limited and multi-constrained qos multicast routing based on the genetic algorithm for manets. *Math Comput Model* 53(11):2238–2250
- Yoldas Y, Onen A, Muyeen S, Vasilakos AV, Alan I (2017) Enhancing smart grid with microgrids: challenges and opportunities. *Renew Sustain Energy Rev* 72:205–214
- Zaballos A, Vernet D, Selga JM (2013) A genetic qos-aware routing protocol for the smart electricity networks. *International Journal of Distributed Sensor Networks* 2013
- Zafar S, Soni M (2014) Trust based qos protocol (tbqp) using meta-heuristic genetic algorithm for optimizing and securing manet. In: *Optimization, reliability, and information technology (ICROIT)*, 2014 International Conference on, IEEE, pp 173–177