

Non-dominated Sorting Genetic Algorithm-II for Throwbox Deployment in Delay Tolerant Networks



C.Poongodi, K.Lalitha, A.Jeevanantham, D Vijay Anand

Abstract— Delay-tolerant networks (DTNs) are basically opportunistic networks and uses store-carry-and-forward switching for message forwarding. Performance of DTNs can be improved by placing stationary nodes, called throwboxes which increases the contact opportunities among nodes. Throwbox is viewed as relaying node and the message delivery is enhanced by spreading as many messages as possible. Increasing the contact opportunities in throwbox based DTN model depends on the deployment of throwboxes in suitable places. The objective of this paper is to identify optimal deployment locations of the throwboxes with a pre-specified transmission range, and to maximize the coverage of all the throwboxes in order to increase average delivery and to reduce the average delay among all the nodes in the network. We use Non-Dominated Sorting Genetic Algorithm-II for optimizing the deployment of throwboxes in DTN. The simulation results are analyzed for better strategy in deploying throwboxes and to improve the performance of throwbox-augmented DTNs.

Keywords : Delay Tolerant Network, Optimal Deployment, Message Forwarding, Throwbox.

I. INTRODUCTION

The DTN are characterized as challenged networks where there is no end-to-end between a data source and its peer(s), excessive maximum round-trip time between any pair of nodes in the network and high end-to-end packet drop probability (Fall,2003). It leads to disconnected partitions in the DTN. Unfortunately, these networks may not be well served by the existing end-to-end TCP/IP model. DTNs rely on the intrinsic mobility of the nodes to deliver packets around the disconnected partitions of the network using a store-carry-and-forward paradigm. This method used to forward information in completely unconnected and

decentralized scenarios like sparse sensor deployments, natural disasters, underwater communications, and highly mobile systems.

A major complexity in DTN is that contact opportunities between nodes cannot be predicted nor modified. An approach proposed in the literature consists of employing some additional mobile nodes, called data mules(DM) or message ferry(MF) nodes, their movement trajectories are altered for routing (Poongodi et al 2010, Zhao et al 2004 and Zhao et al 2005). An alternative approach proposed is in equipping the network with dedicated stationary nodes, called throwboxes (Ibrahim et al 2007). Throwboxes are inexpensive, battery-powered nodes with long radios and high storage. When many nodes pass by the same location at different times, the throwbox acts as an intermediate router among them and creates a new contact opportunity. When a mobile node enters into the transmission range of a throwbox, it will drop the messages to forward or receive data from it. Throwboxes are located at some planned geographical positions to improve the number of contact opportunities among the mobile nodes. Throwboxes are typically low cost devices when compared with Ferry nodes and can be deployed in huge numbers, hence allowing for a better flexibility in the network design.

Deploying throw-boxes as static nodes increases the DTN performance in message dissemination among mobile nodes. Throwboxes are operated without communicating with other throwboxes. The real time deployments and simulation results in (Lloyd et al 2007, Gu et al 2009, Gu et al 2011, Zhiyuan et al 2017) have demonstrated that deploying few active throwboxes can indeed improve the network performances and overall throughputs. This method is especially helpful for localized mobility where mobile nodes may only move within a region of the network, and the entire network lacks in global mobility of most nodes. In this network scenario, communicating through pure encountering of mobile nodes may not be appropriate due to the reduced chances of contact. Due to the mobile nodes, the network topology evolves over time in DTN. These new challenges make existing relay node placement algorithms are not suitable for DTNs. Careful deployment of the throwboxes can increase the network connectivity and improves the DTN network performances.

Manuscript published on November 30, 2019.

* Correspondence Author

Dr.C.Poongodi*, Department of Information Technology, Kongu Engineering College, Erode, Tamilnadu, India. Email: poongs.it@kongu.edu

Dr.K.Lalitha, Department of Information Technology, Kongu Engineering College, Erode, Tamilnadu, India. Email: klalitha.it@kongu.edu

Mr.A.Jeevanantham, Department of Information Technology, Kongu Engineering College, Erode, Tamilnadu, India. Email: jeeva.it@kongu.edu

Mr.D.Vijay Anand, Department of Information Technology, Kongu Engineering College, Erode, Tamilnadu, India. Email: erovijay.it@kongu.edu

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

II. RELATED WORKS

Shah et al (2003), Tariq et al (2006), Zhao et al (2004), and Xian et al (2010) proposed the Message Ferry approach in which it delivers data in DTN by utilizing non-randomness of mobile nodes. Nodes will not move randomly always. All the nodes are aware of some regular node like a MF node which follows a routine path. The use of multiple ferries approach discussed in Zhao et al (2005) may be essential in all the situations to increase connectivity. It uses multiple ferries in networks with Throwboxes, focusing on the design of ferry routes. Chen et al (2007) discussed the task of using small number of Unmanned Aerial Vehicles (UAVs) to relay messages between source and destination of ground nodes. In latency-insensitive bulk data transfer of DTNs, it seeks to increase throughput by having a UAV load from a source ground node, carry the data while moving towards the destination, and finally transfers the data to a destination ground node.

The papers (Zhao et al 2006, Li et al 2016) studied about throwbox deployment problem of selecting or removing throwboxes in a predictable and time-evolving DTN for guaranteed network reliability. In (Banerjee et al 2007), An energy-efficient architecture for throwbox-assisted DTNs is proposed. An approximate heuristic is given for computing the NP-hard problem at meeting places of throwboxes and an average power constraint while maximizing delivery ratio. Energy optimization is performed with individual throwboxes. In (Fan et al 2016) discussed about Throwbox assisted DTNs where throwbox deployment and also routing optimization problem are combined.

A greedy algorithm is discussed which depends on network flow technique to find multiple linear programming problems. The study only focuses on the possible maximum data rate between any two nodes in long term. In (Huang et al 2013, Li et al 2015) they have studied topology control (TC) problem for DTNs, which aims to build a sparse space-time graph while assuring the reliability requirement or connectivity over time. In throwbox optimization problem, if a throwbox is triggered, all its spatial and temporal connections over any time slots are activated. Deploying minimum number of throwboxes and k number of throwboxes problems are solved using greedy algorithm for time-evolving DTNs with the assistance of weighted space-time graphs.

In this proposed scheme, we study about how the throwboxes can be placed effectively for improving the performance of DTN. While deploying the throwboxes the places where the maximum coverage is possible is to be found. Those places should also be useful in reducing the maximum delay in reaching from one source to all other destinations. A single objective scheme cannot be applied since a throwbox should support more number of nodes as well as it should improve the other performance criteria. Based on these objectives, multi-objective optimization algorithms NSGA-II are applied and the results are compared. [Yunjia Yang et al 2019, Jing Liang et al 2019, Ali Hojjati et al 2018] discusses about how multi-objective optimization algorithms like NSGA can be effectively applied for real time multi-objective problems.

III. NSGA – II FOR THROWBOX DEPLOYMENT

DTN environment is unreliable and constantly changes due to their mobility of nodes. However, we can use these same node movements along with Throwboxes to transmit messages in the network. In this section, we present the problem of finding the best locations for Throwboxes in the network to reach from one or more sources to a destination based on the availability probabilities of the links. In our model, we assume that each message is assigned infinite TTL. Hence, nodes and throwboxes can hold any amount of messages for any duration. Practically, nodes and throwboxes may have limited capacity to hold only certain amount of messages and have limited energy resource. Spatio Temporal Parametric Stepping (STEPS) is the mobility model attached to each node. It makes abstraction of spatio-temporal preferences in human mobility preferred for DTN environment (Nguyen 2011).

NSGA-II is a simple and effective algorithm to implement compared to other evolutionary algorithms like PSO and ACO. In this section, we discuss about implementation of the various parts of the general NSGA-II optimization algorithms for the deployment problem in DTNs. We need to define appropriate internal data structure to represent chromosomes and their reproductions for this deployment problem. NSGA-II is the second version of the famous “Non-dominated Sorting Genetic Algorithm” which is based on the work for solving single and multi-objective optimization problems (Deb 2001). Its main features are:

- A non-dominated sorting algorithm where all the individuals are ordered according to the level of non-domination
- It implements elitism property in which it stores all non-dominated solutions and improves convergence properties
- It adapts a crowding distance mechanism to guarantee diversity and spreading of solutions
- Constraints are evaluated using a modified definition of dominance and without the use of penalty functions

Simple NSGA methodology suffers from three weaknesses: computational complexity, non-elitist approach and the need to specify a sharing parameter. NSGA-II which is improved version of NSGA, which resolved the above problems and uses elitism to create a diverse Pareto-optimal front, is subsequently presented in (Deb 2001). NSGA-II possess low computational complexity, elitism parameter less diversity preservation, and real valued representation and are being applied to real time applications for optimization. (Ali Hojjati et al 2018). NSGA-II equipped with elitism for multi-objective search, using an elitism-preserving approach. Elitism is established by storing all non-dominated solutions discovered so far, starting from the initial population.

Elitism improves the convergence properties of the Pareto-optimal set and a parameter-less diversity preservation mechanism is also adopted. Diversity and spreading of solutions are assured without the use of sharing parameters, since NSGA-II adopts a appropriate parameter-less niching approach.

The crowding distance estimates the density of solutions in the objective space and the crowded comparison operator guides the selection process towards a uniformly spread Pareto-frontier (Basu 2008). The flowchart and pseudo code for the NSGA II are shown in Figure 1 and 2 respectively.

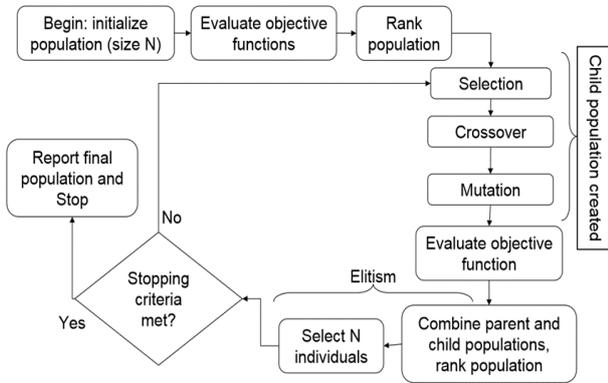


Figure 1. Flowchart for NSGA-II

Algorithm for NSGA II

```

1: Input: N, The population size and CP Random Population for a network area [X, Y]
2: Q->Offspring Population
3: while stop_criteria_not_satisfied() do
4:   for i<-1 to N do
5:     Select two parents CP1 and CP2 from CP
6:     Execute Crossover of CP1 and CP2 and create offsprings CQ1 and CQ2
7:     Mutate CQ1 and CQ2
8:     Evaluate CQ1 and CQ2 and insert them in Q
9:   end for
10:  Create a union set Z ← CP ∪ Q, CP ← ∅
11:  pfs ← EvaluateFitness (Z)
12:  pfs[0] ← pfs
13:  while length(CP) < N do
14:    Sort solutions from the current front by Crowding distance
15:    for each solution(i) in ordered(pfs) do
16:      if length(CP) < N then
17:        CP ← solution(i)
18:      end if
19:    end for
20:    Go to next Pareto Front
21:  end while
22: end while
23: end
  
```

Figure 2. Pseudo Code for NSGA-II

IV. SIMULATION ANALYSIS & RESULTS DISCUSSION

Proposed NSGA-II algorithm is implemented in MATLAB 2013a. For the sample 100 node network problem and network parameters as shown in Table 1. This Pareto-optimal set was obtained by taking only the two objectives Objective 1 as maximizing average node degree and Objective 2 as minimizing average delivery delay and Objective 3 as maximizing average delivery ratio. Further the algorithm was implemented on various network topologies and various network sizes to test the performance in finding the Pareto-optimal solutions. After performing 50 simulation runs, the best results achieved by NSGA-II are recorded.

Figure 3 shows the simulation setup for DTN environment with mobile nodes in blue and throwboxes in red.

Table 1: Simulation Parameters

Parameter(s)	Value(s)
No of nodes	100
Area	1000mx1000m
No of Throwboxes	20
Communication Range	100m
Simulation time(s)	10000
Mobility Model	STEPS Mobility
Crossover Rate(Cr)	0.6
Mutation Factor(F)	0.5

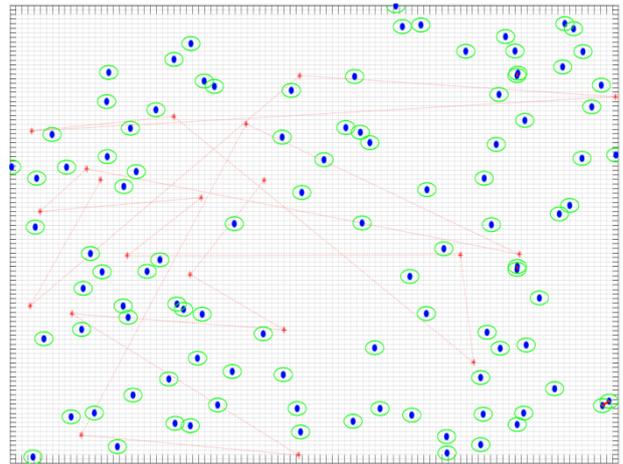


Figure 3. Simulation Environment in MATLAB

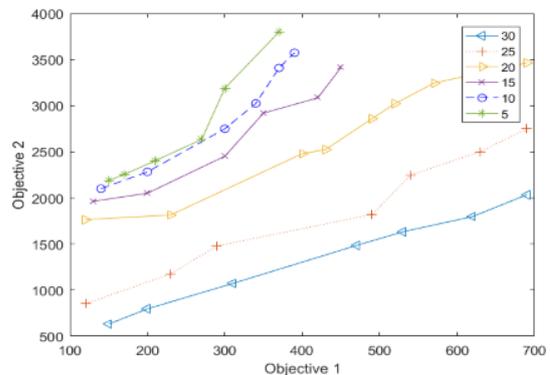


Figure 4. Pareto Fronts obtained for Objective 1(Node Degree) Vs Objective 2 (Delivery Delay)

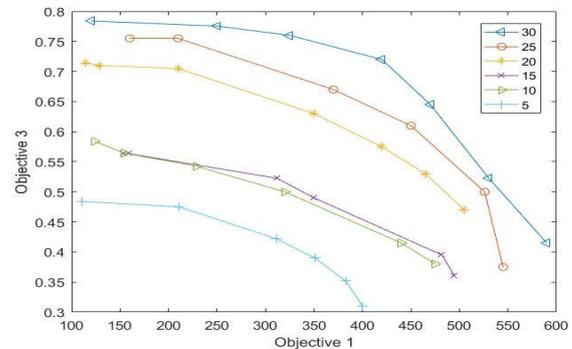


Figure 5. Pareto Fronts obtained for Objective 1 (Node Degree) Vs Objective 3 (Delivery Ratio)

For number of generations 200 and above, the algorithm converged to a Pareto set which are not much different from number of generations 100. It is because of the solution space is small. Hence it is not needed to run as like other Multi-objective Optimization Problems (MOPs) for more than 100 generations and for the given Table 1 simulation parameters. A pattern with clearness appears in the given scenario shows noticeable move in the pareto frontages, presenting correlation in the middle of the quantity of throwboxes, the functional values of phenomenon acquired for node degree, delay as well as ratio. In common, when the throwboxes count get increases, the delivery ratio also increases on an average. On the other hand, the above said results achieved with increase in node degree which was clearly depicted in sample pareto front ages acquired for NSGA II with throwboxes count of 15. Figure 3 and 4 shows the performance comparison for different two objectives case. Observe in Figure 3 that for five numbers of throwboxes, the lowest delay is below 2500ms, as well as numeral elucidations are created given that above only 30% of node degree. As and when the operational units count gets greater than before from 20 to 30, as in Figure 3, the lowest delay is down to below 500ms, but very few solutions are found with over 70% of node degree.

A gradual growth in the count of throwboxes when added count of node degrees and in turn more contacts between nodes, generating virtual infrastructure in which nodes remain get connected. It is quintessential to contemplate when the arrangement team stands matching results with diverse amount of throwboxes. Expecting a better solution with more amount of throwboxes is a negative analogy to reduce delay when associated to one with fewer operational units. It depends on the optimized locations where they are placed. As and when throwboxes grow upwards, a movement of decreased consistency in relations of Pareto fronts attained which is shown in figure 2. A clear gap emerges between the Pareto fronts obtained as the number of throwboxes is increased, indicating a greater variation in performance over the 100 runs. In the middle of Pareto frontiers and throwboxes, a noticeable hole developed when the count gets increased which indicates an inordinate difference over 100 runs. Due to better results obtained at each run, this may likely happened.

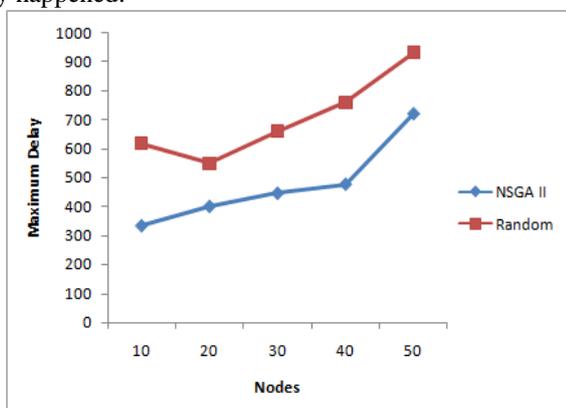


Figure 6. Comparison of Delay for NSGA II Vs Random Deployment

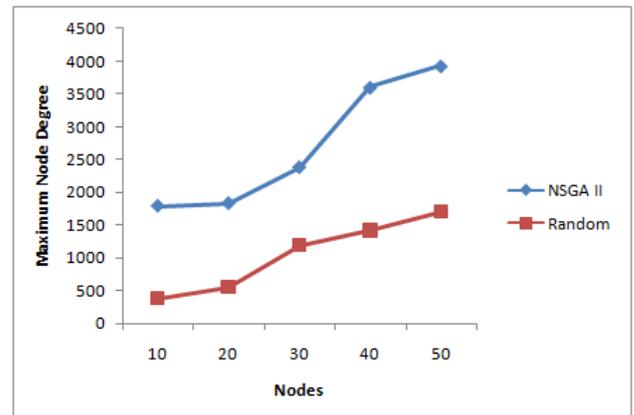


Figure 7. Comparison of Node Degree for NSGA II Vs Random Deployment

The Pareto Fronts obtained for all the three objectives of NSGA II are deployed in DTN environment and the results are compared with Random deployment. The Figure 7 shows the results of maximum node degree by all the throwboxes while the node count gets increased. The obtained result remains compared with the random deployment of throwboxes. Clearly it shows that the proposed deployment of throwboxes using NSGA II algorithm performs better in covering maximum number of nodes when placing the throwboxes at optimum number of places. Similarly Figure 6 compares the results of maximum delay in reaching all the destination from one source. It also shows better results for proposed algorithm rather than random deployment.

V. CONCLUSION

An innovative throwbox deployment approach is projected in this work mainly for DTNs. Compared to the present throwbox positioning strategies, for instance the greedy algorithm based deployment and the other contact-based deployment, multi-objective optimization is most formidable one that would heighten the fault-performance of the discussed area. A self-adaptive NSGA-II which will identify better range of alternative solutions and also better communication near the original Pareto-frontiers, are proposed for solution optimization. The results, associated with all statistics acquired with different quality indicators, indicate the supremacy of NSGA-II in terms of convergence speed as well as quality of final Pareto front. It also provides an alternative means to solve MOPs. In future, the dissemination in objective space to be improved further and also to resolve multimodal objective issues for better performing DTNs

REFERENCES

1. Vahdat A., and Becker D., "Epidemic Routing for Partially-Connected Ad Hoc Networks," Duke University Technical Report CS-20000, 2000.
2. Fall, K., "A delay-tolerant network architecture for challenged internets," Proceedings of ACM SIGCOMM 03, 2003, pp. 27-34.
3. Ibrahim.M, Hanbali A. Al, and Nain P., "Delay and resource analysis in MANETs in presence of throwboxes," Performance Evaluation, 9th ed. Vol. 64, 2007, pp.933-947.

4. Poongodi, C., Natarajan, A. M and Suganthee, R. C, "Cluster Head Bundle Delivery Protocol for Delay Tolerant Networks," Proceedings of CiiT International Journal of Wireless Communication, doi: WC112010005, 2010.
5. Spyropoulos T., Psounis K., and Raghavendra C.S., "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN'05), 2005, pp. 252–259.
6. Spyropoulos T., Psounis K., and Raghavendra C.S., "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," In Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshop, 2007, pp. 79 – 85.
7. Zhao, W., Ammar M., and Zegura, E., "A message ferrying approach for data delivery in sparse mobile ad hoc networks," Proceedings of ACM MobiHoc, (New York, NY, USA), ACM Press, 2004, pp. 187–198.
8. Zhao, W., Ammar, M., and Zegura E., "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," Proceedings of IEEE INFOCOM, vol. 2, 2005, pp. 1407–1418.
9. Lloyd E. L. and Xue G., "Relay node placement in wireless sensor networks," IEEE Transactions on Computers, 1st ed. Vol.56, 2007, pp.134–138.
10. Gu B. and Hong X., "Latency analysis for thrown box based message dissemination," Proceedings of IEEE GLOBECOM, 2010, pp.1 – 5.
11. Gu B. and Hong X., "Capacity-aware routing using throw-boxes," Proceedings of IEEE GLOBECOM, 2011, pp.1 – 5.
12. Li F. , Yin Z., Cheng Y. and Wang Y., "Optimization Problems in Throwbox-Assisted Delay Tolerant Networks: Which Throwboxes to Activate? How Many Active Ones I Need?" IEEE Transactions on Computers, 5th ed. Vol. 65, 2016.
13. Zhao W., Chen Y., Ammar M., Corner M., Levine B., Zegura E., "Capacity enhancement using throwboxes in DTNs," Proceedings of IEEE MASS, 2006.
14. Ibrahim M., Nain P., and Carreras I., "Analysis of relay protocols for throwbox-equipped DTNs," Proceedings of WiOPT, 2009, pp. 1 – 9.
15. Shah, R. C., Roy, S, Jain, S., and Brunette, W., "Data mules: modeling a three-tier architecture for sparse sensor networks," in Proceedings of Sensor Network Protocols and Applications, 2003, pp. 30–41.
16. Bin Tariq M.M, Ammar M., and Zegura E., "Message ferry route design for sparse ad hoc networks with mobile nodes," Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '06), Florence, Italy, 2006, pp. 37–48.
17. Xian Y, Huang C.T., and Cobb J., "Look-Ahead Routing and Message Scheduling in Delay-Tolerant Networks," Proceedings. of 35th IEEE Conference on Local Computer Networks, Denver, Colorado, 2010.
18. Cheng C.M, Hsiao P.H., Kung HT, and Vlah D., "Maximizing Throughput of UAV-Relaying Networks with the Load-Carry-and-Deliver Paradigm," Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 2007.
19. Maurice, J. K., Wissam, F. F., and Chadi, M. A., "Probabilistic Bundle Relaying Schemes in Two-Hop Vehicular Delay Tolerant Networks," IEEE Communications Letter, 3rd ed. Vol. 15, 2011.
20. Banerjee N., Corner M.D., and Levine B. N., "An Energy-Efficient Architecture for DTN Throwboxes", Proceedings of the IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, pp.776-784.
21. Huang M., Chen S., Zhu Y., and Wang Y., "Topology control for timeevolving and predictable delay-tolerant networks," IEEE Transactions on Computers, 11th ed. Vol.62, 2013, pp.2308–2321.
22. Li F., Chen S., Huang M., Yin Z., Zhang C., and Wang Y., "Reliable topology design in time-evolving delay-tolerant networks with unreliable links," IEEE Transactions on Mobile Computing, 6th ed. Vol.14, 2015, pp.1301–1314.
23. Nguyen A. D., Senac P., Ramiro V., Diaz M., "STEPS - an approach for human mobility modeling," Proceedings of the 10th international IFIP TC 6 conference on Networking, 2011, pp.09-13.
24. Storn, R. and Price, K., "Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, Kluwer Academic Publishers, Vol. 11, 1997, pp. 341 – 359.
25. Deb K., "Multiobjective Optimization Using Evolutionary Algorithms", Chichester, U.K., Wiley, 2001.
26. Adeyemo, J.A., Otieno, "F.A.O.: Multi-objective differential evolution algorithm for solving engineering problems", Journal of Applied Sciences 9th ed. Vol.20, 2009, pp.3652–3661.
27. Yunjia Yang et al, "A Modified Multi objective Self-Adaptive Differential Evolution Algorithm and Its Application on Optimization Design of the Nuclear Power System", Science and Technology of Nuclear Installations, 2019, https://doi.org/10.1155/2019/1041486.
28. Jing Liang et al, "Multi modal multi objective optimization with differential evolution", Swarm and Evolutionary Computation, Vol. 44, 2019, pp.1028-1059.
29. Ali Hojjati et al, "Application and comparison of NSGA-II and MOPSO in multi-objective optimization of water resources systems", 3rd ed. Vol.66, 2018, pp. 323–329

AUTHORS PROFILE



Dr.C. Poongodi, is an Associate Professor of Department of Information Technology in Kongu Engineering College, Perundurai. She has obtained her B.E in Electronics and Communication Engineering from Bharathiar University, Coimbatore in 2001, M.E in Computer Science Engineering from Bharathiar University in 2002 and obtained Ph.D in Information and Communication Engineering from Anna University, Chennai during 2013. She has published twenty papers in International conferences and ten papers in International journals. Her research areas include Wireless networks, Mobile Ad-hoc Networks and Delay Tolerant Networks, Embedded Systems. Mail-id: poongs.it@kongu.edu



K.Lalitha, received B.Tech IT from CSI College of Engineering, The Nilgris in 2005 and M.Tech IT from Anna University of Technology in 2009. She completed PhD in 2019 in Wireless Sensor Networks and working as an Assistant Professor in the Department of IT, Kongu Engineering College, Erode. She has conducted various workshops and published several papers in the area of Wireless Sensor Networks, Mobile Computing, IoT and Network Security. Her research area of interest includes Wireless Sensor Networks, Mobile Computing and Internet of Things. She is a Lifetime member of CSI and IAENG. Email:vrklalitha24@gmail.com



Jeevanantham Arumugam, has received his M.S Degree in Computer and Network Security from the Middlesex University, London, UK and received his B.E Degree in Computer Science and Engineering from Anna University, Chennai, India in 2010. He is currently working as an Assistant Professor with the Department of Information Technology, Kongu Engineering College, Erode, TN, India. His research interests include Cryptography and IoT.



D. Vijay Anand, received M.C.A from Bharath Institute of Science and Technology, Chennai in 2002 and M.E CSE from Kongu Engineering College, Perundurai in 2005. He is working as an Assistant Professor (Sr.G) in the Department of IT, Kongu Engineering College, Perundurai, Erode. He has conducted various workshops and published several papers in the area of Data Mining and Networks. His research area of interest includes Data Mining, Networks and Internet of Things. He is a Lifetime member of CSI. Email:erovijay@gmail.com