

TOWARDS AGENT-BASED MODEL SPECIFICATION OF SMART GRID: A COGNITIVE AGENT-BASED COMPUTING APPROACH

Waseem Akram¹, Muaz A. Niazi^{1, *}, Laszlo Barna Iantovics²
and Athanasios V. Vassilakos³

¹COMSATS University Islamabad, Computer Science Department
Islamabad, Pakistan

²Petru Maior University of Tirgu Mures, Informatics Department
Tirgu Mures, Romania

³Lulea University of Technology, Computer Science Department
Lulea, Sweden

DOI: 10.7906/indecs.17.3.14
Regular article

Received: 11 June 2018.
Accepted: 25 November 2018.

ABSTRACT

A smart grid can be considered as a complex network where each node represents a generation unit or a consumer, whereas links can be used to represent transmission lines. One way to study complex systems is by using the agent-based modeling paradigm. The agent-based modeling is a way of representing a complex system of autonomous agents interacting with each other. Previously, a number of studies have been presented in the smart grid domain making use of the agent-based modeling paradigm. However, to the best of our knowledge, none of these studies have focused on the specification aspect of the model. The model specification is important not only for understanding but also for replication of the model. To fill this gap, this study focuses on specification methods for smart grid modeling. We adopt two specification methods named as Overview, design concept, and details and Descriptive agent-based modeling. By using specification methods, we provide tutorials and guidelines for model developing of smart grid starting from conceptual modeling to validated agent-based model through simulation. The specification study is exemplified through a case study from the smart grid domain. In the case study, we consider a large set of network, in which different consumers and power generation units are connected with each other through different configuration. In such a network, communication takes place between consumers and generating units for energy transmission and data routing. We demonstrate how to effectively model a complex system such as a smart grid using specification methods. We analyze these two specification approaches qualitatively as well as quantitatively. Extensive experiments demonstrate that Descriptive agent-based modeling is a more useful approach as compared with Overview, design concept, and details method for modeling as well as for replication of models for the smart grid.

KEY WORDS

agent-based modeling, cognitive agent-based computing, complex networks, smart grid

CLASSIFICATION

JEL: C65

*Corresponding author, η : muaz.niazi@gmail.com; -; -;

INTRODUCTION

A smart grid focuses on the complex interactions between utility service providers and consumers. It involves the non-linear dialogue of power and information data between utility service providers and consumers [1]. The complex interaction in the form of repeated auction, fluctuating supply and demand add complexity to the nature of a smart grid. Because of this complex nature, a smart grid can be considered a complex system.

The study and understanding of any complex system are associated with the modeling of the system. Modeling complex system allows better understanding and analyzing the emergent behavior of each entity involved in the system [2]. Being a complex system, a smart can also essentially be modeled in the form of either agent-based or complex network-based models [3, 4]. These models can well represent the smart grid in term of its various components, their behavior, and communication among them for energy distribution and management.

A particular way of modeling a smart grid as a complex network is by including its various components such as generating units, consumers, distributors, and other components as nodes and communication lines as edges. Chassin et al. in [5] developed the complex network model for the US power grid by considering nodes as power sources and consumers, while edges as communication lines. After developing different complex networks, we are able to use the mathematical tool for computing centrality measures and some metrics on such networks. These measurements allow studying the global behavior of each component in a large-scale power system network.

In scientific literature, agent-based modeling (ABM) and the multi-agent systems (MAS) are successfully used in the smart grid domain. Some of these works have been discussed in the later section of the article (see discussion section). However, these works lack in any ABM specification approach for documenting ABM. An ABM specification is most important for understanding as well as replication ABM. The lack of specification methods causes issues such as low understandability of the model, impossibility to replicate and extend the model, and impossibility to integrate with the existing system. So there is a need for an easily understandable methodology to describe and document an ABM, specifically in the smart grid domain.

This study is motivated by the lack of specification studies in the domain of smart grid. To this end, this article presents a first step towards the use of specification methodology for the ABM development, in particular for the smart grid system. We adopt two approaches. The first method is ODD (short for Overview, Design concept, and Details) [6] and the second is DREAM (short for Descriptive Agent-based modeling) [3]. The proposed study supports ABM developing by using specification methods starting from conceptual modeling to validated ABM through simulation. It also supports modeling complex system, more effective knowledge transfer, and communication between multidisciplinary researchers. To validate our work, we consider a case study from the smart grid domain. In the case study, we consider a large set of the network in which different consumers and power generation units are connected with each other through different configuration. In such a network, communication takes place between consumers and generating units for energy transmission and data routing. We demonstrate how to effectively model a complex system such as a smart grid using specification methods. Finally, we present a comparative analysis of both specification techniques.

Our main contributions can be listed as follows:

- 1) A proposed approach for modeling and simulation of the smart grid using the complex network and agent-based modeling approaches.

- 2) The ODD specification approach used for ABM model of smart grid.
- 3) The DREAM specification approach used for ABM model of smart grid.
- 4) A comparative analysis of ODD and DREAM specification techniques.

The rest of the article is structured as follows: Section 2 presents basic background and concepts, in Section 3 a model development is presented, Section 4 is dedicated for results and discussions, the article ends with conclusions formulated in Section 5.

BACKGROUND

In this section, we present the basic concept and understanding of cognitive agent-based computing approach, DREAM, and ODD specification approaches.

COGNITIVE AGENT-BASED COMPUTING APPROACH

Niazi and Hussain in [7] have presented a unified framework called cognitive agent-based computing framework. The framework is designed for facilitating the development, comparison, communication, and validation of models across different scientific domains. Here, the word Cognitive is used because the goal of the framework is to develop cognition or understanding of the different aspects of the model or system under study. The framework offers tutorials and guidelines in the form of four different modeling levels. This approach involves the process of taking any complex system from the real world and converting it into a suitable simple model by using specific modeling level such as exploratory or descriptive agent-based approach. The exploratory approach involves the use of agents to explore the complex systems, identify which agent-based model is feasible for the specific problem then develops the proof-of-concept and also explains what kind of data is required for validation and verification of the model. The descriptive agent-based modeling level is the process of presenting ABM in the form pseudocode, a complex network of the model, social network analysis. The framework combines other modeling levels named complex network modeling and validation/verification modeling.

DESCRIPTIVE AGENT-BASED MODELING

Descriptive agent-based modeling (DREAM) is a cognitive agent-based computing approach developed in [3]. DREAM offers an ABM specification technique which comprises of developing a complex network of the ABM, pseudocode specification models and social network analysis of the network model. It offers a detailed description of ABM as well as visual based analysis. It provides an easy translation of the network model into pseudocode followed by ABM development. In Figure 1, DREAM methodology is shown.

OVERVIEW, DESIGN CONCEPT AND DETAILS

Overview, Design concept and Details (ODD) is originally developed by [6] and an updated version is presented in [8]. It is a textually based specification technique for documenting ABM. It provides a checklist which covers key features of the model. It comprises of three main sections which are Overview, Design concept, and Details. These sections are further divided into subsections. Figure 2 presents the ODD specification methodology. A detailed description can be found in [8].

MODEL DEVELOPMENT

In this section, we present a case scenario of the smart grid, followed by ODD and DREAM specification of the model. Figure 3 shows our research methodology. First, we start by describing the case model from the smart grid domain. Then, we present a model specification

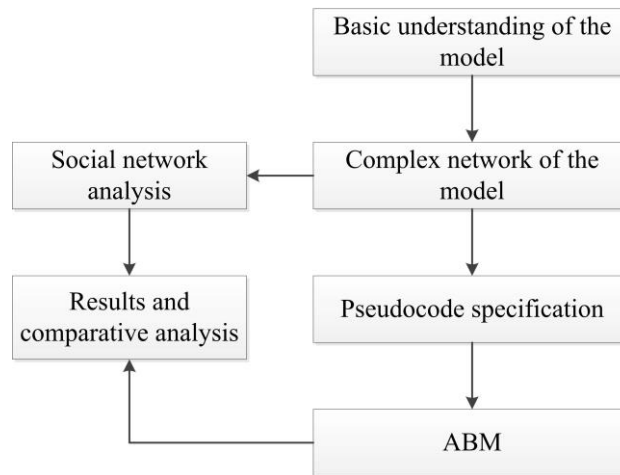


Figure 1. DREAM methodology for ABM specification adopted from [3]. It can be noticed that it starts from a basic understanding of the model, then followed by developing a complex network of the model. There are two steps after network formation, one is to present pseudocode specification, this step allows for the actual code translation, the other is applying social network analysis tool to compute centrality measures of the network. The final step involves the analysis of the results.

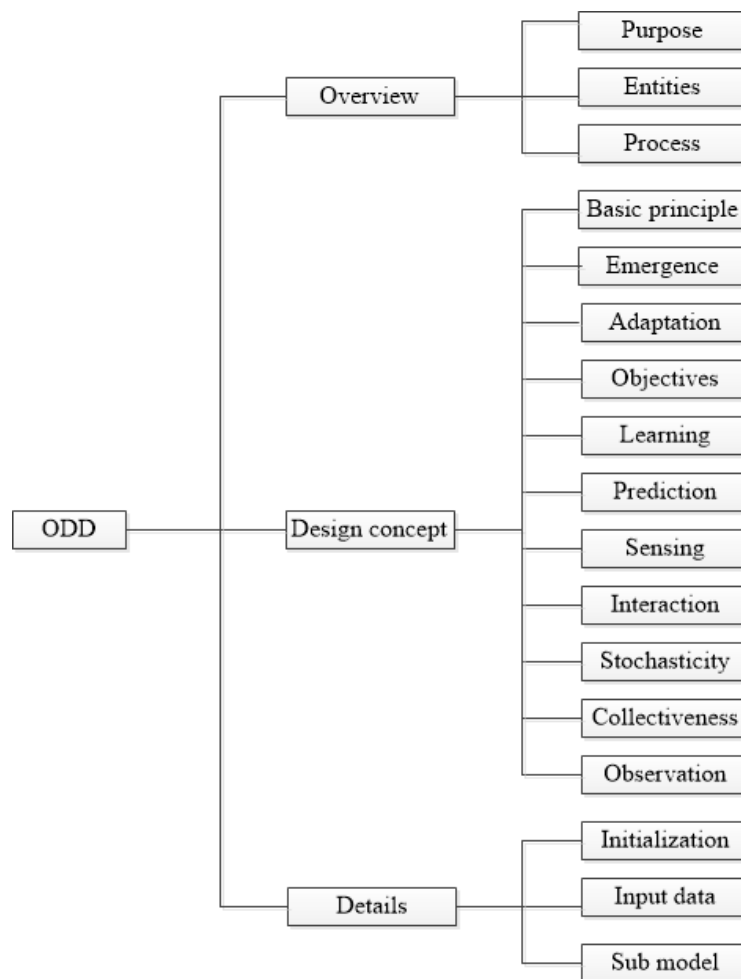


Figure 2. An ODD methodology for ABM specification adopted from [8]. It can be noticed that ODD is divided into three main sections: overview, design concept, and details. Then each section is further divided into subsections. These sections cover key features of the model.

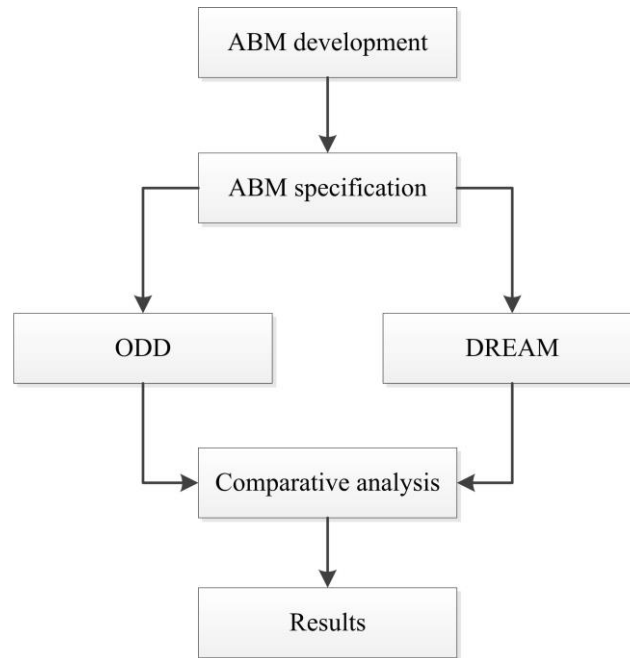


Figure 3. Our research method. It can be noticed that the first step comprises of ABM development followed by the specification of the ABM. Two specification methods are adopted (ODD and DREAM). After specification, the next step is to compare and analyze both specifications approaches qualitatively as well as quantitatively. The final step shows the results of the comparative analysis.

using ODD and DREAM approaches. Through specification methods, we provide guidelines for ABM development of the case scenario. The specification study is complemented by an ABM for the scenario. Finally, the comparative analysis of specification methods as well as simulation results is provided.

SCENARIO OF SMART GRID

To model a smart grid, let us consider a large set of networks, in which different consumers and power generation units are connected with each other through different configuration. To model possibilities of a different configuration of the large-scale power system, we use standard complex networks such as small-world [9], scale-free [10], and random network [11]. For validation, we apply routing technique such as random walk and centrality-based routing.

The routing process involves the selection of a path from the source toward the destination. Routing strategy in complex networks can be categorized into two types, i.e. Local and Global Routing. The local routing strategy needs local information about neighbor nodes. These include local static routing, local dynamic routing, and local pheromone routing [12]. On the other side, the global routing strategy needs global information like topological structure, characteristics of each node and real-time information. These include shortest path routing, efficient routing, and global dynamic routing [13].

For large-scale complex networks, global routing remains problematic. It is difficult to have the characteristics of each node and to have real-time information. Another difficulty consists in the increases in computational time. While on the other hand, local routing remains promising for large-scale real-world complex networks. It offers less computational time as well as easy implementation.

In a smart grid environment, two types of routing occur. The one is energy demand from the consumer's side to the generation unit, while the generation unit responses by providing

energy to consumers. The second is data and information routing about demand profile from consumers and energy cost from grid unit [14].

MODEL SPECIFICATION ACCORDING TO ODD

In this section, we present the model specification by following the ODD model. In table 5, we summarize the ODD specification of the understudy model.

The overview section of ODD

- 1) The purpose of the model: To understand how a combination of agent-based and complex network-based modeling approaches can be used to simulate large-scale power system. Further, how routing techniques can be used to validate the model.
- 2) The involved entities: The model consists of three types of agents named; consumers, producers, and walkers that are represented by nodes in a complex network. The model allows producers and consumers to be generated randomly in a network. Producers generate power and can transmit to the consumers through communication lines that we called links. Consumers demand and finally use energy power. The environment is set as a complex network where nodes represent producers and consumers and links represent transmission lines. State variables `visited?` and `consumer?` are used to mark once a node being visited and to check is any available consumer? – a node in the neighbor list. It is a convention (coding standard for NetLogo) to define a variable name ending with a question mark.
- 3) Routing purpose: For routing purpose, the concept of walkers is deployed. Initially, the walkers are located at the producer's nodes. They search for the neighbor nodes and move to one of the neighbor's node. Once a node is visited, it is marked as `visited?` The walkers also check for the consumer's node. The simulation time is kept as continues. By continues method, the NetLogo continuously updates the plots. At each time step, plots are generated in order to measure visited nodes and visiting consumers.

Design concept section of ODD

- 1) **Basic principles:** The basic hypothesis of our model is that a cognitive agent-based computing approach is better for modeling and simulation of the large-scale power system. In our approach, we used a combination of agent-based and complex network-based modeling approaches. We developed complex network models such as small-world, scale-free, and random network to simulate a smart grid based environment.
- 2) **Emergence:** The “emergence” feature shows information about “what kinds of outputs of the model are modeled?” In other words, we can say what the expected results from the model are? In the case of our approach, the routing techniques (random walk and centrality routing) are used for transmission from producers towards consumers. The key results are the computation of end to end delivery from producers towards consumers.
- 3) **Adaptation:** Adaptive feature of the model shows decision-making capability for the agents against the changing environment. Decisions are taken by using well-defined constraints to adapt the variation in the environment accordingly. There are two rules applied to make a decision. When using a random walk, the walkers search for neighbor's node and select one of them, while by using centrality routing, the walkers search for a neighbor node with maximum value and select that one.
- 4) **Objectives:** In a changing environment, individual agents also receive effects or rewards from the environment for their adaptive behavior to achieve one's objective. In our model, the main objective is to measure how much time is taken while moving from one node to another.

- 5) **Sensing:** In the decision-making process among agents, there are some specific features related to each agent which allow communicating neighbor to make their decision according to the value of those features. In our case scenario, the walkers use the sensing property, if a neighbor node is already been visited then they avoid rerouting. They also sense for consumers if any visited node is a consumer, then they deliver packets or energy.
- 6) **Interaction:** Producers and consumers can communicate with each other for power transmission.
- 7) **Stochasticity:** The routing process is modeled as random.
- 8) **Observation:** When the simulation is running, at each time step the following data are collected.
 - A number of nodes.
 - A number of producers.
 - A number of consumers.
 - A number of walkers.
 - A number of nodes visited.
 - A number of consumers visited.

The details section of ODD

The details section of the ODD specification covers features of the model about what is the initial state of the model, what kind of data is used, and what types of parameters and parameters values are set in the model.

- 1) **Initialization:** the model is implemented in NetLogo agent-based modeling tool. The model environment is initialized by calling “draw-network” method. This method is used to draw any selected network. Then consumers and producers are generated randomly by specifying their number. After this, the walkers are placed at the producer’s location.
- 2) **Input data:** the standard complex network are generated and kept as external source files. These network files are used as input for the model.
- 3) **Submodels:** the model parameters and parameters values are given in Table 1.

Table 1. Evaluation metrics: These parameters and parameter values are used for model simulation. The region shows the simulation environment which is kept as 100 by 100. The number of nodes in the network is considered as 500. The numbers of power sources and consumers used are 10, 50, 100, 150, 200, 250, 300. Three different standard complex networks are used. For routing purpose, random walk and centrality-based routing algorithm are applied. The Performance of the model is measured in term of average delivery rate (computation time from sources to the consumers). A series of experiments were carried out with the simulation model.

Parameter	Value
Region	100X100
No. of Nodes	500
Power sources	10, 50, 100, 150
Power consumer	50, 100, 150, 200, 250, 300, 350
Network	small-world, scale-free, random network
Routing	random-walk, centrality-based routing
Performance measure	Average delivery rate
No. of runs	4(1; 10; 20; 30) runs

MODEL SPECIFICATION ACCORDING TO DREAM

In this section, we present our ABM documentation according to DREAM specification approach. We describe our model using pseudocode specification part of the DREAM as follows.

Agent design

There are two types of agents which are used in the simulation model. *Agents by node types:* in our simulation model, we used complex networks. These complex networks consist of nodes which are connected through communication lines called links. These nodes agents represent producers and consumers in the network. *Agent by walker type:* for routing purpose the walker concept is deployed. These walkers were initially placed on producers nodes. They have the ability to move around the network.

Algorithm 1 Breed Node: This node agent is used to represent producers and consumers in the network

Internal Variables: < source?, target?, visited? >

- 1: **source?:** All nodes that represent sources (used for producers)
 - 2: **target?:** All nodes that represent target (used for consumers)
 - 3: **visited?:** Used to check the status of a Node agent
-

In the Node specification model, first, we described the Breed Node agent. The “Breed” is a global keyword in NetLogo (Agent-based modeling toolkit) describing a set of similar-behavior agents. As we already noted, nodes are used in the network to represent producers and consumers. After this, we specified the internal variables for the Node agent. There are three internal variables used for the Node agent. The source? variable is used to represent producers or generating unit in the power system. The target? variable is used to represent consumers in the power system environment. The last one visited? is used to check the status of the node agent whether it is visited or not. Next, we define a specification for the Walker agents.

Algorithm 2 Breed Walker: This agent is used for walker that can move around the network

Internal Variables: < location, is-finish?, location-list? >

- 1: **location:** Keep current location information of the walker
 - 2: **is-finish?:** Check the finish goal
 - 3: **location-list:** Keep all visited locations record
-

The breed Walker represents Walker agents. These walkers are deployed for routing purpose and can move around the network. Next, we specified three internal variables for the walker agents. The first one is the location. The location variable is used to keep the information about the current location of a walker. The is-finish? is a Boolean variable that returns true if the finish condition is met? The location-list is a list variable that is used as a memory with a walker. This variable keeps the information about all visited locations. After describing the agent specification model, next, we are going to present global variables specification model

Global

For the simulation setup, the key variables are five input global variables.

Algorithm 3 Input Globals: < *get-network-type, num-node, num-walker, num-source, num-target* >

Chooser:

get-network-type: Used for selecting network type from the given list

Slider:

num-node: Used for specifying number of nodes in a network

num-walker: Used for specifying number of walkers in a network

num-source: Used for creating number of producer nodes

num-target: Used for creating number of consumer nodes

Here, five input global variables are used. The *get-network-type* is input provided by “Chooser” (GUI element of NetLogo toolkit). This is used for selecting the network type from the available list. The network list comprises of the small-world, scale-free, and random network. The other four input variables are provided by “Slider” (GUI element of NetLogo toolkit). The *num-node* is used to specify the number of nodes in the network. The *num-walker* is used to specify the number of walkers in the network. The *num-source* and *num-target* are used for the specification of the number of producers and consumers respectively.

Setup procedure

Here, we present the main setup procedure for model initialization.

Algorithm 4 Procedure setup: Creating simulation environment

Input: All global parameters from the user interface

Output: Setup simulation environment

begin

- 1: clear screen
- 2: draw-selected-network
- 3: add-sources
- 4: add-targets
- 5: calculate-centrality

End

The setup procedure is the global simulation setup specification model. This is used to create the simulation model. The input parameters are provided by the user interface. The procedure starts with calling the *clear-all* function. It clears all the previous work. Next, all four individual procedures are called. The first one is “*draw-selected-network*”. Next, we describe these individual procedures.

Algorithm 5 Procedure **draw-selected-network**: Creating the desired network*Input*: network-file, list of available network*Output*: Setup the selected network**begin**

```

1: if network-type = "small-world" then
2:   [draw-small-world]
3: else
4:   if network-type = "scale-free" then
5:     [draw-scale-free]
6:   elsenetwork-type = "random-network"
7:     [draw-random-network]

```

End

“draw-selected-network” is a procedure which is used to set up the selected network from the given options (a list of networks). The procedure checks the input type and calls the appropriate function. Here, three individual procedures are used. Next, we present the specification models of these individual procedures which are called by draw-selected-network procedure.

Small-world network

In this network topology, any individual is linked to any other individual by a maximum of six edges. In social network terminology, this is called “six degrees of separation”. It has fewer nodes with more links. It is specified by: $G(V;E), Lalog(N)$.

Algorithm 6 Procedure **draw-small-world**: Creating small-world network*Input*: N*Output*: Setup small-world network

```

create-nodes N;
for all Nodes do;
  set all-wired? false;
  while all-wired? != true do
    wired-them [connect-nodes];
    set all-wired=Do-calculation [clustering-coefficient];
    Findclustering-coefficient;
    If no Node left then stop;
  end
  create link with one node to other
  calculate shortest path [set node i=1;
  count distance of node i to every other node;
  i++;
  set shortest path=min distance]
  set layout circle;

```

End**Scale-free network**

In this network, the number of links between individuals is uneven. There are some nodes which have dense connections, while some others have fewer connections. The dense connections are called hubs. These hubs have the tendency to join with other new nodes. This

network follows the power law of the degree distribution. The probability of joining new nodes with the existing hub can be defined by:

$$\prod(k_i) = \frac{k_i}{\sum_j k_j}. \quad (1)$$

In (1), k_i represents the degree of hub i .

Algorithm 7 Procedure draw-scale-free: Creating scale-free network

Input: N
Output: Setup scale-free network
set shapes circles;
create node 1;
set node=nobody;
create node 1;
[set node = new-node]
if old-node = nobody **then**
 [create link with old-node];
 while count nodes < total-nodes **do**
 [add nodes];
set layout spring;
set layout circle;
End

Random network

In this network, each individual node is formed randomly; there is no specific structure to be followed. This network can be formed by joining a vertex with other arbitrary vertices. Formally, a random network $G_R(N, P)$ is framed with edges associated with likelihood P , given that $0 < P < 1$. The connectivity of nodes does not depend on the degree of nodes.

Algorithm 8 Procedure draw-random-network: Creating random network

Input: N
Output: Setup random network
create nodes N;
for all Nodes **do**
 set shapes circles;
 set location random location;
 create-links with nodes with p;
 check $p > 0$ and $p < 1$;
End

Next, we present all other procedures that describe different processes which are called by the main setup procedure.

Setup source and target nodes procedures

Procedure “add-source” is used to setup source nodes (producers) on the network. It takes network type, nodes, links, and the number of sources from the user interface. Next, it creates the user-specified number of source nodes randomly on the network.

Algorithm 9 Procedure **add-source**: Setup sources on the network

Input: network-type, num-source
Output: Setup source nodes on the network
begin
 random nodes[
 set source true];
End

Procedure “add-target” is used to setup target nodes (consumers) on the network. It takes network type, nodes, links, and the number of targets from the user interface. Next, it creates the user-specified number of target nodes randomly on the network.

Algorithm 10 Procedure **add-target**: Setup targets on the network

Input: network-type, num-target
Output: Setup target nodes on the network
begin
 Select random nodes[
 set target true];
End

After setting source and target nodes on the network, the main setup procedure calls “calculate-centrality” procedure. Next, we present the calculate-centrality procedure.

Centrality measure

The centrality measure is widely used for measuring the relative importance of nodes within a network. It is a numerical number assigned to each node necessary for pair-wise comparisons with the whole network. There are four types of centrality measures in our model.

- 1) Degree centrality of nodes: it measures the total number of connections that a particular node has in a network. A node with a higher degree has more importance as compared with those which have a lower degree. If a node with a higher degree is removed, then it can disrupt the structure as well as the flow of the network.

Algorithm 11 Procedure **calculate-degree**: Calculates degree centrality on the network

Input: selected network
Output: calculates degree centrality for all nodes
for all Nodes **do**
 set degree **count** all connected links
 set label degree
End

- 2) Closeness centrality of nodes: it is used to find out how much data from a particular node I move to every other node t in a network. Mathematically, it can be written as:

$$C_{closeness(i)} = \sum \frac{1}{dist(i,t)}. \quad (2)$$

Algorithm 12 Procedure **calculate-closeness**: Calculates closeness centrality on the network

Input: selected network

Output: calculates closeness centrality for all nodes

for all Nodes **do**

 set closeness $C_{closeness(i)} = \sum \frac{1}{dist(i,t)}$

 set label closeness

End

- 3) Between centrality of nodes: between centrality is the process of counting the number of times a specific vertex comes in the shortest path between any two vertexes in a network. It has the capability to observe the network transmission. Mathematically, it can be written as:

$$C_{betweenness(i)} = \sum \frac{\delta_{st(i)}}{\delta_{st}}, \quad (3)$$

where $\delta_{st(i)}$ denotes the number of shortest paths between nodes s and t passing through the node I , while δ_{st} is the total number of shortest paths that exist between nodes s and t .

Algorithm 13 Procedure **calculate-betweenness**: Calculates betweenness centrality on the network

Input: selected network

Output: calculates betweenness centrality for all nodes

for all Nodes **do**

 set betweenness $C_{betweenness(i)} = \sum \frac{\sigma_{st(i)}}{\sigma_{st}}$

 set label betweenness

End

- 4) Eigen-vector centrality of nodes: it measures the impact of a particular node in a network. It defines which node is connected to the most important node in a network. It depends on neighbors in term of connection that neighbors have with other nodes in a network.

Algorithm 14 Procedure **calculate-eigenvector**: Calculates eigenvector centrality on the network

Input: selected network

Output: calculates eigenvector centrality for all nodes

for all Nodes **do**

 set eigenvector $C_{eigenvector(i)} = \frac{1}{\lambda} \sum_{t \in g} A_{adj-matrix} * X_t$ OR $Ax = \lambda x$

 set label eigenvector

End

Procedure go

To validate our model, we apply routing techniques on our developed models. There are two routing techniques that we used in our work. The first one is a random walk and the second one is centrality-based routing. Next, we present the details and specification models of these two routing techniques described as procedures.

- 1) Procedure random-walk: in case of a random walk, the walkers are set initially on the source nodes. They search their neighbors and select one of them randomly. This process goes repeatedly until all the target nodes have been visited. Let us consider an undirected graph $G(V, E)$, a random walk is a stochastic process that starts from a given vertex, then select one of its neighbors randomly to visit next. It has no memory that keeps information on previous moves. It stops when the termination condition meets.

Algorithm 15 Procedure **random-walk**: Used for routing on network

Input: All input parameters provided by user interface

Output: End to end delivery from sources to the destinations

Execution: Called repeatedly on simulation execution

begin

- 1: start from sources
- 2: **while** *is – finish* \neq *true* **do**
- 3: search neighbors
- 4: select any of neighbor
- 5: **move** to the selected

End

- 2) Procedure centrality-rw: the procedure “centrality-rw” is another approach which is used in our work for the routing purpose. The technique works as follows. The walkers search for that neighbor node which has maximum centrality value, then move to the selected node. If they find no node with maximum centrality value, then they select randomly a node from the neighbors. The process of centrality-based routing is shown in Figure 4.

Algorithm 16 Procedure **centrality-rw**: Used for routing on network

Input: All input parameters provided by user interface

Output: End to end delivery from sources to destinations

Execution: Called repeatedly on simulation execution

begin

- 1: start from sources
- 2: **while** *is-finish* \neq *true* **do**
- 3: get-list-of-neighbors
- 4: **if** *is-list-empty*=*true* **then**
- 5: die
- 6: stop
- 7: Else
- 8: Search node with max-centrality in the list
- 9: **if** node with max-value exist= *true* **then**
- 10: select node with max-value
- 11: Else
- 12: select any of node in the list
- 13: **move** to the selected

End

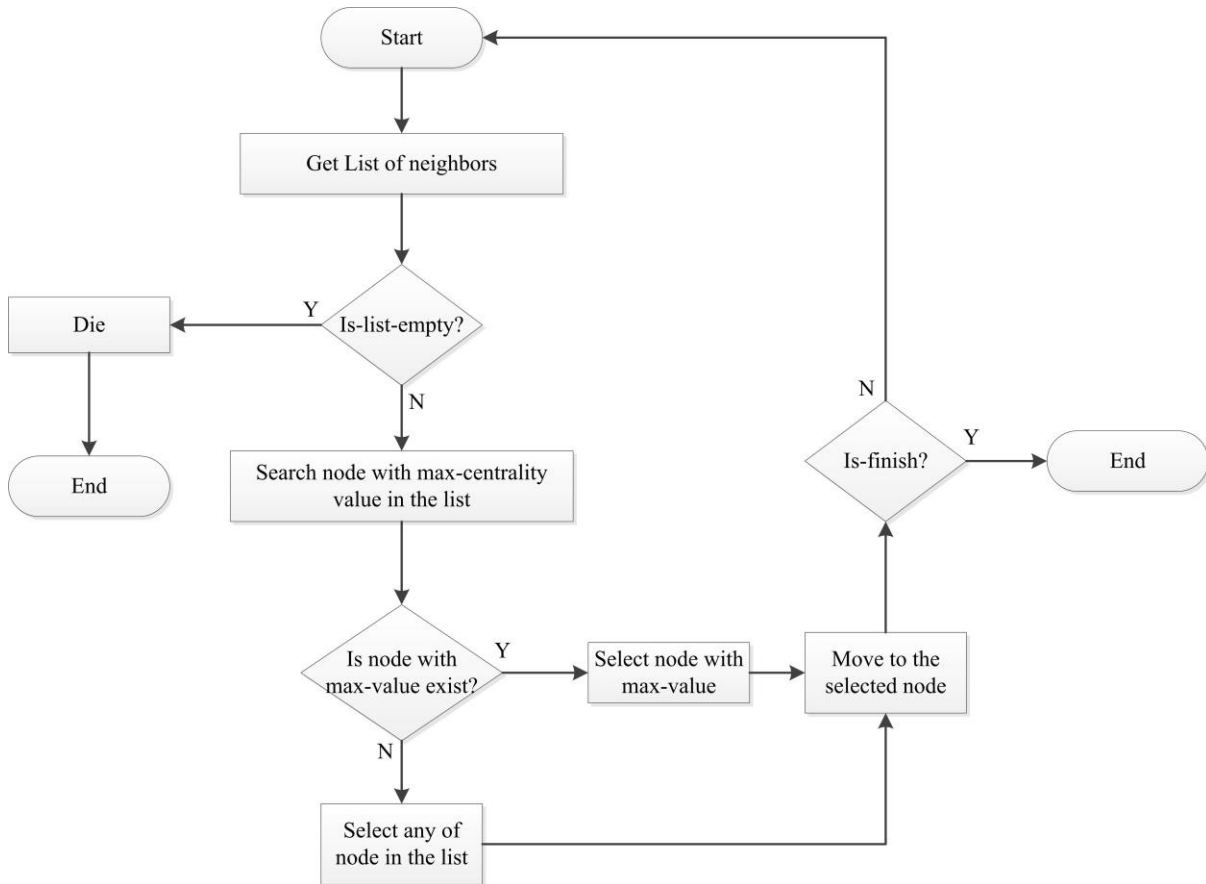


Figure 4. Flowchart for the centrality-based routing algorithm. We can see that the algorithm starts with making a list of neighbor nodes of the current location. In the next step, it checks the content of the list. If the list is empty, then the simulation stops, otherwise, the algorithm searches for the node having maximum centrality value. If such a node exists, it selects that node, otherwise, it selects any node from the neighbor’s list. The next step is to move to the selected node. After this, it checks the terminating condition. If it is satisfied, the simulation stops, otherwise the control goes to the first step.

3) Generate plot: “dot-plot” procedure is used to plot the execution of the simulation of the model. Here, this procedure plots two types of information. First, at each time step, it counts the number of nodes that have been visited. Second, it counts the number of consumers that are visited. Next, we present model specification and details the experiments with our model

Algorithm 17 Procedure **generate-graph**: Generate graphs of the current simulation state

Input: No input required

Output: Generate plot on simulation execution

Execution: Called by Go procedure

begin

Plot the number of nodes that are visited

Plot the number of targets visited

End

Performed experiment

Two types of experiments are performed in our model simulation. The first one is used for the random walk algorithm, the second is used for the centrality-based routing algorithm.

Algorithm 18 Experiment: Demonstrates the effect of random walk and centrality routing on different networks

Input: < num – nodes, num – sources, num – targets, num – walkers, network – type >
Setup Procedure: < setup >
Go Procedure: < random – walk, centrality – rw >
Repetition: 10

Input:

num-nodes: 500
num-sources: 10, 50, 100, 150
num-targets: 50, 100, 150, 200, 250, 300, 350
num-walker: 10, 50, 100, 150

network-type: small-world, scale-free, random-network

Reporter:

measure-nodes: count number of nodes that have been visited
measure-targets: count number of targets that have been visited

Stop condition:

If all targets are visited, Stop

CENTRALITY-BASED ROUTING ALGORITHM: TIME COMPLEXITY ANALYSIS

In this section, we present the centrality-based routing algorithm time complexity analysis. The time complexity of our proposed centrality-based routing is a linear function of n that is $O(n)$. In the algorithm analysis, we analyze the cost and number of times that each step takes for execution. All steps take constant time, except steps 2 and 8 which take n times for execution. In step 2, the while loop executes n times. In step 8, the algorithm searches the list of nodes and then selects the node with the largest value. So it takes n times. The total running time is the sum of the running times and costs of each step in the algorithm.

Procedure **centrality-rw:** Used for routing on network

Input: All input parameters provided by user interface
Output: End to end delivery from sources to the destinations
Execution: Called repeatedly on simulation execution

begin	Cost	time
1: start from sources	C_1	1
2: while is – finish \neq true do	C_2	n
3: get-list-of-neighbors	C_3	1
4: if is-list-empty=true then	C_4	1
5: die	C_5	1
6: stop	C_6	1
7: Else		
8: Search node with max-centrality in the list	C_7	n
9: if node with max-value exist= true then	C_8	1
10: select node with max-value	C_9	1
11: Else		
12: select any of node in the list	C_{10}	1
13: move to the selected	C_{11}	1

End

$$T(n) = C_1(1) + C_2(n) + C_3(1) + C_4(1) + C_6(1) + C_7(n) + C_8(1) + C_9(1) + C_{10}(1) + C_{11}(1) \quad (4)$$

$$c = (C_1 + C_3 + C_4 + C_5 + C_6 + C_8 + C_9 + C_{10} + C_{11}) \quad (5)$$

$$C_2(n) + C_7(n) = (C_2 + C_7)n \quad (6)$$

$$a = C_2 + C_7 \quad (7)$$

By putting the Eq. 5 and 7 values in Eq. 4, we get:

$$T(n) = an + c \quad (8)$$

$$T(n) = O(n) \quad (9)$$

RESULTS AND DISCUSSION

In this section, we present results obtained from the DREAM methodology, then we compare ODD vs DREAM, followed by an empirical analysis and study of some significant representative related researches.

COMPLEX NETWORK OF THE MODEL

Figure 5 shows the complex network of our proposed ABM. This network presents the developed ABM in a visualized form. We develop the network model using Gephi (a network toolkit).

First, we start from the root node “ABM”. This node is expanded into leaf nodes “global variables, agent, procedure, and expts”. The global variables are further expanded into global output and input. These are inputs provided by the user interface to the model. The “agent” node represents the involved entities in the model and it is further expanded into two types of agents that are named as node and walker. After defining global inputs, agent, our next focus is on the “procedure” node. The “procedure” node is the root of all procedures used in the model. This node has the highest node degree in the network model. The node “expts” is the parent for all sub-nodes that represent different experiments carried out during simulation.

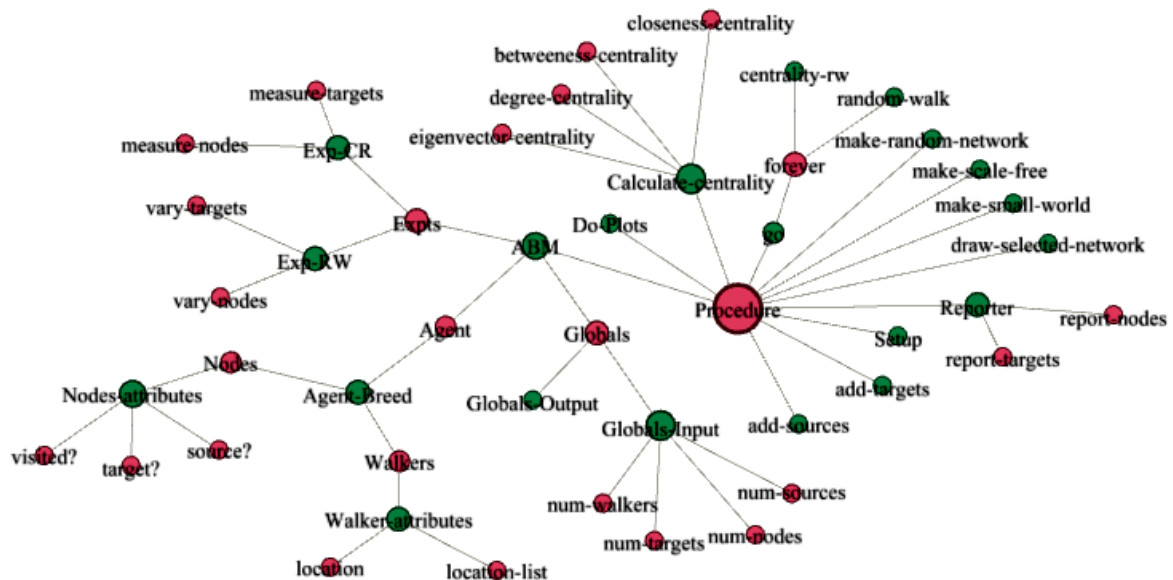


Figure 5. The network model of our proposed ABM for the smart grid. It can be noticed that the root node in the network is “ABM” which is connected to four nodes named “globals, agents, procedure, and expts”. The “globals” node is connected to the globals output and input parameters. The “agent” node is connected with the involved entities in the model. The “procedure” node is connected with all other processes and functions in the model. The “expts” node is connected with all experiments carried out in the model simulation.

SOCIAL NETWORK ANALYSIS

In this section, we present the results obtained by applying social network analysis (SNA) on the network model. The SNA provides quantitative measures to give network topological details. Using these quantitative measures we can perform a comparison of different models.

In Figure 8 degree centrality of the network is plotted. It shows that the procedure node has the highest centrality. Next is the ABM node and third the global input.

In Figure 9 betweenness centrality is presented. It demonstrates that the ABM node has the highest betweenness centrality and the procedure node exhibits the second highest betweenness centrality.

Figure 10 shows the closeness centrality of the network model. It demonstrates that the ABM node has the highest closeness centrality and the procedure node is the second.

Figure 11 shows the eigenvector centrality of the network. It shows that the procedure node has the highest eigenvector centrality followed by the ABM node.

COMPARISON OF ODD AND DREAM

In this section, we provide a qualitative as well as a quantitative comparison between ODD and DREAM specification techniques.

The ODD specification allows a textual-based description of ABM with the purpose to make model readable and it promotes the rigorous formulation of models. It comprises a checklist that covers key features through which one can describe an ABM. The ODD specification also has some limitations that are described in the following.

The ODD specification only provides a textual-based description of ABM. Sometimes for large ABMs, such textual-based description is insufficient to cover all the features of the ABM. It has no quantitative assessment of the ABM on the basis of which one could perform a comparison between different ABMs. Reviewing and comparison of different ABMs are difficult. For comparison and classification purposes, the only possible way is to make a table and put together ODD checklists of different ABMs and then search for similarities and differences.

According to [8], a survey was conducted from 2006 to 2009 of those publications in which ODD was used. According to this survey, only 75% of publications used ODD correctly, while 25% of publications used ODD incorrectly and some parts of the method were compromised. The author formulates the conclusion that it is difficult to write an ABM specification by following ODD method.

Another issue which is identified in the ODD specification is redundancy. Some parts of the specification like the purpose section are also included in the introduction section of the document. The design concept section is also repeated in the sub-model section of the model. The sub-model section is repeated in the process of scheduling sections.

Sometimes there may be different publications with a different version of the same ABM. Then these publications have the same ODD with little modification in the entities and process sections. Another limitation of ODD is that the textual-based description is too specific which is not useful for replication of ABMs; there exist some ambiguities and misunderstanding about ABMs.

On the other side, DREAM allows a detailed specification of ABMs. It comprises of making a complex network of the model, pseudocode specification, and network analysis steps.

This method allows for inter-disciplinary comparative study and communication among different scientific domains. So, if a model is developed in social science, it can be compared visually and quantitatively with a model in biological science and vice versa. For instance,

the social model for aid spreading presented in [15] and biological model developed for the emergence of snake-like structure in [15], by developing complex networks of both ABM models, we can easily compare and analyze both networks in the same manner. DREAM specification approach can be applied to any ABM of any research domain.

DREAM allows presenting ABM in the form of a complex network model. This allows reading and understanding ABM visually without going to the code specification. Performing network analysis on this network of ABM, it also gives a quantitative measurement of the ABM. These quantitative measures are the digital footprint of ABM and can be used to compare different ABMs.

DREAM further allows pseudocode specification and details of the ABM. This specification helps to understand ABM completely and independently of a particular scientific discipline. This specification then offers a translation to the code and facilitates further developing of ABM. It follows that by using DREAM, any ABM can be replicated easily.

Next, we carry out an empirical assessment of ODD and DREAM specification methods, using an evaluation method presented in [17]. In the study, the author presented an empirical analysis of the Ontology-engineering methodologies using different key features as presented in Table 2. That is the first reason we also used the same feature in our study. We selected 10 key features that are desirable and being good for the specification methods.

The purpose of methodologies assessment is to identify which features in the methodology to what extent. In other word, we only consider the level of the feature offered by the specification methodology. We used $H = 2$, $M = 1$, and $L = 0$ for evaluation purpose that demonstrates which methodology offers which features to what extent. For example, if a methodology fully support a specific feature, we assign it H , if a methodology partially support a specific feature, we assign it M , and if a methodology does not support a specific feature, then we assign it L . In last, we compute the rank of each methodology by averaging the results as shown in table 3. The description of the weight assignment is given as follows.

Social and technical process

Both ODD and DRAM provide a description of social and technical processes in the specification, so we assign weight H to both methods.

Adaptability

The DREAM approach is so flexible that can be adopted in the different scientific domain. The pseudocode specification of DREAM allows researchers to implement the model in any programming language. On the other side, the ODD method is a textual-based approach does not target multiple scientific domains. In previous studies, the ODD method is only reported mostly in social sciences. In the evaluation process, we assign L and H to ODD and DREAM respectively against the adaptability feature.

Reusability

In the case of reusability, it can be noted that the ODD specification cannot be used for further development of the models, while the DREAMs pseudocode specification can easily be used for the development purpose. In this case, we assign L and H weight to ODD and DREAM, respectively.

Stepwise approach

Both ODD and DREAM provide sequential steps for documenting a model. That's why we assign equal weight to both ODD and DREAM against the stepwise feature.

Documentation

The ODD offers insufficient documentation of the model while DREAM covers the complete specification in the documentation process. In this case, we assign weights M and H to ODD and DREAM respectively against documentation feature.

Network model

The ODD does not support developing the network model of understudy system, while the DREAM facilitates researchers for developing a network model of understudy system. In this case, we assign weights L and H to ODD and DREAM respectively against the network model feature.

Pseudocode specification

The ODD method does not support the pseudocode specification feature in the documentation process. The DREAM method offers pseudocode specification in the documentation process. In this case, we assign weights L and H to ODD and DREAM respectively against the pseudocode specification feature.

Network analysis

The ODD method does not include the social network analysis approach during the documentation process, while the DREAM offers social network analysis feature. In this case, we assign weights L and H to ODD and DREAM respectively against network analysis feature.

Communication

The ODD does not provide communication across different scientific domains, while the DREAM does. In this case, we assign weights L and H to ODD and DREAM respectively against communication feature.

User satisfaction

The ODD method ignores the user satisfaction feature in the specification process, while DREAM includes user satisfaction in the specification process. In this case, we assign weights L and H to ODD and DREAM respectively against user satisfaction feature.

Table 2. Selected features for empirical analysis of ODD and DREAM.

Feature	Description
<i>Social and technical process</i>	Considers the level of social and technical aspects in the methodology
<i>Adaptability</i>	Referring to how much the methodology is flexible to adoption in different domains
<i>Reusability</i>	It refers to the extent of the methodology to be used for model replication
<i>Stepwise</i>	Measures how much methodology is based on sequential steps
<i>Documentation</i>	It involves the process of documenting the model
<i>Network model</i>	It is the developing of the network for the model
<i>Pseudo-code</i>	Concerns with the presenting pseudo-code specification for the model
<i>Network analysis</i>	Concerns with applying social network analysis tool on network
<i>Communication</i>	Refers to the level of communication among multiple disciplines
<i>User satisfaction</i>	Refers to the level of convenience

Table 3. Empirical analysis of ODD and DREAM.

Feature	ODD	DREAM
F(1)	<i>H</i>	<i>H</i>
F(2)	<i>L</i>	<i>H</i>
F(3)	<i>L</i>	<i>H</i>
F(4)	<i>H</i>	<i>H</i>
F(5)	<i>M</i>	<i>H</i>
F(6)	<i>L</i>	<i>H</i>
F(7)	<i>L</i>	<i>H</i>
F(8)	<i>L</i>	<i>H</i>
F(9)	<i>L</i>	<i>H</i>
F(10)	<i>L</i>	<i>H</i>
Rank	0,5	2,0

EXPERIMENTAL EVALUATION

Simulation setup

To simulate a smart grid-based complex scenario, we developed small-world, scale-free, and random complex networks using agent-based modeling approach. In order to validate our work, we applied routing techniques such as random walk and centrality-based routing on large-scale complex networks, specifically in the smart grid domain. For comparison, we applied random and centrality-based routing on these networks and analyzed their behavior on these networks.

Evaluation metrics

For performance evaluation purpose, we used the average delivery rate parameter. The average delivery rate is defined as the number of packets sent by sources and successfully received by consumers. Mathematically, it can be written as follows:

$$\sum_1^n \frac{Ds}{Dc} 100, \quad (10)$$

where Ds represents data packets sent by the sources and Dc represents data packets received by consumers. The experiments were performed for different case studies such as different numbers of consumers and generation units. Then the simulation results were averaged over 30 executions. To see the behavior of routing techniques, when going from source locations towards the destinations through different paths at each time steps, we used different parameters and observed for which combination it took less convergence time. The simulation environment is set according to the parameters as shown in Table 1.

Results of Random walk routing

We applied the random walk routing technique on different complex networks. The simulation results demonstrate that the random walk routing technique showed less iteration in case of small-world topology as compared to other network topologies.

For the small-world network, Figure 12a shows the simulation results for different numbers of sources and consumers. This shows the convergence rate in different case studies. The results show that the convergence rate lies between 180 and 280 iterations.

For the scale-free network, Figure 12b shows simulation results for the different numbers of sources and consumers. Random walk shows a slow convergence rate as compared to the small-world network.

For the random network, Figure 12c shows simulation results for different numbers of sources and consumers. On this network topology, random walk demonstrates very slow convergence rate as compared to both the scale-free and the small-world networks. This is due to the network topology.

Figure 12d shows the performance of random walk on different network topologies. The small-world topology demonstrates less iteration while the random network has a slow convergence rate.

Issues with Random walk routing

- 1) Agents can move randomly on the network, they select a random node from their neighbor's list.
- 2) Agents can move to previously visited nodes.
- 3) Agents do not maintain records when traversing nodes of the network.
- 4) Sometimes, agents get stuck on the network, which increases computational time.

Results of Centrality-based routing:

In this section, we discuss the simulation results that were carried out using centrality routing (CR) algorithms on different networks.

Figure 13 shows the simulation results based on betweenness centrality routing on different networks. For the small-world network, figure 13a shows the simulation results for different numbers of sources and consumers. The results show that the convergence rate lies between 30 to 50 iterations. For the scale-free network, figure 13b shows simulation results for different numbers of sources and consumers. Betweenness centrality routing shows slow convergence rate as compared to the small-world network. For the random network, figure 13c shows simulation results for different numbers of sources and consumers. On this network, betweenness centrality routing demonstrates very slow convergence rate as compared to both the scale-free and the small-world networks. This is due to network topology. Figure 13d shows the performance of betweenness centrality routing on different network topologies. The small-world network demonstrates less iteration while the random network has very slow convergence rate.

Figure 14 shows the simulation results based on closeness centrality routing. For the small-world network, Figure 14a shows the simulation results for different numbers of sources and consumers. The results show that the convergence rate lies between 30 to 50 iterations. For the scale-free network, Figure 14b shows simulation results for different numbers of sources and consumers. Closeness centrality routing shows slow convergence rate (between 150 and 250) as compared to the small-world network. For the random network, Figure 14c shows simulation results for different numbers of sources and consumers. On this network, closeness centrality routing demonstrates very slow convergence rate as compared to both the scale-free and the small-world networks. Figure 14d shows the performance of closeness centrality routing on different network topologies. The small-world topology demonstrates less iteration while the random network has very slow convergence rate.

The simulation results based on degree and eigenvector centrality routing on different networks have been shown in Figures 15 and 16, respectively. In case of degree and eigenvector centrality routings again the small-world network showed fast convergence rate as compared with other networks.

Figure 17a shows centrality routing on the small-world network with different numbers of consumers and generating units. The simulation results show that on average, each centrality routing has an equal convergence rate. When it is compared with other complex networks, it

is found through simulation results based on centrality routing, the small-world network has lower convergence rate compared with the other networks, in this case, centrality routing on small-world has convergence rate between 30 to 50 iterations.

Figure 17b shows simulation results of the centrality routing on the scale-free network with different numbers of consumers and generation units. The simulation results demonstrate that on the scale-free network, degree centrality routing has a slow convergence rate as compared to other approaches.

Figure 17c shows simulation results of degree, closeness betweenness, and eigenvector centrality routing applied on the random network using different numbers of consumers and generating units. The figure shows that degree centrality routing has lower convergence time as compared to other centrality routing techniques.

Random-walk vs Centrality-based routing

Figure 18a shows the simulation results of different routing techniques on the small-world network. It demonstrates that centrality routing techniques have a similar convergence rate while the random walk has a large convergence rate.

Figure 18b shows simulation results of different routing techniques on the scale-free network. It demonstrates that degree centrality routing and random walk have large convergence time.

Figure 18c shows simulation results of different routing techniques on the random network. It shows that a random walk has large iterations as compared to other routing techniques.

COMPARISON WITH PREVIOUS WORKS

In this section, we present an overview of the previous studies in the smart grid domain. The purpose of this section is to highlight the gaps in the current literature of the smart grid. We focus on the agent-based, complex systems and specification methods for the developing of the smart grid models. From our review, we noticed that the previous agent-based studies do not focus on the specification aspects of the models. A comparative analysis has been shown in Table 4.

In [18], the authors developed a conceptual model for the energy system. This model is integrated with the ODD methodology for documenting ABM. In this model, some other concepts were added like layers, objects, actor and working point to bridge the social and technical systems in the energy model. However, this conceptual model was not validated through ABM.

In [19], the authors proposed a check-in based routing approach for network traffic model. In this work, betweenness centrality was used to assign node as the check-in node between source and destination. The proposed routing strategy was implemented on the scale-free network. However, the optimization of routing remained an open problem in this work.

In [20], the authors presented agent-based tools for modeling and simulation of self-organization in a wireless sensor network. They demonstrated the usability of NetLogo agent-based tool and developed different experiments that show how to model different scenarios in the sensor network domain.

The paper [21], proposed a routing technique for large-scale sensor network-based environments. In this work, local and global updating strategy is introduced for maintenance and efficient routing in the network. This approach monitors any changes in the network and updating the routing path according to the situation. Results demonstrate the effectiveness of the techniques and a reduced end to end delivery rate as compared to the previous techniques.

Wang et al. in [22], worked on frequency synchronization in the power grid system. In this study, the network theory concept was used to monitor, control and exploit the frequency variation of the power system.

Jia et al. [23], worked on security analysis using complex network approach in the power system. In this work, the power adjacency matrix approach is proposed for the analysis and measurement of the power flow and activities of each node and links on the network.

In [24], the author proposed a novel routing strategy based on betweenness centrality in a complex network. In this work, scale-free network is used and routing was performed based on expanding betweenness of each node. This method shifts the load from the node with higher betweenness to the lower.

The study in [25] proposed Honey bee optimization-based routing using the random network-based environment in the smart grid domain. However, this work was not validated on different standard network topologies such as small-world, scale-free and random network. Other limitations of this study are using limited numbers of geographic spaces as well as a limited number of communication ranges.

In [26], the author proposed an efficient probability routing strategy using scale-free complex network topology. This method utilizes the probability concept for redistributing load from critical nodes to the non-critical nodes. Results showed that the routing path is reduced by 30% as compared with a previous probability routing technique.

In [27], the authors presented a novel routing strategy in the wireless sensor network and proposed sink betweenness distributed routing algorithm. In this approach, betweenness of each node is calculated in which the sink node exists as a terminal node. This work was implemented on the random network.

The study in [28], proposed a multi-agent system for the reactive power control system in a smart grid. This proposal reduced power losses and provided the exploitation of available power resources. In [29], the author presented a voltage variation control strategy. This approach controls the voltage profile in the specified range of the studied system, which results in reducing system loss and improving system reliability.

Authors in [30, 31] have worked on fault location and restoration in smart grid by using the complex network approach. These studies demonstrate the modeling of fault location and restoration process in a distributed power network. Likewise, the study [32] proposed the use of the neural network for an adaptive protective system in a large-scale power system.

Authors in [33] worked on fault location and proposed the use of particle swarm optimization the technique for locating voltage disturbance sources in a distributed power grid.

In [34], the authors proposed another novel approach for voltage diagnosis and fault detection in power distribution system. They proposed a negative selection and clonal algorithm inspired by the biological immune system. This method can learn unknown patterns in the system without going to the initial state. The results demonstrate 99% accuracy. Another study in [35] also focused on voltage stability and proposed the use of the genetic algorithm for determining optimal power sources. Likewise, another work [36] was done on contingency selection in a power system network. The proposed system was tested on an IEEE-30 test system and results showed 100% accuracy rate.

Regarding communication management, different studies also have been presented in the smart grid. Wang et al. in [37], proposed an adaptive strategy for energy trading between the utility grid and consumers. In this proposal, each agent can communicate with each other for sharing information about energy usage and cost. In [38], authors have worked on distributed

large-scale consumers load with the conjunction of renewable energy resources. In this work, a neighbor communication strategy is applied. This results in low communication cost. Kremers et al. in [39] presented a bottom-up approach for smart grid modeling. It consists of two layers; physical layer for electrical power transmission and logical layer for communication. This model has the ability to integrate new devices in the smart grid environment. It provides dynamic load management, power, communication control and monitoring.

CONCLUSIONS

In this article, we proposed a novel ABM developing approach by using specification methods such as ODD and DREAM for the smart grid system. The proposed method guides the researcher for developing an ABM starting from conceptual models to validated ABM through simulation. The work is exemplified by considering a case scenario from the smart grid system. We showed how to effectively model the smart grid system by using specification methods. We demonstrate the usefulness of the proposed approach in terms of modeling a complex system, ease of use, and knowledge transfer. The proposed method also supports communication between multidisciplinary researchers. We presented qualitative as well as a quantitative comparison of both ODD and DREAM specification techniques. The comparative study of ODD and DREAM proved that DREAM methodology is the more useful approach for documenting an ABM not only in terms of modeling but also for replication of the models, specifically in the smart grid domain.

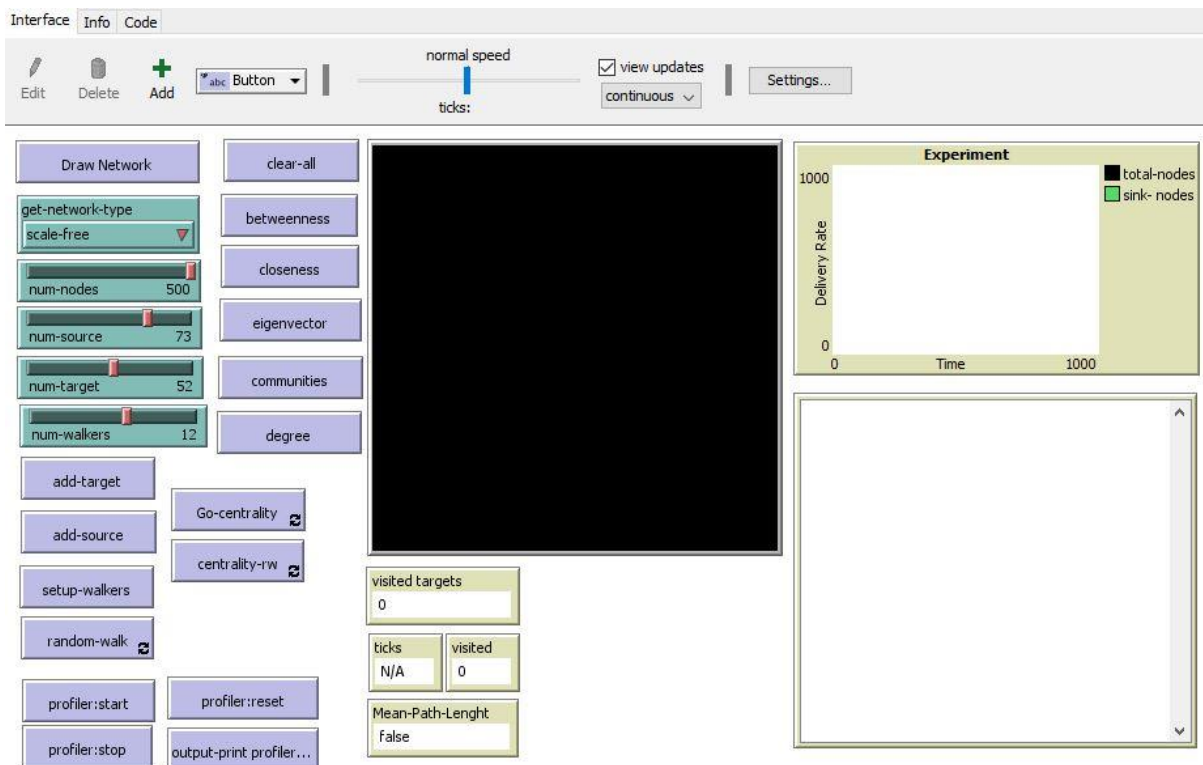
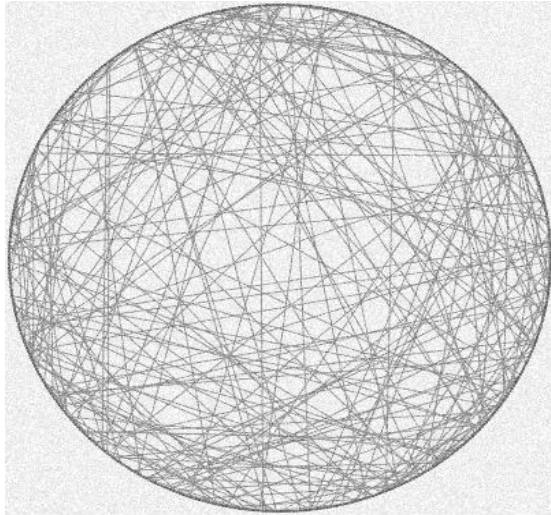
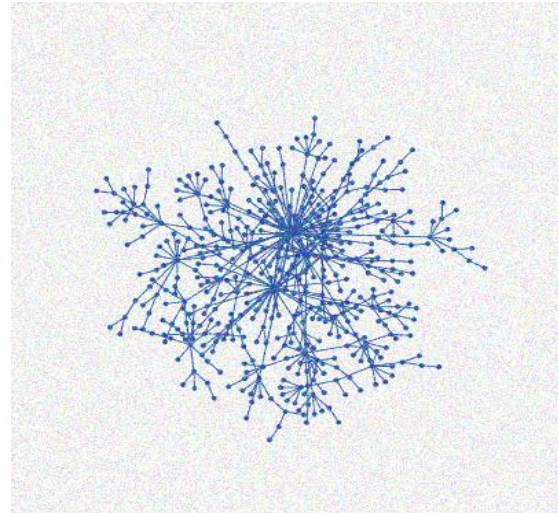


Figure 6. Screenshot of the developed ABM of the smart grid. The image shows the user interface of the NetLogo simulation tool. It consists of sliders, chooser, monitors, buttons, and the world (a simulation environment).

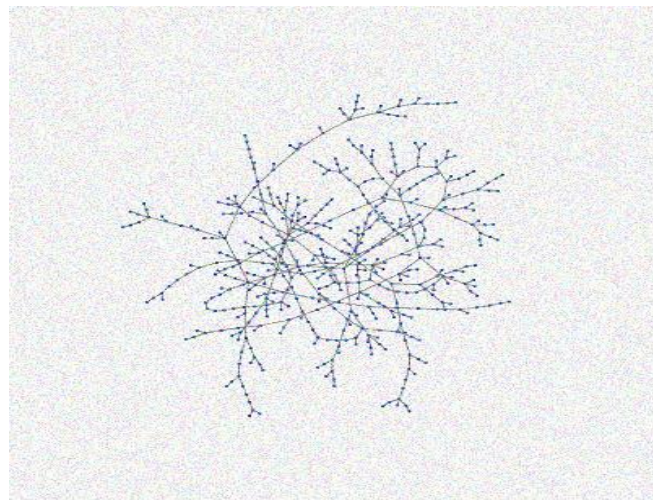
Here we would like to mention that the presented work only focuses on communication among power sources and consumers using routing approaches in the complex networks of the smart grid. However, other than communication, there are several key problems like demand response management, power scheduling, fault control, and storage management. Study on these aspects is also needed from a complex network perspective. These studies will lead to the better utility of complex networks in the smart grid domain.



(a) Small-world network



(b) Scale-free network



(c) Random network

Figure 7. Developed smart grid scenarios based on standard complex networks in our study. These networks were developed using NetLogo tool: Part (a) shows the small-world consisting of 500 nodes, the number of consumers and sources are selected randomly. Part (b) shows the scale-free network with 500 nodes, the number of consumers and sources are selected randomly. Part (c) demonstrates the random network consisting of 500 nodes, the number of consumers and sources are selected randomly.

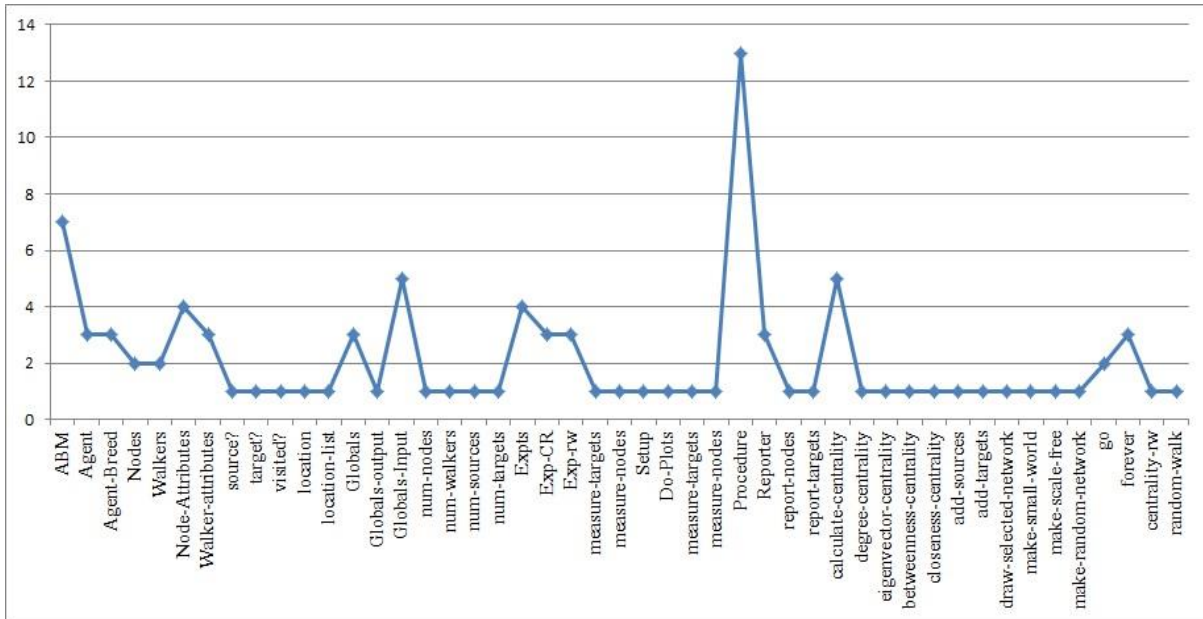


Figure 8. Degree centrality of the network. It shows that the Procedure node has the highest degree in the network.

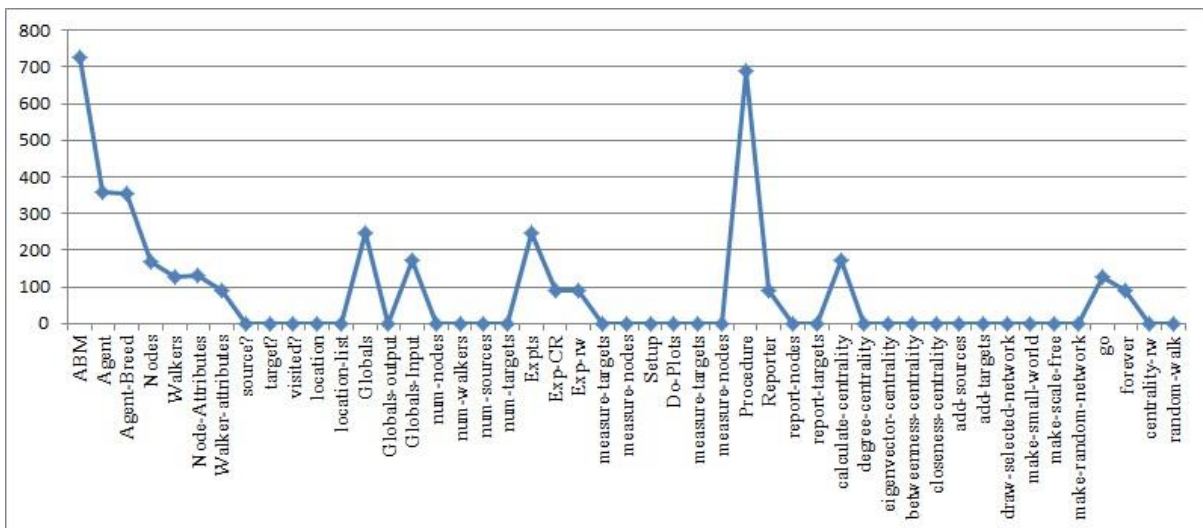


Figure 9. Betweenness centrality of the network. ABM node has the highest betweenness centrality value in the network.

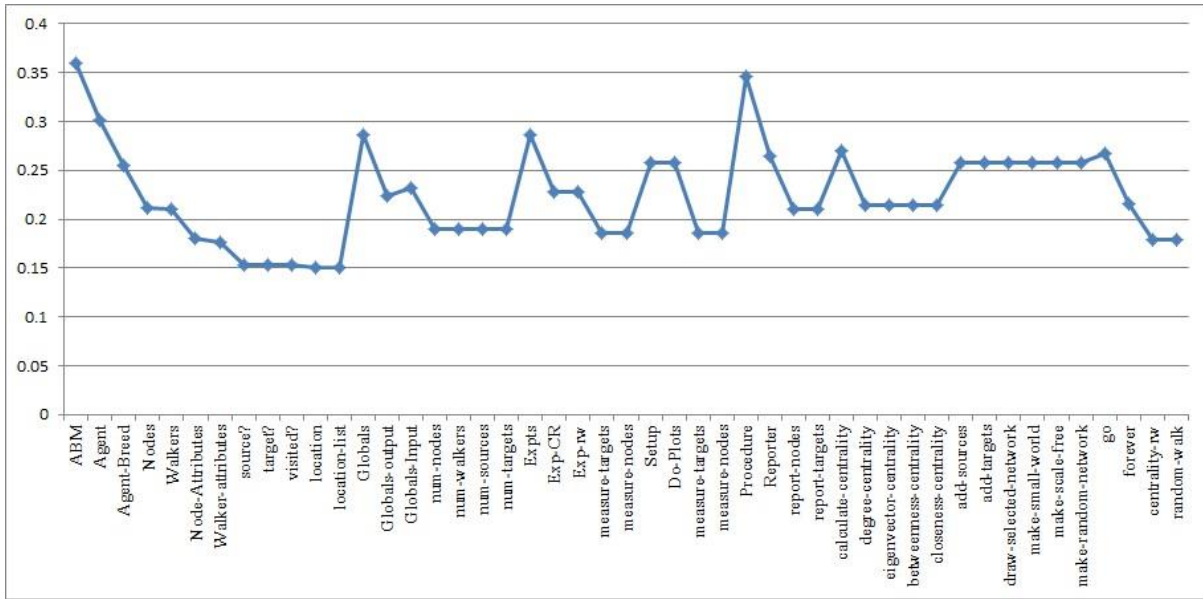


Figure 10. Closeness centrality of the network. ABM and Procedure node has the highest closeness centrality value.

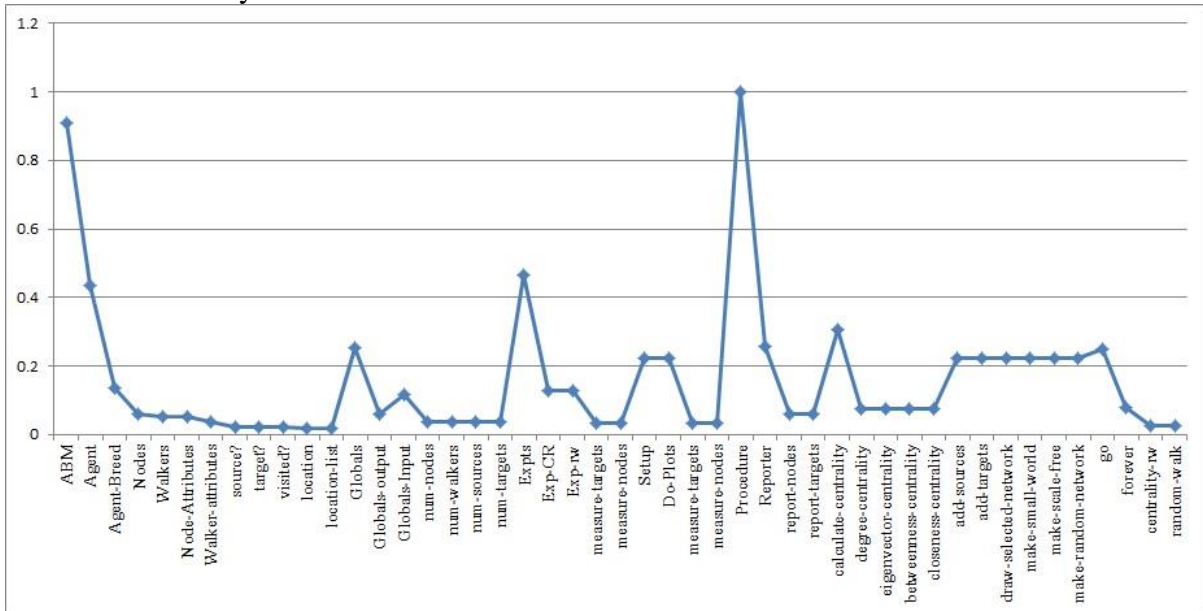
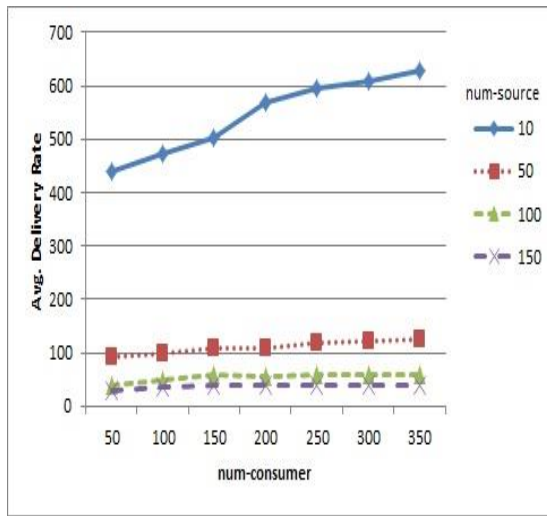
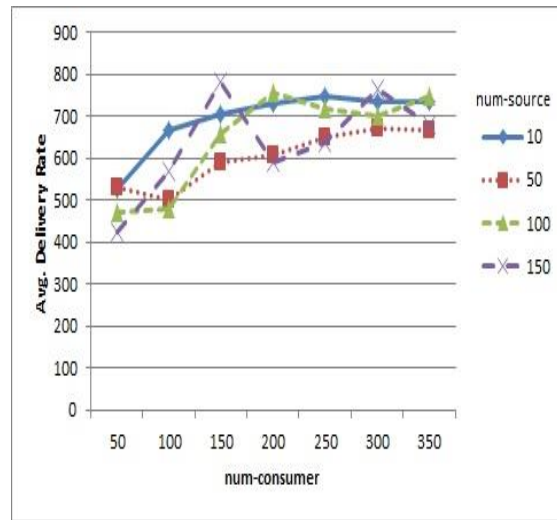


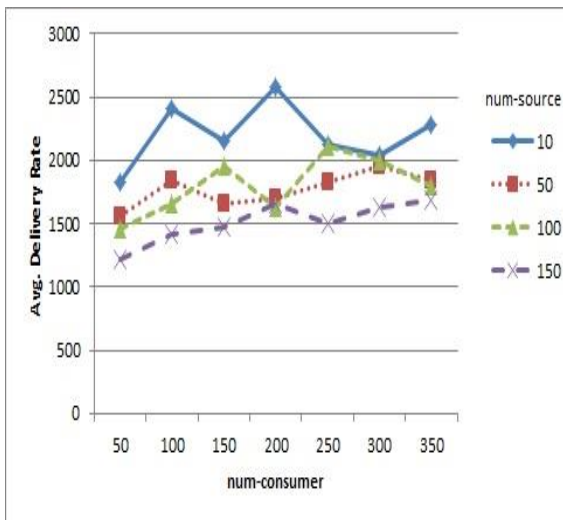
Figure 11. Eigenvector centrality of the network. It shows Procedure node is on the top of the list in the network model.



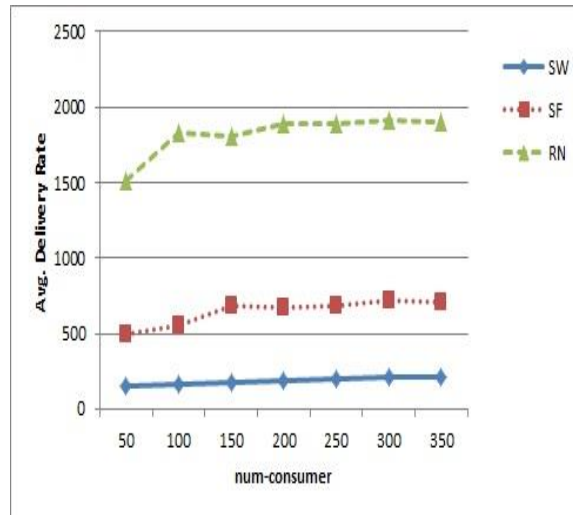
(a) Random walk on Small-world



(b) Random walk Scale-free

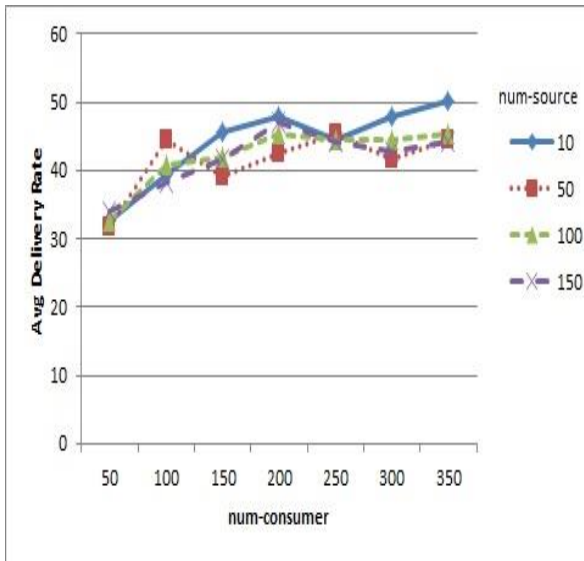


(c) Random walk on Random network

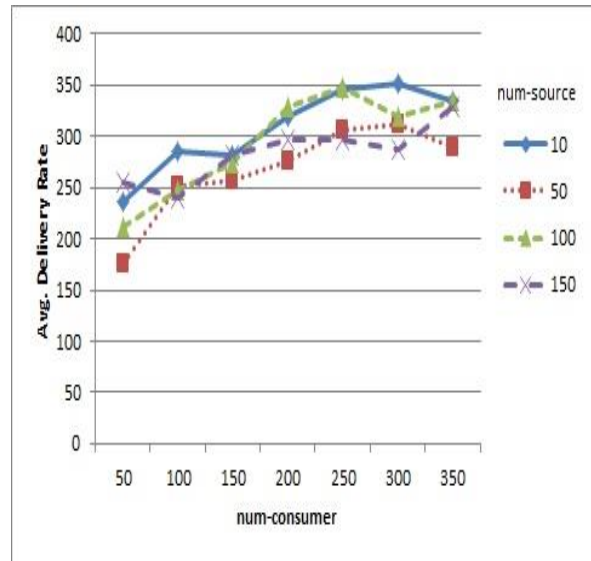


(d) Random walk on all networks

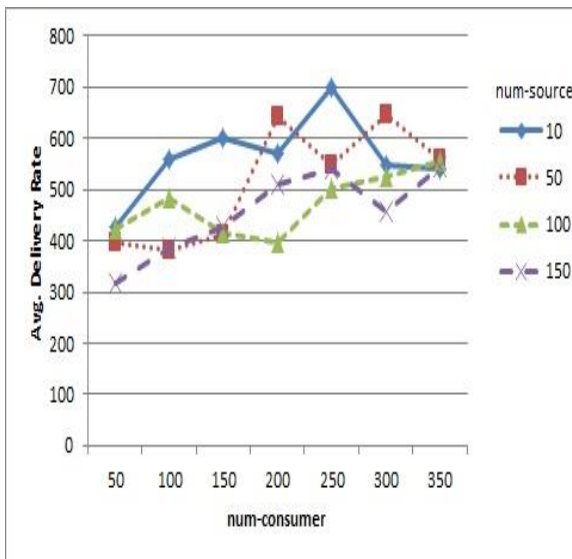
Figure 12. The simulation results of Random walk on different networks: Part (a) shows the results of a random walk on the small-world network. The x -axis shows a number of consumers and the y -axis shows average end to end delivery rate against a different number of sources. Part (b) shows the results of a random walk for different numbers of consumers and sources on the scale-free network of five hundred nodes. Part (c) shows the results of a random walk on the random network. Part (d) shows random walk results on all networks. Results of all three networks are compared (numbers of consumers: 50, 100, 150, 200, 250, 300, 350, number of sources: 10, 50, 100, 150).



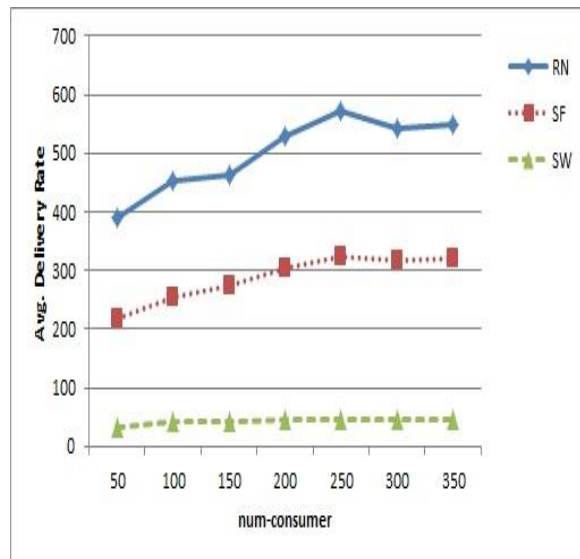
(a) CR-betweenness on Small-world



(b) CR-betweenness Scale-free

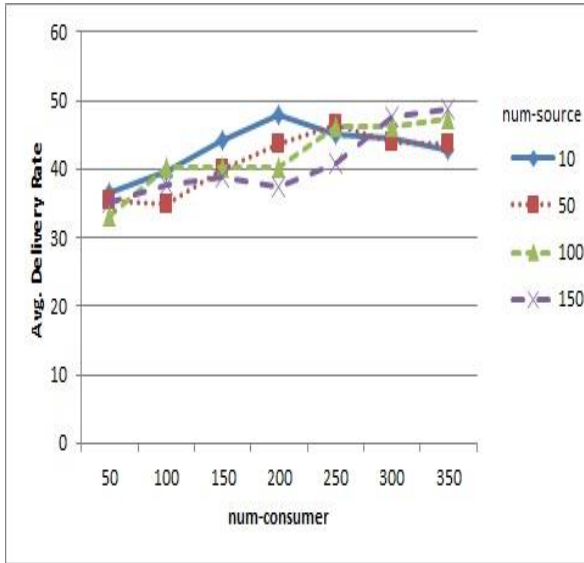


(c) CR-betweenness on Random network

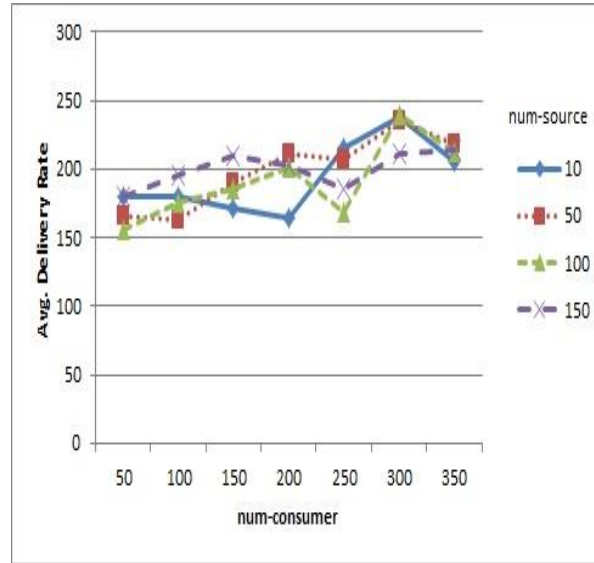


(d) CR-betweenness on all networks

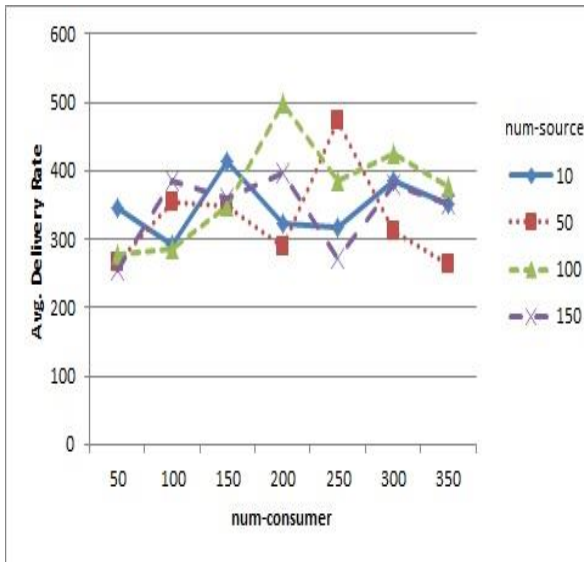
Figure 13. The simulation results of betweenness centrality routing on different networks. Each network is composed of 500 nodes. The simulation was carried out on different numbers of sources and consumers (sources: 10, 50, 100, and 150, consumers: 50, 100, 150, 200, 250, 300, and 350). Part (a) shows simulation results on the small-world network. Part (b) shows simulation results on the scale-free network. Part (c) shows simulation results on the random network. Part (d) shows betweenness centrality routing on different networks.



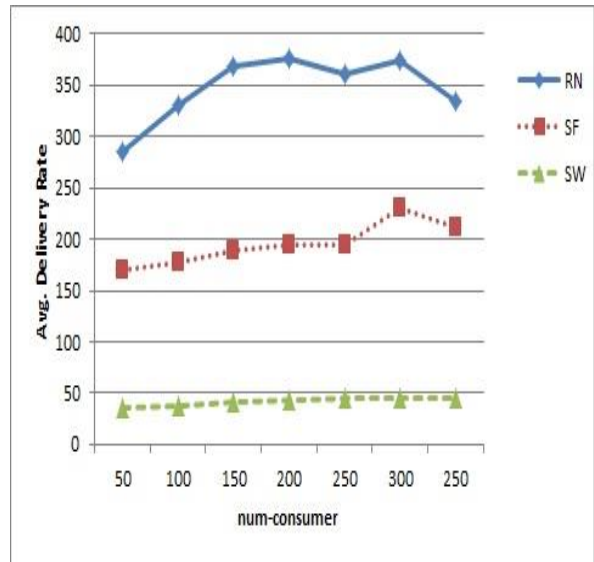
(a) CR-closeness on Small-world



(b) CR-closeness Scale-free

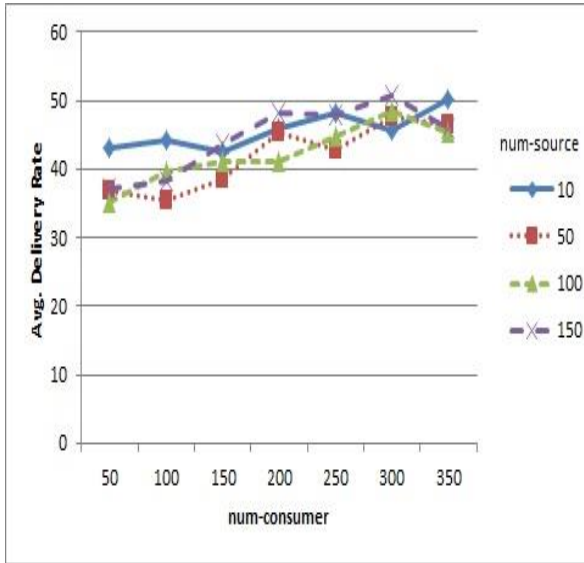


(c) CR-closeness on Random network

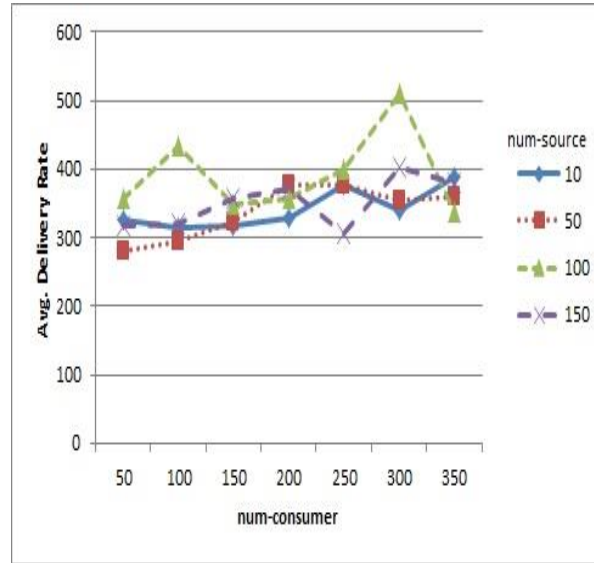


(d) CR-closeness on all networks

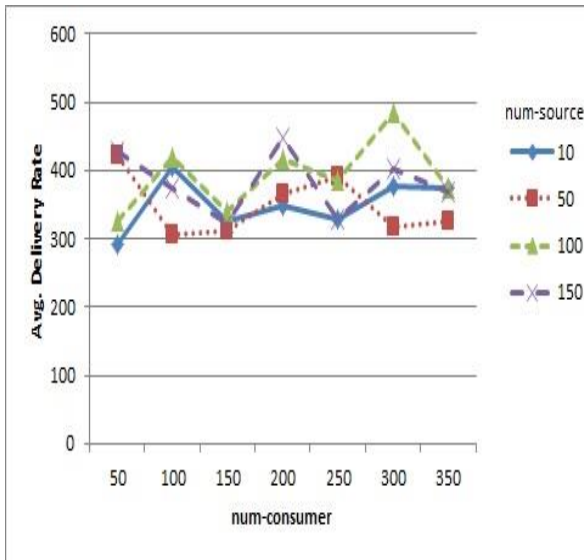
Figure 14. The simulation results of closeness centrality routing on different networks. Each network is composed of 500 nodes. The simulation was carried out on different numbers of sources and consumers (sources: 10, 50, 100, and 150, consumers: 50, 100, 150, 200, 250, 300, and 350). Part (a) shows simulation results on the small-world network. Part (b) shows simulation results on the scale-free network. Part (c) shows simulation results on the random network. Part (d) shows closeness centrality routing on different networks.



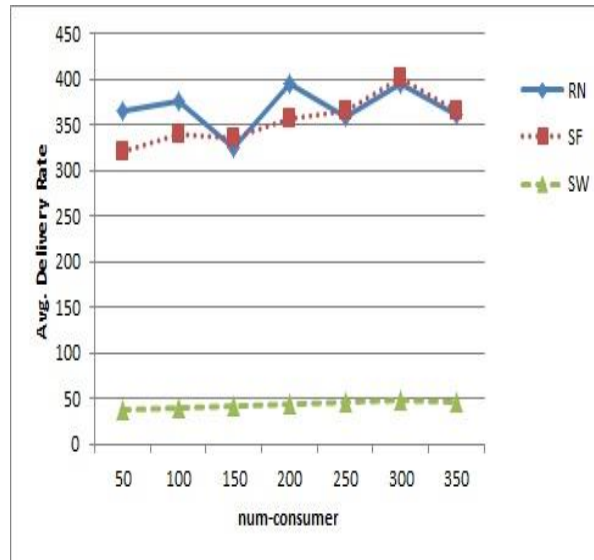
(a) CR-degree on Small-world



(b) CR-degree Scale-free

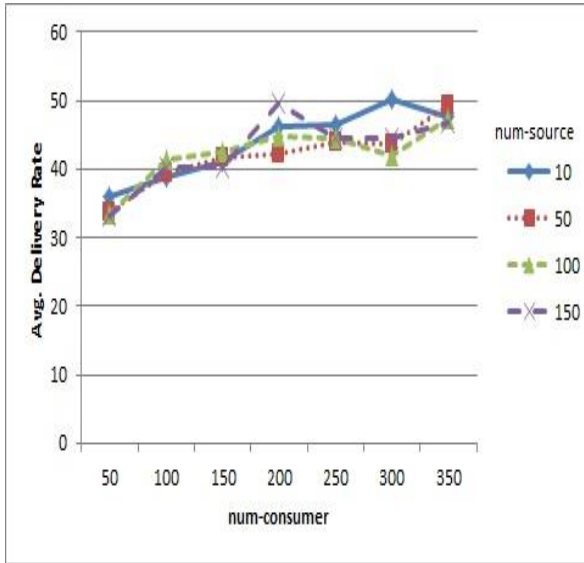


(c) CR-degree on Random network

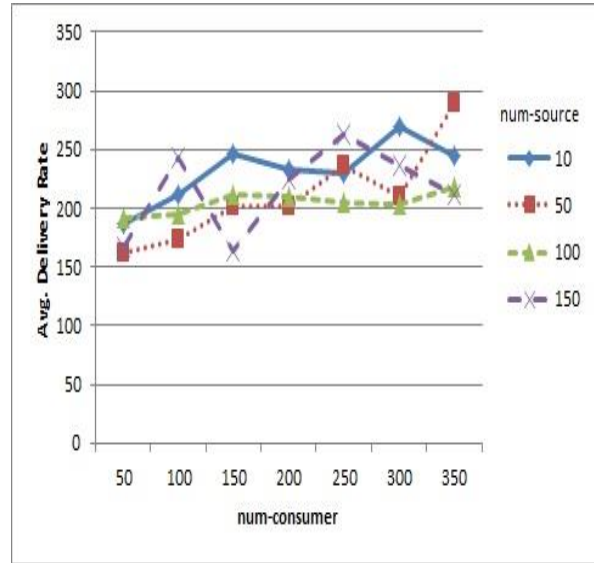


(d) CR-degree on all networks

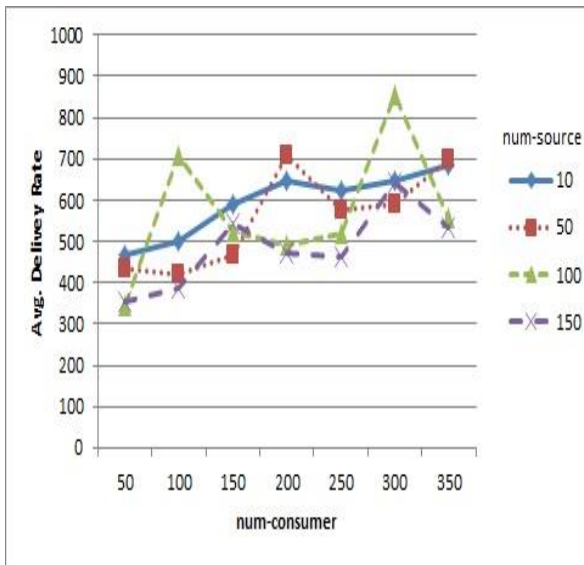
Figure 15. The simulation results of degree centrality routing on different networks. Each network is composed of 500 nodes. The simulation was carried out on different numbers of sources and consumers (sources: 10, 50, 100, and 150, consumers: 50, 100, 150, 200, 250, 300, and 350). Part (a) shows simulation results on the small-world network. Part (b) shows simulation results on the scale-free network. Part (c) shows simulation results on the random network. Part (d) shows degree centrality routing on different networks.



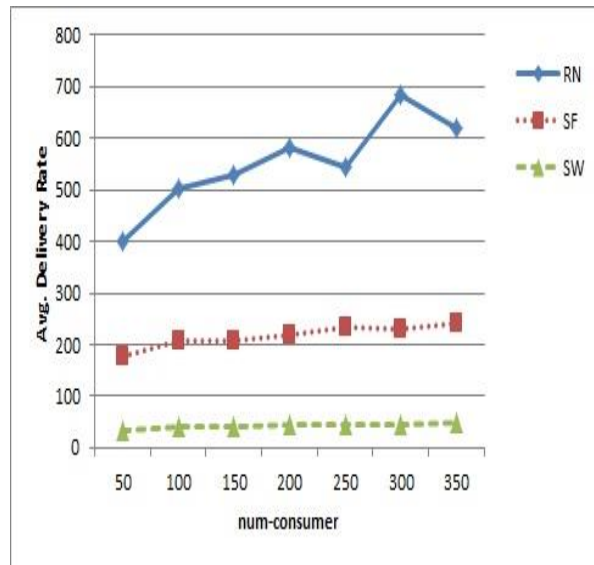
(a) CR-eigenvector on Small-world



(b) CR- eigenvector Scale-free

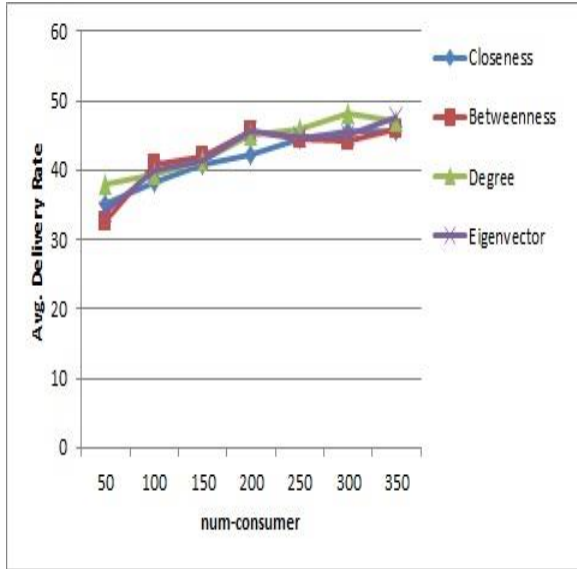


(c) CR- eigenvector on Random network

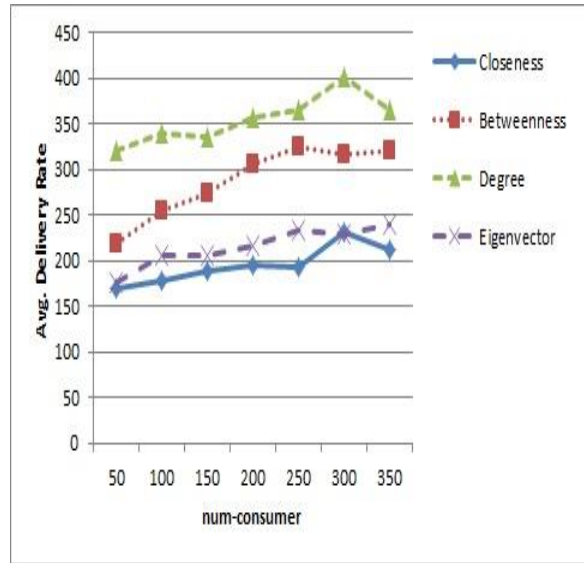


(d) CR- eigenvector on all networks

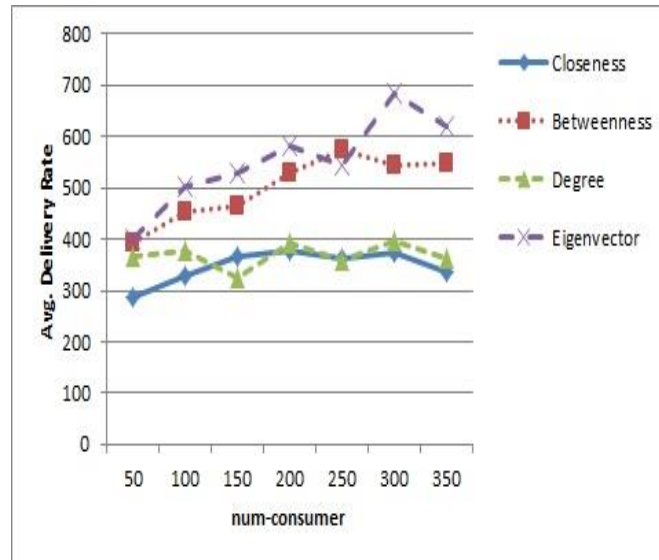
Figure 16. The simulation results of eigenvector centrality routing on different networks. Each network is composed of 500 nodes. The simulation was carried out on different numbers of sources and consumers (sources: 10, 50, 100, and 150, consumers: 50, 100, 150, 200, 250, 300, and 350). Part (a) shows simulation results on the small-world network. Part (b) shows simulation results on the scale-free network. Part (c) shows simulation results on the random network. Part (d) shows eigenvector centrality routing on different networks.



(a) Centrality routing on Small-world network

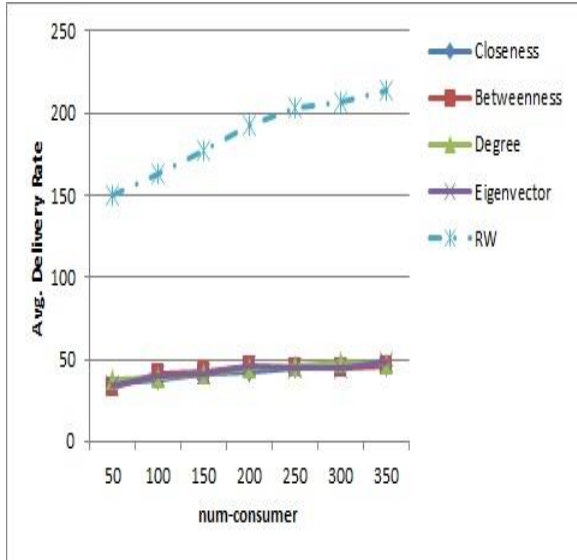


(b) Centrality routing on Scale-free network

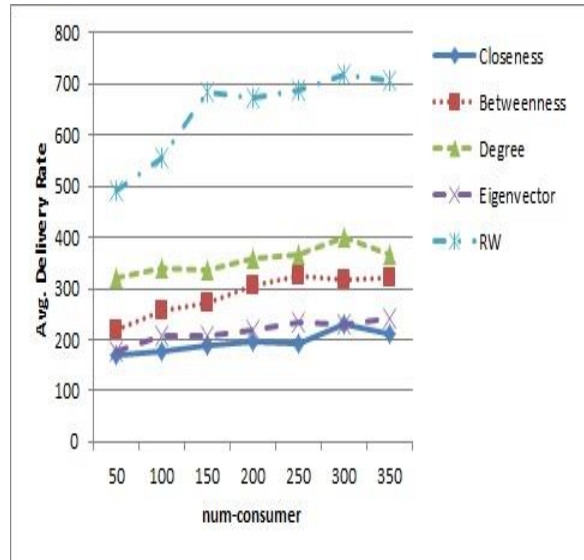


(c) Centrality routing on Random network

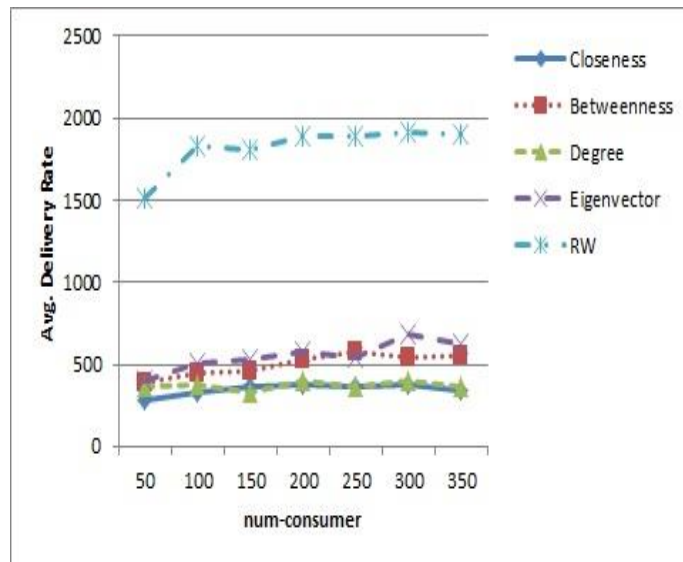
Figure 17. The simulation results of centrality-based routing on complex networks. Results for different types of centrality routing based on Closeness, Betweenness, Eigenvector, and Degree are plotted. The experiments were performed for 50, 100, 150, 200, 250, 300, 350 consumers and 10, 50, 100, 150 sources. Part (a) shows centrality-based routing on the small-world network. Part (b) shows centrality-based routing results on the scale-free network. Part (c) shows centrality-based routing on the random network.



(a) CR vs RW routing on Small-world network



(b) CR vs RW routing on Scale-free network



(c) CR vs RW routing on Random network

Figure 18. Comparison of random-walk and centrality-based routing algorithms on different networks. Experiments were performed for 50,100,150,200, 250, 300, 350 consumers, and 10, 50, 100, 150 sources. Part (a) shows comparative results of random-walk and centrality-based routing on the scale-free network. Part (b) shows comparative results of random-walk and centrality-based routing on the small-world network. Part (c) shows comparative results of random-walk and centrality-based routing on the random network. The simulation results of centrality routing based on closeness, betweenness, eigenvector, and degree are the same for the small-world network.

Table 4. A comparative analysis of previous studies in the smart grid. We analyzed the previous study based on ABM, complex network, specification techniques ODD and DREAM. The comparative study confirms that there is no such specification study for ABM in the smart grid.

Ref.	Objective	ABM	CN	ODD	DREAM
Our study	ABM specification of SG	Yes	Yes	Yes	Yes
[28]	MAS for reactive power management	No	No	No	No
[37]	An adaptive strategy for energy trading	Yes	No	No	No
[30, 31]	Fault location and restoration in the power system network	No	Yes	No	No
[40]	Appliances scheduling in smart home	No	No	No	No
[38]	Communication among large-scale distributed consumers load	No	No	No	No
[39]	Simple power system modeling with consumers and power generators	Yes	No	No	No
[41]	Scheduling of flexible loads	Yes	No	No	No
[42]	Modeling multiple micro grids	Yes	No	No	No
[43]	Battery storage scheduling	Yes	No	No	No
[18]	A conceptual model for smart grid	No	No	Yes	No
[22]	Frequency synchronization in power system	No	Yes	No	No
[23]	Security analysis in power system	No	Yes	No	No

Table 5. Model specification follows ODD (continued on p.582).

Category	Sub-category	Our-model
Overview	Purpose	Modeling a smart grid using agent-based and complex network-based approach.
	Entities	Producers, consumers, walkers
	Entities Process	A hybrid centrality-based routing algorithm for an end to end delivery from producers to consumers
Design concept	Basic principle	A cognitive agent-based computing approach is better for modeling and simulation of the large scale power system.
	Emergence	The computation time of the end to end delivery from producers towards consumers
	Adaptation	Based on connected neighbors
	Objective	To measure how much time is taken while moving from one node to the other
	Sensing	Check the state of the neighbor nodes
	Interaction	Local communication
	Stochasticity	Random process
	Observation	Collect data about the number of consumers, number of producers, the number of nodes visited

Table 5. Model specification follows ODD (continuation from p.581).

Detail	Initialization	Complex network setup
	Input data	External network setup files
	Sub-model	Parameters:
		<ul style="list-style-type: none"> • number of nodes • number of sources • centrality-based routing • average delivery rate calculation

REFERENCES

- [1] Ellabban, O. and Abu-Rub, H.: *Smart grid customers' acceptance and engagement: An overview*. Renewable and Sustainable Energy Reviews **65**, 1285-1298, 2016, <http://dx.doi.org/10.1016/j.rser.2016.06.021>,
- [2] Epstein, J.M.: *Why model?* Journal of Artificial Societies and Social Simulation **11**(4), 12.
- [3] Niazi, M.A.: *Towards a novel unified framework for developing formal, network and validated agent-based simulation models of complex adaptive systems*. Ph.D. Thesis. University of Stirling, 2011,
- [4] Niazi, M.A.: *Complex adaptive systems modeling: a multidisciplinary roadmap*. Complex Adaptive Systems Modeling **1**, No. 1, 2013, <http://dx.doi.org/10.1186/2194-3206-1-1>,
- [5] Chassin, D.P. and Posse, C.: *Evaluating north american electric grid reliability using the Barabasi-Albert network' model*. Physica A: Statistical Mechanics and its Applications **355**(2), 667-677, 2005, <http://dx.doi.org/10.1016/j.physa.2005.02.051>,
- [6] Grimm, V., et al.: *A standard protocol for describing individual-based and agent-based models*. Ecological Modelling **198**(1), 115-126, 2006,
- [7] Niazi, M.A. and Hussain, A.: *Cognitive agent-based computing-I: a unified framework for modeling complex adaptive systems using agent-based & complex network-based methods*. Springer Science & Business Media, 2012,
- [8] Grimm, V., et al.: *The odd protocol: a review and first update*. Ecological Modelling **221**(23), 2760-2768, 2010, <http://dx.doi.org/10.1016/j.ecolmodel.2010.08.019>,
- [9] Watts, D.J. and Strogatz, S.H.: *Collective dynamics of 'small-world' networks*. Nature **393**(6684), 440, 1998, <http://dx.doi.org/10.1038/30918>,
- [10] Barabasi, A.-L.; Albert, R. and Jeong, H.: *Mean-field theory for scale-free random networks*. Physica A: Statistical Mechanics and its Applications **272**(1), 173-187, 1999, [http://dx.doi.org/10.1016/S0378-4371\(99\)00291-5](http://dx.doi.org/10.1016/S0378-4371(99)00291-5),
- [11] Barrenechea, G.; Beferull-Lozano, B. and Vetterli, M.: *Lattice sensor networks: Capacity limits, optimal routing and robustness to failures*. In: *Third International Symposium on Information Processing in Sensor Networks IPSN 2004*. IEEE, pp.186-195, 2004,
- [12] Kleineberg, K.-K. and Helbing, D.: *Collective navigation of complex networks: Participatory greedy routing*. Scientific Reports **7**,

- [13] Lin, B., et al.: *Advanced algorithms for local routing strategy on complex networks*. PLOS ONE **11**(7), e0156756, 2016, <http://dx.doi.org/10.1371/journal.pone.0156756>,
- [14] Rekik, M.; Chtourou, Z.; Mitton, N. and Atieh, A.: *Geographic routing protocol for the deployment of virtual power plant within the smart grid*. Sustainable Cities and Society **25**, 39-48, 2016, <http://dx.doi.org/10.1016/j.scs.2015.12.003>,
- [15] Niazi, M.A.; Siddiqua, A. and Fortino, G.: *Modeling aids spread in social networks*. In: *German Conference on Multiagent System Technologies*. Springer, pp.361-371, 2013,
- [16] Niazi, M.A.: *Emergence of a snake-like structure in mobile distributed agents: an exploratory agent-based modeling approach*. The Scientific World Journal **2014**, No. 168275, 2014, <http://dx.doi.org/10.1155/2014/168275>,
- [17] De Nicola, A. and Missikoff, M.: *A lightweight methodology for rapid ontology engineering*. Communications of the ACM **59**(3), 79-86, 2016, <http://dx.doi.org/10.1145/2818359>,
- [18] Hinker, J., et al.: *A novel conceptual model facilitating the derivation of agent-based models for analyzing socio technical optimality gaps in the energy domain*. Energy **137**, 1219-1230, 2017, <http://dx.doi.org/10.1016/j.energy.2017.06.038>,
- [19] Jiang, Z.-Y.; Ma, J.-F. and Shen, Y.-L.: *Check-in based routing strategy in scale-free networks*. Physica A: Statistical Mechanics and its Applications **468**, 205-211, 2017, <http://dx.doi.org/10.1016/j.physa.2016.11.011>,
- [20] Niazi, M. and Hussain, A.: *Agent-based tools for modeling and simulation of self organization in peer-to-peer, ad hoc, and other complex networks*. IEEE Communications Magazine **47**(3), 166-173, 2009, <http://dx.doi.org/10.1109/MCOM.2009.4804403>,
- [21] Pradittasnee, L.; Camtepe, S. and Tian, Y.-C.: *Efficient route update and maintenance for reliable routing in large-scale sensor networks*. IEEE Transactions on Industrial Informatics **13**(1), 144-156, 2017, <http://dx.doi.org/10.1109/TII.2016.2569523>,
- [22] Wang, C.: *Synchronisation in complex networks with applications to power grids*. Ph.D. Thesis. University of Aberdeen, 2017,
- [23] Jia, Y.; Xu, Z.; Ho, S.-L.; Feng, Z.X. and Lai, L.L.: *Security analysis of smart grids - a complex network perspective*. In: *9th IET International Conference on Advances in Power System Control, Operation and Management (APSCOM 2012)*. IEEE, 2012, <http://dx.doi.org/10.1049/cp.2012.2143>,
- [24] Guan, Z.-H.; Chen, L. and Qian, T.-H.: *Routing in scale-free networks based on expanding betweenness centrality*. Physica A: Statistical mechanics and its applications **390**(6), 1131-1138, 2011, <http://dx.doi.org/10.1016/j.physa.2010.10.002>,
- [25] Fadel, E., et al.: *Spectrum-aware bio-inspired routing in cognitive radio sensor networks for smart grid applications*. Computer Communications **101**, 106-120, 2017, <http://dx.doi.org/10.1016/j.comcom.2016.12.020>,
- [26] Liu, W.y.; Li, X.; Li, J. and Liu, B.: *An efficient probability routing algorithm for scale-free networks*. Chinese Journal of Physics **55**(3), 667-673, 2017, <http://dx.doi.org/10.1016/j.cjph.2017.03.018>,

- [27] Oliveira, E.M.; Ramos, H.S. and Loureiro, A.A.: *Centrality-based routing for wireless sensor networks*.
In: *Wireless Days 2010 IFIP*. IEEE, pp.1-5, 2010,
- [28] Ansari, J.; Gholami, A. and Kazemi, A.: *Multi-agent systems for reactive power control in smart grids*.
International Journal of Electrical Power & Energy Systems **83**, 411-425, 2016,
<http://dx.doi.org/10.1016/j.ijepes.2016.04.010>,
- [29] Zhang, X.; Flueck, A.J. and Nguyen, C.P.: *Agent-based distributed volt/var control with distributed power flow solver in smart grid*.
IEEE Transactions on Smart Grid **7**(2), 600-607, 2016,
<http://dx.doi.org/10.1109/TSG.2015.2423683>,
- [30] Eriksson, M.; Armendariz, M.; Vasilenko, O.O.; Saleem, A. and Nordstrom, L.: *Multiagent-based distribution automation solution for self-healing grids*.
IEEE Transactions on Industrial Electronics **62**(4), 2620-2628, 2015,
<http://dx.doi.org/10.1109/TIE.2014.2387098>,
- [31] Ghorbani, M.J.; Choudhry, M.A. and Feliachi, A.: *A multiagent design for power distribution systems automation*.
IEEE Transactions on Smart Grid **7**(1), 329-339, 2016,
<http://dx.doi.org/10.1109/TSG.2015.2453884>,
- [32] Halinka, A.; Sowa, P.; Szewczyk, M. and Sztandera, L.: *A neural network approach to adaptive protective systems problem in the complex power generating units*.
International Journal of Intelligent Systems **15**(4), 291-302, 2000,
[http://dx.doi.org/10.1002/\(SICD\)1098-111X\(200004\)15:4%3C291::AID-INT2%3E3.0.CO;2-1](http://dx.doi.org/10.1002/(SICD)1098-111X(200004)15:4%3C291::AID-INT2%3E3.0.CO;2-1),
- [33] Weng, G., et al.: *Fault-tolerant location of transient voltage disturbance source for DG integrated smart grid*.
Electric Power Systems Research **144**, 13-22, 2017,
<http://dx.doi.org/10.1016/j.epsr.2016.10.063>,
- [34] Lima, F.P.A.; Lopes, M.L.M.; Lotufo, A.D.P. and Minussi, C.R.: *An artificial immune system with continuous-learning for voltage disturbance diagnosis in electrical distribution systems*.
Expert Systems with Applications **56**, 131-142, 2016,
<http://dx.doi.org/10.1016/j.eswa.2016.03.010>,
- [35] Alonso, M.; Amaris, H. and Alvarez-Ortega, C.: *Integration of renewable energy sources in smart grids by means of evolutionary optimization algorithms*.
Expert Systems with Applications **39**(5), 5513-5522, 2012,
<http://dx.doi.org/10.1016/j.eswa.2011.11.069>,
- [36] dos Santos, J.V.C.; Costa, I.F. and Nogueira, T.: *New genetic algorithms for contingencies selection in the static security analysis of electric power systems*.
Expert Systems with Applications **42**(6), 2849-2856, 2015,
<http://dx.doi.org/10.1016/j.eswa.2014.11.041>,
- [37] Wang, Z. and Wang, L.: *Adaptive negotiation agent for facilitating bi-directional energy trading between smart building and utility grid*.
IEEE Transactions on Smart Grid **4**(2), 702-710, 2013,
<http://dx.doi.org/10.1109/TSG.2013.2237794>,
- [38] Tsai, S.-C.; Tseng, Y.-H. and Chang, T.-H.: *Communication-efficient distributed demand response: A randomized admm approach*.
IEEE Transactions on Smart Grid **8**(3), 1085-1095, 2017,
<http://dx.doi.org/10.1109/TSG.2015.2469669>,
- [39] Kremers, E.; de Durana, J.G. and Barambones, O.: *Multi-agent modeling for the simulation of a simple smart microgrid*.
Energy Conversion and Management **75**, 643-650, 2013,
<http://dx.doi.org/10.1016/j.enconman.2013.07.050>,

- [40] Samadi, P.; Wong, V.W. and Schober, R.: *Load scheduling and power trading in systems with high penetration of renewable energy resources*.
IEEE Transactions on Smart Grid **7**(4), 1802-1812, 2016,
<http://dx.doi.org/10.1109/TSG.2015.2435708>,
- [41] Li, Y., et al.: *A consensus control strategy for dynamic power system look-ahead scheduling*.
Neurocomputing **168**, 1085-1093, 2015,
<http://dx.doi.org/10.1016/j.neucom.2015.05.015>,
- [42] Wang, H.; Huang, T.; Liao, X.; Abu-Rub, H. and Chen, G.: *Reinforcement learning in energy trading game among smart microgrids*.
IEEE Transactions on Industrial Electronics **63**(8), 5109-5119, 2016,
<http://dx.doi.org/10.1109/TIE.2016.2583341>,
- [43] Kuznetsova, E., et al.: *Reinforcement learning for microgrid energy management*.
Energy **59**, 133-146, 2013,
<http://dx.doi.org/10.1016/j.energy.2013.05.060>.