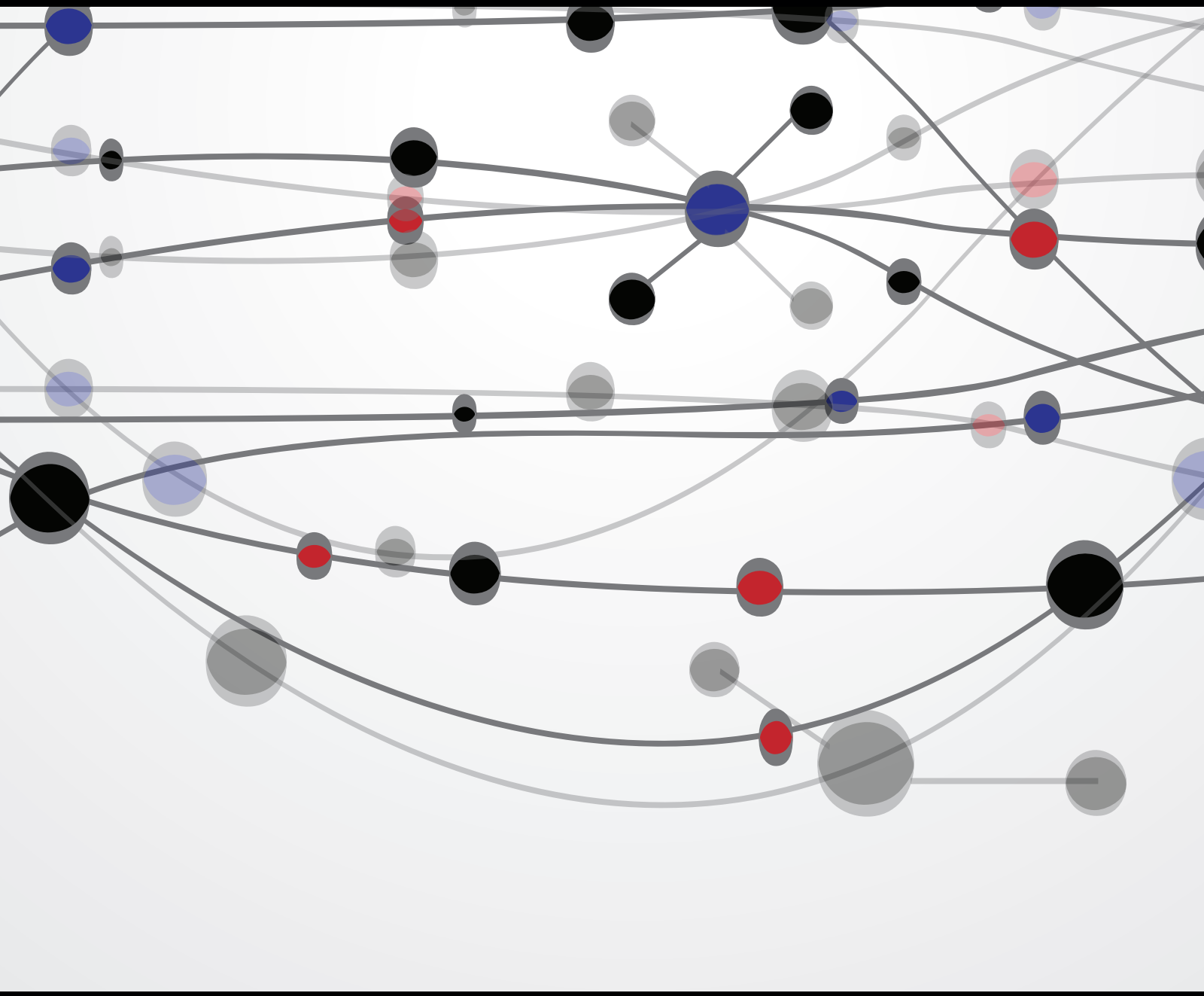
The Scientific World Journal

Soft Computational Approaches for Prediction and Estimation of Software Development

Guest Editors: Xiao-Zhi Gao, Arun Kumar Sangaiah, and Muthu Ramachandran



Soft Computational Approaches for Prediction and Estimation of Software Development

The Scientific World Journal

Soft Computational Approaches for Prediction and Estimation of Software Development

Guest Editors: Xiao-Zhi Gao, Arun Kumar Sangaiah, and Muthu Ramachandran

Contents

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2016, Article ID 3905931, 2 pages
http://dx.doi.org/10.1155/2016/3905931

Editorial

Soft Computational Approaches for Prediction and Estimation of Software Development

Xiao-Zhi Gao,[1] Arun Kumar Sangaiah,[2] and Muthu Ramachandran[3]

[1] *Department of Electrical Engineering and Automation, Aalto University, 00076 Espoo, Finland*
[2] *School of Computing Science and Engineering, VIT University, Vellore 632014, India*
[3] *School of Computing and Creative Technologies, Leeds Beckett University, Leeds LS1 3HE, UK*

Correspondence should be addressed to Xiao-Zhi Gao; xiao.z.gao@gmail.com

Received 22 December 2015; Accepted 22 December 2015

During the past decades, the use of soft computational approaches has been extended to a large variety of software engineering applications. Soft computational methods, such as fuzzy systems, neural networks, evolutionary computation, and probability models including Bayesian network and chaos theory, are currently attractive topics in the extensive software engineering research problems. This special issue provides a platform for the dissemination of knowledge on both the empirical research and applied research in the field of soft computing. The objective of the special issue is to facilitate a forum to a wide spectrum of articles that cover the state-of-the-art techniques and recent results in the research of soft computational approaches. In particular, the special issue focuses on publishing the highly technical articles describing the software development topics: advanced software engineering, computational intelligence, and wireless sensor networks. This special issue has received overwhelming responses from researchers, and it has received a total of 28 high-quality submissions from various countries around the world. All the submitted papers have been evaluated by at least three independent reviewers. However, due to focused scope of the special issue, only a few manuscripts are published. Inevitably, hard editorial decisions had to be made, and some high-quality articles could not even be included. We believe that this special issue presents cohesive information related to the applications of soft computing methods in software development, and it also provides stimulations for future research.

In the paper entitled "An Enhanced PSO-Based Clustering Energy Optimization Algorithm for Wireless Sensor Network," C. Vimalarani et al. propose a novel Enhanced PSO-Based Clustering Energy Optimization (EPSO-CEO) algorithm for the wireless sensor network (WSN), in which the clustering and clustering head selection are done by using the Particle Swarm Optimization (PSO) algorithm with respect to minimizing the power consumption in the WSN.

The results obtained have been compared and evaluated on the basis of energy efficiency.

The paper of Y. Rastegari and F. Shams entitled "Optimal Decomposition of Service Level Objectives into Policy Assertions" presents a method to decompose service level objectives to web service policy assertions. Transformation of Web Service Choreography Description Language (WS-CDL) to Web Service Business Process Execution Language (WSBPEL) has been addressed in some related works, but neither of them considers quality aspects of transformation or run-time adaptation. In this paper, in conformity with web services standards, the authors proposed an optimal decomposition method to make a set of WS-policy assertions. Assertions applied to WSBPEL elements and affect their run-time behaviors. The decomposition method achieves the best outcome for a performance indicator. It also guarantees the lowest adaptation overhead by reducing the number of service reselections. This research considered securities settlement case study to prototype and evaluated the decomposition method. The results show an acceptable threshold between customer satisfaction, the targeted performance indicator through the case study, and adaptation overhead.

G. J. Eanoch and T. S. Sankar present a Neuro-Fuzzy Energy Aware Clustering Scheme (NFEACS) in their paper entitled "Development of Energy Efficient Clustering Protocol in Wireless Sensor Network Using Neuro-Fuzzy

Approach" to form the optimum energy aware clusters. The proposed method consists of two part, fuzzy subsystem and neural network system, which can yield energy efficiency in building up clusters and cluster heads in the WSN. More precisely, it uses neural network that provides effective training sets related to energy and received signal strength of all nodes in order to estimate the expected energy for tentative cluster heads. The sensor nodes with higher energy are trained with the center location of base station for selecting the energy aware cluster heads. Fuzzy IF-THEN mapping rules are also used so as to find the clusters and cluster heads. Experiment results show that the NFEACS performs better than the existing solutions.

In the paper entitled "Software Design Challenges in Time Series Prediction Systems Using Parallel Implementation of Artificial Neural Networks," the authors (N. Manikandan and S. Subha) develop a comprehensive approach to architectural design for financial prediction models. During the past two decades, a large number of artificial neural networks-based learning models have been studied to manipulate with financial data, for example, prediction of the future trends and prices. This paper focuses on the architectural design related issues for performance improvement through vectorising the strengths of multivariate econometric time series models and artificial neural networks. It further provides an adaptive and hybrid method for predicting the exchange rates.

In the paper entitled "Energy-Aware Multipath Routing Scheme Based on Particle Swarm Optimization in Mobile Ad Hoc Networks" written by Y. H. Robinson and M. Rajaram, a novel energy aware multipath routing scheme is proposed, which is based on the Particle Swarm Optimization (EMPSO) that uses Continuous Time Recurrent Neural Network (CTRNN) to deal with optimization problems. The CTRNN has been applied to find the optimal loop-free paths for solving the link disjoint paths in MANET. In other words, it is used as an optimum path selection technique to produce a set of optimal paths between source and destination. In the CTRNN, the PSO method is particularly utilized for training the RNN. The proposed scheme uses the reliability measures, for example, transmission cost, energy factor, and the optimal traffic ratio between source and destination, to increase the routing performance. Therefore, the optimal loop-free paths can be discovered using the PSO to seek better link quality nodes in the route discovery phase suitable for the dynamic topology changing property of a MANET.

Acknowledgments

We would like to express our sincere gratitude to all the contributors who have submitted their high-quality manuscripts and to the experts for their support in providing review comments and suggestions on time.

Xiao-Zhi Gao
Arun Kumar Sangaiah
Muthu Ramachandran

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2016, Article ID 8658760, 11 pages
http://dx.doi.org/10.1155/2016/8658760

Research Article

An Enhanced PSO-Based Clustering Energy Optimization Algorithm for Wireless Sensor Network

C. Vimalarani,[1] R. Subramanian,[2] and S. N. Sivanandam[3]

[1]*Department of Computer Science and Engineering, SNS College of Technology, Coimbatore 641 035, India*
[2]*Department of Electrical and Electronics Engineering, SNS College of Technology, Coimbatore 641 035, India*
[3]*Department of Computer Science and Engineering, Karpagam College of Engineering, Coimbatore 641 032, India*

Correspondence should be addressed to C. Vimalarani; vimalaranis@yahoo.com

Received 5 October 2015; Revised 5 December 2015; Accepted 14 December 2015

Academic Editor: Muthu Ramachandran

Wireless Sensor Network (WSN) is a network which formed with a maximum number of sensor nodes which are positioned in an application environment to monitor the physical entities in a target area, for example, temperature monitoring environment, water level, monitoring pressure, and health care, and various military applications. Mostly sensor nodes are equipped with self-supported battery power through which they can perform adequate operations and communication among neighboring nodes. Maximizing the lifetime of the Wireless Sensor networks, energy conservation measures are essential for improving the performance of WSNs. This paper proposes an Enhanced PSO-Based Clustering Energy Optimization (EPSO-CEO) algorithm for Wireless Sensor Network in which clustering and clustering head selection are done by using Particle Swarm Optimization (PSO) algorithm with respect to minimizing the power consumption in WSN. The performance metrics are evaluated and results are compared with competitive clustering algorithm to validate the reduction in energy consumption.

1. Introduction

Wireless Sensor Network is a network, which can self-organize them with a large number of small sensors. These sensor nodes can perform the packet transmission among themselves within their radio range and also they are organized in a way to sense, observe, and recognize the physical entity of the real world environment. WSN consists of a unlimited number of sensor nodes that can sense their vicinity and communicate either among themselves or to external base transceiver station. The best features of these wireless sensor nodes include small size, low cost, low computation power, multifunctional (can perform sensing, data processing, routing, etc.), and easy communication within short distances. In unattended hostile regions, these devices are deployed in general that make the power source of the sensors difficult to recharge. However, various research works and techniques are carried out for preserving energy in sensor nodes to extend the network lifetime [1]. Prolonged network lifetime, reliable data transfer, energy conservation in sensor nodes, and scalability are the main requirements for WSN applications. Because of the several constraints in the sensor nodes, WSN is having various issues such as coverage area, network lifetime, and scheduling and data aggregation.

The architecture of WSN shows in Figure 1; it comprises wireless sensor nodes in huge number which has been arranged and installed based on the applications and a sink or base station (BT) that is located very near to or within the radio range. The BT transmits the queries to the neighboring sensor nodes which perform the sensing task and return the data to the BT as an answer to the transmitted query.

In WSN nodes utilize disproportionate amount of energy for communication and the required energy in terms of battery power to transmit the packet will differ among the transmissions with respect to the distance between the sender and receiver nodes; therefore multihop communication is recommended. Data transmission using hierarchical routing which increases the lifetime of the sensor network by grouping a number of nodes into clusters. Then a head node is selected for each cluster known as cluster head to collect the data from its members and transmit to the base station with a minimum cost of multihop transmission.

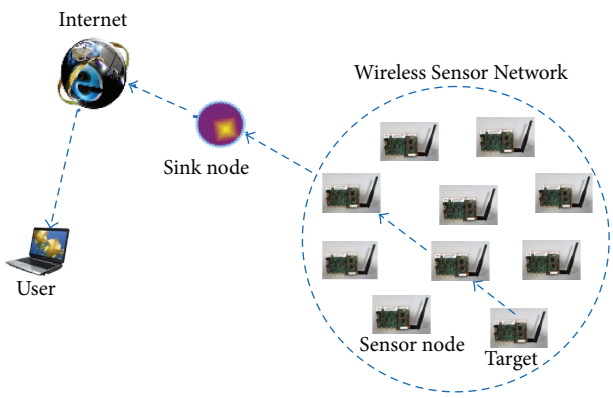


Figure 1: Architecture of Wireless Sensor Network.

Most of the algorithms and protocols [2, 3] tried their best to enhance the performance and throughput of the networks, such algorithms are Low-Energy Adaptive Clustering Hierarchy (LEACH), a Hybrid Energy Distributed Clustering Approach (HEED), and so on. Earlier research papers [4–8] provide different energy efficient techniques that can be used at network layer level by using different routing protocols with energy efficient routing algorithms and reliable communications. The mechanisms described in these algorithms relatively increase the utilization of the power in packet transmission and lengthen the life of the sensor networks. The proposed research work implements the PSO in clustering and for optimal selection of cluster head to enhance the improvement in the residual energy of node by sending a data packet to the cluster head which is located very nearest to the BT.

Particle Swarm Optimization (PSO) is an optimization technique in which natural species social behaviors are considered for the purpose of computation [9]. It is a swarm intelligence technique which is based on population that performs optimization process with the objective of optimizing a fitness function. This approach makes use of a swarm for the purpose of search on every particle and records the fitness value of each particle. Then the particles are linked with their matching velocity. It will help the particle to make a move to a proper location by considering the optimized fitness function's cost [10, 11]. From all the particles of intelligence local, best position optimizes the global best position to identify the cluster head position in order to minimize the overall energy consumption. PSO algorithm has more efficiency and throughput when compared with other mathematical and heuristic approaches.

1.1. Rationale of the Work. To enhance the network lifetime appropriately many routing protocols and cluster-based algorithms are used to fulfill the application requirements in WSN. From existing research methods, optimizing energy dissipation for communication becomes very critical. For maximizing lifetime of the WSN, part of an energy consumption of each sensor node has an important role while communicating among other sensor nodes. This research work focuses on energy conservation in each sensor node by using PSO-based clustering and cluster head selection energy optimization algorithm. The cluster head is selected using PSO, based on the distance from the cluster member node to sink node (BT) and the residual energy in that node. Simulation results show that the motivation of this work improves the life expectancy of the network significantly.

2. Related Work

WSNs have many research challenges and network issues when deploying the sensor nodes to monitor the physical world. Hierarchical routing protocols are appropriate for organizing the nodes to increase the scalability of the WSNs. The traditional clustering algorithm LEACH [12] uses randomized rotation with uniform clustering of local cluster heads to increase the scalability and network performance. The lifetime of the network has extended by utilizing a HEED clustering protocol [13]; this formed the clustering and cluster head selection based on the residual energy of sensor nodes and the cost of communication from source to destination.

Paper [14] proposed Energy Efficient Hierarchical Clustering (EEHC) that increases the lifetime of the sensor network. However, hierarchical clustering made overload in cluster heads and reduces its power sooner than the other nodes. Paper [15] proposed a distribution scheme of cluster heads to reduce energy dissipation by avoiding unnecessary redundancy and compared with existing LEACH it prolongs network lifetime. Paper [16] proposed energy efficient adaptive multipath routing technique to reduce routing overhead and efficiently utilizes the energy availability. In paper [17] competitive clustering (CC) algorithm with sink mobility was proposed to increase residual energy in sensor nodes and improve the network performance. It selects the final head among the competitive candidates based on their remaining energy and competition radio range length. This algorithm forms clusters in small size near the fixed sink node that makes the head node be closer to the BS and consumes lower energy during data gathering between the clusters.

For implementing individual sensor nodes in WSN the better optimization approaches which requires reasonable memory space and resources to produce better results [18]. One of the popular optimization techniques is called Particle Swarm Optimization that has the advantage of solutions with better quality, higher efficiency in computation, easy implementation, and high speed of convergence. PSO-clustering in [14] handled NP-hard optimization problem efficiently by using clustering based on a cluster which lies in a nearby neighborhood, and choosing the sensor node closer to base station becomes header for that particular cluster. PSO-C algorithm considers available energy and distance between the nodes with respect to their cluster heads [19]. The authors in [20] have showed that PSO outperforms both LEACH and LEACH-C in terms of the network span and the overall throughput.

In [21] the author proposed graph theory for routing and PSO for multihop sensor network. For each ith round, the cluster head is selected with the help of a weighted function denoted as $w(i)$, which will be computed in an iterative

manner. Based on the distance taken by the data packet to reach a destination node from the source node and remaining energy, routing of packets is optimized with the fitness function. The simulation results are evaluated with the competitive clustering approach of electing cluster heads and shown as positive results. With the goal of maximizing the network coverage in mobile sensor networks, the author in [22] applied PSO to optimize the sensor deployment strategy. It is executed in a centralized manner which increases the burden of the BS.

For the purpose of reducing the intracluster distance, the authors of paper [23] proposed PSO-based cluster head selection approach to identify the best locality for head nodes with an aim to localize the center of cluster density. Simulation results are matched with the existing LEACH-C and PSO-C and an improvement in network lifetime and saving energy is shown. The recluster construction made network overhead and additional power consumption for communicating clustering information from base station to the sensor nodes.

In this paper, an enhanced clustering algorithm using PSO technique is proposed for energy conservation. The optimal selection of cluster head using PSO reduces the power consumption of each sensor nodes by sending data packets to its cluster head instead of directly forwarding to the base station.

3. Network Energy Model

In this paper, the proposed work simulates the WSN consisting of "n" number of sensor nodes deployed for a temperature monitoring applications using rectangular sensor network. Some assumptions are made regarding the deployment of nodes as [12] given in the following:

(i) All the chosen nodes are considered as static after deployment.

(ii) Two types of nodes are as follows: one is sensor node for sensing temperature monitoring environment and another type of node is sink or base station fixed in the center of the sensor network.

(iii) Sensor nodes are assigned with a distinctive identification (ID) and similar preliminary energy.

(iv) Node is allowed to use transmission power with different levels which are preferred to the remoteness to the target node.

(v) The BT once in a while sends a request message in terms of the packet to the cluster head for getting sampling data from sensors.

(vi) Links are symmetric.

3.1. Energy Model. In WSN an energy model designed in physical layer discussed in [12] used for calculating energy loss in each sensor node while communicating with other sensor nodes. Two channel propagation models used are the free space (d^2 power loss) for the purpose of one-hop or direct transmission and the multipath fading channel (d^4 power loss) for packet transmission via multihop. Thus, the energy exhausted for this kind of transmission of an l-bit packet over distanced d is calculated as

$$E_{\mathrm{TX}}(l,d) = \begin{cases} lE_{\mathrm{elec}} + l\varepsilon_{\mathrm{fs}}d^2, & d < d_0, \\ lE_{\mathrm{elec}} + l\varepsilon_{\mathrm{mp}}d^4, & d \geq d_0, \end{cases} \quad (1)$$

where $\varepsilon_{\mathrm{fs}}$ is free space energy loss, $\varepsilon_{\mathrm{mp}}$ is multipath energy loss, d is distance between source node and destination node, and d_0 is crossover distance:

$$d_0 = \sqrt{\frac{\varepsilon_{\mathrm{fs}}}{\varepsilon_{\mathrm{mp}}}}. \quad (2)$$

The energy spent for the radio to receive this message is

$$E_{\mathrm{RX}}(l) = lE_{\mathrm{elec}}. \quad (3)$$

Thus the transmission power and receiving power energy levels are designed in physical and Mac layer of the WSN [21].

4. Proposed PSO-Based Clustering Algorithm

In this section, we propose an Enhanced PSO-Based Clustering Energy Optimization (EPSO-CEO) algorithm to form clusters and cluster head selection with a combination of centralized and distributed method using static sink node.

4.1. Particle Swarm Optimization (PSO). Particle Swarm Optimization (PSO) is a population-based optimization scheme. The random solutions of the system are initialized with a population and search optimal solutions in each generation [20]. The potential solutions in each generation are called particles. Each particle in PSO keeps the stored record for all its coordinates which are related to obtaining the better solution by following the current best particles.

Fitness function of every particle is executed and the fitness value (best solution) is calculated and stored. The fitness value of the current optimum particle is called "pbest." PSO optimizes the best population value that is obtained so far by any particle in the neighbors and its location is called lbest.

When all the generated populations are considered as topological neighbors by a particular particle, then the best value is chosen among the generated population and that particular best value is the best solution and it is known as gbest. Figure 2 shows the PSO particle movement in a two-dimensional space.

The PSO always try to change the velocity of every particle towards its pbest and lbest. The velocity is determined by random terminologies, which is having randomly generated numbers for velocity towards pbest and lbest localities.

From the large deposit of generated solutions, the best one is selected to resolve the problem. The PSO algorithm always stores and maintains a record of results for three global variables such as target value or condition, gbest, and stopping value.

Every obtained particle of PSO contains the following details.

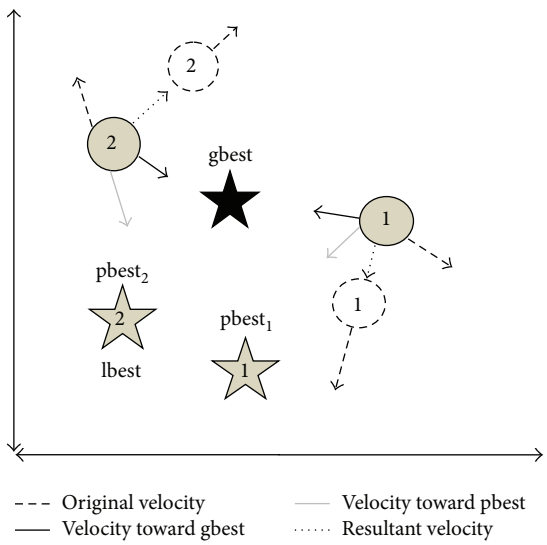


Figure 2: PSO particle movements.

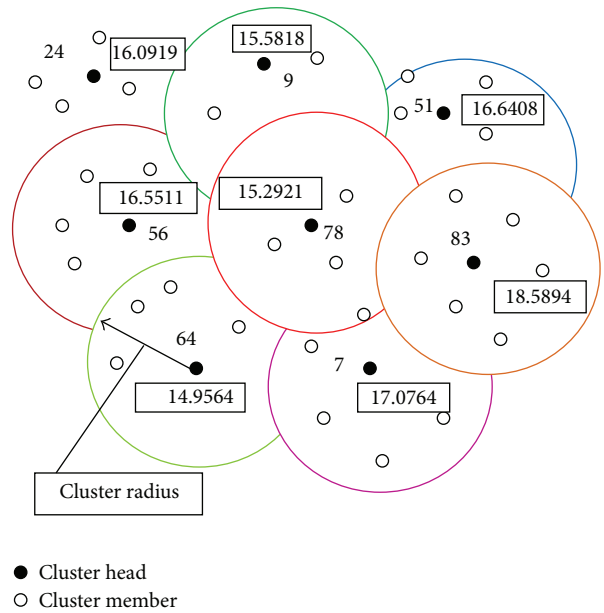


Figure 3: Cluster formation.

(i) A data which can represent a global solution.

(ii) Value for velocity which will indicate the amount of data to be changed.

(iii) lbest value.

4.2. Cluster Formation. The cluster is formed by the base station or sink on the basis of centralized clustering. For clustering base station (sink) broadcasts info collection message to all sensor nodes. Sensor nodes after receiving this message start to send its node information such as node id, location (distance from the base station in X and Y position), energy loss and energy loss ratio (velocity), and current energy to send base station. Then base station initiates the clustering process steps as follows.

Step 1. Conversion of problem into the PSO space in which the PSO particle has two dimensions such as particle position and velocity.

Step 2. Estimation of fitness value using fitness function.

4.3. Fitness Function. Our proposed fitness function for PSO-based clustering is to optimize the average distance and average energy of the member nodes and from the current cluster head and headcount. Figure 3 shows the cluster formation using PSO.

The fitness value is calculated for the particle by using the formula given in the following:

$$\begin{aligned}\text{Fitness value} = \text{Fv} = {} & \alpha_1 \\ & \cdot \frac{\sum_{i=0}^{n} d\,(\text{current node, member } i)}{n} + \alpha_2 \\ & \cdot \frac{\sum_{i=0}^{n} E\,(\text{member } i)}{E\,(\text{current node})} + (1 - \alpha_1 - \alpha_2) \\ & \cdot \frac{1}{\text{No of members covered by current node}},\end{aligned} \tag{4}$$

where α_1 and α_2 are weighing parameters (normalized values) and n denotes number of members covered within the cluster.

Step 3. Generation of new particles from the initial solution. Formation of new particles from the old one is a generation of a new particle.

Step 3.1. Estimation of new velocity: the current velocity of a taken particle is considered to the rate at which the particle's position is changed. New velocity is calculated as follows:

$$\begin{aligned}\text{new velocity} = {} & \omega^{*}\,\text{old velocity} \\ & + w_1\,(\text{local best position} - \text{current best position}) \\ & + w_2\,(\text{global best position} \\ & - \text{current best position}),\end{aligned} \tag{5}$$

where ω is inertia weight and w_1 and w_2 are basic PSO tuning parameters.

Step 3.2. Estimation of new position of the particle is as follows:

$$\text{new position} = \text{old position} + \text{new velocity}. \tag{6}$$

Finally the new particle (new velocity and new position) arrives.

Step 4. Calculation of fitness value for new particles.

Fitness value of the new particles is estimated by using fitness function in Step 2 with new velocity and new position.

Step 5. Fitness value of old particle and new particle is compared and the best one is selected for the next iteration:

If new fitness value > old fitness value

select new particle;

else

old particle is forwarded to next iteration.

Step 6. For every iteration, one best solution is selected as a local best solution.

The particle which has maximum fitness value in the current iteration is selected as lbest solution.

Step 7. The local best solutions from all iterations of the particle in which has maximum among all solutions are selected as a global best solution. The final solutions are decoded into clusters.

The base station forms the cluster using PSO and broadcasts a cluster-announcement message to sensor nodes which contains cluster information as shown in Figure 3. Each sensor node stores this message and initiates round procedure to perform cluster head selection.

4.4. Cluster Head Selection. After clustering, each sensor node maintains "my_cluster_list." It includes current cluster-id, velocity, location, and energy. Then the round procedure is initiated to perform cluster head selection. Cluster head selection by implementing PSO algorithm is shown in Figure 4.

Step 1. The members that are covered by the current node are communicated with each other to select a cluster head which follows as steps mentioned below.

Step 2. Fitness function:

$$\begin{aligned}\text{Fitness value} = \text{Fv} = \alpha_1 &\cdot \frac{\sum_{i=0}^{m} d\,(\text{current node, member } i)}{n}\,\Upsilon + \alpha_2 \\ &\cdot \frac{\sum_{i=0}^{m} E\,(\text{member } i)}{E\,(\text{current node})}\,\Upsilon + (1-\alpha_1-\alpha_2) \\ &\cdot \frac{1}{\text{No of members covered by current node}},\end{aligned} \tag{7}$$

where $\Upsilon = \left\{\begin{matrix}1, & \text{if member } i \text{ is covered by current node} \\ 0, & \text{else}\end{matrix}\right\}$, m is number of members in the current cluster node, α_1 and α_2 are weighing parameters (normalized values), and n denotes the number of members covered within the competition range.

Step 3. Generation of new particles from the initial solution.

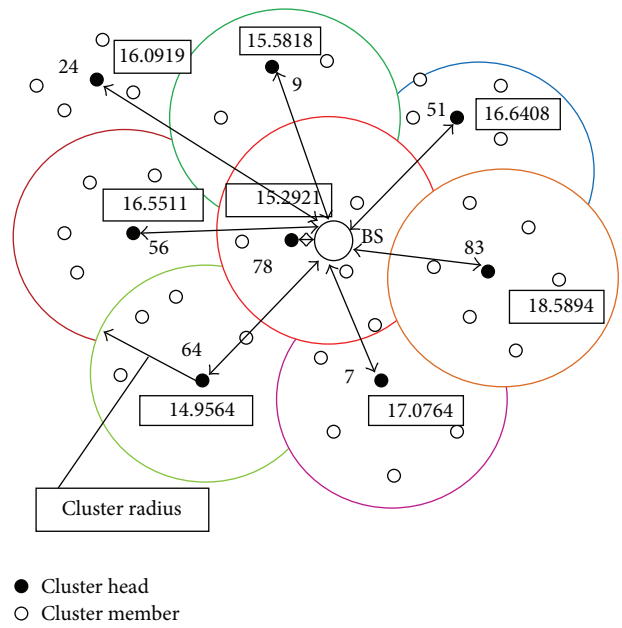


Figure 4: Cluster head selection.

Step 3.1. Estimation of new velocity is as follows:

$$\begin{aligned}\text{new velocity} = \omega^{*}\,&\text{old velocity} \\ &+ w_1\,(\text{local best position} - \text{current best position}) \\ &+ w_2\,(\text{global best position} \\ &- \text{current best position}),\end{aligned} \tag{8}$$

where ω is inertia weight and w_1 and w_2 are the basic PSO tuning parameters.

Step 3.2. Estimation of new position by using new velocity is as follows:

New position = old position + new velocity.

Finally the new particle (new velocity and new position) arrives.

Step 4. Calculate fitness value of new particles.

Fitness value of the new particles is estimated by using fitness function given in Step 2 with new velocity and new position.

Step 5. Fitness value of old particle and the new particle is compared and the best one is selected for the next iteration:

if new fitness value > old fitness value

select new particle;

else

old particle is forwarded to next iteration.

Step 6. For every iteration, one best solution is selected as a local best solution.

The particle which has maximum fitness value in the current iteration is selected as lbest solution.

Step 7. In all iterations one local best solution is found and the particle which has maximum among all local best solutions is selected as a global best solution. Finally, the particle which has a global best solution is chosen as a current cluster head as shown in Figure 4.

4.5. Data Transmission Using Multihop Routing Protocol

4.5.1. Intracluster Multihop Routing Setup. After clustering, routing procedure is invoked during data transmission. Routing consists of two steps; one is route establishment and another one is forwarding sensed data. On demand distance vector routing protocol is used for route establishment [24] between sensor nodes in two occasions: (1) initial route establishment and (2) route unavailability.

In route establishment phase route request message is broadcasted to all nodes with one-hop transmission [20] and unicast route reply message is sent in reverse path to the source node. Once the route is established data transmission with the multihop routing protocol is commenced.

In this work, the multihop communication protocol is used for data transmission [5–8] between the nodes to cluster head (intracluster routing) and cluster head to BT (destination). Data aggregation is done by the head in each cluster for the purpose of saving the residual energy and setting up the threshold value d threshold. In the case of the transmission distance between the cluster head node and the base station is smaller than the threshold value; then the cluster head is committed to transmit the calculated aggregated data to the head with the single hop transmission. Otherwise, cluster head will find next hop with minimum cost neighbor as a relay node [12]. Also, this node will be chosen based on the distance and residual energy. The minimum cost path and highest residual energy node is calculated by using the formula as given in the following:

$$\mathrm{Cost}(j) = \omega^* * \frac{d(s_i, s_j)2 + d(s_j, \mathrm{SN})2}{\max\left(d(s_i, s_j)2 + d(s_j, \mathrm{SN})2\right)} + (1-\omega) * \frac{\max(E(j)) - E(j)}{\max(E(j))}, \quad \omega \in [0,1], \tag{9}$$

where ω is randomized tuning parameter, s_i, s_j are the member node and current head node, and SN denotes sink node.

Relay cluster head node is selected with minimum cost to send data to a destination and intercluster routing is established once the cluster head is selected.

5. Results and Discussion

5.1. Simulation Results. The Network Simulator (NS-2.34) is used for designing the network scenario which executes the PSO algorithm to form clusters and selecting cluster heads in order to reduce energy conservation of sensor nodes. Simulation results are produced, by deploying 100 nodes within a 200 ∗ 200 Sqm area. The sensor nodes are deployed with the task of sensing physical parameters.

The simulation results are evaluated in terms of the following performance metrics.

Total Number of Packets Received. The total number of data packets received by sink node calculated the count value of the total number of data packets transmitted by cluster head node and received by sink node (base station).

Packet Delivery Ratio. The number of packets successfully received with respect to the total number of packets transmitted is known as packet delivery ratio.

Normalized Overhead (NRO). The normalized overhead is defined as the computed ratio between the number of control packets and the number of data packets.

End-to-End Delay. The average time taken to route a data packet from source node to target node is calculated as delay in seconds.

Throughput. It is a measure of a number of packets transmitted per second.

Number of Packet Drop. It is a difference between the total number of data packets sent and the total number of data packets received.

Packet Dropping Ratio. It is a ratio between number of packets dropped and number of packets transmitted.

Network Lifetime. This metric evaluates the time at which the first node failure occurs due to the discharge of battery power charge. The number of active nodes in each round is depicted in

Relative Energy Consumption. It gives the ratio of total amount of energy consumed and the transmission of total packets.

Total Energy Consumption. This calculates the total amount of energy consumed by the nodes to transfer the packets through the simulation.

Average Energy Consumption. It gives the relation of the total energy consumed for total packets received and the total amount of energy consumed by the nodes to transfer the packets.

Total Residual Energy. It is the difference of the initial energy and current energy of each sensor node.

Average Residual Energy. It is the overall residual energy of all sensor nodes over total simulation time.

The network simulation parameters and PSO parameters are listed in the following.

Table 1: Comparison of simulation results of proposed PSO-based clustering algorithm and competitive clustering algorithm.

Number of rounds		50	100	150	200
Number of packets received	CC	103	595	1086	1521
	PSO-based	115	627	1120	1554
Delay (s)	CC	0.04	0.02	0.02	0.02
	PSO-based	0.03	0.02	0.02	0.02
Drops	CC	48	57	58	63
	PSO-based	34	33	30	30
Dropping ratio (%)	CC	0.32	0.09	0.05	0.04
	PSO-based	0.23	0.05	0.03	0.02
PDR (%)	CC	68.21	91.26	94.93	96.02
	PSO-based	77.18	95	97.39	98.11
NRO (%)	CC	16.00	5.83	4.87	4.56
	PSO-based	7.58	3.1	2.62	2.48
Throughput (b/s)	CC	28541.60	37724.50	38789.50	39438.90
	PSO-based	31605.4	39597.8	39977.9	40200.5

5.2. Simulation Parameters. Simulation parameters are as follows:

Number of iterations: 100.

Number of nodes: 100 nodes.

Area (deployment): 200 ∗ 200 Sqm.

Initial energy: 3 joules.

Energy indulgence to run the radio device (E_{elec}): 50 n joule/bit.

Coverage area: 91 metre^2.

MAC type: 802.11.

Antenna model: Omni Direction Antenna.

Propagation model: free space/two-ray ground.

Queue type: priority queue.

Transmission power: 0.02 watts.

Receiving power: 0.01 watts.

Application type: sensing application (temperature).

Connection type: UDP.

Transmission duration: 155 seconds.

Simulation time: 200 sec.

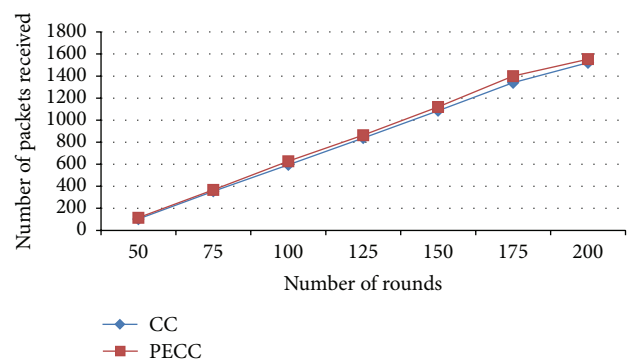


Figure 5: Packets received.

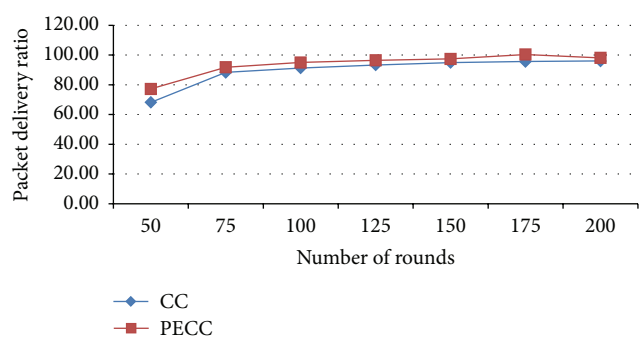


Figure 6: Packet delivery ratio.

5.3. Performance Analysis. In this research work, the following performance metrics are taken for evaluating the WSN to enhance the network efficiency by saving energy. Metrics are evaluated for various rounds from 25, 50, and 75 to 200 and the corresponding outputs depicted as a graph (Figures 5–17). Tables 1 and 2 show the evaluation of the simulation results acquired by using competitive clustering algorithm and PSO-based cluster head selection algorithm. Tables 1 and 2 provide numerical data with a slab of 50 from 50 to 200 rounds.

Simulation results are evaluated with respect to the performance metrics such as number of packets received by sink node, end-to-end delay, packet drop in terms of number of packets, packet dropping ratio, packet delivery ratio, normalized overhead, and throughput of the proposed PSO-based clustering algorithm in contrast with the competitive clustering algorithm that is shown in Table 1.

The other metrics such as overall residual energy, average energy consumption, relative energy consumption, average residual energy, total energy consumption, and lifetime are compared and results are given in Table 2.

From these tables, we observed that the overall network performance of WSN is increased by enhancing the clustering

Table 2: Comparison of simulation results of proposed PSO-based clustering algorithm and competitive clustering algorithm.

Number of rounds		50	100	150	200
Overall residual energy (J)	CC	292.75	288.13	284.73	281.94
	PSO-based	294.52	291.75	289.54	287.5
Average energy consumption (J)	CC	0.04	0.09	0.12	0.15
	PSO-based	0.02	0.05	0.08	0.1
Average residual energy (J)	CC	2.96	2.91	2.88	2.85
	PSO-based	2.97	2.95	2.92	2.9
Relative energy in (l/pkt)	CC	0.04	0.01	0.01	0.01
	PSO-based	0.02	0.01	0.01	0.01
Total energy consumption (J)	CC	4.25	8.87	12.27	15.06
	PSO-based	2.47	5.23	7.44	9.48
Lifetime (s)	CC	13983.80	6697.76	4842.70	3944.72
	PSO-based	24252.1	11393	7998.28	6273.96

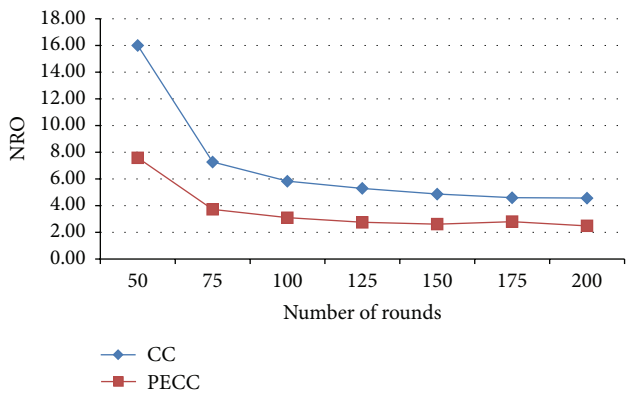


Figure 7: Normalized overhead.

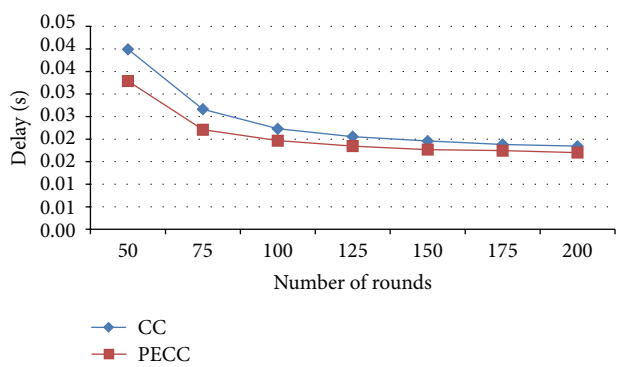


Figure 8: End-to-end delay.

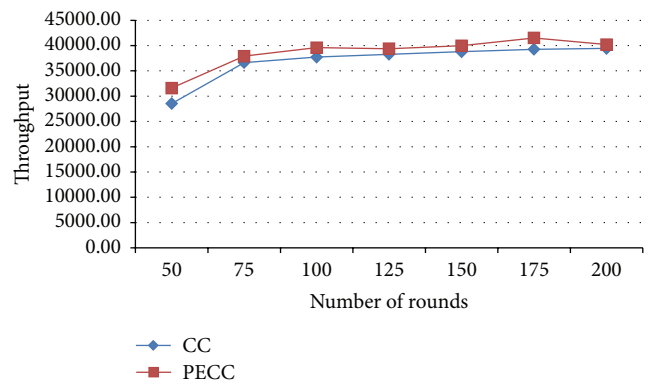


Figure 9: Throughput.

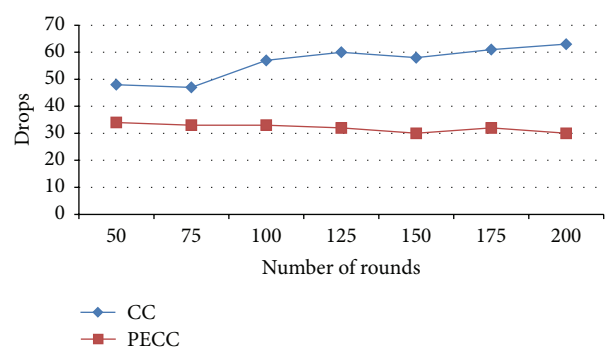


Figure 10: Number of packet drops.

algorithm using PSO-based cluster head selection scheme. Specifically, the average energy consumption and total energy utilization are reduced by 40% and the lifetime has been enhanced by 70%.

The various performance metrics with reference to the graph are discussed as follows.

Figure 5 shows the number of data packets received by the BT with varying rounds from 50, 75, 100, 125, 150, and 175 to 200, respectively. From the results, it can be seen that number of packets received is increased with the optimal selection of cluster head scheme by using proposed PSO-clustering algorithm compared to the competitive clustering algorithm.

Figure 6 shows packet delivery ratio that obtained by using PSO-based energy optimization algorithm is considerably increased compared with the existing system.

Figure 7 depicts the comparison of the normalized overhead using competitive clustering and enhanced PSO clustering algorithm. In proposed system, the number of control packets and normalized overhead of the whole network

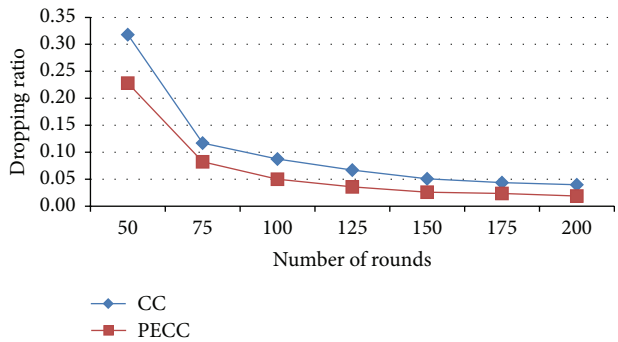


Figure 11: Dropping ratio.

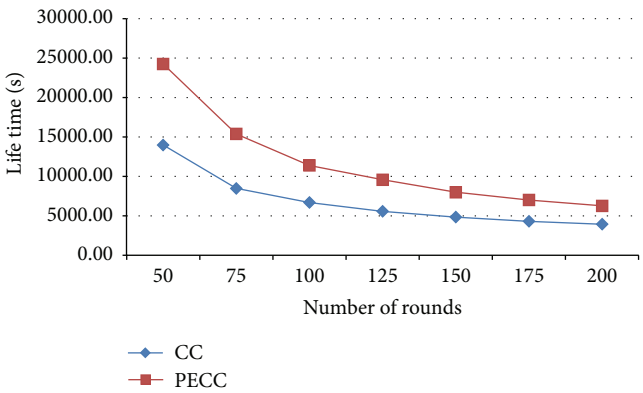


Figure 12: Lifetime.

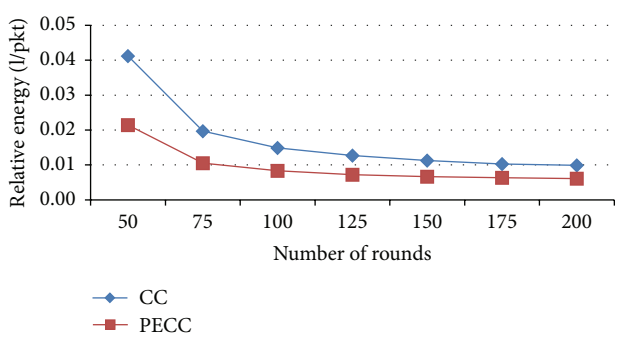


Figure 13: Relative energy.

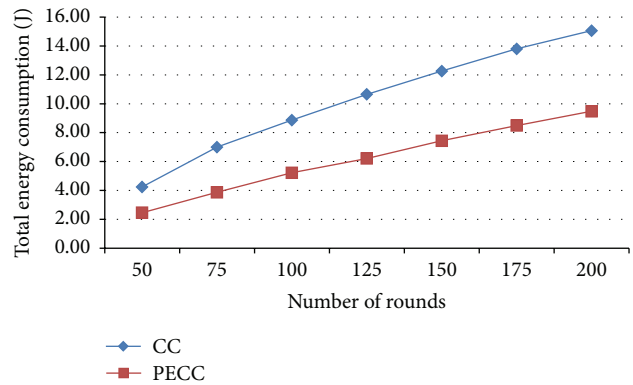


Figure 14: Total energy consumption.

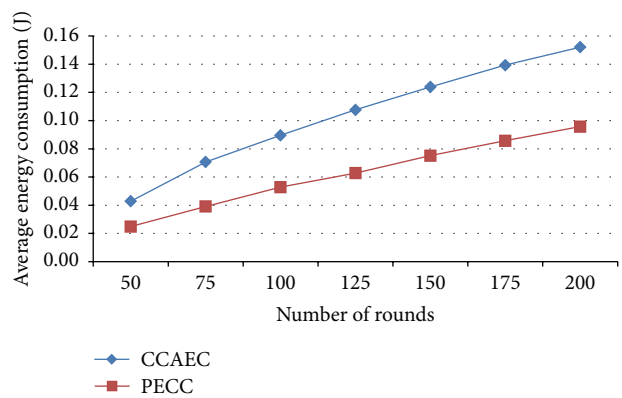


Figure 15: Average energy consumption.

transmission are reduced considerably. Normalized overhead of the overall network control packets is reduced with respect to increased data packets.

Figure 8 shows end-to-end delay of the PSO-based energy algorithm which is minimum compared with a competitive clustering algorithm.

Figure 9 shows an improvement of throughput in varying rounds compared with existing algorithm.

In Figure 10 the number of packets dropped in overall packet transmission of network using PSO clustering and the competitive clustering algorithm is compared. It is observed that the packet drops are reduced considerably.

The dropping ratio is reduced in proposed PSO clustering algorithm relatively in comparison with competitive clustering algorithm which is shown in Figure 11.

Figure 12 which shows the lifetime of the network is increased by consuming less energy of each sensor node with a selection of optimal cluster head using proposed PSO compared with the competitive clustering. The number of active nodes in each round which shows the lifetime of the network is increased.

Figure 13 shows that the relative energy consumption of proposed PSO clustering is less than competitive clustering.

In Figure 14, it is observed that the total energy consumption for various rounds to transmit packets by using proposed PSO-clustering is decreased when compared with competitive clustering considerably.

In Figure 15 it is observed that the energy consumption of all nodes is less considerable by using proposed PSO clustering than competitive clustering.

A considerable improvement is observed in overall residual energy in proposed PSO clustering when compared to the competitive clustering which is shown in Figure 16.

Figure 17 shows that the average residual energy is increased in proposed PSO clustering compared with the competitive clustering.

From the simulation results and performance analysis, we observed that the optimal selection of cluster head using

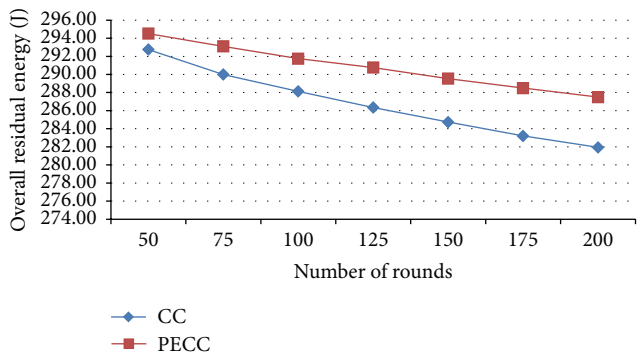


Figure 16: Overall residual energy.

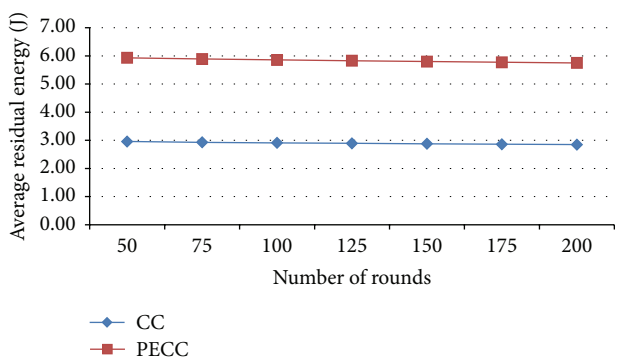


Figure 17: Average residual energy.

PSO clustering algorithm is energy efficient in terms of saving energy and increases the network lifetime to improve the performance of WSNs.

6. Conclusions

The network performance of the WSNs is enhanced by various PSO-based clustering and cluster head selection scheme algorithms in terms of increasing the throughput, packet delivery ratio, residual energy, and number of active nodes. The enhanced PSO algorithm constructs clusters in a centralized manner within a base station and the cluster heads are selected by using PSO in distributed manner. The sensed data from the sensor nodes are aggregated by the head and transmit to the BT directly or using relay node based on the threshold value for which the multihop routing protocol is used. The performance metrics such as throughput, packet delivery ratio, network lifetime, normalized overhead, delay, residual energy, and total energy consumption are evaluated and compared with competitive clustering methodology. The simulation outcome shows that the projected (ECPSO-CEO) scheme gives improved performance in order to minimize the total consumed energy and increase the lifetime of WSN. In future, this work can be extending to improve the network lifetime and data transmission using multiple sink or mobile sink [25] and efficient data collection using data aggregation [6] owing to reduction of the delay in a certain level in the proposed system.

6.1. Contribution to Knowledge. Our research work focuses on energy conservation in each sensor node by using PSO-based clustering and cluster head selection energy optimization algorithm. The cluster head is selected using PSO, based on the distance from the cluster member node to sink node (BT) and the residual energy in that node. To increase the lifetime of the WSN energy conservation measures and energy optimization techniques are enhanced.

Conflict of Interests

The authors proclaim that there is no conflict of interests concerning the publication of this paper.

References

[1] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.

[2] A. Gupta and A. Nayyar, "A comprehensive review of cluster-based energy efficient routing protocols in wireless sensor networks," *International Journal of Research in Computer and Communication Technology*, vol. 3, no. 1, pp. 441–453, 2014.

[3] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[4] M. Younis, I. F. Senturk, K. Akkaya, S. Lee, and F. Senel, "Topology management techniques for tolerating node failures in wireless sensor networks: a survey," *Computer Networks*, vol. 58, pp. 254–283, 2014.

[5] J. Wang, Z. Zhang, F. Xia, W. Yuan, and S. Lee, "An energy efficient stable election-based routing Algorithm for wireless sensor Networks," *Sensors*, vol. 13, no. 11, pp. 14301–14320, 2013.

[6] I. Ha, M. Djuraev, and B. Ahn, "An energy-efficient data collection method for wireless multimedia sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 698452, 8 pages, 2014.

[7] H. R. Dhasian and P. Balasubramanian, "Survey of data aggregation techniques using soft computing in wireless sensor networks," *IET Information Security*, vol. 7, no. 4, pp. 336–342, 2013.

[8] J. Li, X. Jiang, and I.-T. Lu, "Energy balance routing algorithm based on virtual MIMO scheme for wireless sensor networks," *Journal of Sensors*, vol. 2014, Article ID 589249, 8 pages, 2014.

[9] W. Xiaoling, S. Lei, W. Jin, J. Cho, and S. Lee, "Energy-efficient deployment of mobile sensor network by PSO," in *Advanced Web and Network Technologies, and Applications: APWeb 2006 International Workshops: XRA, IWSN, MEGA, and ICSE, Harbin, China, January 16–18, 2006. Proceedings*, vol. 3842 of *Lecture Notes in Computer Science*, pp. 373–382, Springer, Berlin, Germany, 2006.

[10] B. Singh and D. K. Lobiyal, "A novel energy-aware cluster head selection based on particle swarm optimization for wireless sensor networks," *Human-Centric Computing and Information Sciences*, vol. 2, no. 1, article 13, 18 pages, 2012.

[11] S. Thilagavathi and B. G. Geetha, "Energy aware swarm optimization with intercluster search for wireless sensor network," *The Scientific World Journal*, vol. 2015, Article ID 395256, 8 pages, 2015.

[12] J. Wang, X. Yang, T. Ma, M. Wu, and J.-U. Kim, "An energy-efficient competitive clustering algorithm for wireless sensor networks using mobile sink," *International Journal of Grid and Distributed Computing*, vol. 5, no. 4, pp. 79–92, 2012.

[13] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pp. 1–10, Maui, Hawaii, USA, January 2000.

[14] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: a brief survey," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 41, no. 2, pp. 262–267, 2011.

[15] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies (INFOCOM '03)*, vol. 3, pp. 1713–1723, IEEE, San Francisco, Calif, USA, March-April 2003.

[16] T. Kang, J. Yun, and K. Han, "A clustering method for energy efficient routing in wireless sensor networks," in *Proceedings of the 6th International Conference on Electronics, Hardware, Wireless and Optical Communications (WSEAS '07)*, pp. 133–138, Corfu Iceland, Greece, February 2007.

[17] R. Vidhyapriya and P. T. Vanathi, "Energy efficient adaptive multipath routing for wireless sensor networks," *IAENG International Journal of Computer Science*, vol. 34, no. 1, p. 56, 2007.

[18] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, November-December 1995.

[19] S. M. Guru, S. K. Halgamuge, and S. Fernando, "Particle swarm optimizers for cluster formation in wireless sensor networks," in *Proceedings of the IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '05)*, S. K. Halgamuge, Ed., pp. 319–324, Melbourne, Australia, December 2005.

[20] N. M. A. Latiff, C. C. Tsimenidis, and B. S. Sharif, "Energy-aware clustering for wireless sensor networks using particle swarm optimization," in *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '07)*, pp. 1–5, Athens, Greece, September 2007.

[21] N. A. B. A. Aziz, A. W. Mohemmed, and B. S. Daya Sagar, "Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization," in *Proceedings of the International Conference on Intelligent and Advanced Systems (ICIAS '07)*, pp. 961–965, IEEE, Kuala Lumpur, Malaysia, November 2007.

[22] A. Gopakumar and L. Jacob, "Localization in wireless sensor networks using particle swarm optimization," in *Proceedings of the IET International Conference on Wireless, Mobile and Multimedia Networks*, pp. 227–230, IEEE, Beijing, China, January 2008.

[23] M. Saravanan and M. Madheswaran, "A hybrid optimized weighted minimum spanning tree for the shortest intrapath selection in wireless sensor network," *Mathematical Problems in Engineering*, vol. 2014, Article ID 713427, 8 pages, 2014.

[24] C. Zhu, L. Shu, T. Hara, L. Wang, S. Nishio, and L. T. Yang, "A survey on communication and data management issues in mobile sensor networks," *Wireless Communications and Mobile Computing*, vol. 14, no. 1, pp. 19–36, 2014.

[25] T. K. Jain, D. S. Saini, and S. V. Bhooshan, "Lifetime optimization of a multiple sink wireless sensor network through energy balancing," *Journal of Sensors*, vol. 2015, Article ID 921250, 6 pages, 2015.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2016, Article ID 5063261, 8 pages
http://dx.doi.org/10.1155/2016/5063261

Research Article

Development of Energy Efficient Clustering Protocol in Wireless Sensor Network Using Neuro-Fuzzy Approach

E. Golden Julie[1] and S. Tamil Selvi[2]

[1]*Department of Computer Science and Engineering, Regional Office, Anna University, Tirunelveli, Tamil Nadu 627 007, India*
[2]*Department of Electronics and Communication Engineering, National Engineering College, Kovilpatti, Tamil Nadu 628 503, India*

Correspondence should be addressed to E. Golden Julie; goldenjulieauttvl@gmail.com

Received 15 September 2015; Accepted 9 December 2015

Academic Editor: Muthu Ramachandran

Wireless sensor networks (WSNs) consist of sensor nodes with limited processing capability and limited nonrechargeable battery power. Energy consumption in WSN is a significant issue in networks for improving network lifetime. It is essential to develop an energy aware clustering protocol in WSN to reduce energy consumption for increasing network lifetime. In this paper, a neuro-fuzzy energy aware clustering scheme (NFEACS) is proposed to form optimum and energy aware clusters. NFEACS consists of two parts: fuzzy subsystem and neural network system that achieved energy efficiency in forming clusters and cluster heads in WSN. NFEACS used neural network that provides effective training set related to energy and received signal strength of all nodes to estimate the expected energy for tentative cluster heads. Sensor nodes with higher energy are trained with center location of base station to select energy aware cluster heads. Fuzzy rule is used in fuzzy logic part that inputs to form clusters. NFEACS is designed for WSN handling mobility of node. The proposed scheme NFEACS is compared with related clustering schemes, cluster-head election mechanism using fuzzy logic, and energy aware fuzzy unequal clustering. The experiment results show that NFEACS performs better than the other related schemes.

1. Introduction

Wireless sensor networks (WSN) consist of number of sensor nodes with low energy and limited processing capability. WSN is able to sense the physical environment and report about environment data to base station. In critical applications, sensors are randomly deployed in particular region to monitor the environments [1]. Network topology and energy consumption are important issues in WSN for improving network performance in critical applications. Existing clustering schemes are not efficient in terms of energy efficiency. Clustering techniques are playing very important role to maintain the network topology effectively that partition the group of nodes. It maintains the network topology effectively. Artificial neural network (ANN) [2], energy efficient hierarchical unequal clustering [3], continuous time recurrent neural network [4], probabilistic neural network [5], neuro-fuzzy approach [6], and intrusion detection systems based on artificial intelligence technique [7] are classified effectively in WSN. Energy consumption is a great issue in cluster based routing architecture in WSN. It is necessary to develop an energy efficient clustering protocol in WSN for increasing network lifetime.

It is essential to consider energy efficiency for developing cluster based routing schemes in WSN. Many clustering techniques [8] are surveyed in the related works. Energy aware unequal clustering fuzzy (EAUCF) scheme [9] used fuzzy logic to select the cluster heads that considered the residual energy and the distance to the base station of the sensor nodes. Energy efficient cluster formation (EECF) scheme [10] is followed by a three-way message exchange between each sensor and its neighbors. Residual energy and node are considered to select the cluster heads (CHs). The related schemes EAUCF and EECF were not considered a mobility factor to form clusters that do not provide better solution. Balan et al. [11] proposed a system to detect the malicious

behavior of node by intrusion detection system with fuzzy logic technique and identified the attacks such as black hole attack and gray-hole attack. This system also prevents the attacks by using efficient node blocking mechanism.

In this paper, a neuro-fuzzy energy aware clustering scheme (NFEACS) is proposed to form clusters and cluster heads. NFEACS is integrated with neural network system that provides expected energy for tentative cluster heads. Then, fuzzy set approach is used to form clusters and cluster heads that make energy efficiency. NFEACS considered the mobility factor of nodes in WSN. In order to make energy efficiency, the proposed approach uses the mobility factor and residual energy of the sensor nodes to form clusters.

This paper is organized as follows. In Section 2, we present related work related to energy aware clustering schemes. In Section 3, we present proposed NFEACS. In Section 4, we present the simulation results and discussion. Finally, Section 5 presents conclusion and future directions.

2. Related Works

Energy efficient clustering schemes for WSN are widely discussed in the literature. The objective of clustering is the effective topology management and intracluster routing. A fuzzy based clustering protocol is proposed [12] for enhancing energy-efficiency in wireless sensor networks that combines both clustering method and fuzzy logic. Mobility sensor nodes are not considered in scenario. Lee and Cheng [13] proposed a fuzzy-logic-based clustering approach for WSN for energy predication. Many energy aware clustering schemes such as multiobjective fuzzy clustering [14], genetic algorithm [15], fuzzy logic for combining particle swarm optimization and genetic algorithms [16], energy efficient routing protocol with data aggregation [17], and interval type-2 fuzzy logic systems [18] are proposed for addressing energy consumption issues. Nekooei and Manzuri-Shalmani [19] proposed soft computing methods for increasing the network lifetime in WSN.

Low energy adaptive clustering hierarchy (LEACH) [3] was mainly focused on energy metric to form clusters and cluster heads. It also considered load balancing mechanism when cluster heads are elected using rotation basics. Cheng et al. [20] proposed an algorithm that adapts operators of genetic approach with elitism strategy into the iteration in tracking weights. Power efficient gathering in sensor information systems (PEGASIS) [17] was an enhancement of LEACH that forms chains of sensor nodes for transmitting and receiving nodes. It was not suitable for large networks and it wastes much energy to form clusters and cluster heads. Hybrid energy efficient distributed clustering (HEED) [21] used a combined metrics such as residual energy and intracluster communication cost for selecting clusters and cluster heads. Works in [22, 23] proposed fuzzy clustering schemes that use unsupervised learning for remote sensing images when the changes are detected. It was not handling group redundancy in WSN. Sert et al. [14] presented multiobjective fuzzy clustering algorithm to solve hotspots issues in WSN. It consumes more energy in the presence of mobile sensor. Alrajeh et al. [24] presented an artificial neural network based detection of energy exhaustion attacks in WSN that is capable of energy harvesting. They are not considered for mobility of sensors.

3. Proposed Neuro-Fuzzy Energy Aware Clustering Scheme

3.1. Sensor Network Model. The assumptions of that network model are listed:

(i) Mobile sensor nodes are randomly deployed with the same energy level.

(ii) Each sensor node had moved frequently in region.

(iii) Distances between nodes were computed based on the received signal strength.

(iv) The base station needs to be located at the center of the region.

The proposed scheme used Energy aware unequal clustering fuzzy (EAUCF) scheme [9] with some modification. Neural network system is integrated in EAUCF for training network related to energy and received signal strength of all nodes. The radio model [9] is used in simulation for consuming energy. The energy consumed during transmission and reception for a k-bit message to a distance d between transmitter and receiver node is given by

$$\begin{aligned} E_{Tx}(k,d) &= E_{\text{elec}} * k + \varepsilon_{\text{amp}} * k * d, \\ E_{Rx}(k) &= E_{\text{elec}} * k. \end{aligned} \tag{1}$$

3.2. Neural Network System. The supervised learning approach is used to train the network. Single layer perceptron learning rule is used to train the network [2]:

$$W_i(t-1) = W_i(t) + \mu \times (\mathrm{O}_T - \mathrm{O}_A) \times L, \tag{2}$$

where W_i is the weight factor of ith cell, i is the number of input cells, μ is the learning rate, L is the input of that cell, O_A is the output of the network, and O_T is the desired output.

The adjustment of weight is given by

$$W_{\text{new}} = W_{\text{old}} + N \times \mu \times I_p \times (\mathrm{O}_T - \mathrm{O}_A), \tag{3}$$

where

$$\Delta W = N \times \mu \times I \times (\mathrm{O}_T - \mathrm{O}_A). \tag{4}$$

N is the active neuron, and I_p is the input from the previous layer to that active neuron.

Output rate in each layer is expressed as

$$\begin{aligned} &f_1(I_1 \times W_1) \\ &f_2(I_2 \times W_2) \\ &f_3(I_3 \times W_3). \end{aligned} \tag{5}$$

```
    Input x = {I_1, I_2, ..., I_n}
    Initialize I_i = 0 or 1.
Output Co = 0 or 1.
    While {correct output}
        Input x = (I_1, I_2, ..., I_n).
        Po ← y.
    If {Not required output}
        //Change w_i and t
        w_i := w_i + Lr (O-Po) I_i
        t := t − C (O-Po)
    Co ← O.
    Weight change rule:
    If Po = 0, and O = 1, // y is not large enough
        Reduce threshold and increase w_i's.
    if y = 1, O = 0, // output y is too large,
            Increase threshold and reduce w_i
    If Po = 1, O = 1, or Po = 0, O = 0,
        no change in weights or thresholds.
Where Co—correct output, Po—Perceptron output, Lr is positive learning rate
```

Algorithm 1: Perceptron learning rule.

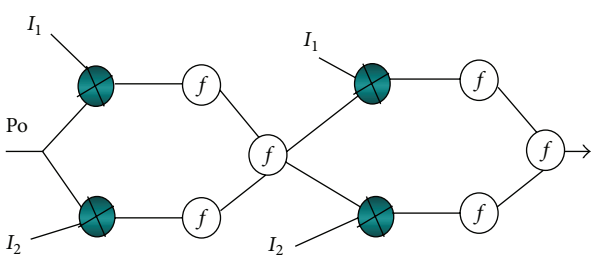


Figure 1: One cell of network layer.

The system has three inputs. The inputs such as the residual energy and received signal strength of sensor nodes are trained by neural network. Input to the system represents energy consumption in network as a continuous variable. Best quality link between node is measured by radio signal strength that is found by

$$L_q = \frac{S_p}{S_{\max}}, \tag{6}$$

where S_p is the signal strength and $S_{\max}$ is the maximum strength available.

Finally, the efficient link quality and also maximum energy sets get trained to select efficient nodes for tentative cluster heads that represent the reliable transmission between source and destination. This input is connected to n first inputs of neural networks. The second input is location of base station that is discrete valued and the last input is feedback outputs. Three inputs make different scenarios in different conditions (Figure 1).

The two upper neurons have fixed weights with linear function. The lower two neurons have trainable weights that were used in sigmoid function of neural network. Neurons can map a set of inputs to a desired output. The sensor network was trained with a scenario-like base station that is located at the centre of the environment. Single layer perceptron is used to train the network to get desired output. Algorithm 1 shows the perceptron learning rule and weight change rule. Network trained and maps outputs with previous learning procedure. The proposed neural network gives energy efficient cluster heads and reliability that is important in a WSN. Result in neural network is the best energy efficiency scenarios for WSN.

3.3. Fuzzy Logic System. The proposed scheme used energy aware unequal clustering fuzzy (EAUCF) algorithm [10] which is integrated with mobility factor. Sensors are randomly deployed in an environment. The sensors are allowed to move freely when they are required. The proposed system chooses a location for a base station at centre of the environment. NFEACS used fuzzy logic approach to select cluster heads. Table 1 shows the fuzzy if-then mapping rule for cluster head chance in NFEACS. NFEACS is on a neural network that provides desired threshold to elect the cluster heads. In this, we considered residual energy and mobility factor weight factors to form efficient cluster heads.

The clustering algorithm is shown in Algorithm 2. It selects a cluster head with low mobility factor and high residual energy.

NFEACS uses residual energy and mobility factor to select the cluster heads. It is derived from fuzzy if-then mapping rule. Fuzzy system used two fuzzy input variables such as mobility factor and residual energy. The fuzzy set linguistic variables such as high, medium, and low are considered for mobility factor. A trapezoidal membership function is chosen for low and high variables. The triangular membership function is selected for medium variable. The linguistic variables like high, medium, and low are selected for residual energy.

Figure 2 shows the fuzzy set input variable for mobility factors that are high, medium, and low. Triangular membership function is chosen for medium linguistic variable. A trapezoidal membership function is chosen for low and

Table 1: Fuzzy if-then mapping rule for cluster head chance.

Mobility factor (speed/sec)	Transmission range (m)	Residual energy (J)	Cluster head chance
Low	High	Low	Medium
Medium	Medium	High	High
Medium	Low	High	Medium
Medium	High	High	Low
Medium	Medium	Medium	Low
Medium	Low	Medium	Low
Medium	High	Medium	High
Medium	Medium	Low	High
Medium	Low	Low	Medium
Medium	High	Low	High
High	Medium	High	High
High	Low	Medium	Medium
High	High	Medium	Medium
High	High	Medium	High
High	Medium	Low	Medium
High	Low	Low	Low

```
T_CH ← output of neural network to become a temporary cluster head
Broadcast changenodestatusMessage as CM
CH ← T_CH
R ← Random value between 0 to 1
   Advertise ElectionMessage
if {R < T}
      Calculate CH_chance using fuzzy if-then mapping rules
      CHMessage (ID, CH_chance, resEnergy, Mobilityfactor)
      if {CH.resEnergy < N.resEnergy}
         CH ← FALSE
         Call QuitElectionProcess
      end if
end if
if {CH = TRUE}
      Broadcast CHMessage with ID
      nodeStatus ← CH
      CH broadcast its nodeStatus with ID
      N Broadcast JoinCMMessage with ID
      C_i = {CM_1, CM_2, ...,CM_n}
else
      Broadcast CHMessages with ID
      CH ← closest CH
      Broadcast JoinCHMessage with ID to the closest CH
end if
```

Algorithm 2: Clustering algorithm of NFEACS protocol.

high variables. Figure 3 illustrates that the residual energy is an input variable for fuzzy set. Trapezoidal membership function is applied for low and high linguistic variables and triangular membership function is chosen for medium linguistic variable. Figure 4 illustrates that the transmission range is also an input variable for fuzzy set. High variable represents a node with maximum transmission range of about more than 10 m selected as cluster head. Membership functions are tested to get the most excellent functions for input variables that give good result.

4. Simulation Results

The proposed scheme NFEACS used neuro-fuzzy system for enhancing the energy efficiency in forming effective clusters. Sensor nodes are varied from 50 to 200 sensor and they were randomly deployed in a 500 m × 500 m simulation area. The transmission range was 20 m. The neuro-fuzzy energy aware clustering scheme (NFEACS) was evaluated and compared with related schemes such as energy aware unequal clustering fuzzy (EAUCF) scheme and energy efficient cluster formation

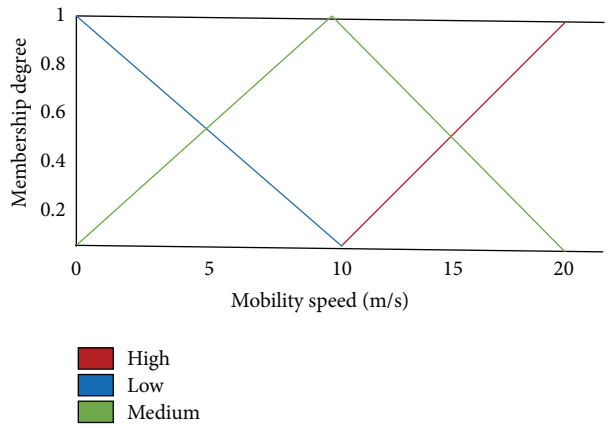


Figure 2: Fuzzy set for fuzzy input variable mobility speed.

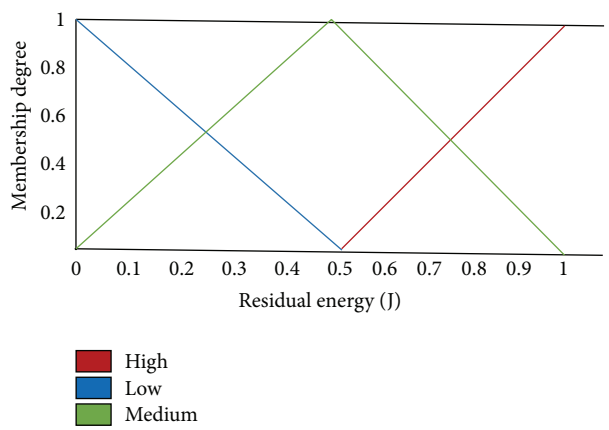


Figure 3: Fuzzy set for fuzzy input variable residual energy.

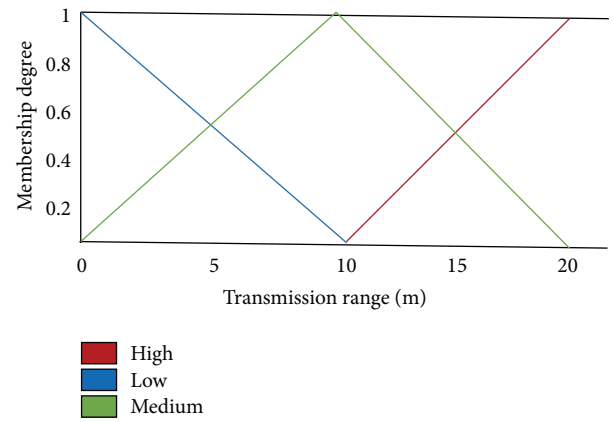


Figure 4: Fuzzy set for fuzzy input variable transmission range.

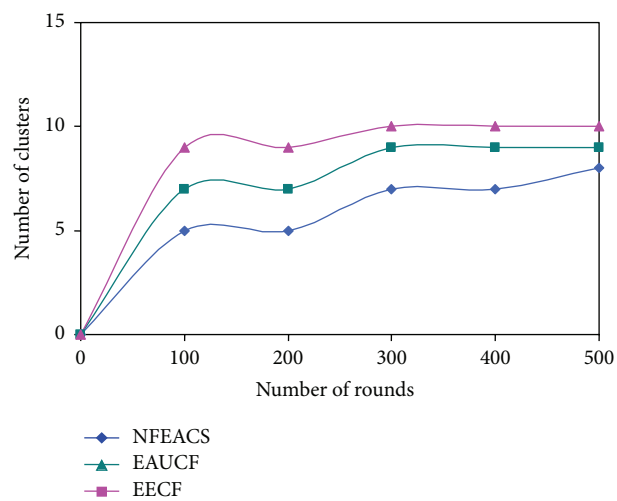


Figure 5: Number of clusters versus number of rounds.

(EECF) scheme. The base station is located at the center of the region of interest. All sensor nodes are initially assigned with 1 J. The NFEACS integrate neural network system with EAUCF scheme instead of probabilistic model for selecting tentative cluster head. Neuro-fuzzy approach was used in NFEACS to get energy aware clusters and cluster heads. In our simulations, we used the same parameter values as [2] such as $\varepsilon_{amp} = 100$ pJ/bit/m^2, $E_{elec} = 50$ nJ/bit, and aggregation ratio 10% and mobility factor is also considered that is not considered in EAUCF.

4.1. Cluster Overhead. Figure 5 shows number of clusters with respect to the number of rounds for proposed scheme and related schemes. EAUCF and EECF generate the higher number of clusters at each round compared to NFEACS. NFEACS generates the minimum clusters in each round. As the number of rounds increases, NFEACS forms minimum clusters and cluster heads when the number of rounds is increased. NFEACS used training set that provides accurate weight value with respect to residual energy and mobility factor for selecting cluster heads, whereas the EAUCF used only fuzzy approach for selecting clusters and cluster heads that provide higher number of clusters than NFEACS. It causes overhead in packet forwarding in the environment.

4.2. Network Lifetime. Figure 6 shows network lifetime with respect to varying number of sensors. It is used to assess the efficiency of the NFEACS in terms of network lifetime. In our simulation, numbers of sensor nodes are varied such as 50, 100, 150, and 200. Performance of NFEACS is evaluated with efficiency of network lifetime with respect to ratio of number of CHs among total number of sensors nodes in the network. Figure 4 notices that NFEACS is very consistent with respect to network lifetime because it used neuro-fuzzy approach to elect the optimal cluster heads compared to other related schemes. Network lifetime is increased when the nodes are increased. NFEACS save 20% of network lifetime compared to related schemes.

4.3. End-to-End Delay. In our simulation, the proposed scheme assumes sensor nodes are mobiles that traverse the network. Figure 7 shows that mobile speed of sensor nodes is varied from 0 to 20 m/s. NFEACS has minimum delay for forwarding packets from destination to base station compared with related schemes because it used neural network approach that provides cluster head reference to select optimal cluster heads in all clusters. NFEACS achieved minimum energy

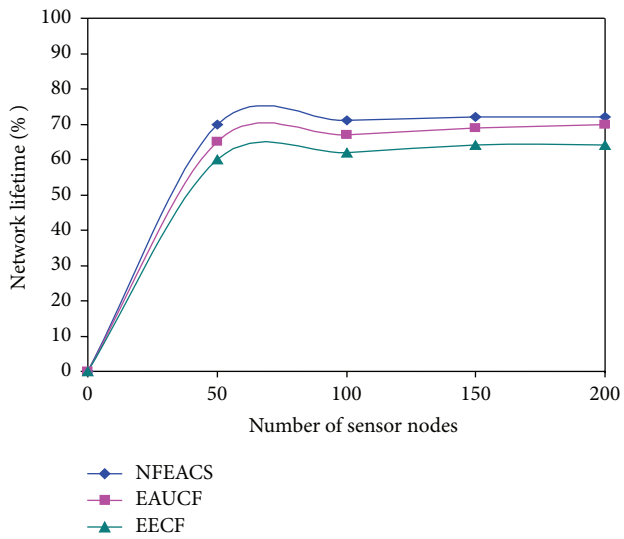


Figure 6: Network lifetime with respect to number of sensors.

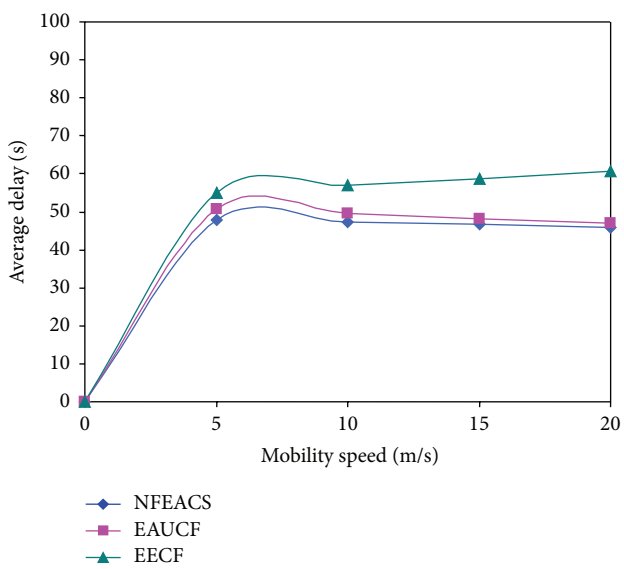


Figure 7: Mobility speed versus delay.

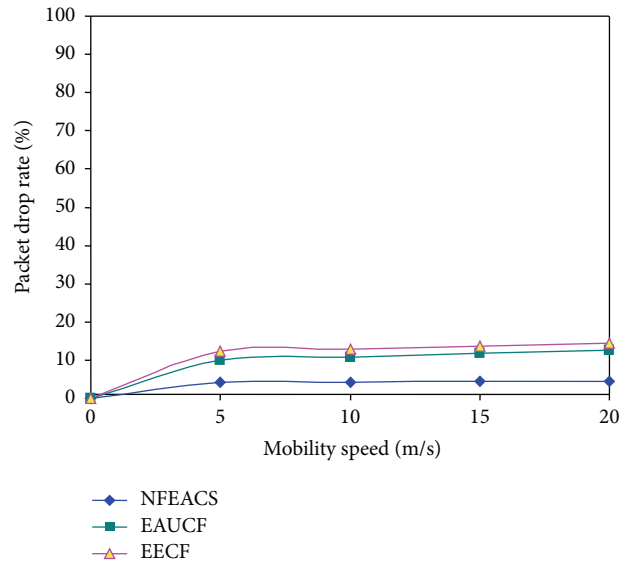


Figure 8: Mobility speed versus packet drop rate.

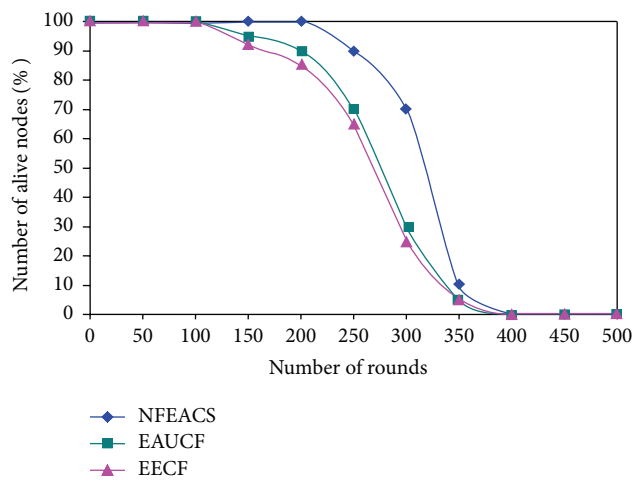


Figure 9: Number of alive nodes in each round.

consumption because of low mobility factor nodes with high residual energy that is elected as cluster heads. It consumes minimum energy that increases network lifetime. The NFEACS has minimum delay even when the sensor nodes are mobile in network. NFEACS also used fuzzy approach to select the low mobility node and neural network to form efficient clusters and cluster heads.

4.4. Packet Drop Rate. Figure 8 shows that the NFEACS has minimum packet drop rate compared with related schemes because the proposed scheme used fuzzy approach to select node with low mobility for selecting cluster heads. The related scheme has more packet drop rate when it considers mobility factor of the sensor nodes. Figure 6 notices that the NFEACS has minimum packet drop rate than related scheme when sensor nodes are mobile in network. NFEACS achieved 37% energy efficiency compared to related schemes.

4.5. Number of Alive Nodes. Figure 9 shows number of alive sensor nodes in each round. It clearly noticed that NFEACS outperforms related schemes because it had alive nodes up to 450 rounds. The related schemes all the sensor nodes are death in 400 rounds that degrade the network lifetime. NFEACS used neuro-fuzzy system that considers three types of metrics such as transmission range, residual energy, and mobility factor to form clusters and cluster heads. The related schemes did not consider the mobility factor to form clusters.

4.6. Signal Strength Ratio. Figure 10 shows the signal strength ratio (SSR) by varying the nodes in the WSN. It shows that the proposed schemes maintained more SSI than related schemes. The SSI of NFEACS had minimum routing packets for transmission compared to related schemes. In related

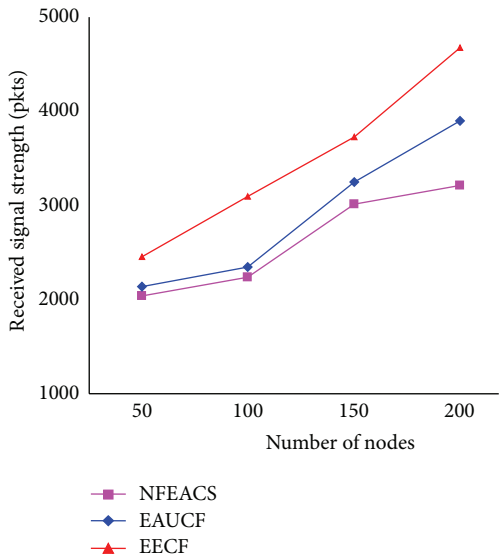


Figure 10: Signal strength ratio.

Table 2: Total remaining energy for three schemes.

Algorithm	Number of alive nodes	Total remaining energy (J)
NFEACS	110	48.54
EAUCF	100	41.19
EECF	80	39.96

schemes, more routing packets are generated for making transmission that causes routing overhead. It shows that the NFEACS provides more link quality between nodes for involving cluster formation and cluster head selection. The related schemes have less SSI because it was not considered as link quality to clusters. Normally the value of SSI gets decreased in the proposed scheme since it sends more numbers of data in a time when being compared to the related schemes.

Table 2 shows the total remaining three schemes. The proposed scheme NFEACS performs better than EAUCF and EECF with respect to mobility and energy metrics. EECF consumes minimum energy and has higher remaining energy about 48.54 J compared to the other two schemes like NFEACS and EAUCF. The proposed scheme NFEACS is more efficient than the other two related schemes about 37% if neuro-fuzzy approach is considered for selecting node with low mobility. EAUCF provides efficient clustering approach than EECF with about 28.5% of energy efficiency because it is considered the fuzzy scheme. The EECF performance is the poorest, because it does not use fuzzy approach during clustering.

5. Conclusion

We proposed a novel scheme named NFEACS that considered neuro-fuzzy approach to form clusters and select energy aware cluster heads. NFEACS analyzed the network lifetime of WSN and the effectiveness of clustering and energy of network is investigated. NFEACS could significantly select energy aware clusters because it used fuzzy system for selecting sensor with low mobility as cluster heads. The feature of neural network can be used to provide cluster head reference that yields optimum results in terms of energy consumption in WSN. We concluded that NFEACS has an increase of 37% in average of energy saving compared to related schemes. Furthermore, the proposed results analyse the energy efficiency of the proposed scheme using the parameters such as energy consumption, packet drop rate, and network lifetime of WSN. A possible future research direction can add weighted metrics like node degree and received signal strength would be used in neural network training set.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] G. Dang and X. Cheng, "Application of wireless sensor network in monitoring system based on Zigbee," in *Proceedings of the IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA '14)*, pp. 181–183, Ottawa, Canada, September 2014.

[2] R. J. Schalkoff, *Artificial Neural Network*, McGraw-Hill, New York, NY, USA, 1997.

[3] B. Baranidharan, S. Srividhya, and B. Santhi, "Energy efficient hierarchical unequal clustering in wireless sensor networks," *Indian Journal of Science and Technology*, vol. 7, no. 3, pp. 301–305, 2014.

[4] Y. H. Robinson and M. Rajaram, "Energy-aware multipath routing scheme based on particle swarm optimization in mobile ad hoc networks," *The Scientific World Journal*, vol. 2015, Article ID 284276, 9 pages, 2015.

[5] J. Jia, C. Liang, J. Cao, and Z. Li, "Application of probabilistic neural network in bacterial identification by biochemical profiles," *Journal of Microbiological Methods*, vol. 94, no. 2, pp. 86–87, 2013.

[6] K. N. Veena and B. P. V. Kumar, "Dynamic clustering for Wireless Sensor Networks: a Neuro-Fuzzy technique approach," in *Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC '10)*, pp. 1–6, Coimbatore, India, December 2010.

[7] N. A. Alrajeh and J. Lloret, "Intrusion detection systems based on artificial intelligence techniques in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 351047, 6 pages, 2013.

[8] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors*, vol. 12, no. 8, pp. 11113–11153, 2012.

[9] H. Bagci and A. Yazici, "An energy aware fuzzy approach to unequal clustering in wireless sensor networks," *Applied Soft Computing*, vol. 13, no. 4, pp. 1741–1749, 2013.

[10] A. Chamam and S. Pierre, "A distributed energy-efficient clustering protocol for wireless sensor networks," *Computers and Electrical Engineering*, vol. 36, no. 2, pp. 303–312, 2010.

[11] E. V. Balan, M. Priyan, C. Gokulnath, and G. U. Devi, "Fuzzy based intrusion detection systems in MANET," *Procedia Computer Science*, vol. 50, pp. 109–114, 2015.

[12] Md. Abdul Alim, Y. Wu, and W. Wang, "A fuzzy based clustering protocol for energy-efficient wireless sensor networks," in *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE '13)*, pp. 2874–2878, Hangzhou, China, March 2013.

[13] J.-S. Lee and W.-L. Cheng, "Fuzzy-logic-based clustering approach for wireless sensor networks using energy predication," *IEEE Sensors Journal*, vol. 12, no. 9, pp. 2891–2897, 2012.

[14] S. A. Sert, H. Bagci, and A. Yazici, "MOFCA: multi-objective fuzzy clustering algorithm for wireless sensor networks," *Applied Soft Computing Journal*, vol. 30, pp. 151–165, 2015.

[15] Y. Yoon and Y.-H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1473–1483, 2013.

[16] F. Valdez, P. Melin, and O. Castillo, "An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2625–2632, 2011.

[17] S. Misra and P. D. Thomasinous, "A simple, least-time, and energy-efficient routing protocol with one-level data aggregation for wireless sensor networks," *Journal of Systems and Software*, vol. 83, no. 5, pp. 852–860, 2010.

[18] H. Shu, Q. Liang, and J. Gao, "Wireless sensor network lifetime analysis using interval type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 2, pp. 416–427, 2008.

[19] S. M. Nekooei and M. T. Manzuri-Shalmani, "Location finding in wireless sensor network based on soft computing methods," in *Proceedings of the International Conference on Control, Automation and Systems Engineering (CASE '11)*, pp. 1–5, Singapore, July 2011.

[20] W. Cheng, H. Shi, X. Yin, and D. Li, "An elitism strategy based genetic algorithm for streaming pattern discovery in wireless sensor networks," *IEEE Communications Letters*, vol. 15, no. 4, pp. 419–421, 2011.

[21] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 1, pp. 366–379, IEEE, Hong Kong, March 2004.

[22] A. Ghosh, N. S. Mishra, and S. Ghosh, "Fuzzy clustering algorithms for unsupervised change detection in remote sensing images," *Information Sciences*, vol. 181, no. 4, pp. 699–715, 2011.

[23] G. E. Tsekouras and J. Tsimikas, "On training RBF neural networks using input-output fuzzy clustering and particle swarm optimization," *Fuzzy Sets and Systems*, vol. 221, pp. 65–89, 2013.

[24] N. A. Alrajeh, S. Khan, J. Lloret, and J. Loo, "Artificial neural network based detection of energy exhaustion attacks in wireless sensor networks capable of energy harvesting," *Ad-Hoc and Sensor Wireless Networks*, vol. 22, no. 1-2, pp. 109–133, 2014.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2016, Article ID 6709352, 10 pages
http://dx.doi.org/10.1155/2016/6709352

Review Article

Software Design Challenges in Time Series Prediction Systems Using Parallel Implementation of Artificial Neural Networks

Narayanan Manikandan and Srinivasan Subha

School of Information Technology & Engineering, VIT University, Vellore, Tamil Nadu 632014, India

Correspondence should be addressed to Narayanan Manikandan; mkyadhav@yahoo.com

Received 1 October 2015; Accepted 29 November 2015

Academic Editor: Muthu Ramachandran

Software development life cycle has been characterized by destructive disconnects between activities like planning, analysis, design, and programming. Particularly software developed with prediction based results is always a big challenge for designers. Time series data forecasting like currency exchange, stock prices, and weather report are some of the areas where an extensive research is going on for the last three decades. In the initial days, the problems with financial analysis and prediction were solved by statistical models and methods. For the last two decades, a large number of Artificial Neural Networks based learning models have been proposed to solve the problems of financial data and get accurate results in prediction of the future trends and prices. This paper addressed some architectural design related issues for performance improvement through vectorising the strengths of multivariate econometric time series models and Artificial Neural Networks. It provides an adaptive approach for predicting exchange rates and it can be called hybrid methodology for predicting exchange rates. This framework is tested for finding the accuracy and performance of parallel algorithms used.

1. Introduction

The universal acceptance of agile methodology provides plenty of evidences in need for rapid adaptation in the current software development life cycle models. However, a number of recent leanings illustrate that a more all-inclusive approach is necessary rather than focusing on continuous integration of software.

Modern time series forecasting involves exchange rates prediction. Many factors that are correlated with each other in a way, namely, economic factors, political factors, and even psychological factors affect the foreign exchange rates by interacting in a complex fashion. Hence, the exchange rates are noisy, chaotic, and nonstationary. But research has shown that nonrandom and predictable behaviour can be emphasized in liquid market areas such as foreign exchange market.

There is a strong dependency between future exchange rates and that of the past. For more than two decades, Box-Jenkins ARIMA was used for time series data forecasting and widely used to benchmark other models. However, it has an assumption that the time series forecasting is linear and stationary by nature. So it results in a need to create a nonlinear model to be used in the prediction exchange rates.

Soft computing techniques are used for predicting currency exchange rates because of the function approximating nature. Parallel Artificial Neural Networks (PANN) in prediction and modelling are used. The multivariate analysis predicts future behaviour and other indicators such as technical, economic, and social indicators are combined along with the time series data in the forecasting process. ANN is more appropriate for the time series forecasting problem because of its nonparametric and adaptive properties. Research has shown that ANNs can map any nonlinear function without any prior assumption about the data. Earlier research in this area has proven that simple technical indicators are enough to obtain useful predictions and significant paper profit, without using any extensive knowledge or data related to the market.

This paper proposes a Heterogeneous Software design mythology in Figure 1 for forecasting currency exchange prices using Artificial Neural Networks. The same design will be validated by implementing of performance improvements through parallel computing. This model uses lagged time

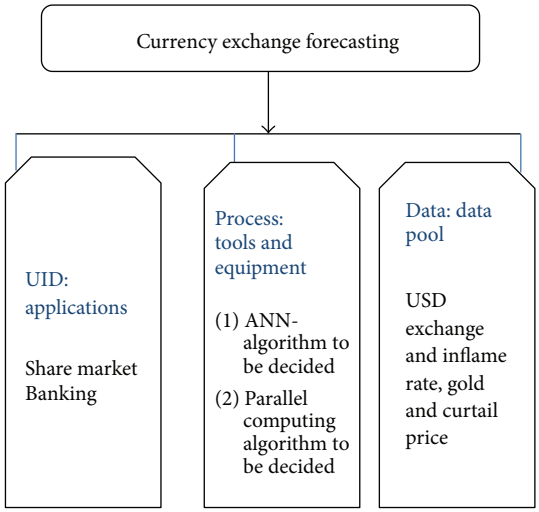


Figure 1: First-level design decisions.

series data along with the technical and economic indicators as inputs and the predicted exchange rates of the desirable currency as output. This work results in forecasting the exchange rates between USD and five major currencies, namely, GBP, JPY, CAD, AUD, and EUR, using Back-Propagation Neural Networks. The Error Back-Propagation Network (EBPN) is trained with parameters like currency exchange rates, gold rates, crude oil rates, and US Inflation Rates for the period of January 2001 to March 2015.

Then the same design is modified without affecting its generality. With the help of Parallel Random Access Machine (PRAM), all processors act in lock-step; that is, the number of processors is not limited and all processors have local memory and one global memory accessible to all processors. Read and write operations are done on global memory and every processor involved in transaction knows its own index. Training and prediction can be done in CPU and GPU (graphics processing unit) based environment to improve the performance.

This work modified the existing scalar algorithms to address design reusability related issues and exploiting naturally parallelisable parts of existing algorithm and injecting brute force methods in each processor to use different initial conditions.

Further organisation of this paper is as follows: Section 2 describes related works and researches in software design, ANN, and parallel processing in the view of performance improvements. Section 3 describes implementation issues, Section 4 shows results and discussion, and Section 5 describes conclusion and future work.

2. Related Works

Major focus is to explain lack of thinking between value and reducing waste. Any product piece or development pace that does not result in adding value is considered as waste. In software design adding and removing appropriate component result in efficiency parameters.

Much research and software development has been going on in forecasting the currency exchange prices in Forex market. ANN based implementation is familiar because of its nonlinear, predictive, and adaptive capabilities. When the amount of training and prediction data increased there is a definite need for parallel processing. In this section some related works done on ANN are presented, Forex market and parallel algorithms that are useful in decision making for effective design.

Useful prediction and significant profit can be made with simple technical indicators without the use of extensive market data or knowledge [1–3].

The author used daily time series data as input and used it to predict the Euro-USD exchange rate using genetic algorithm with Artificial Neural Networks up to three days ahead of the last data available. He used both macroeconomic variables and market data for inputs from which it was learnt that the exchange rate of Euro-USD was conditional. Some of the researchers have shown that a few technical indicators influence the exchange rates strongly. The said indicators are Nasdaq Index, Gold Spot Price, average returns of the government bonds, and crude oil price [4].

After much research using linear and nonlinear models, ANNs are said to perform better than ARIMA model in forecasting foreign exchange rates due to the nonlinear nature of the time series data. Some researchers have used the delayed time series data as input, while few other researchers have used Moving Average (MA) of the time series data as the input [2]. These works [2, 4]impressed us to decide major working components in this design.

The currency exchange rate data in the Forex market was said to be chaotic, random, and noisy in nature. In earlier days, the Random Walk Model and the Efficient Market Hypothesis were the two most widely used models based on fundamental analysis Certainly Forex data was noisy and random. But then, research through statistical tests showed with a significance of 95% that the Forex rates time series are not randomly distributed. To the neural networks since the Moving Average data tends to be a smoothed version of the delayed time series and with much less noise. Also, the MA technique is said to perform well only when the market follows a trend. However, it performs poorly when the index changes direction [5]. This work helps us in analysing feasibility of constructing data model for software.

With the help of above analysis, trade relation, and cost of imports and exports [2], to tackle the market evolution, the input data should be kept consistent. Otherwise, after training, the network is said to degrade in performance. One way to achieve consistency is to periodically replace the past data with recent data [5, 6]. Increasing the number of inputs is said to have not much effect on improving the performance [2].

As [7] specified in his work, the best activation function that can be used in the neural network design for prediction of time series data is a bipolar function $[-1, 1]$ or a binary function $[0, 1]$. Reports [6–8] suggest that performance of the network does not improve when more than 2 hidden layers are used in the network. It has been reported that the presence of more than 2 hidden layers only makes the network more

complex. It also makes the training process difficult and a danger of overfitting is present in such networks having more than 2 hidden layers. All the works in the neural network area have suggested the use of a maximum of 2 to 3 hidden layers as the optimum method to extract the best performance from the network.

To capture the regularities [9] proposed a moving window model that uses a two-layer back-propagation network with a fixed number of inputs modelling a window along the time series in fixed steps to capture the regularities in the underlying data [2]. For a large scale problem back-propagation learns very slowly and convergence is largely dependent on choosing suitable values of learning rate, momentum factor, and step size. Literature review revealed that the testing and validation set should be exactly one-fourth to one-eighth of the training set [5]. Kaastra and Boyd suggests a balanced split 70-15-15 for training, validation, and testing sets. Sigmoid function commonly used transfer function since the time series data is nonlinear in nature [6]. However some researchers have suggested the use of hypertangent function and tangent function too, as transfer functions [4, 8].

Usually the Normalised Mean Square Error is the most widely used metric [2, 3, 7] to measure the efficiency and the correctness of the trained neural network during the testing and validation process. But some researchers [2, 7] have used few other metrics too, in order to compare the performances and get the network to perform the best at any given situation. The error metrics that are used alongside NMSE to measure the correctness of the trained network are MAE (Mean Absolute Error), DS (Directional Symmetry), CU (Correct Uptrend), CD (Correct Downtrend), PMAD (Percentage Mean Absolute Deviation), RMSE (Root Mean Squared Error), MAPE (Mean Absolute Percentage Error), and MAE (Mean Absolute Error). Some have also reported using hit rate as a measure of correctness of a network [3]. Based on performance, it was reported that having small NMSE in validation and testing is more important than having small NMSE for training [3].

Using models based on Artificial Neural Networks, reports show that a correctness of up to 76% has been achieved in the earlier works with the variants of back-propagation algorithm as the learning method [1, 3].

In this work a fully pipelined parallel architecture exploits "mini-batch" training that combines different input cases to compute every set of weight updates to accelerate the power of ANN. Authors implemented this in FPGA; training mechanism is fully implemented in parallelised manner and obtains 100-time performance in running on a Virtex-6 LX760 FPGA [10].

Authors investigated in examining spatial optimization strategies like land allocation and planning will often require multiple data layers and complicated algorithms. It also deals with dynamic processes and the complicated relationships with massive amount of data. Authors developed a parallel geospatial model over the heterogeneous computer architecture of multiple CPU and GPUs. Experiments done with the data sets of California land details resulted in overall computing time for data collected in 50-year simulation which was dropped from 13,000 seconds on a single CPU into 32 seconds using 64 GPU/CPU nodes [11] which gives motivation for us to carry out this work.

In this paper ESR (Evolutionary Swarm Robotics) is an artificial approach for developing collective behaviour of homogenous autonomous robots. Its behaviour is generally controlled by evolving Artificial Neural Networks. However, ESR is unacceptable due to its very high computational cost. Through a detailed study, authors introduced a novel implementation to overcome the computational cost problem. A parallel algorithm for graphics processing unit (GPU) and OpenMP based solutions for multicore CPU. In this approach considerable performance improvement was achieved [12]. To apply parallelisms in currency exchange rate prediction related to Business Intelligence, the following works are considered.

The cloud based Business Intelligence (BI) has been demonstrated with a simulation on OPNET. It is a cloud model with layered OLAP applications with the possibility of applying parallel queries on relational databases. But this work also stated some challenges in taking BI into the cloud because of the restrictions of service providers. So much importance should be given for coordination of elements in architectural design and deploying it to enable the layers of OLAP for better decision making, while designing a BI more significance has to be given for the resource management to avoid bottlenecks. Service providers should plan effectively on available details and implement the same based on infrastructure, platform, and application components to achieve a massively parallel processing system with the support of enhancement framework using all available technologies efficiently [13].

Victor Chang proposed Business Intelligence in cloud based services is very much useful work with respect to predictions in Business Intelligence. Proposed Business Intelligence as a service (BIaas) has Heston model for the investors to take decision before investing and author has used RMSE (Root Mean Squared Error), MA (Moving Average), and EWMA (Exponentially Weighed Moving Average) as calibration. Heterogeneous Software design methodology was used in this implementation [14].

In this work Ramachandran and Chang proposed financial software as a service in cloud environment. The architecture itself sounds good because of its heterogeneity and integrity among the components. Major part of implementations takes Monte Carlo methods, Black Scholes Models, and Variance Gamma process. Here Variance Gamma method is used for outlier removal. Highlights of this paper are MATLAB based implementation and focused towards achieving accuracy and performance in cloud environment [15].

3. System Implementation

3.1. System Description. First some important design decisions should be taken with the following assumptions.

The system is a forecasting model built using neural networks, where the input layer takes the input variables. Both technical and economic variables are taken as input.

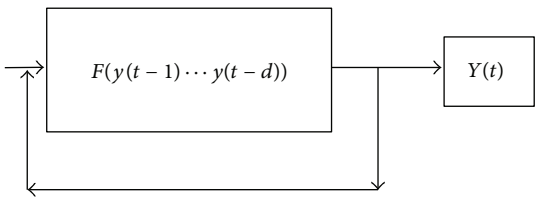


Figure 2: Design for input data generation.

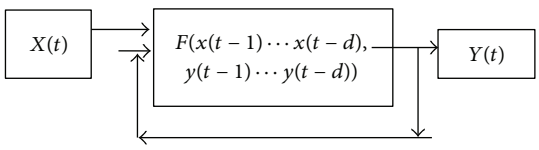


Figure 3: Design of output prediction using NARX.

The hidden layers process the input variables and add value to the system. The hidden layers contain 20 neurons. The output layer yields univariate output. Then a parallel approach for training the algorithm is implemented in the system with the configurations stated in Table 3. This process analyses the impact on change management in software design.

The network is constructed by interconnection of artificial neurons. Different components used in building the parallel forecasting model are as follows.

(i) NAR Network to Predict Input Variables. This section determines prediction details on ask prices of USDJPY, USDAUD, USDCAD, USDCHF, and GBPUSD currencies, respectively, as well as prediction of gold price, crude oil price, CPI, and inflation rate.

The NAR (Nonlinear Autoregressive) network takes the input variable $y(t)$, where the network is trained with the past values of time series to predict its future values. In this example, this approach predicts the variables like ask prices of the five currencies, gold and crude oil price, CPI, and the different inflation rate.

Considering Figure 2, once the network is trained with past values, with the help of parallel algorithms values are predicted from point of prediction and up to the threshold level of expected accuracy.

(ii) NARX to Predict the Output Variable. Figure 3 shows the NARX network that is used for the prediction of the bid price of the desired currency to be forecasted, with the NAR network predictions applied as input to the trained network.

The NARX network takes the input variables as exogenous inputs that all the 10 input variables are taken for input in this network, and the network is trained by assigning the time series output variable that this technique needs to predict as the target variable. Once the network is trained using exogenous input and required output, then, all the trained NAR network predictions of input variables can be supplied to this network as inputs to predict the target output with the presence of delay. This process gives Number of values from the point of prediction to predict the next required number of steps.

Usually the performance degrades with every step of prediction since predicted values are used as feedback to targets instead of original values and it can be solved by using parallel algorithms in appropriate places. Therefore a threshold is set to predict until a particular number of steps to reach good amount of accuracy. NARX network is trained with 25 neurons in hidden layer with the delay of 20.

Number of neurons in this technique is considerably high because of the high processing requirements for data such as a financial time series. Reason for delay of 20 is high correlation because of target values existing in data set (values of one month before the current value used in both input and target variables). This can be reduced by altering design with utilization of parallel resources.

NARX network uses "trainlm" (a predefined algorithm function name, for Levenberg-Marquardt back-propagation algorithm) to train the network. Mean Squared Error (MSE) is used as the performance measure, calculating the mean squared difference between the expected target output and the actual output.

(iii) Time Series Data. MATLAB based implementations are used to promote the reusability for training and prediction process. Excel file is imported using data import Interface and appropriate data for training and Predictions are selected.

Once the required data is selected as in Table 2, it is then converted into either of the following formats, to facilitate the network construction process:

(i) Matrix.

(ii) Cell Array.

(iii) Column Vectors.

In general, Matrix or Cell Array representation is used to store data in the workspace.

The extracted data that is used for training, validating OR predicting, are all stored in the workspace in MATLAB file format with the extension .mat.

This MATLAB file can be accessed when a data is required; this is a simple way of storing data, because all of the data involved is numerical in nature and contains time steps of time series data. There will be less usage if repository based architecture was chosen.

(iv) Architecture Design for Handling Dependencies. The NARX network depends on the other NAR network predictions for its input as it uses those predicted inputs to support the input layer and to predict the target output. Figure 4 represents ancient pipes and filter architecture style which is specialized in designing applications with dependencies.

(v) Choosing Appropriate Algorithm for Low Level Design: Parallel Back-Propagation Learning Approach. This section elaborates the low level design decisions. First Levenberg-Marquardt (LM) algorithm is the most widely used algorithm for optimization. It works well in simple gradient descent and conjugate gradient methods in varieties of problems. LM algorithm gives solutions in the form of *Nonlinear Least Squares Minimization*. The LM algorithm blends speed

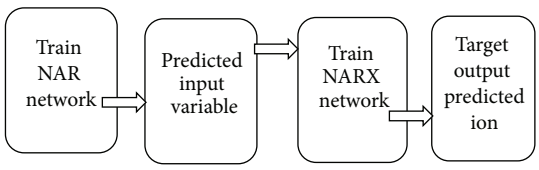


Figure 4: Dependency network.

advantage of Gauss-Newton (GN) algorithm and stability of the steepest descent method. But it is more robust than the GN algorithm, because in many cases it can handle well even if the error surface is much more complex than the quadratic situation.

LM was designed to achieve second-order training speed without calculating Hessian matrix values.

Performance function has sum of squares; Hessian matrix is approximated as

$$H_m = J^T J. \tag{1}$$

Gradient can be computed as

$$G = J^T e, \tag{2}$$

where J is Jacobian matrix containing first derivatives of the network errors based on the weight and biase. e is vector of network errors.

The Jacobian matrix can be computed with less complexity than Hessian matrix through a standard back-propagation. The LM algorithm uses this approximation to the Hessian matrix in the following method called Newton-like update:

$$X_{k+1} = x_k \left[j^T J + \mu I \right]^{-1} J^T e, \tag{3}$$

where scalar μ is zero, in Newton's method for using approximation Hessian matrix.

If μ is large in gradient descent with less steps, then value will be decreased after every successful step.

Algorithm 1 (back-propagation training).

Initial Assumptions. Consider the following:

e_{th}: error threshold.

T_{it}: target iteration.

e: error on output.

o: actual output.

t: target output.

Hid: hidden layer.

P: current pattern.

(1) Divide the training data set D into equal parts D_1, $D_2, \ldots, D_N$, where N is the number of threads in process;

(2) initialize the weights, Desired error threshold e_{th} and Total iterations T_{it}

(3) Channel weight = 0; iteration = 0;

(4) i = size of available patterns;

(5) while i not equal to zero;

(6) Calculate Error in output neuron $e\_i = 0\_i - t\_i$;

(7) Calculate Hid_e_i = e_i ∗ Weight_hid ∗ Derivative_y (P). We_hid. Derivative_y (P); // derivative function for y

(8) Calculate the channel weights W_{ij} for output neurons and hidden layer

(9) Accumulate the newly calculated weight in initial weight for ever iterations.

(10) End while

(11) Update weights in the network based on learning rate

(12) Calculate e_{out}-MSE on training data D,

(13) if $e_{\text{out}} > e_{\max}$ and current iteration > maximum number of iterations increment the iterations and continue.

With the help of Algorithm 1, LM back-propagation algorithms can be parallelised, and batch training approach is used. Since batch training is relatively easy to adapt for multithreaded and multicore CPUs, Steps (4) to (10) can run in threads to perform back-propagation in parallel on different patterns. For each pattern p_i weights and errors can be separately calculated and stored.

Figure 5 represents the high level architecture of training process. Here major responsibility is assigned to synchronizer module that performs the following tasks:

(i) Wait till all threads completes the training process.

(ii) Calculate overall weight for the network.

(iii) Calculate network error for test data.

(iv) Iterate the above process until all patterns are trained sufficiently.

(v) To stop the training process this approach has identified following conditions.

When the maximum number of iterations is reached or maximum amount of time is exceeded, performance is minimized to goal due to performance gradient below min_grad.

In the first step, a time series model based on Artificial Neural Networks generates the estimates of the currency exchange rates and other technical parameters that are used to forecast the exchange price of currency of our choice. In the second step, Error Correction Back-Propagation Neural Network is used to correct the errors of the estimates. The proposed two-step model produces better accuracy in results than the single step models.

4. Results and Discussion

First Heterogeneous Software design merits are given in Table 1.

Ramachandran and Chang and Muntean et al. [15, 16] also used heterogeneous designs for prediction based systems and the results sound good.

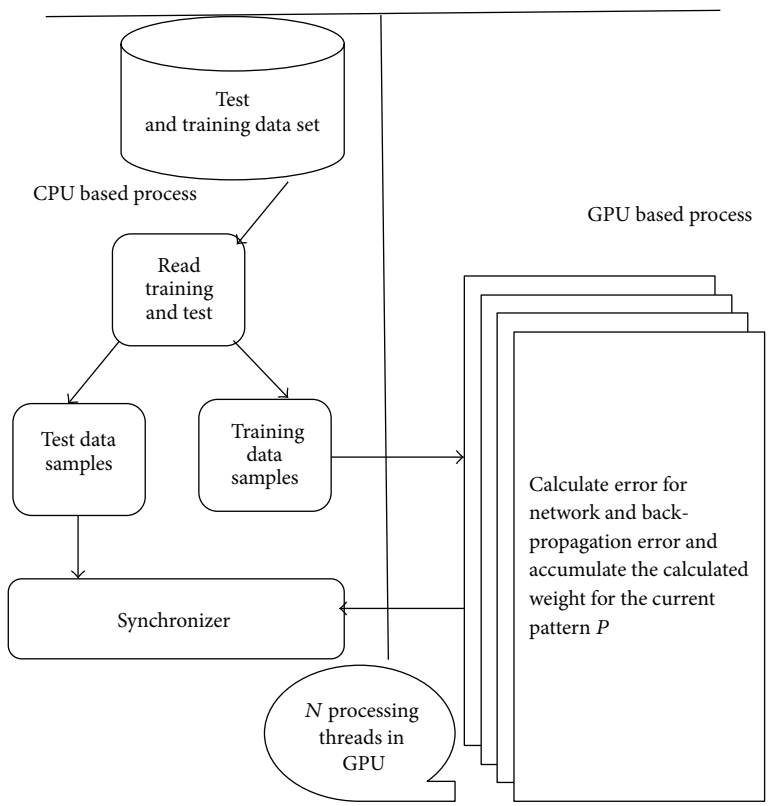


Figure 5: Hybrid architectural designs for training.

Table 1: Quality attributes of software design.

Factors of heterogeneous design	Medium	High
Simplicity		High because well-known components of different technology can be used together
Portability	This design will not fit for implementation in all kinds of platforms	
Modifiability		Actual design and modified design produced the same results
Reliability	Except Japan currency versus USD other predictions are reliable	

Implementation is done with the following steps:

(1) Collected data (as per Section 4.1) is stored in Excel file.

(2) Use NTSTOOL of MATLAB neural network time series tool. With NTSTOOL NAR, NARX, and Levenberg-Marquardt algorithms are implemented.

Table 2: Data collected for analysis.

Process number	Number of daily data	Number of training data	Number of test data
1	48100	33670	14430
2	58200	39500	18700

Table 3: CPU and GPU configurations used.

Processor number	i5-6440HQ
Intel Smart Cache	6 MB
DMI3	8 GT/s
Instruction set	64-bit
Number of cores	4
Number of threads	4
Processor frequency	2.6 GHz
Processor graphics	Intel HD Graphics 530
Graphics frequency	350 MHz
Graphics maximum dynamic frequency	950 MHz

(3) Then implemented Algorithm 1 (parallelised LM algorithm) in MATLAB with parallel processing capability and execution time of each training is noted and compared.

The following sections describe data collections, performance metrics used, and results of implementations.

4.1. Data Collection. The data used in this analysis are the daily foreign exchange rates of five currencies against US Dollar along with prices of crude oil, gold, US Inflation Rate, and CPI from the period of June 1993 to March 2015 made available by Oanda.com. This approach took into consideration the exchange rates of Australian Dollar (AUD), British Pound Sterling (GBP), Canadian Dollar (CAD), Swiss Franc (CHF), and Japanese Yen (JPY).

4.2. Construction Of Performance Metrics. Performance of the above forecasting model is evaluated with the help of three statistical metrics.

Mean Squared Error (MSE) is as follows:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{Y}_i - Y_i\right)^2. \tag{4}$$

Mean Absolute Error (MAE) is as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|f_i - Y_i\right| = \frac{1}{n}\sum_{i=1}^{n}\left|e_i\right|. \tag{5}$$

Sum Squared Error (SSE), also otherwise called Residual Sum of Squares (RSS), is as follows:

$$\mathrm{RSS} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - f\left(x_i\right)\right)^2. \tag{6}$$

Here, MSE and MAE measure the deviation between the actual value and the predicted value. SSE is a measure of discrepancy between the data and the forecasting model.

Higher accuracy of prediction is indicated by the presence of smaller MSE and MAE values.

Performance Measures for Parallel Algorithms. Consider the following:

$$\mathrm{Efficiency} = \frac{\epsilon_{\mathrm{single}}}{N \cdot \epsilon_{\mathrm{parallel}}}, \tag{7}$$

where ϵ is execution time.

*Assumptions.*Single threaded and parallel trainings are initialized with same network weights for training. The experiment is repeated 15 times by changing network configurations. This algorithm is tested with the following Intel based computer.

4.3. Simulation Results. To implement this, at the point of prediction, the predicted value is given as feedback instead of using the original input values; closed loop back-propagation design is chosen. All NAR networks are trained with a maximum of 35 neurons in hidden layer with delay of 20. The maximum performances achieved around 19 to 25 epochs. Results were shown in Table 4.

Table 4: Measurement of prediction performance over 60-day prediction.

Currency	Performance metrics		
	MSE	MAE	SSE
Australian Dollar	$9.74E-05$	$7.64E-03$	$3.45E-03$
British Pound	$2.35E-05$	$3.96E-03$	$8.94E-04$
Canadian Dollar	$3.20E-05$	$4.29E-03$	$1.23E-03$
Swiss Franc	$1.10E-04$	$8.62E-03$	$4.31E-03$
Japanese Yen	$2.21E+03$	$2.83E+01$	$9.65E+04$

The "trainlm" (a predefined algorithm function name, for Levenberg-Marquardt back-propagation algorithm) is used to train the network. And Mean Squared Error (MSE) is used as the performance measure, calculating the mean squared difference between the expected output and the actual output.

A NARX neural network model was trained with 7 technical indicators and 2 economic indicators, a hidden layer, and an output neuron unit to predict the exchange rate. The network uses Levenberg-Marquardt training algorithm which adaptively changes weights during each back-propagation and the training is stopped when the best performance for the given inputs and output is obtained for both training and validation. The number of hidden neuron units was modified between 15 and 20 and the training was terminated at epochs between 60 and 100.

Based on the performance metrics measurements performed on the predicted data, this approach found out that the trained networks gave the best performance predictions with high rates of accuracy for GBP, CHF, AUD, and CAD for 60 days from the point of prediction. But for the JPY currency exchange rate, the prediction accuracy lasted for only 15 days from the point of prediction which is shown in Figure 6(e). This model is created for short term trend forecasting, hence 60-day period of prediction. The model can be extended to 120–150 days with minimum loss of accuracy. The Levenberg-Marquardt algorithm of back-propagation works well for this application when compared to the performance of the previous works in foreign exchange rates prediction.

The performance of such high accuracy is obtained due to the improved technique used for learning, in the Levenberg-Marquardt algorithm, combining the advantages of gradient descent and Gaussian-Newton methods. The MSE, MAE achieved by the trained NARX network is visibly higher than those obtained using other methods in the researches done in the past [2].

The diagrams comparing the actual and the predicted exchange rate of the five currencies are shown in Figures 6(a)–6(e). The plots show that the forecasting follows the actual rates more closely in the case of AUD, GBP, and CAD. For CHF and JPY the prediction is relatively closer to the actual rates.

From the above plots, performance for USD/JPY degrades after 15 days. The other 4 currencies' prediction shows significant accuracy around 90–95% for the first 120 days from the prediction start date and more than 97% accuracy for the first 60 days of the prediction which is

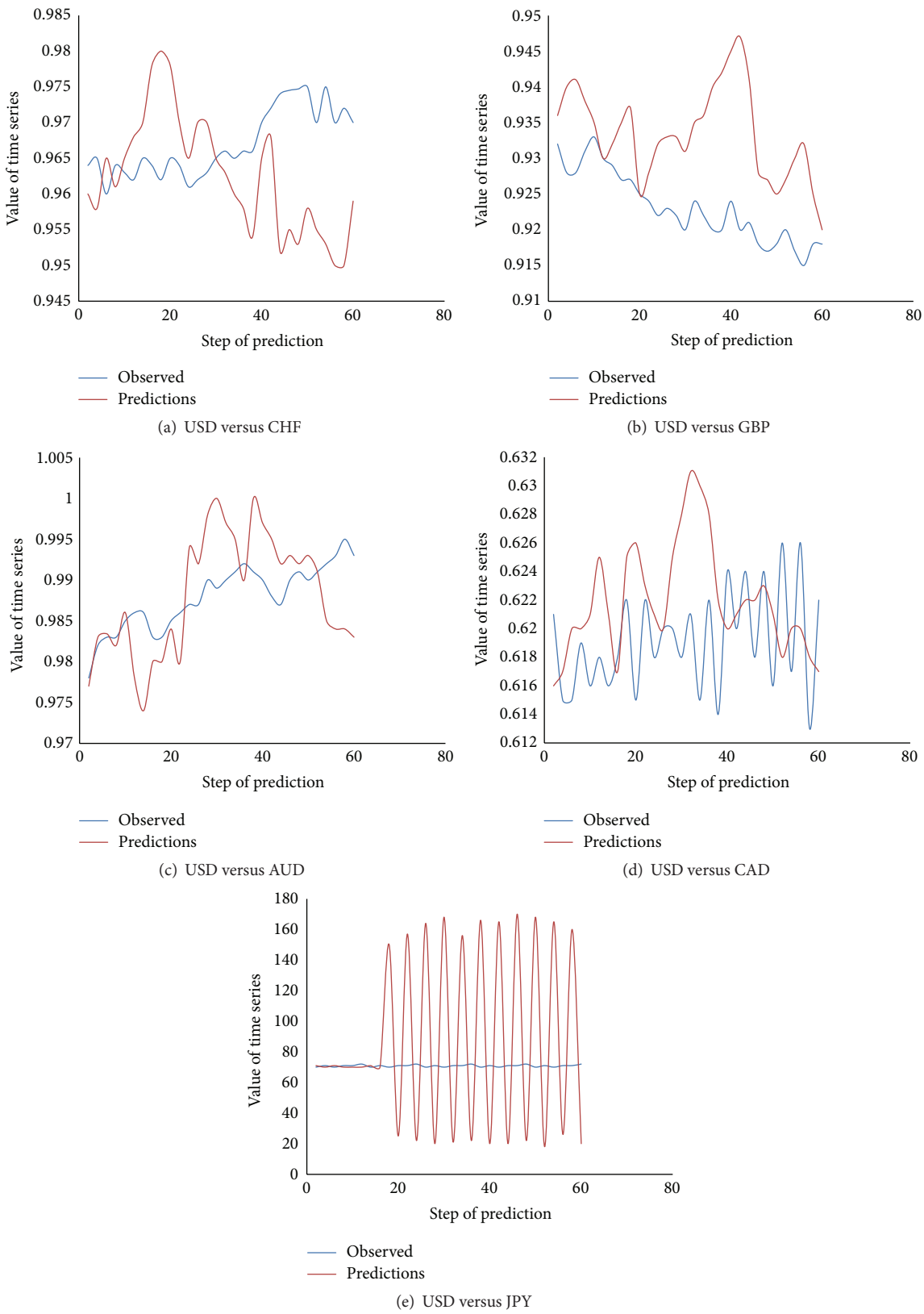


Figure 6: Implementation results of predictions.

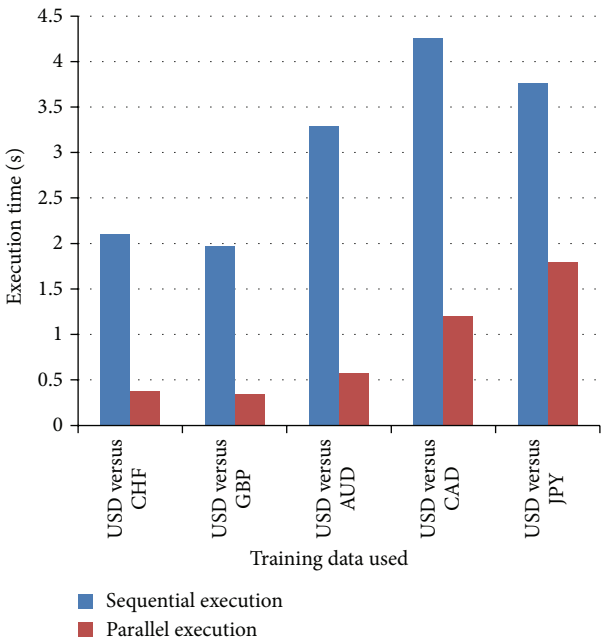


Figure 7: Performance improvement through parallel execution.

significantly higher when compared the other models used in the previous researches [1, 3]. This means that, for improved accuracy, the network has to be retrained every 120 days.

4.4. Parallel Implementation of Same Algorithm. The proposed approach used MATLAB based parallel libraries to make use of library functions. It has implemented changes stated in Figure 5, a hybrid software architecture that addresses the following problems in the sequential execution:

(i) Increasing the execution time of training algorithms. It gives good impact on the overall execution.

(ii) Utilizing maximum processing power of available resources.

(iii) Reducing the cost of implementation through reducing execution time.

Same methodology is executed using parallel training approach. Based on (7), the improvements in execution time are obtained as shown in Figure 7.

5. Conclusion and Future Work

This analysis gave rise to the following conclusions. Heterogeneity based software design is more suitable for soft computing based applications and introducing parallel algorithms at any possible stages will increase the performance. The prediction results are significantly promising for the four currencies GBP, AUD, CAD, and CHF. The prediction performance for Japanese Yen is very poor. Instead of using MSE alone, the proposed approach used two other metrics along with that to measure the performance of the network. But other additional metrics can be used to significantly measure the performance which can be used for comparisons.

The Levenberg-Marquardt back-propagation algorithm that is used in this study to build and train the network has proved to be worthy in combining technical and economic indicators to perform the prediction.

The above observations have confirmed the better performance of Artificial Neural Networks in the forecasting of currency exchange rates.

Further research emphasis will be on using just the technical indicators in the NARX network and obtaining a performance better than the models that were previously used for the purpose of forecasting the currency exchange rates, using the LM algorithm that was used for this study.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. Yao, H.-L. Poh, and T. Jasic, "Foreign exchange rates forecasting with neural networks," in *Proceedings of the International Conference on Neural Information Processing*, pp. 754–759, Hong Kong, China, 1996.

[2] K. Joarder and R. A. Sarker, "Forecasting of currency exchange rates using ANN: A case study," in *Proceedings of the International Conference on Neural Networks and Signal Processing (ICNNSP '03)*, vol. 1, pp. 793–797, IEEE, Nanjing, China, December 2003.

[3] J. Yao and C. L. Tan, "A case study on using neural networks to perform technical forecasting of forex," *Neurocomputing*, vol. 34, no. 1–4, pp. 79–98, 2000.

[4] V. Pacelli, V. Bevilacqua, and M. Azzollini, "An artificial neural network model to forecast exchange rates," *Journal of Intelligent Learning Systems and Applications*, vol. 3, no. 2, pp. 57–69, 2011.

[5] C. Evans, K. Pappas, and F. Xhafa, "Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation," *Mathematical and Computer Modelling*, vol. 58, no. 5-6, pp. 1249–1266, 2013.

[6] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, no. 3, pp. 215–236, 1996.

[7] D. A. Kumar and S. Murugan, "Performance analysis of Indian stock market index using neural network time series model," in *Proceedings of the International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME '13)*, pp. 72–78, IEEE, Salem, India, February 2013.

[8] W.-S. Gan and K.-H. Ng, "Multivariate FOREX forecasting using artificial neural networks," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 2, pp. 1018–1022, December 1995.

[9] A. N. Refenes, M. Azema-Barac, and S. A. Karoussos, "Currency exchange rate forecasting by error backpropagation," in *Proceedings of the 25th Hawaii International Conference on System Science*, vol. 4, pp. 504–515, Kaual, Hawaii, USA, January 1992.

[10] Z. Y. Wu, "Portable GPU-based artificial neural networks for data-driven modeling," in *International Conference on Hydroinformatics. Informatics and the Environment: Data and Model Integration in a Heterogeneous Hydro World (HIC '15)*, New York, NY, USA, 2015.

[11] Q. Guan, X. Shi, M. Huang, and C. Lai, "A hybrid parallel cellular automata model for urban growth simulation over GPU/CPU heterogeneous architectures," *International Journal of Geographical Information Science*, 2015.

[12] K. Ohkura, T. Yasuda, Y. Matsumura, and M. Kadota, "GPU implementation of food-foraging problem for evolutionary swarm robotics systems," in *Swarm Intelligence*, pp. 238–245, Springer International, 2014.

[13] H. Al-Aqrabi, L. Liu, R. Hill, and N. Antonopoulos, "Cloud BI: future of business intelligence in the cloud," *Journal of Computer and System Sciences*, vol. 81, no. 1, pp. 85–96, 2015.

[14] V. Chang, "The business intelligence as a service in the cloud," *Future Generation Computer Systems*, vol. 37, pp. 512–534, 2014.

[15] M. Ramachandran and V. Chang, "Financial software as a service—a paradigm for risk modelling and analytics," *International Journal of Organizational and Collective Intelligence*, vol. 4, no. 3, pp. 65–89, 2014.

[16] M. Muntean, L. G. Cabău, and V. Rînciog, "Social business intelligence: a new perspective for decision makers," *Procedia—Social and Behavioral Sciences*, vol. 124, pp. 562–567, 2014.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 465074, 9 pages
http://dx.doi.org/10.1155/2015/465074

Research Article

Optimal Decomposition of Service Level Objectives into Policy Assertions

Yousef Rastegari and Fereidoon Shams

Department of Computer Engineering and Science, Shahid Beheshti University, P.O. Box 1983969411, Velenjak, Tehran, Iran

Correspondence should be addressed to Yousef Rastegari; y_rastegari@sbu.ac.ir

Received 2 October 2015; Accepted 29 November 2015

Academic Editor: Arunkumar Sangaiah

WS-agreement specifies quality objectives that each partner is obligated to provide. To meet quality objectives, the corresponding partner should apply appropriate policy assertions to its web services and adjust their parameters accordingly. Transformation of WS-CDL to WSBPEL is addressed in some related works, but neither of them considers quality aspects of transformation nor run-time adaptation. Here, in conformance with web services standards, we propose an optimal decomposition method to make a set of WS-policy assertions. Assertions can be applied to WSBPEL elements and affect their run-time behaviors. The decomposition method achieves the best outcome for a performance indicator. It also guarantees the lowest adaptation overhead by reducing the number of service reselections. We considered securities settlement case study to prototype and evaluate the decomposition method. The results show an acceptable threshold between customer satisfaction—the targeted performance indicator in our case study—and adaptation overhead.

1. Introduction

WS-CDL is the choreography standard to describe collaborative business processes. It shows a global view of all interactions among local WSBPEL processes. Since WS-CDL specifies only functional responsibilities of each partner, WS-agreement [1] is used to define service level objectives between a service provider and its consumers. It will assure consumers that they get the service they pay for and will obligate the service provider to achieve its service promises [2]. Therefore, service provider should explicitly manage quality properties of its local WSBPEL processes to realize service level objectives. Assessing the impact of new quality objectives on WSBPEL processes is not straightforward. Although quality objectives are defined at choreography level, they must be achieved at orchestration level by applying appropriate policy assertions (see Figure 1).

There are two types of transformation including model-driven (with the goal of integration) and formal (with the goal of verification) in the literature, but neither of them considers quality aspects of transformation nor run-time adaptation. The model-driven approaches translate a WS-CDL element to its respective replacement in terms of BPEL as well as WSCDL. This enables tracing down changes from choreography to orchestration and vice versa which is an important issue in the choreography adaptation scope. On the other hand, some studies formalize the WS-CDL elements. They tried to verify several aspects of service choreography like protocol compatibility, time constraints, and message ordering.

In this paper, we proposed a method to decompose service level objectives to WS-policy assertions. The assertions can be applied to WSBPEL elements and affect their run-time behaviors. While assertions describe what should be done with regard to quality objectives, adaptation strategies implement activities to achieve the objectives. For example, a performance assertion may cause the substitution of a slow service provider with a more efficient one (i.e., service reselection adaptation strategy).

The rest of the paper is organized as follows. The structure of WS-agreement is described in Section 2. In Section 3, we explain how to optimally decompose service level objectives

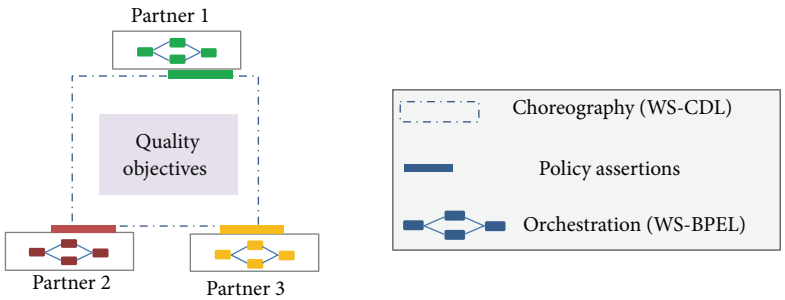


Figure 1: Policy assertions control WS-BPEL processes to satisfy quality objectives.

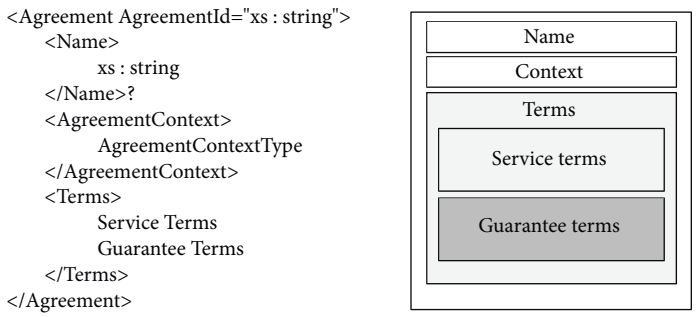


Figure 2: Structure of an agreement.

into policy assertions and associate such policies with service subjects to which they should apply. To prototype and evaluate the proposed method, we define securities settlement case study in Section 4. This section is concluded by adaptation efficiency results. Section 5 presents related works. Finally, Section 6 provides conclusions.

2. Structure of an Agreement

A service provider proposes an agreement template based on its capabilities, resources, and accepted agreement offers with other providers. As shown in Figure 2, an agreement is conceptually composed of several distinct parts. The section after the (optional) name is the context, which contains the meta-data for the entire agreement. It names the participants in the agreement and the agreement's lifetime. The next section contains the terms that describe the agreement itself.

An agreement defines service level attributes and service level objectives (SLOs). SLOs are expressed as assertions over service attributes and/or external factors such as date and time. Service terms contain *ServiceProperties* and *ServiceReference* to describe all aspects of service attributes. Guarantee terms contain *SeviceScope*, *QualifyingCondition*, and *ServiceLevelObjective* to specify a conditional assertion over a specific service term.

3. Proposed Optimal Decomposition Method

The process of handling user request is shown in Figure 3. For each user request, we identify appropriate quality values based on user's preferences and web services' quality assurance. To apply the identified quality values, we define their corresponding policy assertions. Then, to achieve the assertions, we realize renegotiation and reselection adaptation strategies which change the existing agreements or modify the existing service providers, respectively.

To decompose (choreography-level) service level objectives into (orchestration-level) policy assertions, first we integrate their related standards (see Figure 4) and then present the decomposition method.

3.1. Policy Definition and Assignment. A policy contains one or more assertions; each one identifies requirements or capabilities of a policy subject. Policy assertions indicate low-level constraints on quality of services. For example, WS-ReliableMessaging [3] describes a protocol for reliable delivery of SOAP messages, or WS-Security describes enhancements to SOAP messaging to provide quality of protection [4].

As shown in Algorithm 1, we specified a generic format for defining domain-specific assertions. The policy *@id* attribute identifies the policy expression within the enclosing XML document. The *[QoS]Token* identifies what quality of service this assertion relates to (e.g., PerformanceToken). The *weight* attribute defines the impact of this service on the whole WSBPEL process; it mostly relates to user's preferences. The definition of quality of services and their measurement metrics are domain-specific; therefore, assertions are also expected to contain context of use. The *TokenType* is used to specify suitable metric for current context. For example,

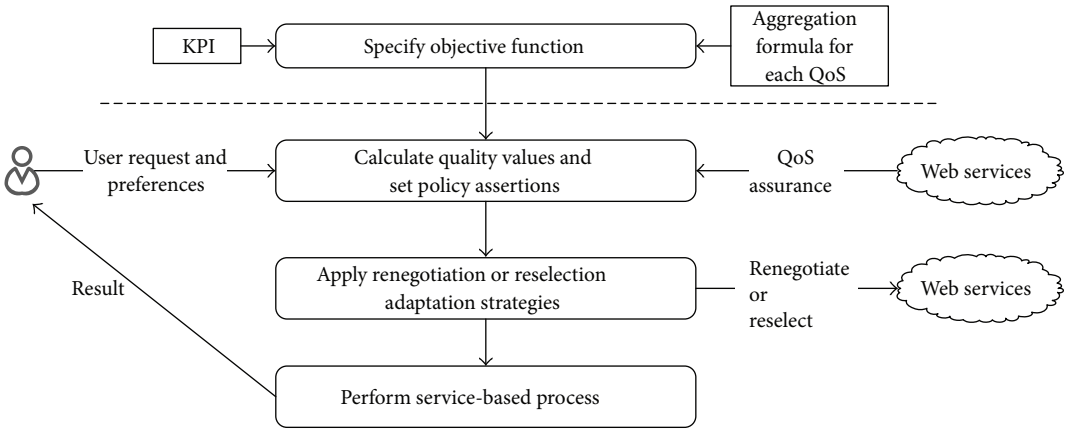


Figure 3: The process of adapting service-based process to user's preferences.

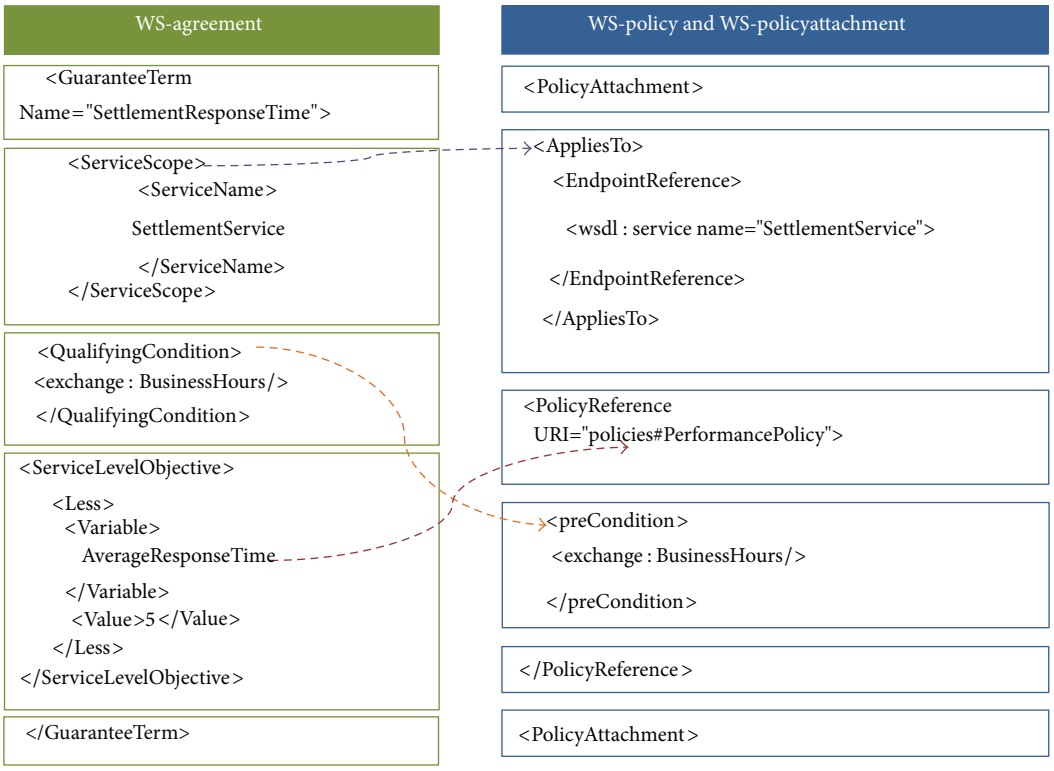


Figure 4: Integrating WS-agreement with orchestration-level standards.

performance can be measured by average response time or throughput in Website Load or Job Submission System, respectively. The *TokenValue* indicates the value of a quality attribute. In our work, the assertions and their attributes and values are specified after decomposition of service level objectives.

Policies can be referenced internally or externally by *PolicyReference* element. The *PolicyReference @URI* attribute references a policy. In Algorithm 2, a policy with *#PerformancePolicy* identity is referenced externally by *wsdl:service* element. In this example, the performance policy will be applied to behaviors or aspects of the *SettlementService* as a whole.

WS-agreement supports both policy definition and policy assignment. WS-Policy is also a standard framework to model and express assertions, but it does not support policy assignment. Therefore, we used WS-PolicyAttachment as an extension for applying policies to service elements. WS-PolicyAttachment defines two general-purpose mechanisms for associating policies with service subjects to which they apply [16]. The policies can be applied to any part of a service such as endpoint, binding, port, portType, operation, and

```
<!- - A generic format for policy assertion - ->
<Policy Id="xs:string" xmlns:wsp="http://www.w3.org/ns/ws-policy">
   <[QoS]Token weight="xs:float" usage="optional | mandatory">
      <TokenType>xs:string[responseTime | throughput |...]</TokenType>
      <TokenValue>xs:float</TokenValue>
   </[QoS]Token>
</Policy>
<!- - A performance policy example - ->
<Policy Id="PerformancePolicy">
   <PerformanceToken weight="0.3" usage="mandatory">
      <TokenType>AverageResponseTime</TokenType>
      <TokenValue>5</TokenValue>
   </PerformanceToken>
</Policy>
```

Algorithm 1: A generic format for service policy assertion.

```
<wsdl:service name="SettlementService">
   <PolicyReference URI="/policies#PerformancePolicy"/>
</wsdl:service>
```

Algorithm 2: Policy reference is used to apply a performance policy to settlement service.

```
<!- - External mechanism- ->
<PolicyAttachment>
   <AppliesTo>
      <EndpointReference>
         <wsdl:service name="SettlementService">
      </EndpointReference>
   </AppliesTo>
   <Policy ReferenceURI="policies#PerformancePolicy" utility="float">
      <preCondition>
         <exchange:BusinessHours/>
      </preCondition>
   </PolicyReference>
</PolicyAttachment>

<!- - Embedded mechanism- ->
<wsdl:service name="SettlementService">
   <PolicyReference URI="policies#PerformancePolicy" utility="float">
      <preCondition>
         <exchange:BusinessHours/>
      </preCondition>
   </PolicyReference>
</wsdl:service>
```

Algorithm 3: Policy attachment mechanisms.

message. As shown in Algorithm 3, the external mechanism consists of *AppliesTo* and *PolicyReference* elements for identifying which service subjects are under control of which policies. The embedded mechanism adds *PolicyReference* besides a service subject to explicitly determine the scope of control.

3.2. Integration. Figure 4 shows the mapping between WS-agreement and "WS-policy and WS-PolicyAttachment" standards. Using policy attachment mechanisms, WS-agreement's *ServiceScope* is realized either by *AppliesTo* element or by adding *PolicyReference* to a specific service subject.

WS-agreement's *QualifyingCondition* is an optional condition that must be met (when specified) for a guarantee to be enforced. It is used to express a precondition for several service level objectives. Therefore, we add *preCondition* to *PolicyReference* to specify under what condition the policy should be applied. The type of *preCondition* is *xs:anyType*. It can be extended with assertion languages to address the requirements of the particular collaborative domain.

3.3. Decomposition. Performance requirements on business processes are specified as performance indicators with target values, which are to be achieved in a certain analysis period [17]. A key performance indicator (KPI) is a key metric which is measured by run-time monitoring mechanisms to prevent violation. It has both quality aspect (e.g., subprocess duration, service availability, and service security) and process aspect (e.g., number of ordered products and type of customer) [18]. In some cases, KPI measurement is not absolute; it relates to user's preferences. For example, customer satisfaction is a relative KPI; it relates to quality levels that fulfill user's preferences. Preference provides a mean by which a user can specify service quality levels that he would be satisfied with. For example, when an individual requests low cost and high performance process execution, all partners should consider these constraints while providing a service. While KPI forces quality requirements on a collaborative business process in general, SLOs specify constraints only on those service properties which are offered to service consumers. Therefore, service providers have to control service policies and apply adaptation strategies to comply with both collaborative agreement and user's preferences.

A business process includes several service-centric tasks; each one binds to a web service at run time. The web services communicate using different message exchange (composition) patterns such as sequence, parallel, loop, fork, and join. Here, we present a method to optimally decompose a service level objective and specify corresponding quality policies over service-centric tasks.

We applied nonlinear programming method to model our optimization problem. As shown below, the objective function is a specific KPI. KPI is defined based on four following factors: (1) context of use, (2) quality aggregation formula [19] per each quality of service, (3) service provider quality assurance, and (4) users' preferences. In (1), Q_i specifies ith quality of service, where i specifies number of effective qualities on KPI, w_i specifies the weight (importance) of Q_i, and F_{Q_i} is the aggregation formula for Q_i. F_{Q_i} would be different per each quality of service and composition pattern. For example, suppose a sequence of web services; the cost and availability aggregation formulas are specified as follows: $F_{\text{cost}} = \sum_{i=1} C(i)$; $F_{\text{availability}} = \prod_{i=1} A(i)$. The model of optimization problem is specified as follows.

Find the maximum (minimum) value of the objective function:

$$\text{KPI} = \sum_{i=1} w_i F_{Q_i}, \tag{1}$$

where

$$F_{Q_i} = \text{QoS aggregation formula for } Q_i, \tag{2}$$

which is subject to the following:

(i) For each Q_i, specify quality assurance of each service.

(ii) For each Q_i, specify users' preferences.

4. Evaluation

4.1. Case Study: Securities Settlement [20, 21]. Clearing and settlement are fundamental processes in financial markets. After the trade is executed, the record is submitted to the clearing agency, which matches the buyer and seller records and confirms that the counterparts agree to the terms. After the clearing process is performed, agencies fulfill the delivery requirements of the securities through settlement process. The settlement agency receives cash from buyers and securities from sellers and, at the end of the process, gives the securities to the buyer and the cash to the seller. The clients of the clearing and settlement agencies are brokerage companies. Brokerage companies can use different settlement and clearing agencies at the securities market. They receive trades from various customers like private individuals, institutional investors, or companies issuing or managing bonds. As shown in Figure 5, the customers can submit trades they made, to be settled by the brokerage companies through *settlement service* interface. From the brokerage company's perspective, it should bind to *clearing* and *settlement* web services to exchange securities. For the customers of the brokerage company, performance and cost of the *settlement service* are essential; they are the parameters of customer's preferences. The performance relates to the response time of both clearing and settlement processes. The cost relates to consumption of clearing and settlement web services and brokerage company's resources. The brokerage company can use different agencies at the securities market. After they receive a trade submission from a customer, they negotiate with existing agencies for new quality agreements or discover suitable agencies for satisfying the customer's preferences.

4.2. Prototype. As shown in Figure 5, the securities settlement process includes a sequence of two service-centric tasks (clearing and settlement). The objective function is customer satisfaction which depends on response time and cost quality attributes. The goal is to achieve the maximum customer satisfaction; therefore we should get the minimum response time and cost by considering both service quality assurance levels and user's preferences. As shown below, Z defines the aggregation formula for a sequence of two web services. We also consider 0.6 and 0.4 weight (importance) factors for response-time and cost, respectively. The maximum customer satisfaction is achieved when the value of Z is minimum. Response-time is abbreviated to rt; cost is

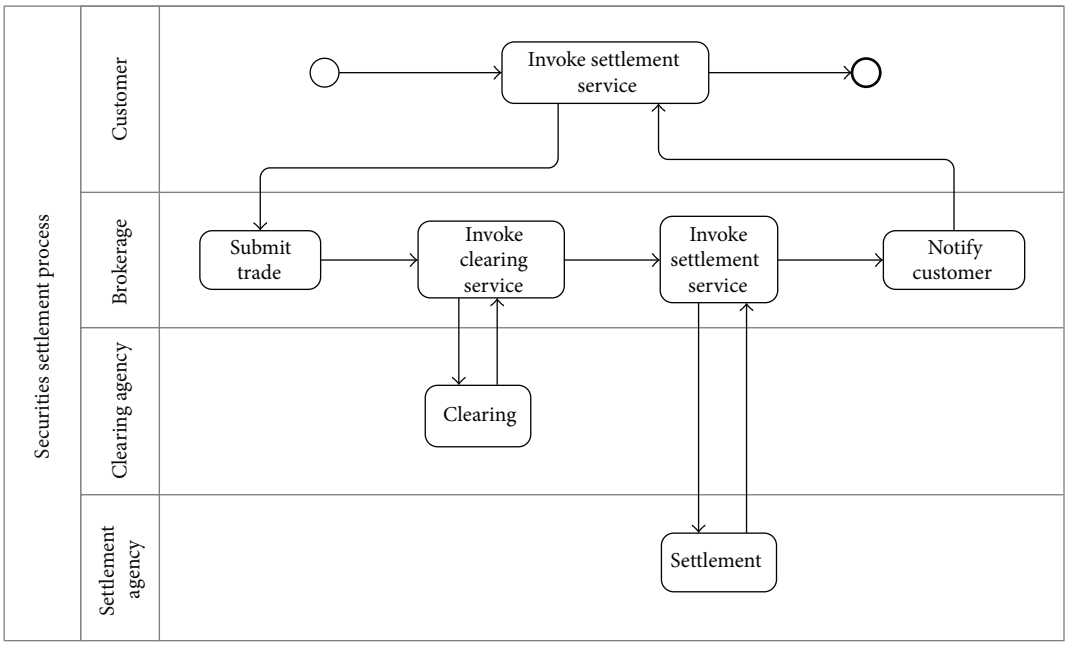


Figure 5: Securities settlement process.

abbreviated to c; clearing and settlement web services are indicated by their names, respectively:

$$\begin{aligned}
&\text{Maximize} \quad \text{Customer Satisfaction} = 100 - Z,\\
&\qquad Z = 0.6 \times \left(\mathrm{rt}_{\text{clearing}} + \mathrm{rt}_{\text{settlement}}\right) + 0.4 \times \left(c_{\text{clearing}} + c_{\text{settlement}}\right)\\
&\text{Subject to} \quad \text{User's preferences} = \begin{cases} \text{medium resonse-time} \\ \text{low cost} \end{cases}\\
&\qquad \text{Quality assurance levels} = \begin{cases} 12 \le \mathrm{rt}_{\text{clearing}} \le 20 \\ 5 \le c_{\text{clearing}} \le 15 \\ 7 \le \mathrm{rt}_{\text{settlement}} \le 15 \\ 15 \le c_{\text{settlement}} \le 30 \end{cases}\\
&\qquad \text{Response-time ranges} = \begin{cases} \text{low: } 0\text{–}30 \\ \text{medium: } 30\text{–}60 \\ \text{high: } 60\text{–}100 \end{cases}\\
&\qquad \text{Cost ranges} = \begin{cases} \text{low: } 0\text{–}20 \\ \text{medium: } 20\text{–}50 \\ \text{high: } 50\text{–}100. \end{cases}
\end{aligned} \tag{3}$$

$$\text{Cost ranges} = \begin{cases} \text{low: } 0\text{–}20 \\ \text{medium: } 20\text{–}50 \\ \text{high: } 50\text{–}100. \end{cases} \tag{4}$$

Applying the simplex method [22] to this problem results in

$$\begin{aligned}
\mathrm{rt}_{\text{clearing}} &= 20,\\
\mathrm{rt}_{\text{settlement}} &= 10,\\
c_{\text{clearing}} &= 5,\\
c_{\text{settlement}} &= 15 \longrightarrow Z = 26.
\end{aligned} \tag{5}$$

So, the maximum value for customer satisfaction is

$$\text{Customer Satisfaction} = 74. \tag{6}$$

According to above results, the brokerage company should apply four new policies to service-centric tasks. The new policies are shown in Table 1. The brokerage company should also realize renegotiation adaption strategy to set new agreement offers based on new policies and send the offers to the clearing and settlement agencies.

4.3. Adaptation Efficiency. The decomposition method insists on staying with existing providers and negotiating for new quality agreements. But if the method does not find any solution to satisfy all constraints, the corresponding partner should apply other adaptation strategies such as reselection or reconfiguration. These strategies impose more overhead than renegotiation but may result in higher customer satisfaction,

Table 1: Performance and cost policies.

	Performance (it is measured by response-time)	Cost (it is measured by dollar)
Clearing service	<Policy Id="clearing-Performance">& <PerformanceToken usage="mandatory">& <TokenType>responseTime</TokenType>& <TokenValue>20</TokenValue>& </PerformanceToken>& </Policy>	<Policy Id="clearing-Cost">& <CostToken usage="mandatory">& <TokenType>dollar</TokenType>& <TokenValue>5</TokenValue>& </CostToken>& </Policy>
Settlement service	<Policy Id="settlement-Performance">& <PerformanceToken usage="mandatory">& <TokenType>responseTime</TokenType>& <TokenValue>10</TokenValue>& </PerformanceToken>& </Policy>	<Policy Id="settlement-Cost">& <CostToken usage="mandatory">& <TokenType>dollar</TokenType>& <TokenValue>15</TokenValue>& </CostToken>& </Policy>

because they discover new service providers and set up an agreement which matches exactly what a user requested. This means that the decomposition method causes a bit of dissatisfaction but guarantees the lowest adaptation overhead. Thus, we set up an evaluation plan to measure the customer satisfaction and the adaptation overhead for the following scenarios, according to the securities settlement case study.

Scenario I. Our proposed decomposition method is used. The method provides user's preferences by using existing clearing and settlement agencies. It insists on renegotiation. The reselection is used, only if the existing agencies cannot provide quality objectives.

The customer satisfaction was measured for 72 requests with different preferences. For each request, the preferences of response-time and cost were defined randomly as either low, medium, or high. These ranges were defined as follows: response-time {low: 5–40, medium: 41–60, high: 61–100}, cost {low: 5–40, medium: 41–70, high: 71–100}. At the beginning of the evaluation, the quality assurance levels of clearing and settlement web services were set as follows: clearing web service {response-time: 17–57, cost: 18–65}, settlement web service {response-time: 25–60, cost: 45–75}. We also considered 10 alternative candidates for clearing and settlement web services; each one offers different quality assurance levels.

The reselection adaptation strategy consists of the three following activities: (1) service discovery, (2) service selection, and (3) service binding. The negotiation for service level agreement is also done during service binding. Therefore, we considered (1) and (3) for the overhead value of realizing renegotiation and reselection adaptation strategies, respectively. The evaluation started based on the above configuration. The decomposition method was used to measure the customer satisfaction for each user's preferences, to see whether they can be accomplished by negotiating with existing web service providers (i.e., renegotiation) or selecting new ones (i.e., reselection).

Scenario II. A reselection algorithm is used to discover and select new clearing and settlement agencies for each user's preferences.

Table 2: Adaptation efficiency results.

	Scenario I	Scenario II
Number of renegotiations	48	0
Renegotiation overhead	48	0
Number of reselections	24	72
Reselection overhead	72	216
Total adaptation overhead value (%)	120 (55%*)	216 (100%)
Overhead reduction	45%	
Customer satisfaction value (%)	4591 (87%*)	5221 (100%)
Customer dissatisfaction	13%	

*The value is calculated as follows: (value of scenario I ÷ value of scenario II) × 100.

Reselection algorithm finds new service providers that match exactly what user requested. As a consequence of selecting new suitable web services, the maximum customer satisfaction is achieved but it comes with more adaptation overhead.

As depicted in Table 2, the results show that the decomposition method (Scenario I) achieves 87% customer satisfaction and 45% adaptation overhead reduction, in comparison with Scenario II.

5. Related Work

There are some related works in transformation of WS-CDL to WSBPEL, but neither of them considers quality aspects of transformation nor run-time adaptation. In [5, 6], a model-driven transformation approach is proposed to drive BPEL process definitions from a global WS-CDL model. It proposes a mapping between WS-CDL and WSBPEL building blocks. In addition, the mapping can be used to generate WS-CDL description from existing WS-BPEL processes. In another model-driven approach, CDL2BPEL [7] algorithm translates WS-CDL to "BPEL and WSDL" elements, according to a knowledge base. The knowledge base contains generic patterns to translate a WS-CDL entity to its respective

Table 3: Classification table.

	Usage	Aspect	Strategy
Our work	Integration, adaptation	Quality requirements, user's preferences	Model-driven approach to generate WS-policy assertions and tune BPEL processes based on user's preferences
[5, 6]	Integration	Functional requirements	Model-driven approach to drive BPEL processes from WS-CDL
CDL2BPEL [7]	Integration	Functional requirements	Knowledge-based approach to translate WS-CDL to "BPEL and WSDL" elements
BPEL4Chor [8]	Integration	Functional requirements	A new language which supports both BPEL and WS-CDL structures
CDL [9], timed automata [10], and colored Petri-net [11, 12]	Verification	Functional requirements	Formal specification and verification of WS-CDL
MOSES [13], fuzzy approaches [14, 15]	Adaptation	Quality, context information	By applying recomposition and reselection strategies to adapt an application to context information

replacements in terms of BPEL as well as optional WSDL. The algorithm extracts WSDL interfaces from interactions and "tokens/token locators." BPEL4Chor [8] is an intermediary language to align choreography and orchestration. BPEL4Chor adds a few constructs on top of BPEL to support all service interaction patterns. In addition to being an executable language, BPEL4Chor is an alternative for WS-CDL. More recently, a simple language (CDL) [9] was introduced to formalize the WS-CDL's participant roles and the collaborations among roles. They used SPIN model-checker to reason about properties that should be satisfied by the specified system automatically. Furthermore, in order to verify WS-CDL protocol mismatches, the transformation rules were proposed to describe the WS-CDL entities with timed automata [10] and colored Petri-net [11, 12] specifications. MOSES [13] is a QoS-based adaptation framework based on MAPE components. It is classified as an adaptive adaptation method. MOSES uses abstract composition to create new processes and also service selection to dynamically bind the processes to different concrete web services. MOSES is applicable where a service-oriented system is architected as a composite service. In [14], fuzzy controllers are applied to improve service-based applications based on context information (e.g., user, environment, and computational contexts). Beggas et al. [15] proposed middleware that calculates ideal QoS model using a fuzzy control system to fit context information and user preferences. Then, the middleware selects the best service among all variants having the nearest QoS value to the ideal. We used the following indicators to classify the related studies: what is the usage of approach? (integration, adaptation, and verification), which aspect does the approach cover? (functionality, quality), and what strategy is used to meet the research objective? The classification results are depicted in Table 3.

6. Conclusion

In this paper, we proposed a method to decompose WS-agreement service level objectives into WS-policy assertions. Assertions can be applied to web service elements and control their run-time behaviors. As a result, we can easily change quality objectives in a collaborative domain and track their effects on all relevant service parameters of each partner. In response to new assertions, service adaptation strategies (e.g., renegotiation, reselection) are applied to modify WSBPEL processes.

Considering renegotiation and reselection service adaptation strategies, the latter has more overhead; because a new service provider should be discovered, a new agreement offer should be sent to the provider and a new binding should be established. Apart from overhead of adaptation, there are other constraints like user's preferences and service quality assurance levels, which should be considered, if we want to get the best outcome for a business performance indicator. We modeled these constraints in an optimization problem using nonlinear programming method. We evaluated the method in securities settlement case study. The method reduces adaptation overhead and achieves acceptable customer satisfaction.

Symbols

Parameters

Q_i: ith quality of service
w_i: Weight (importance) of Q_i
F_{Q_i}: Aggregation formula for Q_i
rt: Response-time
c: Cost.

Abbreviations

WS-CDL: Web service choreography description language
WSBPEL: Web service business process execution language
KPI: Key performance indicator
QoS: Quality of service.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] A. Andrieux, K. Czajkowski, A. Dan et al., "Web services agreement specification (WS-agreement)," in *Proceedings of the 19th Open Grid Forum (OGF19 '07)*, pp. 128–216, Chapel Hill, NC, USA, January-February 2007.

[2] L. Jin, V. Machiraju, and A. Sahai, *Analysis on Service Level Agreement of Web Services*, Software Technology Laboratories, HP Laboratories, Palo Alto, Calif, USA, 2002.

[3] P. Gilbert, D. Doug, A. Karmarkar, S. Winkler, and U. Yalcinalp, *OASIS Web Services Reliable Messaging Protocol*, OASIS, Pasadena, Calif, USA, 2006.

[4] K. Lawrence, C. Kaler, A. Nadalin, R. Monzillo, and P. Hallam-Baker, "Web services security," *Computer Fraud & Security*, vol. 2003, no. 3, pp. 15–17, 2003.

[5] J. Mendling and M. Hafner, "From WS-CDL choreography to BPEL process orchestration," *Journal of Enterprise Information Management*, vol. 21, no. 5, pp. 525–542, 2008.

[6] J. Mendling and M. Hafner, "From inter-organizational workflows to process execution: generating BPEL from WS-CDL," in *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops*, R. Meersman, Z. Tari, and P. Herrero, Eds., vol. 3762 of *Lecture Notes in Computer Science*, pp. 506–515, Springer, Berlin, Germany, 2005.

[7] I. Weber, J. Haller, and J. A. Mülle, "Automated derivation of executable business processes from choreographies in virtual organisations," *International Journal of Business Process Integration and Management*, vol. 3, no. 2, pp. 85–95, 2008.

[8] G. Decker, O. Kopp, F. Leymann, and M. Weske, "BPEL4Chor: extending BPEL for modeling choreographies," in *Proceedings of the IEEE International Conference on Web Services (ICWS '07)*, pp. 296–303, IEEE, Salt Lake City, Utah, USA, July 2007.

[9] H. Yang, X. Zhao, Z. Qiu, G. Pu, and S. Wang, "A formal model for web service choreography description language (WS-CDL)," in *Proceedings of the IEEE 20th International Conference on Web Services*, pp. 893–894, IEEE, Chicago, Ill, USA, September 2006.

[10] G. Diaz, J.-J. Pardo, M.-E. Cambronero, V. Valero, and F. Cuartero, "Automatic translation of WS-CDL choreographies to timed automata," in *Formal Techniques for Computer Systems and Business Processes*, M. Bravetti, L. Kloul, and G. Zavattaro, Eds., vol. 3670 of *Lecture Notes in Computer Science*, pp. 230–242, Springer, Berlin, Germany, 2005.

[11] V. Valero, H. MacI, J. J. Pardo, M. E. Cambronero, and G. Díaz, "Transforming Web Services Choreographies with priorities and time constraints into prioritized-time colored Petri nets," *Science of Computer Programming*, vol. 77, no. 3, pp. 290–313, 2012.

[12] M. S. Benabdelhafid and M. Boufaida, "Toward a better interoperability of enterprise information systems: a CPNs and timed CPNs -based web service interoperability verification in a choreography," *Procedia Technology*, vol. 16, pp. 269–278, 2014.

[13] V. Cardellini, E. Casalicchio, V. Grassi, S. Iannucci, F. L. Presti, and R. Mirandola, "MOSES: a framework for qos driven runtime adaptation of service-oriented systems," *IEEE Transactions on Software Engineering*, vol. 38, no. 5, pp. 1138–1159, 2012.

[14] H. Takatsuka, M. Nakamura, S. Saiki, and S. Matsumoto, "Developing service platform for web context-aware services towards self-managing ecosystem," in *Proceedings of the 3rd International Workshop on Self-Managing Pervasive Service Systems*, pp. 73–82, November 2014.

[15] M. Beggas, L. Médini, F. Laforest, and M. T. Laskri, "Towards an ideal service QoS in fuzzy logic-based adaptation planning middleware," *Journal of Systems and Software*, vol. 92, no. 1, pp. 71–81, 2014.

[16] A. Vedamuthu, D. Orchard, F. Hirsch et al., "Web Services Policy 1.5—Attachment," W3C Recommendation, 2007.

[17] B. Wetzstein, D. Karastoyanova, and F. Leymann, "Towards management of SLA-aware business processes based on key performance indicators," in *Proceedings of the 9th Workshop on Business Process Modeling Development and Support (BPMDS '08)*, Montpellier, France, June 2008.

[18] A. Metzger, K. Pohl, M. Papazoglou, E. Di Nitto, A. Marconi, and D. Karastoyanova, "Research challenges on adaptive software and services in the future internet: towards an S-Cube research roadmap," in *Proceedings of the 1st International Workshop on European Software Services and Systems Research—Results and Challenges (S-Cube '12)*, pp. 1–7, IEEE Press, Zürich, Switzerland, June 2012.

[19] G. Canfora, M. Di Penta, R. Esposito, and M. L. Villani, "An approach for QoS-aware service composition based on genetic algorithms," in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, pp. 1069–1075, ACM, June 2005.

[20] P. Pezzutti, *Trading the US Markets: A Comprehensive Guide to US Markets for International Traders and Investors*, Harriman House, 2008.

[21] H. Ludwig, *WS-Agreement Concepts and Use-Agreement-Based Service-Oriented Architectures*, IBM Research Division, Yorklown Heights, NY, USA, 2006.

[22] G. Dantzig, "Maximization of a linear function of variables subject to linear inequalities," in *The Basic George B. Dantzig*, R. Cottle, Ed., pp. 24–32, Stanford University Press, 2003.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 284276, 9 pages
http://dx.doi.org/10.1155/2015/284276

Research Article

Energy-Aware Multipath Routing Scheme Based on Particle Swarm Optimization in Mobile Ad Hoc Networks

Y. Harold Robinson[1] and M. Rajaram[2]

[1] *Department of Computer Science and Engineering, SCAD College of Engineering and Technology, Tirunelveli, Tamil Nadu 627414, India*
[2] *Anna University, Chennai, Tamil Nadu 600 025, India*

Correspondence should be addressed to Y. Harold Robinson; haroldrobinsony@gmail.com

Received 17 September 2015; Accepted 24 November 2015

Academic Editor: Muthu Ramachandran

Mobile ad hoc network (MANET) is a collection of autonomous mobile nodes forming an ad hoc network without fixed infrastructure. Dynamic topology property of MANET may degrade the performance of the network. However, multipath selection is a great challenging task to improve the network lifetime. We proposed an energy-aware multipath routing scheme based on particle swarm optimization (EMPSO) that uses continuous time recurrent neural network (CTRNN) to solve optimization problems. CTRNN finds the optimal loop-free paths to solve link disjoint paths in a MANET. The CTRNN is used as an optimum path selection technique that produces a set of optimal paths between source and destination. In CTRNN, particle swarm optimization (PSO) method is primly used for training the RNN. The proposed scheme uses the reliability measures such as transmission cost, energy factor, and the optimal traffic ratio between source and destination to increase routing performance. In this scheme, optimal loop-free paths can be found using PSO to seek better link quality nodes in route discovery phase. PSO optimizes a problem by iteratively trying to get a better solution with regard to a measure of quality. The proposed scheme discovers multiple loop-free paths by using PSO technique.

1. Introduction

A MANET [1] is composed of mobile nodes connected by wireless media without centralized infrastructure. Routing schemes such as Dynamic Source Routing [2] and Ad Hoc On-Demand Distance Vector [3] were implemented to perform basic routing operation like forward data packets from a source to a destination. Routing schemes should consider the characteristics of the MANET. The prime needs of a MANET are reliability of data transmission, multipath selection [4], and providing security [5] which increases the network performance. Many of the researches have been developed to achieve this goal. On-demand routing [6] is one of the essential functions in MANET. The routing schemes [7] should be reliable, robust, and flexible in an ad hoc environment. Routing function is restricted by dynamic topology and link failure of the nodes. The mobility of the nodes increases the complexity of routing function because it causes of link failure between nodes. This frequent link failure leads to routing overhead and topology management, reduces the reliability of data transmission, and reduces the efficiency of the network. Hence, the link failure in MANET becomes a vital issue. Further, this sort of link failure also leads to frequent path failures. As a result, the reliability of data transmission gets reduced; the lesser the packet delivery ratio, the longer the end-to-end delay. Retransmission of data packets in a MANET is costly increasing the control message overhead and reducing the efficiency of the routing function. Hence, it is very much essential to select an alternate path or to form multiple paths for ensuring reliability of data transfer when a link failure occurs. And also loop-free path is very much interested in finding an optimal path among multiple paths between source and destination in a network.

Multipath routing in a MANET [8] is established in order to increase the reliability of data transmission that provides load balancing among the nodes. The use of multiple

disjoint paths transferred the data in parallel that significantly increases the packet delivery ratio. Multipath routing schemes [9] deal with the problem of scalability, confidentiality, integrity, and network lifetime. Multiple-path routing [10] between source and destination ensures reliability of the data transmission in a MANET. Existing multipath routing schemes in a MANET lead to problems such as flooding, empty set of neighbors, flat addressing, widely distributed information, large energy consumption, interference, and load balancing issues. Therefore, the efficient multipath routing scheme is proposed to solve one or more of these issues. And also the existing multipath routing schemes do not perform well in dynamic environment change and frequent path failure. They also generate a routing overhead in the network. The routing overhead occupies a considerable portion of network bandwidth and the energy of the mobile node exhausts rapidly. Hence, with minimum overhead, the reliable multipath routing protocol is essential for designing to restrict the participation of mobile nodes in a route discovery phase that ensure reliability of data transmission. Evolutionary mechanism paradigm [11, 12] is most suitable to resolve multiobject problems because they are based on population. It generates a set of solutions in one run. There is no single solution that can be termed as the optimal solution [13] in multiobjective problems. In particle swarm optimization (PSO) [14, 15], the potential solutions of the problem are called particles. The group of particles becomes a swarm that searches for an optimum solution. Particle swarm optimization [16, 17] is a stochastic optimization technique in which the particles fly in the search space and adjust to the velocities dynamically according to their historical behaviors. This process guides the particles to soar toward the better search area in the search space [18]. It is difficult to ensure the reliability of the data transmission from source to destination because of its dynamism property and frequent link failure in MANET. The PSO can be applied to solve this kind of issue.

We design an energy-aware multipath routing scheme based on particle swarm optimization that uses continuous time recurrent neural network to solve optimization problems. Its goal is to discover multiple loop-free paths by using PSO technique. For gaps in the above literatures review, this work gives an optimum path selection technique that produces a set of optimal paths between source and destination to enhance the reliability of the data transmission.

The remainder of the paper contains five sections. Section 2 summarizes previous research works. Section 3 describes the proposed energy-aware multipath routing scheme based on particle swarm optimization technique in a MANET. Section 4 presents our simulation results and a relevant performance analysis. Finally, Section 5 presents our conclusions and future direction.

2. Related Works

Evolutionary mechanisms [19] and clustering scheme [20] play a vital role to find optimal solutions for network partition. A number of evolutionary mechanisms have been proposed such as genetic algorithm [21, 22], artificial neural network system [23], particle swarm intelligence [24], and particle swarm optimization based clustering [25–27] that reveals the best global solutions. Dengiz et al. [28] proposed a particle swarm optimization algorithm that conceptualizes an autonomous topology optimization for mobile ad hoc networks using multiple mobile agents. The MANET communication is represented as network flows and optimization using a maximum flow model. This representation is very responsive to small changes in topology when evaluating network connectivity. Goswami et al. [29] proposed a fuzzy ant colony based routing protocol using fuzzy logic and swarm intelligence to select the optimal path by considering optimization of multiple objectives while retaining the advantages of swarm based intelligence algorithm. Ali et al. [30] proposed a multiobjective solution by using multiobjective particle swarm optimization algorithm to optimize the number of clusters in an ad hoc network and reduce the network traffic. This scheme performs intercluster and intracluster traffic that is managed by the cluster-heads. The authors considered the performance measures such as degree of nodes and power consumption of the mobile nodes.

Nasab et al. [31] proposed a multicast routing based on the particle swarm optimization (MPSO) that focused on efficient energy consumption and delay in multicast routing in MANET. It selects the node with the minimum energy consumption in the route selection and builds a multicast tree with minimum delay. There exists route failure in all route discovery methods resulting in data loss and routing overheads. Manickavelu and Vaidyanathan [32] proposed a Particle Swarm Optimization based Lifetime Prediction (PSOLP) algorithm for route recovery in a MANET. This scheme predicts the lifetime of link and node in the available bandwidth based on the parameters such as relative mobility of nodes and energy drain rate. The parameters are fuzzified and fuzzy rules have been formed to decide on the node status for prediction. Then, this piece of information is exchanged among all the nodes for verifying status of a node before data transmission.

Yun-Sheng et al. [33] proposed a multicast QoS based routing approach using genetic algorithm. It uses the available resources and minimum computation time in a dynamic environment. This scheme optimizes the routes by selecting the appropriate values for genetic operations like crossover, mutation, and population size. Radi et al. [34] proposed Low-Interference Energy-efficient Multipath Routing protocol (LIEMRO) to distribute source node traffic over the established paths. It also provides load balancing that estimates the optimal traffic rate of the paths. LIEMRO starts packet transmission immediately after the first path is established. Whenever a new path is created, the load balancing algorithm redistributes the source node traffic, according to the relative quality of the paths. Hurni and Braun [35] proposed a multipath routing protocol based on AOMDV. The objectives of these multipath schemes are to improve energy efficiency and reduce latency through load balancing and using cross-layer information. In order to reduce the end-to-end delay of data forwarding, each node utilizes the information provided by the MAC layer to transmit its packets to the neighboring node that wakes up earlier. Ghiasi and Karimi [36] proposed an algorithm of learning automata adjusting learning rate on

neural network. It is a combination of the back-propagation algorithm, a local search algorithm, and learning automata to provide efficient global search. Mobile network parameters were measured for training and testing the neural network. The learning automata approach does not find optimal solution when the number of nodes increased in the network.

3. Energy-Aware Multipath Routing Scheme Based on Particle Swarm Optimization

MANET consists of mobile nodes with limited energy and wireless link. Each mobile node forwards the packets from source to destination. MANET is represented as directed graph $G = (V, E)$. The vertices $v \in V$ are a symbol of the mobile nodes and the neighbor node. An edge $(u, v) \in E$ is a symbol of a wireless link between nodes u, v, which forward packets to others. The energy consumption for forwarding packets from a node u to node v is given by

$$E_{tx}(k, d) = E_{\text{elec}}(K) + E_{\text{amp}}(k, d), \tag{1}$$

where k is number bits and d is the distance between nodes. $E_{\text{elec}}, E_{\text{amp}}$ are energy dissipated per bit to forward and receive packets, respectively. Energy consumption for receiver is calculated by

$$E_{rx}(k) = E_{\text{elec}}(K). \tag{2}$$

The proposed EMPSO routing scheme is composed of three phases: (i) route setup phase, (ii) route discovery phase, and (iii) route maintenance phase. In the route setup phase, each node acquires its metadata of the neighborhood. This metadata is used in the route discovery to find the best next-hop node towards the destination node. The route discovery is activated whenever a source wants to transmit data to destination in an on-demand fashion that prevents multiple interference between source and destination. The route maintenance phase handles path failures during data transmission.

3.1. Route Setup Phase. In the route setup phase, source node initiates a data transmission for forwarding packets to the destination. Each node in a MANET obtains its metadata of the neighborhood, which also includes the transmission cost (t_c) of its neighbors towards the destination node. The t_c value of a link indicates the required number of transmissions for a successful packet reception at the receiver. The transmission cost of a link is given as follows:

$$t_c = \frac{1}{p \times q}, \tag{3}$$

where p and q are the probabilities of forward and backward packet reception over a link, respectively. In the initialization phase, each node broadcasts the control packets and stores the number of successfully received packets from its neighbors in the routing neighborhood table. Then, the destination node sets its transmission cost to zero and broadcasts this value to its neighbors, when a node receives a transmission cost included in a packet.

3.2. Route Discovery Phase. Whenever a source node wants to transmit data to destination, the route discovery phase is initiated to find multiple paths from the source to destination. The proposed multipath routing protocol uses reliability measures such as transmission cost, optimal traffic ratio, and remaining energy. The source node starts the route discovery by transmitting a route request packet (RR) towards the destination node. Whenever an intermediate node receives a RR packet, it computes the transmission cost, optimal traffic ratio, and remaining energy for a path that is established between the source and the destination. Then, it also used a found path to forward the RR packet to the neighboring node with minimum cost. The reliability measures are stored in the routing table of a node in MANET. The proposed EMPSO scheme uses a continuous time recurrent neural network to find an optimal path among multiple paths. CTRNNs are more computationally efficient in order to use a system of ordinary differential equations to model. The three weight factors such as transmission cost, energy factor, and optimal traffic ratio are taken into the account in CTRNN to find an optimal path. The weight factor of transmission cost is given in

$$w_{\text{tc}} = \text{CR}_{\text{pkt}}\left(t_{i,j}\right) + \text{DR}_{\text{pkt}}\left(t_{i,j}\right), \tag{4}$$

where w_{tc} is the weight factor for transmission cost, CR_{pkt} is the control packet transmission ratio and DR_{pkt} is the data packet transmission ratio from node I to node j, respectively, and $t_{i,j}$ is the time taken for CR_{pkt} and DR_{pkt}. The weight factor for optimal path ratio and remaining energy of a node are calculated as in

$$\begin{aligned} w_{\text{opr}} &= \frac{1}{p_k \sum_{f=1}^{n} 1/p_f}, \\ w_{\text{RE}} &= E_T - \left(E_{\text{TX}} + E_{\text{RX}} + E_{\text{ideal}}\right). \end{aligned} \tag{5}$$

For a neuron i in the network with action potential y_i the rate of change of activation is given in

$$\begin{aligned} T_i y_i = {} & -y_i + \sigma\left(\sum_{i=1}^{n} w_{\text{tc}} y_i + \sum_{i=1}^{n} w_{\text{opr}} y_i + \sum_{i=1}^{n} w_{\text{RE}} y_i\right) \\ & - \theta_j + I_i(t). \end{aligned} \tag{6}$$

Notations used in (5) and (6) are as follows:

p_k: kth path;

E_T: total energy required for packet forwarding from node I to node j;

$E_{\text{TX}}, E_{\text{RX}}, E_{\text{ideal}}$: transmitted energy, received energy, and ideal energy of a node, respectively;

T_i: time constant of postsynaptic node;

y_i: rate of change of activation of postsynaptic node;

w_{tc}: weight vector of transmission cost from presynaptic to postsynaptic node;

w_{opr}: weight vector of optimal path ratio from presynaptic to postsynaptic node;

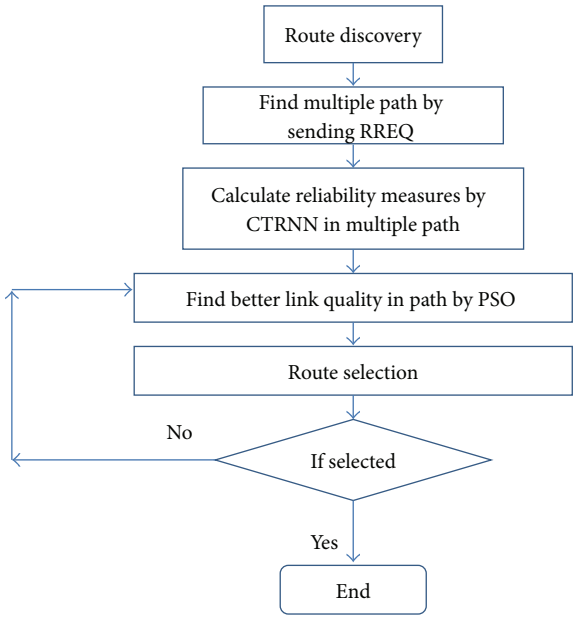


Figure 1: Flow diagram of the proposed system.

w_{RE}: weight vector of remaining energy of node;

$\sigma(x)$: sigmoid of x;

θ_j: bias of presynaptic node;

$I_i(t)$: input to node.

Figure 1 shows the flow diagram of the proposed scheme. Initially, multiple paths are found by route discovery phase. Then, reliability measures for a path are calculated with help of CRRNN. An optimization technique called PSO is used to find better link quality in a path. PSO can be applied to optimization problems that are partially in dynamic topology changing environment. PSO is an evolutionary optimization technique that may be used to seek a good set of weights in CTRNN. PSO is applied to find the best nodes (particles) involved in a path. PSO is metaheuristic that searches large spaces of candidate solutions. A route with a better link quality is selected for forwarding data from source to destination. If a better link quality is not found, PSO function is performed again until global best solution has been found. PSO reduces the traffic and routing overhead of the optimization process and finds the node with best link quality in an ad hoc network.

3.3. PSO Algorithm. Initialize nodes (particle) in a MANET. For particle i, which is at distance $x(i)$

Choose a path p of N particles among multiple paths.

Find $fbest(p)$, the best objective function among the neighbors, and global best $g(p)$, the distance of the neighbor with the best objective function.

Initialize the particle's position with a uniformly distributed random vector (u_1, u_2)

$v = W * v + y_1 * u_1 * (p - x) + y_2 * u_2 * (g - x)$.

This update uses a weighted sum of the following:

(i) The previous velocity v is found by speed of a packet

(ii) The difference between the current distance and the best distance the particle has seen $p - x$

(iii) The difference between the current distance and the best distance in the current neighborhood $g - x$

$x = x + v$ // Update the distance

Enforce the bounds. If any component of x is outside a bound, set it equal to that bound.

$f = \mathrm{fun}(x)$ // Evaluate the objective function

If $f < \mathrm{fun}(p)$, then

$p = x$.

If $f < b$, then

$b = f$ and $d = x$.

$g = p$

g holds the best found solution

If in the previous step or the best function value was lowered, then

flag = true.

Otherwise,

flag = false // The value of flag is used in the next step.

Update the neighborhood.

If flag = true

$c = \max(0, c - 1)$.

minNeighborhoodSize = N

If $c < 2$, then

$W = 2 * W$.

If $c > 5$, then

$W = W/2$ // Ensure that W is in the bounds of the Inertial Range option.

If flag = false:

$c = c + 1$.

$N = \min(N + \mathrm{minNeighborhoodSize}, \mathrm{SwarmSize})$.

PSO is initialized with a group of particles and then searches for an optimal candidate solution by updating generations. Each particle is updated by two best values in the iterations. The first one is the best solution that has been achieved previously. The second best value is tracked by the particle swarm optimizer obtained currently by any particle in the population. The bound of the inertial range option is used

Table 1: Parameter settings for simulation.

Parameter	Value
Simulation area	1000 × 1000 m
Simulation time	1000 sec
Number of nodes	50, 100, 150, and 200
Transmission range	200 m
Speed	0–50 m/sec
Movement model	Random waypoint model
Traffic type	CBR/UDP
Packet size	1000 bytes
Rate	250 kb/s
Pause time	500 sec

for providing a satisfactory solution that eventually is discovered. This best value is a global best. The PSO algorithm significantly reduces the traffic overhead and computation complexity. The proposed PSO scheme reduced the route failure between nodes that minimize the routing overhead. To decrease the effect of random error, every experiment repeats 50 times and the average of experimental results is used as the performance metrics.

4. Results and Discussion

The proposed scheme has been implemented in a network simulator (NS2). The main objective of the simulation was to ensure reliability during data transmission in routing. The parameter settings are listed in Table 1. Nodes were randomly deployed in a 1000 m × 1000 m area of interest. The transmission range was 20 m. Nodes followed the random waypoint model that finds the availability of connection paths in a MANET. The performance of the proposed scheme was evaluated by comparing it with the related PSOLP and MPSO schemes in terms of packet delivery ratio, routing overhead, latency, energy consumption, and path optimality. The simulation results were studied by varying the network size from 50 to 200. The proposed scheme has integrated the PSO and continuous time recurrent neural network to enhance energy efficiency and reliability by selecting an optimal path. With the objective of comparing routing performance with related approaches, the proposed scheme has modeled a PSO in terms of functionality for reliable routing.

4.1. Performance Metrics. The proposed scheme uses five performance metrics to evaluate the proposed scheme and related schemes.

Packet Delivery Ratio. It is the ratio of the number of data packets received successfully by the destination node.

Latency. It is the average time taken by the data packets sent from source node to the destination node.

Routing Overhead. The number of control packets was generated during data transmission in routing.

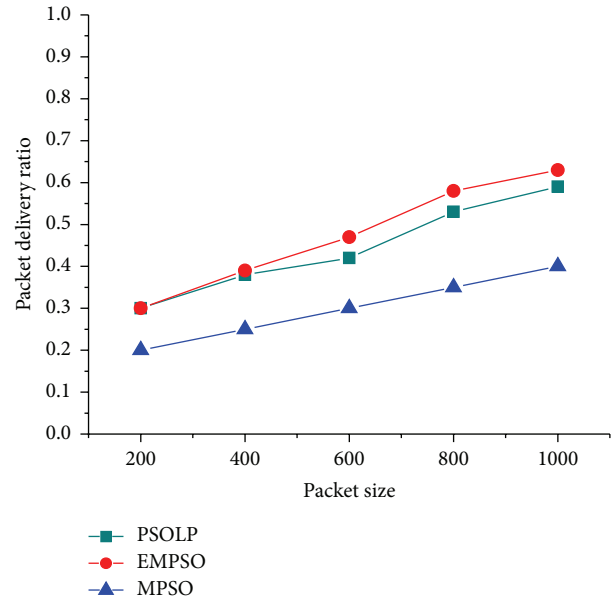


Figure 2: Packet delivery ratio versus packet size.

Path Optimality. It is the ratio of the total number of hops in the shortest paths to the total number of hops in the paths taken by the data packets.

Energy Consumption. It is the average energy consumed for the data transmission in routing.

4.2. Packet Delivery Ratio. In this simulation, the impacts of mobility, packet size, and network size were evaluated while measuring the PDR. It shows that the proposed schemes maintained higher PDR about 63%, while varying packet size increases. An intensive performance evaluation shows that the proposed scheme has better capability of finding an optimal route with the help of PSO approach. Figure 4 shows that the PDR of the proposed schemes gets increased when the number of nodes increases. It shows that the packet delivery ratio increases for the proposed model since it provides the multipath routing when compared with the existing schemes. It was clearly shown that the performance of the proposed scheme is more efficient than the related schemes. Normally, the value of PDR gets increased in the proposed model since it sends the number of data at a time when compared to the related schemes. Figure 3 showed that the PDR increases since it provides multipath routing with the best path. The other two approaches have no effective mechanism to find optimal path for routing. Figure 2 shows that the proposed schemes maintained higher PDR about 90% in mobility scenario. PSOLP and MPSO have a smaller PDR because they have no effective routing mechanism to find optimal path among multiple paths.

4.3. Latency. Figure 5 shows the delays of EMPSO, PSOLP, and MPSO measured from the simulation. It has been seen that the delay occurred for both the techniques when

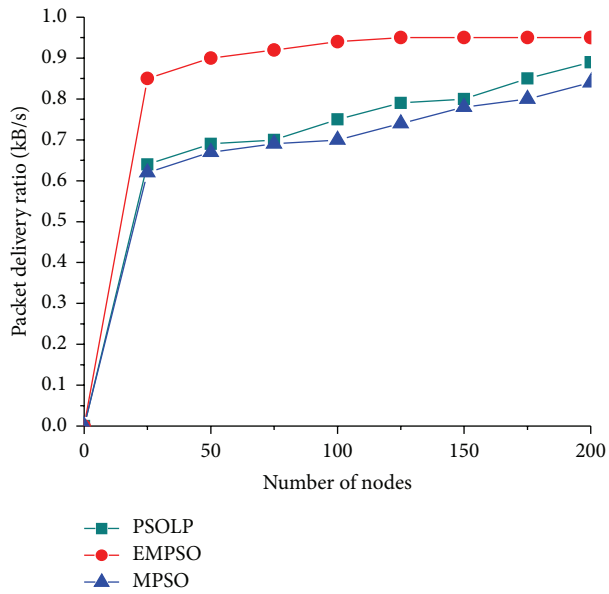


Figure 3: Packet delivery ratio versus number of nodes.

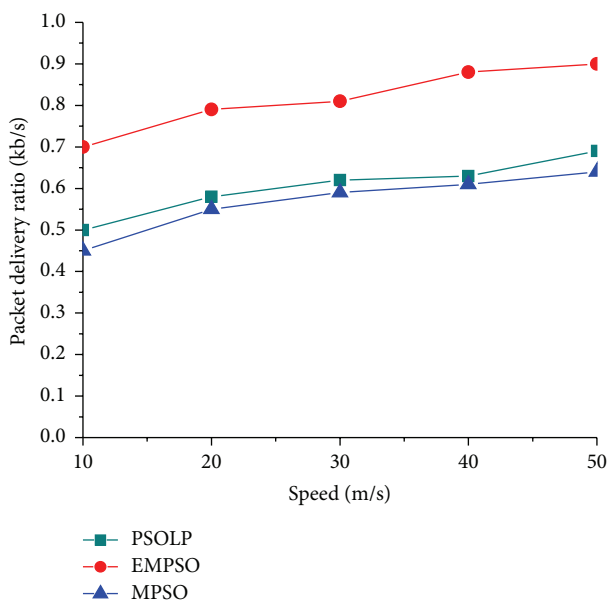


Figure 4: Packet delivery ratio versus speed.

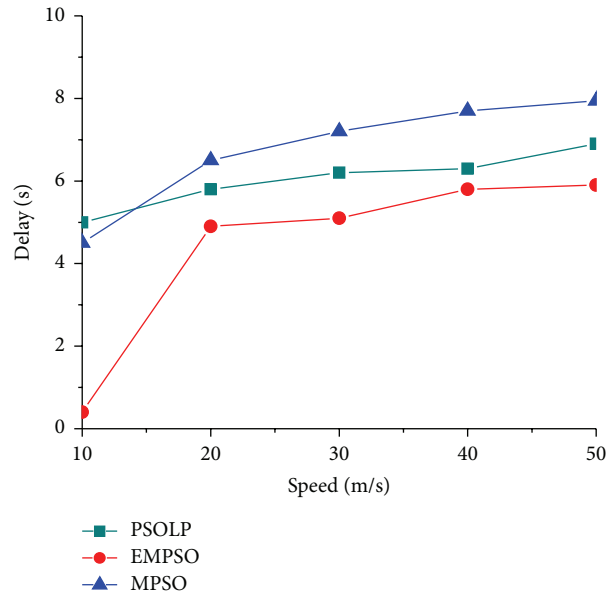


Figure 5: Delay versus speed.

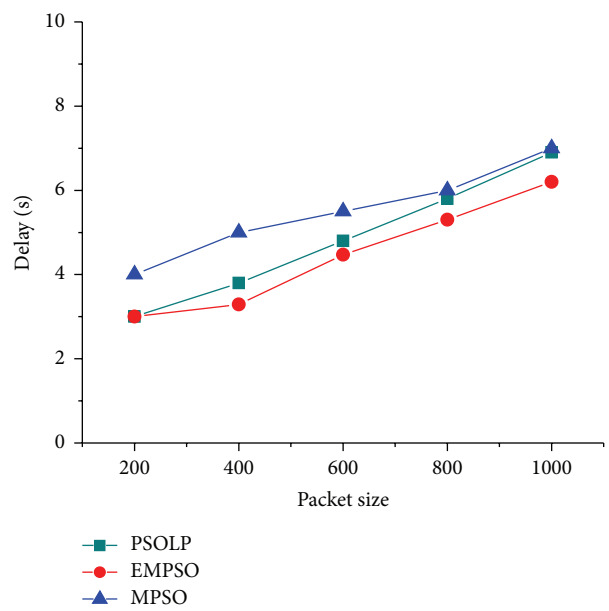


Figure 6: Delay versus packet size.

the speed is increased from 10 to 50 m/s. The delay begins to increase when the speed is increased since the chances of route breakage are more at high speed. Since the proposed EMPSO scheme predicts link quality more accurately than PSOLP and MPSO, the delay is 0.3% less for the related scheme. It was determined that the performance of the proposed scheme is more efficient than the related PSOLP and MPSO schemes. Figure 6 shows the routing latency for three protocols when the number of packets varied. It was observed that the routing latency of the proposed EMPSO is 5.3 sec with multipath scenario on the network and 7 sec and 8.2 sec for PSOLP and MPSO, respectively. The related PSOLP and MPSO schemes took more time to forward packets to their destinations. The delays in both schemes are because of link failures and more exchanges of control messages in the routing. The proposed EMPSO scheme uses reliability measures to form multipath in a network. Therefore, it provides reliable routing and enhances the performance of MANET.

4.4. Routing Overhead. In this simulation, the routing overhead was evaluated for EMPSO, PSOLP, and MPSO while varying the packet size with different stages. Figure 7 depicted the routing overhead that occurred during the routing process. EMPSO has 2.13% lower overhead when compared

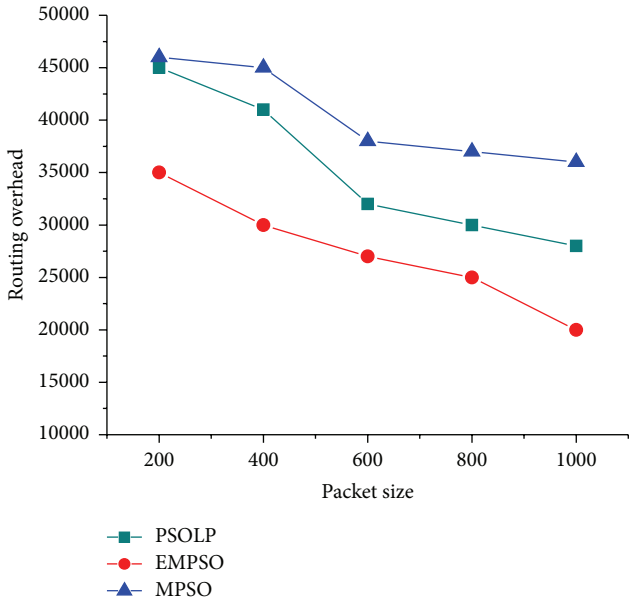


Figure 7: Routing overhead versus packet size.

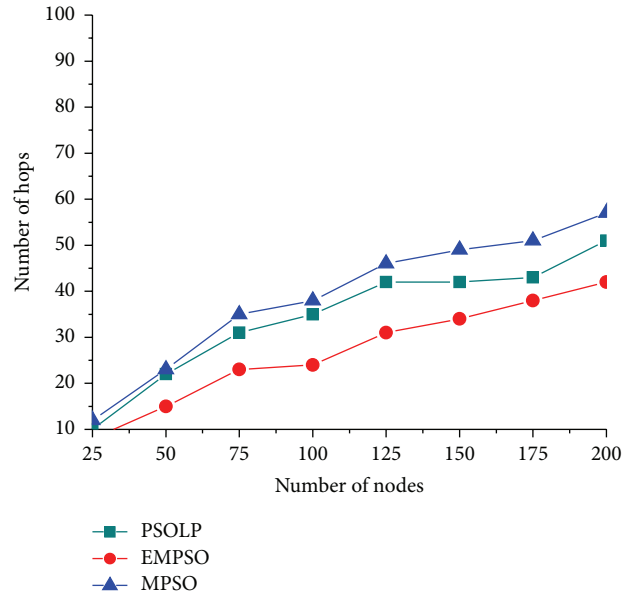


Figure 9: Number of hops versus number of nodes.

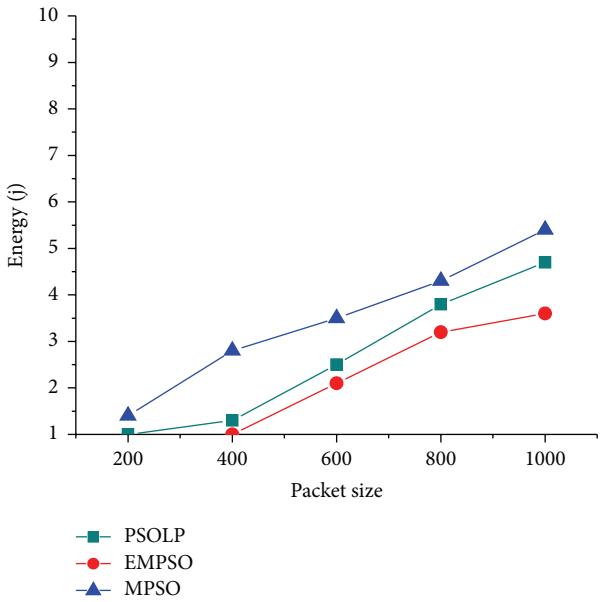


Figure 8: Energy consumption versus packet size.

with PSOLP and MPSO; it reveals in reduced to generate control routing packets and hence alternate routing will be triggered. Therefore, the routing overhead is decreasing. In PSOLP and MPSO, the control packet flooded throughout the network until the destination was reached to find the loop-free routing. It caused the routing overhead. The routing overhead caused by EMPSO performs less than the other two protocols because of its PSO mechanism, a limited exchange of routing control messages among the nodes for the route discovery phase. EMPSO had a minimum routing overhead of approximately 43% when an optimal path of the nodes was selected.

4.5. Energy Consumption. Figure 8 shows the energy consumption for three techniques when the packet size is increased from 250 to 1000 bytes. It is clearly seen that EMPSO has 0.28% less energy consumption when compared to related schemes since it considers reliability measures by CRNN mechanism for finding multipath. Figure 8 shows the total energy consumption of a network while varying packet size. It is noticed that the EMPSO consumes minimum energy about 3.22 j while varying the packet size. PSOLP and MPSO have larger energy consumption because they tend to establish longer routes and to find as many nodes-disjoint routes as possible in a route discovery attempt. The other two related schemes such as PSOLP and MPSO do not consider the energy factor, and also they required more control packets to establish routing that causes more energy consumption. The proposed EMPSO selects the optimal path that reveals efficient solution with less energy consumption. In the proposed EMPSO, the energy consumption of a node increases slightly as the number of nodes also increases.

4.6. Path Optimality. Figure 9 shows the path optimality of the different routing protocols. It is noticed that the EMPSO used minimum hops to forward the packets from source to destination because the EMPSO protocol uses the PSO algorithm for constructing the loop-free routing between source and destination. This algorithm achieves the minimum number of hops due to PSO mechanism. This leads to elimination of faraway nodes. So the amount of hops is minimized. As a consequence of this, the average hop count is reduced in the proposed EMPSO scheme. Thus the proposed EMPSO scheme achieves path optimality compared with related scheme.

5. Conclusion

The prime objective of the proposed EMPSO scheme is to develop energy-aware multipath routing based on particle swarm optimization in mobile ad hoc networks. Development of reliable routing with the use of PSO is conversed. The proposed EMPSO scheme has used a PSO mechanism to reveal the optimal route to minimize the routing overhead and ensure the reliability in MANET. A novel scheme of energy-aware multipath routing based on particle swarm optimization is developed for ensuring reliable routing in MANET. CRNN is used to find multipath among nodes using reliability measures such as transmission cost, energy factor, and optimal traffic ratio. A packet can be forwarded through a selected optimal path. This is a novel way to enhance reliability in routing among nodes. It emphasizes ensuring reliable routing during data transmission when it follows the PSO mechanism. Simulation results show that the proposed EMPSO scheme reveals good results compared with related schemes. It is concluded that the proposed EMPSO based multipath routing can be a potential solution for real time multimedia applications. Future research direction is to use simulated annealing technique to enhance energy efficiency.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] U. Shuchita and G. Charu, "Node disjoint multipath routing considering link and node stability protocol: a characteristic evaluation," *International Journal of Computer Science Issues*, vol. 7, no. 1, article 2, pp. 18–25, 2010.

[2] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad-hoc wireless networks," in *Mobile Computing*, vol. 5, Kluwer Academic Publishers, 1996.

[3] C. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop of Mobile Computing Systems and Applications*, pp. 90–100, August 1999.

[4] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for wireless Ad-Hoc and mesh networks," *Computer Networks*, vol. 56, no. 2, pp. 940–965, 2012.

[5] M. Kaliappan and B. Paramasivan, "Enhancing secure routing in Mobile Ad Hoc Networks using a Dynamic Bayesian Signalling Game model," *Computers & Electrical Engineering*, vol. 41, pp. 301–313, 2015.

[6] J. Boice, J. J. Garcia-Luna-Aceves, and K. Obraczka, "Combining on-demand and opportunistic routing for intermittently connected networks," *Ad Hoc Networks*, vol. 7, no. 1, pp. 201–218, 2009.

[7] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, "Routing protocols in ad hoc networks: a survey," *Computer Networks*, vol. 55, no. 13, pp. 3032–3080, 2011.

[8] J. Yi, A. Adnane, S. David, and B. Parrein, "Multipath optimized link state routing for mobile ad hoc networks," *Ad Hoc Networks*, vol. 9, no. 1, pp. 28–47, 2011.

[9] S. Chakraborty, S. Chakraborty, S. Nandi, and S. Karmakar, "Fault resilience in sensor networks: distributed node-disjoint multi-path multi-sink forwarding," *Journal of Network and Computer Applications*, vol. 57, pp. 85–101, 2015.

[10] M. Tarique, K. E. Tepe, S. Adibi, and S. Erfani, "Survey of multipath routing protocols for mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 32, no. 6, pp. 1125–1143, 2009.

[11] A. K. Daniel, R. Singh, and J. P. Saini, "Swarm intelligence based routing technique for call blocking in heterogeneous mobile Adhoc network using link stability factor and buffering technique for QoS," *International Journal of Research & Reviews in Computer Science*, vol. 2, no. 1, pp. 65–72, 2011.

[12] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[13] J. Alvarez-Benitez, R. M. Everson, and J. E. Fieldsend, "A MOPSO algorithm based exclusively on Pareto dominance concepts," in *Evolutionary Multi-Criterion Optimization*, vol. 3410, pp. 459–473, Springer, Berlin, Germany, 2005.

[14] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft Computing Journal*, vol. 11, no. 4, pp. 3658–3670, 2011.

[15] M. R. AlRashidi and M. E. El-Hawary, "A survey of particle swarm optimization applications in electric power systems," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 913–918, 2009.

[16] S. Sakamoto, T. Oda, M. Ikeda, and L. Barolli, "Design and implementation of a simulation system based on particle swarm optimization for node placement problem in wireless mesh networks," in *Proceedings of the International Conference on Intelligent Networking and Collaborative Systems (INCOS '15)*, pp. 164–168, IEEE, Taipei, Taiwan, September 2015.

[17] S. Waseem, A. K. Farrukh, and B. S. Abdul, "Weighted clustering using comprehensive learning particle swarm optimization for mobile ad hoc networks," *International Journal of Future Generation Communication Networking*, vol. 3, no. 1, pp. 61–70, 2010.

[18] W. Ziqiang, S. Xia, and Z. Dexian, "A PSO-based multicast routing algorithm," in *Proceedings of the 3rd IEEE International Conference on Natural Computation (ICNC '07)*, vol. 4, pp. 664–667, IEEE, Haikou, China, August 2007.

[19] Z. Ali and W. Shahzad, "Critical analysis of swarm intelligence based routing protocols in adhoc and sensor wireless networks," in *Proceedings of the 1st International Conference on Computer Networks and Information Technology (ICCNIT '11)*, pp. 287–292, Abbottabad, Pakistan, July 2011.

[20] B. Paramasivan and M. Kaliappan, "Secure and fair cluster head selection protocol for enhancing security in mobile Ad hoc networks," *The Scientific World Journal*, vol. 2014, Article ID 608984, 6 pages, 2014.

[21] J. Jia, X. Wang, and J. Chen, "A genetic approach on cross-layer optimization for cognitive radio wireless mesh network under SINR model," *Ad Hoc Networks*, vol. 27, pp. 57–67, 2015.

[22] C. Jian, J. Jie, W. Yingyou, Z. Dazhe, and L. Jiren, "A genetic approach to channel assignment for multi-radio multi-channel wireless mesh networks," in *Proceedings of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation (GEC '09)*, pp. 39–46, Shanghai, China, 2009.

[23] J. P. Singh, P. Dutta, and A. Pal, "Delay prediction in mobile ad hoc network using artificial neural network," *Procedia Technology*, vol. 4, pp. 201–206, 2012.

[24] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, 2008.

[25] N. Keerthipriya and R. Latha, "Adaptive cluster formation in MANET using particle swarm optimization," in *Proceedings of the 3rd International Conference on Signal Processing, Communication and Networking (ICSCN '15)*, pp. 1–7, IEEE, Chennai, India, March 2015.

[26] S. T. Kurdi and M. Y. Abdullah, "Weighting for mobile ad hoc networks clustering based on K-HPSO and its application in aircraft networks," *Journal of Babylon University/Pure and Applied Sciences*, vol. 22, no. 3, pp. 958–965, 2014.

[27] W. Shahzad, F. A. Khan, and A. B. Siddiqui, "Clustering in mobile ad hoc networks using comprehensive learning particle swarm optimization (CLPSO)," in *Communication and Networking*, D. S. Ślęzak, T.-H. Kim, A. C.-C. Chang, T. Vasilakos, M. Li, and K. Sakurai, Eds., vol. 56 of *Communications in Computer and Information Science*, pp. 342–349, Springer, Berlin, Germany, 2009.

[28] O. Dengiz, A. Konak, and A. E. Smith, "Connectivity management in mobile ad hoc networks using particle swarm optimization," *Ad Hoc Networks*, vol. 9, no. 7, pp. 1312–1326, 2011.

[29] M. M. Goswami, R. V. Dharaskar, and V. M. Thakare, "Fuzzy ant colony based routing protocol for mobile ad hoc network," in *Proceedings of the International Conference on Computer Engineering and Technology (ICCET '09)*, vol. 2, pp. 438–444, IEEE, Singapore, January 2009.

[30] H. Ali, W. Shahzad, and F. A. Khan, "Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization," *Applied Soft Computing*, vol. 12, no. 7, pp. 1913–1928, 2012.

[31] A. S. Nasab, V. Derhamia, L. M. Khanlib, and A. M. Z. Bidoki, "Energy-aware multicast routing in MANET based on particle swarm optimization," *Procedia Technology*, vol. 1, pp. 434–438, 2012.

[32] D. Manickavelu and R. U. Vaidyanathan, "Particle swarm optimization (PSO)-based node and link lifetime prediction algorithm for route recovery in MANET," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, article 107, pp. 1–10, 2014.

[33] Y. Yun-Sheng, C. Han-Chieh, C. Ruay-Shiung, and V. Athanasios, "Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs," *Mathametical and Computation Model*, vol. 53, no. 11-12, pp. 2238–2250, 2011.

[34] M. Radi, B. Dezfouli, K. Abu Bakar, S. Abd Razak, and M. A. Nematbakhsh, "Interference-aware multipath routing protocol for QoS improvement in event-driven wireless sensor networks," *Tsinghua Science and Technology*, vol. 16, no. 5, pp. 475–490, 2011.

[35] P. Hurni and T. Braun, "Energy-efficient multi-path routing in wireless sensor networks," in *Proceedings of the 7th International Conference on Ad-Hoc, Mobile and Wireless Networks*, pp. 72–85, Sophia Antipolis, France, 2008.

[36] M. Ghiasi and A. Karimi, "Fault diagnosis method for mobile Ad-hoc network by using smart neural networks," *Procedia Computer Science*, vol. 42, pp. 222–227, 2014.