Green Energy and Technology

Tutut Herawan
Haruna Chiroma
Jemal H. Abawajy *Editors*

Advances on Computational Intelligence in Energy

The Applications of Nature-Inspired Metaheuristic Algorithms in Energy

Springer

Green Energy and Technology

Climate change, environmental impact and the limited natural resources urge scientific research and novel technical solutions. The monograph series Green Energy and Technology serves as a publishing platform for scientific and technological approaches to “green”—i.e. environmentally friendly and sustainable—technologies. While a focus lies on energy and power supply, it also covers “green” solutions in industrial engineering and engineering design. Green Energy and Technology addresses researchers, advanced students, technical consultants as well as decision makers in industries and politics. Hence, the level of presentation spans from instructional to highly technical. **Indexed in Scopus**.

More information about this series at http://www.springer.com/series/8059

Tutut Herawan · Haruna Chiroma ·
Jemal H. Abawajy
Editors

Advances on Computational Intelligence in Energy

The Applications of Nature-Inspired Metaheuristic Algorithms in Energy

Springer

Editors

Tutut Herawan
University of Malaya
Kuala Lumpur, Malaysia

Haruna Chiroma
Federal College of Education (Technical)
Gombe, Nigeria

Jemal H. Abawajy
Deakin University
Geelong, VIC, Australia

ISSN 1865-3529 ISSN 1865-3537 (electronic)
Green Energy and Technology
ISBN 978-3-319-69888-5 ISBN 978-3-319-69889-2 (eBook)
https://doi.org/10.1007/978-3-319-69889-2

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Nature-Inspired metaheuristic algorithms are algorithms inspired from nature ranging from a natural biological system such as the human brain, animal behaviour and plants; human activities including but not limited to football and interior decoration; natural phenomenon such as waterfall, gravitational force and others including chemistry and physics. These algorithms are known for their efficiency and effectiveness in solving optimization problems across different domain of applications. This makes these algorithms not to be problem dependent.

In previous years, statistical and mathematical methods are the dominant methods in solving problems in energy including predictions for energy management, but the present uncertainty nature of the energy makes the conventional methods unfit for the task. Nature-inspired metaheuristic algorithms such as neural networks, genetic algorithm and more recently cuckoo search, artificial bee colony, firefly and hybrid systems have gained an unprecedented attention in the literature because of their ability to model any nonlinear problem more than the conventional methods. Presently, metaheuristic algorithms are very active in the literature for solving problems in the energy domain.

This book summarizes contributions from experts in the research area. The book can serve as a platform for dissemination and sharing of the latest scientific development for nature-inspired metaheuristic algorithms in energy that present state-of-the-art interesting topics and a consolidated expert view ready for practical applications to the interested readers as well as industry experts. The book is covered by ten chapters dedicated to significant contributions on the applications of metaheuristic algorithm in energy issues as follows:

Chapter “A Theoretical Framework for Big Data Analytics Based on Computational Intelligent Algorithms with the Potential to Reduce Energy Consumption”, by Haruna Chiroma, Usman Ali Abdullahi, Ibrahim Abaker Targio Hashem, Younes Saadi, Rawaa Dawoud Al-Dabbagh, Muhammad Murtala Ahmad, Dada Emmanuel Gbenga, Sani Danjuma, Jafaar Zubairu Maitama, Adamu Abubakar and Shafi’i Muhammad Abdulhamid, discusses theoretical work involving computational intelligence algorithm in big data analytics. The chapter proposes a theoretical framework for big data analytics using computational intelligent algorithms that have the potential to reduce the consumption of energy and advanced performance for big data analytics.

Chapter "Artificial Bee Colony for Minimizing the Energy Consumption in Mobile Ad Hoc Network", by Mustafa T. Abd, Saad Adnan Abed and Elankovan A. Sundararaja, discusses the searching behaviour of the artificial bee colony (ABC) motivated the authors to apply the algorithm for finding the possible optimal route from the source to the destination. As such, minimize the average energy consumption for the routes selected by the ABC algorithm. Evaluation result indicated that the proposed ABC was found to improve the energy efficiency of the network.

Chapter "A Novel Chicken Swarm Neural Network Model for Crude Oil Price Prediction" by Abdullah Khan, Rahmat Shah, Junaid Bukhari, Nasreen Akhter, Attaullah, Muhammad Idrees, and Hilal Ahmad, discusses the extraction of chicken swarm intelligence of the chicken smarm optimization to train an artificial neural network for building a model for the prediction of crude oil price. The chicken swarm based artificial neural network is applied to predict crude oil price, and it was found to be better than the comparison algorithms.

Chapter "Forecasting OPEC Electricity Generation Based on Elman Network Trained by Cuckoo Search Algorithm", by Abdullah Khan, Rahmat Shah, Nasreen Akhter, Awais Qureshi, Kamran Ullah, Shah Zaman, Hilal Ahmad and Muhammad Idrees, discusses the hybridization of Elman network by training it using cuckoo search algorithm to build a model for the prediction of OPEC electricity generation. Experimental results have shown that the proposed algorithm performs better than classical algorithms.

Chapter "Variable Neighborhood Search-Based Symbiotic Organisms Search Algorithm for Energy-Efficient Scheduling of Virtual Machine in Cloud Data Center", by Mohammed Abdullahi, Shafi'i Muhammad Abdulhamid, Salihu Idi Dishing and Mohammed Joda Usman, proposes modified symbiotic organisms search (SOS) algorithm to minimize the number of the active server by consolidation virtual machines on few servers for energy savings. The SOS mimics mutualism, commensalism and parasitism forms of relationship for traversing the search space. The SOS is hybridized with variable neighbourhood search (SOS-VNS), and it is applied to various workload instance with varying number of virtual machines in a simulated IaaS cloud. The proposed algorithm is found to efficiently minimize energy consumption and improving resource utilization. The results obtained showed that SOS-VNS outperforms the heuristics algorithms.

Chapter "Energy Savings in Heterogeneous Networks with Self Organizing Backhauling", by Nasir Faruk, Abdulkarim A. Oloyede, Abubakar Abdulkarim, Lukman A. Olawoyin and Yinusa A. Adediran, discusses the comparison of conventional microwave backhauls and self-backhauling for typical dense and spare environments in which the heterogeneous network backhaul-energy model is used for the investigation of their energy efficiency. The chapter presents a break-even power point and the load threshold level for safe operating regions towards achieving optimum utilization of self-backhauling in a way for higher energy-efficient and sustainable networks compared to traditional homogeneous macro-network deployments.

Chapter “Integrated Resource Allocation Model for Cloud and Fog Computing: Toward Energy-Efficient Infrastructure as a Service (IaaS)”, by Mohammed Joda Usman, Abdul Samad Ismail, Hassan Chizari, Abdulsalam Ya’u Gital, Haruna Chiroma, Mohammed Abdullahi and Ahmed Aliyu, proposes the integration of the resource allocation model for energy-efficient Infrastructure-as-a-Service (IaaS). It extends from the network edge of the fog to the cloud data centre. The propose architecture have a policy on the fog end where a decision can be made to either process the user request on the fog or move to the cloud data centre. The new architecture can improve resource utilization and the reduction of energy consumption by a data centre. Therefore, improves energy efficiency and optimum resource utilization.

Chapter “Energy-Efficient Bias-Based User Association for Heterogeneous Networks in LTE-Advanced” by Ayuba K. Danburam, Mohammed A. Gadam, Aliyu D. Usman and Suleiman M. Sani, proposes an adaptive BRSRP cell selection algorithm that uses energy efficiency as cell load metric for adaptive pico-cell range extension (CRE). The algorithm was found to effectively estimate the varying energy efficiency in each of the cells. Subsequently, based on the optimality gap of the energy efficiency a corresponding bias value is configured per cell. The simulation results show that the proposed adaptive bias setting improves energy efficiency, average UE throughput and system capacity compared to the classical methods.

Chapter “Energy Efficiency of Backhauling Options for Future Heterogeneous Networks”, by Nasir Faruk, Abubakar Abdulkarim, Nazmat T. Surajudeen-Bakinde, and Segun I. Popoola discusses various backhauling options for future HetNet. For a HetNet system of seven SBS scenario, the efficiencies of the backhauling technologies are evaluated based on power consumption analysis. The implementation of energy-efficient microwave communication links is also considered based on realistic power consumption. It is found that massive MIMO (M-MIMO) backhauling system consumes the highest power at maximum load. Power consumption in M-MIMO tends to rise as the number of transmitting antennas increases. The contribution of this work will help mobile network operators (MNO) in better decision making towards achieving a sustainable backhauling in future HetNet deployments.

Chapter “Analysis of Energy Consumption Model in Cloud Computing Environments” by Zhou Zhou, Jemal H. Abawajy and Fangmin Li discussed energy consumption in cloud computing data centres. The high energy consumption in data centres not only causes the energy wastes and system instability, but also generates low return on the investment and adverse effects on the environment. Therefore, it is extremely necessary to reduce energy consumption while meeting the quality of service. This chapter presents a fine-grained energy consumption model and analysis its effectiveness in energy consumption of data centres.

Lastly, we will like to thanks all the reviewers by providing sufficient and relevant comments that significantly improve the quality of the book chapters. Our profound appreciation goes to the authors for submitting their contributions to the edited book.

Kuala Lumpur, Malaysia	Tutut Herawan
Gombe, Nigeria	Haruna Chiroma
Geelong, Australia	Jemal H. Abawajy

Reviewers

Dr. Haruna Chiroma, Federal College of Education, Gombe, Nigeria,
e-mail: freedonchi@yahoo.com
Prof. Philippe Fournier-Viger, Harbin Institute of technology Shenzhen, China,
e-mail: philfv8@yahoo.com
Dr. Mohamed M. Mostafa, Gulf University for Science and Technology, Kuwait,
e-mail: moustafa.m@gust.edu.kw
Dr. Tarik Rashid, University College Dublin, Ireland, e-mail: tarik.rashid@ucd.ie
Assc. Prof. Gai-Ge Wang, University of Alberta, Canada,
e-mail: gaigewang@163.com
Assc. Prof. Lukas Pichl, International Christian University, Tokyo, Japan,
e-mail: lukas@icu.ac.jp
Dr. Shafi'i Muhammed Abdulhamid, Federal University of Technology, Minna,
Nigeria, e-mail: shafii.abdulhamid@futminna.edu.ng
Dr. Roberto A. Vázquez, Universidad La Salle, Mexico,
e-mail: ravem@lasallistas.org.mx
Dr. Rawaa Dawoud Al-Dabbagh, University of Baghdad, Iraq,
e-mail: rawaaiq04@yahoo.com
Dr. Adamu Ibrahim Abubakar, International Islamic University Malaysia,
e-mail: 100adamu@yahoo.com
Dr. Mukhtar Fatihu Hamza, Bayero University Kano,
e-mail: emukhtarfh@gmail.com
Dr. Yu Wue, Beijing University of Posts and Telecommunications,
e-mail: wuyue@bupt.edu.cn
Dr. Gokop Goteng, Queen's Marry University, London,
e-mail: g.l.goteng@qmul.ac.uk
Dr. Paul brogan, QUB Belfast United Kingdom, e-mail: pbrogan02@qub.ac.uk
Dr. Gabralla Lubna, University of Science and Technology, Kartum, Sudan,
e-mail: lubnagabralla@gmail.com
Dr. Mustapha Ismail, University of Bagdad, Iraq, e-mail: mofisml3@gmail.com
Dr. Paul Brogan, Queen's University Belfast, e-mail: pbrogan02@qub.ac.uk

Dr. Olugbenga Ogunmodimu, Lancaster University, United Kingdom, e-mail: O.Ogunmodimu@mmu.ac.uk
Dr. Tim Brown, Carnegy Mellon University, e-mail: timxb@cmu.edu
Prof. A. A. A. Atayero, Covenant University, Ota, Nigeria, e-mail: atayero.pa@gmail.com
Dr. Gaddafi Abdul-Salam, Kwame Nkrumah University of Science and Technology, Ghana, e-mail: gaddafi.ict@knust.edu.gh
Dr. Syed Hamid Hussain Madni, University of Technology Malaysia, e-mail: madni4all@yahoo.com
Sani Saminu, Hebei University of Technology China, e-mail: sansam4k@gmail.com

GET Authors

Abdul Samad Ismail
Abdulkarim A. Oloyede
Abdullah Khan
Abdulsalam Ya'u Gital
Abubakar Abdulkarim
Adamu Abubakar
Ahmed Aliyu
Aliyu D. Usman
Attaullah
Awais Qureshi
Ayuba K. Danburam
Dada Emmanuel Gbenga
Fangmin Li
Haruna Chiroma
Hassan Chizari
Hilal Ahmad
Ibrahim Abaker Targio Hashem
Jaafar Zubairu Maitama
Jemal H. Abawajy
Junaid Bukhari
Kamran Ullah
Lukman A. Olawoyin
Mohammed A. Gadam
Mohammed Abdullahi
Mohammed Joda Usman
Muhammad Idrees
Muhammad Murtala Ahmad
Mustafa T. Abd
Nasir Faruk
Nasreen Akhter
Nazmat T. Surajudeen-Bakinde

Rahmat Shah
Rawaa Dawoud Al-Dabbagh
Saad Adnan Abed
Salihu Idi Dishing
Sani Danjuma
Segun I. Popoola
Shafi'i Muhammad Abdulhamid
Shah Zaman
Suleiman M. Sani
Usman Ali Abdullahi
Yinusa A. Adediran
Younes Saadi
Zhou Zhou

Contents

A Theoretical Framework for Big Data Analytics Based on Computational Intelligent Algorithms with the Potential to Reduce Energy Consumption

Haruna Chiroma, Usman Ali Abdullahi, Ibrahim Abaker Targio Hashem, Younes Saadi, Rawaa Dawoud Al-Dabbagh, Muhammad Murtala Ahmad, Gbenga Emmanuel Dada, Sani Danjuma, Jaafar Zubairu Maitama, Adamu Abubakar and Shafi'i Muhammad Abdulhamid

Abstract Within the framework of big data, energy issues are highly significant. Despite the significance of energy, theoretical studies focusing primarily on the issue of energy within big data analytics in relation to computational intelligent

H. Chiroma (✉) · U. A. Abdullahi
Department of Computer Science, Federal College of Education (Technical), Gombe, Nigeria
e-mail: freedomchi@yahoo.com

U. A. Abdullahi
e-mail: danzazzau12@gmail.com

U. A. Abdullahi
Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Perak, Malaysia

I. A. T. Hashem
Centre for Data Science and Analytics, School of Computing & Information Technology, Taylor's University, Subang, Jaya, Malaysia, Kuala Lumpur, Malaysia
e-mail: ibrahimabaker.targiohashem@taylors.edu.my

Y. Saadi
Department of Computer Science, University of Batna, Fésdis, Algérie
e-mail: younessaadi@gmail.com

R. D. Al-Dabbagh
Department of Computer Science, University of Baghdad, Baghdad, Iraq
e-mail: rawaaiq04@yahoo.com

M. M. Ahmad
Department of Information Technology, National Open University of Nigeria, Lagos, Nigeria
e-mail: mursuw@gmail.com

G. E. Dada
Department of Computer Engineering, University of Maiduguri, Maiduguri, Nigeria
e-mail: gbengadada2004@yahoo.com

S. Danjuma
Department of Mathematical Science, North-West University Kano, Kano, Nigeria
e-mail: sani_danjuma@yahoo.com

T. Herawan et al. (eds.), *Advances on Computational Intelligence in Energy*,
Green Energy and Technology, https://doi.org/10.1007/978-3-319-69889-2_1

algorithms are scarce. The purpose of this study is to explore the theoretical aspects of energy issues in big data analytics in relation to computational intelligent algorithms since this is critical in exploring the emperica aspects of big data. In this chapter, we present a theoretical study of energy issues related to applications of computational intelligent algorithms in big data analytics. This work highlights that big data analytics using computational intelligent algorithms generates a very high amount of energy, especially during the training phase. The transmission of big data between service providers, users and data centres emits carbon dioxide as a result of high power consumption. This chapter proposes a theoretical framework for big data analytics using computational intelligent algorithms that has the potential to reduce energy consumption and enhance performance. We suggest that researchers should focus more attention on the issue of energy within big data analytics in relation to computational intelligent algorithms, before this becomes a widespread and urgent problem.

Keywords Big data analytics · Energy · Cluster systems · Computational intelligent algorithms · Artificial neural networks · Cuckoo search algorithm

1 Introduction

The International Energy Agency (IEA) has estimated that the global consumption of energy is expected to surge by 53% by the year 2035 [1]. Energy is viewed as the largest industry across the globe [2]. The consumption of energy involves all sectors of society, including information and communication technology. Shojafar et al. [3] have argued that real-time processing and energy efficiency are hot topics in the management of information and communication technology platforms. Currently, one of the most widely discussed topics in the science and technology community is big data. Big data has potential for applications in all sectors of society, such as climate, economics, health, social science [4]. The data collected from various sources in society is growing exponentially and is estimated to grow to 44 ZB (trillion gigabytes) by 2020, from 4.4 ZB in 2013 [5]. In big data, a diverse field of study which includes natural language processing, medical science, secu-

J. Z. Maitama
Department of Information Technology, Bayero University Kano, Kano, Nigeria
e-mail: jzmaitama.it@buk.edu.ng

A. Abubakar
Department of Information Systems, International Islamic University Malaysia, Gombak, Malaysia
e-mail: 100adamu@gmail.com

S. M. Abdulhamid
Department of Cyber Security Science, Federal University of Technology Minna, Minna, Nigeria
e-mail: shafii.abdulhamid@futminna.edu.ng

rity and business management depends heavily on knowledge discovery through big data analytics. The effective and efficient processing of big data requires computer systems [6] involving Hadoop which offer the MapReduce framework for parallel computation [7].

The transfer of large volumes of data between users, service providers and data centres requires a high bandwidth connection. This consumes large amounts of energy, more than simply processing and storing the big data within cloud-based data centres. Therefore, emits high carbon dioxide. The transfer of big datasets into remote data centres consumes a significant quantity of power [8], and these carbon dioxide emissions contribute to global warming [9]. The optimisation of energy consumption for data transmission requires the network to reduce redundant and duplicate traffic [10].

Applications for future generations of parallel and distributed systems in big data analytics are a major issue. These applications generate datasets in repositories that exceed exabytes, and the size of these datasets is speedily increasing. These datasets and their associated applications pose a challenge to both software techniques and software development [11]. The task of analysis frequently has strict targets, and one of the major issues for applications in this field is the quality of data. Most of the emerging applications, data-driven models and techniques with the capability of operating at large scales are not yet widely known [12]. In real-time systems, the amount of energy is increasing; thus, the application of big data methodologies can be used to handle these operations [13]. Significant developments in big data have arisen from various research communities, for example data mining and learning algorithms from the artificial intelligence research community [4].

Big data offers important opportunities for organisations who can analyse it and gain critical intelligence for effective decisions [14]. Within various industries, data processing and analysis plays a significant role, especially in situations where nonlinear dynamics, comprising various uncertainties and mathematical models, fails. Computational intelligent algorithms such as fuzzy logic, artificial neural networks and evolutionary algorithms have demonstrated their ability to deal effectively with data modelling, and research on computational intelligent algorithms has attracted unprecedented attention from researchers. Computational intelligent algorithms have successfully solved real-world problems, as reported in the literature. Examples of the effectiveness of computational intelligent algorithms in solving real-world problems include control engineering, the modelling of unknown nonlinear dynamics using artificial neural networks and the implementation of controllers using adaptive neuro-fuzzy inference systems. Despite the overwhelming successes recorded by computational intelligent algorithms in solving real-world problems, those within the research community are still facing obstacles to the processing of industrial data, such as feature extraction from large-scale measurements that are distributed in nature, machine learning algorithms for highly robust data modelling and signal processing [15].

The world is experiencing a data revolution in terms of gleaning knowledge from big data. Computational intelligent algorithms are among the mainstream tools of big data analytics: computational intelligence has played an important role in artificial intelligence, which focuses on the design of algorithms. Such algorithms can be used to analyse huge amounts of structured and unstructured data which help in the discovery of approximate solutions for many complex problems [16]. Meanwhile, the use of virtualised clouds is currently under consideration in big data analytics, based on new machine learning theories and artificial intelligence. It is now common that intuitive physical interpretations affect the use of machine learning and artificial intelligence. It is therefore important that a suitable knowledge interpretation is provided, in order to make sound decisions based on the intelligence derived from machine learning and artificial intelligence [17].

In this chapter, we propose a theoretical framework for big data analytics, based on computational intelligent algorithms with the potential to reduce energy consumption and improve performance. It is necessary to explore the theoretical aspects of energy issues in big data analytics in relation to computational intelligent algorithms; it is critical to explore the theoretical aspects of big data in view of the fact that this can point the way towards effective and efficient applications [4].

The remaining sections of this chapter are organised as follows: Section 2 introduces the computational intelligent algorithms; Sect. 3 presents a discussion of big data analytics and the energy consumption of cluster systems; Sect. 4 discusses big data analytics and computational intelligent algorithms; Sect. 5 describes the issue of energy consumption in the application of computational intelligent algorithms in big data analytics; Sect. 6 describes the proposed framework for big data analytics based on computational intelligent algorithms; and Sect. 7 presents the conclusions.

2 Computational Intelligent Algorithms

The computational intelligent algorithm is a name recently given to the branch of artificial intelligence that deals with sub-symbolic techniques. It offers a description of techniques that mainly focus on strategies and results. Computational intelligent algorithms include sub-disciplines that deal with adaptive and intelligent systems such as evolutionary computing, artificial neural networks, fuzzy systems, artificial immune systems, particle swarm optimisation (PSO) and ant colony optimisation [18, 19]. The primary source of inspiration for these intelligent systems is nature; most of these algorithms are inspired by the characteristics of biological systems, referred to as biology-inspired algorithms [20].

Solving real-world problems generally involves challenging and NP-hard problems which require optimisation techniques, with no guarantee of obtaining an optimal solution. There are no effective and efficient algorithms for all NP-hard problems; therefore, experimentation with various optimisation algorithms is required to find the algorithm that produces the optimal solution. Many computational intelligent algorithms such as PSO, cuckoo search and firefly have been introduced to deal with the challenges of optimisation problems [20]. Computational intelligent algorithms have become widespread, and this has significantly expanded the literature [21].

Recently, a new computational intelligent algorithm inspired by nature has been added to the literature almost every month. It is likely that there are more than 200 of such algorithms in the literature. As these algorithms have flooded the literature, many researchers have found that the newly created algorithms are the existing algorithms disguised as new ones [22]. Figure 1 illustrates the number of computational intelligent algorithms introduced into the literature per year. In 2009, the literature witnessed a drastic influx of computational intelligent algorithms. Figure 2 presents the classification of computational intelligent algorithms inspired by nature and is based on the classification proposed by Fister et al. [20].

2.1 Characteristics of Computational Intelligent Algorithms

In general, computational intelligent algorithms aim to generate a new solution which is superior to the existing one. Ideally, computational intelligent algorithms are expected to generate solutions superior to current solutions with minimal effort [21]. We now examine the major characteristics of computational intelligent algorithms: exploitation and exploration, parameter tuning/control, diversity and adaptation and algorithm parameters.

Exploitation and Exploration Exploitation is a local search process using local information for a problem and uses information obtained from a problem to generate new

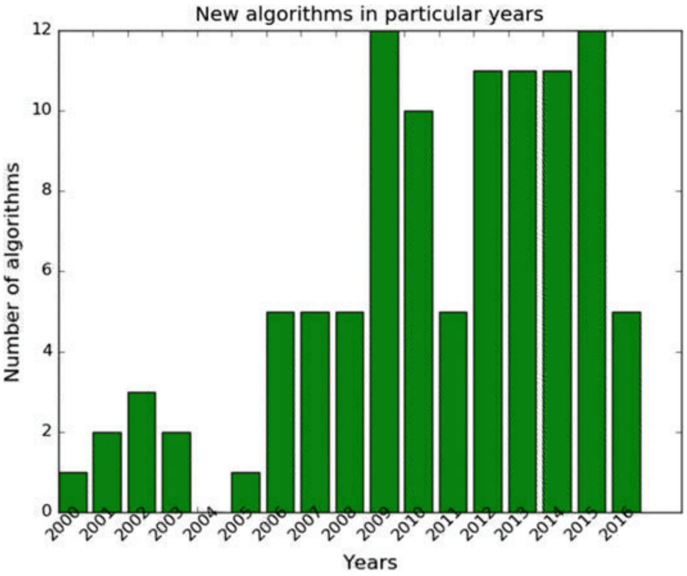


Fig. 1 Number of computational intelligent algorithms in the literature per year

solutions which are superior to the existing ones. The major strength of exploitation is its ability to give a high convergence rate. However, it has the possibility of becoming stuck in local minima. Conversely, exploration is a global search process which allows computational intelligent algorithms to explore the larger search space in far regions efficiently; it has the ability to generate solutions with sufficient diversity which is far from the existing solutions. Exploration has a lower propensity to become stuck in local minima, but it has a slow convergence rate and involves a high computational cost. Good performance for an algorithm requires a balance between exploitation and exploration: high exploitation and low exploration lead to faster convergence, but the possibility of finding a true global solution is low, while low exploitation and high exploration can lead to the meandering of the search path with a slow convergence rate [21].

Parameter Control/Tuning The values of parameters obtained through parameter tuning are fixed during iterations, whereas the parameters of an algorithm are varied for the purpose of control. Parameter control aims to find the algorithm with the best convergence rate for better performance; parameter tuning is carried out to find the optimal parameter settings for the running of the algorithm, in order to solve a broader array of problems. There is currently no systematic and efficient method of tuning to obtain optimal parameter settings; this is often realised through extensive experiments on parameter studies [23].

Diversity and Adaptation The computational intelligent algorithms have both diversity and adaptation, which are evident from the balance between exploitation and exploration. For example, ways of balancing exploration and exploitation are the key form of adaptation. For instance, the representations of solutions in genetic algorithms are either in binary or in real number form, whereas swarm intelligence-

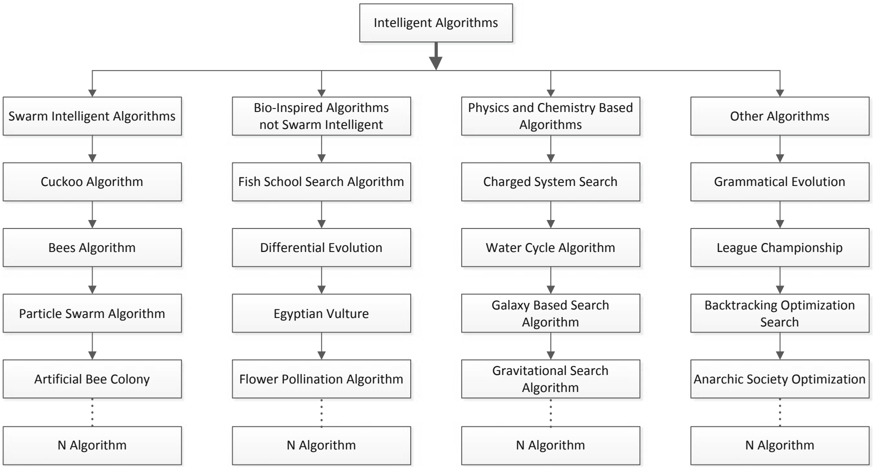

Intelligent Algorithms
Swarm Intelligent Algorithms
Bio-Inspired Algorithms not Swarm Intelligent
Physics and Chemistry Based Algorithms
Other Algorithms
Cuckoo Algorithm
Fish School Search Algorithm
Charged System Search
Grammatical Evolution
Bees Algorithm
Differential Evolution
Water Cycle Algorithm
League Championship
Particle Swarm Algorithm
Egyptian Vulture
Galaxy Based Search Algorithm
Backtracking Optimization Search
Artificial Bee Colony
Flower Pollination Algorithm
Gravitational Search Algorithm
Anarchic Society Optimization
N Algorithm
N Algorithm
N Algorithm
N Algorithm

Fig. 2 Classification of intelligent algorithms

based algorithms generally use real numbers for solutions. The population size of an algorithm can either be fixed or varying, and variation in population size is therefore a typical example of adaptation.

Algorithm Parameters These algorithms involve parameters, and algorithm operators are used to construct the algorithms. In genetic algorithms, crossover, mutation and selection are used. Crossover is the operation used to create new solutions [21]. As an example of the differences between computational intelligent algorithms in terms of parameters, strengths, weaknesses, generation of new solution and solution representation, Table 1 presents five different well-established algorithms from the literature with their differences.

3 Big Data Analytics and Energy Consumption by Cluster Computing Systems

3.1 Big Data Analytics Platforms

The unprecedented accumulation of data in the information technology world has given rise to the concept of big data. Volumes are extremely large, with petabytes (PB) and even zettabytes (ZB) of data handled by organisations. The velocity and time-based variability of this data involve high speeds. The formats in which this data is created and stored are inconsistent, although these may originate from the same source and/or be generated by the same user. The veracity of the data, as opposed to the noise inherent in it, is of the highest concern. Despite all these features, big data offers high value when properly stored and analysed [29, 30]. Internet companies handle large volumes of Internet requests from their users using big data analytical platforms running on clusters of commodity hardware. Facebook and Walmart are two good examples [31].

Due to the above-mentioned characteristics, the storage and analytics of big data require large hardware resources. For example, in order to store 1 PB of data on a cluster with a 6 TB capacity hard disk, 163 hard-disk units are required. Assuming each node in the cluster can host five hard-disk units, then a 100-node cluster is required. In addition, when analysing this 1 PB of data, each hard disk must be accessed for IO, depending on the platform used for big data analysis. This is due to the fact that big data analytics platforms use the sequential access method by default. The most popular big data analysis framework, Hadoop, uses a full scan (sequential access) by default for the targeted data [32]. Each IO operation on a hard disk leads to a maximum consumption of energy in the form of electricity. *Consumption due to powering of the cluster and cooling*: The scenario described above can be used to give a wider picture of how much energy is consumed by companies that deal with big data. This explains why power and energy are always of first-order priority in the design of computing systems infrastructure [31].

Table 1 Major differences between five well-established computational intelligent algorithms

Algorithm	Strengths	Weaknesses	Generation of new solutions	Parameters required for settings	Solution representation	Reference
CS	Strikes a balance between exploitation and exploration. Requires few parameter settings	Sensitive to parameter settings	Lévy walk and far-field randomisation	Probability of the worst nest and step size	Real number	Yang and Deb [24]
FP	Balance between consistency and exploration	Tuning switching probability can create fluctuations	Pollination	Switching probability	Real number	Yang [25]
ABC	Strong exploration	Weak exploitation. Requires many parameter settings	Randomness	Number of scouts, swarm size, number of onlookers and number of employed bees	Real number	Karaboga [26]
PSO	Computationally inexpensive and very simple to implement	Possibility of premature convergence	Adding velocity to a position	Population size, probability, gamma and neighbourhood	Real number	Eberhart and Kennedy [27]
GA	Strong exploitation	Unnecessary number of iterations. Weak exploration	Crossover	Probability of crossover, probability of mutation, population size, generations	Real number/binary number	Holland [28]

CS Cuckoo search algorithm, *GA* genetic algorithm, *FL* flower pollination algorithm, *ABC* artificial bee colony

Initially, information technology companies used clusters of commodity hardware and networking to avoid the high costs of hardware. These companies see this as a better option for providing the infrastructure necessary to accumulate and process large amounts of data, in comparison with the expense of purchasing and maintaining supercomputers and mainframe computers which would allow the system to achieve the same purpose. However, the amount of power and energy consumed by these clusters, particularly during big data storage and analysis, is growing at an alarming rate. The cost of the energy used by servers within their lifetime is expected to supersede the cost of the hardware itself, if current trends continue unchecked [31]. For example, Yahoo has installed a Hadoop cluster of over 2000 servers, while that of Facebook has more than 600 servers. Similarly, General Electric has deployed Hadoop on a cluster of 1700 servers. Energy is a crucial issue in view of these massive deployments of Hadoop over thousands of cluster systems and has influenced the cost of exploiting cluster systems. In 2007, there was a high cost of energy consumption for cluster systems [33]. This example shows the massive deployment of Hadoop over a cluster of hundreds of servers. In addition, the deployment and operation of Hadoop, the hardware required to build the cluster systems and the energy required to run them incur very high costs [31].

3.2 Energy Consumption Over Big Data Platforms

Energy consumption in big data platforms is related to several factors such as physical resources and computing resources. Big data platforms have the ability to model, organise, store and process large amounts of data. Development of information technology platforms and the massive generation of data in the world, big data technologies has become the battlefield of information technology service providers in terms of high performance and cost. The research community started to focus on energy consumption in big data platforms [34]. With the fast development of the global economy, energy consumption will keep increasing in the upcoming years.

Figures 3 and 4 are created based on the data presented in [35]; it shows the carbon dioxide emissions of data centres according to [36]. Whereas Fig. 4 depicts the distribution of energy consumption in data centres, Figure 3 shows that almost 6% per year of the emissions are caused by information technology servers. The consumption of energy in big data platforms can be expressed as follows:

$$E_{\mathrm{c}} = C_{\mathrm{r}} + P_{\mathrm{r}} \tag{1}$$

where E_{c} denotes the energy consumption, C_{r} represents the computing resources and P_{r} denotes the physical resources. According to [37], the energy consumption of computing resources accounts for about 50% of the total energy consumption as shown in Fig. 4. The percentage of energy consumes by servers' computation; the communication equipment, and the storage devices are depicted in Fig. 4.

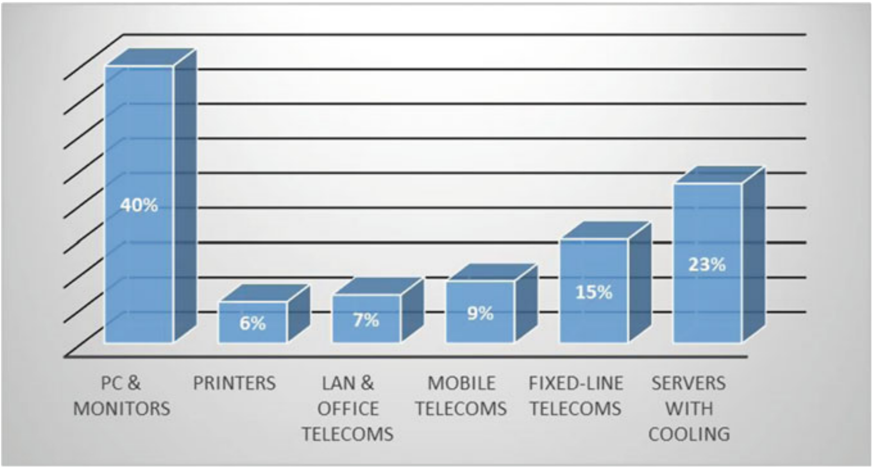


Fig. 3 Estimated information and communication technology carbon dioxide emissions

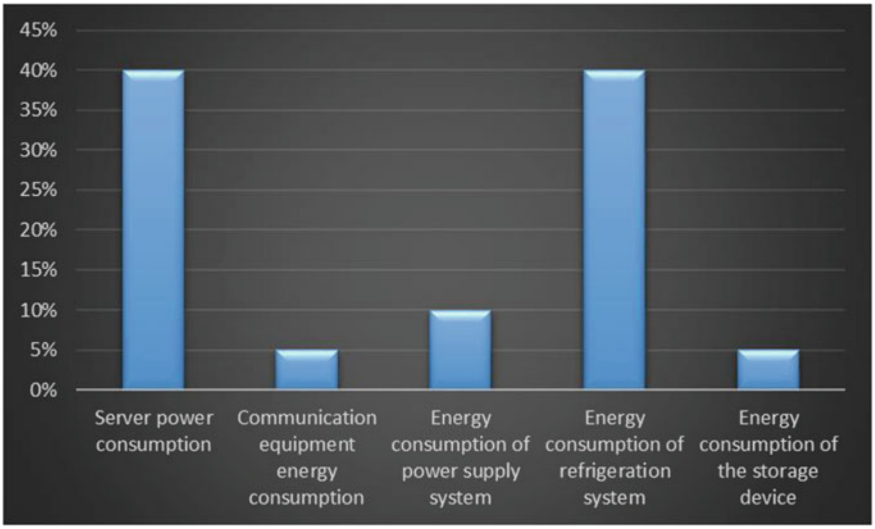


Fig. 4 Energy consumption distribution of data centres

From Fig. 4, it is clear that the data servers are consuming a big part of the energy consumed by data centres. This amount grows exponentially in case of processing large datasets which is the case of big data platforms. Therefore, reducing energy consumption for big data platforms is the key issue for sustainable big data platforms.

3.3 Metrics Used for Measuring Power in Big Data Platforms

The management of energy consumption can be formulated as a multi-objective optimisation problem, where several performances and energy metrics are used to measure the performance [38–42]. It is very important to highlight that the following two objectives are mostly considered in the literature: minimisation and maximisation. The minimisation consists of reducing the consumption of the data platforms during peak power. The maximisation consists of increasing energy efficiency.

In fact, limiting the consumption during peak power is very crucial to maintain the reliability of big data platforms, escape system overheating and avoid power capacity overloads. It is shown that reducing power consumption is strongly correlated with the cost of power provisioning [43]. Energy efficiency can be expressed as follows:

$$\text{Energy efficiency} = \frac{\text{Computing performance}}{\text{Total energy consumed}} \tag{2}$$

This metric represents the main focus of energy management of data centres and processing systems. From the perspective of power management of data centres, energy consumption control is viewed as a result of [39]:

Power Usage Effectiveness Facility efficiency is the ratio of the total amount of energy used by a data centre facility to the energy delivered to computing equipment.

Server Power Usage Effectiveness Server power conversion efficiency is the ratio of the total server input power to its useful power consumed by the electronic components directly involved in the computation.

Server's Architectural Efficiency Server's architectural efficiency is the ratio of computing performance metric to the total amount of energy used by electronic components.

4 Computational Intelligent Algorithms and Big Data Analytics

In today's world, almost everything is online, and organisations intending to improve their services analyse big data to gain knowledge to be used in improving their services [44]. Big datasets are beyond the scope of relational or object-oriented databases, and traditional computer applications and normal computers cannot handle the analytics involved. These big datasets require very large parallel processing power, from clusters of computers, for analysis. The processing of big data is generally based on nonlinear systems, and actions are not predictable in some cases [45].

To discover the knowledge required for decision making, data mining algorithms are applied to the datasets extracted from data sources. In recent years, much attention

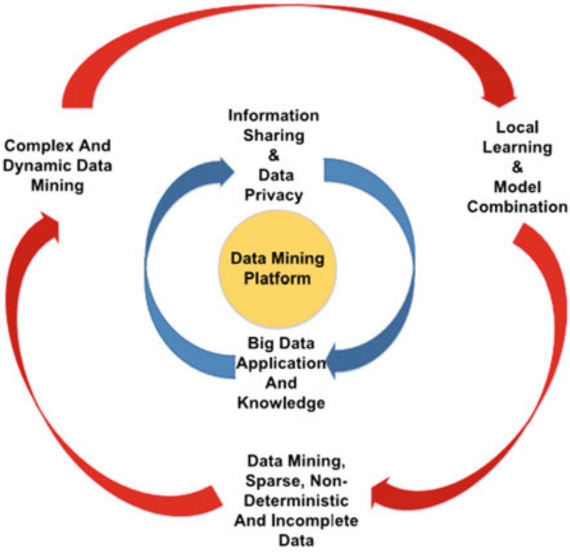

Information Sharing & Data Privacy
Local Learning & Model Combination
Complex And Dynamic Data Mining
Data Mining Platform
Big Data Application And Knowledge
Data Mining, Sparse, Non-Deterministic And Incomplete Data

Fig. 5 Three primary layers of big data analytics [49]

has been given to data mining, probably due to the popularity of big data concepts [44]. Big data analytics involves modelling, analysis and interpretation [46]. It has been shown that computational intelligent algorithms can be applied to solve big data problems effectively from the perspective of hardware and software design [29].

The application of computational intelligent algorithms in big data analytics is severely limited, however, since recent intelligent algorithms have difficulty in analysing big data. This is because the nature of big data makes it difficult for these intelligent algorithms to analyse it [47]. The proposed basic framework for big data analytics in relation to data mining is shown in Fig. 5; these data mining algorithms also include computational intelligent algorithms [48] (discussed in Sect. 2).

The commonly accepted framework for big data analytics is shown in Fig. 5. It comprises three layers [50] as follows:

i. Data access and computing,
ii. Data privacy and
iii. Domain knowledge and data mining algorithms.

The core of Fig. 5 is the data mining platform, which is responsible for data access and computing processes. With the increasing accumulation of high volumes of data, the distributed storage of large-scale data is required to be considered during computation. In brief, data analytics and the processing of the task are partitioned into sub-tasks in multiple forms for parallel execution on a large number of computing nodes. The role of the middle layer structure is to connect the outer and inner layers. The inner layer contains data mining technology, responsible for providing a platform for the execution of data-related activities in the middle layer. Examples of data-related work include information sharing, privacy protection and the acquisition of knowledge from areas and applications.

Information sharing is the concern of the whole framework, including processing and big data analytics in smart grid. The outer layer of Fig. 5 shows the data fusion technology necessary for the preprocessing of the heterogeneous, uncertain, incomplete and multi-source data. Complex and dynamic data is extracted after the data preprocessing phase. Subsequently, pervasive smart-grid global knowledge can be obtained through local learning and fusion of the model [31]. Of the decision tree, ridge regression and support vector machine algorithms, the decision tree is found to be the most efficient algorithm for managing energy data based on big data. However, when efficiency is the priority, for example in real-time applications, ridge regression is the most effective algorithm of these three algorithms [51].

Learning is a subfield of machine learning that has the potential to solve a range of complex problems within mobile big data analytics, including classification and regression. Mobile big data samples can be modelled using deep learning consisting of neurons and synapses for training mobile big data samples to learn hierarchical features.

The application of deep learning within mobile big data has the advantages of a high level of accuracy, which is a priority in mobile systems, and multimodal deep learning; intrinsic features are generated by deep learning, necessary in mobile big data analytics, and unlabelled mobile data can be learned using deep learning, which reduces the effort required for data labelling. However, the large number of deep model parameters and the large size of mobile big datasets mean that deep learning is slow and computationally expensive [52].

More recently, deep learning has become a common technique in big data analytics, especially in the retrieval of images with a high level of accuracy [53]. Supervised deep learning and unsupervised deep learning are the two types of deep learning discussed in the literature [54]. A battery with a limited capacity requires an energy efficiency of hundreds of giga floating-point operations per second per watt for a mobile embedded system. This can allow mobile embedded systems to achieve both the required portability and performance [55].

5 Energy Consumption in the Application of Computational Intelligent Algorithms in Big Data Analytics

In the design of computing systems, energy efficiency is one of the most significant issues to be considered. However, the termination of Moore's law has imposed a limit on additional improvements to energy efficiency, which is unfortunate. Recently, the use of physical memristors has shown that it is possible to generate a solution for the integrated hardware of artificial neural networks. This can heavily influence energy efficiency and improve performance [10, 55]. The artificial neural network is one of the more powerful algorithms in computational intelligence and has received unprecedented attention from researchers; it is believed to constitute one of the major breakthroughs in artificial intelligence. Hu et al. [56] used memristors in the design of a power neuromorphic framework for approximating computation with programmability and computational generality. This design was motivated by the theory of artificial neural networks, which shows that multilayer neural networks are universal approximators, and their wide range of applications in signal processing, pattern recognition, computer vision and natural language processing. A neuromorphic architecture for computing and a tolerance for uncertain computing can generate significant performance and gains in energy efficiency. Wang et al. [55] have found that large-scale artificial neural networks constitute one of the most mainstream algorithms in big data analytics. Two phases are involved in the processing of big data using large-scale artificial neural networks: a training phase and an operational phase. The training phase in big data processing requires a very high amount of computing power and energy efficiency, and this is one of the primary considerations in the operational phase. For example, a ~100 MB training dataset is needed with >100 TOP computation capability, ~40 GB/s IO and SRAM data bandwidth. A 3.4 GHz CPU therefore requires >10 h of learning time for ~100 K input-vector datasets; this requires ~1 s for recognition and is far from real-time processing [57].

The use of computational intelligent algorithms in big data analytics requires high bandwidth interconnection networks with low latency and low power consumption, which are essential for data and storage systems [58]. For example, Wang et al. [55] presented a promising ultrahigh energy-efficient implementation by taking advantage of emerging memristor techniques involving the computing power of GPUs for big data analytics. The results showed a high speedup compared with the basic CPU implementation [55]. Big data analytics using artificial neural networks poses the challenge of how to achieve better training within a lower convergence time and with lower energy consumption [55]. Another computational intelligent algorithm related to energy itself is the deep belief network. The deep belief network consists of the stack of a restricted Boltzmann machine. This is based on the model of energy and certain stochastic methods. Binary values are generated by each of the nodes in the restricted Boltzmann machine, between each node, symmetric link (weight) that can have negative and positive numbers exist. The two types of node in a restricted Boltzmann machine are visible and hidden nodes. The state of a restricted Boltzmann machine is associated with the energy of the restricted Boltzmann machine; a higher energy of restricted Boltzmann machine gives a lower probability of node activation [59].

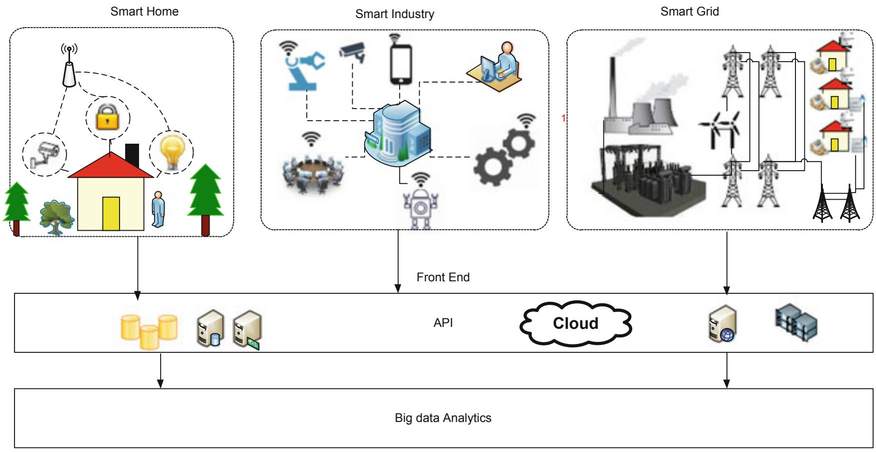


Fig. 6 Large-scale data processing using big data analytics tools such as computational intelligent algorithms

6 A Proposed Framework for Big Data Analytics Using Computational Intelligent Algorithms

Previously, several big data analytics techniques have required an investment in computer hardware and software in order to overcome energy computations. This can result in the downgrading of the performance of the systems used in highly computational environments, which need to process massive amounts of energy data. However, cloud computing and software as a service (SaaS) have now made on-premise solutions unnecessary. In addition, fog computing now allows much of the analytics for these tasks to be moved to the grid edge, to further support the implementation of forecasting and optimisation programmes in real time and at large scales. Two primary big data intelligence applications are load modelling and forecasting, which have been used for energy. These applications are necessary to understand the behaviour of the system in achieving efficient energy management and to enable generic load forecasting [17].

One of the forces driving the adoption of big data analytics is the development of smart grids, although the data generated through these smart grids is increasing in size and difficult to process. Advancements in big data and cloud computing technologies are therefore crucial for a sustainable energy system. Figure 6 shows the processing of large-scale data using big data analytics techniques such as computational intelligent algorithms.

The framework proposes three steps in big data analytics. The first is to collect the required data from sources such as smart industry, smart grid and smart home applications. Subsequently, these data can be stored in a database and servers using cloud storage. A parallel algorithm for optimal power flow based on MapReduce, proposed by Liang et al. [60], is to be applied for power flow calculation in a smart grid, since this reduces computational complexity. Lastly, the stored data can be processed using computational intelligent algorithms in order to gain insights from the big data.

The application of computational intelligent algorithms consumes a large amount of energy during the training phase as discussed earlier. The proposed framework involves the application of an energy-efficient emerging memristor, which is embedded with an artificial neural network for big data analytics. New and unexpected challenges have been created for the research community, since the current theories and techniques cannot handle big data analytics. Therefore, extension and upgrading of the existing techniques and theories are required to handle big data analytics [4]. As a result, there is no need to propose additional computational intelligent algorithms in the literature (refer to Sect. 2 for a justification); attention should be focused on the modification of the existing ones, since these all have limitations that require improvements in order to allow them to work on big data. The existing computational intelligent algorithms should be modified to handle big data analytics and to require a low energy consumption during the training phase. When processing big data for analytics using a computational intelligent algorithm, a hybrid storage device which combines hard disks and solid-state disks, as proposed by Polato et al. [61], should be used for data storage, as this reduces energy consumption and enhances performance. In addition, the storage of only a proportion of the data in solid-state disks has the potential to significantly save energy and speed up convergence. One typical example of the very high volumes of data generated from smart meters that have the potential to be analysed by computational intelligent algorithms is the case of the 200 TB data generated in Jiangsu, China. Data generation is increasing by 90 GB each day. These data were generated by over 1.81 million acquisition terminals, 1.54 million concentrated meter-reading terminals and 38 million smart meters [62]. Modern data centres should give priority to massive parallel processing in order to enhance computing speeds and reduce energy consumption. In turn, large amounts of data can be moved between the various virtual machines [63]. This framework suggests the use of the following techniques to cushion the high cost of energy consumption by distributed system clusters:

Proportioning the Use of Power The power consumed by hardware components can be proportioned to optimise the operational efficiency of a particular server. The proportioning can also be based on a capped budget, within which the server can be underprovided, in order to be more efficient, or oversubscribed, to operate at high load levels in carrying out its task [31, 62, 64].

Performance Improvement High energy consumption can also be optimised by improving the performance of big data analytical platforms. One example of these improvements is the indexing approach, which works with the main analytical processes and ensures that a full scan of the input data is prevented. This indexing helps to restrict the range of input data to be processed to only that required to carry out the task. Another approach in this category is the scheduling of jobs by considering data locality [65].

Iqbal et al. [66] have pointed out the potential application areas of computational intelligent algorithms within big data analytics which are as follows: personalised health services, biometrics and surveillance, transportation, visualisation of data and interpretation, business and governance, sentimental analysis, models for population displacement, effective computation, fault detection and manufacturing.

7 Conclusions

This chapter presents a theoretical perspective of energy issues within big data analytics, as related to computational intelligent algorithms. Theoretical issues of energy consumption related to big data analytics are described based on computational intelligent algorithms. It is found that the high consumption of energy in big data analytics using computational intelligent algorithms occurs mostly during the training phase of big data processing. We propose a big data analytics theoretical framework based on computational intelligent algorithms with the potential for low energy consumption and performance improvement. The theoretical study presented in this chapter may guide researchers to apply computational intelligent algorithms efficiently and effectively in big data analytics, with the possibility of consuming low energy, and improve performance. Future research directions should be focused on the application of a deep belief network in big data analytics, which consists of the stack of a restricted Boltzmann machine.

References

1. IEA (2016, 5 May, 2017) World Energy Outlook. Available: http://www.iea.org/newsroom/news/2016/november/world-energy-outlook-2016.html
2. Horn M, Mirzatuny M (2013) Mining big data to transform electricity. In: Broadband networks, smart grids and climate change. Springer, Berlin, pp 47–58
3. Shojafar M, Cordeschi N, Amendola D, Baccarelli E (2015) Energy-saving adaptive computing and traffic engineering for real-time-service data centers. In: 2015 IEEE international conference on communication workshop (ICCW), pp 1800–1806
4. Yu S, Wang C, Liu K, Zomaya AY (2016) Editorial for IEEE access special section on theoretical foundations for big data applications: challenges and opportunities. IEEE Access 4:5730–5732
5. Salinas S, Chen X, Ji J, Li P (2016) A tutorial on secure outsourcing of large-scale computations for big data. IEEE Access 4:1406–1416
6. Wu K, Barker RJ, Kim MA, Ross KA (2013) Navigating big data with high-throughput, energy-efficient data partitioning. In: ACM SIGARCH computer architecture news, 2013, pp 249–260
7. Li C, Zu Y, Hou B (2016) A feature selection method of power consumption data. In: International conference on computational science and its applications, 2016, pp 547–554
8. Baker T, Al-Dawsari B, Tawfik H, Reid D, Ngoko Y (2015) GreeDi: an energy efficient routing algorithm for big data on cloud. Ad Hoc Netw 35:83–96
9. Chiroma H, Abdul-Kareem S, Khan A, Nawi NM, Gital AYU, Shuib L et al (2015) Global warming: predicting OPEC carbon dioxide emissions from petroleum consumption using neural network and hybrid cuckoo search algorithm. PloS One 10:e0136140

10. Li R, Harai H, Asaeda H (2015) An aggregatable name-based routing for energy-efficient data sharing in big data era. IEEE Access 3:955–966
11. Dean J, Ghemawat S (2004) MapReduce: simplified data processing on large clusters. In: OSDI'04 Proceedings of the 6th conference on symposium on operating systems design and implementation (Int J Eng Sci Invent). URL: http://static.googleusercontent.com/media/research.google.com (diunduh pada 2015-05-10), pp 10–100
12. Kambatla K, Kollias G, Kumar V, Grama A (2014) Trends in big data analytics. J Parallel Distrib Comput 74:2561–2573
13. Fernández MR, García AC, Alonso IG, Casanova EZ (2016) Using the Big Data generated by the Smart Home to improve energy efficiency management. Energ Effi 9:249–260
14. Abawajy J (2015) Comprehensive analysis of big data variety landscape. Int J Parallel Emergent Distrib Syst 30:5–14
15. Wang D, Yu W, Chai T (2015) Guest editorial: special issue on computational intelligence for industrial data processing and analysis. Neurocomputing 358–360
16. Cuadra L, Salcedo-Sanz S, Nieto-Borge J, Alexandre E, Rodríguez G (2016) Computational intelligence in wave energy: comprehensive review and case study. Renew Sustain Energy Rev 58:1223–1246
17. Hu J, Vasilakos AV (2016) Energy big data analytics and security: challenges and opportunities. IEEE Trans Smart Grid 7:2423–2436
18. Engelbrecht AP (2007) Introduction to computational intelligence. In: Computational intelligence: an introduction, 2nd edn, pp 1–13
19. Păun G (2005) Bio-inspired computing paradigms (natural computing). In: Unconventional programming paradigms, pp 97–97
20. Fister Jr I, Yang X-S, Fister I, Brest J, Fister D (2013) A brief review of nature-inspired algorithms for optimization. arXiv preprint arXiv:1307.4186
21. Yang X-S, He X (2016) Nature-inspired optimization algorithms in engineering: overview and applications. In: Nature-inspired computation in engineering. Springer, Berlin, pp 1–20
22. Fister Jr I, Mlakar U, Brest J, Fister I (2016) A new population-based nature-inspired algorithm every month: is the current era coming to the end. In: StuCoSReC: proceedings of the 2016 3rd student computer science research conference. University of Primorska, Koper, pp 33–37
23. Yang X-S (2014) Cuckoo search and firefly algorithm: overview and analysis. In: Cuckoo search and firefly algorithm. Springer, Berlin, pp 1–26
24. Yang X-S, Deb S (2009) Cuckoo search via Lévy flights. In: World congress on nature & biologically inspired computing, NaBIC 2009, pp 210–214
25. Yang X-S (2012) Flower pollination algorithm for global optimization. In: International conference on unconventional computing and natural computation, 2012, pp 240–249
26. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report-tr06. Erciyes University, Engineering Faculty, Computer Engineering Department
27. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, MHS'95, pp 39–43
28. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, MI. Reprinted in 1998
29. Chen M, Mao S, Liu Y (2014) Big data: a survey. Mob Netw Appl 19:171–209
30. Chen CP, Zhang C-Y (2014) Data-intensive applications, challenges, techniques and technologies: a survey on Big Data. Inf Sci 275:314–347
31. Gupta A, Gupta S, Ge R, Zong Z (2015) CRUSH: data collection and analysis framework for power capped data intensive computing. In: 2015 sixth international green computing conference and sustainable computing conference (IGSC), pp 1–6
32. Yang H-C, Parker DS (2009) Traverse: simplified indexing on large map-reduce-merge clusters. In: International conference on database systems for advanced applications, 2009, pp 308–322
33. Jlassi A, Martineau P (2016) Benchmarking Hadoop performance in the cloud-an in depth study of resource management and energy consumption. In: The 6th international conference on cloud computing and services science

34. Rabl T, Gómez-Villamor S, Sadoghi M, Muntés-Mulero V, Jacobsen H-A, Mankovskii S (2012) Solving big data challenges for enterprise application performance management. Proc VLDB Endowment 5:1724–1735
35. Rong H, Zhang H, Xiao S, Li C, Hu C (2016) Optimizing energy consumption for data centers. Renew Sustain Energy Rev 58:674–691
36. Kumar R, Mieritz L (2007) Conceptualizing green IT and data center power and cooling issues. Gartner research paper, 2007
37. Johnson P, Marker T (2009) Data centre energy efficiency product profile. Pitt & Sherry, report to equipment energy efficiency committee (E3) of The Australian Government Department of the Environment, Water, Heritage and the Arts (DEWHA)
38. Karpowicz M, Niewiadomska-Szynkiewicz E, Arabas P, Sikora A (2016) Energy and power efficiency in cloud. In: Resource management for big data platforms. Springer, Berlin, pp 97–127
39. Barroso LA, Clidaras J, Hölzle U (2013) The datacenter as a computer: an introduction to the design of warehouse-scale machines. Synth Lect Comput Archit 8:1–154
40. Lefurgy C, Rajamani K, Rawson F, Felter W, Kistler M, Keller TW (2003) Energy management for commercial servers. Computer 36:39–48
41. Mastelic T, Oleksiak A, Claussen H, Brandic I, Pierson J-M, Vasilakos AV (2015) Cloud computing: survey on energy efficiency. ACM Comput Surv (CSUR) 47:33
42. Wang L, Khan SU (2013) Review of performance metrics for green data centers: a taxonomy study. J Supercomput 63:639–656
43. Dongarra J, Beckman P, Moore T, Aerts P, Aloisio G, Andre J-C et al (2011) The international exascale software project roadmap. Int J High Perform Comput Appl 25:3–60
44. Khalifa S, Elshater Y, Sundaravarathan K, Bhat A, Martin P, Imam F et al (2016) The six pillars for building big data analytics ecosystems. ACM Comput Surv (CSUR) 49:33
45. Cheng S, Liu B, Shi Y, Jin Y, Li B (2016) Evolutionary computation and big data: key challenges and future directions. In: International conference on data mining and big data, pp 3–14
46. Gandomi A, Haider M (2015) Beyond the hype: big data concepts, methods, and analytics. Int J Inf Manage 35:137–144
47. Hashem IAT, Chang V, Anuar NB, Adewole K, Yaqoob I, Gani A et al (2016) The role of big data in smart city. Int J Inf Manage 36:748–758
48. Chiroma H, Abdul-Kareem S, Abubakar A (2014) A framework for selecting the optimal technique suitable for application in a data mining task. In: Future information technology. Springer, Berlin, pp 163–169
49. Jiang H, Wang K, Wang Y, Gao M, Zhang Y (2016) Energy big data: a survey. IEEE Access 4:3844–3861
50. Hu H, Wen Y, Chua T-S, Li X (2014) Toward scalable systems for big data analytics: a technology tutorial. IEEE Access 2:652–687
51. Kang D, Kim S, Lee T, Hwang J, Lee S, Jang S et al (2016) Energy information analysis using data algorithms based on big data platform. In: High performance computing and communications; IEEE 14th international conference on smart city; IEEE 2nd international conference on data science and systems (HPCC/SmartCity/DSS), 2016 IEEE 18th international conference on, pp 1530–1531
52. Alsheikh MA, Niyato D, Lin S, Tan H-P, Han Z (2016) Mobile big data analytics using deep learning and apache spark. IEEE Network 30:22–29
53. Lee H, Grosse R, Ranganath R, Ng AY (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th annual international conference on machine learning, pp 609–616
54. Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554
55. Wang Y, Li B, Luo R, Chen Y, Xu N, Yang H (2014) Energy efficient neural networks for big data analytics. In: Design, automation and test in Europe conference and exhibition (DATE), 2014, pp 1–2

56. Hu M, Li H, Wu Q, Rose GS (2012) Hardware realization of BSB recall function using memristor crossbar arrays. In: Proceedings of the 49th annual design automation conference, pp 498–503
57. Yoo H, Park S, Bong K, Shin D, Lee J, Choi S (2015) A 1.93 TOPS/W scalable deep learning/inference processor with tetra-parallel MIMD architecture for big data applications. In: IEEE international solid-state circuits conference, pp 80–81
58. Mehdipour F, Noori H, Javadi B (2016) Chapter two-energy-efficient big data analytics in datacenters. Adv Comput 100:59–101
59. Park S-W, Park J, Bong K, Shin D, Lee J, Choi S et al (2015) An energy-efficient and scalable deep learning/inference processor with tetra-parallel MIMD architecture for big data applications. IEEE Trans Biomed Circuits Syst 9:838–848
60. Liang B, Jin S, Tang W, Sheng W, Liu K (2016) A parallel algorithm of optimal power flow on Hadoop platform. In: Power and energy engineering conference (APPEEC), 2016 IEEE PES Asia-Pacific, pp 566–570
61. Polato I, Barbosa D, Hindle, Kon F (2016) Hadoop energy consumption reduction with hybrid HDFS. In: Proceedings of the 31st annual ACM symposium on applied computing, pp 406–411
62. Nan Z, Hanyong H, Haiyan Z (2016) Efficient stereo index technology for fast combination query of electric power big data. In: 2016 IEEE international conference on computer communication and the internet (ICCCI), pp 329–333
63. Baccarelli E, Cordeschi N, Mei A, Panella M, Shojafar M, Stefa J (2016) Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study. IEEE Network 30:54–61
64. Zhu N, Rao L, Liu X, Liu J, Guan H (2011) Taming power peaks in mapreduce clusters. In: ACM SIGCOMM computer communication review, pp 416–417
65. Lee S, Jo J-Y, Kim Y (2016) Performance improvement of mapreduce process by promoting deep data locality. In: 2016 IEEE international conference on data science and advanced analytics (DSAA), pp 292–301
66. Iqbal R, Doctor F, More B, Mahmud S, Yousuf U (2016) Big data analytics: computational intelligence techniques and application areas. Int J Inf Manage

Artificial Bee Colony for Minimizing the Energy Consumption in Mobile Ad Hoc Network

Mustafa Tareq, Saad Adnan Abed and Elankovan A. Sundararajan

Abstract Wireless ad hoc network is widely used nowadays, in particular the mobile type, known as the mobile ad hoc network (MANET). This type of network consists of sets of mobile nodes that do not require a fixed infrastructure such as an access point or base station. The common use of MANET is to enable nodes contacting in the absence of the typical communications infrastructure. Constantly changing topology and having no fixed infrastructures are some of the challenges confronted through a MANET designing. Hence, emphasizing the need to establish an efficient connection inside the network we use for a routing protocol to explore paths among nodes. The guarantee of finding optimum path formation among the nodes is the primary goal of the routing protocol, in order to ensure that messages would be delivered timely. The aim of this paper is to find the best possible route from the source to the destination based on a method inspired by the searching behaviour of bee colonies, i.e. artificial bee colony (ABC) algorithm. This algorithm works on minimizing the average energy consumption of the selected route. For evaluation purposes, the proposed model has been applied on two protocols, i.e. the Destination-Sequenced Distance-Vector Routing (DSDV) and Ad hoc On-demand Distance-Vector (AODV). The evaluation is based on node speed and packet size topology parameters. The results show that the network nodes can save more energy in AODV as compared to DSDV. As such, it can be concluded that optimizing the path from the source to the destination has a significant impact on the quality of the network performance.

Keywords MANET · Energy consumption · AODV · DSDV · Artificial bee colony

M. Tareq (✉) · E. A. Sundararajan
Centre of Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia
e-mail: P89127@siswa.ukm.edu.my

E. A. Sundararajan
e-mail: elan@ukm.edu.my

S. A. Abed
Computer and Information Sciences Department, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perak, Malaysia
e-mail: saad.adnan_g03549@utp.edu.my

T. Herawan et al. (eds.), *Advances on Computational Intelligence in Energy*,
Green Energy and Technology, https://doi.org/10.1007/978-3-319-69889-2_2

1 Introduction

The mobile ad hoc network (MANET) permits a promising future for wireless networks, with the ability of establishing networks at anytime, anywhere. MANET is a wireless network that includes mobile nodes which has a dynamic topology without infrastructure or centralized management. Ad hoc is a workable solution for various applications and environments including emergency operations (e.g. disaster recover) and military environments. It is also easily deployed since it does not need expensive infrastructure like other conventional wireless networks do [1, 2].

Wireless network topology enables nodes to communicate wirelessly with each other within a fixed network that can be classified as either infrastructure or infrastructure-less network [3]. In an infrastructure network, wireless nodes communicate through a base station or an access point connected to the fixed network like the Internet. Figure 1 illustrates an example of infrastructure networks. Infrastructure networks have many advantages which include being reliable and is less likely to undergo any changes. Furthermore, the existence of a base station facilitates resource management and the routing process, as routing decisions among nodes are centralized [4]. In contrast to centralized networks, infrastructure-less networks or MANETs do not need access points or other base stations to communicate within the wireless nodes, as shown in Fig. 2. In MANETs, nodes locations have dynamic behaviour while communicating over wireless links. Typically, infrastructure-less networks are invoked in situations where support from networking infrastructure or administration is absent [4].

In MANET, the nodes have limited resources, and the connectivity to each other is determined by a particular time. This type of networks is devoted to provide connectivity among users, regularly raised in the area, where traditional infrastructures

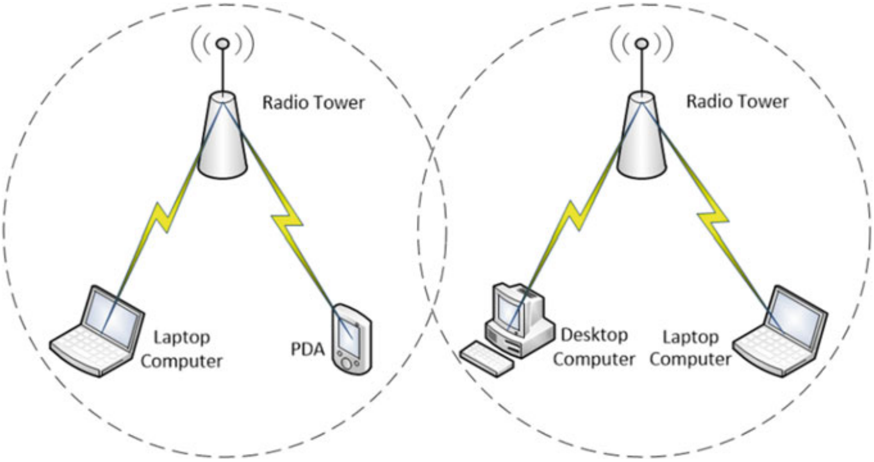


Fig. 1 Infrastructure wireless network [4]

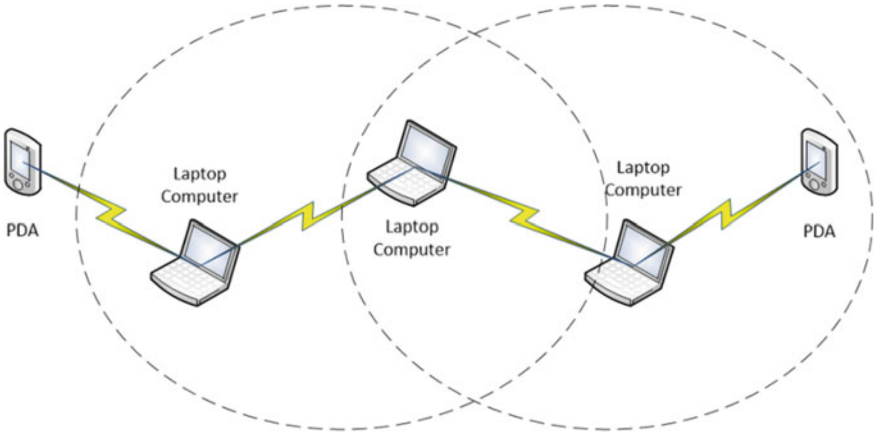


Fig. 2 Infrastructure-less wireless network [4]

for communication between nodes are not available. Also, nodes are not able to play the role of servers in MANETs all the time, due to its low capacity and battery energy limitations. The connection among nodes is established only at the time of communication and mostly disconnected from the network because of a network partition, link breakage and changes in topology. Routes are created in this network based on hop-by-hop method, but these are limited by the range of transmission of nodes. The data communication in MANETs is conducted by assigning a definite set of contents or data to a node that are shared in the network, and each node manages its own contents [5].

Routing protocols are categorized into three main types: the proactive, reactive and hybrid routing protocols. In the first category, a recent map of the network topology is maintained. In a proactive mechanism, the packet transmission from the source to the destination requires predefined routes to be available. While in the second routing protocols type, which is also named on-demand-driven routing protocols, the route availability is not continuous. Consequently, the data packet transmission invokes a procedure for discovering the route, which is done by sending requests that are flooded into the networks. The advantage of the first two categories can be integrated to consist of a hybrid routing approach. This is achieved by calling the reactive mechanism when a node does not exist within the transmission range and switches to a proactive technique if a node is within the range [6, 7]. Specifically, this paper is focused on optimizing two types of protocols that fall under reactive and proactive categories.

In the context of protocols optimization, the applications of the tools that are based on computational intelligence (CI) have grown dramatically since its initiation [8]. Previously, CI tools were limited to neural networks, fuzzy logic and evolutionary computing as well as their hybrid methods [9]. However, in recent years, swarm intelligence (SI) which is one of the CI classes is broadly used for many domains, in which the protocol optimization is one of them. SI is the collective behaviour of decentralized and self-organized systems. The classic SI system consists of a set of simple agents which can communicate with each other by acting on their local environments. Though the agents in a swarm follow very simple rules, the interactions between such agents can lead to the emergence of very complicated global behaviour, far beyond the capability of a single agent [10, 11]. The natural system of SI includes, but is not limited to, ant foraging, bird flocking and fish schooling. In specific, ABC is one of the SIAs, which mimics the behaviour of honeybee swarms. ABC received a lot of attention in various domains and specifically for protocols optimization [12].

In recent years, several works have been devoted to optimize the routing protocols based on finding the best path from the source to the destination based on ABC algorithm. These works employed the ABC algorithm for different types of protocols, i.e. bee-inspired protocol (BeeIP) [13], predicted energy-efficient bee-inspired routing (PEEBR) [14], sensorbee protocol [15] and energy-aware reliable routing protocol (EARRP) [16, 17].

Based on the previous works of ABC algorithm in routing protocols domain, we have investigated its efficiency to tackle the problem of finding the shorted path in reactive and proactive protocols. The rest of this paper is organized into four sections. The first one describes the routing protocols used in this study. The next one illustrates the proposed ABC for the used protocols. The third section shows the results obtained and a comparison of the effectiveness of ABC with other protocols. Finally, the conclusion includes some directions and guidance for future works in this field of study.

2 Energy-Aware Routing Protocol

Energy-aware routing protocols concentrate on the methods to reduce battery energy usage by diverting traffic from low-energy nodes to ones with high battery energy [17]. This approach increases the overall network lifetime by saving network nodes' battery life.

These routing protocols suggest the use of various metrics, whereas finding the routing path can help reduce the energy usage of the nodes. These metrics include minimizing energy consumption, minimizing variance in node energy levels, maximizing time to a network partition, minimizing cost or minimizing maximum node cost. These metrics are suggested for use instead of only finding the shortest route from the source node to the destination node [18]. The metric 'minimizing energy consumed' is useful for minimizing the overall energy consumption for delivering a data packet from the source node to the destination node.

If a node is overused for a certain period of time due to the transmission of packets through it, then the operational time of such nodes is reduced consistently, and the system leads to energy exhaustion. In turn, this process results in the disconnection of links and in network partitioning. Therefore, the best routing path is the one with good energy sources in its intermediate nodes even if it costs an additional hop count.

Maximizing the third metric, time to a network partition, is considered as the ultimate goal of a MANET because it directly addresses the lifetime of the network [19].

3 Routing Protocols in MANET

The process of exchanging information from one host to another in a network is referred to as routing. This also refers to the way of forwarding packets towards its destination using an effective path. The efficient performance of a path is determined based on specific criteria, such as traffic, number of hops, security and so on. In ad hoc networks, each host node represents a specialized router. Therefore, routing is identified as an essential problem in MANETs [20]. Designing an effective routing protocol encounter mainly the problem of maintaining and finding the best route from the source to the destination [21].

MANETs have many protocols to address the networks' typical limitations. Some of these typical limitations include high energy consumption, node mobility, high inaccuracy rates and low bandwidth. These routing protocols can be divided according to the routing information mechanism and are classified into three main categories (unicast, multicast and broadcast routing protocols) as shown in Fig. 3. Unicast routing protocols can be divided into reactive, proactive, hybrid, flow-oriented routing and adaptive routing protocols.

This work aims to find the shortest path using two routing protocols, namely Destination-Sequenced Distance-Vector Routing (DSDV) and Ad hoc On-Demand Distance-Vector Routing Protocol (AODV). This section describes in detail each of these protocols.

3.1 Destination-Sequenced Distance-Vector Routing

In 1994, Perkins and Bhagwat [22] developed a table-driven routing protocol known as DSDV for MANET based on the Bellman–Ford algorithm.

The Bellman–Ford algorithm has been adjusted further to address the problem of routing loops in the routing tables. Each node in this protocol maintains routing table. This routing information must be continuously updated. Routing table lists all available destinations, generally, all the possible destinations and the number of hops to each destination of every node are maintained in the routing table. Using such a routing table stored in each mobile node, the packets are transmitted between the nodes of an ad hoc network. Each node of the ad hoc network updates the routing table with advertisement periodically or when significant new information is available to maintain the consistency of the routing table with the dynamically changing topology of the ad hoc network. In the routing table, every entry is marked by a sequence number which is allocated by the destination node. Using the assigned number, the mobile node recognizes old and new routes, thus the routing loop problem

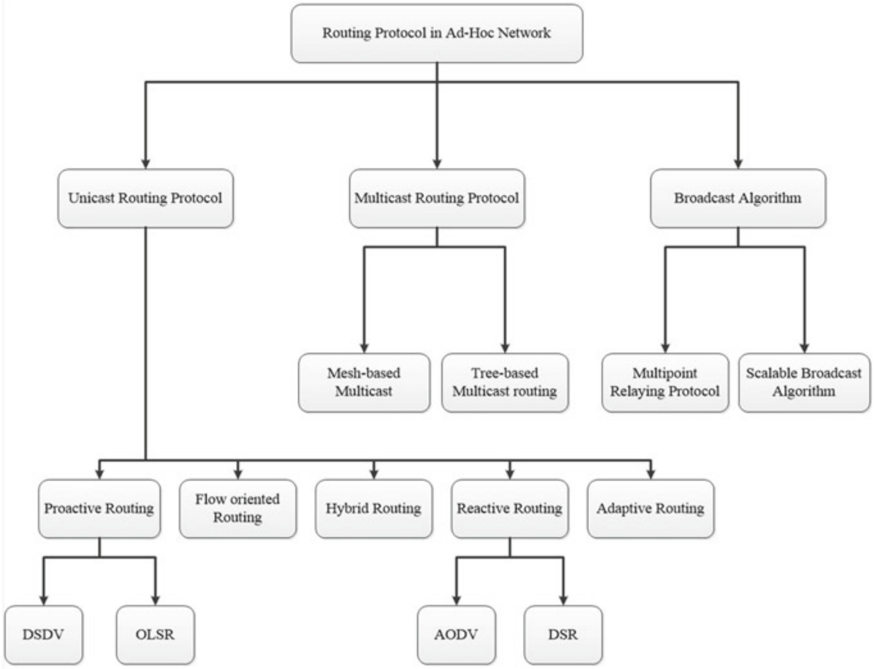


Fig. 3 Mobile ad hoc routing protocols [1]

is lowered. The node continuously uses the route associated with the most recent sequence number, and in case two routes hold the same sequence, the route with the smaller metric will be used to ensure an optimized path. The advantages of DSDV are attributed to the algorithm's simplicity, and it can be applied conveniently to few numbers of nodes. However, this protocol consumes a lot of battery even when the network is idle due to the continuous updates in the routing table [21, 23].

3.2 Ad Hoc On-Demand Distance-Vector Routing Protocol

AODV belongs to the reactive routing protocols, which is based on a unicast transmission mode. This protocol creates paths among nodes only as wanted by source nodes, and hence, it is an on-demand algorithm. It preserves these paths as long as they are required by the sources. In AODV, every node stores the information about active routes in the routing table. The stored data are a number of hops, next hop, destination, sequence number for the destination, the expiration time for a route table entry and active neighbours for a route. To find out and maintain links, AODV protocol uses three control packets, namely route requests (RREQs), route replies

(RREPs) and route error messages (RERRs). AODV consists of two mechanisms, which are route discovery and route maintenance [24–28].

The source node would first check its routing table when it needs to send data to the desired destination. If the route to the desired destination is found in the source node routing table, it starts to send data to the next hop. Otherwise, it begins a route discovery mechanism. In this mechanism, RREQs packets are sent throughout the network. Nodes receiving this packet would update their information for the source node and create reverse routing entries towards the source node in the route tables. In addition to the source node's IP address, broadcast ID and current sequence number, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. An RREP would be sent by the destination node or by an intermediate node that has a valid route to the destination. Nodes that receive RREPs also create reverse routing entries towards the nodes that sent the RREQs [24, 25, 27, 29].

Every node along an active route will send HELLO messages to all nearby nodes. The links between two neighbouring nodes would be broken if a node does not receive a HELLO message or any data. A local repair procedure may be used to repair broken links in a route if the destination of the route using this link is nearby the next hop from the neighbour. Otherwise, a RERR message would be sent to the nearby nodes, which then broadcasts the RERR message towards other nodes that may have routes affected by the broken link. If the route is needed by the affected source, the route discovery process will then be repeated [24, 25, 27, 30]. Figures 4 and 5 portray the RREQ packet and the RREP packet in AODV, respectively.

4 Artificial Bee Colony for AODV and DSDV

The proposed ABC applied to two types of protocols which are a unicast reactive routing protocol called AODV, and the other one is table-driven routing protocols named known as DSDV. This section describes the ABC algorithm and its application to DSDV and AODV.

In recent years, a lot of studies concentrated on the ABC algorithm due to its efficiency in solving the combinatorial optimization problem. ABC is inspired from the natural foraging behaviour of honeybee swarms [31]. In ABC algorithm, the artificial bee colony can be divided into three groups based on the functionality, which are employed bees, onlookers and scouts. The employed bees are filling the first half of the colony, and the second part includes the onlooker bees. These bees will be waiting on the waggle dance area to make a decision on choosing a food source. Onlooker bees watch the waggle dance of the employed bees within the hive to choose a food source [12]. While scout bees are those that conduct random searches to find new food sources. Figure 6 shows the process of the scout bee. In this figure, A, B and C are the food sources, and the scout bee begins searching randomly from one source to another [32].

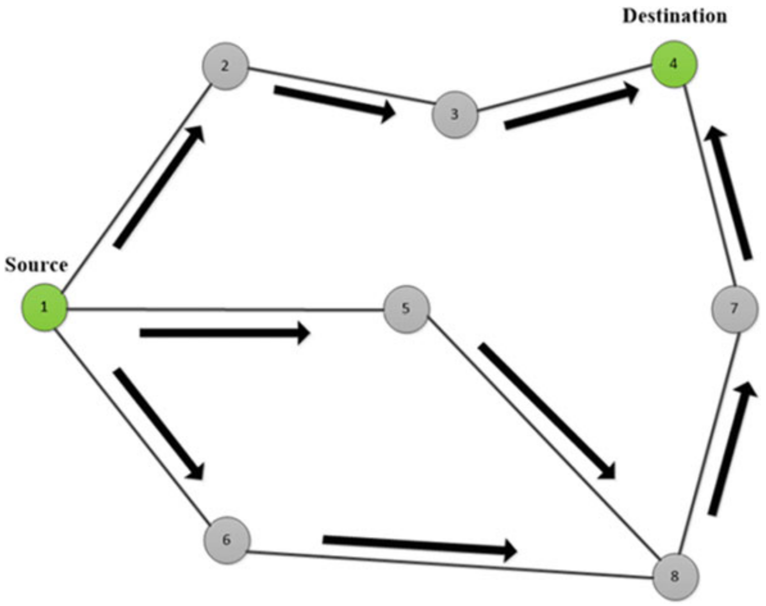


Fig. 4 AODV routing protocol RREQ packet [30]

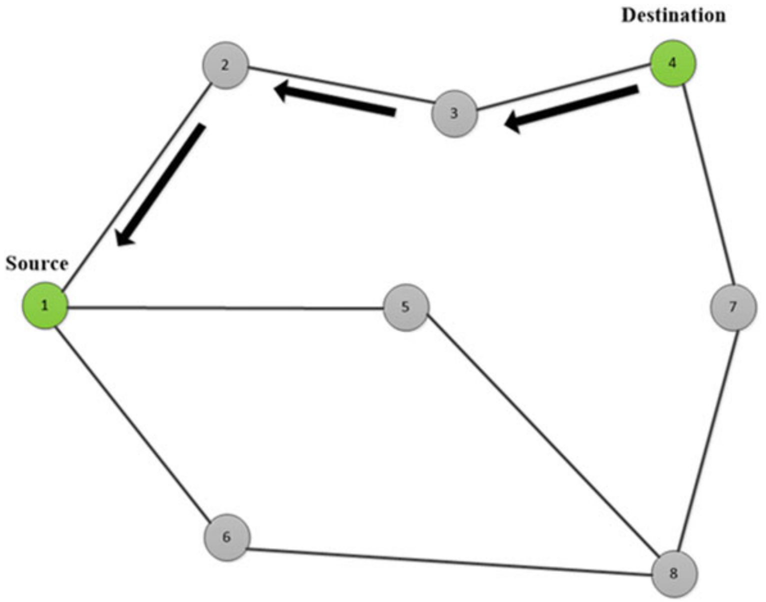
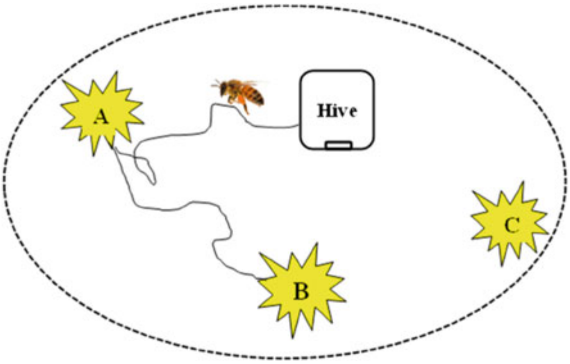


Fig. 5 AODV routing protocol RREP packet [30]

Fig. 6 Scouting process [32]



The population size of the employed bees equals to the number of food sources. The search process of artificial bees can be shown as follows:

- Employed bees detect a food source within the neighbourhood of the food source in their memory.
- Employed bees disseminate the information, i.e. distance, direction, and profitability, of food sources when returning to their hive with the onlooker bees.
- Onlookers select a food source within the neighbourhood of the food sources chosen by the onlookers themselves.
- Finally, if the source has been abandoned, the employed bees will be transformed to scout and start searching randomly for new food sources.

The search behaviour of ABC shown above can be mapped to algorithmic steps as follows:

- Initialize.
- REPEAT.
- The employed bees begin moving to their food sources to specify their nectar amounts.
- Onlooker bees watch the waggle dance and go to the sources of food according to their nectar amounts.
- Adjust the scouts to search for new sources of food.
- Throughout the search process, the best food will always be maintained.
- UNTIL (stopping criteria are met).

The searching cycle of food sources depends on three steps. First, the food source location will be regenerated and randomly distributed. Second, moving the employed bees onto the food source and calculating their nectar amounts [12]. Furthermore, when the employed bees return to the hive they share the information of these food sources (distance, direction and profitability) with the onlooker bees. Third, the forager bees continue visiting the same food sources that are already stored in their memory. Depending on the nectar amount, the onlooker bees will decide on the best food sources to visit so that they can extract the nectar from it. After a while, when the nectar becomes exhausted, these food sources will be abandoned since the position cannot be further improved over a number of cycles called limit. The value of limit in our method adapts the original value in ABC algorithm to be equivalent to a number of onlookers multiplied by the solution dimensions. Thereafter, a forager bee will become a scout bee, and it starts a new journey of discovering new food sources to replace abandoned ones. The ABC search procedure controlled by the colony size was determined experientially to 50 bees divided equally into onlooker and employed bees. Also, the stopping criterion is identified in the experiment to stop after 30 successive non-improved cycles.

The foraging behaviour of ABC is similar to find a path from a source to the destination. In the ad hoc network, all the nodes work cooperatively by sharing the information through the links and the partial routes among the nodes. The nature of ABC algorithm makes it suitable for dynamic and multi-objective problems. Due to the mobility of the nodes in MANET, the best path from the source to destination changes from time to time. Hence, the DSDV and AODV recruit the ABC to find the best path that minimizes energy consumption. In ABC, the solution produced by an employed bee via modification of its memory using the local information (visual information). Thereafter, the new solution (new source) is evaluated by measuring its energy consumption. An employed bee disseminates its position and energy information to the onlooker bees after the employed bees complete the search process. The onlooker bees evaluate the energy of the employed bees and then select the food source of the probability related to the energy amount.

5 Experimental Results

This section discusses the results obtained from using ABC algorithm for DSDV and AODV separately. The section is organized into three subsections discussing the simulation settings, performance metrics, and simulation results, respectively. In the simulation settings, the simulator and the ABC parameters are described. The metrics subsection, on the other hand, portrays the formulations of each result obtained. Finally, the simulation results subsection displays the yielded results, discuss and compare the results for each of the used protocols.

5.1 Simulation Settings

The results of the used protocols, i.e. DSDV and AODV, were obtained using a network simulation (NS2). This is developed to simulate a multi-hop wireless ad hoc environment achieved with medium, data link and physical access control (MAC) layer models. In this simulation, a network setting on OTcl script has been declared, such as the routing protocol, network traffic, propagation model, packet size and node speed to be used.

Table 1 Simulation parameters

Parameter	Value	Unit
Simulator	NS-2.35	–
Number of runs	**5**	–
Number of nodes	50	Node
Simulation area	1670 × 970	m^2
Routing protocols	DSDV and AODV	–
Mobility model	Random Way Point	–
Packet size	128, 256, 512 and 1024	Bytes
Node speed	5, 10, 15 and 20	m/s
Type of traffic	CBR	–
Initial energy	100	J
Idle energy consumption	0.05	mW
Transmission energy consumption	1.35	mW
Receive energy consumption	1.7	mW
Sleep energy	0.001	mW
Simulation time	50	s

To evaluate the performance of AODV and DSDV, two different scenarios were used. In the first scenario, nodes are allowed to move within the speed range off 5–20 m/s. The second scenario packet size has different sizes 128, 256, 512 and 1024 bytes. The mobility model used in simulating the ‘Random Way Point (RWP) model’, with dimensions of 1670 × 970 m, had 50 nodes. The rest of the settings used in the simulation model are shown in Table 1.

5.2 Performance Metrics

Throughout the simulation experiments, different performance metrics were used for the AODV and DSDV routing protocols. These metrics are defined as follows [33]:

(i) **Average End-to-End Delay (Average E2E delay)**
This metric refers to the time consumed to transmit successfully a packet of data from the source to the destination. E2E delay includes every possible delay such as the buffering during discovery latency of the route, propagation, retransmission delay at the MAC, time of the transfer and queuing at the interface queue. The average E2E delay can be calculated using the following formula:

$$\text{E2E delay} = \frac{\sum_{i=1}^{n} (R_i - S_i)}{n}$$

where n is the number of packets successfully broadcasted, i is the unique packet identifier, R_i is the total packet received, S_i is the total packet sent and n is the number of data packets.

(ii) **Packet Delivery Ratio (PDR)**

PDR is the ratio of data packets received by destinations divided to packets sent by the sources. It is calculated using the following formula:

$$\text{PDR} = \frac{\text{Received packets}}{\text{Transmitted packets}} * 100 \tag{2}$$

(iii) **Average Energy Consumption**

It is measured as the ratio of average energy consumed by each node in the network. The average energy consumption is calculated as follows:

$$\text{AVG Energy Consumption} = \frac{\text{Consumed Energy}}{\text{Initial Energy}} * 100 \tag{3}$$

(iv) **Packet Loss Ratio (PLR)**

PLR is the ratio of the difference between the data packets sent and data packets received to the data packets sent by the source. It is calculated using the following formula:

$$\text{PLR} = \frac{S - R}{S} * 100 \tag{4}$$

where S refers to the data packet sent, while R is the data packet received.

5.3 Simulation Results and Performance Comparison

The simulation was carried out using NS-2.35, and the compression of the performance analysis was conducted using average energy consumption, average end-to-end delay, packet delivery ratio and packet loss ratio evaluation metrics. To demonstrate the effectiveness of the DSDV and AODV routing protocols using ABC, a simulation study was conducted using different node speeds and packet sizes. Initially, the nodes were placed at certain specific locations, and then, the nodes move towards new locations. Five simulations were carried out for each of the two protocols. In order to evaluate the performance of ad hoc network routing protocols, the following parameters were considered:

Fig. 7 E2E delay based on node speed

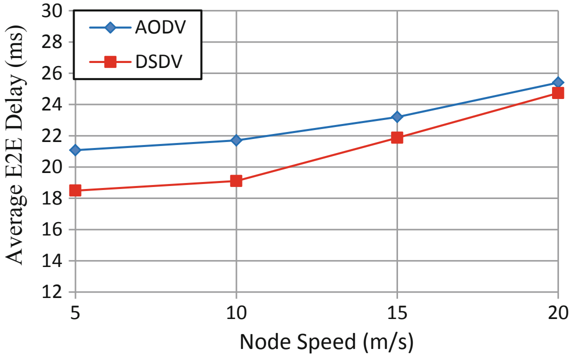
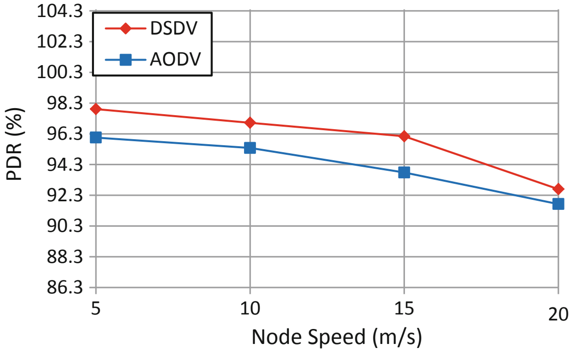


Fig. 8 PDR based on node speed



(i) Impact of node speed

The average E2E delay tested was from the time the data packet was sent by the source until the time the data packet was received at the destination. Figure 7 shows the average E2E delay for the DSDV and AODV protocols as a function of the node speed. The result clearly shows that the average E2E delay increased when the mobility of the nodes increases from 5 to 20 m/s in both of the DSDV and AODV routing protocols. DSDV keeps a steady performance and slightly deteriorating as the mobility of the nodes increases. It can be observed that DSDV performs better than AODV.

In terms of PDR in Fig. 8, the routing protocols studied decreased in a similar manner. The result clearly shows the PDR decreased when the speed of the nodes was increased from 5 to 20 m/s in both of the DSDV and AODV routing protocols. DSDV is, again, proven to perform better than AODV.

Figure 9 presents the average energy consumption for DSDV and AODV routing protocols. Due to the highly dynamic network when the node speed range is 5–20 m/s, AODV has less energy consumption than the DSDV. Average energy consumption increases when the nodes' speed increases from 15 to 20.

Fig. 9 Energy based on node speed

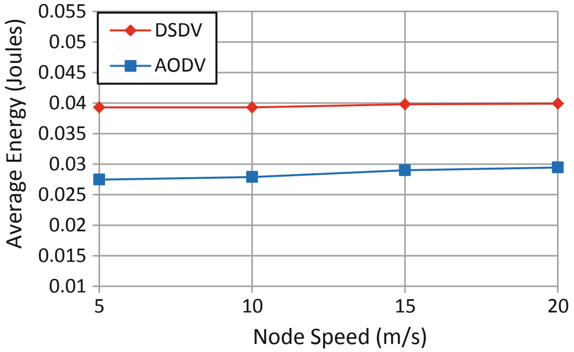


Fig. 10 PLR based on node speed

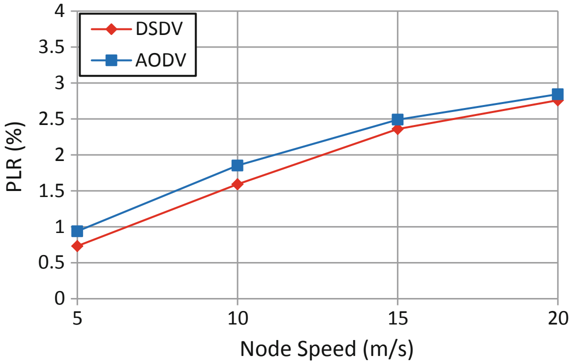


Figure 10 shows the number of packet loss ratio for the two routing protocols as a function of the mobility of nodes speed. The result clearly shows the PLR increased when the mobility of the nodes increases in both of the DSDV and AODV routing protocols. It can be observed that the nodes' speed has less effect on the performance of the DSDV as compared to AODV.

Obviously, the node speed affects overall performance protocols. Also, it affects each of the tested metrics. Generally, the high speed of the nodes has a negative impact on the calculated metrics. As this study is concerned about energy consumption, the node speed is directly proportionate to the energy consumption. Hence, the high speed reflects a high energy consumption and vice versa. The AODV shows less energy consumption as compared to the DSDV in terms of the use of node speeds.

(ii) **Impact of Packet Size**

Average E2E delay is affected by high rate of packets transfer as well. The buffers become full much quicker before the packets sent it has to stay in the buffers for a period of time. This can be seen at the AODV routing protocol. Figure 11 shows the average E2E delay for the DSDV and AODV protocols as a function of the packet size. The performance result shows the average E2E delay slightly deteriorating when the packet size increased in both of the DSDV and AODV routing protocols. This

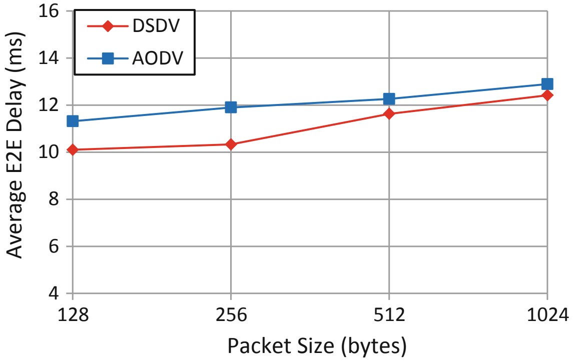


Fig. 11 E2E delay based on packet size

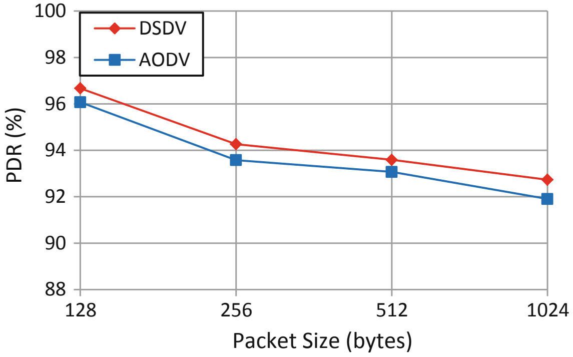


Fig. 12 PDR based on packet size

is because of the extremely high data rate and low mobility. It can be observed that DSDV has less delay compared to AODV.

Figure 12 presents the performance of the DSDV and AODV protocols as a function of the packet size. The result of PDR decreased when the packet size increased in both routing protocols. It can be seen from the graph that the DSDV is able to perform better than AODV.

Figure 13 depicts that the energy consumption for AODV and DSDV routing protocols is increased especially from the packet size 256 bytes. It can be observed that AODV outperforms DSDV. The energy consumption for DSDV protocol increases for each successful data delivery because the probability of link breakage also increases. That needs additional energy to establish new routes.

Figure 14 shows the number of packet loss ratio for the AODV and DSDV protocols as a function of packet size. The result of PLR increased when the packet size increased. AODV has higher PLR than DSDV protocol. As the PDR decreased means that more packet loss in the network.

In the context of packet size impact, the AODV and DSDV protocols show retrograde performance in the bigger packet size. The energy consumption in both protocols increased with each increment in the packet size. However, AODV is less

Fig. 13 Energy based on packet size

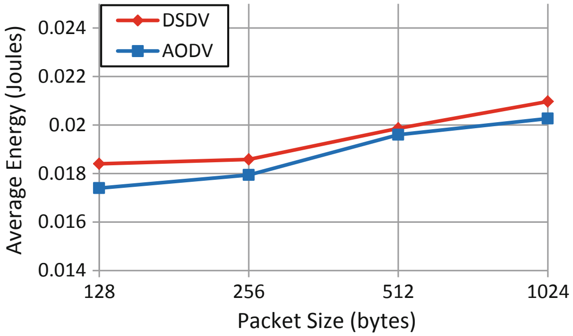
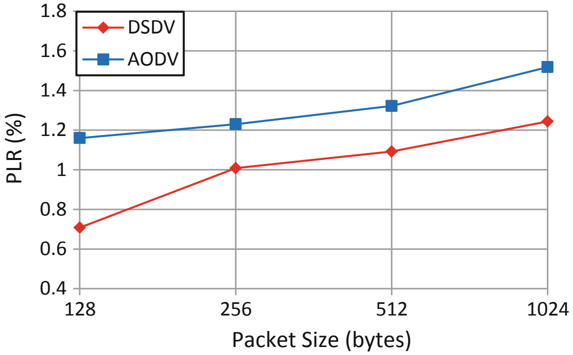


Fig. 14 PLR based on packet size



energy consuming at each point of the packet size increment. In conclusion, AODV outperforms the DSDV in term of energy saving under the changes in node speed and packet size.

6 Conclusion

The proposed study in this paper would work on optimizing two ad hoc routing protocols AODV and DSDV, based on a swarm intelligence method, namely artificial bee colony (ABC) algorithm. The ABC for protocol optimizations mimics the behaviour natural bees foraging, to determine the best routing paths for reducing the energy consumption that is caused by overloading packets and nodes speed. Our comparison is based on two testing network parameters which are node speed and packet size. We conclude that the node speed and packet size have a severe effect on the performance metrics of the studied routing protocols. From the energy perspective, we found that AODV outperforms the DSDV in terms of energy consumption under the changes in node speed and packet size. Due to continuous updates in the routing table, DSDV consumes a lot of battery even when the network is idle. On the other hand, AODV routing protocol makes update only when the packet sends from source to the destination.

References

1. Safdar M, Khan IA, Ullah F, Khan F, Jan SR (2016) Comparative study of routing protocols in mobile adhoc networks. Int J Comput Sci Trends Technol ISSN 2347–8578
2. Zhao X, Hung WN, Yang Y, Song X (2013) Optimizing communication in mobile ad hoc network clustering. Comput Ind 64:849–853
3. Duggi MR (2008) Apparatus and method for collecting active route topology information in a mobile ad hoc network. Google Patents
4. Taneja S, Kush A (2010) A survey of routing protocols in mobile ad hoc networks. Int J Innov Manage Technol 1:279
5. Amin R, Akhtar MB, Khan AA (2010) Analyzing performance of ad hoc network mobility models in a peer-to-peer network application over mobile ad hoc network. In: 2010 International conference on electronics and information engineering (ICEIE), pp V2-344–V2-348
6. Abolhasan M, Wysocki T, Dutkiewicz E (2004) A review of routing protocols for mobile ad hoc networks. Ad Hoc Netw 2:1–22
7. Raut SH, Ambulgekar HP (2013) Proactive and reactive routing protocols in multihop mobile ad hoc network. Int J Adv Res Comput Sci Softw Eng 3:152–157
8. Alba E, Dorronsoro B, Luna F, Nebro AJ, Bouvry P, Hogie L (2007) A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan MANETs. Comput Commun 30:685–697
9. Tettamanzi AG, Tomassini M (2013) Soft computing: integrating evolutionary, neural, and fuzzy systems. Springer Science & Business Media
10. Engelbrecht AP (2006) Fundamentals of computational swarm intelligence. Wiley, New York
11. Zhang S, Lee CK, Chan HK, Choy KL, Wu Z (2015) Swarm intelligence applied in green logistics: a literature review. Eng Appl Artif Intell 37:154–169
12. Karaboga D, Gorkemli B, Ozturk C, Karaboga N (2014) A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artif Intell Rev 42:21–57
13. Giagkos A, Wilson MS (2014) BeeIP–A Swarm Intelligence based routing for wireless ad hoc networks. Inf Sci 265:23–35
14. Choudhury D et al (2015) Energy efficient routing in mobile ad-hoc networks. In: 2015 International conference and workshop on computing and communication (IEMCON). IEEE
15. Sridhar H, Siddappa M, Prakash GB (2013) Power aware routing protocol for MANET's using swarm intelligence. Power 2
16. Phil M, Arumugam N (2012) Energy aware reliable routing protocol (EARRP) for mobile ad hoc networks using bee foraging behavior and ant colony optimization
17. Mohan BC, Baskaran R (2011) Energy aware and energy efficient routing protocol for adhoc network using restructured artificial bee colony system. In: High performance architecture and grid computing. Springer, Berlin, pp 473–484
18. Feng D, Jiang C, Lim G, Cimini LJ, Feng G, Li GY (2013) A survey of energy-efficient wireless communications. IEEE Commun Surv Tutor 15:167–178
19. Manweiler J, Santhapuri N, Sen S, Choudhury RR, Nelakuditi S, Munagala K (2012) Order matters: transmission reordering in wireless networks. IEEE/ACM Trans Netw 20:353–366
20. Jiang D, Xu Z, Li W, Chen Z (2015) Network coding-based energy-efficient multicast routing algorithm for multi-hop wireless networks. J Syst Softw 104:152–165
21. Loo J, Mauri JL, Ortiz JH (eds) (2016) Mobile ad hoc networks: current status and future trends. CRC Press, Boca Raton
22. Perkins CE, Bhagwat P (1994) Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. ACM SIGCOMM Comput Commun Rev 24(4):234–244

23. Valivety S (2009) Affect of handover on the performance of routing protocols in WiMax. California State University, Long Beach
24. Shaheen A, Gaamel A, Bahaj A (2016) Comparison and analysis study between AODV and DSR routing protocols in VANET with IEEE 802.11 b. J Ubiquitous Syst Pervasive Netw 7(1):07–12
25. Liu J, Wan J, Wang Q, Deng P, Zhou K, Qiao Y (2016) A survey on position-based routing for vehicular ad hoc networks. Telecommun Syst 62(1):15–30
26. Royer EM, Perkins CE (1999) Multicast operation of the ad-hoc on-demand distance vector routing protocol. In: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pp 207–218
27. Xiong W, Li Q-Q (2015) Performance evaluation of data disseminations for vehicular ad hoc networks in highway scenarios. Int Arch Photogrammetry Remote Sens Spat Inf Sci 37
28. Tareq M, Alsaqour R, Abdelhaq M, Uddin M (2017) Mobile ad hoc network energy cost algorithm based on artificial bee colony. Wirel Commun Mob Comput 2017:1–14
29. Royer EM, Toh C-K (1999) A review of current routing protocols for ad hoc mobile wireless networks. Personal Commun IEEE 6:46–55
30. Tariq M, Fareed H, Alsaqour R (2016) Performance analysis of reactive routing protocols in mobile ad hoc network using NS2
31. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report-tr06. Erciyes University, Engineering Faculty, Computer Engineering Department
32. Sivakumar D, Suseela B, Varadharajan R (2012) A survey of routing algorithms for MANET. In: 2012 International conference on advances in engineering, science and management (ICAESM), pp 625–640
33. Macker J (1999) Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations
34. Zafar S, Tariq H, Manzoor K (2016) Throughput and delay analysis of AODV, DSDV and DSR routing protocols in mobile ad hoc networks. Int J Comput Netw Appl (IJCNA) 3(2):1–7

A Novel Chicken Swarm Neural Network Model for Crude Oil Price Prediction

Abdullah Khan, Rahmat Shah, Junaid Bukhari, Nasreen Akhter, Attaullah, Muhammad Idrees and Hilal Ahmad

Abstract For optimization applications, an innovative bio inspired algorithm of Chicken Swarm Optimization (CSO) is suggested, the CSO represents hierarchy of chicken swarm. Chicken Swarm Optimization extracts the chickens swarm intelligence that can be used efficiently to optimize problems. This research investigates performance of proposed model Chicken-Swarm Optimization in hybrid with neural network (Chicken S-NN) to find the local minima and slow convergence. Performance of the Chicken S-NN model is compared with ABCNN (Artificial Bee Colony Neural Network) and ABCBP (Artificial Bee Colony Back-Propagation). From the results of training and tested data, this is evident that the proposed (Chicken S-NN) algorithm performs better than the other models with respect to accuracy and Mean Square Error (MSE).

A. Khan (✉) · J. Bukhari · Attaullah · H. Ahmad
Institute of Business and Management Science, Agricultural University, Peshawar, Pakistan
e-mail: Abdullahdirvi@gmail.com

J. Bukhari
e-mail: baba.khan993@gmail.com

Attaullah
e-mail: Atta12319@yahoo.com

H. Ahmad
e-mail: hilal_a24@yahoo.com

R. Shah
Department of Computer Science, CECOS University of IT & Emerging Sciences, Peshawar, Pakistan
e-mail: rahmat@cecos.edu.pk

M. Idrees
Department of Agriculture Extension Education and Communication, Agricultural University, Peshawar, Pakistan
e-mail: dridreesext@yahoo.com

N. Akhter
Department of Computer Science, FAST National University, PFD Campus, Islamabad, Pakistan
e-mail: nasreen.akhtar@nu.edu.pk

T. Herawan et al. (eds.), *Advances on Computational Intelligence in Energy*,
Green Energy and Technology, https://doi.org/10.1007/978-3-319-69889-2_3

Keywords Chicken swarm optimization · Local minima · Back-propagation · Artificial Bee Colony

1 Introduction

Forecasting the consumptions of crude oil prices has become one of the hot questions in the area of research due to its wide usage and importance in the world economic arrangements. Crude Oil price has been a part of the process of decision-making for the growth in industries and the government planning in the short-term and long-term, export strategy and for national reserves [1].

Generally, crude Oil prices forecast shows the future possibility in development of relevant marketplace. Crude oil is crucial to the world economy. Similarly from industrial point of view, it is vigorous production inputs while the prices are controlled by supply and demand. Changes in crude Oil prices are driven at various levels with collective or precautionary demands that are linked to the market concerns about the availability of oil supplies in the future [2].

Unpredictability of the crude Oil prices has been debated and gagged in numerous researches. The determination of this unpredictability in the prices of crude Oil gets more importance for the periods during which crude Oil prices show fast and sudden variations. Secondly, to predict the prices of crude Oil gets more significance in the future, especially in periods of high volatility. There is considerable works aimed in improving the abilities of econometric models to model the price of crude Oil [3]. Many articles have already been successfully submitted ANN applications for modeling and predicting a series of crude oil prices likewise for predicting time series of the real world [4–10].

ANN offers other approaches to solve complex problems. In many different ANN models, the most famous ANN is Multi layer Feed Neural Network (MLFF) and Recurrent Neural Network (RNN). Feed forward neural network sends unidirectional signals, from the level of input, hidden layers to output layer. While Recurrent Neural Network hold a short term memory for activation feedback [11]. The network has local feedback loops allowing storing previous state of hidden unit. This opportunity makes it progressive as compared to a feed forward network, because the network outputs are calculated from current inputs and hidden states, so it has at least one feedback link [12].

The results of experiments prove that ANN performance is much higher in numerous conventional models used for statistical purposes. Ability of ANN is to learn complex as well as nonlinear time-series that are hard to using the traditional models. ANN also has some disadvantages such as the performance of ANN is inconsistent in some explicit situation. ANN can do accurate prediction [13]. While RNN can perform particular nonlinear and dynamic behaviour. Which is applicable in different area such are classification models of space and time, electrical power demand, financial data forecasting [14], associative memories, forecasting, management optimization and simplify sequences of patterns [12].

The complex application such as time-series prediction and nonlinear modeling makes RNN topologically more powerful and compact than FFNN. For complex problem fully recurrent networks are still complex, so using the partially recurrent networks, where the connections have particularly feedback, but contain selected set of association's feedback. Without any complication such model allows the system to memorize past history. Elman neural network is an example of partial RNN [11].

Some features make RNN algorithms less efficient and often requires a huge amount of time to form a network of even for moderate size of problem. RNN has complex error surface which makes many other training algorithms more flat stuck in local minima [14]. An artificial neural network is famous algorithm because of its ability to simplify and solve problems in function approximation, pattern classification, pattern matching and associative memories as well as its flexibility and competence. The success of ANN depends largely on their design, learning algorithm, and the choice of structures used during training [15]. ANN has ability to solve complex problem by classification prediction and forecasting nonlinear system and huge datasets with high accuracy. Without having prior knowledge, the neuron store, recognized, estimates and adopt new patterns on training, which make it superior and attractive model.

To overcome the weakness ANN many evolutionary techniques and global search techniques such as genetic algorithm, Ant Bee Colony (ABC), Particle Swarm Optimization (PSO), Cuckoo Search and Bat algorithm were used that formulates a kind of neural network more superior than conventional neural network models. CS algorithm provides more optimal solution then an efficient particle swarm algorithm and genetic algorithm [15]. Different modification have been done by various researchers such as GANN [16], CSBP, [17], BAT-BP, [18], ACPSOBP [14].

But this research proposed a new technique Chicken Swarm optimization (CSO) algorithm probes to boost the performance of neural network (NN). CSO development was credited to Xianbing Mengin 2014. Through observations of an individual or whole flock of chicken, the researchers extracted the chicken that have of mature intellectual ability, communication-skills and ability to learn and they exists in a strict hierarchical order in the flock. The CSO has the characteristics of simplicity and scalability, is a natural multi-swarm adaptive algorithm. CSO, a type of stochastic optimization algorithm that uses repetitive approach for solving the objective problem. CSO can only not handle continuous problems as well as discrete problems. In addition, configuration of parameters is simple [19].

The proposed model is trained on crude oil data to check the performance and then compared with others similar algorithms. The next remaining sections of the paper is organized such is Sect. 2, and Sect. 3, give the basic idea of Neural Network and Cuckoo search algorithm. Proposed technique is discussed in Sect. 4. Section 5, will represent experimental setup and results and discussion. Section 6, presents conclusion.

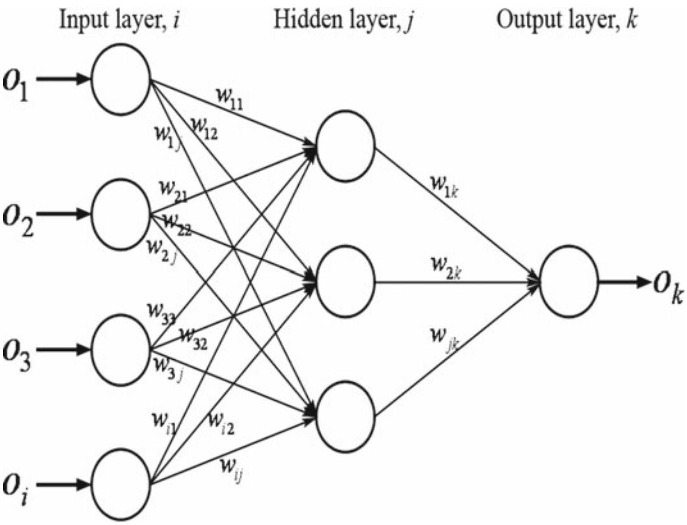


Fig. 1 Artificial Neural Network [25]

2 Artificial Neural Network

The concept of ANN is an engineering approach which is inspired by the concept of biological neurons. These neurons consist of inputs and outputs neurons that are interconnected with each other. Each neuron receives data from other neurons and weight of neurons link is added and output is generated for target neurons by applying activation function [20]. Layers of input feeds input to hidden layers and move forward direction to nodes in output layer. Nodes of input layer are configured based upon variables that are independent in dataset while dependent variables determined the output nodes. [21, 22]. There may more than one layer, but theoretical works such as [23] argues stating this that one hidden layer is enough for approximation any complex non linear function. Nodes in hidden layer is usually determined using trial an error [24]. Structure of ANN is shown in Fig. 1. ANN algorithm is used to process information in parallel. This can also be applied to model complex and non linear relationships through input output training from data sets that are collected from application domain.

The inherent capabilities of the Neural Network allow the algorithm to facilitate a non linear mapping of both input and output vectors [22, 25].

3 Chicken Swarm Optimization

Chicken Swarm optimization a new bio inspired algorithm developed by Meng et al. [26]. The Chicken Swarm optimization algorithm selected the behaviors of chickens by the following rules. For simplicity, the behavior of chickens can be idealized in the procedures [27].

(1) In a chicken flock, there are several groups and each group consists of a rooster, some hens and chickens.
(2) As the chicken flock can be further divided into different groups and establish the identity of chickens, hens (hens and chicks). They all are dependent on the values of fitness of the chickens. The chicken with the best range of fitness values precede as roosters. The worst chickens with different health values will be elected as the chicks. Rest is to be considered as hens. Hens selects group randomly that which group to join. The relationship of mother and their child is also determined randomly.
(3) The hierarchy, relations of domination and the relationship between parents and children in a group remain unchanged.
(4) Chicken rate after network group to find food, and can avoid those that eats their own food. Assuming steal chickens were randomly good food that is found by others. Chicks look for food near to their own mother (hen). Prominent one has a benefit in the race for food.

The pseudo code of the Chicken Swarm optimization algorithm is

```
Initialize a population of N chickens and define the related parameters;
Evaluate the N chickens fitness values, t = 0;
While MSE < Stopping Criteria
    If (t % G = 0)
        Rank the chickens' fitness values and establish a hierarchal order in the swarm;
        Divide the swarm into different groups, and determine the relationship between
        the chicks and mother hens in a group;
    End if
    For i = 1 : N
        If i == rooster
            Update its solution/location using Eq. (1);
        End if
        If i == hen
            Update its solution/location using Eq. (3);
        End if
        If i == chick
            Update its solution/location using Eq. (6);
        End if
        Evaluate the new solution;
        If the new solution is better than its previous one, update it;
    End for
End wile
```

4 The Proposed Chicken S-NN Algorithm

Chicken swarm optimization is bio inspired constructed on behavior of chicken swarm. Like other bio-inspired algorithms of global research metaheuristic keep both local search and global search. Chicken swarm is composed of distinct groups and every individual group revising chicks, hens, and a rooster. Head rooster is a chicken with excellent value given the conditions. In mean while the worst chickens with peculiar fitness values are considered as chicks. The remaining is seen as chickens. Chicken will try the best food around parent (Hen). Chickens without exceptions indicate rational solutions in the proposed CS-NN algorithm (i.e. weight density and linked basis to optimize neural network in this investigation). The weight optimizing as well as the sizes of pack dilemma imitates the solution for various application [28]. Weight of optimization problem and size of the populace serve as quality of solution. In the beginning cycle, the initial values are taken for weights and biases for the said Chicken S-NN algorithm and pass these weights values to NN. In the following approach, Chicken swarm will modify weights and the value of biases with probable nest or solutions and the search will be continued by Chicken swarm for best possible solution or research for best possible weight value till last round of network or objective MSE is obtained.

The proposed Chicken S-NN algorithm Pseudo code is as under:

```
Step 1: initializes Chicken swarm population
Step 2: Load the training data
Step 3: While MSE < stopping criteria

    If ($t \% G = 0$)
    Rank the chickens' fitness values and establish a hierarchal order in the swarm;
    Divide the swarm into different groups, and determine the relationship between the
    the
    chicks and mother hens in a group; End if
    For $i = 1$ : N
    If $i$ = rooster Update its solution/location using Eq. (1);
```

$$x_{i,j}^{t+1} = x_{i,j}^{t} * \left(1 + rand\left(0, \partial^2\right)\right) \tag{1}$$

$$\partial = \begin{cases} 1 \quad \text{if } f_i \le f_k \\ \exp\left(\frac{(f_k - f_i)}{|f_1| + \varepsilon}\right), & otherwise,\ k\varepsilon[1, N]\, K \ne i. \end{cases} \tag{2}$$

where rand($0 \sim \partial^2$) is a Gaussian distribution with mean 0 and standard deviation ∂^2. ϵ which is used to avoid zero-division-error, is the smallest constant in the computer. k, a rooster's index, is randomly selected from the roosters group, f is the fitness value of the corresponding x. End if

If i == hen Update its solution/location using Eq. (3)

$$x_{i,j}^{t+1} = x_{i,j}^{t} + S_1 * rand * \left(x_{r1,j}^{t} - x_{i,j}^{z}\right) + S_2 * \text{rand} * \left(x_{r2,j}^{t} - x_{i,j}^{t}\right) \tag{3}$$

$$S_1 = \exp((f_i - f_{r1})/(abs(f_i) + \varepsilon)) \tag{4}$$

$$S_2 = \exp(f_{r2} - f_i) \tag{5}$$

```
    End if
    If $i =$ chick Update its solution/location using Eq. (6)
```

$$x_{i,j}^{t+1} = x_{i,j}^{t} + FL * \left(x_{m,j}^{t} - x_{i,j}^{t}\right) \tag{6}$$

```
End if
Evaluate the new solution;
If the new solution is better than its previous one, update it;
End for
Step 4: Pass the chicken as weights to network
Step 5: Feed forward neural network runs using the weights initialized with chicken swarm
Step 6: Calculate the error
Step 7: Chicken Swarm keeps on calculating the best possible weight at each epoch until the network is converged.
End While
```

5 Results & Discussion

Primarily, the core concept behind this research is to improve slow convergence and obtains high accuracy. Before heading towards the results of simulation some terminologies used suchas structure of network, tools and technologies, as well as the dataset used for experiments have to be explained. Discussion of these are given below.

5.1 Preliminaries

Workstations used for experimentations comes equipped with a processor 2.33 GHZ (Intel Core i5 Processor) and Microsoft Windows 7 is used as Operating system. MATLAB version 2012 is used for simulation purposes. In comparison evaluation, learning rate and hidden layers are kept same. Three layers feed-forward neural networks have been used for models training. 5 nodes are fixed for hidden layers. Whereas the activation function used is log sigmoid function as transfer function from input till hidden layers and from hidden up to output layers. For evaluation of each problem, 1000 Epochs is restricted for solitary trial. 25 trails are executed for every value to validate the algorithm. Result of each trail is saved in result file. Number of failures as well as Mean Square Error (MSE) is documented in one by one distinct files for independent trial on crude oil classification.

5.2 Data

The dataset for OPEC petroleum consumption in thousands of barrels per day (tb/d) are obtained from the EIA of the US Department of Energy, which is freely available online on the organization's official website [29]. Dataset are collected for the domestic petroleum consumption of each 12 OPEC member countries, including Algeria, Angola, Ecuador, Iran, Iraq, Kuwait, Libya, Nigeria, Qatar, Saudi Arabia, UAE, and Venezuela. Also, the total petroleum consumption for OPEC is collected, given that it is the target forecasting value. The dataset are collected on yearly basis from 1980 to 2012 [30]. This research is based upon predicting OPEC of crude oil. Two different experiments are carried out on data division. Firstly the data is divided in two ways. In the first of applied technique, data is partitioned into 70 and 30%. 70% of the given data is used as training set and the rest of 30% data are used as testing set. While using second technique, data is partitioned into two slices of 60 and 40%. The first slice of 60% is used for training and on the other hand 40% of the remaining data is used for testing the data.

The dataset is normalized into a specified range of [0, 1] with the help of Eq. (7) to improve accuracy and convergence speed of prediction.

$$n_0 = \frac{k_{i-x_{\text{min}}}}{P_{\text{max}-x}\text{min}} \tag{7}$$

where

no normalize dataset,
ki raw dataset,
$xmin$ lowest value of dataset,
$pmax$ largest value of dataset [25].

5.3 Discussion

This section illustrates the performance of proposed Chicken Swarm Neural Network (Chicken S-NN) algorithm which is compared with Artificial Bee Colony Neural Network (ABCNN) and Artificial Bee Colony Back Propagation (ABCBP). The performance of the mentioned algorithm is measured in term of Epochs, Accuracy, Mean Square Error and CPU Time that can be seen in Table 1.

Table 1 Epochs, CPU time, accuracy and MSE for 12-5-1 NN-structure

Algorthms	Epochs		CPU time		Accuracy		MSE	
	Training (70%)	Testing (30%)	Training (70%)	Testing (30%)	Training (70%)	Testing (30%)	Training (70%)	Testing (30%)
Chicken S-NN	30	127	86.5244	352.7355	99.99807	99.99888	0.000101	0.0001104
ABCNN	1000	1000	2134.577	1484.712	99.87918	99.82674	0.001202	0.001311
ABCBP	1000	1000	2087.435	2784.457	99.81945	99.70366	0.001271	0.000544

From Table 1 this can be viewed that the performance of proposed Chicken S-NN algorithm is better on crude oil data set as compared to ABCNN and ABCBP. From these results, this is evident that for 70% of training data, the proposed Chicken S-NN algorithm converges to the global minimum in 30 Epochs, consuming 86.52 s of CPU time and attain 99.99 accuracy with an average MSE equals to 0.0001. While the other models such as ABCNN obtain an accuracy of 99.87 together with an average MSE is 0.0012. Similarly, the accuracy of ABCBP is 99.81 with the average MSE is 0.001271.

Furthermore for 30% testing data, the proposed Chicken S-NN algorithm performs well and converges to the global minimum in 127 Epochs obtaining 99.99 percent of average accuracy with an MSE equals to 0.000104. The accuracy of ABCNN and ABCBP is 99.82674 and 99.70366 respectively, with an average MSE of ABCNN is 0.001311 and ABCBP has an average MSE equals to 0.000544. Results of different models applied on data set has been illustrated below.

Figures 2, 3 and 4 display convergence performance of suggested Chicken S-NN algorithm, ABCNN and ABCBP on 70% Training data set of crude oil dataset.

The convergence of proposed Chicken S-NN algorithm, ABCNN and ABCBP can be seen in Figs. 1, 2 and 3 together with their MSE.

Figures 1, 2 and 3 shows that the proposed Chicken S-NN algorithm performs well and reaches to global minima in 30 Epochs.

Figures 5, 6 and 7 showed the Convergence Performance of proposed Chicken S-NN algorithm, ABCNN and ABCBP on 30% testing data of crude oil.

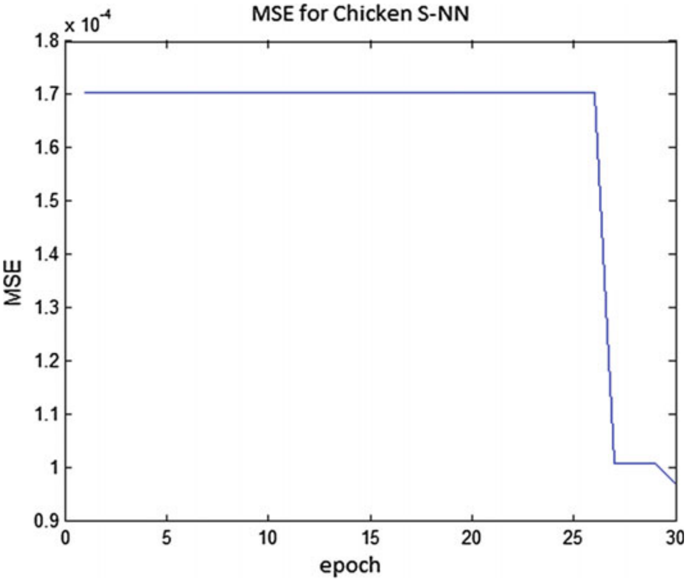


Fig. 2 MSE convergence performance of Chicken S-NN algorithm for 70% training dataset

The accuracy of proposed Chicken S-NN algorithm, ABCNN and ABCBP can be seen in Figs. 5, 6 and 7 together with their MSE.

Figures 5, 6 and 7 show that the proposed Chicken S-NN algorithm performs better and converges to the global minima in 127 Epochs.

Figures 8 and 9 displays a comparison of actual and predicted value of proposed Chicken S-NN algorithm. This can be seen clearly that the predicted value of proposed Chicken S-NN algorithm is near to the actual value.

Table 2 represents the performance of proposed Chicken S-NN Algorithm, ABCNN and ABCBP algorithm. Results show that the proposed Chicken S-NN algorithm performs superior than rest of the models. This is apparent from results that for 60% of training data, the proposed Chicken S-NN algorithm meets to global minima in 15 Epochs while consuming 47 s of CPU time with an average accuracy matches to 99.99% and 9.23E−05 is their average MSE. Whereas ABCNN model shows 99.50 average accuracy and reach an MSE of 0.000366. The accuracy of ABCBP model is 99.60 and gets an MSE of 0.004226.

Similarly on 40% testing data Chicken S-NN algorithms perform better and reaches to the global minima in 36 Epochs while gaining 99.99 percent of average accuracy and MSE equals to 0.004226. The other two models ABCNN and ABCBP gain 99.80 and 99.30 average accuracy respectively. ABCNN achieve MSE of 0.001054 while ABCBP gets 0.004226 averages MSE.

Figures 10, 11 and 12 of Chicken S-NN algorithm, ABCNN and ABCBP shows the convergence performance on 60% training data of crude oil prices. The average accuracy of suggested algorithm Chicken S-NN and other two models ABCNN and ABCBP can be viewed in Figs. 10, 11 and 12 with their MSE. In Fig. 9 show that the

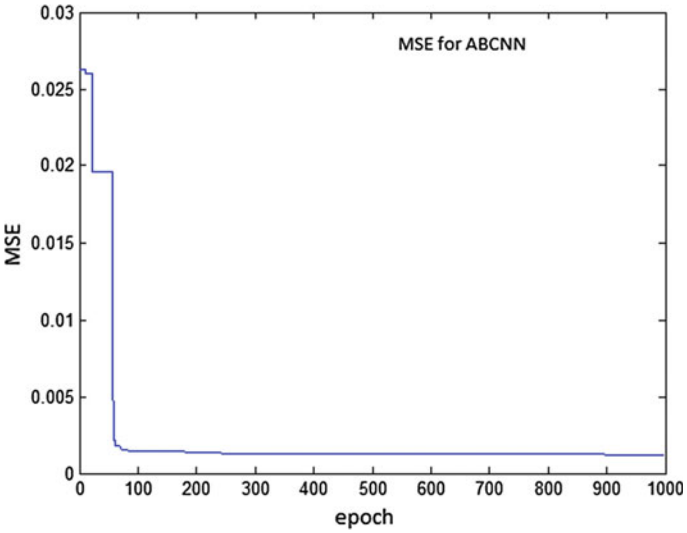


Fig. 3 MSE convergence performance of ABCNN algorithm for 70% training dataset

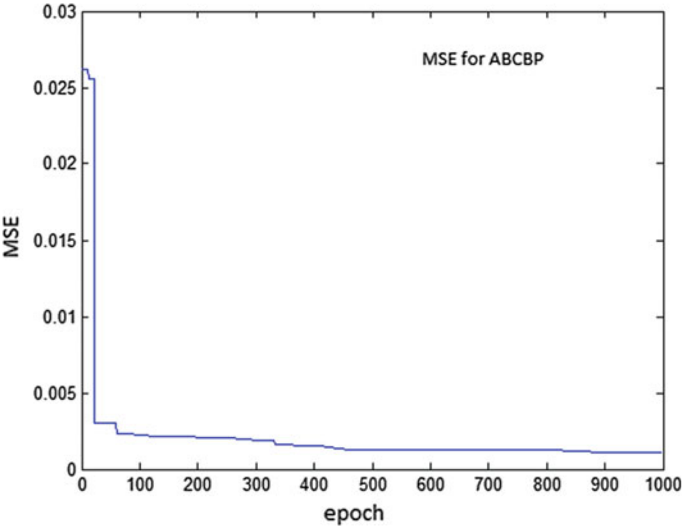


Fig. 4 MSE convergence performance of ABCBP algorithm for 70% training dataset

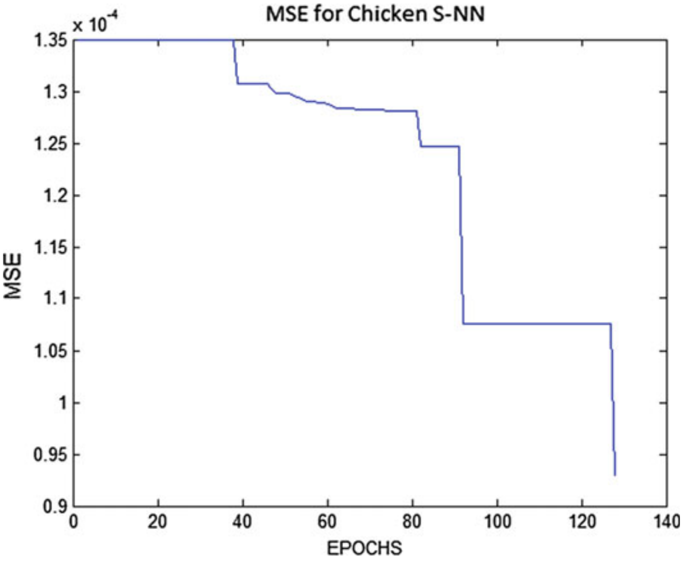


Fig. 5 MSE convergence performance of Chicken S-NN algorithm for 30% testing dataset

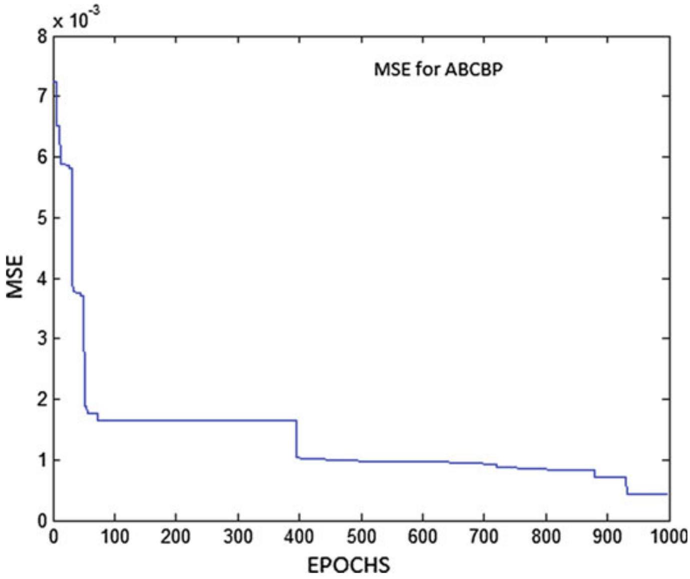


Fig. 6 MSE convergence performance of ABCBP algorithm for 30% testing dataset

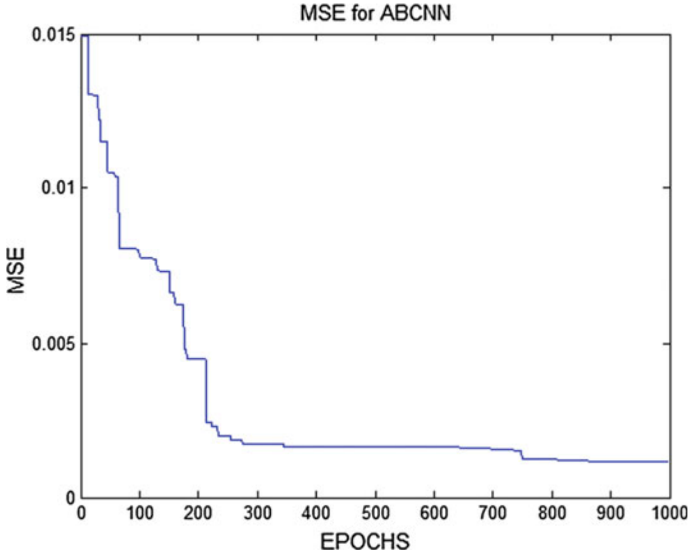
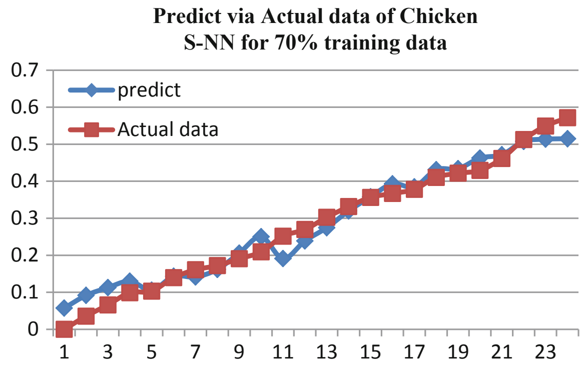


Fig. 7 MSE convergence performance of ABCNN algorithm for 30% testing dataset

Table 2 Epochs, CPU time, accuracy and MSE for 12-5-1 NN-structure

Algorithms	Epochs		CPU time		Accuracy		MSE	
	Training (60%)	Testing (40%)	Training (60%)	Testing (40%)	Training (60%)	Testing (40%)	Training (60%)	Testing (40%)
Chicken S-NN	17	36	47	99	99.99	99.99	9.23E−05	0.0001107
ABCNN	1000	1000	1497	1418	99.50	99.80	0.000366	0.001054
ABCBP	1000	1000	1789	1495	99.60	99.30	0.001332	0.004226

Fig. 8 Actual via predict data convergence of proposed algorithms for 70% training and 30% testing dataset



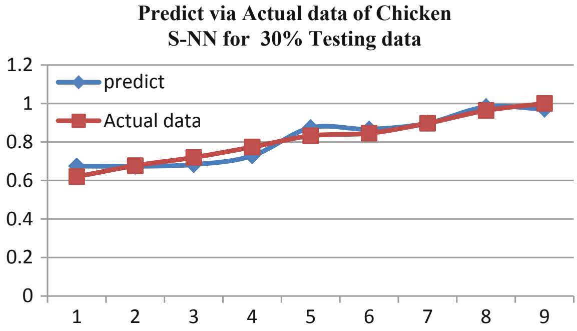


Fig. 9 Actual via predict data convergence of proposed algorithms for 70% training and 30% testing dataset

suggested Chicken S-NN algorithm performs better and converges to global minima in 15 Epochs.

The average accuracy of suggested algorithm Chicken S-NN and other two models ABCNN and ABCBP can be viewed in Figs. 13, 14 and 15 with their MSE. Figures 13, 14 and 15 shows that the suggested Chicken S-NN algorithm performs better and converges to global minima in 36 Epochs.

Figure 16 illustrates comparison of predicted value and actual value. Proposed algorithm Chicken S-NN performs very well and predicts almost near to the actual value.

6 Conclusion

Chicken Swarm is an assumptive based optimization algorithm inspired by chicken capabilities. Chicken-swarm algorithm has capability to search both local and global large space in the candidate solution. This paper proposed Chicken swarm based Neural Network to investigate and to overcome the shortcoming of NN algorithm. The

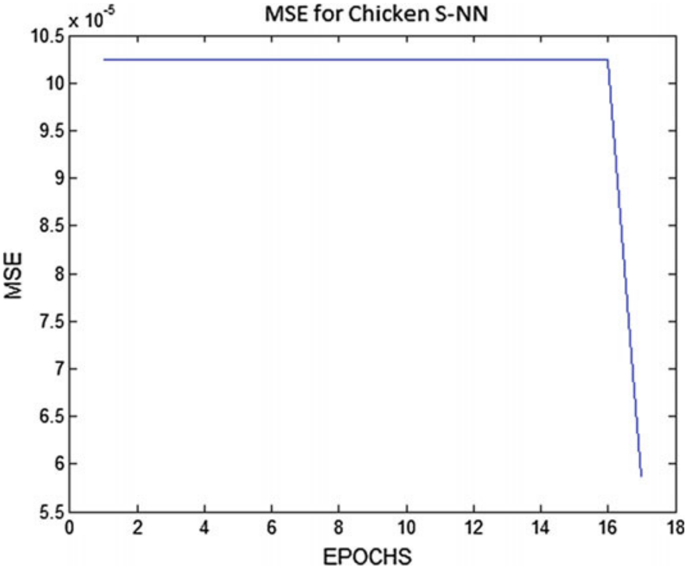


Fig. 10 MSE convergence performance of Chicken S-NN algorithm for 60% training dataset

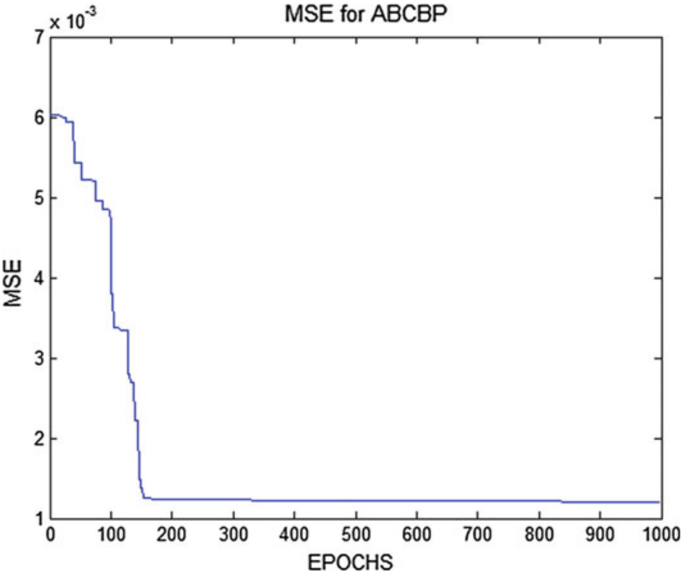


Fig. 11 MSE convergence performance of ABCBP algorithm for 60% training dataset

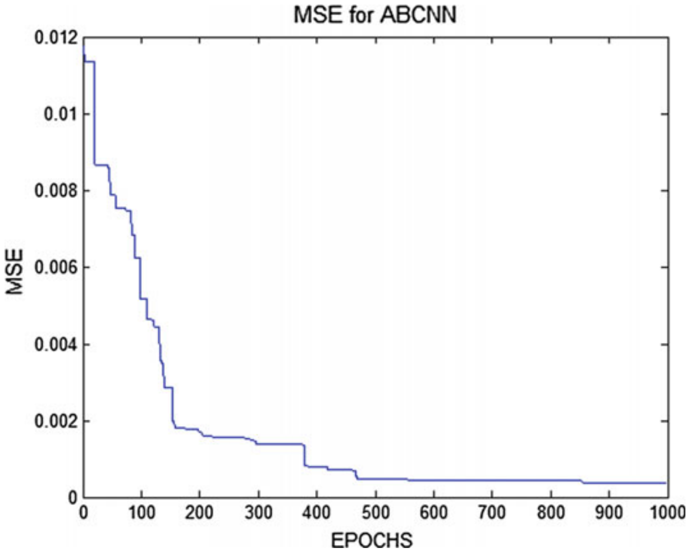


Fig. 12 MSE convergence performance of ABCNN algorithm for 60% training dataset

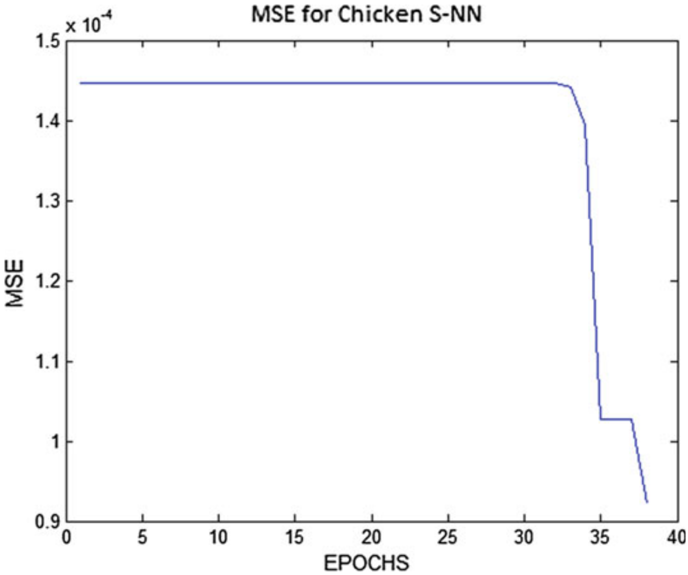


Fig. 13 MSE convergence performance of Chicken S-NN algorithm for 40% testing dataset

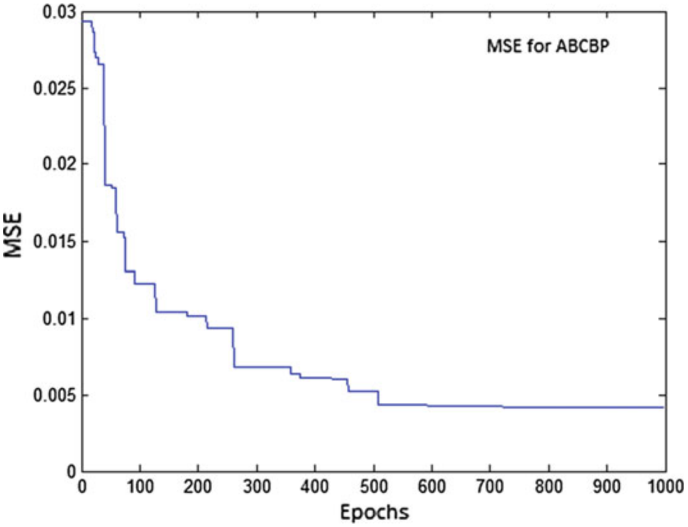


Fig. 14 MSE convergence performance of ABCBP algorithm for 40% testing dataset

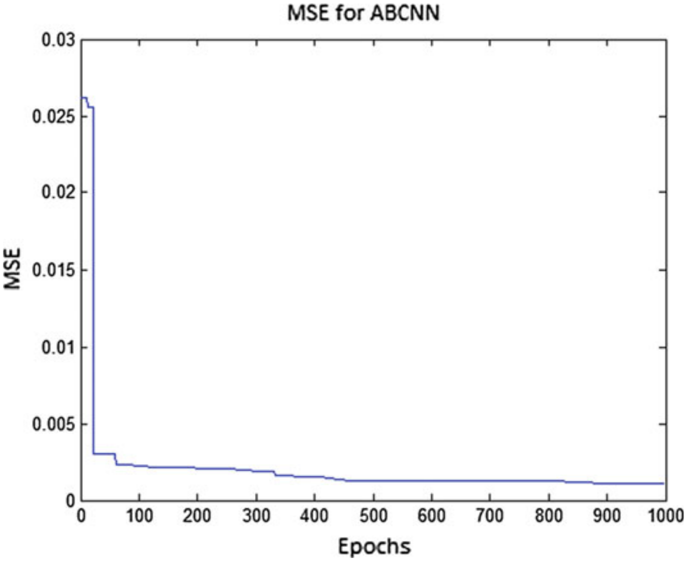


Fig. 15 MSE convergence performance of ABCNN algorithm for 40% testing dataset

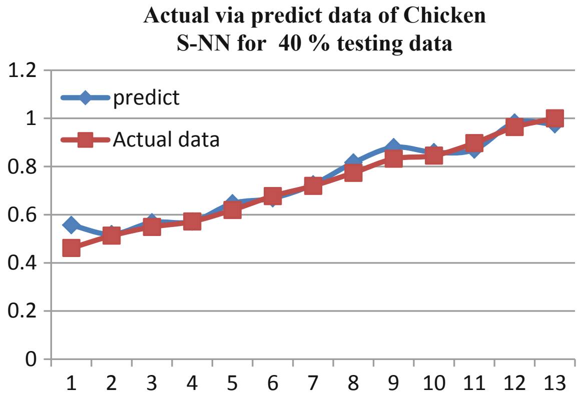


Fig. 16 Actual via predict data convergence of proposed algorithms for 40% testing dataset

performance of the proposed Chicken S-NN model is better to Artificial Bee Colony Back Propagation (ABCBP) and Artificial Bee Colony Neural Network (ABCNN). The network is trained as well tasted on the basis of Crude Oil price dataset. The simulation result concludes that the proposed model outperforms existing methods like ABCBP, and ABCNN in term of MSE accuracy. From Experiments it is also clear that with Chicken swarm neural network avoid the slow convergence and convergence to optimal solution.

Acknowledgements I would like to say special thanks to all the authors for helping in this research.

References

1. Yu L, Wang Z, Tang L (2015) A decomposition–ensemble model with data-characteristic-driven reconstruction for crude oil price forecasting. Appl Energy 156:251–267
2. Barunik J, Malinska B (2015) Forecasting the term structure of crude oil futures prices with neural networks. arXiv preprint arXiv:1504.04819
3. Bildirici M, Ersin ÖÖ (2013) Forecasting oil prices: smooth transition and neural network augmented GARCH family models. J Petrol Sci Eng 109:230–240
4. Shabri A, Samsudin R (2014) Daily crude oil price forecasting using hybridizing wavelet and artificial neural network model. Math Prob Eng 2014:201402 (10 pages). https://doi.org/10.1155/2014/201402
5. Lackes R, Börgermann C, Dirkmorfeld M (2009) Forecasting the price development of crude oil with artificial neural networks. In: Distributed computing, artificial intelligence, bioinformatics, soft computing, and ambient assisted living. Springer, pp 248–255
6. Khazem H, Mazouz A (2013) Forecasting the price of crude oil using artificial neural networks. Int J Bus Mark Decis Sci 6(1)
7. Mirmirani S, Li HC (2004) A comparison of VAR and neural networks with genetic algorithm in forecasting price of oil. Adv Econ 19:203–223
8. Kulkarni S, Haidar I (2009) Forecasting model for crude oil price and commodity futures prices using artificial neural networks. Int J Comput Sci Inf Secur 2(1):81–88
9. Yu L, Wang S, Lai KK (2008) Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. Energy Econ 30(5):2623–2635

10. Hu JWS, Hu YC, Lin RRW (2012) Applying neural networks to prices prediction of crude oil futures. Math Prob Eng 2012:959040 (12 pages). https://doi.org/10.1155/2012/959040
11. Nawi NM, Khan A, Rehman MZA (2014) New optimized Cuckoo Search Recurrent Neural Network (CSRNN) algorithm. In The 8th international conference on robotic, vision, signal processing & power applications. Springer
12. Nawi NM, Khan A, Rehman M (2012) CSLMEN: a new optimized method for training Levenberg Marquardt Elman network based Cuckoo search algorithm
13. Khashei M, Bijari M (2014) Fuzzy artificial neural network p, d, q model for incomplete financial time series forecasting. J Intell Fuzzy Syst Appl Eng Technol 26(2):831–845
14. Nawi NM et al (2015) Weight optimization in recurrent neural networks with hybrid metaheuristic Cuckoo search techniques for data classification. Math Probl Eng 501:868375
15. Nawi NM, Khan A, Rehman MZ (2013) A new cuckoo search based Levenberg-Marquardt (cslm) algorithm. In: Computational science and its applications–ICCSA 2013. Springer, pp 438–451
16. Chiroma H et al (2014) Optimization of neural network through genetic algorithm searches for the prediction of international crude oil price based on energy products prices. In: Proceedings of the 11th ACM conference on Computing Frontiers. ACM
17. Nawi NM, Khan BV, Rehman MZ (2013) A new back-propagation neural network optimized with cuckoo search algorithm. In: Computational science and its applications–ICCSA 2013. Springer, pp 413–426
18. Nawi NM, Rehman MZ, Hafifie N, Khan A, Siming A (2014) Bat-BP: a new bat based back-propagation algorithm for efficient data classification
19. Chen YL, P He, Zhang Y (2011) Combining penalty function with modified chicken swarm optimization for constrained optimization. In: First international conference on information sciences, Machinery, materials and energy. Atlantis Press
20. Sharma V, Rai S, Dev A (2012) A comprehensive study of artificial neural networks. India Int J Adv Res Comput Sci Softw Eng 2(10)
21. Haykin S (2005) Neural network: a comprehensive foundation. Pearson Education, New Delhi
22. Peter GZ, Patuwo B, My H (2001) A simulation study of artificial neural networks for nonlinear timeseries forecasting. Comput Oper Res 28:381–396
23. Hornick KSM, White H (1999) Multilayer feed forward networks are universal approximates. Neural Netw 2:359–366
24. Pan TY, Wamg RY (2004) State space neural networks for short term rainfall-runoff forecasting. J Hydrol 297(1–4):34–50
25. Chiroma H, Abdul-Kareem S, Khan A, Nawi NM, Gital AYU, Shuib L, Herawan T (2015) Global Warming: predicting OPEC carbon dioxide emissions from petroleum consumption using neural network and hybrid cuckoo search algorithm. PLoS ONE 10(8)
26. Meng X et al (2014) A new bio-inspired algorithm: chicken swarm optimization. In: Advances in swarm intelligence. Springer, pp 86–94
27. Hafez AI, Zawbaa HM, Emary E, Mahmoud HA, Hassanien AE (2015) An innovative approach for feature selection based on chicken swarm optimization. In: 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR), IEEE, pp 19–24
28. Khan A et al (2017) Chicken S-BP: an efficient chicken swarm based back-propagation algorithm. In: Herawan T et al (eds) Recent advances on soft computing and data mining: the second international conference on Soft Computing and Data Mining (SCDM-2016), Bandung, Indonesia, August 18–20, 2016 Proceedings, Springer International Publishing, Cham, pp 122–129
29. Energy Information Administration of the US Department of Energy. [Online]. Available http://www.eia.gov/. 2014
30. Chiroma H et al (2016) A new approach for forecasting OPEC petroleum consumption based on neural network train by using flower pollination algorithm. Appl Soft Comput 48:50–58

Forecasting OPEC Electricity Generation Based on Elman Network Trained by Cuckoo Search Algorithm

Abdullah Khan, Rahmat Shah, Nasreen Akhter, Awais Qureshi, Kamran Ullah, Hilal Ahmad and Muhammad Idrees

Abstract All over the world, energy is vitally important because it affects life-standard economy, and social growth. Electricity is one of the main significant forms of energy which is usually needed to be generated and cannot be stored physically. It was the main goal to generate the required electricity to meet a future need in previous studies. To maintain the level of electricity needed constantly, a good system needs to be designed for avoiding waste or shortage. This paper proposes an alternative topology of neural network which is known as Elman network. In the case of Elman networks (ENs), majority techniques only identify topologies in which neurons are connected to each other from hidden to input layer. However, training algorithm of Elman network has a number of disadvantages, like network stagnation and getting stuck in a minimum and low local speed of convergence. This study suggests

A. Khan (✉) · K. Ullah · H. Ahmad
Institute of Business and Management Science, Agricultural University Peshawar, Peshawar, Pakistan
e-mail: Abdullahdirvi@gmail.com

K. Ullah
e-mail: kamranullah@aup.edu.pk

H. Ahmad
e-mail: hilal_a24@yahoo.com

R. Shah
Department of Computer Science, CECOS University Peshawar, Peshawar, Pakistan
e-mail: rahmat@cecos.edu.pk

N. Akhter
Department of Computer Science, FAST National University, Faisalabad Campus, Faisalabad, Pakistan
e-mail: nasreen.akhtar@nu.edu.pk

A. Qureshi
Department of Computer Science, Preston University, Kohat, Pakistan
e-mail: awais.qureshi@preston.edu.pk

M. Idrees
Department of Agriculture Extension Education, Communication, Agricultural University Peshawar, Peshawar, Pakistan
e-mail: dridreesext@yahoo.com

T. Herawan et al. (eds.), *Advances on Computational Intelligence in Energy*,
Green Energy and Technology, https://doi.org/10.1007/978-3-319-69889-2_4

Cuckoo search algorithm (CS) to enhance training time of EN for high precision and fast convergence. Performance of Cuckoo search Elman (CSElman) is compared to artificial bee colony (ABC) and with similar hybrid techniques. Simulation results are displayed that prove the proposed CSElman is better than the other algorithms in this research with respect to accuracy and the speed of convergence.

Keywords Slow convergence · Local minima · Cuckoo search algorithm · Artificial bee colony · Elman network

1 Introduction

The world has become more populated and the electricity demand has increased extremely in the world in recent years. In daily lives of people, electricity-consuming devices and appliances have become more common. For a country, it is very important to be able to supply exactly equal electricity to the demand. If blackouts occur, it will affect negatively electricity-dependent industry; If a country's electricity generation capacity is lower than the demand, or on the other hand, electricity demand is higher than the electricity generation capacity leads to the power plants with idle capacity. Hence, to maintain the electricity demand and supply balance, it is very important to correct plan and develop new electricity generation investments for accurate prediction of the electricity demand for the future [1, 2].

Nowadays, several organizations adopting information technologies are used to help their working, providing enough memory units for storing up-to-date information and used this information for maximum benefit with different methods. Programming to anticipate changes. In future to find a change a way to analyze the statistical knowledge in the past that relates to applying to the current event. The results were used to predict future events. This seems like the future predict event are like that group data or observations that have been recorded according to a continuous period of time, such as yearly, quarterly, monthly, weekly, and daily, or format depending on appropriate to use [3].

To forecast the electricity demand with good accuracy, need to determine accurately the variables that can influence the demand for electricity in the country. One of its main factors is a population that strongly correlated with the demand for electricity (most people consume more electricity). However, the population is insufficient for clarifying variations in demand for electricity in the past years. This is also fair common to study some common considerations and some economic gages in correlation together with the demand for electricity [4].

One of the main scales for the living people in a country is the gross domestic product (GDP); it is the main factor that may be used for such purpose. And, it is also a scale of the capital for living people in a country [5]. GDP per capita increases the standard of living of the people and their lifestyles that dependent on devices of energy generation and utilization. Another two other economic factors employment and inflation that can influence the demand for electricity in addition to per capita GDP [6]. Electricity price is another possible factor that can affect electricity generation. If there are alternatives available for the electricity generation in a country, this is expected that the demand for electricity is price elastic (a price increase leads to a significant reduction in demand); otherwise, the application must be inelastic (a price increase caused a slight drop in demand) [7].

Different forecasting techniques based on data mining were used for the prediction of electricity or future demand of energy, such as artificial neural network (ANN)-based models [8], support vector regression models [9], and methodologies based on fuzzy logic [10].

Artificial neural network is a famous algorithm for function approximation, pattern classification, pattern matching, and associative memories because of its ability to simplify and solve problems as well as its flexibility and competence [11]. ANN has the ability to solve complex problems of huge data sets with high accuracy by classification, prediction, and forecasting nonlinear system. Without having prior knowledge, the neuron store recognized estimates and adopted new patterns on training, which makes it superior and attractive model [12]. Mostly, ANN depends on their design, learning algorithm, and the choice of structures used during training [11]. ANN offers other approaches to solve complex problems. In many different ANN models, the most famous ANN is multi-feed neural network (MLFF) and recurrent neural network (RNN). The FNN sent data only in a single direction from input layer to hidden layers and then to the output layer. The RNN has feedback connection form hidden layer to input layer with short term memory has activation-feedback short-term memory [12], the network has local feedback loops allow to store the previous state of hidden unit. This opportunity makes it progressive as compared to a feed-forward network because the network outputs are calculated from current inputs and hidden layers; so, it has at least one feedback link [13].

RNN has dynamic and nonlinear behaviour. It's is used for different application such as classification electrical power demand, financial data forecasting [14], associative memories, forecasting, management optimization, and simplified sequences of patterns [13]. The complex application like nonlinear modeling and time-series prediction makes RNN topologically more powerful and compact than FFNN. For complex and large problem fully recurrent networks are still has some limitation, so using the partially recurrent networks with the particularly feedback connections from hidden layer to input layer. Without any complication, such model allows the system to memorize past history [12]. Elman neural network is an example of partial RNN. Some features make RNN algorithms often require a huge amount of time to form a network even for the moderate size of the problem and also less efficient. RNN has complex error surface which makes other training algorithms stuck in local minima [14].

To overcome the limitation of Elman network different global search techniques such as genetic algorithm, Ant Bee Colony (ABC), Particle Swarm Optimization (PSO), Cuckoo Search and Bat algorithm used [11], that formulated a type of neural-network that is more superior than conventional Neural-Network models [14]. CS algorithm provides more optimal solution than efficient particle swarm algorithm and genetic algorithm [11].

This paper is organized in such a way that Sect. 2 and Sect. 3 provide the basic idea of Elman network and Cuckoo search algorithm. Proposed technique is discussed in Sect. 4. Section 5 will represent experimental setup with results and discussion. Section 6 presented as conclusion.

2 Elman Network

Elman network was first developed by Elman in 1990 [3, 15–18]. It has hidden input and output layer and, in addition, it has context layer. Context units are equal to the number of neurons in hidden layer that takes values at t iteration from hidden layer being calculated at $t-1$ iteration. Thus, at each iteration, the value of hidden unit being calculated and feed back to the context unit. The value from the hidden unit are feedback to the context unite by set of weights The feedback is improved by set of weights that enables regular adaptation by means of learning. This research uses three-layer network with a single input layer, one hidden or "state" layer, and one "output" layer [12].

In this network structure, each layer has its own index variable: for input nodes, this network uses i *similarly* j and l for hidden and context layer nodes, while k is used for output nodes. In a feed-forward network, the input vector, x is propagated through a weight layer, V;

$$y(t) = \sum_{i}^{n} x_i(t)v_{ji} + b_j \tag{1}$$

where n the number of inputs b_j is a bias.

$$Y_j(t) = f(y(t)) \tag{2}$$

$$Y_j(t-1) = \sum_{i}^{n} x_i(t)v_{ji} + \sum_{l}^{m} Y_j(t)u_{jl} + b_j(t) \tag{3}$$

$$Y_k(t) = g\big(Y_j(t-1)\big) \tag{4}$$

$$\text{out}_k(t) = \text{purline}\left(\sum_{j}^{m} Y_k(t)w_{kj} + b_k\right) \tag{5}$$

where g is an output function and w_{kj} represents weights from hidden to output layer.

3 Cuckoo Search

"Cuckoo search (CS) is meta-heuristic approach developed by Yang and Deb [11, 19, 20] and is applied to solve numerous engineering design optimization problems, such as the design of springs and welded beam structures, and forecasting [13].

The idea of this algorithm was inspired by the obligate brood parasitism of some cuckoo species that use the nest of the host birds and laying their eggs in it. The cuckoos hatch their eggs to increase the likelihood of other eggs in the nest by eliminating some host's eggs [12, 21–24]. Each egg in nest represents solution and cuckoo egg represents new solution; so, the idea of using Cuckoo search is to select optimal solution and discarding the previous one.

There are three main rules that CS algorithm follow

1. At a time, each cuckoo lays only one egg and dumps it in randomly chosen host's nest.
2. The nest with high-quality egg will be the best and selected for future generation.
3. The total numbers of available host's nests are fixed and the probability to discover the cuckoo egg by the host bird is pa ϵ [0, 1]".

Based on these rules, if cuckoo's egg is discovered by host bird, it either abundant that nest and built a new one or either simply throw the egg away.

4 The Proposed CS Elman Algorithm

In the proposed Cuckoo search Elman (CSElman) algorithm, each best nest in CS algorithm represents a possible solution (i.e., the initial weight space and the corresponding biases for Elman network (EN)). The weight optimization problem and the size of the population represent the quality of the solution. In the first epoch, the best weights and biases are initialized with CS and then those weights are passed on to the ERN. The weights in the EN are calculated. In the next cycle, CS updates the weights with the best possible solution and CS will continue searching for the best weights until the last cycle/epoch of the network is reached or either the MSE is achieved. Figure 1 shows the proposed flowchart for the CSElman algorithm.

The CS is a population-based optimization algorithm; it starts with a random initial population. In the proposed CSElman algorithm, the weight value of a matrix is calculated as follows;

$$W_n = U_n = \sum_{n=1}^{N} \alpha \left(\text{rand} - \frac{1}{2} \right) \tag{6}$$

$$B_n = \sum_{n=1}^{N} \alpha \left(\text{rand} - \frac{1}{2} \right) \tag{7}$$

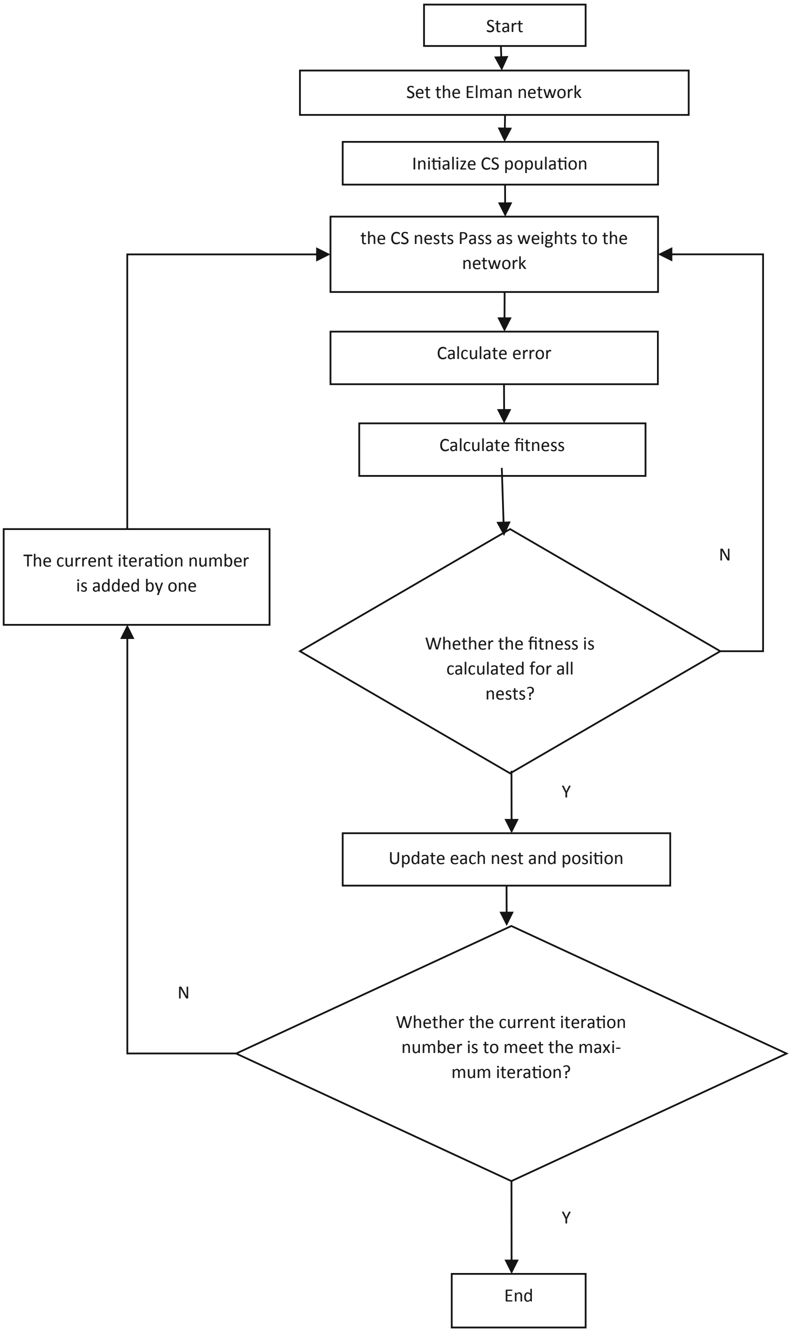

Start
Set the Elman network
Initialize CS population
the CS nests Pass as weights to the network
Calculate error
Calculate fitness
Whether the fitness is calculated for all nests?
N
Y
Update each nest and position
Whether the current iteration number is to meet the maxi-mum iteration?
N
Y
The current iteration number is added by one
End

Fig. 1 Flowchart for proposed CSElman algorithm

where, $W_n = N^{\text{th}}$ is the weight value in a weight matrix. The rand in Eqs. (6 and 7) is the random number between $\left[\,0\ 1\,\right]$, α is any constant parameter for the proposed method it being less than one, and B_n bias value. So, the list of weight matrix is as follows;

$$W^c = \left[W_n^1, W_n^2, W_n^3, \ldots W_n^{N-1},\right] \tag{8}$$

Now from the neural network process, MSE is easily calculated from every weight matrix, in W^c.

$$E = (T_k - Y_k) \tag{9}$$

The performance index for the network is calculated as;

$$V(x) = \frac{1}{2}\sum_{k=1}^{K}(T_k - X_k)^{\mathrm{T}}(T_k - Y_k) \tag{10}$$

$$V_F(x) = \frac{1}{2}\sum_{k=1}^{K} E^T.E \tag{11}$$

In the proposed method, the MSE is considered as the performance index and calculated as;

$$V_\mu(x) = \frac{\sum_{j=1}^{N} V_F(x)}{P_i} \tag{12}$$

where, y_r is the output of the network when the kth input net$_i$ is presented, and $E = (T_k - Y_k)$ is the error for the kth input, $V_\mu(x)$ is the average performance, $V_F(x)$ is the performance index, and P_i is the number of cuckoo population in ith iteration. At the end of each epoch, the list of average sum of square error of ith iteration MSE can be calculated as;

$$\text{MSE}_i = \left\{V_\mu^1(x), V_\mu^2(x), V_\mu^3(x) \ldots V_\mu^n(x)\right\} \tag{13}$$

The Cuckoo search duplicates the MSE and is found when all the inputs are processed for each population of the cuckoo nest. So, the Cuckoo search nest x_j is calculated as;

$$x_j = \text{Min}\left\{V_\mu^1(x), V_\mu^2(x), V_\mu^3(x) \ldots V_\mu^n(x)\right\} \tag{14}$$

And the rest of the average sum of square is considered as other Cuckoo nest.

A new solution x_i^{t+1} for Cuckoo i is generated, using a levy flight according to the following Eq. (15);

$$x_i^{t+1} = x_i^t + \alpha \oplus \mathrm{levy}(\lambda) \tag{15}$$

So, the movement of the other Cuckoo x_i toward x_j can be drawn from Eq. (16);

$$X = \left\{ \begin{matrix} x_i + \mathrm{rand} \cdot \left(x_j - x_i\right)\mathrm{rand}_i > p_\alpha \\ x_i \qquad\qquad\qquad\qquad \mathrm{else} \end{matrix} \right\} \tag{16}$$

The Cuckoo search can move from $p\alpha \in [\,0\ 1\,]$ toward x_j through levy fight and can be written as;

$$\nabla X_i = \left\{ \begin{matrix} x_i + \alpha \oplus \mathrm{levy}(\lambda) \sim 0.01 \cdot \left(\frac{U_j}{|V_j|^{\frac{1}{\mu}}}\right) \cdot (X - X_{\mathrm{best}})\mathrm{rand}_i > p_\alpha \\ x_i \mathrm{else} \end{matrix} \right. \tag{17}$$

where ∇V_i is a small movement of x_i toward x_j. The weights and biases for each layer are then adjusted as;

$$W_n^{s+1} = U_n^{s+1} = W_n^s - \nabla X_i \tag{18}$$

$$B_n^{s+1} = B_n^s - \nabla X_i \tag{19}$$

The pseudocode for the CSElman is given as:

```
1.  Initializes CS population size dimension and Elman structure
2.  Load the training data
3.  While MSE < stopping criteria
4.  Pass the cuckoo nests as weights to network
 5  Feed forward network runs using the weights initialized with CS
 6  Calculate the error using the Eq. (9)
 7  Minimize the error by adjusting network parameter using CS.
 8  Generate Cuckoo egg $(x_j)$ by taking levy flight from random nest.

                              $x_i = x_j$

9.  Abandon a fraction $p\alpha \in [\,0\ 1\,]$ of the worst nest. Build new nest at new location
    via levy flight to replace the old one.
10. Evaluate the fitness of the nest, Chose a random nest i
11. If
        $X_j > X_i$ Then
12. $x_i \leftarrow x_j$
13. $X_i \leftarrow X_j$
14. End if
```

15. *CS keeps on calculating the best possible weight at each epoch until the network is converged.*
16. *End While*

5 Results and Discussion

The workstation used for evaluation of results is equipped with an (Inter Processor Core i5) and Microsoft Windows 7 is used as Operating-System (OS). With the help of MATLAB version 2012, results have been simulated for data. Keeping the number of layers same in hidden as well as in learning layer. A total number of three feed-forward layers have been used to train the model, while the quantity of hidden layers remained fixed to five. Log sigmoid transfer function is used from input to hidden layers, and from hidden to output layers purlin function is used. One thousand epochs have been kept static for the evaluation of each problem. For validation of algorithms, 25 numbers of trails are executed for each model. A different file has been used to record number of failures and mean square errors (MSE) of each independent trial for the classification of electricity data.

The OPEC Electricity Generation Data Set Collection and Pre-Processing

The data set for the OPEC electricity generation in billion kilowatt hours (Bkwh) are collected from Energy Information Administration (EIA), 2015 on a yearly frequency from 1980 to 2012. The data set is freely available and is the current data set at the time of conducting the research. A revisit to the EIA (2015) data repository, on the February 10, 2016, indicated that the data set for 2013, 2014, and 2015 are not yet available. As such, we believed that the data set from 1980 to 2012 are the updated data set. The EIA is one of the credible sources of energy data in which many researchers rely on it to collect their energy research data set [25]. The data set contained OPEC electricity generation in Bkwh for each of the 12 OPEC member countries as listed in the introduction section of the paper as well as totals for the OPEC. In this study, the NN also requires initial parameter settings. The number of inputs and output neurons is set to 12 and 1, respectively, because the number of OPEC countries is 12 and the total OPEC electricity generation in is 1.

DATA: This research will forecast the generation of electricity. Two different experiments are conducted on data. Data is divided using two techniques. By using the first technique, data is divided into two sets, 70% set of data is used for training while the remaining set of 30% data is used for testing the suggested model. In the second technique, 60% data used to train while 40% to test the model. In another technique, suggested model is trained with 80% of electricity generation data and then tested with 20% of data.

5.1 Discussion

This part of research reveals performance of proposed CSElman algorithm in comparison to artificial bee colony with Elman (ABCElman) and genetic algorithm neural network (GANN). Performance is distinguished with respect to CPU time, epochs, accuracy, and mean square error that have been elaborated in Table 1.

All the simulation results are given in Table 1. Table 1 represents the performance of different models in term of epochs, CPU time, accuracy, and MSE used for forecasting the generation of electricity. From simulation result, it shows that the suggested CSElman model is good for the generation of electricity data as compared to ABCElman and GANN. Results display that for the evaluation of 70% training data, suggested CSElman algorithms reach to global minima in six numbers of epochs

Table 1 Convergence performance in term epochs, CPU time, accuracy and MSE

Data set	Algorithms	CSElman	ABCElman	GANN
Training 70%	Epochs	6	1000	1000
	CPU Time	3	539	12
	Accuracy	99.99	99.58	99.54
	MSE	9.36E-07	3.30E-05	9.38E-03
Testing 30%	Epochs	3	1000	1000
	CPU Time	2	529	11
	Accuracy	99.99	99.79	99.53
	MSE	4.68E-08	1.52E-05	0.009047
Training 60%	Epochs	8	1000	1000
	CPU Time	5	534	12
	Accuracy	99.99	99.56	99.44
	MSE	1.64E-07	1.07E-05	1.12E-02
Testing 40%	Epochs	4	1000	1000
	CPU Time	4	511	12
	Accuracy	99.97	99.39	99.68
	MSE	4.86E-06	3.06E-05	0.006354
Training 80%	Epochs	8	1000	1000
	CPU Time	5	534	12
	Accuracy	99.99	99.69	99.57
	MSE	8.00E-08	1.86E-05	0.008502
Testing 20%	Epochs	3	1000	1000
	CPU Time	2	215	12
	Accuracy	99.97	98.79	99.35
	MSE	4.70E-06	9.25E-05	0.01284

with a time spent of 3 s. Obtained accuracy of CSElman is 99.99 having an average mean square error equals to 9.36E-07.

The second model ABCElman attains average accuracy equals to 99.58 with MSE of 3.30E-05. Likewise, GANN gets an average accuracy equals to 99.54 together with average MSE is 9.38E-03. Similarly the performance of the model on 30% testing data, suggested model CSElman performs very well and reaches to global minima in three epochs while attaining 99.99 accuracy with generation of CPU time is 2 s. CSElman achieves an average MSE of 4.68E-08. ABCElman attains 99.79 accuracy while GANN gets 99.53 accuracy. MSE of ABCElman is 3.30E-05 and GANN is 0.009047.

Figure 2 shows the simulation results of the used models in the study. This can be obtained from Fig. 2 that suggested CSElman algorithm performs better in terms of mean square error than rest of the applied models and reaches to global minimum in three epochs. Comparison of the actual and predicted value can be seen in the next figure, i.e., Fig. 3. After analyzing Fig. 3, this can be extracted that suggested CSElman prediction value is nearer to the actual value. Figure 2 displays the performance of convergence of suggested CSElman, ABCElman, and GANN model on 30% testing data of electricity generation.

Similarly, Table 1 further shows the performance analyses of applied algorithms on 60% training data of electricity generation. Results suggest that CSElman algorithms act better than others two algorithms. CSElman achieved global minima in three epochs with CPU time generation equals to 2 s. CSElman has the highest accuracy among the implemented models, an average accuracy of 99.99 with average MSE of 1.64E-07. On the hand, the ABCElman and GANN take 1000 epochs to accuracy is 99.56 having MSE 1.07E-05, and GANN obtains 99.44% accuracy with average MSE is 1.12E-02.

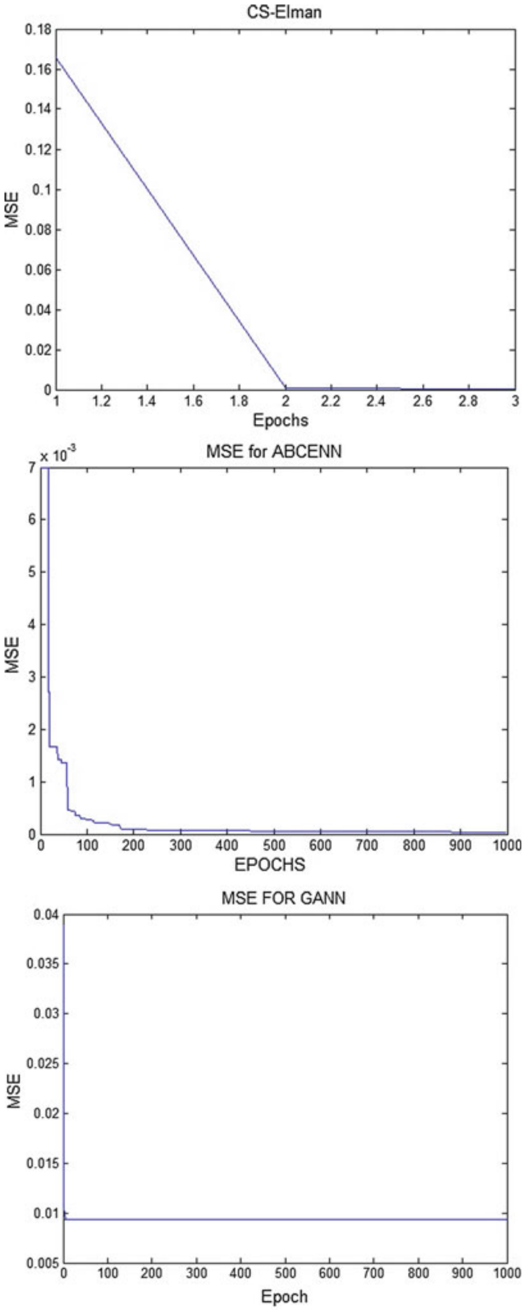
Also to check the performance of the proposed models on 40% tested data, it shows that the performance of suggested CSElman algorithm is better than others implemented models in term of epoch, accuracy, and MSE. CSElman touches to global minima in four epochs taking 4 s of CPU time while attaining 99.9 accuracy. Average MSE of CSElman is 4.86-06. Similarly, ABCElman and GANN obtain accuracy of 99.39 with 3.06 E-05 MSE and GANN has an MSE equals to 0.00635 with 99.68, respectively.

Figure 4 illustrates convergence performance of the CSElman, ABCElman and GANN algorithms on 40% electricity generation tested data set. This can be judged in Fig. 4 that suggested CSElman model performs better in comparison with ABCElman and GANN and converges to global minima in four epochs.

Figure 5 displays the comparison of the actual value of electricity data and predicted value of suggested CSElman model. This is evident that CSElman predicted value is much nearer to actual value. From the predicted value, it shows that the proposed CSElman model predicated well as compared to the other used algorithms.

Furthermore, Table 1 illustrates more details of the proposed model for the electricity generation data on 80% training and 20% testing data. For 80% of training data, results suggest that CSElman algorithm performed better than other two algorithms. The proposed CSElman achieved global minima in eight epochs with CPU

Fig. 2 MSE convergence performance of the used algorithm 30% testing data



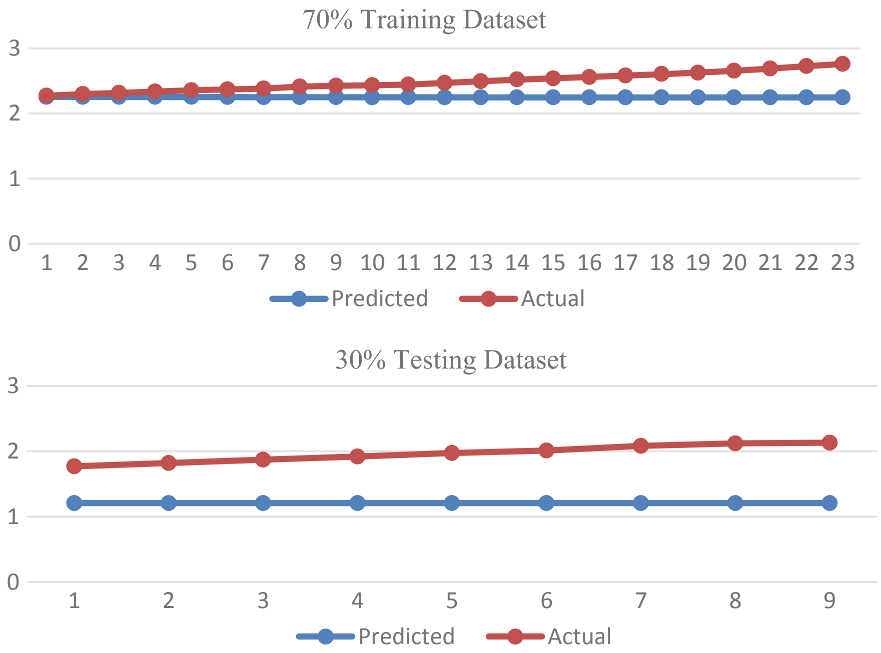


Fig. 3 Comparison of the actual versus predicted value of CSElman on 30% test data

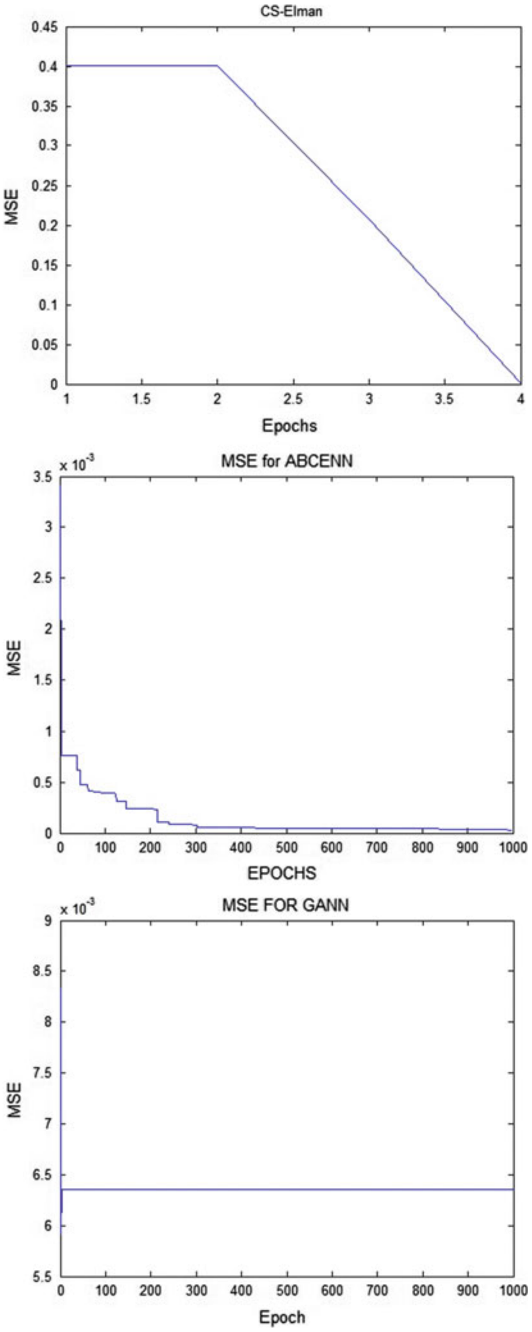
time generation equals to 5 s. CSElman has the highest average accuracy of 99.99 with average MSE of 8.00E-08. On the other hand, ABCElman's accuracy is 99.69 having MSE 1.86E-05, and similarly GANN obtains accuracy equals to 99.57 with average 0.008502 MSE.

Also on 20% tested data, performance of suggested CSElman algorithm is better than other implemented models. CSElman converges to global minima in three epochs taking 2 s of CPU time while attaining 99.97 accuracy. Average MSE of CSElman is 4.70E-06. Similarly, ABCElman and GANN obtain accuracy of 99.79 and 99.35, respectively. 9.25E-05 is the MSE of ABCElman while GANN has an MSE equals to 0.01284.

Figure 6 illustrates convergence performance of 20% tested data of electricity generation. This can be judged from Fig. 6 that suggested CSElman model performs better in comparison with ABCElman and GANN and converges to global minima in three epochs.

Figure 7 shows the comparison of the actual versus predicted value of suggested CSElman model for electricity generation data. This is evident that CSElman predicted value is much nearer to actual value.

Fig. 4 MSE convergence performance of the used algorithm 40% test data



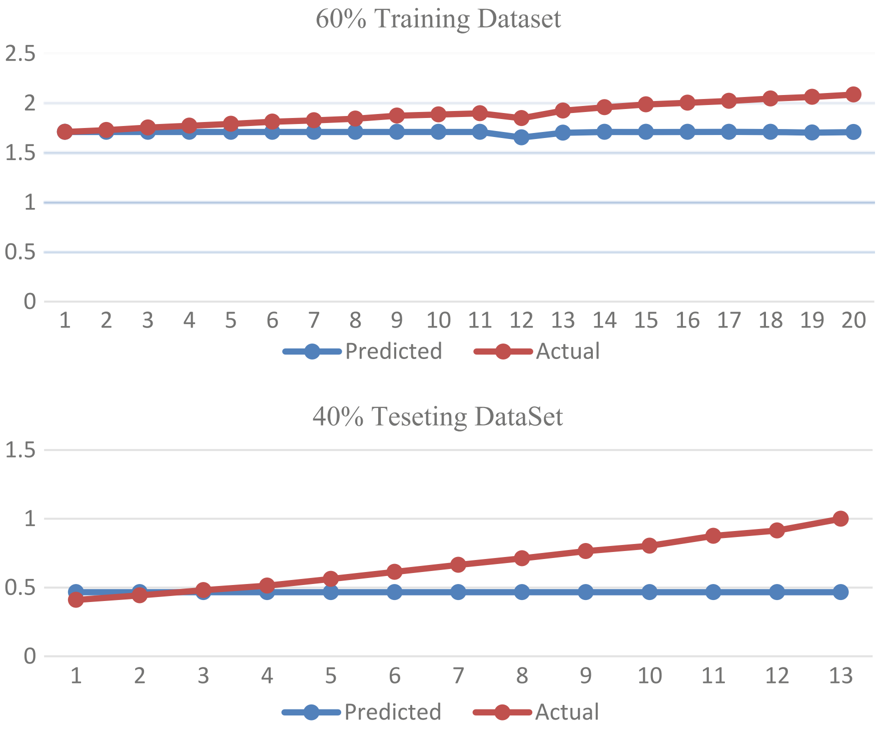
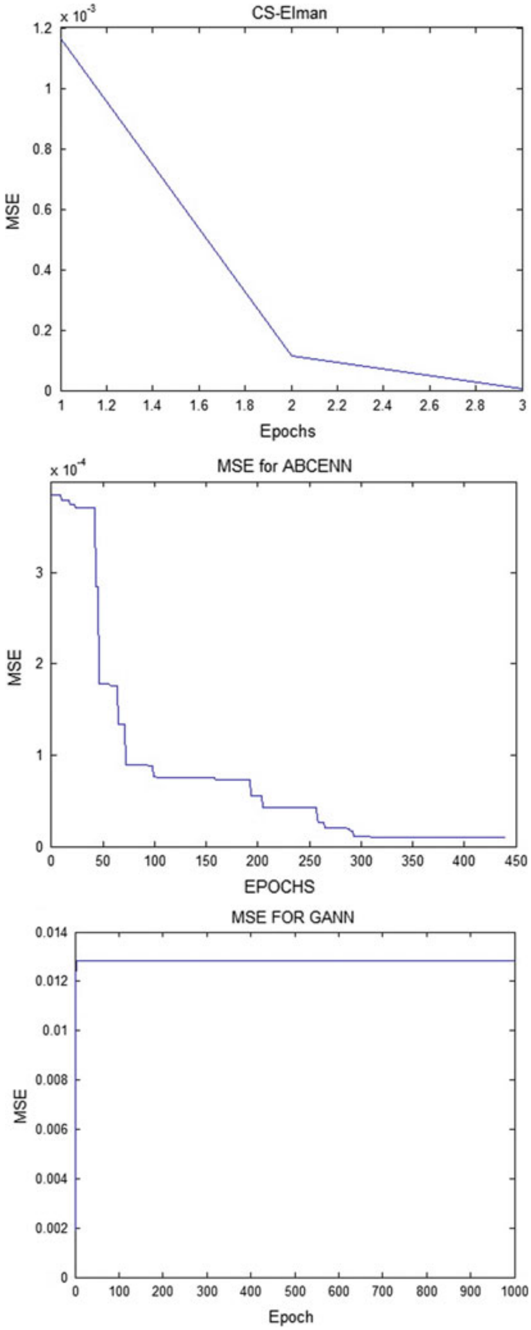


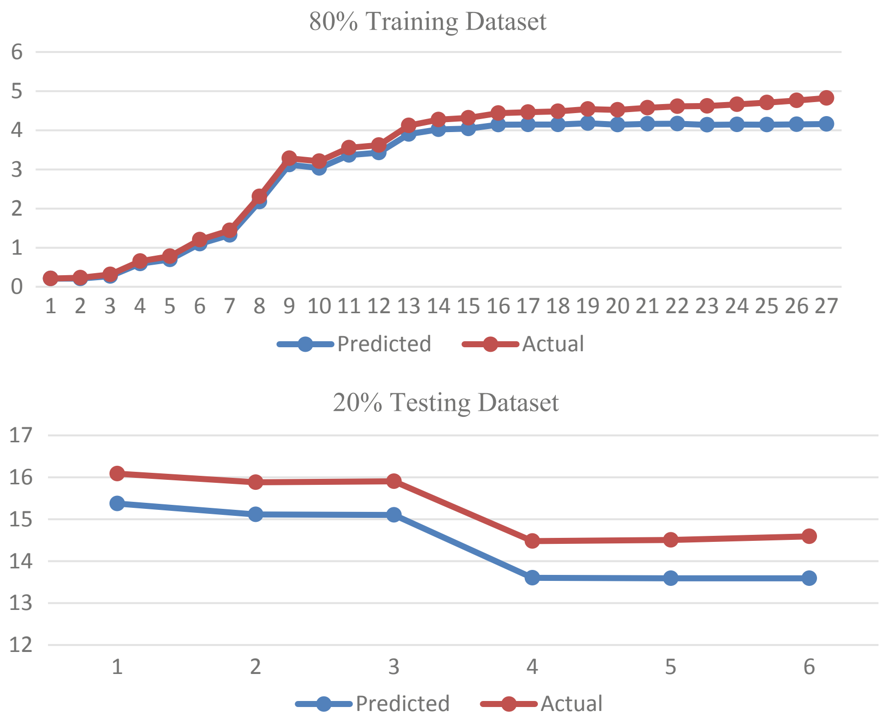
Fig. 5 Comparison of the actual versus predicted value of CSElman on 40% test data

6 Conclusions

To improve the accuracy of forecasting electricity generation, convergence time, robustness, and consistency, the Elman network modeling of OPEC electricity generation based on CS via levy flights has been proposed in this study code named CSElman. For evaluation purpose, the performance of the CS on training Elman network to build a model for the forecasting of OPEC electricity generation is compared with GANN and ABCElman. The CSElman is found to be effective, efficient, robust, and consistent compared to the comparison algorithms. The CSElman forecasting method can be an alternative approach for forecasting the consumption of OPEC electricity, which can be of benefits to the 12 OPEC member countries. This paper proved that the CSElman can effectively, efficiently, robustly, consistently, and reliably model the generation of electricity by OPEC. Performance of proposed Cuckoo search Elman (CSElman) algorithm is compared to artificial bee colony (ABC) as well as with hybrid similar variants. Simulation results illustrate that suggested algorithm CSElman is better as compared to other models with respect to convergence and accuracy rate. We believed that the approach presented in the study can be explored in another domain of applications such as electricity consumption, nuclear energy,

Fig. 6 MSE convergence performance of the used algorithm 20% test data




80% Training Dataset

20% Testing Dataset

Fig. 7 Comparison of the actual versus predicted value of CSElman on 20% test data

renewable energy, weather, and OECD energy consumption. Future work will mainly focus on the application of soft computing techniques in energy big data analytics.

Acknowledgements The researchers would like to say special thanks to “University of Agriculture Peshawar Pakistan for supporting this project.”

References

1. Ozoh P et al (2014) A comparative analysis of techniques for forecasting electricity consumption. Int J Comput Appl 88(15)
2. Günay ME (2016) Forecastingannualgrosselectricitydemandbyartificial neural networks using predicted values of socio-economic indicators and climatic conditions: case of Turkey. Energy Policy 2016(90):92–101
3. Chujai P, Kerdprasop N, Kerdprasop K (2013) Time series analysis of household electric consumption with ARIMA and ARMA models. In Proceedings IMECS conference Hong Kong (2013)
4. Askarzadeh A (2014) Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. Commun Nonlinear Sci Numer Simul 19(4):1213–1228

5. Kucukali S, Baris K (2010) Turkey's short-term gross annual electricity demand forecast by fuzzy logic approach. Energy Policy 38(5):2438–2445
6. Zahedi B, Norum LE (2013) Modeling and simulation of all-electric ships with low-voltage DC hybrid power systems. IEEE Trans Power Electron 28(10):4525–4537
7. Nawaz S, Iqbal N, Anwar S (2014) Modelling electricity demand using the STAR (Smooth Transition Auto-Regressive) model in Pakistan. Energy 78:535–542
8. Kialashaki A, Reisel JR (2014) Development and validation of artificial neural network models of the energy demand in the industrial sector of the United States. Energy 76:749–760
9. Kavaklioglu K (2011) Modeling and prediction of Turkey's electricity consumption using support vector regression. Appl Energy 88(1):368–375
10. Zahedi G et al (2013) Electricity demand estimation using an adaptive neuro-fuzzy network: a case study from the Ontario province–Canada. Energy 49:323–328
11. Nawi NM, Khan A, Rehman MZ (2013) A new cuckoo search based levenberg-marquardt (cslm) algorithm. In Computational science and its applications–ICCSA 2013, Springer, pp 438–451
12. Nawi NM, Khan A, Rehman MZ (2014) A new optimized cuckoo search recurrent neural network (CSRNN) Algorithm. In the 8th international conference on robotic, vision, signal processing and power applications, Springer, 2014
13. Nawi NM, Khan A, Rehman M (2014) CSLMEN: a new optimized method for training Levenberg Marquardt Elman network based cuckoo search algorithm, 2014
14. Nawi NM et al (2015) Weight optimization in recurrent neural networks with hybrid metaheuristic Cuckoo search techniques for data classification. Math Prob Eng 501:868375
15. Aziz M, Hamed HNA, Shamsuddin SMH (2008) Augmentation of elman recurrent network learning with particle swarm optimization. Modeling and simulation, 2008. AICMS 08. In: Second Asia international conference on, 2008: pp 625–630
16. Elman JL (1990) Finding structure in time. Cognitive Sci 14(2):179–211
17. Pasila F, Lesmana T, Ferdinando H Elman neural network application with accelerated LMA training for east java-bali electrical load time series data forecasting
18. Toha SF, Tokhi MO (2008) MLP and Elman recurrent neural network modelling for the TRMS. In: 7th IEEE international conference on cybernetic intelligent systems CIS 2008, pp 1–6
19. Xin-She Y, Deb S (2009) Cuckoo search via levy flights. In: World congress on nature and biologically inspired computing, 2009, pp 210–214
20. Yang X.-S (2010) Nature-inspired metaheuristic algorithms, Luniver Press
21. Tuba M, Subotic M, Stanarevic N (2011) Modified cuckoo search algorithm for unconstrained optimization problems. In: Proceedings of the 5th European conference on European computing conference, 2011, pp 263–268
22. Rajabioun R (2011) Cuckoo optimization algorithm. Appl Soft Comput 11(8):5508–5518
23. Valian E, Mohanna S, Tavakoli S (2011) Improved cuckoo search algorithm for feedforward neural network training. Int J Artif Intell Appl 2(3):36
24. Walton S et al (2011) Modified cuckoo search: a new gradient free optimisation algorithm. Chaos, Solitons Fractals 44(9):710–718
25. Chiroma H et al (2016) A review on artificial intelligence methodologies for the forecasting of crude oil price. Intell Autom Soft Comput, pp 1–14

Variable Neighborhood Search-Based Symbiotic Organisms Search Algorithm for Energy-Efficient Scheduling of Virtual Machine in Cloud Data Center

Mohammed Abdullahi, Shafi'i Muhammad Abdulhamid, Salihu Idi Dishing and Mohammed Joda Usman

Abstract The quest for energy-efficient virtual machine placement algorithms has attracted significant attention of researchers in the cloud computing platform. This paper applied a novel symbiotic organisms search (SOS) algorithm to minimize the number of active server by consolidation VMs on few servers for energy savings. SOS algorithm was inspired by symbiotic relationship exhibit by organisms in an ecosystem to boost their chances of survival. Essentially, SOS mimics mutualism, commensalism, and parasitism forms of relationship for traversing the search space. Hybridized with variable neighborhood search, the hybrid algorithm is termed SOS-VNS. SOS-VNS algorithm is efficient in minimizing energy consumption and improving resource utilization. The SOS-VNS algorithm is applied to various workload instances with varying number of VMs in a simulated IaaS cloud. The results obtained showed that SOS-VNS outperforms the heuristics and achieved reasonable energy savings while improving resource utilization.

Keywords Energy efficiency · Cloud computing · Virtual machine placement · Symbiotic organisms search

M. Abdullahi · S. I. Dishing
Department of Computer Science, Ahmadu Bello University, Zaria, Nigeria
e-mail: abdullahilwafu@abu.edu.ng

S. I. Dishing
e-mail: sidishing@abu.edu.ng

S. M. Abdulhamid (✉)
Department of Cyber Security Science, Federal University of Technology Minna, Minna, Niger, Nigeria
e-mail: shafii.abdulhamid@futminna.edu.ng

M. J. Usman
Department of Mathematics, Bauchi State University Gadau, Gadau PMB 068, Bauchi, Nigeria
e-mail: umjoda@gmail.com

T. Herawan et al. (eds.), *Advances on Computational Intelligence in Energy*,
Green Energy and Technology, https://doi.org/10.1007/978-3-319-69889-2_5

1 Introduction

Cloud computing is a large-scale distributed computing system which presents virtualized computing resources that are dynamically controlled, supervised, sustained, and administered using market principles [1, 2]. It is a subscription-based computing facility, which makes available an appropriate platform for user applications because of its features like application scalability, heterogeneous resources, dynamic resource provisioning, and pay-as-you-go cost model.

Cloud computing technology is rapidly emerging as the de facto paradigm of cybercomputing, storage space and host virtual infrastructures, platforms, and software both in the industries and academia. The vast scalability prospects presented by cloud infrastructures can be simply exploited not just for services and applications hosting but also as an on-demand computing paradigm [3, 4]. Modern cloud data centers consume a huge sum of energy, and the expended energy largely comes from the conventional energy that is produced using fossil fuels. The consequence of this is the high cost of electricity that is being accumulated by the day, and in addition, it leads to high carbon radiations and the large quantity of pollution [5]. With the speedy growth in cloud computing technology, minimizing energy consumption in addition to sustaining high-level computation capacity has become a timely and vital issue in such an environment. Current virtual machines (VMs) scheduling systems have primarily concentrated on improving the resource utilization and minimizing energy usage by improving some of the classical optimization schemes. Nonetheless, many large resource demanding cloud systems are executing VMs in a very realistic situation to have substantial effects on the organization performance and energy consumption. Likewise, sudden peak loads could result in serious scheduling error that can considerably hinder the energy efficiency of scheduling schemes [6].

The approaches that shut down VMs that are not in use can have an undesirable influence in relation to performance of the entire scheme. The idle VMs would not be able to perform using large workloads in a short period. Thus, the optimum energy-aware schemes are required to assure a proper level of minimization of energy consumption. To assess these approaches and to measure the level of impact of performance and energy consumption, a very reliable computational intelligence (CI) technique needed. Furthermore, the selected CI technique has to be able to work with the current conditions of cloud data centers. CI algorithms are inspired by natural process, animals' behavior, or sports to optimize complex real-world problems or systems. CIs as used in cloud scheduling for energy minimization in cloud data centers can be classified into population-based like genetic algorithms [7], ant colony algorithm [8], symbiotic organisms search [9], and particle swarm optimization [10]; and trajectory-based like the simulated annealing [11].

Researchers have employed different metaheuristic algorithms [5, 12–14] (such as PSO, GA, ABC, and ACO) to solve energy-efficient optimization problems in cloud. The results obtained by these algorithms are superior to other heuristic algorithms. However, these metaheuristic algorithms still suffer from the shortcoming of being time-consuming or inefficient in optimizing energy consumption and resource utilization. Therefore, more effective alternative metaheuristic algorithms need to be developed to improve the optimization of energy-efficient parameters. SOS algorithm is a metaheuristic optimization algorithm based on symbiotic association of organisms in an ecosystem, it was first introduced in [15], and being widely applied to optimization problems in various domains such as economic dispatch [16, 17], power optimization [18, 19], construction project scheduling [20], task scheduling [9, 21, 22], design optimization of engineering structures [23, 24], wireless communication [25], and machine learning [26, 27]. SOS is influenced by random oscillation effect in later evolution, which makes it to be easily trapped in local minima, thereby making the convergence rate very slow. The VNS is a robust local search procedure that systematically uses the concept of neighborhood change for search space exploitation and avoidance of likely entrapment in local minima [28, 29]. SOS-VNS combines SOS and VNS to enhance the ability of the proposed algorithm to jump out of likely local maxima, and speed up convergence rate.

In this chapter, we present a variable neighborhood search (VNS)-based symbiotic organisms search (SOS) algorithm for energy-efficient scheduling of VM in cloud data center. The main contributions of this chapter are to:

1. Formulate an energy-efficient VM scheduling optimization technique model for minimizing energy in a cloud data center.
2. Design a variable neighborhood search-based symbiotic organisms search algorithm for energy-aware scheduling in the cloud data center.
3. Evaluate of the proposed technique using standard energy efficiency performance metrics for VM scheduling.

The remaining parts of the chapter are organized as follows. In Sect. 2, literature related to cloud scheduling and energy awareness of VM management in cloud data centers is presented. Section 3 chronicles the formulation and design of energy-efficient VM scheduling optimization using VNS and SOS algorithms. In Sect. 4, the performance metrics were used to comparatively evaluate the proposed technique, and Sect. 5 presents conclusion and future works.

2 Related Works

Computing and VM management in data centers with different features to achieve a particular goal, using computations intelligence algorithms. These include genetic algorithm [30, 31], ant colony optimization [32], particle swamp optimization [33], SOS [9, 21, 22], and BAT [34]. Scheduling of VMs to different virtual resources has a substantial effect on both energy consumption and resource utilization in the cloud data center [35, 36]. Conversely, it is vital to develop an energy-aware VM scheduling technique that retains a balance between energy efficiency and VM utilization in this environment. Ibrahim et al. [12] present an integer linear programming (ILP) algorithm, which reduces the energy used in a cloud computing data centers. In addition, an adaptive genetic algorithm (AGA) was put forward for the reflection of the dynamicity of the data centers and to deliver an optimal scheduling result that reduces the energy usage in the cloud network. Experimental results show that the ILP-AGA method used performs comparatively well in terms of response time and energy minimization. However, the ILP-AGA technique did not consider of task preemption in the process of dynamic scheduling decisions. Similarly, a repairing genetic algorithm (RGA) was developed to address the problem of the large-scale optimization problem in cloud data centers for energy efficiency. It was designed to improve the penalty-based GA scheme by integrating it with Longest Cloudlet Fastest Processor (LCFP) algorithm, through which a preliminary population size was produced with an infeasible solution repairing procedure (ISRP). The VM scheduling with RGA was incorporated into a three-stage energy-aware methodology for cloud data centers. The simulation results show that the proposed framework produces minimization of 23% of energy usage and 43% increase in resource utilization as compared with the steady-state GA in the simulated case study [37]. However, the proposed framework did not consider multi-objectivity and instances in a heterogeneous distributed cloud data centers.

A Euclidean distance founded multi-objective resource scheduling algorithms for VMs and a migration strategy in the cloud data center for energy efficiency was proposed. In addition, the sharing of VMs to physical machines (PMs) was done using the proposed hybrid method of GA and PSO called the HGAPSO [14]. The method, HGAPSO, uses VM scheduling and migration to minimize energy usage and consumption of cloud resources but also it avoids SLA violation in the cloud environment. In order to evaluate the performance of the HGAPSO technique and VM migration policy in relation to energy usage, VM utilization, and SLA violation, an experiment was conducted in both heterogeneous and homogeneous cloud computing systems. The investigation outcomes show the supremacy of HGAPSO and the VM migration strategy over legacy algorithm in terms of energy minimization and best resources utilization strategy. However, the HGAPSO did not consider the effect of computational complexity on the makespan time. Duan et al. [6] present a novel scheduling method termed PreAntPolicy, which comprises of a prediction model using fractal mathematics with scheduling algorithm by improving the ACO algorithm. It uses the proposed model to generate the implementation of the scheduling algorithm through the load tendency estimate. Also, the scheduler is accountable for VM scheduling while reducing the energy usage to guarantee the quality of service (QoS). Experimental results show that using real workload traces gathered from the compute clusters of Google, the PreAntPolicy method demonstrations good energy minimization and resource usage. Furthermore, the method provides an effective dynamic capability scheduling technique for resource-intensive tasks in a distributed environment and may minimize the usage of VMs and energy when scheduling is started by prompt peak loads. However, the computation intelligence algorithm used for optimization is inherently deficient in local entrapment avoidance during scheduling.

Fernandez-Caro et al. [38] developed a tool called SCORE, which was design to simulate energy-aware huge and parallel scheduling techniques and for the implementation of varied and artificial workloads. The experimental outcome shows that the SCORE performance well in terms of energy awareness, security, and scheduling policies in cloud data centers. However, the SCORE simulations need to be compared with real-world data and emulators' results. Similarly, another study by Luo et al. [39] concentrates on the IaaS cloud model, where convention VMs were ran in applicable servers accessible in cloud environment. The research work presented a cloud data center resource scheduling algorithm. The system was designed to deliver QoS (via SLAs) and also reduces energy usage and green computing objectives. Taking into account that the cloud data center host is regularly in thousands of magnitude and that using an exact procedure to address the resource scheduling problem is difficult. The modified shuffled frog leaping algorithm (MSFLA) and enhanced extremal optimization are deployed to resolve the dynamicity of scheduling problem of cloud resources. Investigative outcomes show that the MSFLA system shows good performance in the cloud computing environment. However, the MSFLA scheduling method is relatively weak in energy management in heterogeneous cloud data centers. A computational intelligence algorithm is used for VM placement to reduce the amount of active physical servers running, in order to allocate underutilized servers to minimize energy lost. Encouraged by the performance of the ACO algorithm for undeterministic social problems, an ACO-based method is advanced to accomplish the VM placement objective. In addition to order exchange and migration (OEM) local search procedure, the hybridized system is called an OEMACS. It minimizes the number of active servers used for the scheduling of VMs from exploration viewpoint through an innovative approach for pheromone deposition that guides the artificial ants to encouraging results that cluster contestant VMs in the same place. The proposed hybrid scheme is used in a diverse VM placement with divergent sizes in cloud data centers of different cloud servers. Experimental outcome indicates that the OEMACS largely overtakes some computation intelligence algorithms and other evolutionary algorithms, particularly on VM placement with logjam resource features, and presents substantial savings of energy and more efficient usage of diverse VMs [13]. However, the OEMACS algorithm performs better in a homogeneous cloud data center environment than in a heterogeneous cloud data center environment. Likewise, another similar research presented a multi-objective energy-aware VM scheduling problem for cloud data centers within the concept of green computing. This is partly driven by renewable energy computing strategy, where the computing VMs of the cloud data centers are based on DVFS. Therefore, an improved multi-objective computational intelligence algorithm called OL-PICEA-g was put forward to address the issue. The PICEA-g system with the general opposition-based learning was used for searching the appropriate computing VM in the cloud data center. Experimental results of the OL-PICEA-g system were evaluated with the PICEA-g method. The outcome shows the dominance and efficiency of the OL-PICEA-g algorithm over PICEA-g scheme [5]. However, the OL-PICEA-g algorithm cannot handle the energy-aware scheduling of tasks and VMs on a data center on batteries. Therefore, our proposed variable neighborhood search-based symbiotic organisms search algorithm for energy-aware scheduling of VMs in the cloud data center environment is put forward to address this problem.

3 Energy-Efficient Virtual Machine Scheduling Optimization

In cloud computing, users' applications requirements are stated in the form of VMs features (CPU, memory, and storage) and operating system for executing user application. Then, the VMs that meet the user application requirements are assigned to a server based on a placement strategy. The choice of a suitable server for VMs assignment while minimizing energy consumption is a challenging problem. This paper studies the VM placement problem for optimizing the number of running servers where VMs are servers are balanced and consolidated based on CPU, memory, and storage constraints.

3.1 Problem Definition

Suppose there are m VMs and n servers, with $V = \{V_1, V_2, V_3, \ldots, V_m\}$ and $H = \{H_1, H_2, H_3, \ldots, H_n\}$ as sets of VMs and servers, respectively. Also, let cV_j, mV_j, and sV_j be the CPU, memory, and storage requirements of a VM $V_j \in V$, respectively. In a similar manner, let cH_j, mH_j, and sH_j be the CPU, memory, and storage capacities of server $H_j \in H$, respectively. The objective is to obtain a VM assignment schedule that will reduce the energy consumption and improve the resource utilization. It is assumed that a server has the capacity to meet the resource needs of a VM. That is, in a VM assignment schedule, a VM can only be mapped to one and only one server which is a kind of zero-one adjacency matrix P, where its element p_{ij} indicates the VM V_j is mapped to a server H_i. If V_j is mapped to H_i then $p_{ij} = 1$, otherwise $p_{ij} = 0$. Each server H_i must be able to meet the resource requirements of all the VMs assigned to it.

The VM placement problem for reducing the number of active servers can be formally written as:

$$\text{minimize}\, f(p) = \sum_{i=1}^{n} q_i \tag{1}$$

$$\text{subject to}\; p_{ij} \begin{cases} 1; & \text{if } V_j \text{ is assigned to } H_j \\ 0; & \text{otherwise} \end{cases} \tag{2}$$

$$p_{ij} \begin{cases} 1; & \text{if } \sum\limits_{j=1}^{m} p_{ij} \geq V_j \text{ for all } V_j \in V \text{ is assigned to } H_j \\ 0; & \text{otherwise} \end{cases} \tag{3}$$

$$\sum_{i=1}^{n} p_{ij} = 1;\; \text{for } V_j \in V \tag{4}$$

$$\sum_{j=1}^{m} cV_j p_{ij} \leq cH_i q_i \tag{5}$$

$$\sum_{j=1}^{m} mV_j p_{ij} \leq mH_i q_i \tag{6}$$

$$\sum_{j=1}^{m} sV_j p_{ij} \leq sH_i q_i \tag{7}$$

The constraint 3 indicates that a server H_i is selected when $q_i = 1$ while constraint 4 ensures that a VM is assigned to one and only one server. Constraints 5, 6, and 7 make sure that a server assigned to a VM must be able to satisfy its requirements. The power consumption model of a server is linearly dependent on its CPU utilization [40]. An active server in an idle state consumes about 50–70% of its power consumption when active at full load [41]. The power model is defined in Eq. 8.

$$P(\alpha) = \beta P_{\text{max}} + \alpha(1 - \beta)P_{\text{max}} \tag{8}$$

where P_{max} is the power consumption at full load, $\alpha \in [0,1]$ is the CPU utilization, β is the fraction of power consumed in an idle state. Since a reasonable amount of energy is wasted when a server is in an idle state, minimization of the number of active servers will amount to significant energy savings. During the experiment, the power consumption of a serve will be estimated using Eq. 8.

3.2 Basic Concepts of Symbiotic Organisms Search

SOS is a new and promising metaheuristic algorithm inspired by forms of interaction adopted by organisms for their survival in the ecosystem. The algorithm simulates mutualism, commensalism, and parasitism forms of interaction to evolve candidate solutions. Mutualism interaction involves two organisms which cohabit together for the benefit of each other, none of the organisms loose in the interaction. In commensalism interaction, one of the pair of the organisms involved in the interaction benefits while another organism neither loses nor gained from the relationship. In the parasitism relationship, one organism benefits from the relationship while the other is harmed.

In SOS algorithm, potential solutions are represented by a population of organisms which evolved through successive iterations. An initial ecosystem (population) of organisms (candidate solutions) with *ecosize* (number of organisms in the ecosystem) is generated as $E = \{X_1,X_2,X_3,\ldots,X_{\text{ecosize}}\}$. The position of organism i is denoted as $X_i = [x_{i1},x_{i2},x_{i3},\ldots,x_{id}]^T$ for optimizing a d–dimensional problem. The positions of each organism in the ecosystem are updated using mutualism, commensalism, and parasitism phases, respectively. The following sections describe the mutualism, commensalism, and parasitism phases of SOS algorithms.

Mutualism Phase In this phase, an organism X_i randomly selects an organism X_j $(i \neq j)$ for mutual interaction to improve the survival of both X_i and X_j. The new organisms are obtained according to Eqs. 9 and 10. MV is the mutual relationship vector between X_i and X_j as defined in Eq. 11. X_{best} represents the organism with best fitness value. B_1 and B_2 represent the benefit factors between organism X_i and X_j. B_1 and B_2 stochastically determined are either 1 or 2. The values 1 and 2 denote light and heavy benefits, respectively. The fitness value of the new organisms $f\left(x_i^{\text{new}}\right)$ and $f\left(x_j^{\text{new}}\right)$ are evaluated, then X_i and X_j are updated to x_i^{new} and x_j^{new}, respectively, if the fitness of the new organisms are better as represented in Eqs. 12 and 13, respectively.

$$X_{i,\text{new}} = X_i + R_1(0, 1 - \beta) * (X_{\text{best}} + \text{MV} * \beta_1) \tag{9}$$

$$X_{j,\text{new}} = X_j + R_2(0, 1 - \beta) * (X_{\text{best}} + \text{MV} * \beta_2) \tag{10}$$

$$\text{MV} = \frac{1}{2}(x_i + x_j) \tag{11}$$

$$X = \begin{cases} X_i^{\text{new}} & \text{if } f\left(X_i^{\text{new}}\right) > f(X_i) \\ X_i & \text{if } f\left(X_i^{\text{new}}\right) \leq f(X_i) \end{cases} \tag{12}$$

$$X = \begin{cases} X_j^{\text{new}} & \text{if } f\left(X_j^{\text{new}}\right) > f(X_j) \\ X_j & \text{if } f\left(X_j^{\text{new}}\right) \leq f(X_j) \end{cases} \tag{13}$$

where R_1(0,1) and R_2(0,1) are vectors of random numbers in the range 0–1; $f(.)$ is the fitness function.

Commensalism Phase In commensalism phase, an organism X_i interacts with a random organism x_j $(j \neq i)$ for the improvement of survival of X_i. The new organism is obtained using Eq. 14; X_{best} is the fittest organism. X_i is updated to X_i^{new}, if $f\left(X_i^{\text{new}}\right)$ is better than that of $f(X_i)$ according to the relation in Eq. 15.

$$X_i^{\text{new}} = X_i + R(-1, 1) * (X_{\text{best}} + X_j) \tag{14}$$

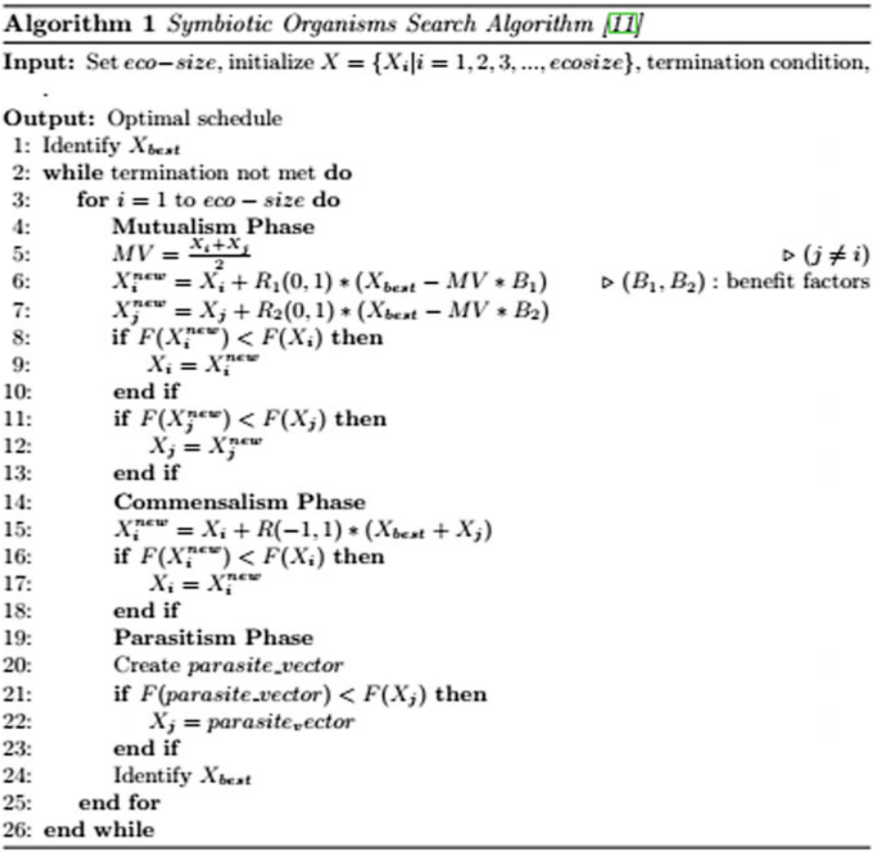
$$X_i = \begin{cases} X_i^{\text{new}} & \text{if } f\left(X_i^{\text{new}}\right) > f(X_i) \\ X_i & \text{if } f\left(X_i^{\text{new}}\right) \leq f(X_i) \end{cases} \tag{15}$$

where $R(-1,1)$ is a vector of random numbers between -1 and 1.

Parasitism Phase In parasitism phase, an organism X_i is utilized to create an artificial parasite called parasite vector by mutating X_i using uniformly generated random number. The parasite vector is evaluated against a randomly selected organism X_j, and the parasite vector replaces X_j if the parasite vector is fitter. The relationship in Eq. 16

$$X_j = \begin{cases} \text{PV if } f(\text{PV}) > f(X_j) \\ X_j \text{ if } f(\text{PV}) \le f(X_j) \end{cases} \tag{16}$$

where PV is the parasite vector.

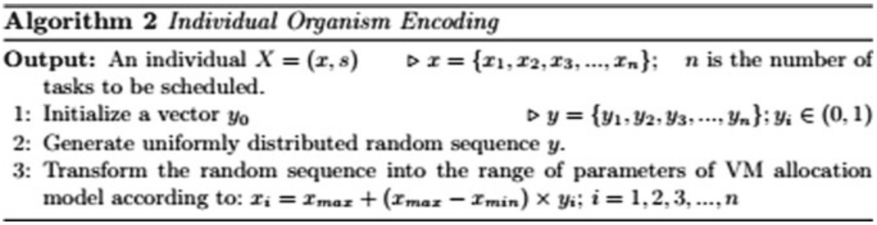


Algorithm 1 *Symbiotic Organisms Search Algorithm* [11]

```
Input: Set $eco-size$, initialize $X = \{X_i | i = 1, 2, 3, ..., ecosize\}$, termination condition,
Output: Optimal schedule
Identify $X_{best}$
while termination not met do
    for $i = 1$ to $eco - size$ do
        Mutualism Phase
        $MV = \frac{X_i + X_j}{2}$                                    ▷ $(j \neq i)$
        $X_i^{new} = X_i + R_1(0, 1) * (X_{best} - MV * B_1)$        ▷ $(B_1, B_2)$ : benefit factors
        $X_j^{new} = X_j + R_2(0, 1) * (X_{best} - MV * B_2)$
        if $F(X_i^{new}) < F(X_i)$ then
            $X_i = X_i^{new}$
        end if
        if $F(X_j^{new}) < F(X_j)$ then
            $X_j = X_j^{new}$
        end if
        Commensalism Phase
        $X_i^{new} = X_i + R(-1, 1) * (X_{best} + X_j)$
        if $F(X_i^{new}) < F(X_i)$ then
            $X_i = X_i^{new}$
        end if
        Parasitism Phase
        Create $parasite\_vector$
        if $F(parasite\_vector) < F(X_j)$ then
            $X_j = parasite_vector$
        end if
        Identify $X_{best}$
    end for
end while
```

Solution Encoding The meaning and dimension of an organism determine the encoding for the problem at hand. In the proposed algorithm, each organism is an individual in the ecosystem that represents a part of the solution search space. To define the solution representation for the problem, each organism represents a complete VM allocation; thus, the dimension of an organism is the same as the number of VMs. The real values are used to represent the servers to be selected. The coordinate system for determining the position of an organism in the solution search space is dependent on the dimension of the organism. The organism is a five-dimensional one and its position on the search space is defined by coordinates 1 through 5. For instance, an organism which encodes an allocation with five VMs and three servers can have a solution $S = \{(1), (1,5), (2,3), (3,4), (2,3)\}$; the first element of the ordered pair represents the index of server and the second element of the ordered pair represents the index of the VM. For instance, the ordered (1) indicates that VM 1 is allocated to server 1, the ordered pair (2,3) indicates that VM 2 is allocated to server 3.

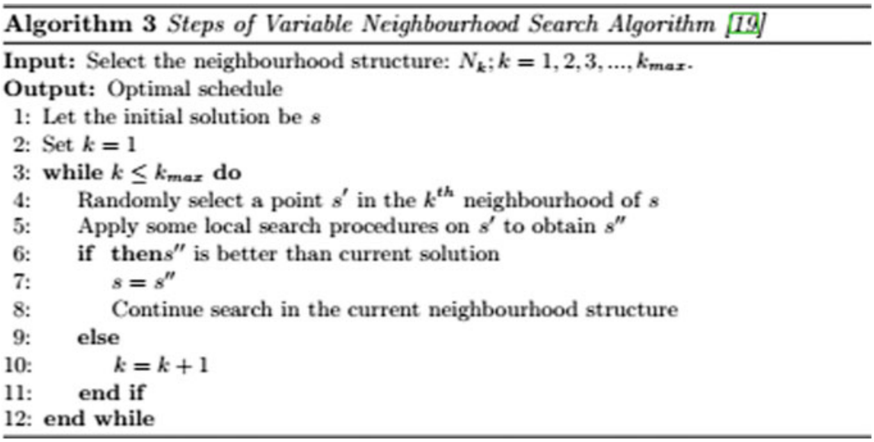
Initialization Uniformly generated random numbers are used to initialize the ecosystem (population) which also serves as the source of the randomness for updating the positions of the organisms during the searching procedure.



```
Algorithm 2 Individual Organism Encoding
Output: An individual $X = (x, s)$    ▷ $x = \{x_1, x_2, x_3, ..., x_n\}$; $n$ is the number of tasks to be scheduled.
1: Initialize a vector $y_0$    ▷ $y = \{y_1, y_2, y_3, ..., y_n\}; y_i \in (0, 1)$
2: Generate uniformly distributed random sequence $y$.
3: Transform the random sequence into the range of parameters of VM allocation model according to: $x_i = x_{max} + (x_{max} - x_{min}) \times y_i$; $i = 1, 2, 3, ..., n$
```

SOS Operators for Position Update The candidate solutions are represented by ecosystem (population) of organisms, while mutualism, commensalism, and parasitism operators to direct the search process by candidate solutions. Each organism is represented by a coordinate system in the search space, and organisms keep an update of global best position X_{best} which is determined based on the fitness function of the problem at hand. The fitter organisms are allowed to proceed to the next generation of potential solution, while the unfitted organisms are discarded. The fitter organisms are those with good solution, while the unfitted organisms hold bad solution. The positions of the organisms are then updated toward the X_{best} locations using mutualism, commensalism, and parasitism phases, respectively. The rate of movement of organisms toward the X_{best} locations is moderated by chaotic random sequence to improve global search ability of the organisms. The SOS operators are continuously applied to the population of organisms which represents candidate solutions until the stopping criterion are reached.

Local Search Using Variable Neighborhood Search Variable neighborhood search (VNS) is a robust local search procedure that systematically uses the concept of neighborhood change for search space exploitation and avoidance of likely entrapment in local minima [28, 29]. VNS incrementally explores the distant neighborhoods of the present solution and move to the next neighborhood if there is an improvement in the current solution which makes VNS more robust than is in contrast to other local search approaches that traverse the trajectory of the solutions [42]. With this, the current solution is used to identify the potential neighborhood solutions. VNS has been successfully hybridized with metaheuristic algorithms in improving their search efficiency. Considering the fact that VNS has a strong local search ability, a novel hybrid algorithm (SOS-VNS) is presented in order to further improve the local search ability of SOS algorithm. The proposed SOS-VNS algorithm utilizes the exploration ability of SOS and exploitation ability of VNS to improve the efficiency of cloud data centers. The framework of VNS algorithm is given as Algorithm 3.



Algorithm 3 *Steps of Variable Neighbourhood Search Algorithm* [19]

```
Input: Select the neighbourhood structure: $N_k; k = 1, 2, 3, ..., k_{max}$.
Output: Optimal schedule
Let the initial solution be $s$
Set $k = 1$
while $k \leq k_{max}$ do
    Randomly select a point $s'$ in the $k^{th}$ neighbourhood of $s$
    Apply some local search procedures on $s'$ to obtain $s''$
    if then$s''$ is better than current solution
        $s = s''$
        Continue search in the current neighbourhood structure
    else
        $k = k + 1$
    end if
end while
```

The neighborhood structures must ensure that the imposed optimization constraints are not violated. This study employs four different kinds of neighborhood structures.

Neighborhood Structure N_1: The neighborhood structure N_1 intends to change the server that a VM belongs to, seeking to improve resource utilization, thereby improving energy efficiency. Suppose that s is a candidate solution and i is the index of the randomly selected VM. Let $s[i] = h_i$ be the server that host VM i. The value of $s[i] = h_i$ is changed as follows: First, a check is done if VM with index i can be moved to a new server without violating the imposed constraints, in this case, the processing, memory, and storage constraints. The choice of the server to which VM i will be moved to is determined as follows: for each VM$_j$ $(i \neq j)$, if $s[i] \neq s[j]$ then $[j]$ is a candidate server to accept the VM$_i$. If the candidate servers are more than one, then VM i is moved to the server with lesser utilization. As a result, a solution s^0 is better than a solution x if and only if it has a better resource utilization with respect to s.

Neighborhood Structure N_2: While N_1 changes the server that a VM belongs to, seeking to improve resource utilization, neighborhood structure N_2 migrates a VM i to another server to improve the degree of load balance. For a candidate solution s, the value of $s[i] = c_i$ is changed by the neighborhood as follows: the first step is to ensure that the server constraints (CPU, memory, and storage capacity) are not violated which is similarly done like for the N_1 neighborhood. If the constraints are not violated, then N_2 is applied. The server to which VM i will be migrated to is determined as follows: For each VM VM$_j$ $(i \neq j)$, if $s[i] \neq s[j]$, then $[j]$ is a candidate server to accept the VM i. If the candidate servers are more than one, then

VM i is moved to the server with lesser load. As a result, a solution s^0 is better than a solution x if and only if its degree of load balance is lower than that of s.

Neighborhood Structure N_3: Like N_1 neighborhood, neighborhood structure N_3 is to achieve a better resource utilization. This is achieved by exchanging the VM i and VM j belonging to different servers h_i and h_j. In particular, to exchange VMs between servers h_i and h_j, the following conditions must be satisfied.

- The removal of VM i from server h_i does not violate the imposed constraints.
- The removal of VM j from server h_j does violate the imposed constraints.
- If VM i shares the same server with at least one VM k $(k \neq j)$, in server h_j
- If VM j shares the same server with at least one VM k $(k \neq i)$ in server h_i.

If the above conditions are satisfied, then VM i is migrated to server h_j and VM j is migrated to server h_i. In this case, a solution s^0 is better that s if and only if it has a better resource utilization with regard to s.

Neighborhood Structure N_4: Like N_3 neighborhood, neighborhood structure N_4 exchanges VM i and VM j belonging to different servers h_i and h_j. Different from N_3. In contrast to N_3, the essence of N_4 is to achieve more compact servers. As for N_3, to exchange VMs between servers h_i and h_j, the following conditions must be satisfied.

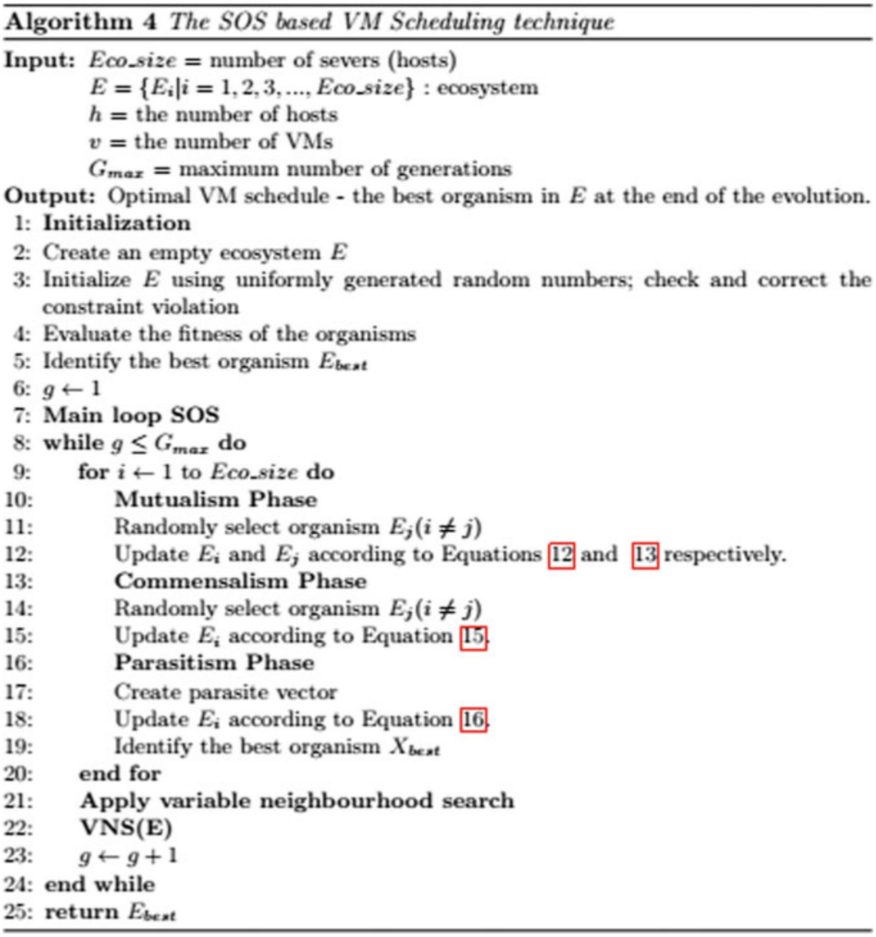
- The removal of VM i from server h_i does not violate the imposed constraints.
- The removal of VM j from server h_j does violate the imposed constraints.
- If VM i shares the same server with at least one VM k $(k \neq j)$ in server $h_{j.}$
- If VM j shares the same server with at least one VM k $(k \neq i)$ in server h_i.

If the above conditions are satisfied, then VM i is migrated to server h_j and VM j is migrated to server h_i. In this case, a solution s^0 is better that s if and only if it has a better degree of load balance with regard to s.

The Proposed VM Scheduling Algorithm In this section, the VM scheduling algorithm is presented. The various components of the proposed algorithms have been given in the previous sections. Hence, the problem initialization, application of SOS algorithm to optimize the objective function, and the use of variable neighbourhood search. The different constituents of the technique address different problems as they affect existing VM scheduling techniques. More specifically:

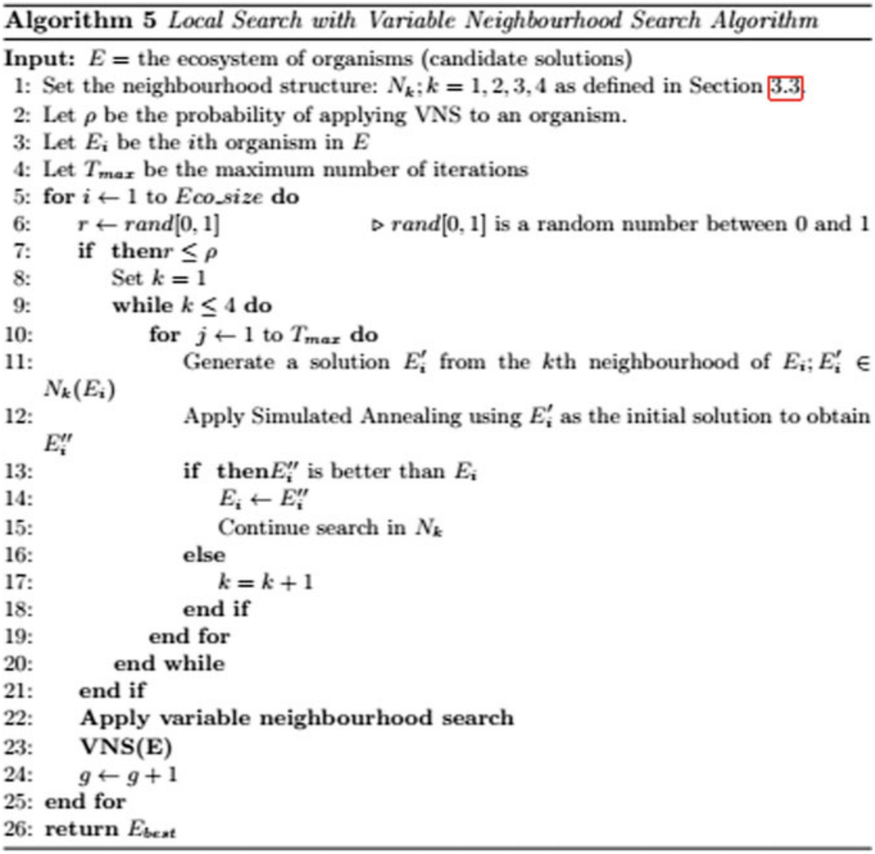
- The initialization method produces initial solutions to speed up the convergence rate of the SOS algorithm.
- The SOS algorithm optimizes the objective function in a more efficient manner.
- The use of VNS further improves the quality of the obtained solution, hybridizing the ability of SOS in efficiently exploring the solution search space with the capability of VNS (local search) in exploiting a specific region of the solution search space.

The resulting VM scheduling technique is given as Algorithm 4, while the VNS algorithm is given as Algorithm 5.



Algorithm 4 *The SOS based VM Scheduling technique*

```
Input: $Eco\_size$ = number of severs (hosts)
       $E = \{E_i | i = 1, 2, 3, ..., Eco\_size\}$ : ecosystem
       $h$ = the number of hosts
       $v$ = the number of VMs
       $G_{max}$ = maximum number of generations
Output: Optimal VM schedule - the best organism in $E$ at the end of the evolution.
Initialization
Create an empty ecosystem $E$
Initialize $E$ using uniformly generated random numbers; check and correct the constraint violation
Evaluate the fitness of the organisms
Identify the best organism $E_{best}$
$g \leftarrow 1$
Main loop SOS
while $g \leq G_{max}$ do
    for $i \leftarrow 1$ to $Eco\_size$ do
        Mutualism Phase
        Randomly select organism $E_j (i \neq j)$
        Update $E_i$ and $E_j$ according to Equations 12 and 13 respectively.
        Commensalism Phase
        Randomly select organism $E_j (i \neq j)$
        Update $E_i$ according to Equation 15.
        Parasitism Phase
        Create parasite vector
        Update $E_i$ according to Equation 16.
        Identify the best organism $X_{best}$
    end for
    Apply variable neighbourhood search
    VNS(E)
    $g \leftarrow g + 1$
end while
return $E_{best}$
```



Algorithm 5 *Local Search with Variable Neighbourhood Search Algorithm*

```
Input: $E$ = the ecosystem of organisms (candidate solutions)
Set the neighbourhood structure: $N_k; k = 1, 2, 3, 4$ as defined in Section 3.3.
Let $\rho$ be the probability of applying VNS to an organism.
Let $E_i$ be the $i$th organism in $E$
Let $T_{max}$ be the maximum number of iterations
for $i \leftarrow 1$ to $Eco\_size$ do
    $r \leftarrow rand[0, 1]$                ▷ $rand[0, 1]$ is a random number between 0 and 1
    if then$r \leq \rho$
        Set $k = 1$
        while $k \leq 4$ do
            for $j \leftarrow 1$ to $T_{max}$ do
                Generate a solution $E_i'$ from the $k$th neighbourhood of $E_i$; $E_i' \in N_k(E_i)$
                Apply Simulated Annealing using $E_i'$ as the initial solution to obtain $E_i''$
                if then$E_i''$ is better than $E_i$
                    $E_i \leftarrow E_i''$
                    Continue search in $N_k$
                else
                    $k = k + 1$
                end if
            end for
        end while
    end if
    Apply variable neighbourhood search
    VNS(E)
    $g \leftarrow g + 1$
end for
return $E_{best}$
```

4 Performance Evaluation

This section presents the performance evaluation of the proposed VM scheduling algorithm.

4.1 Experimental Setup

This study used CloudSim simulator toolkit [43] for performance evaluation of the proposed technique. The simulator toolkit is the most popular simulator used by cloud computing researchers because of its support modeling and simulation of cloud computing infrastructures [44]. It provides support for simulating resource management and scheduling algorithms through its cloud information service and DataCenterBroker components for the realization of resource discovery and information exchange.

Table 1 Workload traces characteristics in CPU utilization

Instance name	Date	Number of VMs	Mean (%)	STD (%)
t_1	03/03/2011	1052	12.31	17.09
t_2	06/03/2011	898	11.44	16.83
t_3	09/03/2011	1061	10.70	15.57
t_4	22/03/2011	1516	9.26	12.78
t_5	25/03/2011	1078	10.56	14.14
t_6	03/04/2011	1463	12.39	16.55
t_7	09/04/2011	1358	11.12	15.09
t_8	11/04/2011	1233	11.56	15.07
t_9	12/04/2011	1054	11.54	15.15
t_{10}	20/04/2011	1033	10.43	15.21

The development of new algorithms is carried in the DataCenterBroker which equally provides support for the development of energy-aware algorithms (Table 1).

The simulated IaaS cloud platform is configured with two data centers, 200 hosts of two types, each type consisting of 100 hosts. The configuration settings of the hosts are given as Table 2. Equation 8 is used to estimate the energy consumption of each host with $E_{\max}$ value of 259 W. The popular workload from monitoring facilities of PlanetLab was used to generate tasks for VMs which is available in CloudSim simulator. The workload primarily contains CPU utilization of VMs across 500 data centers for about 5 min. The workload traces were collected for 10 days, and the characteristics of the traces for each day are shown in Table 1. All the workload traces were used for the evaluation of the proposed algorithm. The study adopted a task scheduling algorithm of CloudSim for scheduling of tasks on VMs.

The performance of SOS and SOS-VNS algorithms was carried using similar conditions. Each algorithm is run independently for 30 times using each workload. For each run, 100 organisms (individuals) constitute the ecosystem (population) which evolves for 100 generations as the stopping criteria. The values of the benefit factors β_1 and β_2 are, respectively, determined stochastically as either 1 or 2. The probability of applying VNS to an organism is 0.2 and $T_{\max}$ of Algorithm 5. A comparison was conducted in terms of SLAV and energy consumption in the data center to evaluate the performance of the algorithm. Resource utilization and energy consumption are used as the performance metrics for evaluating the proposed technique.

4.2 *Results and Discussion*

This section reports the experimental results obtained from the simulation of the proposed technique. Energy consumption and resource utilization are used as performance metrics to evaluate the efficiency of the proposed SOS-VNS algorithm.

Table 2 Host configuration settings

Host type	Parameter	Value
ProLiant ML110 G3	Number	2
	Processor	3000 MHz * dual-core
	RAM	6 GB
	Storage	8 GB
	Bandwidth	1 GB
	Operating system	Linux
	VMM	Xen
ProLiant DL360 G4p	Number	2
	Processor	3400 MHz * dual-core
	RAM	8 GB
	Storage	6 GB
	Bandwidth	1 GB
	Operating system	Linux
	VMM	Xen

Figures 1, 2, and 3, respectively, show statistics (best, average, and standard deviation) of the energy consumption of the hosts for different workload instances (t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8,t_9,t_{10}). The SOS algorithm allocates VMs using the standard SOS concepts, while SOS-VNS algorithm dynamically allocates VMs to achieve lower energy consumption while improving the utilization of compute resources. Each workload instance was executed 30 times, and the best, average, and standard deviation of energy consumption of the hosts are recorded. As it can be observed from Figs. 1, 2, and 3, the proposed technique produces the server schedules with lower energy consumption. According to the results of the experiments, the SOS and SOS-VNS can the performance of the system in terms of energy consumption and resource utilization. This is due to the strong global search ability of SOS algorithm and robust local search ability of VNS algorithm which equips the SOS-VNS with both exploration and exploitation capability, thereby leading to the enhanced performance. Clearly, the proposed SOS-VNS algorithm produces lower energy consumption as compared to SOS and LR-MMT algorithm [44].

The average CPU, memory, and storage utilization for different number of VMs on workload instance t_4 obtained by SOS-VNS, SOS, and LR-MMt algorithms are shown in Figs. 4, 5, and 6, respectively. The SOS-VNS algorithm outperformed SOS and LR-MMT algorithm in terms CPU, memory, and storage utilization, the SOS-VNS algorithm obtained the highest CPU, memory, and storage utilization averaged 90.52, 86.71, and 86.33%, respectively. The average CPU, memory, and storage utilization obtained by SOS algorithm are 82.89, 81.85, and 80.41%, respectively, while those of LR-MMT algorithm are 73.83, 71.66, and 74.36%, respectively. The proposed algorithm obtained better resource (CPU, memory, and storage) utilization

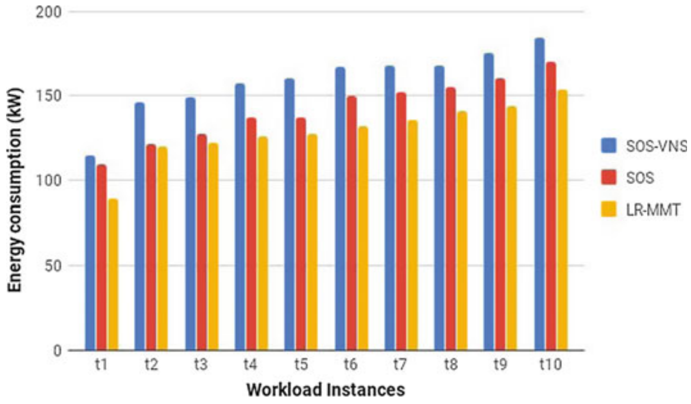


Fig. 1 Best energy consumption

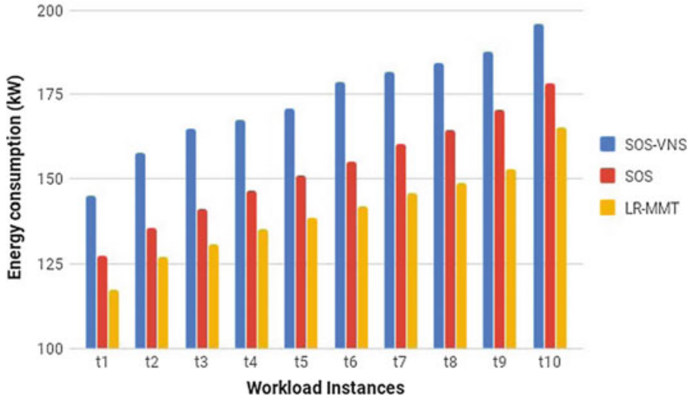


Fig. 2 Average energy consumption

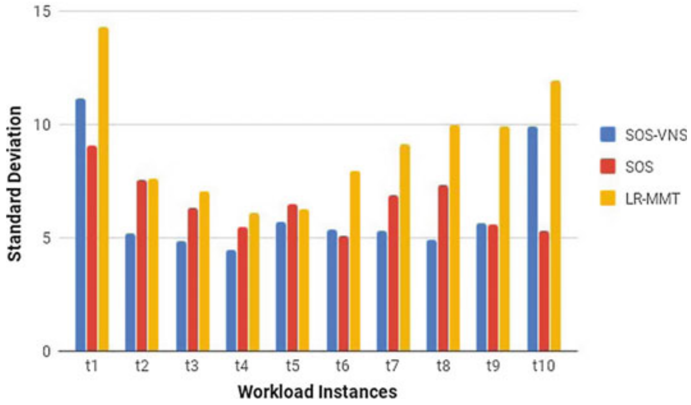


Fig. 3 Standard deviation of energy consumption

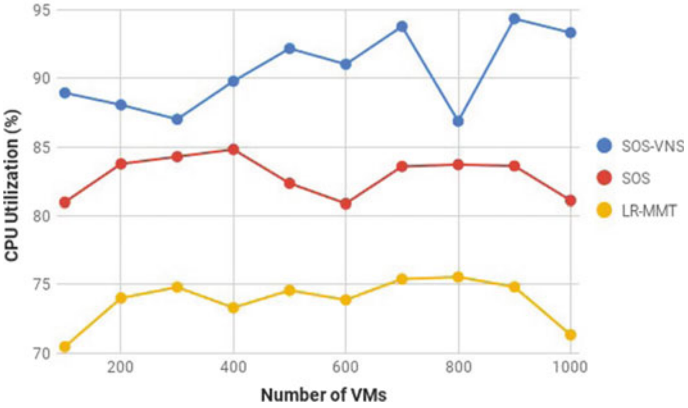


Fig. 4 Average CPU utilization

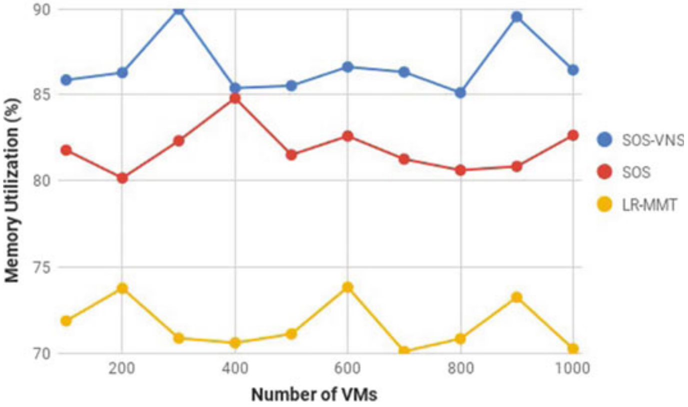


Fig. 5 Average memory utilization

compared to SOS and LR-MMT algorithm. This higher performance is likely due to the efficient local search capability of VNS algorithm integrated into SOS algorithm. Utilizing the SOS-VNS algorithm to optimize energy consumption parameters of the cloud model can improve the energy consumption optimization for cloud computing data centers.

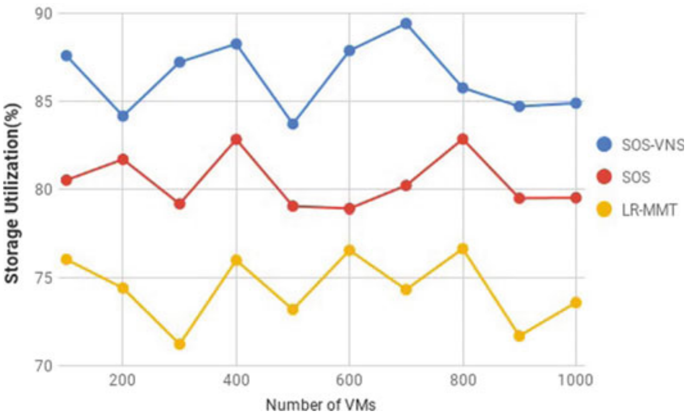


Fig. 6 Average storage utilization

5 Conclusion and Future Work

High energy consumption of data centers has been a major concern for cloud for researchers and cloud stakeholder, the cost of settling energy bills constitutes a significant cost of running cloud systems. This concern motivated the development of an efficient SOS-VNS algorithm for energy-efficient management in cloud computing. The optimal VM allocation has been achieved with minimal number of active hosts and turning off the idle servers for energy saving. VM scheduling problem is a NP-Complete problem, and a SOS-VNS algorithm has been proposed to solve this problem. The schedule of VMs is constructed using symbiotic organism search algorithm which is characterized by strong global convergence to optimal solution. The strong global convergence of SOS makes it suitable for optimizing large-scale problems. This is a great advantage over heuristic algorithms which fail to find optimal solutions for relative large-scale cloud computing scheduling problems. The SOS-VNS algorithm is applied IaaS cloud system of different sizes and characteristics. The results of the experiment prove that SOS-VNS algorithm is able to minimize number of active servers, resource balancing, improve resource utilization, and reduction in consumption. The enhanced performance is enabled by robust local search ability of variable neighborhood search. Conclusively, the SOS-VNS is an efficient approach to VM scheduling problem. As part of future work, rigorous experimentation of the proposed technique can be considered. Also, design and development of efficient VM allocation algorithms for large-scale heterogeneous and distributed data centers will be our focus of future.

References

1. Chandrashekar DP (2015) Robust and fault-tolerant scheduling for scientific workflows in cloud computing environments. Ph.D. thesis
2. Poola D, Ramamohanarao K, Buyya R (2014) Fault-tolerant workflow scheduling using spot instances on clouds. Proc Comput Sci 29:523–533
3. Vouk MA (2008) Cloud computing–issues, research and implementations. J Comput Inf Technol 16(4):235–246
4. Caron E, Desprez F, Loureiro D, Muresan A (2009) Cloud computing resource management through a grid middleware: a case study with diet and eucalyptus. In: IEEE international conference on cloud computing, 2009. CLOUD'09, IEEE, pp 151–154
5. Lei H, Wang R, Zhang T, Liu Y, Zha Y (2016) A multi-objective coevolutionary algorithm for energy-efficient scheduling on a green data center. Comput Oper Res 75:103–117
6. Duan H, Chen C, Min G, Wu Y (2017) Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems. Future Gener Comput Syst 74:142–150
7. Kolodziej J, Khan SU, Xhafa F (2011) Genetic algorithms for energy-aware scheduling in computational grids. In: 2011 international conference on P2P, parallel, grid, cloud and internet computing, IEEE, pp 17–24
8. Achary R, Vityanathan V, Raj P, Nagarajan S (2015) Dynamic job scheduling using ant colony optimization for mobile cloud computing. In: Intelligent distributed computing, Springer, pp 71–82
9. Abdullahi M, Ngadi MA et al (2016) Symbiotic organism search optimization based task scheduling in cloud computing environment. Future Gener Comput Syst 56:640–650
10. Zhong SB, He ZS (2010) The scheduling algorithm of grid task based on pso and cloud model. Key Eng Mater 439:1487–1492
11. Geng J, Huang ML, Li MW, Hong WC (2015) Hybridization of seasonal chaotic cloud simulated annealing algorithm in a svr-based load forecasting model. Neurocomputing 151:1362–1373
12. Ibrahim H, Aburukba RO, El-Fakih K (2018) An integer linear programming model and adaptive genetic algorithm approach to minimize energy consumption of cloud computing data centers. Comput Electr Eng 67:551–565
13. Liu XF, Zhan ZH, Deng JD, Li Y, Gu T, Zhang J (2016) An energy efficient ant colony system for virtual machine placement in cloud computing. IEEE Trans Evolut Comput
14. Sharma N, Guddeti RM (2016) Multi-objective energy efficient virtual machines allocation at the cloud data center. IEEE Trans Serv Comput
15. Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112
16. Dosoglu MK, Guvenc U, Duman S, Sonmez Y, Kahraman HT (2016) Symbiotic organisms search optimization algorithm for economic/emission dispatch problem in power systems. Neural Comput Appl pp 1–17
17. Secui DC (2016) A modified symbiotic organisms search algorithm for large scale economic dispatch problem with valve-point effects. Energy 113:366–384
18. Duman S (2016) Symbiotic organisms search algorithm for optimal power flow problem based on valve-point effect and prohibited zones. Neural Comput Appl pp 1–15
19. Zamani MKM, Musirin I, Suliman SI (2017) Symbiotic organisms search technique for svc installation in voltage control. Indones J Electr Eng Comput Sci 6(2):318–329
20. Tran DH, Cheng MY, Prayogo D (2016) A novel multiple objective symbiotic organisms search (mosos) for time-cost-labor utilization tradeoff problem. Knowl-Based Syst 94:132–145
21. Abdullahi M, Ngadi MA (2016) Hybrid symbiotic organisms search optimization algorithm for scheduling of tasks on cloud computing environment. PLoS ONE 11(6):e0158229
22. Abdullahi M, Ngadi MA, Dishing SI (2017) Chaotic symbiotic organisms search for task scheduling optimization on cloud computing environment. In: 6th ICT international student project conference (ICT-ISPC), IEEE, 2017, pp 1–4

23. Panda A, Pani S (2016) A symbiotic organisms search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems. Appl Soft Comput 46:344–360
24. Prayogo D, Cheng MY, Prayogo H (2017) A novel implementation of nature-inspired optimization for civil engineering: a comparative study of symbiotic organisms search. Civ Eng Dimens 19(1):36–43
25. Dib NI (2016) Design of linear antenna arrays with low side lobes level using symbiotic organisms search. Prog Electromagn Res B 68:55–71
26. Nanda SJ, Jonwal N (2017): Robust nonlinear channel equalization using wnn trained by symbiotic organism search algorithm. Appl Soft Comput
27. Wu H, Zhou Y, Luo Q, Basset MA (2016) Training feedforward neural networks using symbiotic organisms search algorithm. Comput Intell Neurosci 2016
28. Hansen P, Mladenović N (2001) Variable neighborhood search: principles and applications. Eur J Oper Res **130**(3):449–467
29. Hansen P, Mladenović N, Urošević D (2006) Variable neighborhood search and local branching. Comput Oper Res **33**(10):3034–3045
30. Gasior J, Seredyński F (2013) Multi-objective parallel machines scheduling for fault-tolerant cloud systems. In: International conference on algorithms and architectures for parallel processing, Springer, pp 247–256
31. Jung D, Suh T, Yu H, Gil J (2014) A workflow scheduling technique using genetic algorithm in spot instance-based cloud. KSII Trans Internet Inf Syst 8(9)
32. Tawfeek MA, El-Sisi A, Keshk AE, Torkey FA (2013) Cloud task scheduling based on ant colony optimization. In: 2013 8th international conference on computer engineering & systems (ICCES), IEEE, pp 64–69
33. Ramezani F, Lu J, Hussain FK (2014) Task-based system load balancing in cloud computing using particle swarm optimization. Int J Parallel Prog 42(5):739–754
34. Raghavan S, Sarwesh P, Marimuthu C, Chandrasekaran K (2015) Bat algorithm for scheduling workflow applications in cloud. In: 2015 international conference on electronic design, computer networks & automated verification (EDCAV), IEEE, pp 139–144
35. Madni SHH, Latiff MSA, Abdullahi M, Usman MJ et al (2017) Performance comparison of heuristic algorithms for task scheduling in Iaas cloud computing environment. PLoS ONE 12(5):e0176321
36. Madni SHH, Latiff MSA, Coulibaly Y et al (2016) Resource scheduling for infrastructure as a service (iaas) in cloud computing: challenges and opportunities. J Netw Comput Appl 68:173–200
37. Vasudevan M, Tian YC, Tang M, Kozan E, Zhang X (2018) Energy-efficient application assignment in profile-based data center management through a repairing genetic algorithm. Appl Soft Comput 67:399–408
38. Fernandez-Caro D, Fernández-Montes A, Jakóbik A, Kołodziej J, Toro M (2018) Score: simulator for cloud optimization of resources and energy consumption. Simul Model Pract Theory 82:160–173
39. Luo J, Li X, Chen M (2014) Hybrid shuffled frog leaping algorithm for energy efficient dynamic consolidation of virtual machines in cloud data centers. Expert Syst Appl 41(13):5804–5816
40. Greenberg A, Hamilton J, Maltz DA, Patel P (2009) The cost of a cloud: research problems in data center networks. ACM SIGCOMM Comput Commun Rev
41. Liu X-F, Zhan Z-H, Deng JD, Li Y, Gu T Zhang, J (2018) An energy efficient ant colony system for virtual machine placement in cloud computing. IEEE Trans Evolut Comput
42. Vanneschi L, Henriques R, Castelli M (2017) Multi-objective genetic algorithm with variable neighbourhood search for the electoral redistricting problem. Swarm Evolut Comput
43. Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R (2011) Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw: Pract Exp 41(1):23–50
44. Beloglazov A, Abawajyb J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of datacenters for cloud computing. Future Gener Comput Syst

Energy Savings in Heterogeneous Networks with Self-Organizing Backhauling

Nasir Faruk, Abdulkarim A. Oloyede, Abubakar Abdulkarim, Lukman A. Olawoyin and Yinusa A. Adediran

Abstract Heterogeneous network (HetNet) deployment of a large number of low-power small base stations (SBS) is expected to meet up coverage and capacity challenges arising from the global upsurge of mobile data traffic volumes, driven mostly by increase of data-intensive devices, such as smartphones and tablets. However, effective backhaul implementation for the SBS still remains the main bottleneck, as the ever-increasing SBS density will lead to a more complex backhauling and, as such, increased risk of raising capital, and operational and network energy costs. In this paper, the conventional microwave backhauls are compared with self-backhauling for typical dense and spare environments. A heterogeneous network backhaul-energy model is proposed and used to investigate the energy efficiency of the two systems. The impacts of network traffic load and small cells density on power consumption for both backhaul systems were investigated. Furthermore, we present a break-even power point and load threshold level for safe operating regions toward achieving optimum utilization of self-backhauling in a way for higher energy-efficient and sustainable networks compared to traditional homogeneous macro network deployments.

Keywords Base station powering · Self-backhauling · Small cells · Energy saving · Heterogeneous networks

N. Faruk (✉) · A. A. Oloyede · L. A. Olawoyin
Department of Telecommunication Science, University of Ilorin, P.M.B 1515, Ilorin, Kwara State, Nigeria
e-mail: faruk.n@unilorin.edu.ng

A. A. Oloyede
e-mail: oloyede.aa@unilorin.edu.ng

L. A. Olawoyin
e-mail: lolawoyin@unilorin.edu.ng

A. Abdulkarim · Y. A. Adediran
Department of Electrical and Electronics Engineering, University of Ilorin, P.M.B 1515, Ilorin, Kwara State, Nigeria
e-mail: abdulkarim.a@unilorin.edu.ng

Y. A. Adediran
e-mail: adediran.ya@unilorin.edu.ng

T. Herawan et al. (eds.), *Advances on Computational Intelligence in Energy*,
Green Energy and Technology, https://doi.org/10.1007/978-3-319-69889-2_6

1 Introduction

The demand for wireless data services, coupled with the global upsurge in mobile connected devices, has created a capacity challenge for the next-generation networks. The number of global mobile connected devices is increasing and has exceeded the world population. This is expected to reach 24 billion in 2019 as predicted by Cisco Visual Networking Index [1]. This has pushed the world's Information Communication Technology (ICT) systems' power consumption to grow by 8% in 2013 from 2% as it was in 2007 [2]. A large share of this growth is due to the power consumption of mobile networks. Globally, mobile network operators (MNOs) have been challenged by this increase of mobile data traffic which is mostly driven by the adoption of data-intensive devices, such as smartphones, leading to continuous network capacity and coverage enhancements so as to meet users' quality of experience thereby to remain competitive and sustain the market share. Network densification using small base stations (SBS) is highly anticipated to meeting up this constraint [3]. This is achieved by replacing the high-powered macro base stations (MBS) with numerous SBS. This approach was found to be efficient in meeting up the network capacity constraints. However, increasing the number of base station (BS) sites would, in turn, increase the energy consumption of the network, rendering the approach to be unsustainable. Typically, in developed countries with stable power supplies, the energy consumption costs for the MNOs contribute to 15% relative to the total network operation expenses [4]. This could exceed 50% in countries, mostly in Africa where there are power generation deficits. In most of these countries, the energy supply is far below the demand; this forced the MNOs to be operating mainly off-grid sites thereby relying on alternative power sources such as the diesel generators and recently solar and battery inverters. Moreover, the on-grid sites still experienced irregular power supply.

Most implementations of the energy-saving strategies are mainly in the radio access network (RAN) and cooling. Some of these include: remote radio units (RRU), radio standby mode, passive cooling, advanced climate control for air conditioners, higher efficiency rectifiers, and DC power system ECO mode [5], little or no effort has been devoted to backhauling. The strategy of scaling network capacity through heterogeneous network (HetNet) deployment of a large number of low-power small base stations (SBS) to complement existing macro base stations (MBS) umbrella coverage (see Fig. 2) is an approach that also has the potential to minimize overall network energy consumption [6]. However, while energy-efficient operation is possible in the small cell radio access, there is always a need for powering in the high-capacity backhaul connection between SBS and the mobile core network.

Currently, SBS backhauling is mostly implemented using point-to-point (PtP) or point-to-multipoint (PtMP) microwave radio links, due to unavailability of cabling and prohibitive cost of using a wired backhaul link toward each SBS [7]. The need for powering of the SBS backhaul links would, in turn, increases risk of raise in capital and operational costs associated with the energy consumption of the SBS sites beyond what could be supported with low-cost renewable energy solutions. Some efforts in

[8–10] were made to investigate the effect of backhaul power consumption and it was established that this cannot be neglected as it affects the total power of a HetNet. The increased SBS deployment density would, in turn, requires more backhauling and almost certainly accompanied by increased overall energy consumption mostly attributed to the SBS backhauling. In addition, complexities of the backhaul network could be a big issue since the SBSs would be placed both indoors and outdoors and, considering the usual height of the SBSs that could be in the range 3-6 m above the ground level, the presence of building structures and urban clutter could, therefore, make line-of-sight (LOS) configuration for PtP very challenging. The time required for setting up the network, as well as manpower and network management systems could also be challenging. Self-organizing backhaul networks [11] or self-backhauling is a general concept where the MBS provides backhauling to SBSs via the existing macro RAN. This automation would mitigate backhaul connectivity and management bottlenecks associated with existing backhaul system such as the PtP. Furthermore, it would help in faster deployment and network rollout with minimal human intervention.

This paper, therefore, proposes an energy-efficient backhaul solution for next-generation wireless systems. The conventional microwave backhauls are compared with self-backhauling for typical dense and sparse environments. The paper also presents a break-even power point and the load threshold level for safe operation regions toward achieving optimum utilization of self-backhaul deployment and green backhauling for the deployment of HetNets in a way that is more energy-efficient and sustainable compared to traditional homogeneous macro network deployments.

2 Base Station Types in HETNET and Power System Consideration

2.1 Base Station Types in HetNet

HetNet architecture consists of various base station types, each having different coverage and operating functionalities. In a typical LTE-Advanced multitier HetNet rollout, the network may consist of macro, micro, pico, and femto base stations, remote radio heads (RRHs), and as well as relay stations. Each of these base stations, aside the macro BS, is commonly referred to as low-power nodes (LPN). We generalize this concept as small base stations (SBS). Figure 1 shows typical deployment scenario of HetNet which consists of macro base station and small base stations.

I. **Macro Base Station (MBS)**

Macro base stations are deployed outdoors and are able to provide 2G, 3G, and 4G services in the range of kilometers. They are backhauled via fiber optics and/or microwave. A macro base station consists of three main segments: the transmission equipment (TE), cooling equipment, and auxiliary equipment (AE). The TE includes

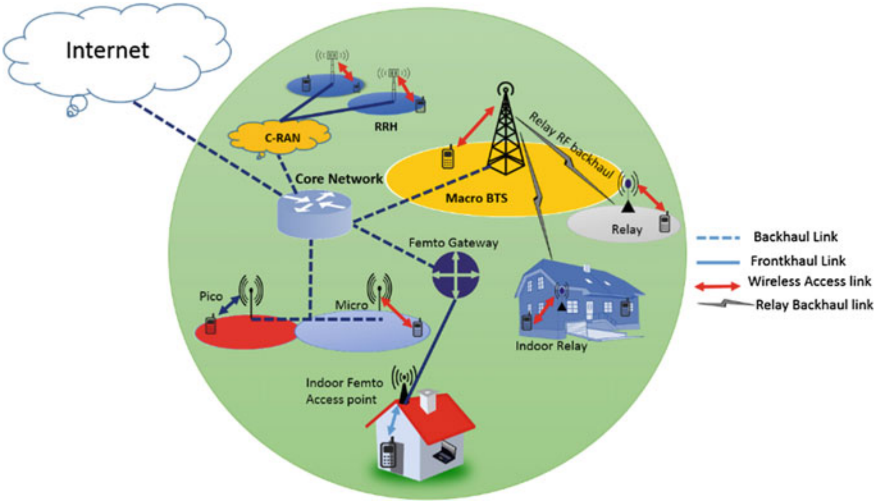


Fig. 1 Typical heterogeneous network deployment scenario

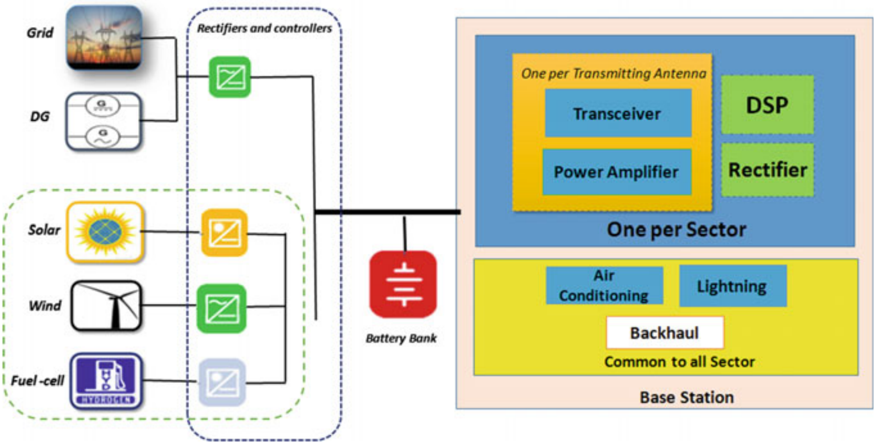


Fig. 2 Block diagram of base transceiver station (BTS) site and complete power system

the power amplifiers (PA), transceivers (TRX), digital signal processing (DSP), and rectifiers, and these are deployed per sector or cell at a given BS site as shown in Fig. 2.

Therefore, the overall power consumption budget of the BS site is per sector consumption multiplied by the number of sectors and the number of TXR per sector in the case of MIMO antenna systems. The cooling systems, lightning, security alarms, and backhauls are all common to all sectors and are deployed to serve the whole BS site. Among these components, the PA and air conditioners (ACs) are the most power-consuming components with AC consuming about 50–60% of the total power consumed by the BS site (see for instance, measurement conducted in [12]).

II. **Small Base Stations (SBS)**

Compared to MBS, a micro BS consumes less power since it does not need power hungry cooling solution. Moreover, the power consumption of the PA, which is a function of the radiated power is also reduced significantly due to smaller coverage footprint of the micro BS. The power consumption of pico and femto BSs is even a smaller fraction of the MBS power consumption (see Table 1), as only TRX, PA, microprocessor, and FPGA are present to make up the functioning BS. Moreover, omnidirectional (single sector) antenna configuration is being used. Pico BS could be deployed outdoor or indoor, while Femto BS is completely meant for indoor deployments in the residential or business premises. Relay nodes (RNs) are used to improve coverage at cell edge of the existing network. It is defined in LTE-A relay standard [13]. This could be deployed in indoor or outdoor locations and could exhibit similar features of the femto cells. RNs are backhauled using the macro RAN. Remote radio heads (RRHs) are high powered, low weights nodes with the capability of reducing cost, improving efficiency and performance when deploying new base stations. RRHs can operate in multimode where different standards such as GSM, HSPA, LTE, and WiMAX could be operated [14]. They can be used to extend the coverage of BTS/NodeB/eNodeB in rural locations and tunnels. Table 1 provides the power consumption and coverage range for different base station types.

2.2 *Power System Consideration of BS Sites*

The geographical location, deployment sites (e.g., urban, suburban, or rural rollout), and climatic condition are some factors that may influence the power requirements for a BS site. In tropical dry climates (e.g., Nigeria), two ACs with minimum power capacity of 2500 W each may be needed to maintain the indoor temperature of the BS shelter, usually, below 25 °C [18]. This is to avoid excessive heat dissipation from the TE; this means additional cooling for the SBS which was originally omitted from

Table 1 Base station types in HetNet [12, 15–17]

Base state type	Coverage	Radiated power (W)		Power consumption (W)	Commonly used backhaul type
		Indoor	Outdoor		
Macro	<35 km	–	5–40	1000–5000	Fiber/Microwave
Micro	<2 km	–	0.5–2	100–300	Microwave
Pico	<200 m	0.1	0.25–2	9–15	Microwave/Fiber/DSL
Femto	10–15 m	<0.1	–	6–14	Fiber/DSL
RRH	<2 km	–	5–20	550–760	Fiber
Relay	<2 km	<1	0.25–7	10–120	via DeNB

the new green design architecture (Refer to [9 and 16] for EARTH project). However, deployments in polar climate may not require any cooling as the climate is characterized by average temperatures below 10 °C throughout the year, with temperatures typically ranging from −47 °C in February to −11 °C in August [19]. Base stations TEs in these locations could therefore be cooled by natural air convention. It is on this note that the BS power requirement varies with geographical locations due to differences in climatic conditions. In addition, temporal variability of the traffic density associated with the location-dependent population distribution is another factor as the BS consumes more power during the peak hours [15, 17, 20].

Provision of sustainable energy source to the BS sites, while minimizing the network energy cost which constitutes about 60% of the total network OPEX, has been a major bottleneck in most emerging markets. The requirement of achieving 99.95% availability necessitates the MNO to provide power 24 × 7 throughout the year, thus the need for reliable power supply. However, developing countries are faced with limited grid power infrastructure, limited grid coverage, and erratic power supply. The unreliability of the grid power and grid reach (coverage) forces the MNOs to rely on diesel generators to power up their BS sites. In some situations, diesel generators (DGs) are used as both primary and backup power sources depending on the grid and power supply situation. For example, in Europe and North America, one diesel generator is used for emergency backup, while two generators are used in developing countries for primary and backup [21]. For off-grid sites, DGs act as the primary source only as efforts have been made recently to cut down on the carbon footprint, thus making the MNOs to look inward to renewable energy sources such as wind and solar. In Fig. 2, we provide different power sources options to a BS site.

3 Small Cells Deployment and Backhauling Options

Network densification through the use of small cells has been identified as the key enabler for the success of next-generation wireless systems [6]. The market growth for small cells have been very encouraging, from less than 2 million units in 2011 to about 4 million units in 2015, and this is expected to grow up to 10 million in 2018 according to Small Cell Forum [22]. From the technical and economic points of view, deployment of small cells would enable the mobile operators to ease significant amount of traffic from the macro base stations and this would, subsequently, reduce the network overhead and increase the network capacity without necessarily, adding new macro sites which are inherently costlier. Small cells could boost spectral efficiency per unit area through spatial reuse of the spectrum, thereby improving both network coverage and capacity. Furthermore, deployment of small cells would drastically reduce the network energy expenditures, both capital (CAPEX) and operation (OPEX). Figure 3 provides an illustration of network densification. However, this paradigm shift from the use of conventional macro cells only to small cells have been seen to improve the network performance but also introduces challenges. For instance, effective backhaul implementation still remains the main challenge as the

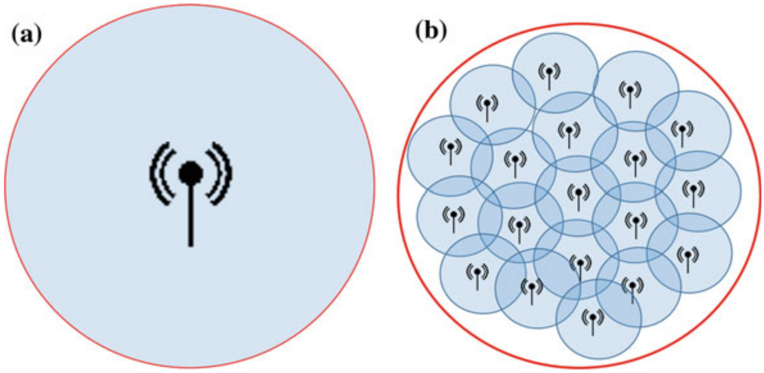


Fig. 3 Network densification (ND) **a** single-tier cellular network. **b** ND with small cells

ever-increasing number of small cells or SBS to meet up the need to support more services may lead to a more complex backhaul and, as such, increase risk of rising operational costs.

It is therefore very essential for the network operators when designing backhaul solution to consider scalability, maintainability, fast and efficient installation, and cost [7]. In this section, we provide various backhaul options for SBS deployments and some challenges that may be encountered during rollout.

3.1 Wired Backhaul Options for Small Cells

Wired backhaul solutions have been deployed and proved promising for years for cellular mobile base stations. There are so many options available for small cells such as:

i. **Direct Fiber Optics**

Fiber optics could provide direct end-to-end, point-to-point high speed, and unlimited backhaul capacity with minimal latency. Gigabit passive optical network (GPON), which is based on PtMP technology that incorporates passive filters, could link multiple small cells' nodes. This is particularly attractive in dense urban rollout [10]. Direct fiber backhaul infrastructure for new small cell sites could be cost prohibitive and may take years to roll out, aside difficulties that might arise laying the fiber cables to each closet location of the small cells. Terrain and clutter features along the backhaul-access path could hinder deployments; bypassing such features could increase both CAPEX and manpower costs.

ii. **Digital Subscriber Line (xDSL)**

xDSL is a general term for the broadband access technologies based on digital subscriber line (DSL) technology. It uses the existing copper wires and due to its lower cost compared with fiber, it could be used for small cell backhaul. Cost of deployment may relatively be the same as in the case for fiber, but the speed provided will be very limited. The backhaul capacity of xDSL depends on the technology type and the distance from the exchange. Typical urban rollout is usually 300–400 m from the cabinet [23]. Very-high-bit-rate digital subscriber line 2 (VDSL 2) is based on ITU G.993.2 standard and can provide peak data rates above 100 Mbps [23]. However, this may require careful planning to ensure that the distance between the VDSL2 and the digital subscriber line access multiplexer (DSLAM) does not exceed the maximum 300 m required by the VDSL2.

iii. **Hybrid Backhaul Solutions**

Hybrid approach could also be used where the fiber from the exchange will terminate at the cabinet (i.e., fiber to the cable (FTTC)) and thereafter, VDSL 2 cable will take the capacity to the locations where the small cells are deployed. In the case of fiber to the node (FTTN), each small cell is connected to a VDSL 2 modem that is connected to a DSLAM which is connected to a fiber switch using 1 Gbps point-to-point optical link [23]. This type of solution is usually available for dense urban areas and can provide capacity to urban small cells.

iv. **Cloud Radio Access Network (Cloud-RAN)**

Cloud-RAN (C-RAN) is one candidate technology aimed at solving the challenges related to densification and increased base station cooperation. In C-RAN architecture, remote radio units (RRU) of different cells are connected to the cloud server via a high-speed front-haul link, such as a fiber network [24]. Recent deployment trials by China Mobile [24] provide 30 and 53% CAPEX and OPEX reduction, respectively. Ericsson-LG also showed the possibility of replacing 28 3G NodeBs with 195 RRUs which reduced the total network power consumption by 2/3; from user's experience, the call drop ratio reduced by 70% with increase in throughput gain of 100%. However, the latency requirements for the base band unit (BBU) impose an upper bound for BBU processing time and the front-haul transport latency. Real-time requirement for operating system, clock synchronization, and capacity requirement for the front-hauls could be challenging [24]. Despite this, the network would in practice eliminate the need for local backhaul since end user data traffic will be delivered to and from the centralized location. Still, it will require high-speed fiber front-haul connections to the small cells and therefore, may not be cost-effective for massive small cell deployments particularly in the developing economies.

3.2 Wireless Backhaul Options

The deployment of small cells in outdoor urban and rural areas or inside building where cabling infrastructure is not available makes the adoption of wired backhaul very challenging. It is also important to note that the increasing traffic demands would require the installation of wired backhauls, supposedly fiber, in all new small cell sites locations; this approach may therefore not be cost-effective. Wireless backhauling could therefore provide cost-effective means of connecting these cells.

i. **Microwave Radio links**

Microwave links still remain the most dominant backhaul solution in the mobile industry. It provides an alternative to complement the fiber optics which is inherently costly and almost practically impossible to deploy in some designated locations. With the advent of millimeter waves, i.e., 60 GHz V-band and 70/90 GHz E-band, end-to-end high-capacity backhaul could be achieved. Microwave radio links could operate in line of sight (LOS) or non-line of sight (NLOS) depending on the operating frequency, and these could be in PtP or PtMP topology depending on the choice of deployment and capacity requirements. The LOS typically requires the use of parabolic dishes which could be unsuitable for many small cells since the outdoor small cells are typically 3–6 m above the ground level and direct LOS is not always the case in urban deployments. Moreover, at extremely high frequencies, there is a risk of increased rain attenuation, oxygen absorption, and, possibly, antenna alignment complications. These problems could hinder the performance of the network. From technical design perspectives, the realization of suitable integrated circuits in those bands could be challenging due to nonlinear distortion of power amplifiers, IQ imbalance, and the need for highly directional antennas [25]. The need for more backhauls as the cells become dense would, in turn, consume more energy and also increase the complexity in designing the network.

ii. **Satellite Backhaul**

This type of solution is preferred in remote areas where fiber, copper, or microwave solutions are uneconomical to deploy. Satellites have universal availability and could provide data rate up to 350 Mbit/s with the size of the dish and power amplifier constraints. A typical capacity that a satellite can provide to small cells will be in the range 1 M–10 Mbps [22]. The most commonly used satellite band is Ku-band which has operating frequency of 10–12 GHz and is slightly affected by rain. The lower-frequency bands (i.e., C-band: 4–6 GHz) is practically unaffected by weather conditions. However, the 20–30 GHz Ka-band, which is predominantly used, is heavily affected by weather and fading [23]. In addition, delay and jitter are constraints to deployment of satellite backhaul. For example, typical expected average jitter is between 5 and 50 ms [22], while the round trip signal propagation delay could be 240–260 ms depending on the small cell location, hub site and, also the satellite. Moreover, packetization and processing delay is typically 35–50 ms, which would increase the total delay to 275–310 ms far, beyond the acceptable thresholds [22]. The adoption of satellite for small cell backhauling has not been widely reported, but there is a huge market potential for the system.

iii. Free Space Optics

Free space optics (FSO) [26] is given attention due its associated lower cost as no underlaying fiber cabling is needed. In addition, very high-capacity data rates could be achieved with minimal interference. However, atmospheric disturbance, such as absorption and scattering, could hinder the signal propagation. Most importantly, eye safety regulations for the use of laser beams, designing backhaul systems in dense environments while meeting up with the international safety regulations, and specifications defined by international standard could be very challenging.

iv. TV White Space

Unused portion of the electromagnetic spectrum within the TV bands is referred to as TV white space (TVWS). Secondary operation on TVWS is still under process although some countries, such as USA, UK, Japan, Finland, and Singapore, have already developed a regulatory framework for its use [23]. Substantial research studies and piloting have been conducted globally; there are also success stories from Africa [27]. Due to its favorable propagation characteristics, TVWS has been found to be very attractive for small cell deployment [28]. The availability of TVWS varies with location and the transmit power of the primary users (i.e., TV broadcast stations); these could limit the number of available TV channels to be used in dense environments. Although considerable capacity up to 18 Mb/s with 600 MHz band could be achieved in NLOS, radio design, and secondary user coexistence to comply with the regulator standard on interference could still pose some challenges.

v. Long-Range Wi-Fi

Wi-Fi was primarily designed for local area networks and it uses the 2.4 and 5 GHz ISM radio bands to provide telecommunication services such as VoIP. This technology is based on 802.11 standards and has recently been proved to provide cost-effective communication for long-distance applications. Very long-distance PtP Wi-Fi network was proposed wherein 279 and 133 km links were successfully connected in Venezuela and in northern Italy, respectively [29]. The long-range capabilities of Wi-Fi networks could be leveraged to provide backhaul services to small cell sites, particularly given the constraints where fiber POP is unlikely to be deployable due to the small cell locations and microwave point-to-point (PtP) links that could exacerbate both the OPEX and CAPEX costs. Sites acquisition, backhaul installation, and maintenance will drastically reduce and no license fee for the spectrum is required since it operates in the ISM band. However, long-range Wi-Fi backhaul would require LOS. It is also vulnerable to interference since it operates in an unlicensed band, though enhanced technologies such as the adaptive directional antennas coupled with smart meshing and predictive channel management [30] could drastically minimize the interference. However, in typical dense urban deployments where hundreds of cells could be deployed within a close proximity, the backhaul interference will be very high and unsustainable. Moreover, the predictive channel management may not attain maximum throughput due to the available frequency pool within the 2.4 and

5 GHz ISM bands. Therefore, interference avoidance by simply frequency hopping may still not yield sufficient gain.

vi. **Self-Backhauling of Small Cells**

Self-backhauling is a general concept where a macro base station (MBS) provides backhaul to a small base station (SBS) via the existing macro RAN. This has already been introduced by mobile standardization bodies, such as in the Third Generation Partnership Project (3GPP) in the form of the relaying concept standardized for 3GPP Release 10 LTE-Advanced networks [13]. The relay base stations are low-power nodes that enhance achievable performance of macro UEs in areas of poor coverage, such as the cell edge. In 3GPP standard, the relay node backhaul is connected to a "donor" macro cell. The LTE or LTE-Advanced (Release 8/9 or 10) UEs are then connected to a macro cell via the relay access network within a relay BS coverage area. Self-backhauling generalizes this approach by considering other radio access technologies (not only LTE or LTE-Advanced) in both the SBS backhaul and SBS radio access. As a result of this flexibility in radio technology selection, self-backhauling is able to be implemented with little or no impact on the existing standards. For instance, Qualcomm proposed a so-called Velcro relays self-backhauling approach, whereby macro LTE provides backhauling for evolved 3G BS (e.g., HSPA+) to efficiently exploit the spare LTE resources in places with low LTE UE penetration [31]. Recent deployment case studies in Addis Abba show the feasibility of self-backhauling [32]. When compared with other backhaul systems, self-backhauling could help in reducing the cost of deployment as a result of minimal on-site installation efforts from the operators. This could drastically also reduce the cost of maintenance; moreover, network management and control entities such as the radio resource management and security management are been shared. The RAN mainly is operating on lower frequencies usually below the 6-GHz bands. This would further simplify the installation cost, time, and efforts since the string LOS requirements that could demand careful installation is no longer needed, thereby making the deployment faster and flexible for dense urban environments. The need for building another infrastructure as in the case of fixed-wireless backhauling is also not needed since self-backhauling reuses the macro site infrastructures.

Backhaul capacity could be an issue in self-backhauling, since the base station serves multiple small cells, and access and backhaul links shared the total pool of the available resources. This might cause capacity bottleneck. Also, with the large number of SBS, signaling load and excessive handover could manifest. However, there have been active research efforts trying to mitigate these problems. In [33], significant spectral efficiency and throughput gains was achieved for the SBS backhaul through the use of selected CoMP technique under realizable feedback overhead, even under feedback bit error. In [34], LTE-based backhaul concept (self-backhauling) for heterogeneous networks was simulated using microwave frequencies for two virtual cities (European and USA) and sufficient capacity was obtained.

4 System Concept

The system concept introduced in this paper, as illustrated in Fig. 4, consists of a conventional fixed microwave backhauling and the self-backhauling. The first scenario is a fixed-wireless backhauling which is achieved with PtP or PtMP radio links. In most cases, this requires LOS clearance between the radio links backhaul (wireless hub) and the SBS due to operation in higher-frequency bands (Fig. 4a). The LOS requirement for fixed-wireless backhauling links can be relaxed through use of alternative multihop topologies (e.g., mesh) to route the fixed-wireless links around shadowing objects (e.g., buildings, mountains, etc.). In the second scenario, we assume the SBS to be distributed within the MBS coverage area. The MBS users (MBS UE) and SBS users (SBS UE) are randomly distributed within the MBS and SBS coverage areas, respectively, while the SBSs are uniformly distributed. The MBS radio access link is used to transmit the backhaul traffic between the SBS and MBS. However, due to the capacity requirements of the traffic aggregates from multiple SBS within the coverage area of the MBS, all other UE traffics both from the MBS and SBS are routed to the core network via fiber links as shown in Fig. 4b. For fairness, in the two scenarios, we assumed that all the SBS transmits on the same power level, and therefore under normal condition, their power consumption, sizes of their coverage footprint are expected to be same. Moreover, the number of SBSs and the aggregated backhaul capacity for both scenarios are also assumed to be the same. The SBSs are connected to the MBS and RRL with star topology. It is a natural connection configuration since star topology is more power efficient than ring and mesh [9].

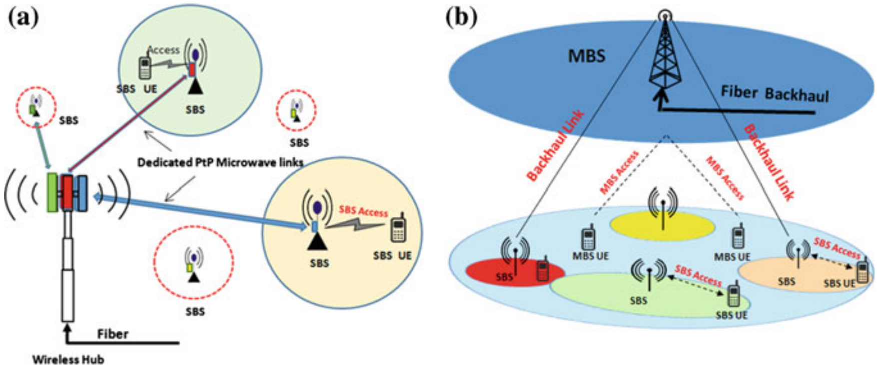


Fig. 4 Backhauling SBS using **a** fixed-wireless microwave backhauling and **b** self-backhauling

5 Backhaul-Energy Model

Using the HetNet architecture presented in Fig. 4, the power consumption of each HetNet scenario is the sum of the power consumption of the SBS and the backhauling. Let $P_{c,t}^{\text{bh}}$ and $P_{\text{SBS},t}^{\text{bh}}$ represent the power consumption of the microwave unit(s) at the wireless hub and SBS, respectively. The total power consumption of microwave radio link (RRL) $P_{\text{tot},t}^{\text{bh}}$ can be written as:

$$P_{\text{tot},t}^{\text{bh}} = P_{c,t}^{\text{bh}} + P_{\text{SBS},t}^{\text{bh}} \tag{1}$$

Let SBS (C_j^{SBS}) be the aggregated backhaul capacity of the RRL at the SBS at site j and N_k^{SBS} the total number of RRLs at SBS, then the power consumption of the RRL at the central point is derived as:

$$P_{\text{SBS},t}^{\text{bh}} = P_{\text{SBS}}^{\text{agg}}(C_j^{\text{SBS}}) + P_{\text{switch}}^{\text{SBS}}(N_k^{\text{SBS}}, C_j^{\text{SBS}}) \tag{2}$$

$$P_{c,t}^{\text{bh}} = P_c^{\text{agg}}(C^c) + P_{\text{switch}}^{c}(N_k^{c}, C^c) \tag{3}$$

A threshold capacity C_{th} is defined to classify two traffic conditions: low and high demands. With this assumption, the aggregated power consumption limits for the central hub $P_c^{\text{agg}}(C^c)$ and SBS $P_{\text{SBS}}^{\text{agg}}(C_j^{\text{SBS}})$ are derived as follows:

$$P_{\text{SBS}}^{\text{agg}}(C_j^{\text{SBS}}) = \begin{cases} P_{\text{low}}, & \text{if } C_j^{\text{SBS}} \le \text{C}_{\text{th}} \\ P_{\text{high}}, & \text{otherwise} \end{cases} \tag{4}$$

$$P_{\text{switch}}^{\text{SBS}}(N_k^{\text{SBS}}, C_j^{\text{SBS}}) = \begin{cases} 0, & \text{if } N_k^{\text{SBS}} = 1 \\ P_{\text{switch},t} \times \left[\dfrac{C_j^{\text{SBS}}}{C_{\text{Max-Switch},t}} \right], & \text{otherwise} \end{cases} \tag{5}$$

And

$$P_c^{\text{agg}}(C^c) = \begin{cases} P_{\text{low}}, & \text{if } C^c \le \text{C}_{\text{th}} \\ P_{\text{high}}, & \text{otherwise} \end{cases} \tag{6}$$

$$P_{\text{switch}}^{c}(N_k^{c}, C_j^{c}) = \begin{cases} 0, & \text{if } N_k^{c} = 1 \\ P_{\text{switch},t} \times \left[\dfrac{C_j^{c}}{C_{\text{Max-Switch},t}} \right], & \text{otherwise} \end{cases} \tag{7}$$

In [9], C_{th} was assumed to be 500 Mbps. We, therefore, generalize the total backhaul power consumption for the RRL (i.e., PtP and PtMP) network topology as:

$$P_{\text{tot},t}^{\text{bh}} = \begin{cases} P_{c,t}^{\text{bh}} + N_k P_{\text{SBS},t}^{\text{bh}} \; t \in \text{PtMP} \\ (P_{c,t}^{\text{bh}} + P_{\text{SBS},t}^{\text{bh}}) N_k t \in \text{PtP} \end{cases} \tag{8}$$

where P_{low} and P_{high} are the power consumption of the microwave antennas associated with the low and high traffic conditions, respectively. $P_{\text{switch}}^{\text{SBS}}$ and P_{switch}^{c} represent the power consumption of the switches used at SBS and central hub, respectively. C_j^{SBS} and C_j^c are the total backhaul capacity at the SBS and hub site, while N_k^{SBS} and N_k^c are the respective number of microwave links at the SBS and hub site, respectively. $C_{\text{Max}-\text{Switch},t}$ is the maximum capacity of the switch of type t. $P_{\text{switch},t}$ is the fixed power consumed by switch of type t irrespective of the load, and ψ_{bh} is the backhaul type (PtP or PtMP).

We introduce an indicator variable $\chi_{k,t}^{j}$, the site selector, which indicates whether the backhaul device t located in j is PtP or PtMP type. The total backhaul power consumption could be written as:

$$P_{\text{tot},t}^{\text{bh}} = \sum_{t\in\psi_{\text{bh}}} \sum_{k\in \text{K}_t} P_{\text{tot},t}^{\text{bh}}(N_k)\chi_{k,t}^{j} \tag{9}$$

where k represents the set of backhaul devices of type t, and

$$\chi_{k,t}^{j} = \begin{cases} 1, & \text{if type } t \text{ used in location } j \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Therefore, the power consumption of the HetNet for the SBS backhauled with RRL and also self-backhauled are derived in Eqs. (11) and (12).

$$P_{\text{RRL}}(f_b) = \sum_{i\in M_t} N_i P_i^{\text{SBS}}(fb) + \sum_{t\in\psi_{\text{bh}}} \sum_{k\in \text{K}_t} P_{\text{tot},t}^{\text{bh}}(N_k)\chi_{k,t}^{j} \tag{11}$$

$$P_{\text{Self}}(f_a, f_b) = \sum_{j\in Q_t} N_j P_j^{\text{MBS}}(f_a) + \sum_{v\in S_t} N_v P_v^{\text{SBS}}(f_b) \tag{12}$$

where f_a, f_b, and f are load factors for MBS, SBS, and HetNet, respectively, and the HetNet load factor, f, is given by:

$$f = f_a + f_b \text{ and } f_a + f_b \leq 1 \tag{13}$$

Using Eqs. (11) and (12), the power savings for the self-backhauling compared with point-to-point (PtP) or point-to-multipoint (PtMP) fixed-wireless backhaul is expressed as:

$$\begin{aligned} P_S = \lambda_p \Bigg[& \sum_{i\in M_t} N_i P_i^{\text{SBS}}(fb) + \sum_{t\in\psi_{\text{bh}}} \sum_{k\in \text{K}_t} P_{\text{tot},t}^{\text{bh}}(N_k)\chi_{k,t}^{j} \\ & - \left\{ \sum_{j\in Q_t} N_j P_j^{\text{MBS}}(f_a) + \sum_{v\in S_t} N_v P_v^{\text{SBS}}(f_b) \right\} \Bigg] \end{aligned} \tag{14}$$

The power savings can be represented by an indicator function $1_{D(P_S)}$,

$$1_{D(P_S)} = \begin{cases} \text{Gain}, & \text{if } P_S \geq 0 \\ \text{Loss}, & \text{otherwise} \end{cases} \tag{15}$$

where M and S represent set of SBS types t used in the networks.N_i and N_v represent the total number of SBS of specific type. Q represents the set of MBS type t used as self-backhaul. N_j represents the total number of MBS of specific type t. P_j^{MBS}, P_i^{SBS}, and P_v^{SBS} are the power consumption of MBS and SBS of types t, respectively. f, f_a, and f_b are the instantaneous load on the HetNet, MBS and SBS, respectively, λ_p is the self-backhaul factor representing the percentage of number of sites equipped to have self-backhauling capabilities and p is the number of mobile operator's MBS sites.

6 Results and Discussions

In this work, simple and validated power consumption models presented in [15–17, 35] for macro and micro base stations were adopted. These were used to analyze the power consumption of a typical three-sector macro base station, PtP, and PtMP microwave backhauls. The BS equipment power consumption without cooling and, other auxiliary equipments is between 1 and 2 kW and, that of two ACs as used in most tropical and dry climate regions ranges between 2 and 4 kW [36].

6.1 Typical Power Consumption of Macro BS and Microwave Backhaul Hub Sites

Figure 5 shows the load-dependant power consumption of the MBS with and without backhaul; the peak power is 4.3 kW at maximum traffic load condition. We also present the power consumption of the backhaul system. The backhauling power varies from less than 50 W for fiber backhaul to about 500 W for long-range PtMP microwave links. The most widely used backhaul system in emerging markets is PtP backhaul link which consumes 100–200 W [36]. In this analysis, we use 100 and 500 W for PtP and PtMP backhauls, respectively, while and the load-dependent component is neglected. The results show that for microwave backhaul with 50 PtP backhaul links (i.e., $N_k^c = 50$), the peak power is 5 kW and this increases to 10.025 W when 25% of the backhauls are PtMP. This is power consumption at the hub site only and did not include that of the SBS links (N_k^{SBS}). When SBS links are included, the overall backhaul power requirement for the PtP backhaul system will be 10 kW and that of PtMP will depend on the mapping of PtMP to SBS backhauls.

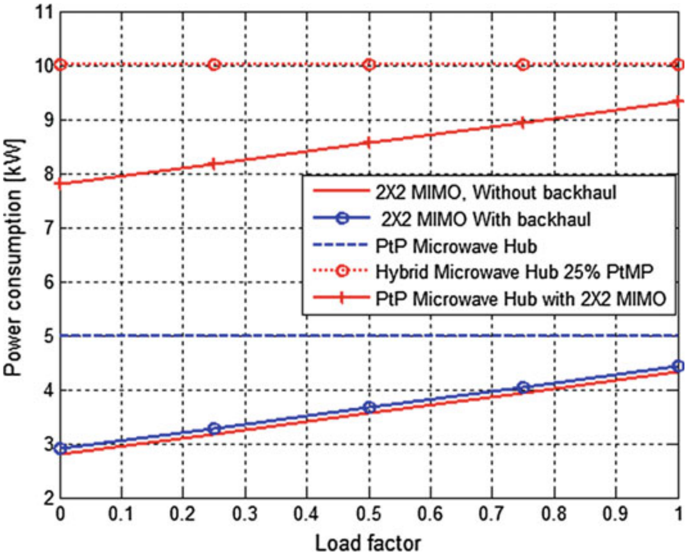


Fig. 5 Power consumption of macro BS and microwave backhaul hub

In Fig. 5, we have also illustrated the power consumption of a hub site with three-sector MBS. In Fig. 6, we show the distribution of power consumption of the BS site. The power consumption share of the BS equipment, cooling, backhauling, and auxiliary equipments are 35, 59, 2, and 4%, respectively. Recent energy-efficient BS equipment consumes less energy and the MBS site's power is between 1 and 1.5 kW [36] as opposed to the 4 kW for the current deployed networks. In the EARTH project, highly energy-efficient power amplifiers for the MBS, SBS, and feeder loss factor of 38.8, 28.5 and 0.5%, respectively, were used [16]. Passive cooling system of 170 W was used for the MBS and no active cooling for SBS [16]. In our analysis, we considered worst-case scenario that could occur in tropical areas, for instance, where 60 W of cooling would be required for the SBS. This is in accordance with the assumption presented in [15].

The transmitter power levels for the MBS and SBS can be assumed to be 20 and 2 W, respectively, and power consumption of the DSP, transceiver, rectifier, and backhaul, respectively, are 29.5, 13, 100, and 35 W [37]. Then, the load-dependent power consumption for three-sector LTE system with 10 MHz bandwidth and 2 × 2 MIMO system for the MBS and SBS gave the highest value at peak load as 1109 and 166.3 W, respectively. These values are few watts higher than the values obtained in the EARTH project [16] as active cooling, which could reduce the power for SBS to about 106 W, was not considered. However, these values are still within the range obtained from various publications for MBS and SBS [16, 37]. Deployment of the SBS has decreased the power consumption significantly from 4300 to 166 W, but the proportion of the backhaul power increased from 2 to 21% which is expected to increase further to 43% if the current 100 W PtP link is used at the SBS site.

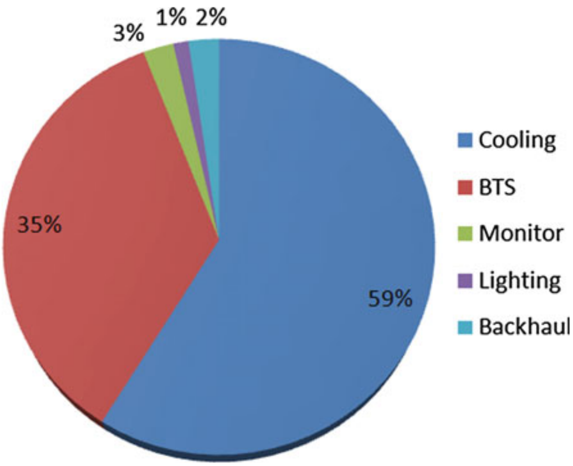


Fig. 6 Distribution of power consumption of a BS site

6.2 *Power Consumption of HetNet and the Break-Even Load*

Using Eqs. (11) and (12), we analyzed a future HetNet power consumption by modeling the whole MBS and SBS power consumption. In this analysis, we consider a HetNet with 10 SBSs. We also considered specific implementation for the energy-efficient microwave links based on realistic power consumption typically, varying from 25 to 50 W [8]. However, in this analysis, we focused more on the average value of 35 W and assumed to be the maximum power consumption of each unit from the backhaul side (i.e., the wireless hub) and the SBSs locations. Even though, it is expected that the power consumption to be higher at the hub than the SBSs. For simplicity, we did not consider the load varying power consumption of the microwave units. In Fig. 7a, we show the power consumption of (i) MSB (macro) cell only without any SBSs deployment, (ii) PtP wireless backhaul only, (iii) SBSs backhauled with microwave links and finally, and (iv) SBSs backhauled with MBS (self-backhauling). Starting with the PtP wireless backhaul only, 25 W microwave units were deployed, since we have neglected the load varying power consumption part, it is easy to deduce that the total power consumption for the hub is 250 W (i.e., 10 PtP links serving 10 SBSs). For the MSB (macro only), the power consumption increases with an increase in traffic load with maximum peak achieved when the traffic load reached 100%. The power consumption at this load is around 1 kW. It should be noted that this is anticipated power consumption when highly efficient systems are used based on the system parameters such as air conditioning with a power rating of 170 W, provided in [37]. The current deployed MBS particularly the ones in temperate regions consumes high power as described I Sect. 6.1.

Furthermore, in Fig. 7a, we have analyzed the power consumption of the two systems backhauling 10 SBSs in the HetNet. For the microwave HetNet systems, the traffic load effect of the SBS manifested on the total power consumption as the power consumption increases with the traffic load which was earlier invariant when

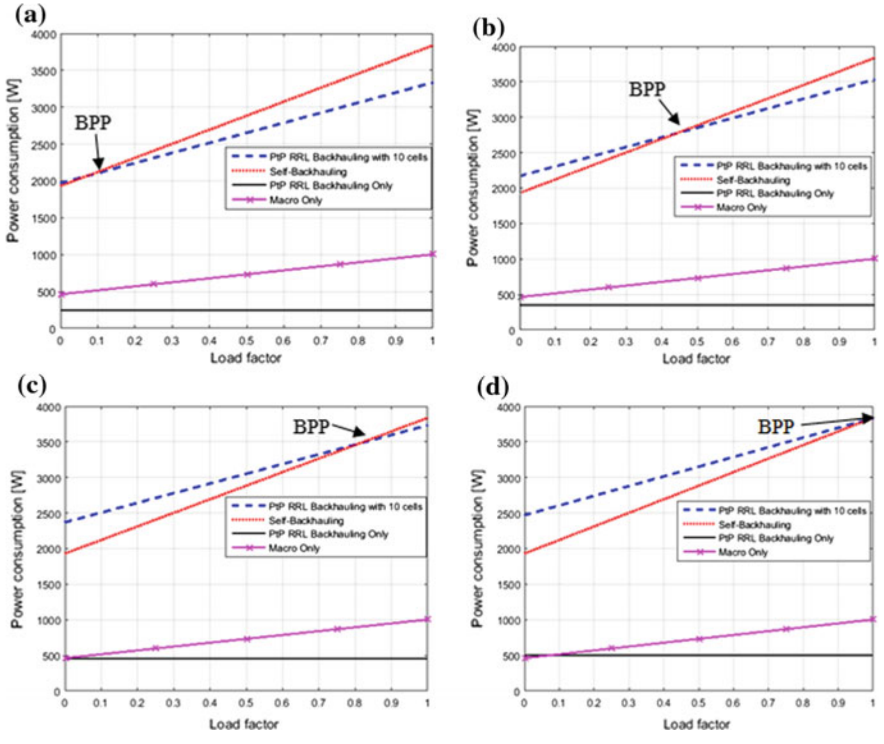


Fig. 7 Power consumption for fixed-wireless and self-backhaul networks with **a** 25 W RRL, **b** 35 W RRL, **c** 40 W RRL, and **d** 50 W RRL

backhaul links were deployed only. In this analysis, the power consumption of the traffic aggregating switch P^{c}_{swtich} at the hub was not included. This was considered to be negligible when compared to the HetNet power demand. Based on this analysis, the numerical result as shown in Fig. 7a indicated that backhauling the 10 SBS with 25 W microwave units is more energy-efficient solution than the self-backhauling at all traffic loads. However, in Fig. 7b, when the 25 W radio units are being replaced with 35 W radio units, at lower traffic conditions (i.e., f < 0.5), the self-backhauling becomes a better solution. Even though, it's rare to have 100% loaded BS and sites with less than 50% load may not be popular. This type of configuration may be favoured in a site with low traffics typically in rural area deployments. It is worthy to mention that at some traffic load, there exists a point named *break-even power point (BPP),* where one backhaul solution becomes more energy efficient than the other. In Fig. 7b, the point exist when the traffic load is about 45%, below this load, the self-backhaul solution is more preferable and above, otherwise. When high power radio units are used as backhaul (i.e., RRL = 40 W/50 W) in Fig. 7c and d, the BPP shift to almost 100% of the traffic load. This is an indication that the Self-B could be useful at low traffic condition or high traffics when the RRL power rating is above

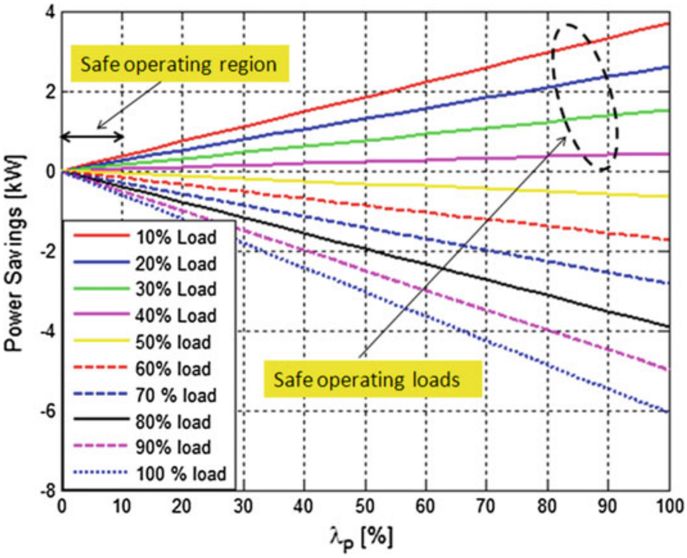


Fig. 8 Impact of load factor and self-backhaul factor on power saving. Network size, 1 MBS $p = 1$, 10 SBS, 35 W RRL

25 W. However, in view of the fact that the self-backhaul does not need radio link infrastructure as the case may be, but, it is clear that it is not an optimal solution at high load conditions, particularly when the radio links of 25 W and below are deployed. However, due to its simplicity and relatively, low loss of power efficiency still make it an attractive solution.

Figure 8 shows the impact of load factor and self-backhaul factor on power savings. The figure demonstrates how much power network operator would lose (or gain) if only self-backhauling solution is deployed. The percentage of the SBS that is self-backhauled is indicated by λ. The curves are presented as the difference from the network when all the SBS use RRL. The negative power saving is indicating that the network wastes power with respect to all RRL connections. We define the BPP as a point where the two backhaul options consume the same amount of power at a predefined traffic load and this is characterized in Fig. 9. When the curves reach 100% of load, it means self-backhaul solution would always be preferred for higher RRL power. For instance, we would always prefer self-backhaul if wireless links consume more than 17, 26, 35, and 55 W for 30, 20, 15, and 10 SBS per HetNet, respectively.

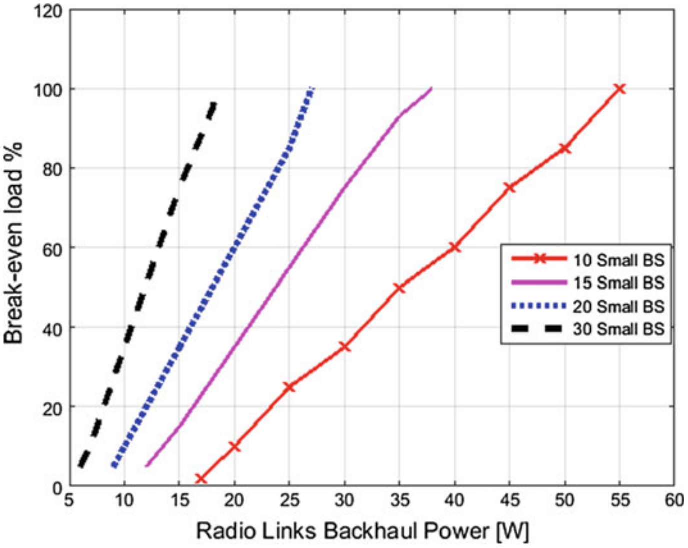


Fig. 9 Break-even load and power point as a function of SBS density [37]

6.3 *Impact of Macro Base Station Load on Power Consumption*

Power consumption largely depends on how traffic load is distributed on the network as illustrated in Fig. 7. The MBS serves as self-backhauling SBS but it also can serve UEs within its coverage area as illustrated in Fig. 4. In Fig. 10, the ratio of UE traffic served by MBS to UE traffic served by SBS is used as a parameter. The traffic loads for MBS and SBS are defined in Eq. (14) as f_a and f_b. We implement different load scenarios, whereby the MBS is fully loaded (i.e., 100%, $f_a = 1$), or partially loaded (i.e., 60%, $f_a = 0.6$) with the remaining 40% of the traffic loads routed to the SBS. The load-sharing effects on power consumption for the self-backhauled HetNet is observed. At low MBS: SBS load ratios of 40:60, even when the SBS is fully loaded with the remaining 60% (i.e., f_b= 100%), the self-backhauled network is more energy efficient than the conventional RRL.

6.4 *Energy Savings of Self-Backhauling*

Assume a HetNet system consisting of 20 MBS, each of them containing 10 SBS some of which are connected over the microwave links, while the rest are self-backhauled. In Eq. (14), λ represents the percentage of SBS that are self-backhauled. The negative power saving (i.e., $P_S < 0$) indicates that the self-backhauled network

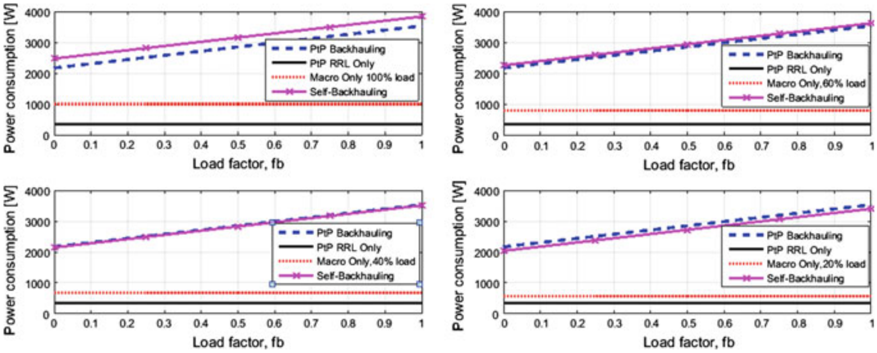


Fig. 10 Impact of macro base station load on power consumption

losses power with respect to microwave backhauled network and otherwise if $P_S \geq 0$ as indicated in Eq. (15). Figure 11 shows that when the network is fully loaded and RRL uses more than 50 W, then it is more preferable to use self-backhauling solutions. For lower loads, the self-backhauling is energy efficient and provides higher gains over the microwave backhauls. We also vary the percentage of SBS connected with self-backhauling, λ, and provide the curves for different traffic load. For lightly loaded network of 50%, the gain due to the use self-backhauling for all SBS (i.e., $\lambda = 100\%$) over 50 W RRL backhaul is about 5 k; this increases to 7.5 kW when the network load decreases to 30%.

Therefore, with the current deployed 100 W PtP microwave backhauls, we provide energy-saving prospects for deploying self-backhauling in emerging markets. Our analysis is focused on Asia and Africa. As an illustration, Nigeria, Ghana, Kenya, Tanzania, Uganda, Bangladesh, Indonesia, Pakistan, and India are projected to have at least 37,651, 7052, 7029, 5801, 3874, 41,686, 88,425, 40,422, 400,000 MBS sites, respectively, in 2015 [38–40]. With 8.1% CAGR, except India where 3.1% is used, in the next five years, about 13,762 new MBS sites will be added in Nigeria and 2578 in Ghana. Assuming 1:10 ratio of MSB-SBS replacement, the network densification projection will be 137, 620 and 25,780 SBS, respectively.

If all the SBS are self-backhauled (i.e., $\lambda = 100\%$) the resulting energy saving in the year 2020 will be 383.35 MWh/day (i.e., about 16 MW power) and 71.82 MWh/day (3 MW). This is expected to be higher for countries like India with over 400,000 MBS sites and which is expected to grow to 511,000 by 2020 [41] amounting to savings of 4.07 GWh/day (170 MW) and 32 GWh/day (1336 MW) in 2035. Figure 12 shows the energy-saving forecast for the selected countries for the period of 20 years, from 2015 to 2035. Already,e energy production in most of the developing countries is far below the demand. For example, Nigeria has a per capita consumption of 115 kWh/year [39]. Also, India, Tanzania, and Bangladesh have 565, 68.55, and 234 kWh/year, respectively [42], these are very insignificant when compared with some developed countries like the USA's 12,185 kWh/year, Finland's 16,100.44 kWh/year, and Denmark's 6026 kWh/year [43]. In Table 2,

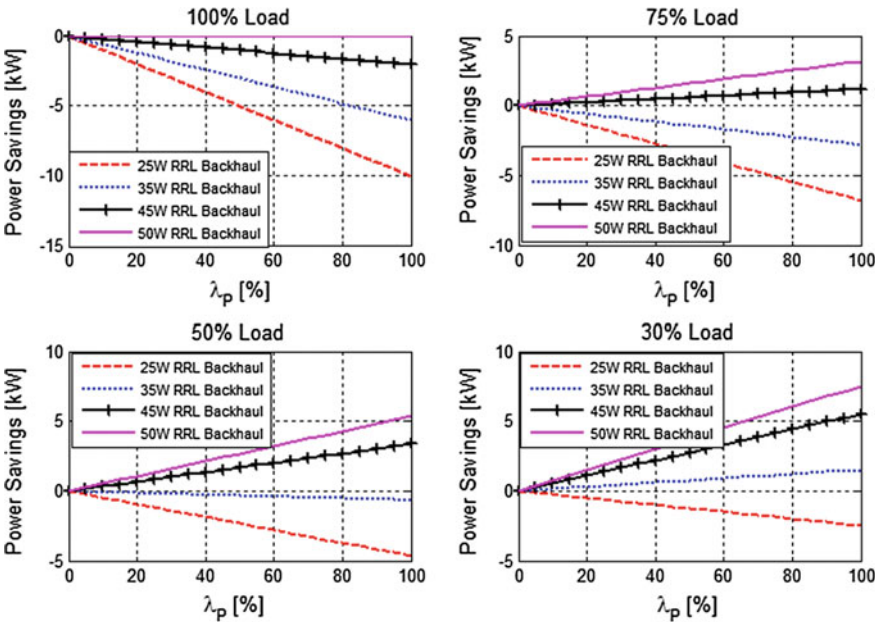


Fig. 11 Power savings for different RRL backhaul

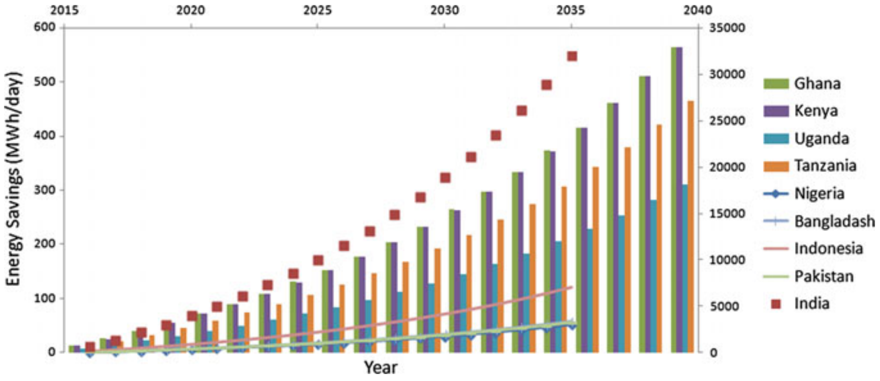


Fig. 12 Energy-saving forecast

we provide a summary of basic power indicators for some selected countries and prospective energy saving in 2035. Therefore, this energy-saving strategy would relieve the extra burden on the power network which will be more beneficial to the developing economies with a shortfall in energy production. While for the developed economies, it would yield a significant decrease in network energy costs.

Table 2 Electricity indicator indicies and energy savings

Country	Population (Thousand)	GDP per capita	Power consumption per capita (Kwh)	Land mass (Km^2)	No. of BTS	Electricity production (GWh)	Energy savings (MWh/day)
Nigeria	182,201.96	2640.3	115.04	923,768	37,651	3.3	3018
Ghana	27,409.89	1369.7	206.19	239,460	7052	3.5	565
Kenya	46,050.30	1376.7	136.64	582,650	7029	1.8	564
Tanzania	53,470.42	879.0	68.55	945,087	5801	1.7	465
Uganda	39,032.38	705.3	61.03	236,040	3874	1.5	311
Bangladesh	160,995.64	1211.7	233.88	130,170	41,686	7.349	3341
Indonesia	257,563.82	3346.5	623	1,811,570	88,425	232,000	7087
Pakistan	188,924.87	1434.7	357.34	770,880	40,422	7487	3239
India	1,311,050.53	1598.3	565.21	2,973,190	400,000	1,368,000	32,059

7 Conclusions

Network densification using a large number of small cells (small base stations) has been identified as the key enabling strategy for meeting up coverage and capacity demands in next-generation cellular systems. Providing effective backhauling to the SBS locations have also been identified as the main bottleneck, as the ever-increasing SBS density will lead to a more complex backhauling and, as such, increase risk of raising capital, and operational and network energy costs. This paper highlighted various backhauling options for SBS deployments and some of the key challenges the mobile network operators may be encountered during rollout. The paper also provides power system requirements for various network configuration and their respective power options that would be sustainable. The use of self-organizing backhaul networks (i.e., self-backhauling) for SBS is highlighted and compared in terms of energy efficiency with existing microwave backhauling solutions. Numerical results of the comparison show that when the network is fully loaded and the microwave backhaul uses more than 50 W, then it is more preferable to use self-backhauling solutions. For lower loads, the self-backhauling is energy efficient and provides higher gains over the microwave backhauls. Furthermore, a break-even power point and load threshold level for safe operation regions are presented, and this will help toward achieving optimum utilization of self-backhaul deployment and green backhauling for the deployment of HetNets in a way that is more energy efficient and sustainable compared to traditional homogeneous macro network deployments. The prospect of the self-backhauling in terms of energy saving is presented by providing energy-saving forecast for some selected emerging economies for the period of 20 years, from 2015 to 2035. These energy savings would relieve the overburden on the power network which will be more beneficial to the developing economies with shortfall in energy production. While for the developed economies, it would yield significant a decrease in network energy costs.

References

1. Cisco. Visual Networking index. Available on http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html [Accessed 17 Sept 2015]
2. Mills MP (2013) An overview of the electricity used by the global Digital Ecosystem. Digital Power Group, August 2013, pp 1–45. Available on http://www.tech-pundit.com/wp-content/uploads/2013/07/Cloud_Begins_With_Coal.pdf?c761ac&c761ac [Accessed on 27 Nov 2015]
3. Hwang I, Song B, Soliman SS (2013) A holistic view on hyper-dense heterogeneous and small cell networks. IEEE Commun Mag 51(6):20–27
4. Nokia Networks. Technology Vision 2020 Flatten Network Energy Consumption. White Paper, available on http://networks.nokia.com/sites/default/files/document/technology_vision_2020_energy_consumption_white_paper.pdf. [Accessed on 10 Nov 2015]
5. Lubritto C (2010) Telecommunication power system: energy saving, renewable sources and environmental monitoring. In: Bouras CJ (ed) Trends in telecommunications technologies. ISBN: 978-953-307-072-8, InTech, Available from http://www.intechopen.com/books/trends-in-telecommunicationstechnologies/telecommunication-power-system-energy-saving-renewable-sources-and-environmentalmonitoring
6. Bhushan N et al (2014, February) Network densification: the dominant theme for wireless evolution into 5G. IEEE Commun Mag 82–89
7. Bhushan N et al (2013, March) Small cells backhaul. Strategic White Paper, Alcatel-Lucent, pp 1–8. http://www.stjohnpatrick.com/12/ec/7Small-Cells-Backhaul.pdf
8. Tombaz S, Monti P, Wang K, Vastberg A, Forzati M, Zander J (2011) Impact of backhauling power consumption on the deployment of heterogeneous mobile networks. In 2011 IEEE Global Telecommunications Conference (GLOBECOM 2011), 5–9 Dec 2011, pp 1–5. https://doi.org/10.1109/glocom.2011.6133999
9. Luis S, Abbes BM, Ali MM, Michel M, Loutfi N (2015) Energy efficiency and cost issues in backhaul architectures for high data-rate green mobile heterogeneous networks. In 2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 30 Aug 2015–2 Sept 2015, pp 1563–1568. https://doi.org/10.1109/pimrc.2015.7343547
10. Farias FS, Monti P, Vastberg A, Nilson M, Costa JCWA, Wosinska L (2013) Green backhauling for heterogeneous mobile access networks: What are the challenges? In 2013 9th International Conference on Information, Communications and Signal Processing (ICICS), 10–13 Dec 2013, pp 1–5. https://doi.org/10.1109/icics.2013.6782868
11. Nokia. Self-organizing networks for mobile backhaul. Available on http://resources.alcatel-lucent.com/asset/200255
12. Spagnuolo A, Petraglia A, Vetromile C, Formosi R, Lubritto C (2015) Monitoring and optimization of energy consumption of base transceiver stations. Energy 81:286–293
13. GPP Release 10 available on http://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced [Accessed on 24 Sept 2015]
14. Lanzani CF, Kardaras G, Boppana D. Remote radio heads and the evolution towards 4G networks. Radiocom 1–5. Availbale on https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/wp/wp-01096-rrh-4g.pdf
15. DeruyckM, Emmeric T, Wout J, Luc M (2011) Modelling and optimization of power consumption in wireless access network. Comput Commun 43:2036–2046 (Elsevier)
16. Auer G et al (2011) How much energy is needed to run a wireless network? IEEE Wirel Commun 18(5):40–49

17. Deruyck M, Vereecken W, Joseph W, Lannoo B, Pickavet M, Martens L (2012) Reducing the power consumption in wireless access networks: overview and recommendations. Prog Electromagnet Res 132:255–274
18. Spagnuolo A, Petraglia A, Vetromile C, Formosi R, Lubritto C (2015) Heat flows and energetic behavior of a telecommunication radio base station. Energy 89:75–83
19. Pidwirny M (2006) Climate classification and climatic regions of the world. Fundam Phys Geogr (2nd Ed). Accessed 13 June 2016 from http://www.physicalgeography.net/fundamentals/7v.html
20. Holtkamp H, Auer G, Giannini V, Haas H (2013) A parameterized base station power model. IEEE Commun Lett 17(11):2033–2035
21. Balshe W. Power system consideration for cell tower applications. White paper, Cummins Power Gener 1–2, available on http://power.cummins.com/sites/default/files/literature/technicalpapers/PT-9019-Cell-Tower-Applications-en.pdf [Accessed on 10 June 2016]
22. Small Cell Forum (2014, February) Small cells, what's the big idea. White Paper, Issue 25, version 030.05.05, pp 1–27
23. Small Cell Forum (2013, February) Backhaul technologies for small cells: use cases, requirements and solutions, version 049.05.02, pp 1–78
24. C-RAN The Road Towards Green RAN. White paper, China Mobile, October 2011. Available on http://labs.chinamobile.com/cran/wp-content/uploads/CRAN_white_paper_v2_5_EN.pdf
25. Niu Yong, Li Yong, Jin Depeng, Li Su, Vasilakos AV (2015) A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges. Wirel Netw 21:2657–2676. https://doi.org/10.1007/s11276-015-0942-z
26. Fakidis J, Kucera S, Claussen H, Haas H (2015) On the design of a free space optical link for small cell backhaul communication and power supply. In IEEE ICC Workshop on Visible Light Communication and Networking (VLCN), pp 1428–1433
27. Opawoye I, Faruk N, Bello OW, Olufemi M (2015, July) Recent trends on TV white space deployments in Africa. Niger J Technol (NIJOTECH) 34(3):556–563 (University of Nnsuka)
28. Gerami, C, Mandayam N, Greenstein L (2010) Backhauling in TV white spaces. In 2010 IEEE on Global Telecommunications Conference (GLOBECOM 2010), 6–10 December 2010, pp 1–6. https://doi.org/10.1109/glocom.2010.5684131
29. Flickenger R, Okay S, Pietrosemoli E, Zennaro M, Fonda C (2008) Very long distance wi-fi networks. In NSDR'08 Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions. ACM New York, NY, USA ©2008, pp 1–6. https://doi.org/10.1145/1397705.1397707
30. http://www.theruckusroom.net/2012/01/wi-fi-for-cellular-backhaul-really.html
31. Qualcomm. Rising to meet the 1000x mobile data challenge. White paper. Available on https://www.qualcomm.com/documents/rising-meet-1000x-mobile-data-challenge [Accessed 24 Sept 2015]
32. Haile BB, Mutafungwa E, Hamalainen J (2015) LTE-advanced enhancements for self-backhauled LTE-U small cells: an Addis Ababa case study. In AFRICON, 2015, pp 1–5. https://doi.org/10.1109/afrcon.2015.7332025
33. Haile BB, Mutafungwa E, Hämäläinen J (2015) Coordinated multi-point transmission for relaxation of self-backhauling bottlenecks in heterogeneous networks. EURASIP J Wirel Commun Network 78:1–17
34. Coldrey M et al (2014, November 14) Wireless backhaul in future heterogeneous networks. ISSN 0014-0171284 23-3234. Ericsson, pp 1–11
35. Faruk N, Ayeni AA, Muhammad MY, Olawoyin LA, Abdulkarim A, Agbakoba J, Olufemi MO (2013) Techniques for minimizing power consumption of base transceiver station in mobile cellular systems. IJS Int J Sustain 2(1):1–11. http://dx.doi.org/10.4156/ijs.vol2.issue1.1
36. GSMA (2014, December) Green power for mobile: the global telecom tower ESCO market overview of the global market for energy to telecom towers in off-grid and bad-grid areas. Technical White Paper, pp 1–52. Available on http://www.gsma.com/mobilefordevelopment/wp-content/uploads/2015/01/140617-GSMA-report-draft-vF-KR-v7.pdf [Accessed on 18 June 2016]

37. Faruk N, Ruttik K, Mutafungwa E, Jäntti R (2016, November 15) Energy savings through self-backhauling for future heterogeneous networks. Energy 115 (Part 1):711–721, ISSN 0360-5442 (Elsevier), http://dx.doi.org/10.1016/j.energy.2016.09.059, http://www.journals.elsevier.com/energy
38. GSMA (2012, December) Powering telecoms: East Africa market analysis sizing the potential for green telecoms in Kenya, Tanzania and Uganda. White Paper, pp 1–28
39. GSMA (2013) Powering telecoms: West Africa market analysis sizing the potential for green telecoms in Nigeria and Ghana. Technical White Paper, pp 1–29. Available on http://www.millennia2015.org/files/files/Zero_mothers_die/gpm_market_analysis_west_africa_.pdf [accessed on 25 June 2016]
40. GSMA (2014, December) Green power for mobile: the global telecom tower ESCO market overview of the global market for energy to telecom towers in off-grid and bad-grid areas. Technical White Paper, pp 1–52. Available on http://www.gsma.com/mobilefordevelopment/wp-content/uploads/2015/01/140617-GSMA-report-draft-vF-KR-v7.pdf [Accessed on 18 June 2016]
41. Delloite (2015, June) Indian tower industry: the future is data. http://www2.deloitte.com/content/dam/Deloitte/in/Documents/technology-media-telecommunications/in-tmt-indian-tower-industry-noexp.pdf [Accessed on 08 July 2016]
42. https://yearbook.enerdata.net/world-electricity-production-map-graph-and-data.html
43. Country Comparison Electricity consumption per capita. Available on http://www.indexmundi.com/g/r.aspx?v=81000 [accessed on 30 June 2016]

Integrated Resource Allocation Model for Cloud and Fog Computing: Toward Energy-Efficient Infrastructure as a Service (IaaS)

Mohammed Joda Usman, Abdul Samad Ismail, Hassan Chizari, Abdulsalam Ya'u Gital, Haruna Chiroma, Mohammed Abdullahi and Ahmed Aliyu

Abstract Cloud is transmigrating to network edge where they are seen as virtualized resources called "Fog Computing" that expand the idea of Cloud Computing perspective to the network edge. This chapter proposes an integrated resource allocation model for energy-efficient Infrastructure as a Service (IaaS) that extends from the network edge of the Fog to the Cloud datacenter. We first developed a new architecture and introduced a policy on the Fog end where a decision will be made to either process the user request on the Fog or it will be moved to the Cloud datacenter. We developed a decision model on top of the architecture. The decision model takes into consideration of the resource constraints of CPU, Memory, and Storage. Using this will improve resource utilization as well as the reduction in energy consumption by

M. J. Usman (✉) · A. Aliyu
Department of Mathematical Sciences, Bauchi State University Gadau, 75110 Itas-Gadau, Nigeria
e-mail: umjoda@gmail.com

A. Aliyu
e-mail: ahmedaliyu8513@gmail.com

A. S. Ismail
School of Computing, Universiti Teknologi Malaysia, Skudai Johor 81310, Malaysia
e-mail: absamad@utm.my

H. Chizari
School of Computing and Technology, University of Gloucestershire, Cheltenham Park Campus GL50 4SH, UK
e-mail: h.chizari@imperial.ac.uk; chizari@utm.my

A. Y. Gital
Department of Maths and Computer, Abubakar Tafawa Balewa University Bauchi, Bauchi 740272, Nigeria

H. Chiroma
Department of Maths and Computer Science, Federal College of Education Technical Gombe, Gombe 072158, Gombe, Nigeria
e-mail: freedonchi@yahoo.com

M. Abdullahi
Department of Computer Science, Ahmadu Bello University Zaria, Zaria 810211, Kaduna, Nigeria
e-mail: abdullahilwafu@abu.edu.ng

T. Herawan et al. (eds.), *Advances on Computational Intelligence in Energy*,
Green Energy and Technology, https://doi.org/10.1007/978-3-319-69889-2_7

a datacenter. Finally, we addressed future research direction considering the model components and its performance.

Keywords Virtual machine · Cloud datacenter · Resource allocation · Energy efficiency · Fog Computing

1 Introduction

Cloud Computing is a universal term used for the delivery of hosted services and resources connected with the Internet that are pre-packaged using virtualization technology. The technology allows the creation of duplicate servers or computer system known as a virtual machine (VM). The concept of utilizing Cloud Computing allows companies, industries, and organization to have their Computing resources as a utility that is on a pay-per-use basis instead of building and keeping computing infrastructures in your premises. More so, Cloud Computing has rapidly become known as a successful paradigm for providing ICT infrastructure support. There is an enormous growth in this area for the past few years. The broader acceptance of Cloud Computing domain contributes to the development of large-scale datacenters. The datacenters provide services through their respective deployment and service models presented in various means and at different stages of the datacenters as shown in Fig. 1. The services are IaaS, Software as a Service (SaaS), and Platform as a Service (PaaS) [1, 2]. There are a lot of benefits using Cloud which includes pay-per-use, instant on-demand self-service provisioning, rapid elasticity, and resource pooling [3].

Cloud datacenters are the strength of today's demanding ICT infrastructure. Therefore, energy-efficient resource utilization is growing gradually and becoming important [4, 5]. Furthermore, the Cloud is transmigrating to the network edge where they are seen as virtualized resources called "Fog Computing" that expand the idea of Cloud Computing perspective to the network edge. The Fog and Cloud used similar resources that are identical in characteristics such as networking and computing devices with storage. They also share the same techniques associated with energy-efficient resource allocation and utilization. Although Fog Computing relies on routers and edge devices that are usually away from the hub datacenter. On the other hand, there are applications such as mobility-based services, health-monitoring devices, sensor networks with actuators, and emergency response services that require real-time response. If Cloud Computing alone will be used, it will require many resources from the sources to the edge and round it back to the Cloud datacenter. This leads to inefficient usage of the Cloud datacenter infrastructure, poor scheduling policies, and resource wastage are the reason for high-energy consumption in datacenters, not their size. Therefore, energy efficiency remains a serious problem for datacenter service providers.

Among the means to address this problem is to reduce the potentialities of the Cloud datacenters due to the used of virtualization technology [2]. The virtualization

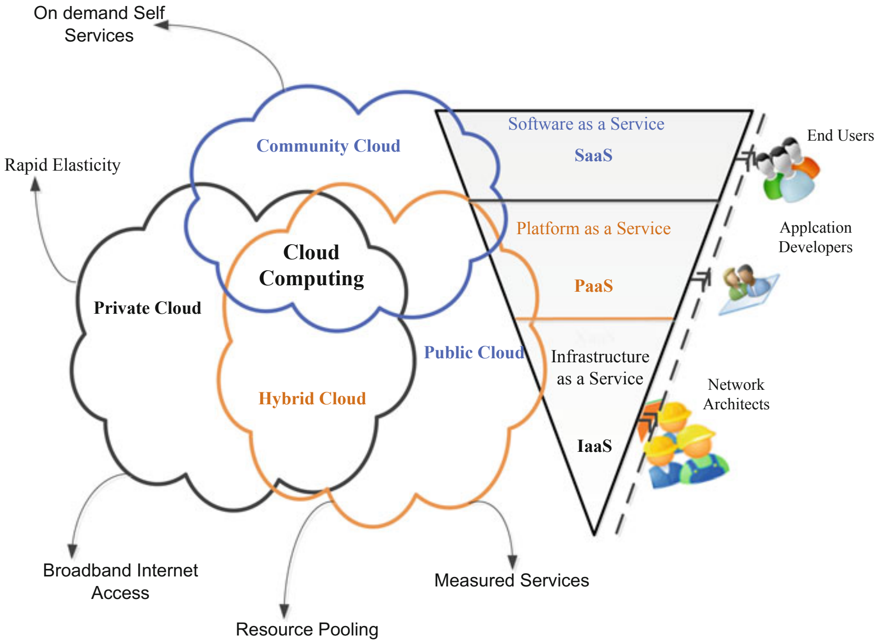


Fig. 1 Classification and models of Cloud Computing

allows Cloud administrators to create many VM instances on a physical machine (PM) or server. Thereby demanding few physical resources unlike in the past, thus the resource utilization is improved. Alternatively, reducing energy consumption will be realized by turning off or switching PMs that are in the idle state to low-power mode state (i.e., sleep, hibernation). This approach also reduces the idle power consumption in the datacenters. This method enhances the condition of power consumption in the datacenter, but it is often not sufficient, due to the dynamic and heterogeneous nature of the Cloud environment. Yi et al. [6] in their survey they have find out unsolved Cloud Computing challenges because of their underlying problems such as resource utilization, central access, inadequate mobility support, and location awareness. Therefore, Fog Computing can address these challenges if properly apply in the Cloud domain. Combining the Fog with Cloud Computing is one of the most logical steps for building an adaptable and scalable platform that strengthens the IoT [7]. This will continue to support and increase the inescapable interplay of the Fog and the Cloud domain in years to come. Therefore, there is a need for integrating the Cloud and Fog Computing architecture for realizing energy efficiency in Cloud datacenter IaaS.

The focus of this research work is how to realize energy efficiency using resource allocation strategies and policies that can be applied to virtualized Cloud datacenter. Unlike existing studies [8–13], our approach will integrate Cloud IaaS with Fog Computing resources on a single datacenter. The integration will bring a new form of applications and services within the Cloud datacenter environment that reduces resource under-utilization and energy consumption. We also develop a decision model that takes into consideration of the resource constraints and their capacity at the Fog edge. Furthermore, the model determines whether the user's request will be executed on the computational infrastructure of Fog or Cloud datacenter. Also, we recommend future research direction considering the operational components, execution environment, and performance of the integrated environment.

The rest of the book chapter is organized as follows. Section 2 discusses the related work. Section 3 presents the Cloud and Fog Computing overview. Section 4 describes the needs for Cloud and Fog Computing with the proposed model. Section 5 outlines the future issues and direction of the research and Section 6 concludes the book chapter.

2 Related Works

Deng et al. [14] and Al Faruque and Vatanparvar [15] presented their work with resource provisioning of Fog and its related services. They focus mainly on the energy consumption of Fog–Cloud interaction of devices installed at smart houses or cities. Furthermore, they consider the different stages of energy consumption as well as the latency trade-off between the Cloud–Fog. Deng et al. [14] and Tang et al. [16] introduce a distributed architecture of Fog Computing that has big data analysis capabilities used in smart cities. They designed a four-layered architecture that supports immediate response and intelligence report for future smart cities. The computing nodes at each layer of the architecture execute applications that are latency-sensitive and support quick control loop to protect the safeness of some vital infrastructure resource components. They focus on the integration of Cloud IaaS with next-generation smart cities. It provides high computing performance that supports future intelligence smart cities. However, the authors consider response time only. Thus, leaving abundant resources under-utilized. Stojmenovic [17] expands the concept of Fog and Cloud Computing to the decentralized smart controller in survey paper. They recognize cloudlets often known as task as defined in Fog domain. The authors describe the cloudlets using scenarios in software-defined networks (SDN). They make a recommendation on the existing approach, which needs modification to adopt two layers Fog–Cloud Computing. Smart Cargo concept has been introduced by Costa et al. [18]. The authors integrate traditional technologies and frameworks ascertain from the cargo environment. Besides, the concept combined the capabilities of a Fog Computing centralization and scalability offered by Cloud paradigm applied to the freight transport domain. They explore home energy management (HEM) and a micro-grid-level energy management over a Fog Computing platform with a customized controller. This method shows a paradigm shift that takes away computing application and data services control from Cloud to Fog Computing. Therefore, the approach makes multimodal freight transport attained real-time awareness with future environmental prediction, but they did not consider the limitation of the Fog devices and/or nodes.

Fog Computing has been used as a new platform for energy management by Vatanparvar et al. [19] and Al Faruque and Vatanparvar [15]. The platform allows the interaction of different devices' interoperability and operational flexibility. Furthermore, they implemented two prototypes of HEM and micro-grid-level energy management that separates control and data communication layers. The platform is adaptive and scalable due to its open architecture. However, they focus on energy management of residential houses, not datacenters. Masip-Bruin et al. [20] proposed a layered architecture that shows the need for the coordinated management of Fog–Cloud systems. The authors use medical emergency scenario to evaluate the performance of the architecture. The F2C (Fog 2 Cloud) architecture optimizes and improves the execution time of applications compared with traditional Cloud services.

Madsen et al. [21] present the challenges and reliability posed by current Computing paradigms for reliable Fog platforms. The approach incorporates the networks of smart devices communicating with the Cloud. While it may be true, the authors conclude that the design and development of Fog Computing projects are demanding. Therefore, this strategy did not offer a new approach that is reliable for the network of smart devices in Fog Computing standard. Thus, the acceptance of Fog-based access call for the implementation of algorithms that deals with network reliability of smart devices. The authors focus on Fog devices that are operating under specific circumstances that inquire for fault tolerant technique, not resource utilization or energy management. Similarly, Bonomi et al. [22] analyze key distinct features of Fog and how it complements Cloud Computing. Main goal of this research work is to examine the used instances that motivate the need for Fog. This approach emphasizes the relevancy of Fog to IoT and Big Data Space (BDS) paradigm. Finally, the authors provide a high-level characterization of Fog's software architecture, giving prominence to the different components and the technological requirement to achieve the Fog vision.

A placement resource migration technique for Cloud and Fog is presented by Ottenwälder et al. [23]. The authors demonstrate how application understanding of complex event processing (CEP) system is applied to make live VM migration better. The needed bandwidth for the technique to perform the VM migration has been reduced substantially. The resources that are considered to be network-intensive are directed on distributed Fog nodes. On the other hand, the computationally intensive resources were hosted in the datacenter. This method reduces the migration cost due to the direct selection of the destination of resources to be migrated which assures sufficient network utilization over a period of time. The network utilization and mobility pattern have been improved. Thus, other resources and workload mobility have not been optimized due to the computationally intensive task by the Fog devices. The immense demand for geo-distribution, mobility support, low latency, and location-based application is becoming high in both Fog and Cloud domains. Therefore, to solve the above-mentioned problem, a resource-provisioning mechanism for VM-based tactical cloudlets has been proposed by Lewis et al. [24]. It is a strategy that can be hosted on a vehicle or other platforms for providing IaaS that endorses computation offloading and data forwarding at the edges of the Fog devices. The aim is to increase position awareness that will improve the decision making of moving the resources used in Cloud close to the Fog. This will encourage energy efficiency in mobile devices, lower latency and change provisioning policy in tactical environment. Although the cloudlets has been successful in supporting the mobile devices, the approach did not consider the limitation of the mobile devices in terms of running applications.

Aazam and Huh [25] proposed a solution using smart gateway-based communication (SMBC) with Fog Computing to realize smart communication and resource management. It alleviates the core network communication overhead of the Cloud datacenter. Using this method makes it easier to create better services for the Cloud in an efficient manner. However, resource utilization and energy saving are not their focus. Similarly, Aazam and Huh [26] presented another service-oriented resource management model (SORMM) for Fog Computing. The model performs adequate resource management for IoT deployments. It also uses a framework for resource management to predict user's resource usage and to also pre-allocate resources based on the probabilistic behavior that determines the future usage by the user. The used of this prediction strategy by the consumed resources bring good standard and efficiency. However, they did not present any evaluation or simulation result that supports their model. Nishio et al. [27] presented a mathematical framework for service-oriented utility in heterogeneous resource environment. In mobile Cloud concept, the heterogeneous resources are measured in standards, such as power, bandwidth, and latency. Neighboring nodes within a local area network (LAN) form a cluster group termed local Cloud. The nodes shared the resources within the LAN and other nodes from the clusters in the same local Cloud. Then from there, the Fog device is selected from the nodes in each local Cloud by local resource coordinator. However, the model did not take into consideration of resource utilization and energy consumption.

Dsouza et al. [28] proposed a relevant schema with a preliminary policy management framework followed by a detail description of the policy. The authors identified Fog Computing novel features. These key features have been identified with challenges that are inherent in managing the policies. It becomes vital to endorse ensure resource sharing and collaboration to reuse data within heterogeneous Fog environment. The model is effective in resource management for enterprise used but not Cloud datacenter. Recent development of Fog Computing is presented by Dastjerdi et al. [29], and the authors design a reference architecture for Fog and discussed their challenges. They pointed out issues that need urgent attention including security, resource management, and energy usage minimization are still in needs of the solution.

3 Overview of Cloud–Fog Computing

The Cloud Computing is the Cloud datacenter structural system, which is a building or room full of computing servers on premises or remote location. It also includes other related equipment and resources. Whereas the Fog Computing is being considered as a term that is originally coined by Cisco, that is, in many ways the same or nearly the same with edge Computing or nano-Cloud. The main distinct feature of the Fog when compare with the Cloud is that the platform and resources used by Fog are always at the network edge of an enterprise or institution with limited computational capabilities. The attribute of the Fog and other associated components and platforms are discussed in the following sections.

3.1 The Cloud Computing Architecture

The Cloud Computing architecture is the structure and organization of the system, which comprise on premises and computing resources (physical or logical) such as available services, middleware, software components, storages, geo-location, the externally visible properties of resources and their relationships to provide services to users through the network [30]. Recently, Cloud Computing reference architecture recognizes the major actors and classifies them based on types, models, as well as their behavior. More so, Cloud architecture has been considered as a generic high level that is proposed to help the understanding of the user demands, characteristics, and principles used in this domain.

This architecture comprises two parts: the front end and the back end. The front cannot operate independently without the support of the back and vice versa. All requests coming into the Cloud mostly come from the front end. This connects usually with the back end through a network communication system. What differentiates the front with the back end is the source of the request. The request always comes from the users who are usually known as client or consumers of services from the front end. The back end is where the requested services by the clients are hosted which is the Cloud side of the system. The users do not care where the request or information is stored or processes rather they are after service delivery and reliability from the Cloud side. Therefore, the front end is composed of user's resources such as computing and network resources with the required application to access the Cloud Computing resources. The Cloud Computing resources usually include physical machines (PM) also known as servers, storages, network communications, and different applications and operating systems running concurrently to service user's request. Requests of application in the Cloud are usually run on virtual machines (VM) hosted on the PM. These resources are managed by the administrator who manages and monitors the resource as well as the traffic within the Cloud environment. The management of the resources in the Cloud has to follow a set of rules known as a protocol with the special help of middleware. The middleware will allow the computers to communicate with each other over the network [30].

3.2 *The Fog Computing Architecture*

The Fog Computing system-level architecture has been modified to achieve maximum efficiency for the scattered Computing nodes used to perform the computational task with storages, processes, and networking capabilities according to various criteria within an Internet of Everything (IoE) domain [31]. The Fog Computing encompasses a tiered organization of Fog nodes all over the network amid sensors and the Cloud at the central core of the network system [32]. The Fog architecture constitutes a change from conventional access systems and trust on Cloud models, to the latest computation model that moves computational request close to the network edge often referred to edge Computing [33]. The nodes are not completely fixed to the edge but should be seen as a variable within the system of connectivity.

The Fog architecture provides the correct support and attempts to bring into the balance between resource content of the three most basic capabilities at the edge of the network level where they are optimally located. The Fog layers as shown in the figure above are located between different network levels of the Cloud and the Fog resources known as nodes. As a result, the Fog makes a connected graph, with the corresponding Fog nodes located at different levels, which forms an interconnected topology between core smart objects and layers Cloud components [32–35]. Storage and network bandwidth are the essential attributes of Fog architecture found at different levels.

4 The Need for Integrating Cloud and Fog Computing

There is always a need to utilize the Computing power available at the edge of the network which is the Fog. This can only be realized by taking Cloud concept which includes flexibility, agility, and distributed nature as well as network storages and moving them toward the devices that generate the data. Integrating Fog with Cloud Computing is more convenient than migration. The Cloud has been standardized in terms of information exchange and emerging use of services such as web, machine-to-machine communication (M2M) and used open sources. As the Fog devices are becoming smarter and intelligence that serve as the extension of the Cloud where user's requests are no longer centralized. Therefore, bringing in new concept of decentralization due to the massive computation at the edges of the network. The integrated architecture has been designed based on the Cloud and Fog characteristics as shown in Fig. 2.

This clearly shown the limitation of each of the architecture in terms of latency, hardware used, location awareness, and response time. Moreover, this has also lead to the need for their complimentary services toward energy efficiency in the Cloud datacenter. The integration of the Fog–Cloud architecture take into consideration the limitation of both Cloud and Fog Computing, thereby providing a new solution that improves energy efficiency, resource utilization, and data optimization.

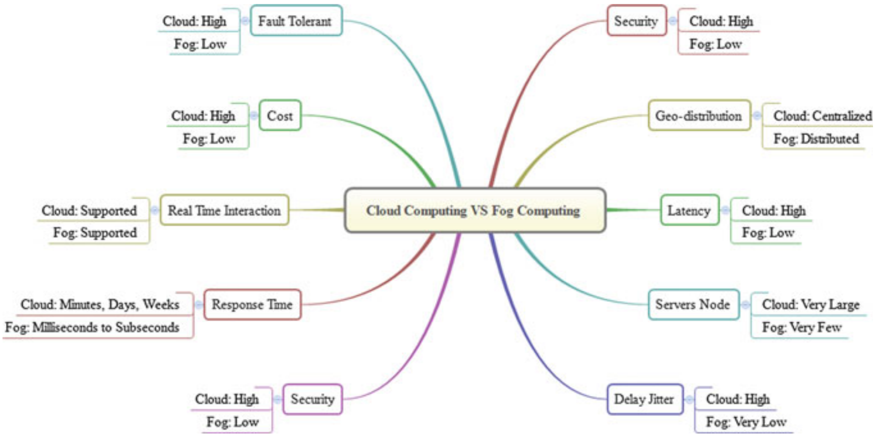


Fig. 2 Challenges of Cloud and Fog Computing architecture

4.1 Operational Framework of Cloud–Fog Computing

Figure 3 shows the framework for the integrated model of Cloud–Fog Computing. The framework is divided into four distinct parts with the objective of achieving energy efficiency in the Cloud datacenter. The users are the consumers of the services and always make application request. These requests are usually processed at the datacenter which consumed high energy due to inefficient resource utilization. Therefore, we have introduced a new concept whereby a user's request is process on the Fog using workload allocation model. This model has three decision criteria due to the inherent limitation of the Fog resources in terms of processing, storage, and memory else the request will be sent to the Cloud. The Cloud will now look at the request and allocate appropriate resource for it using resource allocation linear model. We also show how the Fog interacts with the Cloud through the gateway.

4.2 The Proposed System Architecture of Cloud–Fog Computing

Figure 4 shows the overall system component of the proposed architecture with a brief explanation of their role while Fig. 5 shows the integrated architecture of Fog/Cloud.

The proposed system architecture shows the integration of Cloud and Fog architecture. The datacenter resources such as server processor and network which enable on-demand and dynamic allocation of resources while minimizing the datacentre energy consumption. The approach is model-based using resource allocation linear model of the IaaS that is handled by the Cloud Management System (CMS) which include brokers, service monitors, and energy optimizers. While for the Fog is handle by the integrated resource management system (IRMS) which includes the sensor as well as resource constraints. The technique will overcome the limitations of both the Cloud and the Fog.

The Cloud Computing players include users, application request, service providers, and CMS and the other side of the architecture is connected with the communication gateway through the IRMS. The Fog physical resources are organized based on their application areas and execution environment. In Cloud datacenter setting, users are regarded as one of the important entities since they are the end users

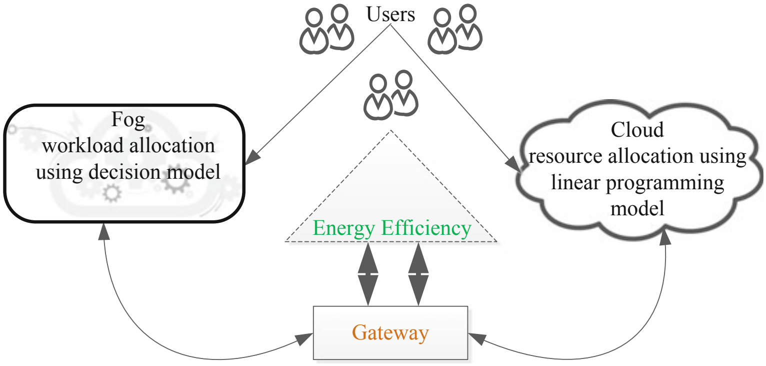


Fig. 3 Operational framework of energy efficient Cloud–Fog Computing

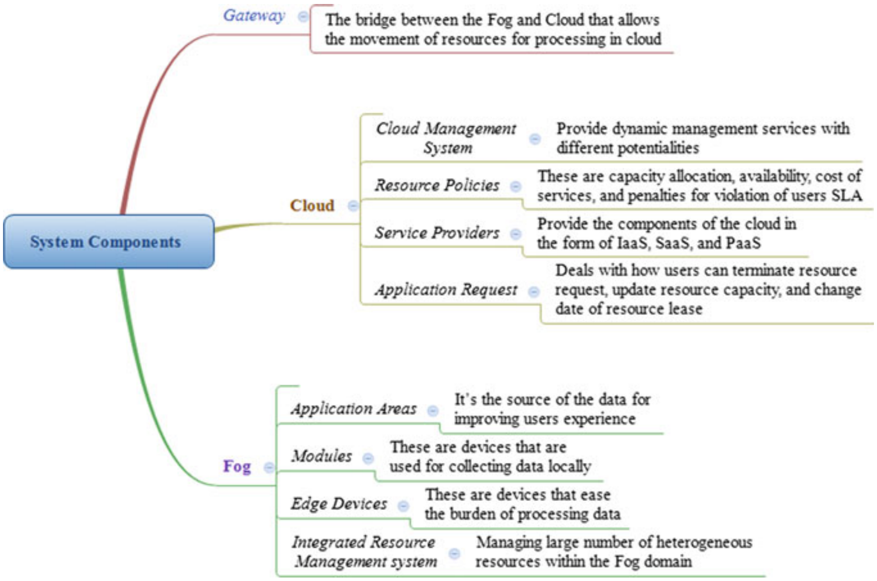

Gateway
The bridge between the Fog and Cloud that allows the movement of resources for processing in cloud
System Components
Cloud
Cloud Management System
Provide dynamic management services with different potentialities
Resource Policies
These are capacity allocation, availability, cost of services, and penalties for violation of users SLA
Service Providers
Provide the components of the cloud in the form of IaaS, SaaS, and PaaS
Application Request
Deals with how users can terminate resource request, update resource capacity, and change date of resource lease
Fog
Application Areas
It's the source of the data for improving users experience
Modules
These are devices that are used for collecting data locally
Edge Devices
These are devices that ease the burden of processing data
Integrated Resource Management system
Managing large number of heterogeneous resources within the Fog domain

Fig. 4 Components of the architecture

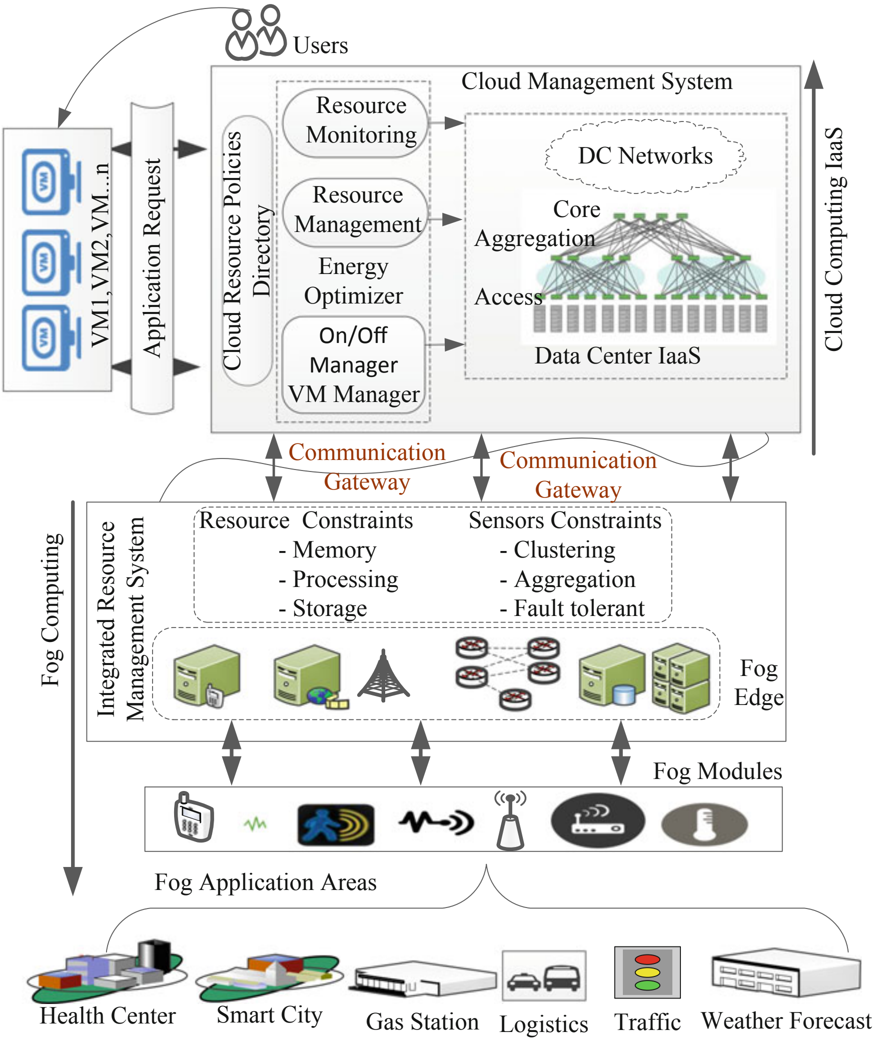


Fig. 5 Integrated architecture of Cloud and Fog Computing

of the services provided by the Cloud IaaS. They sent application requests that are operated in distinguish VMs hosted on different servers while the service providers owned the applications that provide services to the users.

More so, users submit their request to the Cloud provider through the CMS, specifying the number of VMs needed for each application and their requirement in terms of memory, storage, CPU, and network. Other functional components of the CMS collect run-time statistics of the resources including availability, utilization, and status of the resources. This information will be published for decision making by the data center based on their resource policy requirement define in the service level agreement (SLA).

On the other part of the architecture, we push away the applications, data, and services from distributed nodes to the network edge. Therefore, this will allow the IRMS to perform its function and decide where to place the applications and data based on the application requirement.

4.3 Assumption of the Cloud–Fog Model

The integration of Cloud–Fog Computing separates the workloads between the two components. This also leads to the efficient utilization of resources hence reduces the energy consumption. The detail assumptions of our architecture are as follows:

1. A request represents a user's computational request or task.
2. All requests come from Fog Computing environment considering three constraints such as CPU, storage, and memory.
3. The constraints are presented as $U(R_1)$, $U(R_2)$, and $U(R_3)$.
4. $U(R_i)$ is the cumulative resource requirement of the user request, Ø is the threshold of the fog resource capacity, where α, β, and γ represent resource requirement of a given task, where the task will be T_1, T_2, T_3, …, T_n.
5. Based on the assumption, the workload forwarded to the Cloud is in the form of tasks.
6. The resource allocation requests by users must provide the detail of resource usage for the task. Otherwise, if the user requests are more than the available resources, the request has to wait in the queue for the next allocation.
7. All hosts in Cloud–Fog consume energy in idle state and denoted as $P_{\min}$.
8. Hosts consume maximum energy at the highest stage of utilization which is denoted as $P_{\max}$.
9. Hosts put the request in the waiting queue if its CPU utilization is at pick level or it reaches its threshold.

4.4 Decision Model for Cloud–Fog Architecture

In optimization, the resource constraints usually affect the solution. Therefore, how to select and decide where the resource will be run is an essential element for realizing efficient resource utilization and energy consumption in optimization, the resource constraints usually affect the solution. Therefore, how to select and decide where the resource will be run is an essential element for realizing efficient resource utilization and energy consumption. For the Fog Computing resources, the assumption is that the number of constraints which includes CPU, storage, and memory are presented as $U(R_1)$, $U(R_2)$, and $U(R_3)$. $U(R_i)$ is the cumulative resource requirement of the user request, Ø is the threshold of the fog resource capacity, and where α, β, and γ represent resource requirement of a given task, where the task will be $T_1, T_2, T_3, \ldots, T_n$.

Thus, the resource utilization function for single resource can be denoted as:

$$U(R_1) = \sum_{i=1}^{n} \alpha_i \tag{1}$$

$$\text{Single Decision Creteria} \begin{cases} U(R_1) < \theta; & \text{assign to Fog resources} \\ \text{Otherwise}; & \text{assign to Cloud resources} \end{cases}$$

The decision criteria for single-resource constraint take into consideration the size of the resource α_i which is being requested by the user. It checks the requested resource threshold Ø capacity whether it can be processed at the fog level else it will forward the request to the cloud.

For double-resource constraint's request, assuming we have a task that requires more than one resource at a time for execution. Thus, the resource utilization function can be denoted as:

$$U(R_1) = \sum_{i=1}^{n} \alpha_i \tag{2}$$

$$U(R_2) = \sum_{i=1}^{n} \beta_i \tag{3}$$

$$\text{Double Decision Creteria} \begin{cases} U(R_1) < \text{Ø}_1; \text{ and } U(R_2) < \text{Ø}_2; & \text{assign to Fog resources} \\ \text{Otherwise}; & \text{assign to Cloud resources} \end{cases}$$

The decision criteria for double-resource constraint also take into consideration the size of the resource α_i and β_i which is being requested by the user. It checks the requested resource threshold Ø capacity whether it can be processed at the fog level else it will forward the request to the cloud. Whereas the multiple resource constraints request, assuming we have a task that requires more than two resource at a time for execution. Thus, the resource utilization function can be denoted as:

$$U(R_1) = \sum_{i=1}^{n} \alpha_i \tag{4}$$

$$U(R_2) = \sum_{i=1}^{n} \beta_i \tag{5}$$

$$U(R_3) = \sum_{i=1}^{n} \gamma_i \tag{6}$$

$$\text{Multiple Decision Creteria}\begin{cases} U(R_1) < \emptyset_1;\ U(R_2) < \emptyset_2;\ \text{and } U(R_3) < \emptyset_3; \\ \quad \text{assign to Fog resources} \\ \text{Otherwise;} \quad \text{assign to Cloud resources} \end{cases}$$

The decision criteria for multiple resource constraints take into consideration the size of the resource α_i, β_i, and γ_i which are being requested by the user. It checks the requested resource threshold ∅ capacity whether it can be processed at the fog level else it will forward the request to the cloud. The Fog Computing resource capacity is given as

$$\begin{cases} 0 \le \theta \le f_i; \quad \text{allocate to fog } i \\ \text{otherwise;} \quad \text{allocate to cloud} \end{cases} \tag{7}$$

N is the number of available fog devices i
f_i resource capacity in terms of RAM, CPU, and storage.

4.5 Optimization Problem

Resource allocation and energy efficiency of the overall Cloud datacenter can be denoted as linear programming (LP) problem to reduce the energy consumed E, as represented in Eq. (8).

$$\text{Minimize } E = \sum_{r=1}^{n} \sum_{i=1}^{r} E_i(\tau) \tag{8}$$

Subject to

$$E_i(\tau) = \left(P\text{max} - P\text{min}\right) * \frac{U_i(\tau)}{100} + P\text{min} \tag{9}$$

$$U_i(\tau) = \sum_{j=1}^{n} U_{ij} \le \text{Peakload at time } \tau, \quad \forall R_i \in R \text{ and } \forall t_j \in T \tag{10}$$

$$U_{ij} = 0, \quad \text{when the user request } j \text{ is not serve on node } R_i$$
$$U_{ij} = 1, \quad \text{when the user request } j \text{ is serve on node } R_i$$

Equations (8) and (9) indicate that minimization of energy consumption is proportionally subjected to the utilization of resources based on the user request for the time τ. Our models relate to the overall Cloud Computing environment's energy consumption to total utilization represented by U as can be formulated in Eq. (8). In the equation, u is the percentage value of the processor utilization, E(u) is the energy consumed by CPU at the utilization $u\%$, and $P_{\max} - P_{\min}$ are the power consumption at maximum performance in watt and at idle, respectively.

4.6 Performance Evaluation

Fog does not replace the Cloud which have lots of advantages because of its central and scalable features. The Cloud datacenters offer unlimited virtual processing capabilities and an on-demand resource usage model. To explain our contribution, the storage, processing, and memory constraints have been considered in the decision criteria model. Therefore, we use the idea of resource modeling criteria as the key idea for the integration of Cloud and Fog to reduce energy consumption and maximize resource utilization of IaaS. To ensure the quality of our developed model, we cautiously perform some experiments based on assumption. From the assumption, all requests are coming from the Fog environment handle by IRMG. Therefore, the decision models will now decide where the request will be processed based on the threshold limit set by the model. In the following, we show some results obtained by using the decision model which includes single-, double- and multiple decision criteria.

From Fig. 6, it shows the uses of single-resource request from fog environment. When the requested resource by the user has not exceeded the fog capacity as define in the model, we can say that it has an upper and lower boundary of the requested resource. If resource happens to be at the lower boundary (less than θ). Then, the request will be handled by the Fog schedulers. Likewise, when the request is greater than θ then the resource request will be forwarded to the cloud datacenter through the gateway. In the datacenters, the CMS will handle the request and allocate an appropriate resource for the user request. The request will be executed based on first-come first-serve. As shown in Fig. 6 anything above the θ will be processed in the cloud datacenter. This will improve resource utilization of the Fog and Cloud, as well as the energy efficiency of the datacenters.

Figure 7 shows the user request for double resource. In this case, the decision model will make a complex decision when the requested resources are not same. This is due to the nature of the application. Some applications require much memory than storage and vice versa. In this case, the decision criteria will not compare the two requests and in case of any of the request exceed Fog capacity θ, then the request will be moved to the cloud datacenter. However, when the request has not been exceeded θ, then it will be processed within the Fog environment. In addition, Figs. 8 and 9 show multiple decision criteria. In this case, the user's request is treated in the same way as single- and double-decision criteria. However, we have shown how the multiple resource requests will be processed in Fog. This can only be achieved when the user request has not exceeded the threshold for the three constraints. In case of any of the requests have been exceeded the threshold, the resource request will be moved to the Cloud datacenter.

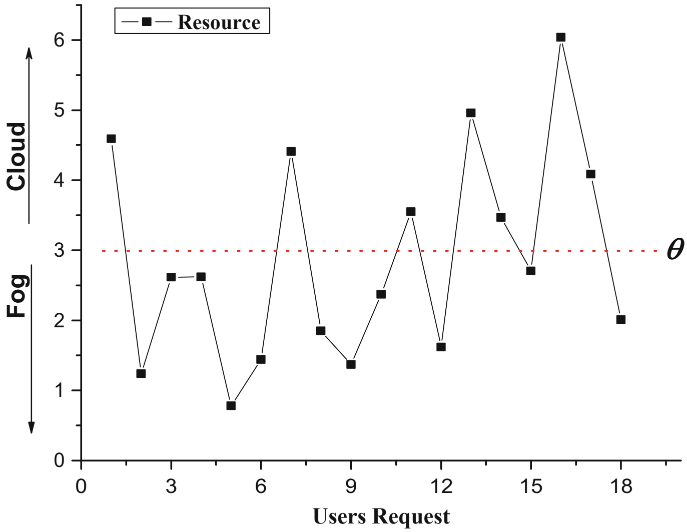


Fig. 6 Single-decision criteria

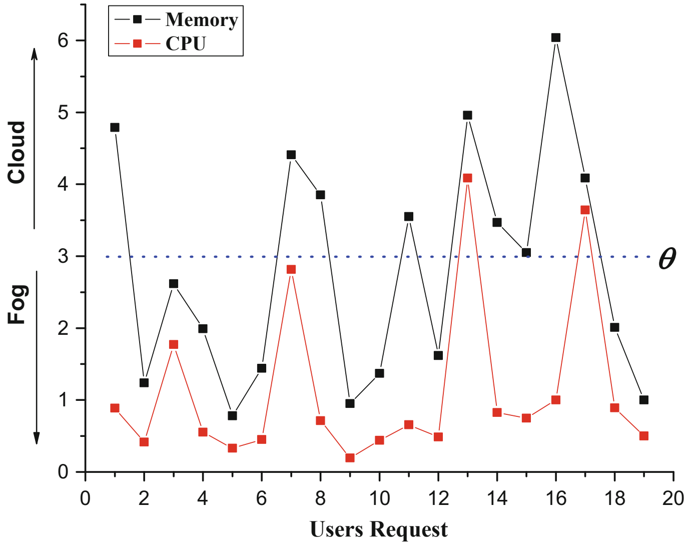


Fig. 7 Double-decision criteria

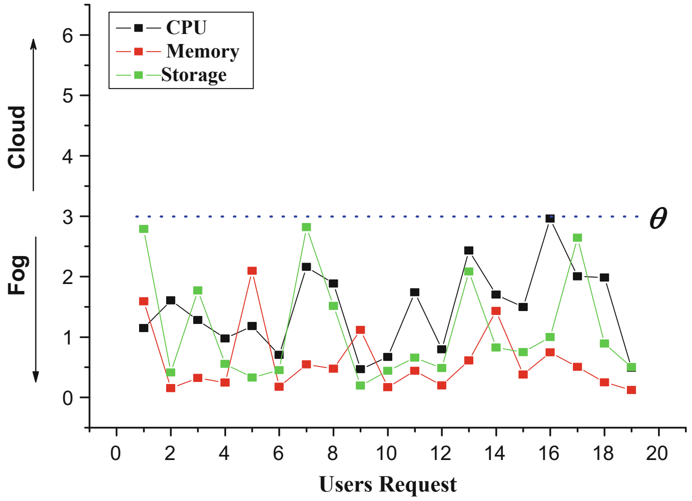


Fig. 8 Multiple decision criteria

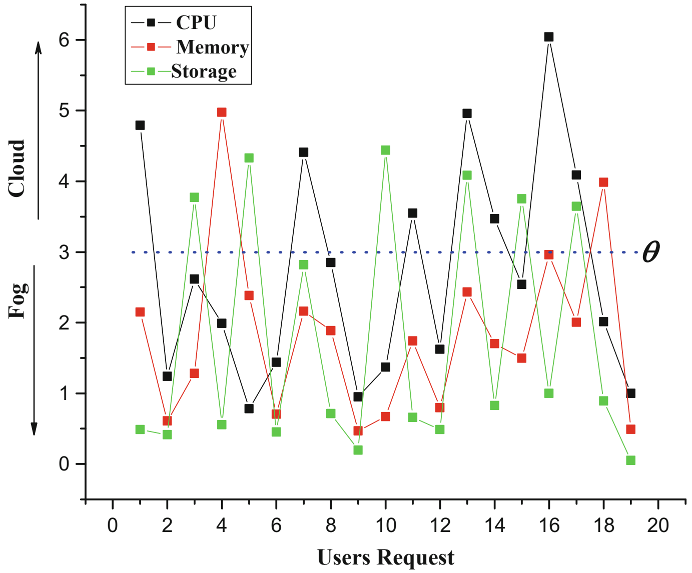


Fig. 9 Multiple decision criteria

Based on this model and its analysis, we can say that the energy efficiency of the datacenter and resource utilization has been improved. This is due to the workload and application processing migration when the user's request exceeded the threshold of the Fog environment. Therefore, this triggers the interaction of the two-environment Fog–Cloud resource. But when the average workload is less than the threshold value then the user's request will be processed at the Fog environment. Therefore, only application that require much CPU, Memory and Storage will be sent to Cloud datacenter for processing. This allows the efficient use of the Cloud—Fog resource that eliminates the loss of data, massive communication bandwidth, and energy consumption.

5 Future Issues and Direction

Based on our discussion in the literature, there are important aspects of Fog and Cloud Computing paradigm due to their inherent finite resource capacity offered as a service to users. Therefore, there is a need to explore the issues that are left behind because of the compromise in solving the existing problems. This will bring advancement in finding solutions to the existing issues that require urgent attention of the researchers in this domain. Some of the issues and future research direction are discussed in the following section.

Resource Provisioning This deals with choosing and deployment of programs run time for applications which can be dynamic or static. It is found from the literature, very few research works have considered dynamic resource provisioning or using both static and dynamic in Cloud–Fog environment. Many important aspects such as latency, mobility, network traffic, and resource availability are still unexplored. To efficiently use both Cloud and Fog resources, application of different techniques on resource provisioning and service management will be a promising subject in the theme of Cloud–Fog research.

Multitenancy Resource Management Existing Fog Computing resources can be virtualized and at the same time allocated to numerous user requests. The current multi-tenants' research works do not take into consideration of Fog Computing resource scheduling of computational tasks and request based on the user application requirements (performance degradation, makespan and throughput). This can be carried out in future researches.

Parameters The parameters used by the implemented techniques to determine their performance are shown in the table. The choice of the parameters on Cloud–Fog user perspective or Cloud service provider are not the same. The provider is concerned more about efficient resource utilization while the users are focused on application performance. In that regards, the user's parameters include latency, response time, execution time, fairness, turnaround time and tardiness, and the provider parameters include energy consumption reduction, resource utilization, VM migration, workload, budget, and other dependency constraints (death line, reliability and priority constraint). Not all these parameters were used. There is a need to explore more on the parameters to see their performances in Cloud–Fog environment.

The Sustainable of the Cloud–Fog Architecture The Fog Computing sustainability reflects both economic and the effect of the environmental to certain degree. Though, sustaining the Fog Computing architecture has been struggling with issues such as the QoS, service availability, energy efficiency, and resource utilization. Additional research in this domain is highly required for the performance and accomplishment of Fog Computing. This is because the present research work presents a limited analysis of sustainability and reliability of the Fog Computing architecture.

Interoperability It is the ability to share information between systems and application in a meaningful way. This will enable users to understand why they are receiving and sending information over the network with different configurations. Although, the Cloud Computing is already standardized and the operational risk is minimal. Current future of the Cloud Computing is on the federation of different infrastructure across different sites. But when they are interacting with Fog nodes that are on premises platform which are entirely different in terms of processing and storage capabilities. This shows that the users are now operating in a highly diverse environment, different facilities located around the world. There is a need to support this diversity in that environment so that they can run their applications on Cloud environment without service interruption or resource luck-in. In general, interoperability will allow new efficiency that will not be interrupted by some constraints from the Fog environment. Therefore, there is a need to adapt standard or common interface that will create more innovation, support for different applications by different vendors and data-driven insight.

6 Conclusion

Energy efficiency is important for Cloud datacenters and Fog Computing. Fog Computing is becoming a desirable answer to application processing issues in the IoT paradigm. In this chapter, three parts of the developed model have been explained which include Cloud-only application module placement and a technique that pushes applications toward Fog devices. The technique allows the Fog to operate while disconnected from a larger network. It also supports the intelligent decision of processing, storing, memory, and actuating when enough resources are available. Nevertheless, we recommend that the usage of Cloud Computing paradigm has several advantages as a support infrastructure and large-scale reasoning, but part of the control and trust decisions should be shifted to the Fog and allow for innovative computing applications. The model has been tested using the origin platform. Some issues related to double and multi-resource constraints have not been addressed in this work. This limitation will be addressed in future work.

References

1. Foster I et al (2008) Cloud computing and grid computing 360-degree compared. In: IEEE 2008 grid computing environments workshop, Austin, TX, pp 1–10
2. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener Comput Syst 28(5):755–768
3. Usman M et al (2016) A conceptual framework for realizing energy efficient resource allocation in cloud data centre. Indian J Sci Technol 9(46):1–8
4. Madni SHH, Latiff MSA, Coulibaly Y (2016) An appraisal of meta-heuristic resource allocation techniques for IaaS cloud. Indian J Sci Technol 9:1–14
5. Madni SHH, Latiff MSA, Coulibaly Y (2016) Resource scheduling for infrastructure as a service (IaaS) in cloud computing: challenges and opportunities. J Netw Comp Appl 68:173–200
6. Yi S, Li C, Li Q (2015) A survey of fog computing: concepts, applications and issues. In: 15th proceedings of the 2015 workshop on mobile big data, Hangzhou, pp 37–42
7. Yannuzzi M et al (2014) Key ingredients in an IoT recipe: fog computing, cloud computing, and more fog computing. In: 19th IEEE international workshop on computer aided modeling and design of communication links and networks (CAMAD), Athens, pp 325–329
8. Bose R, Saha MK, Sarddar D (2015) Fog computing made easy with the help of Citrix and Billboard manager. Int J Comput Appl 121:19–23
9. Masip-Bruin X et al (2016) Fog-to-cloud computing (F2C): the key technology enabler for dependable e-health services deployment. In: 15th Mediterranean ad hoc networking workshop (Med-Hoc-Net), Vilanova i la Geltru, pp 1–5
10. Deng R et al (2015) Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing. In: 5th IEEE international conference on communications (ICC), London, pp 3909–3914
11. Gital AY et al (2014) A framework for the design of cloud based collaborative virtual environment architecture. In: Proceedings of the international multiconference of engineers and computer scientists (IMECS), Hong Kong, pp 1–5
12. Usman MJ et al (2014) Modified low energy adaptive clustering hierarchy protocol for efficient energy consumption in wireless sensor networks. Int Rev Comput Softw (IRECOS) 9:1904–1915
13. Gital AY et al (2015) A zone-based CVE architectural model for improving scalability and consistency. Electrical Engineering and Information Technology (IEET), Hangzhou, China, pp 63–68
14. Deng R et al (2016) Optimal workload allocation in fog-cloud computing towards balanced delay and power consumption. IEEE Internet Things J 3:1171–1181
15. Al Faruque MA, Vatanparvar K (2016) Energy management-as-a-service over fog computing platform. IEEE Internet Things J 3:161–169
16. Tang B et al (2015) (2015) A hierarchical distributed fog computing architecture for big data analysis in smart cities. In: 2015 proceedings of the ASE big data & social informatics. ACM, Kaohsiung, Taiwan, pp 1–28
17. Stojmenovic I (2014) Fog computing: a cloud to the ground support for smart things and machine-to-machine networks. In: 26th Australasian telecommunication networks and applications conference (ATNAC), Southbank, VIC, pp 117–122
18. Costa R et al (2016) Smart cargo for multimodal freight transport: when “Cloud” becomes “Fog”. IFAC-PapersOnLine 49:121–126
19. Vatanparvar K, Faruque A, Abdullah M (2016) Energy management as a service over fog computing platform. IEEE Internet Things J 3:163–169
20. Masip-Bruin X et al (2016) Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. IEEE Wirel Commun 23:120–128
21. Madsen H et al (2013) Reliability in the utility computing era: towards reliable fog computing. In: 20th international conference on systems, signals and image processing (IWSSIP), Bucharest, pp 43–46

22. Bonomi F et al (2014) Fog computing: a platform for internet of things and analytics. In: Big data and Internet of Things: a roadmap for smart environments. Springer International Publishing, pp 169–186
23. Ottenwälder B et al (2013) MigCEP: operator migration for mobility driven distributed complex event processing. In: 7th ACM international conference on distributed event-based systems, Arlington, Texas USA, pp 183–194
24. Lewis G et al (2014) Tactical cloudlets: moving cloud computing to the edge. In: IEEE military communications conference (MILCOM), Baltimore, MD, pp 1440–1446
25. Aazam M, Huh E-N (2014) Fog computing and smart gateway based communication for cloud of things. In: 2014 2nd IEEE international conference on future Internet of Things and cloud (FiCloud), Bercelona, pp 464–470
26. Aazam M, Huh E-N (2015) Dynamic resource provisioning through fog micro datacenter. In: IEEE international conference on pervasive computing and communication workshops (PerCom Workshops), St. Louis, MO, pp 105–110
27. Nishio T et al (2013) Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud. In: ACM proceedings of the first international workshop on mobile cloud computing & networking, Bangalore, India, pp 19–26
28. Dsouza C, Ahn G-J, Taguinod M (2014) Policy-driven security management for fog computing: preliminary framework and a case study. In: IEEE 15th international conference on information reuse and integration (IRI), Redwood City, CA, pp 16–23
29. Dastjerdi AV et al (2016) Fog computing: principals, architectures, and applications. arXiv preprint arXiv:1601.02752
30. Liu F et al (2012) NIST cloud computing reference architecture: recommendations of the National Institute of Standards and Technology (special publication report 500–292). CreateSpace Independent Publishing Platform
31. Byers CC, Wetterwald P (2015) Fog computing distributing data and intelligence for resiliency and scale necessary for IoT: the Internet of Things (ubiquity symposium). Ubiquity, pp 1–12
32. Gupta H et al (2016) iFogSim: a toolkit for modeling and simulation of resource management techniques in Internet of Things, edge and fog computing environments. arXiv preprint arXiv:1606.02007
33. Bonomi F et al (2012) Fog computing and its role in the internet of things. In: ACM proceedings of the first edition of the MCC workshop on mobile cloud computing, Helsinki, Finland, pp 13–16
34. Perera C, Qin Y, Estrella JC, Reiff-Marganiec S, Vasilakos AV (2017) Fog computing for sustainable smart cities: a survey. ACM Comput Surv 50:32–43
35. Moreno-Vozmediano R, Montero RS, Huedo E, Llorente IM (2017) Cross-site virtual network in cloud and fog computing. IEEE Cloud Comput 4:46–53

Energy-Efficient Bias-Based User Association for Heterogeneous Networks in LTE-Advanced

Ayuba K. Danburam, Mohammed A. Gadam, Aliyu D. Usman and Suleiman M. Sani

Abstract Heterogeneous network (HetNet) deployment is a promising technique for improving energy efficiency in 4G and beyond wireless cellular systems. The major challenge of enhancing energy efficiency in HetNet is a poor cell selection when the conventional reference signal received power (RSRP) or biased RSRP (BRSRP) cell selection algorithm is employed. These cell selection techniques limit the potential of HetNet in improving transmission energy efficiency. The proposed energy-efficient bias setting strategy is an adaptive BRSRP cell selection algorithm. It uses energy efficiency as cell load metric for adaptive picocell range extension (CRE). The algorithm efficiently estimates the varying energy efficiency in each cell, then, based on the optimality gap of the energy efficiency, it adopts an optimized bias value per cell. Simulations using LTE system level simulator shows the proposed adaptive bias setting improves energy efficiency, average UE throughput and system capacity by 6.7, 9.7 and 6.9%, respectively when compared with BRSRP with a fixed bias of 6 dB. Although the proposed adaptive bias exhibits low offloading gain from PeNB to MeNB as against BRSRP, the system load balance has improved when compared with RSRP.

Keywords Heterogeneous network · Picocell range expansion · Cell selection · Energy efficiency · Traffic load balance

A. K. Danburam (✉) · A. D. Usman · S. M. Sani
Department of Electrical and Computer Engineering, Ahmadu Bello University, Zaria, Nigeria
e-mail: danburamayuba@gmail.com

A. D. Usman
e-mail: aliyuusman1@gmail.com

S. M. Sani
e-mail: smsani@abu.edu.ng

M. A. Gadam
Department of Electrical and Electronic Engineering, Federal Polytechnic Bauchi, Bauchi, Nigeria
e-mail: agmohammed@fbtb.edu.ng

T. Herawan et al. (eds.), *Advances on Computational Intelligence in Energy*,
Green Energy and Technology, https://doi.org/10.1007/978-3-319-69889-2_8

1 Introduction

The demand for mobile broadband services and improved device capabilities drives the strong increase in the unprecedented traffic volumes and consumer data rate [1]. Deploying more macro base stations (BS) hereinafter referred to as Macro evolved NodeB (MeNB) onto an existing homogeneous network is limited by poor cell splitting gain due to high co-channel interference [2]. More so, the high cost of site acquisition and operational cost due to high energy consumption associated with MeNB deployment make it difficult to achieve desired revenue and quality of service. Therefore, with homogeneous deployment, the mobile data traffic revenues are not commensurate with the actual traffic growth. The mobile network operators spend about 25% of the total network operation cost on electric energy, which is largely generated from fossil fuel [3]. Therefore, the challenge is providing quality service while operating within acceptable cost of operation to the expanding mobile networks. To cope with this challenge, Heterogeneous Network (HetNet) deployment strategy was proposed and standardized by the 3rd Generation Partnership Program (3GPP) [4]. HetNet is realized by overlaying low power nodes (LPNs) onto high power macro area through spectrum reuse of one. The LPNs deployed in HetNet includes Pico eNodeB (PeNB), femtocell and relay nodes. HetNets are being increasingly deployed by operators, and PeNB is most preferred because of ease of planning and deployment [5]. Apart from improving capacity, another benefit of deploying PeNBs is to reduce coverage holes. Especially where radio signal strength from MeNB is low that user equipment (UE) is not served by MeNB [6]. More also, network deployment based on PeNB is a potential solution for reducing total power consumption of a cellular network [7]. The fact being that a base station referred to as eNodeB (eNB), closer to mobile users, lowers the required transmit power due to advantageous lower path loss [8].

However, HetNet deployment brings about new challenges due to the diverse transmit power levels of MeNB and PeNB in HetNet. Most UEs prefer to associate with the MeNBs, with the conventional reference signal received power (RSRP)-based user association scheme. This results in uneven distribution of traffic load and in turn underutilization of the resource in PeNBs [9]. In order to solve the problem of traffic imbalance, 3GPP as part of its standardization effort proposed the biased reference signal received power (BRSRP) user association also known as cell range expansion (CRE). This is aimed to proactively offload users to PeNBs by utilizing an association bias [10]. However, CRE with fixed bias lacks information on traffic load conditions in the cell due to the dynamic UE distribution as well as varying environmental conditions in the system [11]. This shortcoming results in a wrong bias setting, consequently, poor traffic load balance, reduced throughput, and hence reduced energy efficiency in the system. Therefore, bias for CRE needs to be dynamically set to adapt to traffic load for improved system performance [6].

In this chapter, an energy-efficient adaptive bias setting for optimal system performance is proposed. The proposed technique uses energy efficiency threshold estimated per macrocell area coverage, to represent cell load condition in order to set

PeNB bias value per cell. On one hand, the estimated energy efficiency per macro area represents the load condition in the cell and reduces the complexity associated with UE distribution. On the other hand, configuring bias values per cell reduces high signaling overhead and UE frequent handover associated with setting bias value per UE. More so, the bias value is dynamically set, if and only if a defined network energy efficiency threshold is reached in order to avoid frequent handover. The bias value is dynamically set to avoid poor bias setting due to varying load conditions and UE service demand.

Therefore, the main contributions of this chapter are as follows;

1. It proposed an adaptive bias cell association which uses energy efficiency as cell load metric and considers spatial distribution of UEs.
2. It developed a dynamic bias configuration strategy which enhances energy efficiency and traffic load balance with reduced network complexity and signaling overhead.

This chapter is organized as follows: Section 2 present related literature, Sect. 3 introduces cell association and its procedures and Sect. 4 presents the system model. Description of the proposed algorithm is presented in Sect. 5. Section 6 deals with the scenario description and the simulation assumptions, Sect. 7 presents results and discussion while Sect. 8 provides the conclusion of the chapter.

2 Related Literature

The fixed CRE bias value method implemented by authors in [11] is not practical due to the fact that networks need to adapt to the variations in environmental conditions. Hence, the adaptive bias proposed and studied by the authors in [11–16] was meant to address the issues of dynamics in environmental conditions. Proposed also by authors in [12, 15] was an adaptive cell range control in HetNet utilizing the cell edge UE capacity. This scheme assumed cell load metric to be represented by the number of pieces of UE as resource block utilization ratio (RBUR) acquired through the network. Near-optimal cell edge UE throughput gain of over 6 dB static bias setting was achieved by this scheme. Furthermore, since cell load estimation is acquired from the network side, this makes the scheme simple, devoid of feedback delays and errors. However, since traffic load varies with user mobility and the scheme is lacking criteria for deciding the cell load threshold, then number of pieces of UE employed by the authors cannot efficiently represent cell load condition.

The work in [13] presented a distributed Q-learning-based CRE in HetNet to improve both cell edge and average UE throughput, and outage reduction was employed as performance metric. The ratio of resource blocks to the UE distribution was also considered as cell load metric. The input to the algorithm is based on the previous experience of distributed pieces of UE that learn their optimal bias values. The algorithm achieves maximized network throughput since pieces of UE learn its bias values from past experience. However, due to the long convergence time associated with the Q-learning, it is therefore not suitable for real systems.

Further, parameter optimization for adaptive control CRE in HetNet based on SINR was studied by the authors in [14]. The scheme utilizes the ratio of the number of pieces of UE connected to PeNBs and MeNBs as cell load metric. The scheme input centralization was achieved due to the SINR feedback from UE. The ability of the algorithm to solve the trade-off between cell edge UE throughput and the average UE throughput makes it simple and with a near-optimal performance. However, since traffic load depends on resource block and SINR, the feedback from each UE also brought about delay in the system; therefore, the number of pieces of UE cannot efficiently estimate the cell's load condition.

The authors in [15] presented a simple cell association method for HetNets based on expected minimum average UE throughput. A combined metric comprising of MeNB index and resource index to maximize the UE throughput was employed as cell load metric. The decentralization of the inputs in the algorithm has therefore removed the need for coordination among MeNBs. It was evident that faster convergence was achieved by the algorithm, and further improved performance was also recorded with enhanced interference and inter-coordination (eICIC). The algorithm is robust in canceling the effect of interference because the eICIC has the ability to adapt according to the variation of UE distribution. However, large overheads due to feedbacks were observed in the algorithm which leads to frequent handover problems with multiple pieces of UE. It is a known fact that frequent handover leads to scheduling outage.

The authors in [16] proposed PeNB CRE-based cell association algorithm employing adaptive CRE bias in HetNet. The algorithm utilized the measurement in the uplink interference to adjust PeNB coverage areas dynamically. For the PeNB that suffers high interference, small CRE bias values are acquired to shrink their coverage areas. On the other hand, PeNBs subjected to less interference has their coverage areas extended to provide services for those areas previously covered by the loaded neighboring cells. To increase the uplink transmission rates, the algorithm makes cell selection decision based on the uplink interference. However, this may not necessarily provide the best downlink rates because the PeNB UEs in the range extended are exposed to severe interference from MeNB in the downlink, consequently, reducing their downlink data rates. Therefore, MeNB that gives the maximum uplink rates can be different from the one for the downlink.

The authors in [17] proposed a distributed, priority and non-biased-based channel access and load-aware association technique with interference mitigation in their most recent work. Their work proposed combined priority access based on maximum channel gain for high priority UEs and hybrid channel gain and access-based cell association for low priority UEs for rate maximization and load balancing. The work achieved a good traffic load balance and the best offloading gain from MeNB to PeNB when compared with RSRP and PeNB CRE with the fixed bias of 6 dB, respectively. Lower UE throughput was achieved despite employing interference mitigation because the resource blocks available in the PeNB were not able to compensate for the severe interference suffered by PeNB cell edge UEs. This ultimately affects the overall system performance and thereby reducing the energy efficiency of the network.

In the most recent work of authors in [18], the effect of RSRP cell selection algorithm and BRSRP with the different bias on energy efficiency and traffic load balance of downlink in LTE-Advanced HetNet was investigated and verified through simulation. The modeling of power consumption of base station was based on base station power consumption parameters, where power consumption was assumed to be constant irrespective of traffic [19]. The data rate was modeled based on link adaption considering spatial distribution of UEs. From simulation result, RSRP performed better in terms of total throughput and overall energy efficiency. However, RSRP achieved poor traffic load balance due to poor offloading gain from MeNB to PeNB. BRSRP with a bias of 16 dB achieved the worst performance due to severe interference. BRSRP with a bias between 3 dB and 9 dB achieved a trade-off between traffic load balance and system performance. Therefore, adaptive bias setting has the potential to improve system performance and traffic load balance.

3 Theoretical Background

Cell selection is the process in which pieces of UE attach themselves to the serving cell for communication based on certain criteria [11]. The cell selection criteria are determined by utilizing quality, coverage and load-based cell selection [16]. The quality-based cell selection is initiated when the UE has better channel quality with a candidate eNB than its serving eNB. And the coverage-based cell selection occurs when UE has measured better RSRP with a candidate eNB than it serving eNB. While in load-based cell selection, the congested eNB needs to handover some users to a neighboring eNB to balance the traffic load [16].

3.1 RSRP Cell Selection

The RSRP cell selection is a coverage-based cell selection, where UE connects to eNB that has the highest RSRP. For a HetNet, let RSRP from eNB n be denoted by P_n in dB and defined according to [6] as follows:

$$\text{RSRP} = \arg\ \max_n\{P_n\} \tag{1}$$

The method in Eq. (1) results in the underutilization of the resource at low powered eNB(s) having lower transmit power, and it also tends to overload the high power eNB(s) having higher transmit power thereby reducing system performance [9]. This is due to the fact that data rate varies linearly with resource block. Hence, load balancing is very critical in achieving high data rates which is the key performance metric [17].

3.2 Biased RSRP (BRSRP)-Based Cell Selection

To tackle the traffic load balancing problem, the BRSRP-based association also known as cell range extension (CRE) is considered [10]. In such a scheme, an arbitrary bias is added to the RSRP from PeNBs which offloads users from MeNBs to PeNBs [18, 20]. CRE connects UEs to PeNB(s) rather than MeNB(s) by adding a bias value called the "PCRE bias value," to the signal level received from PeNBs transmission power [18]. Hence, with CRE, PeNBs seem to have greater reference signal strength than usual. Let RSRP from PeNB n and the CRE bias value for the PeNB be denoted by P_n [dB] and B_n [dB], respectively. Then, the UE selects the serving eNB that connects with the UE by the following equation according to [2].

$$\text{BRSRP} = \arg\ \max_n\{P_n + B_n\} \tag{2}$$

For the purpose of pico CRE, B_n is selected to be a positive value so that the handover boundary is shifted closer to the MeNB as in [6] depicted in Fig. 1.

CRE bias value does not increase the transmission power from the PeNBs but makes UEs do handovers earlier to the PeNBs since they have a positive CRE bias value [6]. The coverage area is not affected by load imbalance in the uplink because the UE holds equal transmit power [16]. CRE provides significant improvement for UEs in the uplink as a result of reduced path loss since the link distance are

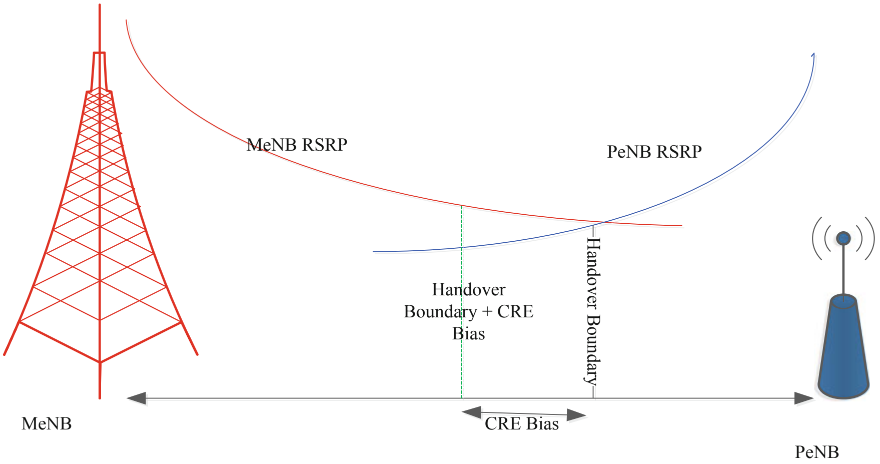


Fig. 1 Biased receive signal power based association scheme

reduced. However, in the downlink transmission, picocell edge UEs are exposed to severe interference from MeNB for two reasons: First, the picocell edge UEs are furthest away from the serving PeNB. Secondly, this UEs are much closer to the interfering macrocells. These consequently reduce the throughput of the picocell edge UEs as data rate varies logarithmically with SINR. CRE for picocells leads to uplink-downlink traffic imbalance [16].

Therefore, with RSRP, heavily loaded MeNB provides lower data rates despite holding higher SINR, and with BRSRP, the available resource in PeNB cannot compensate for the lower SINR. Hence, adaptive bias setting that considers the dynamic distribution of UEs will achieve a better system performance.

4 System Model

The default cell selection criteria in HetNet are the conventional RSRP for 3GPP release 8/9 and BRSRP for 3GPP release 10/11 [17]. However, both RSRP and BRSRP cell selection criteria suffer the same problem of poor energy efficiency and traffic imbalance even with the best network configuration. Therefore, an energy-efficient cell selection criterion is proposed based on adaptive bias setting. RSRP for MeNB and PeNB was modeled according to [5] as follows:

$$\mathrm{RSRP}_m = \mathrm{PTX}_m - \mathrm{PL}(m) - \mathrm{SF}(m) + \mathrm{GA}(m) - L_{\mathrm{misc}} \tag{3}$$

$$\mathrm{RSRP}_p = \mathrm{PTX}_p - \mathrm{PL}(p) - \mathrm{SF}(p) + \mathrm{GA}(p) - L_{\mathrm{misc}} \tag{4}$$

where RSRP_m and RSRP_p are RSRP from MeNB and PeNB, respectively, PTX_m and PTX_m are transmit power of MeNB and PeNB, respectively in dB, $\mathrm{PL}(m)$ and $\mathrm{PL}(p)$ are path loss of UE from MeNB and PeNB, respectively, $\mathrm{SF}(m)$ and $\mathrm{SF}(p)$ are shadows fading for MeNB and PeNB, respectively, $\mathrm{GA}(m)$ and $\mathrm{GA}(p)$ are antenna gain of MeNB and PeNB, respectively, and L_{misc} is any miscellaneous losses. The conventional RSRP and BRSRP cell association was modeled as follows:

$$\mathrm{RSRP} = \max\left\{\mathrm{RSRP}_m, \mathrm{RSRP}_p\right\} \tag{5}$$

The BRSRP was modeled as follows:

$$\mathrm{BRSRP} = \max\left\{\mathrm{RSRP}_m, \mathrm{RSRP}_p + B_n\right\} \tag{6}$$

whereas the proposed energy-efficient bias setting (EEBS) was modeled as follows:

$$\mathrm{EEBS} = \max\left\{\mathrm{RSRP}_m, \mathrm{RSRP}_p + \beta_n(\mathrm{EE}_H)\right\} \tag{7}$$

where B_n is the fixed CRE bias value for PeNB in dB, β_n is the variable CRE bias value which varies in the range of 0–16 dB and EE_H is highest downlink energy efficiency for each network realization. Therefore, UE to eNB association decision will vary for different association schemes as follows:

i. RSRP: The new users will likely associate to MeNB due to higher transmitted power.
ii. BRSRP: For reasonably large values of bias which is static, the new user will be forced to select the PeNB.
iii. EEBS: Depending on the estimated downlink energy efficiency, a dynamic bias value is set and the new users can choose to associate between MeNB and PeNB.

Other benchmark models for measuring system performance such as throughput, energy efficiency and load balance are also presented in this section. In this section data rate and throughput are used interchangeably. The detailed scenario description is presented in Sect. 6.

4.1 Capacity Model

For this chapter, round-robin resource allocation was used, where it was assumed that users within a cell share the available resource block equally so that they can attain higher data rate [21]. Therefore, the number of resource block allocated to a user at distance d from an eNB per transmission time interval was modeled according to [22]:

$$\mathrm{NRB}(\mathrm{uid}, d) = \frac{\mathrm{NRB}/\mathrm{TTI}}{\mathrm{NumUEpercell}} \tag{8}$$

where NRB is the number of resource block per cell, NumUEperCell is the number of UE per cell; TTI is the transmission time interval. Therefore, from Eq. (8), the number of resource block for a pico UE and a macro UE was modeled respectively as follows:

$$\mathrm{NRB}_{\mathrm{Pico}}(h, k) = \frac{\mathrm{NRB}}{q} \tag{9}$$

$$\mathrm{NRB}_{\mathrm{macro}}(h) = \frac{\mathrm{NRB}}{p} \tag{10}$$

where h has values from 1 to the number of MeNB and k has values from 1 to the number of PeNB per macrocell, $\mathrm{NRB}_{\mathrm{Pico}}(h, k)$ is the resource block available to a user connected to a picocell k in a macrocell h, $\mathrm{NRB}_{\mathrm{macro}}(h)$ is the resource block available to a user connected to a macrocell h, while q and p are the number of users connected to picocell k and macrocell h, respectively. The data rate for a

UE is calculated based on the UE distance d from the eNB, the average signal to interference and noise ratio (SINR) at the UE is defined according to [22, 23] as;

$$\mathrm{SINR}(\mathrm{uid}, d) = \mathrm{PTX} + \mathrm{GA} - N - I - \mathrm{SF}(d) - \mathrm{PL}(d) - L_{\mathrm{misc}} \tag{11}$$

where PTX is the eNB transmission power (per cell sector); GTx *and* GRx are the eNB and UE antenna gains, respectively. N and I are the noise and the inter-cell interference (ICI) power from all the interfering eNBs at the UE location respectively. L_{misc} is any miscellaneous loss e.g. the wall penetration loss for signals received by indoor UE. Finally, PL(d) and SF(d) are the path loss and shadow loss in dB, respectively, measured at different UE positions.

In LTE, the Media Access Control (MAC) layer allocate the physical Transmission Block Size (TBS) which depends on the Modulation and Coding Scheme (MCS) selected by the MAC layer scheduler. This is based on the Channel Quality Indicator (CQI) reported by the UE after every Transmission Time interval (TTI) of 1 ms [22]. Link adaptation requires the selection of a proper MCS according to the channel quality which is usually indicated by the SINR reported by each UE [24]. Following the LTE specification in [25], three modulation levels of QPSK 16-QAM and 64-QAM are supported. Together with turbo coding, there are 26 MCSs, this implies that there are 26 CQI. The SINR to TBS mapping for these MCSs, assuming a block error rate (BLER) target of 10% was modeled using the following procedure. First, the effective SINR of a UE was modeled according to [24] as;

$$\mathrm{SINR}_{\mathrm{eff}}(\mathrm{uid}, d) = \min\{\mathrm{SINR}(\mathrm{uid}, d), \mathrm{SINR}_{\mathrm{threshold}}\} \tag{12}$$

where $\mathrm{SINR}_{\mathrm{eff}}$(uid) is the effective SINR of a UE for mapping to corresponding CQI and TBS. SINR(uid, d) is the SINR as a result of the UE's instantaneous channel conditions as in Eq. (11). And $\mathrm{SINR}_{\mathrm{threshold}}$ is the SINR value corresponding to the 26 MCSs level. The mapping of SINR to TBS was modeled as follows:

$$\mathrm{TBS}(\mathrm{uid}, d) = \mathrm{TBS}(\mathrm{SINR}_{\mathrm{eff}}(\mathrm{uid}, d)) \tag{13}$$

where TBS(uid,d) is the TBS in bits allocated to UE based on $\mathrm{SINR}_{\mathrm{eff}}$(uid).

In LTE, there are 7 OFDMA symbols in a resource block (RB) and TTI of 0.5×10^{-3} s. For two RB pairs, the TTI is 1×10^{-3} s and number of subcarriers is 12 [22]. Hence, the number of symbols in an RB is equal to 7 multiplied by 12 which is 84 in TTI of 0.5×10^{-3} s and 168 in TTI of 1×10^{-3} s for an RB pair. The number of bits in an RB is the number of symbols multiplied by the number of bits per symbol. The number of bits per symbol is the modulation index multiplied by the coding rate which is the symbol efficiency. Therefore, the number of bits in an RB pair is the symbol efficiency multiplied by 168, and this is the TBS. Therefore, data rate (R) for a UE i is given according to [22] as follows:

$$R(i) = \frac{\mathrm{TBS}(i) \times \mathrm{NRB}(i)}{\mathrm{TTI}}(1 - \mathrm{BLER}(i)) \tag{14}$$

where $\mathrm{TBS}(i)$ is the physical transmission block information capacity (in bits) for the CQI state I, and $\mathrm{BLER}(i)$ is the average block error rate (BLER), TTI is the transmission time interval and $\mathrm{NRB}(i)$ is the number of resource block allocated to UE i. The achievable data rate for each UE has been modeled based on instantaneous channel conditions. Therefore, from Eq. (14), the data rate $R(\mathrm{uid},d)$ delivered to a UE (*uid*) placed at distance d from the eNB assuming constant targeted BLER was modeled as follows:

$$R(\mathrm{uid}, d) = \frac{\mathrm{TBS}(\mathrm{uid}, d) \times \mathrm{NRB}(\mathrm{uid}, d)}{\mathrm{TTI}}(1 - \mathrm{BLER}) \tag{15}$$

Hence, putting the value of $\mathrm{NRB}(\mathrm{uid}, d)$ of macro and pico UE from Eqs. (9) and (10), respectively, the data rate for macrocell UE and picocell UE was modeled as follows:

$$\mathrm{RMUE} = \frac{\mathrm{TBS}(\mathrm{uid}, d) \times \mathrm{NRB}_{\mathrm{macro}}(h)}{\mathrm{TTI}}(1 - \mathrm{BLER}) \tag{16}$$

$$\mathrm{RPUE} = \frac{\mathrm{TBS}(\mathrm{uid}, d) \times \mathrm{NRB}_{\mathrm{Pico}}(h, k)}{\mathrm{TTI}}(1 - \mathrm{BLER}) \tag{17}$$

where RMUE and RPUE are the data rate for UE connected to macrocell and picocell, respectively. Therefore, the total data rate within a macro area coverage was modeled as follows:

$$\mathrm{RallUE} = \sum \mathrm{RMUE} + \sum \mathrm{RPUE} \tag{18}$$

where RallUE is the total data rate. Therefore, from Eq. (18), the average macro area throughput and average UE throughput was modeled as follows:

$$\text{Average Macro Area Throughput} = \frac{\mathrm{RallUE}}{\mathrm{NumCell}} \tag{19}$$

$$\text{Average UE Throughput} = \frac{\mathrm{RallUE}}{\mathrm{NumUE}} \tag{20}$$

where NumCell is the total number of MeNB and NumUE the total number of UE.

4.2 *Energy Efficiency Model*

Assuming static power consumption irrespective of traffic load situations, the base station power consumption is defined as in [19] by:

$$\mathrm{Pc}_i = \mathrm{Nsec\ Nant(Ai\ PTX + Bi)} \tag{21}$$

where Nsec and Nant denote the eNBs's number of sectors and the number of antennas per sector, respectively. Pc_i is the average total power per base station, and PTX is the power fed to the antenna as in Eq. (3). The coefficient Ai accounts for the part of the power consumption that is proportional to the transmitted power, which includes radio frequency amplifier power and feeder losses. While Bi denotes the power that is consumed independently of the average transmit power which includes signal processing and site cooling [19]. The energy efficiency is defined as the ratio of the total data rate (RCi) delivered within a cell and power consumption of the cell (Pci), which is defined as in [19] by:

$$\text{EE}_{\text{Ci}} = \frac{\text{overall data rate}}{\text{total power consumed}} = \frac{\text{RCi}}{\text{PCi}} \tag{22}$$

where RCi is the overall data rate in bits/s within a cell, and PCi is the total power consumption of the cell in watts and EE_{Ci} is the transmission energy efficiency for all UE in bits/joule within the cell. Therefore, from Eq. (22), the total transmission energy efficiency for HetNet was modeled as follows:.

$$\text{EE}_{\text{HetNet}} = \frac{\text{RallUE}}{\text{Pc}_{\text{macro}} + \text{Pc}_{\text{pico}} \times \text{k}} \tag{23}$$

where RallUE is the total data rate obtained using Eq. (18). Pc_{macro} and Pc_{pico} are the power consumption of MeNB(s) and PeNB(s), respectively, obtained using Eq. (20), while k is the number of PeNB per macrocell. Also from Eq. (23), the transmission energy efficiency of macrocell and picocell was modeled as follows:

$$\text{EE}_{\text{macro}(i)} = \frac{\text{RMUE}}{\text{Pc}_{\text{macro}}} \tag{24}$$

$$\text{EE}_{\text{pico}(i)} = \frac{\text{RPUE}}{\text{Pc}_{\text{pico}}} \tag{25}$$

where $\text{EE}_{\text{macro}(i)}$ is the energy efficiency of macrocell in bits/joule, $\text{EE}_{\text{pico}(i)}$ is the energy efficiency of picocell in bits/joule. Pc_{macro} is the power consumption of MeNB in watts, Pc_{pico} is the power consumption PeNB in Watts. The average energy efficiency is given as follows:

$$\text{AV.EE} = \frac{\sum \text{EE}_{\text{HetNet}}}{\text{NumCell}} \tag{26}$$

where AV.EE is the average energy efficiency in bits/joules, NumCell is the number of macrocells.

4.3 Load Balancing Fairness Measure

The total number of UEs in a MeNB area comprises of UEs connected to MeNB and UEs connected to PeNB(s). The same frequency is reused in the MeNB and PeNB, and using round-robin resource allocations, equal time resources are assigned to each UE. Therefore, it is expected that the average MeNB (MUE) throughput and average PeNB UE (PUE) throughput given as U_M and U_P, respectively, is the same for optimal load balance between MeNB and PeNB(s) according to [2]. Hence, the load balancing fairness index U_K is formulated as follows:

$$U_K = |U_M - U_P| \tag{27}$$

where $|U_M - U_P|$ is the difference between the average MeNB (MUE) throughput and average PeNB UE (PUE) throughput or vice versa. The smaller the value of U_K the more balanced the system load distribution between MeNB and PeNB.

5 Description of Proposed Energy-Efficient Bias Setting

In the conventional BRSRP cell selection algorithm, fixed bias values are set. Due to varying load condition in each cell of a mobile wireless HetNet, fixed bias setting will lead to poor network performance [11]. It will only make sense if the bias value is set based on current traffic load. Therefore, an improved cell selection algorithm is proposed to dynamically set bias value based on estimated energy efficiency as a load metric. The algorithm efficiently estimates the varying load in each cell, then, based on the estimated load, a bias value is set per cell. This will ensure that an optimized energy efficiency and traffic balance are achieved in the HetNet system.

The proposed algorithm requires that transmission energy efficiency for different bias values (α) is first estimated, and then the bias value that yields the optimal energy efficiency is configured per cell. It is noteworthy that energy efficiency is estimated for every network realization, but adaptive bias configuration is carried out only when an energy efficiency optimality gap is exceeded. This is to ensure that frequent handover especially for cell edge UE is avoided. The optimality gap (λ) is modeled as the ratio of the absolute difference between the energy efficiency associated with configured bias (EE_H) and measured energy efficiency (EE_M). The energy efficiency optimality gap is expressed as follows:

$$\lambda = \left(\frac{|\mathrm{EE}_H - \mathrm{EE}_M|}{\max(\mathrm{EE}_H, \mathrm{EE}_M)} \right) * 100 \tag{28}$$

where EE_H is the energy efficiency of the bias value initially configured, and EE_M is the new energy efficiency measured for each network realization. The higher the difference between EE_H and EE_M or vice versa the higher the value of λ, and the higher the value of λ the more significant change between the estimated energy efficiency and the previous value of energy efficiency. Therefore, for a small value of λ, the current bias value remains unchanged else the process of adaptive configuration is initialized to obtain the new bias value that yields an optimized performance. In this chapter, the value of λ was set to be 10%. This value was obtained after running several simulation runs, and worst case was considered for the same number of UE to mitigate frequent handover.

There must be a limit to the value of CRE bias to be selected even though a larger value of bias yields high offloading gain; however, the larger the bias value the poorer the SINR of picocell edge UEs [2]. In the proposed algorithm, 16 dB is considered as the maximum CRE bias value β_n. This is because the difference between MeNB and PeNB transmit power is 16 dB. CRE bias interval of 3 dB is considered to minimize the number of bias values for each network realization. This is done to avoid frequent handover and reduce simulation time. This means for each network realization energy efficiency for each of the bias values will be estimated and stored with the corresponding bias value. The bias value that yields the overall highest energy efficiency will be automatically configured on the PeNB. The bias value will be configured per macro area coverage and it will be implemented by the MeNB. Therefore, PeNB(s) in the same macro area coverage will have the same bias value and will be controlled by the MeNB following similar procedure with the X-2-based handover [26]. The energy efficiency estimation will be carried out periodically such that as the load condition varies, the network will automatically adapt a new bias setting that will yield optimized performance.

The input to the proposed algorithm is acquired from the network side. The proposed method is similar to the CRE-based cell selection which is employed in the 3GPP standard. In both methods, the RSRP measurement and CRE bias values are used to consider the best serving cell. The novelty in this chapter is the development of an adaptive CRE bias setting using energy efficiency as cell load metric. Algorithm 1 further explains the process of adaptive bias setting.

```
Algorithm 1: Energy Efficient Adaptive Bias Setting
initializations
 $\beta_n$= 1:3:$\beta_n$max, $EE_{TH}$= 0
      For k = 1: Number of Drops
           For u = NumUEs
              For m = NumMeNB
                     Calculate UE $RSRP_m$ according to eqn (3)
                     For n = NumPeNB
                     Calculate UE $RSRP_p$ according to eqn (4)
                            If $RSRP_m > RSRP_p + \beta_n$
                         MeNB UE
                     Else
                         PeNB UE
                  End
              Compute SINR for all UEs according to eqn (11)
           End
     Compute PeNB throughput according to eqn (16)
 Compute PeNB energy efficiency according to eqn (24)
```

```
18.0   End
19.0   Compute MeNB throughput according to eqn (17)
20.0  Compute MeNB energy efficiency according to eqn (25)
21.0  End
22.0   Compute overall energy efficiency according to  equation (26)
23.0   If $\beta_n$ < $\beta_n$max
24.0      Increment $\beta_n$ by 3 and repeat step 1.0 to 23.0
25.0         Else
26.0         End
27.0           Compute $\lambda$ according to eqn (28)
28.0             If $\lambda$ < 10%
29.0             Repeat 1.0 to 28.0
30.0               else
31.0              Configure bias with highest energy efficiency
32.0   Compute load balance fairness according to eqn (27)
33.0                End
34.0 End
```

6 Scenario Description and Simulation Assumption

Based on the 3rd Generation Partnership Project (3GPP) LTE system level simulations toolbox defined in [27], a system of seven wraparounds sectored MeNB (21 cells) with four PeNB per sector is considered in this chapter. The PeNBs are randomly dropped within a MeNB area with minimum inter-site distance constraints. Each sector has a directional antenna at 120° apart one for each sector, while the PeNB has an omni-directional antenna. Users are uniformly distributed throughout the coverage area. Mobility is represented by users having different locations in each drop. Other related system level simulation parameters are specified in Table 1.

6.1 *Propagation Models*

For this chapter, single antenna receivers and transmitters are assumed, and therefore, only large-scale parameters are considered in the channel model according to [25].

$$P_{\mathrm{RX}} - P_{\mathrm{TX}} = \mathrm{PL} + \mathrm{SF} + G_A + L_{\mathrm{misc}} \tag{29}$$

where P_{RX} and P are the received and transmit powers,respectively, and PL and SF are the path loss and fading due to shadowing, respectively. The directional antenna gain is given as G_A and L_{misc} is any miscellaneous loss such as feeder cable loss. The path loss and shadow fading are carrier frequency dependent, and UE communication link is either line of sight (LoS) or non-line of sight (NLoS) depending on whether the location of the UE is indoor or outdoor. The path loss models of MeNB and PeNB are expressed by Eqs. (30) and (31), respectively, according to [25].

Table 1 System level simulation parameters

Parameter		Setting/Description
Cell layout		7 Hexagonal MeNBs; 3 sectors; reuse 1
MeNBs radius		500 m
Bandwidth and carrier frequency		10 MHz at 2000 MHz
Number of PeNBs per sector		4
Hotspot radius		40 m
Minimum distances between	MeNBs and PeNBs	75 m
	Among PeNBs	40 m
	MeNBs and UE	35 m
	PeNBs and UEs	10 m
Transmission power	MeNBs	46 dBm
	PeNBs	30 dBm
Path loss	MeNBs	128.1 + 37.6 $\log_{10}$ (r [km]) [24]
	PeNBs	140.7 + 36.7 $\log_{10}$ (r [km]) [24]
Number of UEs per sector		10, 20, …, 100
UE distribution		Uniform distribution [27]
Packet scheduler		Round-Robin
Power consumption parameters		Macro: Ai = 21.45; Bi = 354.44, Pico: Ai = 5.5; Bi= 38 [19]

$$128.1 + 37.6 \log(r[\text{km}]) \tag{30}$$

$$140.7 + 37.6 \log(r[\text{km}]) \tag{31}$$

where r is the three-dimensional (3D) distance between the UE and the MeNB or PeNB which is expressed as follows:

$$r = \sqrt{a^2 + b^2} \tag{32}$$

where the absolute antenna height difference between an eNB and a UE is denoted by b and a, respectively, is the two-dimensional (2D) distance between an eNB and a UE.

6.2 Antenna Patterns

The 3-sector antenna pattern used for each sector, reverse link and forward link is specified according to [25] by:

$$A\theta = -\min\left[12\left(\frac{\theta}{\theta_{3\mathrm{dB}}}\right)^{2}, A_{\mathrm{m}}\right] \tag{33}$$

θ is defined as the angle between the direction of interest and the boresight of the antenna, is the 3 dB beamwidth in degrees, and A_{m} is the maximum attenuation. For a 3-sector scenario, $\theta_{3\mathrm{dB}}$ is 70°, and A_{m} is 20 dB.

7 Results and Discussion

This section evaluates the performance of the proposed adaptive bias setting and draws comparisons with the existing cell association techniques. The metrics to be compared are the connection proportion of UEs to PeNBs, SINR, load balancing fairness using the difference between average UE throughput for PeNB and MeNB [2], MeNB area throughput and average UE throughput and energy efficiency, respectively.

7.1 Proportions of UEs Connected to PeNB

The simulation was carried out for different number of UEs for the HetNet configuration 1. The proposed adaptive bias setting has the highest proportions of UEs connected to the PeNBs when 10 UEs per cell were allowed into the network, after which the BRSRP with 6 dB bias maintains the highest, as shown in Fig. 2.

This is due to the offloading of more UEs from MeNB to PeNBs as a result of the effect of pico CRE associated with BRSRP and the proposed adaptive bias. The proportion of UEs connected to PeNB for BRSRP with bias of 6 dB is about 10% higher than the conventional RSRP for the different number of UE simulated, whereas, for the proposed adaptive bias, it is about 15% for 10 UE per cell, about 5% for 20 and 30 UE per cell, after which there is no significant difference with RSRP. This implies that as the number of UE increases, the proposed adaptive bias tends to configure smaller bias values.

For the proposed adaptive bias, the proportion of PeNB UEs decreases for up to 20 UEs in the system, but allowing up to 30 UEs into the system; however, the connection ratio stabilizes. Therefore, it can be deduced that the best offloading gain for the proposed adaptive bias is achieved when 10 UEs are allowed in the system.

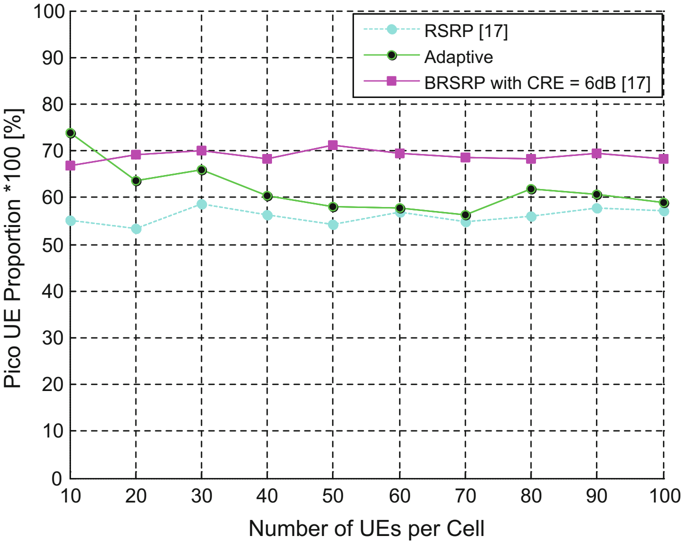


Fig. 2 PeNB UE proportion

Nevertheless, for both RSRP and BRSRP with 6 dB bias, the connection ratio does not show a significant difference in all the number of UE considered.

7.2 SINR CDF

The cumulative distribution functions (CDF) of the SINR of BRSRP with 6 dB bias and the proposed adaptive bias with 4 PeNBs lie slightly above the SINR CDF of RSRP as the reference cell association scheme, as shown in Fig. 3.

The worst affected UE by interference in all the cell association schemes is the cell edge (worst 5%) UE categories according to [2]. Essentially, any offloading due to increase in PeNB cell range will result in SINR performance degradation of the offloaded UEs. This is due to the interference effect suffered by picocell edge UEs from the high transmission power of MeNBs. Consequently, the SINR CDF for the cell edge UEs of the BRSRP with 6 dB was found to be slightly worse than the SINR CDF of the RSRP. Nevertheless, the proposed adaptive bias shows no significant difference with the RSRP. This shows that, with the proposed adaptive bias, the picocell edge UEs will not be in an outage; however, with BRSRP and larger bias value, the picocell edge UEs will be in an outage.

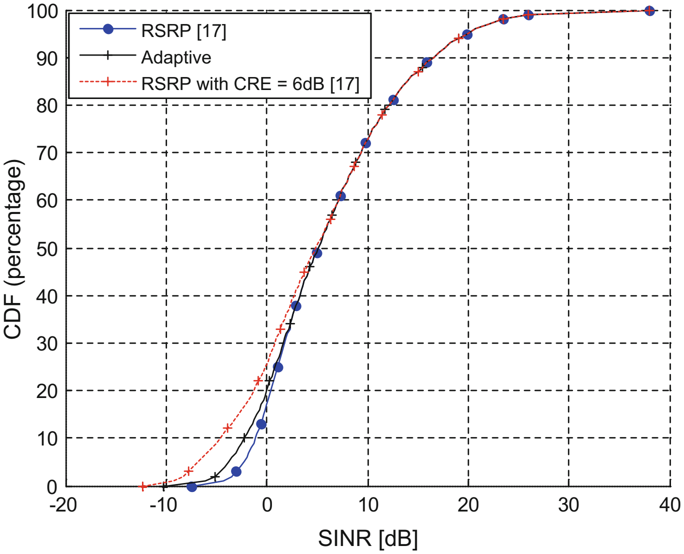


Fig. 3 The CDF of SINR

7.3 *Average UE Throughput as Load Balancing Fairness Measure*

Even though the proposed adaptive bias improves the traffic load balance in the system when compared with RSRP, RSRP with a bias of 6 dB exhibited a more balanced average UE throughput performance between PeNB and MeNB. The difference between the average throughput performance of the PeNB UEs and MeNB UEs is 5.32, 3.62 and 1.6 Mbps for RSRP, proposed adaptive bias and BRSRP with 6 dB, respectively, as shown in Fig. 4.

Hence, BRSRP with 6 dB has the lowest difference in the average UE throughput between the MeNB UEs and PeNB UEs, which shows a more balance in the system's load condition.

7.4 *Throughput Performance*

For all the traffic load considered, the average UE throughput and average macrocell area throughput decrease with the proposed adaptive bias and BRSRP with a bias of 6 dB as depicted in Fig. 5.

This can be attributed to the fact that BRSRP and the proposed adaptive bias essentially offload UE from MeNB to PeNB, and the larger the bias the more the offloading gain. Therefore, large bias tends to overload the PeNB thereby lowering the

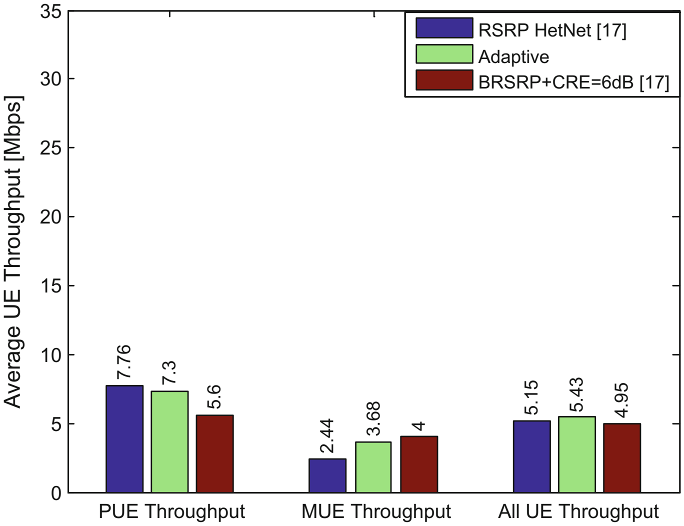


Fig. 4 Average UE throughput for 30 UE per cell

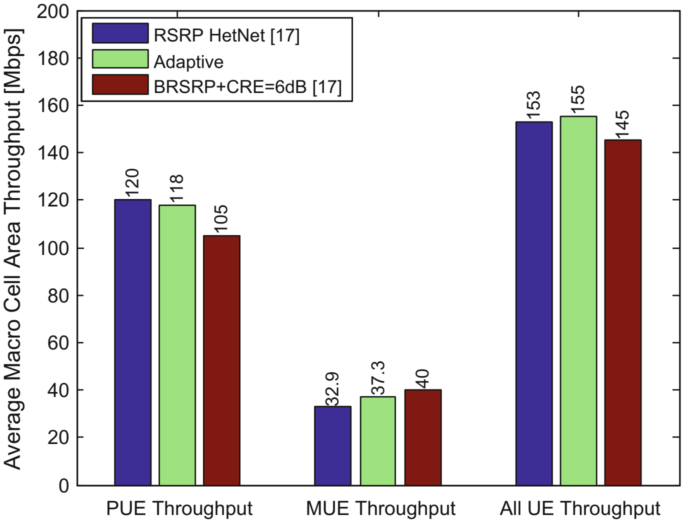


Fig. 5 Average macrocell area throughput for 30 UE per cell

average achievable throughput of the PUE due the round-robin scheduler employed. The round-robin resource allocation makes UEs share the limited resource blocks in the picocell equally. Also as the bias increase, picocell edge UE increase, such UEs are greatly impacted by interference from MeNB which consequently reduce their rate. Conversely, the average UE throughput and average macrocell area throughput increase with the proposed adaptive bias and BRSRP with a bias of 6 dB. This can be attributed to the fact that, as UEs are offloaded to PeNBs from MeNB, fewer UEs are left in the MeNB to share the available resources and such UEs are not affected by interference. Therefore, such UEs achieve higher throughput.

Even though BRSRP with 6 dB bias achieves the best traffic load balance, it has the worst average UE throughput and average macrocell area throughput performance for the traffic load considered. This can be attributed to poor SINR performance with 6 dB and redundancy introduced to the MeNB due to limited UE allowed in the MeNB as a result of biasing. It can also be observed that the proposed adaptive bias achieved the best average UE throughput and average macrocell area throughput. This is because the proposed adaptive bias was able to dynamically configure an optimized bias that maximizes throughput and energy efficiency. However, the proposed adaptive bias and RSRP achieve a poor traffic load balance. This is due to low offloading of UEs from PeNB to MeNB.

7.5 Energy Efficiency Performance

The proposed adaptive bias achieved the best energy efficiency for all the traffic load simulated as depicted in Fig. 6.

BRSRP with 6 dB achieved the worst energy efficiency due to poor SINR performance which lowers the total achievable throughput. The PeNB have very high energy efficiency compared to MeNB. This can be attributed to the fact the PeNB utilizes a lower amount of power due to its small area coverage and delivers higher throughput due to its lower path loss. On the other hand, the MeNB utilizes a higher amount of power due to large area coverage and delivers lower throughput due to higher path loss. Consequently, the energy efficiency of HetNet is greatly impacted by the high power consumption of the MeNB.

8 Conclusion

HetNet deployment has the potential to improve capacity as well as energy efficiency. However, cell selection based on BRSP with fixed bias limits the energy efficiency in LTE-Advanced HetNet. Therefore, in this chapter, an energy-efficient adaptive bias setting strategy is proposed to dynamically configure bias using energy efficiency as a cell load metric. The energy efficiency was modeled as a ratio of data rate to base station power consumption. Thus, the power consumption is evaluated using

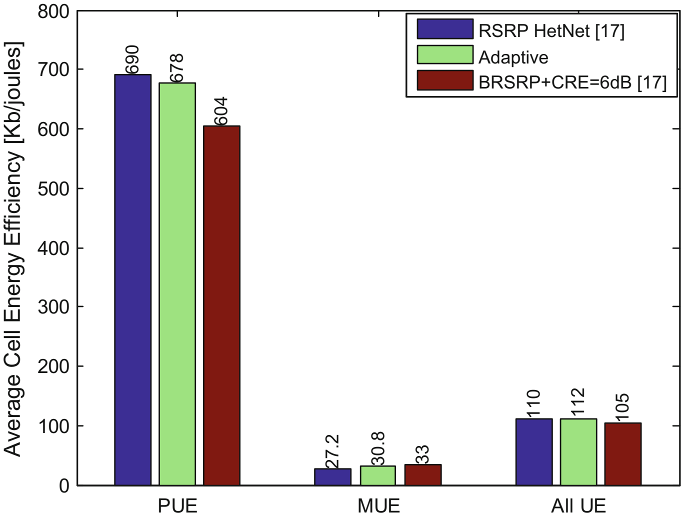


Fig. 6 Average macrocell area energy efficiency

power consumption parameters while the system capacity was modeled based on link adaptation, considering spatial distribution of UEs. From simulation results, it was found that the proposed adaptive model achieves an improved energy efficiency, average UE throughput and system capacity by 6.7, 9.7 and 6.9%, respectively, when compared with BRSRP with a fixed bias of 6 dB as benchmark algorithms. The proposed adaptive bias improves traffic to load balance in the system when compared with RSRP. However, BRSRP with a fixed bias of 6 dB performs better in terms of traffic load balance between PeNB and MeNB.

References

1. Joud MA (2013) Pico cell range expansion toward LTE-advanced wireless heterogeneous. M.Sc. Thesis, Department of information and communication technologies, Universitat Politècnica de Catalunya (UPC)
2. Gadam MA, Ng CK, Nordin NK, Sali A, Hashim F (2016) Hybrid channel gain and access cell association for load balancing in downlink LTE-advanced HetNets. In 6th IEEE international conference on computer and communication engineering (IEEE ICCCE 2016)
3. Khadka SK, Shrestha J, Shakya SR, Lal L (2015) Energy demand analysis of telecom towers of Nepal with strategic scenario development and potential energy cum cost saving with renewable energy technology options. Int J Res Eng Sci (IJRES) 3:01–08
4. Damnjanovic A, Montojo J, Wei Y, Ji T, Luo T, Vajapeyam M, Yoo T, Song O, Malladi D (2011) A survey on 3GPP heterogeneous networks. Wirel Commun, IEEE 18(3):10–21
5. AcharyaJ, Gao, L, Gaur S (2014) Heterogeneous networks in LTE-advanced. Wiley
6. Konishi S (2013) Comprehensive analysis of heterogeneous networks with picocells in LTE-advanced systems. IEICE Trans Commun E96-B(6):1243–1255

7. Humar I, Ge X, Xiang L, Jo M, Chen M, Zhang J (2014) Rethinking energy efficiency models of cellular networks with embodied energy. IEEE Network 25:40–49
8. Guo W, Turyagyenda C, Hamdoun H, Wang S, Loskot P, O’Farrell T (2011) Towards a low energy LTE cellular network: architectures. In 2011 19th European Signal Processing Conference, pp. 879–883
9. Zhou T, Huang Y, Huang W, Li S, Sun Y, Yang L (2014) QoS-aware user association for load balancing in heterogeneous cellular networks. In 2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall), pp 1–5
10. Andrews JG, Singh S, Ye Q, Lin X, Dhillon H (2014) An overview of load balancing in HetNets: old myths and open problems. Wirel Commun, IEEE, 21(2):18–25
11. Gadam M, Ahmed MA, Ng CK, Nordin NK, Sali A, Hashim F (2016) Review of adaptive cell selection techniques in LTE-Advanced Heterogeneous Networks. J Comput Netw Commun 2016
12. Gu W, Li W, Zhang L (2013) Adaptive cell range control in heterogeneous network. In 2013 international conference on wireless communication and signal process, October 2013, pp 1–5
13. Kikuchi K, Otsuka H (2013) Parameter optimization for adaptive control CRE in HetNet, pp 3349–3353
14. Koizumi T, Higuchi K (2013) A simple decentralized cell association method for heterogeneous networks. IEICE Trans Commun 96(6):1358–1366
15. Gu X, Deng X, Li Q, Zhang L, Li W (2014) Capacity analysis and optimization in heterogeneous network with adaptive cell range control. Int J Antennas Propag
16. Davaslioglu K (2015) Energy efficiency and load balancing in next-generation wireless cellular networks. Ph.D. Dissertation, Department of Electrical and Computer Engineering, Faculty of Engineering, University of California, Irvine
17. Gadam MA, Ng CN, Nordin NK, Sali A, Hashim F (2016) Hybrid channel gain prioritized access-aware cell association with interference mitigation in LTE-Advanced HetNets. Int J Commun Syst
18. Danburam AK, Usman AD, Sani SM, Gadam MA (2016) Analysis on energy efficient traffic load balancing in downlink LTE-advance heterogeneous network. In International conference on information and communication technology and its applications (ICTA 2016), pp 191–197
19. Abdulkafi AA, Tiong SK, Chieng D, Ting A, Ghaleb AM, Koh J (2013) Modeling of energy efficiency in heterogeneous network. Eng Technol 6(17):3193–3201
20. Siddique U, Tabassum H, Hossain E, Kim DI (2015) Channel access-aware user association with interference coordination in two-tier downlink cellular networks. IEEE Trans Veh Technol 9545(c):1–1
21. Gadam MA, Maijama’a L, Usman IH (2013) A review on resource allocation techniques in downlink 29 LTE. J Mobile Commun 17–23
22. Khirallah C, Rastovac D, Vukobratovic D, Thompson J (2014) Energy efficient multimedia delivery services over LTE/LTE-A. IEICE Trans Commun 97:1504–1513
23. Network T (2009) Tr 36.814-further advancements for e-utra: physical layer aspects (release 9). 3rd Generation Partnership Project Tech Rep, Tech Rep
24. Wang Y (2010) System level analysis of LTE-advanced: with emphasis on multi-component carrier management. Videnbasen for Aalborg UniversitetVBN, Aalborg UniversitetAalborg University, Det Teknisk-Naturvidenskabelige FakultetThe Faculty of Engineering and Science, Institut for Elektroniske SystemerDepartment of Electronic Systems
25. Access EUTR (2010) Further advancements for E-UTRA physical layer aspects. 3GPP Technical Specification TR, vol 36, p V2
26. Zineb AB, Ayadi M, Tabbane S (2017) An enhanced vertical handover based on fuzzy inference MADM approach for heterogeneous networks. Arab J Sci Eng 1–12
27. Acharya J (2014) Cellular network topology toolbox, pp 1–6

Energy Efficiency of Backhauling Options for Future Heterogeneous Networks

Nasir Faruk, Abubakar Abdulkarim, Nazmat T. Surajudeen-Bakinde and Segun I. Popoola

Abstract Deployment of heterogeneous networks (HetNets) is a veritable solution to the challenges of coverage and capacity in meeting the unprecedented future mobile data traffic. However, the high density of small base stations (SBS) in future HetNets may increase the complexity of backhauling with higher capital expenditure (CAPEX), operating expenditure (OPEX), and energy cost. In this paper, a comprehensive review of various backhauling options for future HetNet is provided. For a HetNet system of seven SBS scenario, the efficiencies of the backhauling technologies are evaluated based on power consumption analysis. The implementation of energy-efficient microwave communication links is also considered based on realistic power consumption. Findings show that massive MIMO (M-MIMO) backhauling system consumes the highest power at maximum load. Power consumption in M-MIMO tends to rise as the number of transmitting antennas increases. In the same vein, power consumption in self-backhauling is relatively high when compared with conventional backhauling systems such as the microwave point-to-point (P2P) and point-to-multipoint (P2MP). On the other hand, total power consumed by satellite hub site, fiber optics, and cloud radio access network (CRAN) technologies are found to be relatively low (271.0571, 96.8083 and 90.1920 W, respectively). Fiber optics, CRAN, satellite hub site, and P2MP backhauling options proved to be more energy efficient in a decreasing order, when coverage and capacity are considered. The con-

N. Faruk (✉)
Department of Telecommunication Science, University of Ilorin, P.M.B 1515, Ilorin, Kwara State, Nigeria
e-mail: faruk.n@unilorin.edu.ng

A. Abdulkarim · N. T. Surajudeen-Bakinde
Department of Electrical and Electronics Engineering, University of Ilorin, P.M.B 1515, Ilorin, Kwara State, Nigeria
e-mail: abdulkarim.a@unilorin.edu.ng

N. T. Surajudeen-Bakinde
e-mail: deenmat@unilorin.edu.ng

S. I. Popoola
Department of Electrical and Information Engineering, Covenant University, P.M.B 1023, Ota, Ogun State, Nigeria
e-mail: segun.popoola@covenantuniversity.edu.ng

T. Herawan et al. (eds.), *Advances on Computational Intelligence in Energy*,
Green Energy and Technology, https://doi.org/10.1007/978-3-319-69889-2_9

tribution of this work will help mobile network operators (MNO) in better decision making toward achieving a sustainable backhauling in future HetNet deployments.

Keywords Backhaul technologies · Small cells · Heterogeneous networks · Energy efficiency

1 Introduction

In this era of Internet of Things (IoT), the unprecedented and continuous growth in the demand for mobile data services by wireless devices and sensor nodes has compelled mobile network operators (MNOs) to seek for how to significantly increase their network capacity while decreasing the cost per bit of data delivered [1]. The traditional cellular architectures cater for large coverage area but they often fail in achieving the expected throughput due to high inter-cell interference, bandwidth, and backhauling air interface and network configuration [2]. Hence, the need for disruption in the network architecture of future mobile communication system so as to meet the ever-increasing mobile data traffic. Meanwhile, most of the methods proposed in the literature seek to increase network capacity and fill coverage gaps at the expense of increased network energy consumption. The bulk of the energy is consumed by the active nodes such as the macrobase stations (MBSs) and cooling systems. Therefore, the cost of energy consumed may introduce new challenges in the bid to finding traditional network energy solutions [3].

Several methods have been proposed to reduce the high power consumption in future mobile networks. For instance, network densification can be achieved through cell shrinking, wherein a large number of low power base stations are deployed to complement existing MBSs. The deployment of these SBS (femto-, micro-, and pico-base stations) have small coverage areas, may be exploited to improve the spectral efficiencies and to achieve low path losses and low power consumption. This low power can be as small as 500 W, which is feasible and sustainable, especially when renewable energy sources are employed. The 4 kW power consumed by the MBS is quite high when compared to that of SBS [3–5]. The resulting network consists of various types of base stations each with different coverage areas, typically referred to as heterogeneous network (HetNet) [1, 2].

Interestingly, an energy-efficient operation can be achieved by the deployment of SBS since they consume less power. However, the high-capacity backhaul connection between the SBS and the mobile core network since more SBS would be required to complement a single high-powered MBS. These SBS-backhaul links may in turn increase the energy consumption of the overall SBS sites beyond what could be supported with low-cost renewable energy solutions. An extensive deployment of SBS will eventually increase the overall energy consumption due to SBS backhauling energy requirements [3]. Therefore, it is of paramount necessity to provide a holistic and comprehensive approach toward achieving a more energy-efficient and sustainable green backhauling for future HetNets.

This paper, therefore, explores and provides a critical and systematic review of the various backhauling options available today for future HetNet. Considering a HetNet system of seven SBS scenario, the efficiencies of the backhauling technologies are evaluated based on power consumption analysis. The implementation of energy-efficient microwave communication links is also considered based on realistic power consumption. The backhauling options covered in this paper include: massive MIMO (M-MIMO), satellite hub site, fiber optics, and cloud radio access network (CRAN) technologies. The contribution of this paper will help mobile network operators (MNO) in better decision making toward achieving a sustainable backhauling in future HetNet deployments. The descriptions of the mathematical notations used in this paper are provided in Table 1.

Table 1 Mathematical notations and their descriptions

Notation	Description
H	Efficiency
$\sum$	Summation
Δ	Delta
α	Alpha
φ	Quotient
P_{PA}	Power of transceiver
P_{RF}	Power amplifier
P_{DSP}	Power of baseband processing
P_{cool}	Power of cooling unit
W	Transmission bandwidth
P_{BB}	Baseband engine
P_{RF}	Small-signal transceiver
n_{PA}	PA efficiency
δ_{feed}	Feeder cable loses
$P_{\max}$	Maximum transmission power
P^{SBS}	Power consumption of the SBS
N_j and N_v	Represent the total number of MBS and SBS, respectively
P_j^{MBS} and P_v^{SBS}	The power consumption of each MBS and SBS
f, f_{a} and f_{b}	The instantaneous load on the HetNet, MBS and SBS, respectively
n	Power amplifier efficiency
P_{t}	Transmit power
M	Number of transmit antennas
P_{c}	Circuit power of the corresponding RF chain

(continued)

Table 1 (continued)

Notation	Description
P_{o}	Determined by the non-transmission power consumption
f_{a}	The load factor representing the number of active users served by MBS network
P_{t}	Transmit power
P_{c}	Power consumption of the transceiver
$P_{\mathrm{c,t}}^{\mathrm{bh}}$, $P_{\mathrm{SBS,t}}^{\mathrm{bh}}$	Power consumption of microwave unit(s) at the wireless hub and SBS, respectively
$P_{\mathrm{tot,t}}^{\mathrm{bh}}$	Total power consumption of microwave radio link (RRL)
$\mathrm{SBS}(C_j^{\mathrm{SBS}})$	Aggregate backhaul capacity of the RRL at the SBS at site $j\cdot$
$N_{\mathrm{k}}^{\mathrm{SBS}}$	Total number of microwave at SBS
P_{lc}, P_{hc}	Power consumption of the microwave antennas associated with low and high-capacity demand, respectively.
$P_{\mathrm{Switch}}^{\mathrm{SBS}}$, $P_{\mathrm{Switch}}^{\mathrm{c}}$	Power consumption of the switches used at SBS and the hub site, respectively
C_j^{SBS} and C_j^{c}	Total backhaul capacity at the SBS and hub site
$N_{\mathrm{k}}^{\mathrm{SBS}}$ and $N_{\mathrm{k}}^{\mathrm{c}}$	Number of microwave links at the SBS and hub site, respectively
$C_{\mathrm{Max\text{ - }Switch,t}}$	Maximum capacity of the switch of type t
$P_{\mathrm{Switch,t}}$	Fixed power consumed by switch type t irrespective of the load
φ_{bh}	Backhaul type (either P2P or P2MP)
N	Number of base station types used in the network
Mi	Total number of base stations of a specific type i
$P_{\mathrm{bh}}^{\mathrm{sat}}$	Entire backhaul power consumption
$P_{\mathrm{hub}}^{\mathrm{sat}}$	The power consumption of the area hub node
M_{bs}	Total number of base stations in the entire area
P_j^{sbs}	Power associated with satellite backhaul operations at the base station site j
C_{th}	Threshold capacity
V_{low} and V_{high}	Power consumption of the very small aperture terminal associated with low and high-capacity demand, respectively

(continued)

Table 1 (continued)

Notation	Description
p_i	Wireless transmission power of the RRH
η	Power amplifier efficiency
P_{RF}	Circuit power consumption for a RRH
P_0 and $P_{\max}$	Power consumption of BBU in idle mode and in full usage mode, respectively
Δp	Slope of the equivalent linear power model
k	Relevant coefficient
X_{cap}	Baseband processing capacity of one BBU in the BBU pool
P_{tx}	Average radiated power
a_i	Power consumption that scales with the transmit power due to RF amplifier and feeder losses
b_i	Models the power consumed independently of the transmit power due to signal processing and site cooling
c_i	Power consumption of the SFP used to transmit over the backhauling fiber
P_{bh}	Power consumed by the backhaul
P_i	Power consumption of a base station of type i
P_{dl}	Power consumed by one downlink interface in the aggregation switch used to receive the backhauled traffic
N_{ul} and P_{ul}	Total number of uplink interfaces and the power consumption of one uplink interface
P_s	Power consumed by a switch
$P_{\max}$	Maximum power consumption of the switch when all the downlink interfaces are in use
$Ag_{\max}$	Maximum amount of traffic a switch can handle

2 Heterogeneous Networks (HetNets)

HetNet architecture involves the interconnection of different base station types (macro-, micro-, pico-, and femto-base stations) with varying coverage footprints and power requirements. The primary aim of introducing HetNet is to increase the network capacity while ensuring higher Quality of Service (QoS) and bandwidth usage in an energy-efficient manner. Table 2 presents the different base stations, their coverage distances, power radiated, and consumed with the various backhauling technologies for connecting the base stations and the core networks. Figure 1

Table 2 Various base station types [3–5]

Base station type	Coverage distance (m)	Radiated power (kW)		Consumed power (kW)	Backhaul type
		Indoor	Outdoor		
Macrocell	<35,000	N/A	0.005–0.040	1–5	Fiber or microwave
Microcell	<2000	N/A	0.0005–0.002	0.1–0.3	Microwave
Picocell	<200,000	0.0001	0.00025–0.002	0.009–0.015	Microwave or fiber or DSL
Femtocell	10,000–15,000	>0.0001	N/A	6–14	Fiber/DSL
RRH-cell	<2000	N/A	5–20	550–760	Fiber
Relay-cell	<2000	<0.001	0.25–7	10–120	via DeNB

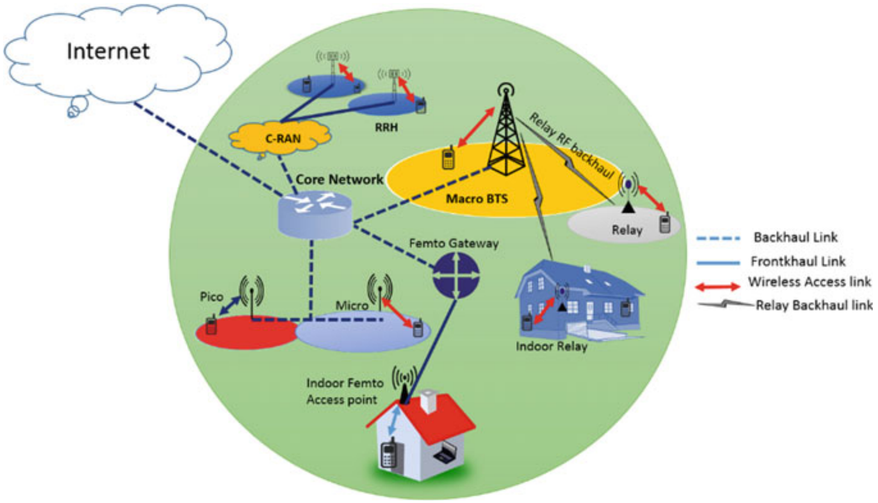


Fig. 1 Typical HetNet deployment scenario

is a schematic diagram which shows how the SBS are connected to the MBS. The figure also shows the different links vis-à-vis backhaul, fronthaul, wireless access, and relay backhaul links. This is a typical scenario of how the core network, which accesses the Internet, can be connected to the various SBS.

3 Backhaul Technology Options

HetNet is described as the combination of different types of base stations with various building components, form factor, coverage area, operation, and cost of procurements. The coverage footprints of the SBS are called small cells while those of

Table 3 Copper cables available for backhaul [7]

Cable	Standard	Types	Maximum downstream (Gbps)	Maximum upstream (Gbps)
Data over cable service interface specification (DOCSIS)	ITU DOCSIS 1.0	1.0	0.01	0.01
		1.1	0.01	0.01
		2.0	0.03	0.03
		3.0	0.2	0.2
		3.1	1–2	1–2
		3.1 full duplex	10	10
Digital subscriber line (DSL)	ITU G992	SADSL	0.012	0.0018
		ADSL 2	0.012	0.0035
		ADSL 2+	0.024	0.0013
		ADSL 2 + M		0.0033
		VDSL	0.055	0.003
		VDSL 2+	0.055	0.003
		G. FAST	1	1
Fiber optics cable	ITU G.707, G.783, G.784, and G.803	Single mode	>10	>10
		Multimode		
		Plastic		

MBS are referred to as macrocells [6]. Backhauling, which is of two types, wired and wireless, is the connection between the SBS and the MBS. It also links the MBS to the core network.

3.1 Wired Backhauling

Wired backhauling is a type of backhauling that uses cables to connect nodes together. It is the most suitable solution for dense urban deployment due to its high reliability, high data rate, and high availability. However, these advantages come with an extra cost [6]. The two types of wired backhauling are copper and optical fiber cables. Table 3 presents the different types of copper cables available for backhauling with their corresponding maxima downstream and upstream data rates.

3.1.1 Wired Backhauling

Copper cables are in different kinds; they can be the E1/T1 or the Digital Subscriber Line (DSL). E1/T1 cables are the ancient backhaul medium between base transceiver stations (BTS) and base station controller (BSC). They operate using time division multiplexing techniques known as plesiochronous digital hierarchy (PDH). There are two standard PDH delivering different bit rates: the T-carriers (T1–T4) and E-carriers (E1–E5). Leased T1/E1 copper lines dominate the backhaul solutions in the 1G and 2G networks as they provide suitable support for voice traffic, deterministic QoS, low latency, and low delay variations (jitter). However, the increase in the required backhaul capacity with the later generations of mobile networks has caused a huge growth in the number of T1/E1 connections and cost. As a result, leased T1/E1 is not a suitable backhaul option for future networks [8]. DSL over existing copper infrastructure is a good backhaul option for short distances. This is mostly used for indoor backhauling and for broadband applications [8]. However, its backhaul capacity depends on the technology and the distance from the exchange. The continuous improvements in DSL technology led to the evolution of ADSL2+, GSHDSL.DOC, VDSL2, VDSL2+, etc. making them viable for mobile backhaul in future networks [9].

3.1.2 Optical Fiber Cable

An optical fiber cable comprises of a transparent core and a cladding with rays of light kept in the core by total internal refraction, while information is transmitted as rays of light from one point to another within the cable. Fiber, which is widely used for a shorter length, can support multiple propagation paths known as multimode Fiber (MMF). A single-mode fiber (SMF) has a single path, and it provides high-speed connectivity, making it suitable for future networks because any backhaul capacity can be served [10]. On the other hand, a huge capital expenditure (CAPEX) of up to $100,000 per km is incurred by the MNOs. It also requires permits, trenching, boring, and ducting leading to high deployment duration [11]. Direct fiber could be used to connect SBS or a hybrid or fiber and any other backhauling technologies such as microwave. Figure 2 shows a typical fiber optics backhaul architecture.

3.1.3 Cloud Radio Access Network (C-RAN)

Cloud RAN (C-RAN) is a network architecture which maximizes baseband unit (BBU) utilization by pooling network resources into a centralized, virtual pool called the BBU/DU pool/hotel. These resources are then shared between the remote radio heads (RRH). This architecture was first referred to as wireless network cloud by IBM, but its name was later changed to C-RAN where the letter “C” could mean “Cloud” or “Centralized.” However, in massive deployment, its high-speed fronthaul connection makes it less cost-effective. Also, the latency requirements for the BBU, real-time requirement for the operating system, clock synchronization, and capacity requirement are its major limitations [12].

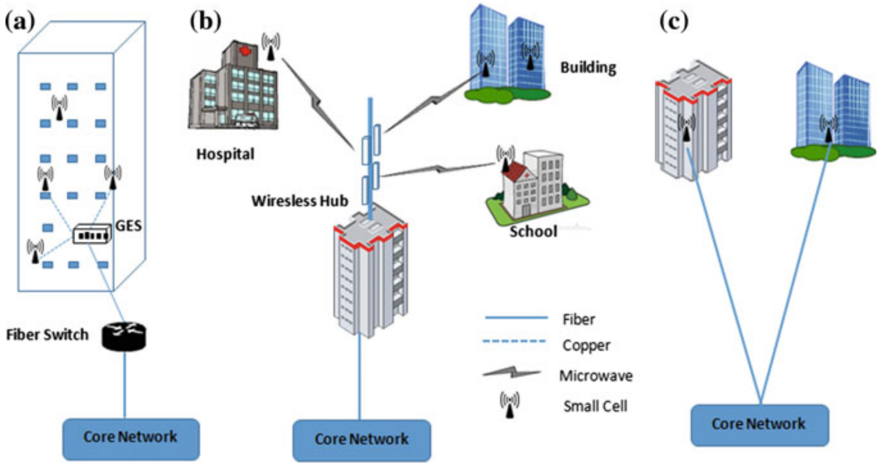


Fig. 2 Fiber optic backhaul architecture **a** fiber–copper architecture, **b** fiber–microwave, **c** direct fiber

3.2 Wireless Backhauling

Although wireless backhauling, when compared to the wired counterpart, has lesser capacity and reliability, it becomes useful in areas where wired backhauls are difficult to deploy in terms of location or cost [10]. Wireless backhauls are of two major types: line of sight (LoS) and non-line of sight (NLoS). LoS backhaul requires that both communicating antennas have clear visibility to each other for communication to be made, while the NLoS does not require clear visibility to establish information exchange [13]. Wireless backhaul includes microwave, satellite, TV white space, self-backhauling, etc. Parameters such as communication channel, spectrum efficiency, cost of the backhauling, and backhaul capacity among others, for the two types of wireless backhauls, are compared in Table 4.

3.2.1 Self-backhauling System

Self-backhauling system uses the existing cellular network infrastructures to provide backhaul traffic. It involves the use of existing macro-RAN to provide backhaul to SBS as shown in Fig. 4. Its merits include flexible deployment in dense SBS scenarios using NLoS requirement, cost saving through reusing of macrocell infrastructures such as spectrum, and leveraging the existing macrosite management and control (such as security management, resource management, and fault management) [14].

Table 4 Comparison of LoS and NLoS wireless backhauls

Parameters	LoS	NLoS
Communication channel	Requires a clear unobstructed visibility between antenna	Requires only to be placed within range of the backhaul radio unit
Multipath fading and signal interference	A highly directional beam transmits the data in a straight line with little or no fading or multipath radio interference	NLoS systems using OFDM present a level of tolerance to multipath fading and other wireless channel impairments not possible with LoS systems
Spectrum efficiency	This is a highly efficient use of spectrum, as multiple microwave transceivers can function within a close proximity to each other and reuse the frequency band for transmitting separate data streams	It has limited spectrum efficiency; hence, frequency planning would have to be planned to avoid too much interference
Areas of application	Mainly used for high-bandwidth applications for outdoor small cell deployments rather than indoor cells	It can provide coverage for various types of small cell setups with proper design
Tolerance to environment changes	Pole tilting or swaying are problems for deploying small cell backhaul on structures like utility, lighting, and traffic poles. Also, environments with many trees, such as park, could block LoS making them impractical location for small cells backhauled through LoS technology	A single NLoS base station can provide coverage for multiple small cells within the coverage area without the need for an unobstructed path between the transceivers, this makes the technology highly helpful for future planning and upgrades
Cost of the backhauling	The cost of the backhaul rises quickly when compared to NLoS in cases of huge deployment especially if daisy chains are involved as significant number of skilled technicians are usually required for antenna alignment for LoS technologies	NLoS technologies are easy to deploy as they are mostly "plug and play" and can be set up in a short time with reduced labor costs
Backhaul capacity	LoS technologies have no upper limit lower than that offered by the network	NLoS technology has an upper limit to the amount of data that each coverage area can backhaul

3.2.2 Microwave and Millimeter Wave Backhaul

Microwave and millimeter wave backhauls have dominated mobile networks for years, especially in challenging geographical areas. They are easy to deploy and they support a distance of up to 50 km. New innovations (such as adaptive coding and modulation (ACM), compression accelerators) are being made so as to increase the bandwidth on both links, hence making them operate in both LoS and NLoS (but microwave is mostly used in LoS backhaul) using point-to-point (P2P) or point-to-multipoint (P2MP) topologies [15]. The setback to microwave is that operating in the licensed band increases the CAPEX while operating in the unlicensed band and E-band takes less CAPEX (which is lightly licensed and relatively easy permission, with frequency ranging from 70 to 80 GHz). However, it suffers attenuation from atmospheric effects and other signals. In addition, it has been established that the operating frequency and bandwidth of E-band are inversely proportional to coverage distance. This makes the E-band mostly used in small cell backhauling and short distance links. Also, when the gap between coverage areas is large and unpopulated, it is not economically feasible to add towers to bridge the distance.

3.2.3 Satellite Backhauling

This type of backhaul is preferred in remote areas where other backhaul solutions are uneconomical to deploy [6]. Backhaul over satellite was demonstrated in 2012 to deliver a speed of 10 Mbps at download speed and 7 Kbps upload speed [7]. In many areas where terrestrial infrastructure is limited satellite becomes the primary option for transporting voice and data services. Satellite communication is not affected by topological variations such as distance, terrain or LoS, making it the most viable option for backhauling. Although satellite might be more expensive than other wireless solutions, it is scalable, highly reliable and can be deployed quickly, even under the most challenging geographical and climatic conditions. Also, the invention of small cell technology has motivated some MNOs to consider the use of carrier-class satellite backhaul as a viable option to more traditional backhaul types. Compared to macrocell solutions, these small cell networks are less expensive; when coupled with a low-cost satellite MODEM/router. Moreover, it enables MNOs to expand coverage into rural areas quickly and economically or operate smaller networks on board ships, in aircraft, or in remote mining areas [16]. In future network, the satellite will contribute in areas like coverage extension, enhanced spectral utilization integrated signaling systems, and providing resilience [17–19].

3.2.4 Wireless Fidelity (Wi-Fi) Network Technology

Wi-Fi has been modified from its initial indoor usage to feasible backhaul connectivity of up to 38 km. It is based on the IEEE 802.11 standard using 5.8 and 2.4 GHz unlicensed band. This feature, together with its low cost and flexible deployment, makes it a good alternative for microwave backhaul. However, it has design limitations relevant to the achieved throughput, distance coverage, packet overhead, timing, and synchronization [8, 9].

3.2.5 Massive Multiple-Input Multiple-Output (M-MIMO) Backhauling

Massive MIMO system involves scaling up the antenna system of MIMO whereby hundreds of MBS antennas serve thousands of Mobile terminals in the same time–frequency resource. With aggressive spatial multiplexing and array gain, massive MIMO could achieve capacity increase and energy efficiency improvement [20, 21]. However, M-MIMO provides limited throughput to mobile end users in high path loss channels, but this could be improved with the introduction of small cells. This results in M-MIMO-based HetNet [22]. Other merits of M-MIMO include energy and cost efficient components, reduced air interface latency, and in-band wireless backhaul [23].

3.2.6 Worldwide Interoperability for Microwave Access (WiMAX)

WiMAX is a broadband broadcasting technology designed using IEEE 802.16 standards, and researches are done for outdoor purposes [8, 24]. It uses orthogonal frequency division multiplexing (OFDM) to provide higher throughput, guaranteed QoS, and larger coverage area than Wi-Fi. Theoretically, it can deliver data rates from 75 Mbps in a single channel and up to 350 Mbps via multiple channels). It can work in both unlicensed (typically 2.4 and 5.8 GHz) and licensed (typically 700 MHz, 2.3, 2.5, and 3.5 GHz) bands. Hence, the technology reduces the CAPEX more than microwave as it offers a cheaper license spectrum as well as unlicensed option. It uses the IEEE 802.16-2004 standard for backhauling in fixed connectivity applications, P2P, P2MP, and mesh topology. It also supports high-throughput data aggregation [6]. There are two main sets of WiMAX: the fixed WiMAX and the IEEE 802.6e-2005 [24].

The fixed WiMAX was originally designed using IEEE 802.16-2001 standard for fixed wireless broadband air interface with LoS and P2MP applications only but was modified in subsequent standards up to 802.16d-2014 to target NLOS and add WiMAX system profiles and Errata for 2–11 GHz. The 802.16e-2015 is an amendment of the fixed WiMAX for mobile wireless broadband providing up to vehicular speeds in licensed bands from 2 to 8 GHz. It also enables roaming for portable users (laptops, tablets, etc.) within and between service areas [25].

4 Performance Evaluation of Different Backhauling Options

Minimum delay and latency, low cost and time to deploy, high-energy efficiency, and availability of bandwidth for future improvements are the vital backhaul requirements to sustain a network for future HetNet. These requirements vary with the available backhaul options. Hence, there is a need to provide a critical and thorough comparison among the available backhauling options as shown in Tables 5 and 6.

5 System Concept

The system concept introduced in this paper, as illustrated in Figs. 2, 3, 4, 5, 6, and 7, consists of a conventional fixed microwave backhauling that is configured in P2P and P2MP topology, the self-backhauling, M-MIMO, satellite, fiber, and C-RAN. These are used to backhaul seven (7) SBS. For the microwave backhauls, we assume LOS clearance between the microwave units (wireless hubs) and the SBS due to operation in higher frequency bands. However, this LoS requirement can be relaxed by using alternative multihop topologies (e.g., mesh) to route the fixed wireless links around shadowing objects (e.g., buildings, mountains, etc.). For C-RAN, the SBS/RRHs are connected to the BBU. Then, a fronthaul fiber is used to link the BBU to the core network. In the case of fiber, we consider direct fiber connections to each SBS. The satellite system consists of VSAT hubs connecting remote SBS. The M-MIMO and self-backhaul require umbrella macrocells since the macroradio access link is used to backhaul the small cells. We assume the SBS to be distributed within the MBS cell coverage area. The MBS users (MBS UE) and SBS user (SBS UE) are randomly distributed within the MBS and SBS coverage areas, respectively, while the SBS are uniformly distributed. The MBS radio access link is used to transmit the backhaul traffic between the SBS and MBS. However, due to the capacity requirements of the traffic aggregates from multiple SBS within the coverage area of the MBS, all other UE traffics, both from the MBS and SBS, are routed to the core network via fiber links. For fairness, in all the scenarios, we assumed that all the SBS transmits on the same power level. Therefore, under normal condition, their power consumption and the sizes of their coverage footprint are expected to be the same.

6 Backhaul Power Consumption Models

6.1 Power Consumption in MBS

The power consumption of the MBS (in watts) is given in Eq. (1) [3]:

Table 5 Performance evaluation of fiber cable, copper cables, self-backhauling, and microwave and millimeter wave backhauling options

Parameters	Fiber cable	Copper cables	Self-backhauling	Microwave and millimeter wave
Cost	Huge CAPEX because it requires permits, trenching, boring and ducting	Low if already existing but the cost is high for new deployment or leasing	Low as existing cellular network infrastructures are used	Operating in the licensed band cost more CAPEX while operating in the unlicensed band and E-band requires lower cost
Energy efficiency	Highest	High	Moderate	Moderate but can be improved using energy-saving devices
Quality of Service (QoS)	Excellent	Guaranteed	Depends on the presence of a regulatory framework for sharing of network resources	Guaranteed
Delay and latency	Low	Low	Low	Low
Network capacity	Unlimited capacity	High with recent brands (10 Gbps)	Limited but can be improved by adding more carriers	High (2–170 Mbps)
Available bandwidth for future improvement	High	Not imminent as capacity depends on number of cables	Medium with modifications	High available spectrum especially in the V and E-band
Suitability for heterogeneous network	Aggregation and core	Indoor small sites and for low traffic MBS backhauling	Small cell backhauling	MBS and SBS backhauling
Duration of deployment	Months and possibly years	Months and possibly years	Days	Weeks
License required	No, but permit required for laying cables	No but permit required for laying cables	No new licensing is required as same licensing for RAN	Licensed for microwave but light licensed/unlicensed for V-band

(continued)

Table 5 (continued)

Parameters	Fiber cable	Copper cables	Self-backhauling	Microwave and millimeter wave
Synchronization and timing	Available	Available	Not available	Available
Power consumption	Least	Least	Low	Moderate
Ease of deployment	Difficult	Difficult	Easy	Easy

Table 6 Performance evaluation of Wi-Fi, M-MIMO, satellite, and WiMAX backhaul options

Parameters	Wi-Fi	M-MIMO	Satellite	WiMAX
Cost	Low	Low	Small cell satellite requires low CAPEX	Low but standardization is expected to decrease the CAPEX
Energy efficiency	Moderate as it has low power, low consumption but low throughput also	Better spectral efficiency and energy efficiency	Energy-saving mechanism such as split architecture can improve the energy efficiency	Moderate but better than Wi-Fi as it offers greater throughput
QoS	Not guaranteed	Guaranteed	Suffers propagation delay	Guaranteed in P2MP model
Delay and latency	Low	Low	High	Low
Network capacity	High (11, 54 or 600 Mbps)	High as it multiplies the capacity without requesting for more spectrum	Medium (384 Kbps to 4.81 Mbps)	High (75–359 Mbps)
Available bandwidth for future improvement	High	Sufficient bandwidth using same frequency resources	Bandwidth depends on some parameters such as size of antenna, etc.	Higher than Wi-Fi
Suitability for heterogeneous network	Small cell backhauling	MBS and SBS backhauling	Geographically challenged areas and low density area backhauling	Outdoor and indoor Small cell

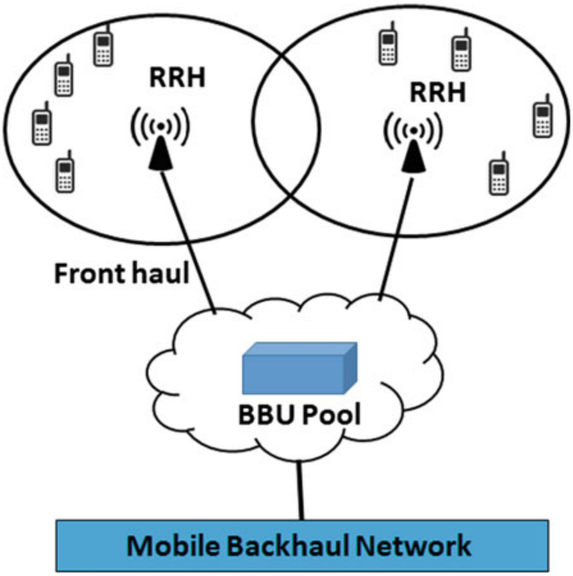


Fig. 3 Architecture of C-RAN HetNet

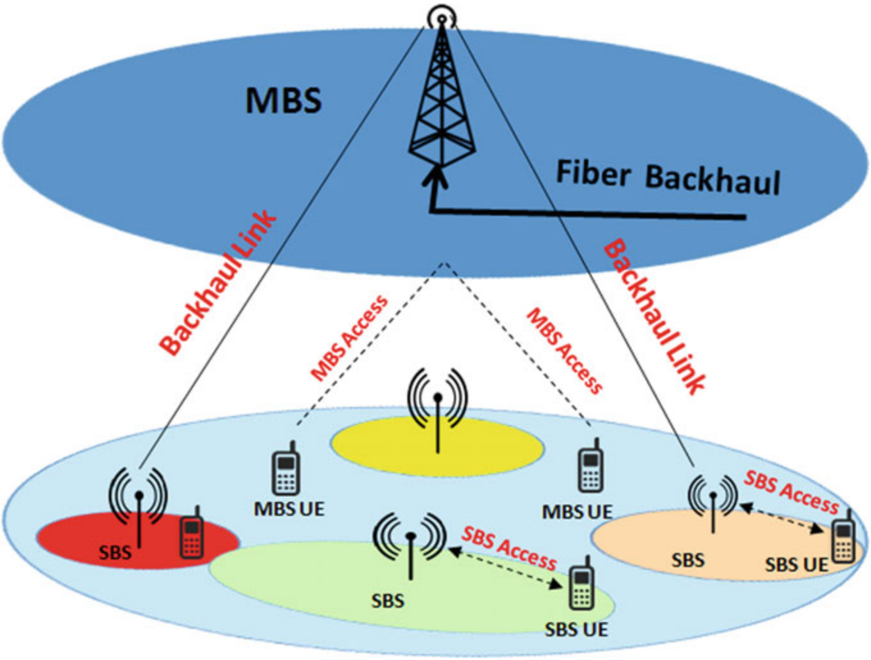


Fig. 4 Architecture of self-backhauling network

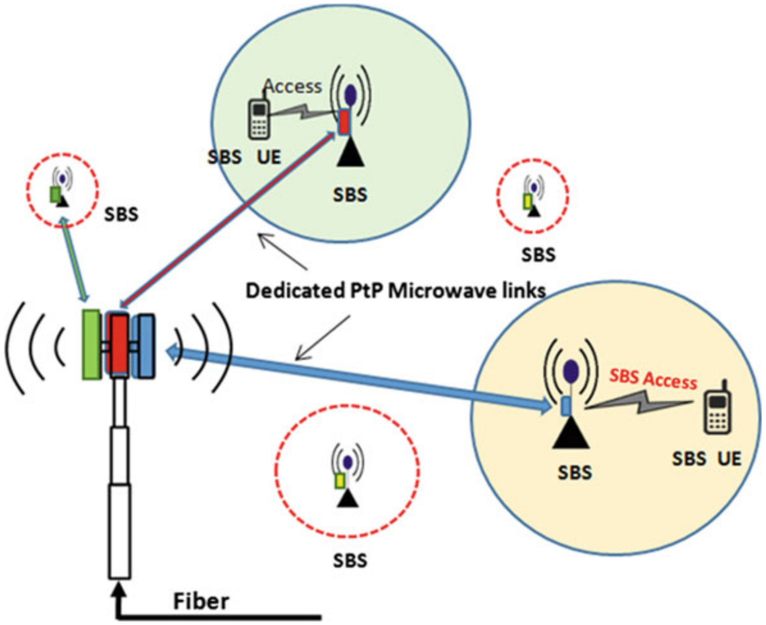


Fig. 5 Architecture of microwave and millimeter wave backhaul

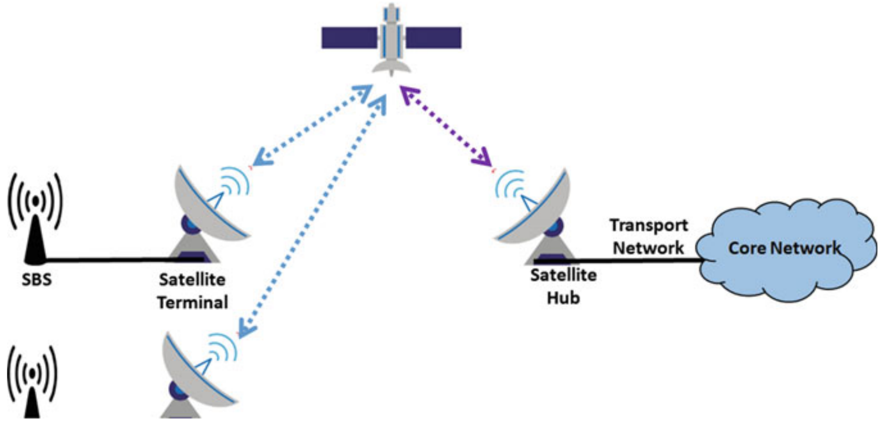


Fig. 6 Satellite backhaul architecture

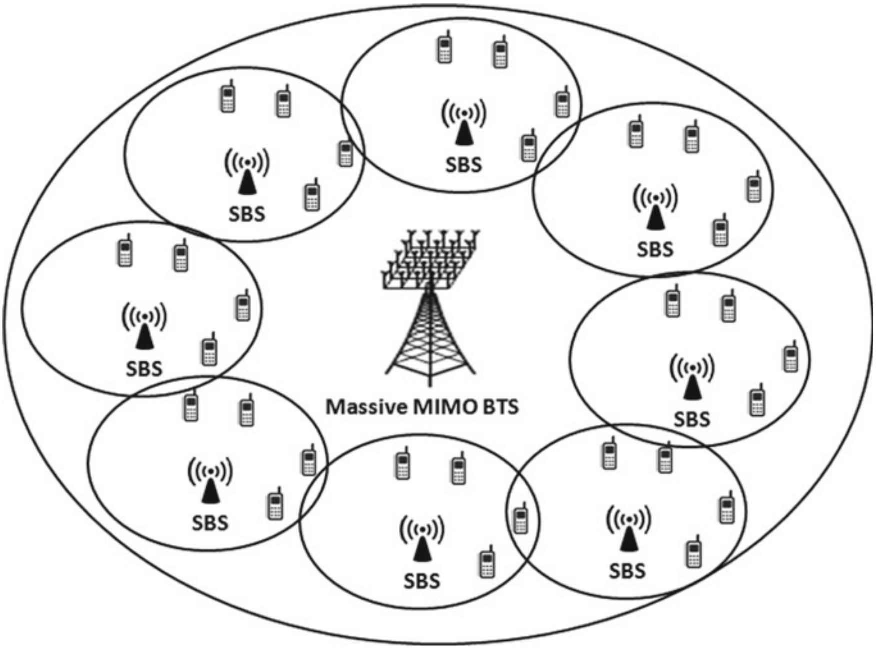


Fig. 7 Architecture of M-MIMO HetNet

$$P^{\text{MBS}}(f_{\text{a}}) = n_{\text{sector}} \times \left(f_{\text{a}} \times \left(n_{\text{TX}} X \left(P_{\text{PA}} + P_{\frac{\text{TRX}}{P_{\text{RF}}}} \right) + P_{\text{DSP}} \right) + P_{\text{rec}} \right) + \sum_{i=1}^{n} P_i^{\text{link}} + P_{\text{cool}} \quad (1)$$

6.2 Power Consumption in SBS

The power consumption of the SBS P^{SBS} (in watts) is given in Eq. (2) [3]:

$$P^{\text{SBS}}(f_{\text{b}}) = P_{\text{rec}} + f_{\text{b}} \times (P_{\text{PA}} + P_{\text{DSP}} + P_{\text{rec}}) + P_{\text{cool}} \quad (2)$$

6.3 Self-backhauling

The power consumption for the self-backhaul is given in Eq. (3) [3]:

$$P_{\text{selfB}}(f_{\text{a}}) = \sum_{j=Q}^{r=1} N_j P_j^{\text{MBS}} f_{\text{a}} \quad (3)$$

6.4 M-MIMO Backhauling

The power consumption of M-MIMO MBS can be modeled adopting a linear BS power consumption model as shown in Eq. (4).

$$P^{\mathrm{MB}}(f_{\mathrm{a}}) = f_{\mathrm{a}} \times (1/n_{\mathrm{PA}} \times P_{\mathrm{t}} + M \times P_{\mathrm{c}}) + P_{\mathrm{DSP}} + P_{\mathrm{rec}} + P_{\mathrm{cool}} \tag{4}$$

6.5 Microwave Backhauling

Given SBS $\left(C_j^{\mathrm{SBS}}\right)$ be the aggregate backhaul capacity of the microwave links at the SBS at the site j and $N_{\mathrm{k}}^{\mathrm{SBS}}$ the total number of microwave at SBS, then the power consumption of the microwave at the central point is given as [3]:

$$P_{\mathrm{SBS,t}}^{\mathrm{bh}} = P_{\mathrm{SBS}}^{\mathrm{agg}}\left(C_j^{\mathrm{SBS}}\right) + P_{\mathrm{switch}}^{\mathrm{SBS}}\left(N_{\mathrm{k}}^{\mathrm{SBS}},\ C_j^{\mathrm{SBS}}\right) \tag{5}$$

$$P_{\mathrm{c},t}^{bh} = P_{\mathrm{c}}^{\mathrm{agg}}(C^{\mathrm{c}}) + P_{\mathrm{switch}}^{\mathrm{c}}\left(N_k^{\mathrm{c}}, C^{\mathrm{c}}\right) \tag{6}$$

If C_{th} is assumed to be 500 Mbps, therefore, generalize the total backhaul consumption for the microwave (i.e., P2P and P2MP) network topology as:

$$P_{\mathrm{tot,t}}^{\mathrm{bh}} = \begin{cases} P_{\mathrm{c,t}}^{\mathrm{bh}} + N_{\mathrm{k}} P_{\mathrm{SBS,t}}^{\mathrm{bh}} & t \in \mathrm{P2MP} \\ \left(P_{\mathrm{c,t}}^{\mathrm{bh}} + P_{\mathrm{SBS,t}}^{\mathrm{bh}}\right) N_{\mathrm{k}} & t \in \mathrm{P2P} \end{cases} \tag{7}$$

6.6 Satellite Backhaul Power Model

The power consumption of a heterogeneous satellite VSAT network including the backhaul part can be written as:

$$P_{\mathrm{tot}}^{\mathrm{sat}} = \sum_{i=1}^{N} M_i P_i + P_{\mathrm{hub}}^{\mathrm{sat}} + P_{\mathrm{switch}} + P_{\mathrm{sbs}}^{\mathrm{sat}} \tag{8}$$

$$P_{\mathrm{bh}}^{\mathrm{sat}} = P_{\mathrm{hub}}^{\mathrm{sat}} + \sum_{j=1}^{M_{\mathrm{bs}}} P_j^{\mathrm{sbs}} \tag{9}$$

6.7 C-RAN Backhauling

Regarding the total power consumption in C-RAN where power consumed by RRHs as well as that by the BBU pool is concerned. It can be expressed as follows:

$$P_{\mathrm{total}} = \sum_{i\mathrm{EN}} P_{\mathrm{R}} + P_{\mathrm{B}}, \tag{10}$$

As for the power consumption at RRH, P_{R} can be expressed as:

$$P_{\mathrm{R}} = \frac{x_i . P_i}{\eta} + P_{\mathrm{RF}}, \tag{11}$$

Thus, P_{B} can be expressed as follows:

$$P_{\mathrm{B}} = m.\big(P_0 + \Delta_{\mathrm{p}} \cdot P\mathrm{max} \cdot y\big), \tag{12}$$

where m denotes the number of active BBUs, P_0, and P_{max} are the power consumption of BBU in idle mode and in full usage mode, respectively. Besides, Δp is the slope of the equivalent linear power model which depends on the specific server adopted, and y denotes the average utilization of each active BBU. Furthermore, the number of active BBUs and the average utilization of each active BBU are related to the load of each RRH. Based on this fact, parameters m and y can be expressed as follow:

$$m = \left[\frac{\sum_{i\mathrm{EN}} k.x_i}{X_{\mathrm{cap}}} \right] \quad y = \frac{\sum_{i\mathrm{EN}} k \cdot x_i}{m \cdot X_{\mathrm{cap}}} \tag{13}$$

where k is a relevant coefficient, X_{cap} denotes the baseband processing capacity of one BBU in the BBU pool. In addition, load balancing is considered in calculating the average utilization of each active BBU. Also, P_{B} can be further expressed as follows:

$$P_{\mathrm{B}} = \left[\frac{\sum_{i\mathrm{EN}} k \cdot x_i}{X_{\mathrm{cap}}} \right] P_0 + \frac{\Delta_p . P\mathrm{max}}{X_{\mathrm{cap}}} \sum_{i\mathrm{EN}} k \cdot x_i \tag{14}$$

In summary, the total power consumption in C-RAN, where both wireless transmission power and baseband processing power are concerned, can be written as follows:

$$p_{\mathrm{total}} = m \cdot P_0 + \sum_{i\mathrm{EN}} (a \cdot p_i \cdot x_i + b \cdot x_i + P_{\mathrm{RF}}) \tag{15}$$

where $a = 1 = \eta$, $b = \Delta p \cdot P_{\mathrm{max}} \cdot k/X_{\mathrm{cap}}$.

6.8 Fiber Optic Backhauling

For the fiber-based case, the total power consumption of a heterogeneous mobile radio network including the mobile backhaul can be written as:

$$P_{\text{tot}}^{\text{FIB}} = \sum_{i=1}^{m} N_i P_i + P_{\text{bh}}^{\text{FIB}} \tag{16}$$

where according to the parameters defined below:

$$P_i = a_i P_{\text{tx}} + b_i + c_i \tag{17}$$

$$P_{\text{bh}} = \left[\frac{1}{\max_{\text{dl}}} \left(\sum_{i=1}^{m} N_i \right) \right] p_{\text{s}} + \left(\sum_{i=1}^{m} N_i \right) P_{\text{dl}} + N_{\text{ul}} P_{\text{ul}}$$

7 Simulation Parameters and Energy Efficiency Metrics

The energy consumption metric is used to quantify the energy consumption of HetNets in order to characterized and measure our findings. This is to compare the various HetNets system under investigation and to evaluate their efficiency and identify where backhaul solution can be deployed while highlighting the trade-off for such deployment. Our metrics were defined energy consumption per some entities which include coverage area (w/m^2), capacity (w/bits), and per load factor. The two major energy standard metrics are the energy consumption ratio (ECR) and the telecommunications energy efficiency ratio (TEER).

The ECR metric is defined as the ratio of the peak power (measured in Watts) to the peak data throughput rate in bits per second and thus can be described as energy consumed per bit of information transported express in joules per bit. This can be mathematically express as:

$$\text{ECR} = \frac{\text{POWER}}{\text{DATA RATE}} = \frac{\text{Watt}}{\text{Bps}} = \frac{\text{joul}}{\text{bit}} \tag{18}$$

The TEER metric is a generic metric which is defined the ratio of useful work done to the total power consumed; hence, we defined our TEER as coverage area per consumed power which is mathematically expressed as:

$$\text{TEER} = \frac{\text{POWER}}{\text{COVERAGE}} = \frac{\text{Watt}}{\text{Area}} = \frac{\text{joul}}{\text{m}^2} \tag{19}$$

8 Results and Discussion

The power consumption of different backhaul systems is given by Eqs. (10)–(19). In this analysis, a HetNet system with 7 SBS was considered. Specific implementation of the energy-efficient microwave links was also considered based on realistic power consumption. This typically ranges from 25 to 50 W. We also considered the load-varying power consumption of the microwave units. In Fig. 8, the power consumption profiles (i) M-MIMO, (ii) self-backhauling, (iii) P2P wireless backhaul, (iv) P2MP wireless backhaul, (v) satellite backhauling, (vi) CRAN, and (vii) fiber are shown. The HetNet system comprises of seven (7) small base stations (SBS). The power consumption of the SBS was kept low (<200 W) as possible, even though, the load-dependent components only contributed 39.5 W of the total power consumption, while the non-load dependent amounted to 160 W and this is about 80% of the total power which is mainly attributed to cooling. The MIMO backhaul system consumes the highest power of 3.6341 kW at maximum load. In this analysis, we used 48 transmitting antennas ($M = 48$). This power budget is rather conservative as 48 transmitting antennas were used since no specific figure is set that constitutes M-MIMO setup. However, Huawei, ZTE, and Facebook used as many as 96–128 antennas to demonstrate M-MIMO systems. Therefore, the power requirement would be high. For the self-backhauling, the transmitter power levels for the MBS is assumed to be 20 W and the power consumption of the DSP, transceiver, rectifier, and backhaul, respectively, are 29.5, 13, 180, and 35 W [3]. Then, the load-dependent power consumption for a 3-sector LTE system with 10 MHz bandwidth and 2 × 2 MIMO system for the MBS gave the highest value at peak load as 2.7693 kW.

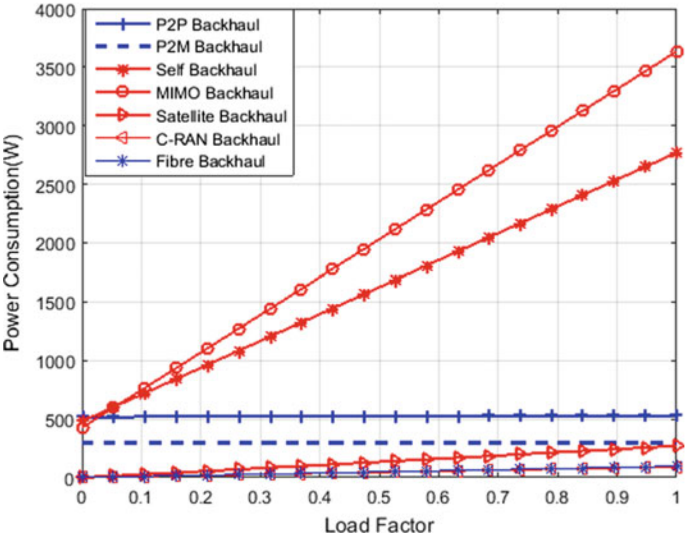


Fig. 8 Power consumption profile of different backhaul systems

The power consumption of the microwave P2P backhaul for both at the hub site and SBS location are 528.6 and 216.44 W, respectively. The backhaul power was computed for the P2MP HetNet at both hub site and SBS as 302.0571 and 216.4400 W, respectively. In this analysis, a microwave unit that supports up to seven small cells was considered. Furthermore, the power consumption of the switch at the hub site and SBS were observed to be 53 and 37 W, respectively. As shown in Fig. 8, the load-dependent component of the power is negligible. The P2MP system consumed less power as few backhaul units are needed to support many cells. In this situation, one microwave unit was used to backhaul seven SBS. On the other hand, the satellite, fiber, and C-RAN are at the lower part of the curve with fiber optics backhaul consuming the least. Moreover, the load-dependent component of the power is negligible, and little effects were observed. Almost no load impact was observed for C-RAN and fiber as both backhaul system operates similarly. The total power consumed by the satellite hub site (backhaul) and SBS are 271.0571 and 216.4400 W, respectively. Component power of 25, 30, 96.8083, and 90.1920 W were consumed by the transponder, VSAT hub, fiber, and C-RAN backhaul, respectively.

Figures 9 and 10 depict the capacity efficiencies of all the backhaul systems under consideration. The capacity efficiency shows how much power is consumed to receive, process, and transmit a bit of information. This is computed by normalizing the power consumption for each backhaul system relative to the backhaul capacity. In this analysis, we used a fixed and uniform capacity of 500 Mbps. Figure 9 shows that fiber is the most efficient as it consumed less power to deliver the expected capacity, followed by the CRAN, satellite, P2MP in that order. Figure 10 shows how this power demand varies with capacity requirements. Similarly, peak power consumption for each backhaul was used, although this is expected to significantly higher than the values used. For a simple illustration, we maintained these peak values. Also, Fig. 10 shows that the capacity efficiency decreases with an increase in data rates, fiber, and C-RAN yielding best results as only about 10 W is required to process and transmit a bit of information at 10 Gbps for fiber, whereas, about 364 W is needed for massive MIMO and 278 W for self-backhauls.

Figure 11 presents the coverage effectiveness for each backhaul system. This is defined as the amount of power consumed per unit square area for each backhaul system. It indicates how much area is covered when 1 W power is consumed. We obtained the coverage effectiveness of self-backhaul and massive MIMO backhaul to be in a range of 0.0016–0.0025 and 0.0015–0.0028, respectively. At medium and high network load, self-backhaul consumes less power than the M-MIMO backhaul because the lower the W/m^2 the better the efficiency. But at extremely low network traffic, less than 10%, the massive MIMO backhaul is more efficient. We obtained the coverage effectiveness of point-to-point, point-to-multipoint microwave, and satellite backhauls to be in a range of 0.3898–0.04996 W/m^2, 0.4503–0.5619, and 0.3808–0.4911, respectively. Fiber and C-RAN backhauls consume negligible power to cover the area.

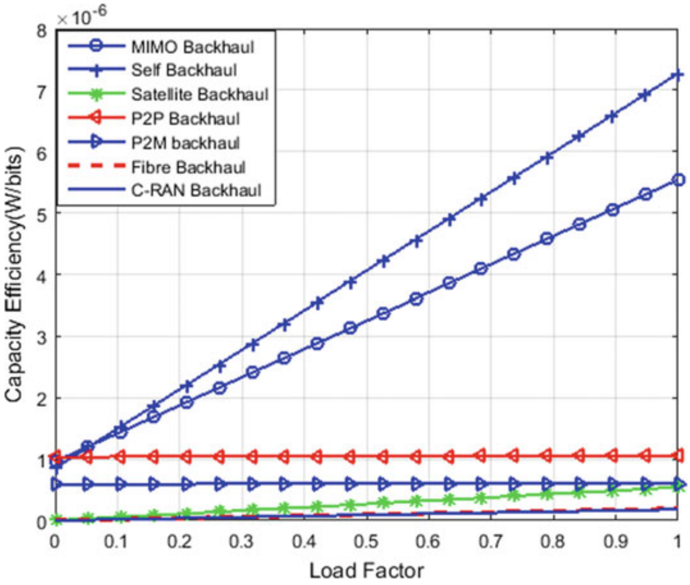


Fig. 9 Capacity efficiency of different backhaul systems

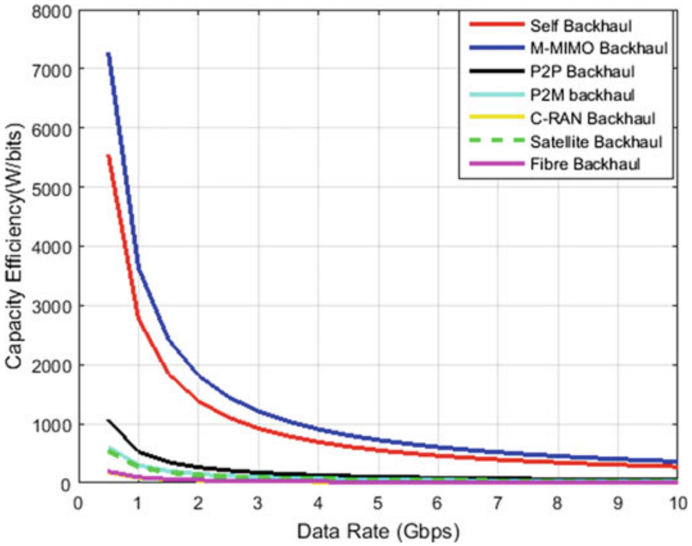


Fig. 10 Capacity efficiency of different data rates

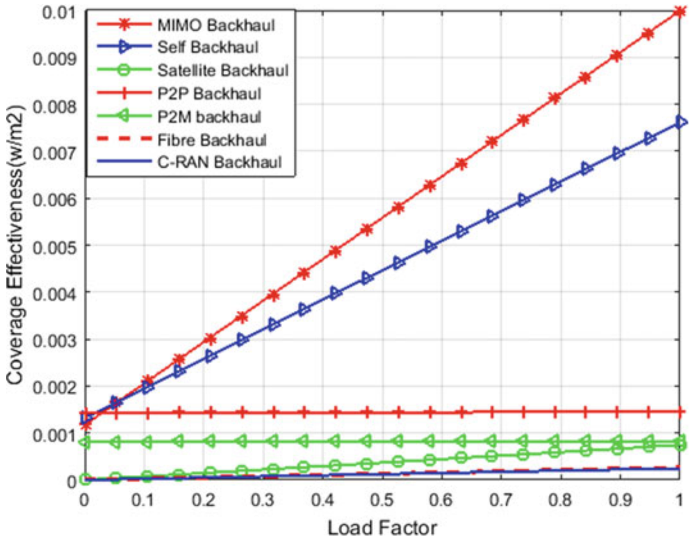


Fig. 11 Coverage effectiveness of different backhaul systems

9 Conclusion

HetNet densification, by the means of deploying a large number of SBS, has been identified as a key enabling strategy in meeting up the unprecedented mobile data demands in next-generation cellular systems. Provision of effective backhaul system for a large number of SBS without a high risk of increasing the CAPEX, OPEX, and network energy costs still remains the main bottleneck.

This paper highlights various backhauling options for future SBS HetNet deployments. The pros and cons for each backhaul system technologies are compared with respect to popular performance metrics that are major drivers for future networks. Fiber optics and C-RAN still remain the most contending options as they are the most energy efficient and could provide unlimited data rates with minimal delay and latencies. The initial CAPEX and ease of deployment, however, may be the issue. Self-backhauling and M-MIMO consumed high power. The power demand varies with capacity requirements. The finding of this study will help the network service providers to make more informed decisions in achieving optimum, sustainable, and energy-efficient backhaul deployment for future heterogeneous networks.

References

1. Ghosh A et al (2012) Heterogeneous cellular networks: from theory to practice. IEEE Commun Mag 50(6)
2. Soh YS et al (2013) Energy efficient heterogeneous cellular networks. IEEE J Sel Areas Commun 31(5):840–850
3. Faruk N et al (2016) Energy savings through self-backhauling for future heterogeneous networks. Energy 115:711–721
4. Spagnuolo A et al (2015) Monitoring and optimization of energy consumption of base transceiver stations. Energy 81:286–293
5. Deruyck M et al (2011) Modelling and optimization of power consumption in wireless access networks. Comput Commun 34(17):2036–2046
6. Zhang Z et al (2015) Large-scale MIMO-based wireless backhaul in 5G networks. IEEE Wirel Commun 22(5):58–66
7. Jaber M et al (2016) 5G backhaul challenges and emerging research directions: a survey. IEEE Access 4:1743–1766
8. Tipmongkolsilp O, Zaghloul S, Jukan A (2011) The evolution of cellular backhaul technologies: current issues and future trends. IEEE Commun Surv Tutorials 13(1):97–113
9. Farias F et al (2016) Cost-and energy-efficient backhaul options for heterogeneous mobile network deployments. Photon Netw Commun 32(3):422–437
10. Muhammad AI, Mona J, Rahim T (2016) 5G Backhual and research direction: a survey. IEEE Access 1743–1766
11. Laxmi RK (2018) Fibre to the BTS—improving network flexibility & energy efficiency. FLA division study paper
12. Tombaz S, Monti P, Wang K, Vastberg A, Forzati M, Zander J (2011) Impact of backhauling power consumption on the deployment of heterogeneous mobile networks
13. GSMA (2014) Wireless backhaul spectrum policy recommendations & analysis. ABI Research Technology and Market Intelligence
14. Faruk N et al (2018) Green energy and technology. In: Advances on computational intelligence in Energy; the application of natural-inspired metahueristic Algorithm. Springer, Berlin
15. Alcatel-Lucent (2018) Small cells backhaul in strategic white paper
16. Oren D (2011) Satellite communication for efficient cellular backhaul
17. Evans BG (2014) The role of satellites in 5G. In: Advanced satellite multimedia systems conference and the 13th signal processing for space communications workshop (ASMS/SPSC), 7th 2014. IEEE
18. NetWorld's–SatCom W (2014) The role of satellites in 5G. White paper
19. Onireti O et al (2015) The role of satellites in 5G
20. Rusek F et al (2013) Scaling up MIMO: opportunities and challenges with very large arrays. IEEE Signal Process Mag 30(1):40–60
21. Larsson EG, Marzetta T, Rusek F (2013) Scaling up MIMO: opportunities and challenges with very large arrays. Sig Process Mag IEEE 30(1):40–60
22. Hoydis J, Ten Brink S, Debbah M (2013) Massive MIMO in the UL/DL of cellular networks: how many antennas do we need? IEEE J Sel Areas Commun 31(2):160–171
23. Li B, Zhu D, Liang P (2015) Small cell in-band wireless backhaul in massive MIMO systems: a cooperation of next-generation techniques. IEEE Trans Wire Commun 14(12):7057–7069
24. Hassan AHM et al (2015) Performance evaluation of qos in wimax network. arXiv preprint arXiv:1506.04902
25. Golshan R (2013) Fixed and mobile WiMAX overview

Analysis of Energy Consumption Model in Cloud Computing Environments

Zhou Zhou, Jemal H. Abawajy and Fangmin Li

Abstract Cloud computing offers software as a service (SaaS), infrastructure as a service (IaaS), and platform as service (PaaS) on pay-as-you-go model over the Internet. Although Cloud have been attractive to businesses and other domains to accommodate their increasing demand for computational power on demand bases, the high energy consumption of Cloud data centers has recently become a serious issue. The high energy consumption not only causes the energy wastes and system instability but also generates low return on the investment (ROI) and adverse effects on the environment. Therefore, it is extremely necessary to reduce energy consumption while meeting the quality of service (QoS). This chapter presents a fine-grained energy consumption model and analyzes its effectiveness in energy consumption of data centers.

Keywords Energy consumption · Energy models · Cloud computing · Quality of service

1 Introduction

Large-scale applications with high computational power demand continue to grow exponentially. At the same time, the construction of large-scale Cloud data centers to meet the requirements of the end-user applications continue to evolve rapidly. As a result, Cloud computing has gained more and more attention from both industrial and academic circles. Cloud computing provides access to a large pool of shared computational resources as a service over the Internet in an on-demand, self-service,

Z. Zhou · F. Li
School of Software, Changsha University, Changsha, China
e-mail: zhouzhou03201@126.com

F. Li
e-mail: lifangmin@whut.edu.cn

J. H. Abawajy (✉)
School of Information Technology, Deakin University, Melbourne, Australia
e-mail: jemal@deakin.edu.au; muc@deakin.edu.au

T. Herawan et al. (eds.), *Advances on Computational Intelligence in Energy*,
Green Energy and Technology, https://doi.org/10.1007/978-3-319-69889-2_10

automatically scalable, and pay-per-use model [1, 2]. Although Cloud computing provides many benefits, the high energy consumption of Cloud data center is a serious concern [3]. The global data center electricity consumption in 2013 is estimated to be more than 4.35 GW with an estimated annual growth rate of up to 15% [4, 5]. Moreover, it is reported that only 10–15% of the supplied electricity to the data center is consumed by servers in data centers [6, 7]. The high Cloud data centers energy consumption has received significant attention recently due to its (i) high operating costs, (ii) adverse effect on the environment, and (iii) a significant impact on performance.

The problem of high energy consumption in Cloud data centers has recently received unprecedented attention. This is because of the fact that the high energy consumption not only causes the energy wastes and system instability, but also generates low return on the investment (ROI) and adverse effects on the environment. Therefore, it is extremely necessary to reduce energy consumption while meeting the quality of service (QoS). As Cloud data centers energy consumption has been steadily increasing over the last few years, the minimization of Cloud data center power and energy consumption has become a challenging problem. A variety of energy-aware algorithms and mechanisms have been proposed to manage and control energy consumption in Cloud data centers. Energy consumption model plays an important role in Cloud data center energy management and control [8]. Thus, any practical approach for minimizing Cloud data center energy consumption requires an accurate modeling of the Cloud data center energy consumption. An energy consumption model is essential for guiding energy-aware algorithms such as resource provisioning policies and mechanisms such as virtual machine migration policies. Moreover, it affects the pricing mechanism which Cloud service providers charge their customers. Therefore, it is necessary to propose an accurate energy consumption model to perform effective management and control.

This chapter presents and analyzes a fine-grained energy consumption model. A fine-grained energy consumption model is essential for guiding energy-aware algorithms such as resource provisioning policies and VM migration policies, while VM migration algorithms leverage both VM migration and host consolidation technology to reduce the energy consumption of data centers. In this chapter, we will explore the fine-grained energy consumption model in data centers, for the purpose of saving energy consumption.

The rest of this chapter is organized as follows: Sect. 2 describes related work. The fine-grained energy consumption model is presented in Sect. 3. Performance analysis of the fine-grained energy consumption model is discussed in Sect. 4. The chapter is concluded with a discussion in Sect. 5.

2 Related Work

The need for managing energy consumption level has become important in various domains [9–14]. Generally speaking, computational resources (e.g., CPU, memory, disk, and networking) and cooling system such as air conditioning equipment are the main energy consumption sources in data centers. There are many energy-aware algorithms such as resource provisioning policies and virtual machine migration policies that aim to minimize the energy consumption of Cloud data center. For example, a three-threshold energy-saving algorithm based on the empirical power model is proposed in Zhou et al. [9, 10]. Beloglazov and Buyya [15] explored the virtual machine migration based on an empirical power model. This power model can be obtained through recording energy consumption and CPU utilization at different load level. An approach that tracks per-VM power consumption is proposed in Kansal et al. [16]. These energy-aware algorithms generally depend on the underlying power models used to develop them. Therefore, an accurate power model is the prerequisite to achieve the fine-grained power control and management in the Cloud data center.

Prior works on power modeling focus on three main aspects: (i) performance-monitor-counter (PMC)-based models [17–23]; (ii) resource utilization-based model [24–30]; and (iii) their usage to guide energy-aware algorithms [9, 10, 15, 16]. PMC-based approaches have three main steps. In the first step, events related to hardware units such as CPU, memory, disk, and NIC are monitored. In the second step, the events are analyzed, and those events that are related to the PMC set are screened out. In the final step, the energy consumption model is built based on the relationship between the PMC events and energy consumption by the system components.

Min et al. [17] proposed a surrogate model that is based on the PMC method. The model can sustain the absolute estimation error of 5.32% when running the SPEC benchmark. In addition, the authors validated the nonlinear relationship between the server energy consumption and CPU utilization. In Bircher and John [18] and Bertran et al. [19], the authors used the microprocessor performance counters to account for the entire system power consumption. Although the approach is promising, the power model did not solve the high relative error problem. The PMC-based model proposed by Bertran et al. [20] utilized the CPU and memory models for virtualized environments. The analysis confirmed that: (1) Virtual machines (VMs) assigned the same amount of CPU cycles do not consume an equal amount of energy; (2) PMC-based method can be used in virtualized environments; (3) Dynamic voltage and frequency scaling (DVFS) method does not affect the accuracy of the power model. The methodology proposed in this paper is promising, albeit it did not consider the consumption of disk and NIC.

In Singh et al. [21], Contreras and Martonosi [22], the authors leveraged the PMC method to build real-time power model. In order to accurately measure the energy consumption of a virtual machine, authors in Xiao et al. [23] proposed a new virtual machine power model based on a concept called a “relative PMC.” Based on the power model, the authors proposed a virtual machine scheduling algorithm to reduce the energy consumption and minimized service level agreement (SLA) violations. However, the approach is complicated as it collects too many events leading to high overheads. Furthermore, the approach is not suitable to extend to other servers or VMs in a data center.

The resource utilization-based approaches leverage the resource utilization of a server (such as CPU utilization, memory utilization, and so on) to construct an energy consumption model. Garg et al. [24] proposed an approach based on the CPU utilization. This model does not reflect the true energy consumption in data centers [26]. Beloglazov et al. [25] leveraged the maximum power consumption, idle power consumption, and CPU utilization of a server to build a power model. Hsu and Poole [26] proposed an exponential model based on CPU-intensive tasks. Zhang et al. [27] argue that the relationship between the energy consumption and the CPU utilization is not linear and instead it is a cubic. Thus, the authors proposed a modified power model named Cubic Model in order to improve the accuracy of the power model. The model proposed in Fan et al. [28] and Economou et al. [29] estimates the energy consumption of the system component (such as CPU) and then builds a linear power model based on statistics. The E-mc2 framework [30] models the requirements of energy consumption in Cloud computing systems. This framework is easy to perform although the accuracy of the power model should be further improved.

There are many energy-aware algorithms such as resource provisioning policies and virtual machine migration policies that aim to minimize the energy consumption of Cloud data center. For example, a three-threshold energy-saving algorithm based on the empirical power model is proposed in Zhou et al. [9, 10]. Beloglazov and Buyya [15] explored the virtual machine migration based on an empirical power model. This power model can be obtained through recording energy consumption and CPU utilization at different load levels. An approach that tracks per-VM power consumption is proposed in Kansal et al. [16]. These energy-aware algorithms generally depend on the underlying power models used to develop them. Therefore, an accurate power model is the prerequisite to achieve the fine-grained power control and management in the Cloud data center.

In summary, although both the disk and NIC subsystems consume considerable energy as compared to CPU and memory subsystems, they are generally ignored in the development of the model. Moreover, different application characteristics lead to different energy consumption. Thus, an accurate Cloud data center energy consumption model must consider not only the CPU, memory, disk, and NIC subsystems but also the application characteristics. By accounting for CPU, memory, disk, and NIC subsystems contribution to the total energy consumption as well as the application characteristics, our approach tackles the shortcomings of the existing models.

3 Fine-Grained Energy Consumption Model of Servers

Existing approaches [20, 24, 27] on energy management models in data centers primarily focus on CPU energy consumption [31], while ignoring the energy consumption by other subsystems such as memory, disk, and NIC subsystems. As CPU is only one of the critical resources in Cloud data centers, data center energy consumption minimization techniques should consider all resources contributing to the overall energy consumption at the same time. With Cloud data centers using huge storage subsystems to store and process data and the increasing communication traffic seen by the data centers make the disk and NIC subsystems significant contributors to the energy consumption of the data centers. Therefore, in addition to the energy consumed by CPU and memory subsystems, the energy consumed by the disk and NIC subsystems should be considered in building the energy consumption model for Cloud data centers. Furthermore, existing approaches do not consider application characteristics when modeling the energy consumption model for Cloud data centers. The fact that different applications impose different resource requirements, considering application characteristics in the development of the model also becomes a primary concern.

In this section, we propose a holistic Cloud data center energy consumption model that is based on the Principal Component Analysis (PCA) and regression methods. Unlike the existing approaches that focus on a single system components in the data center, the proposed approach takes into account the energy consumption of the processing unit, memory, disk, and network interface card (NIC) as well as the application characteristics. The experimental results of the proposed approach show that the proposed energy consumption model achieves more than 95% prediction accuracy. The main contributions of the section are summarized as follows.

- A novel holistic Cloud data center energy consumption model that considers CPU, memory, disk, and NIC subsystems, as well as the application characteristics (CPU-intensive task, transactional web task, and I/O-intensive task).
- Principal Component Analysis (PCA) and regression methods are used to analyze each subsystem parameter's contribution accounting for total energy consumption.
- Extensive experimental analysis to validate the proposed model using widely adopted benchmarks to evaluate the power and performance characteristics of servers [27].
- Comparison of the proposed model with three baseline energy consumption models, the Ramon Model [20], Linear Model [24], and Cubic Model [27].

3.1 Methodology

Figure 1 shows the general steps used to develop the proposed energy consumption model. The methodology consists of the feature extraction, feature selection, modeling, and evaluation steps. The feature extraction step is responsible for collecting features of the resources and applications relevant to energy consumption modeling. This step can be performed by using either resource utilization-based method or PMC-based method. Some features extracted in this step may be related to the power model while others may not be related to the power model.

The feature selection step is responsible for finding good feature representation. This step can be accomplished by deploying approaches such as Correlation Matrix (CM) or Principal Component Analysis (PCA). The power model is then built using the subset features returned from the feature selection step. In this section, we will use regression methods to build the model. An effective power model is characterized

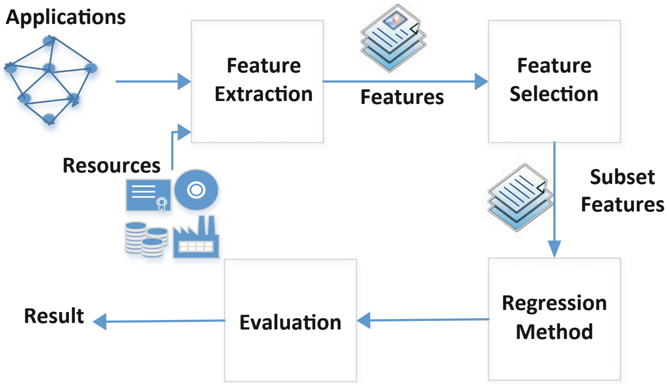


Fig. 1 Construction of energy consumption modeling approach

by its accuracy, representativeness, and extensibility. Finally, in the “evaluate” step, the accuracy of the model is assessed in order to insure its effectiveness.

3.2 *Feature Extraction and Selection*

In this section, we discussed the steps performed to produce the collection of features used to build the energy consumption model.

3.2.1 Feature Extraction

In building a proper power model, it is necessary to include appropriate subsystem parameters related to energy consumption. Let P_{system} be the total power that can be consumed by a server in a data center. Therefore, P_{system} can be modeled with the following equation:

$$P_{\text{system}} = P_{\text{CPU}} + P_{\text{memory}} + P_{\text{disk}} + P_{\text{network}} + \sigma \tag{1}$$

where P_{CPU}, P_{memory}, P_{disk}, and P_{network} are the power of CPU, memory, disk, and NIC, respectively. The parameter σ can be considered as a constant and represents the power of other subcomponents of a system excluding CPU, memory, disk, and NIC. The parameter P_{CPU} can be modeled with the following equation [30]:

$$P_{\text{CPU}} = (P\text{max} - P_{\text{idle}}) \times U + P_{\text{idle}} \tag{2}$$

where $P_{\max}$ represents the maximum power of the server, P_{idle} is the power consumed when the server is idle while U denotes the CPU utilization of the server. As the value of P_{CPU} is related to U, we choose parameter "Processor Time" as energy consumption representative of the CPU. Note that the "Processor Time" *refers to the percentage of an elapsed time that the processor spends executing a non-idle thread. We can monitor the value of* "Processor Time" to get the CPU utilization.

The parameter P_{memory}, it can be modeled as follows [30]:

$$P_{\text{memory}} = P_{\text{PRE}} + P_{\text{ACT}} + P_{\text{RD}} + P_{\text{WR}} + P_{\text{REF}} \tag{3}$$

where P_{PRE}, P_{ACT}, P_{RD}, P_{WR}, and P_{REF} are the power of pre-charge (P_{PRE}), activate (P_{ACT}), read (P_{RD}), write (P_{WR}), and refresh (P_{REF}), respectively. As the value of P_{memory} is associated with writing and reading, we choose "Memory Used" and "Page Fault/Sec" parameters as energy consumption representative of the memory subsystem. "Memory Used" and "Page Fault/Sec" represent the memory utilization and an average number of error pages per second, respectively.

For the parameter P_{disk}, it could be modeled with the following equation [30]:

$$P_{\text{disk}} = P_{\text{READ}} + P_{\text{WRITE}} + P_{\text{IDLE}} \tag{4}$$

where P_{READ}, P_{WRITE}, and P_{IDLE} represent the power needed for reading, writing, and remain idle, respectively. We select parameters "Disk time" and "Disk Bytes/Sec" as the energy consumption representative of the disk. "Disk time" is the percentage of elapsed time that the selected disk drive was busy servicing the read or write requests. The "Disk Bytes/Sec" refers to the total number of bytes sent to the disk (write) and retrieved from the disk (read) over a period of one second.

As for the parameter of P_{network}, it can be modeled as [30]:

$$P_{\text{network}} = C_0 + C_1 \times \frac{S}{B} \tag{5}$$

where parameters C_0 and C_1 can be considered as constants, parameter S is the file size in MB; parameter B is the bandwidth in MB/s. We choose the parameters "Bytes Total/Sec" and "Current Bandwidth" as energy consumption representative of the NIC. "Bytes Total/Sec" is the rate at which the network adapter is processing data bytes; "Current Bandwidth" is the bandwidth.

3.2.2 Feature Selection

In this section, we discuss the process used to select a subset of relevant features from those extracted in the preceding section for use in building power model. In order to determine the subset features, we use a Principal Component Analysis (PCA) [32]. To select a subset features, we deployed three application domains: CPU-intensive applications, transactional web and I/O-intensive applications on DELL PowerEdge R720 with the configuration shown in Table 1.

The SPEC CPU2006 benchmark [33] is a typical example of CPU-intensive task, and it includes "401.bzip2," "403.gcc," "429.mcf," "453.povray," and "450.soplex" subsets. For the transactional web application, we used the HP LoadRunner [34], which is a typical transactional web application. For I/O intensive application, we used Iozone dataset [35], which is a typical I/O-intensive task.

Table 2 shows the feature values (Processor Time, Memory Used, Page Fault/Sec, Disk Time, Disk Bytes Total/Sec, Bytes Total/Sec, and Current Bandwidth) and corresponding energy consumption under three application domains (CPU-intensive applications, Transactional Web, and I/O-intensive applications).

For example, regarding CPU-Intensive application, when "Processor Time" = 4.23%, "Memory Used" = 4.47%, "Page Fault/Sec" = 512.78, "Disk Time" = 0.66, "Disk Bytes/Sec" = 4102.28, "Bytes Total/Sec" = 562.00, and "Current Bandwidth" = 9.22 × 1018, the total energy consumption is 122.49 W. Similarly, for transactional web application, when "Processor Time" = 6.90%, "Memory Used" = 4.29%, "Page Fault/Sec" = 28,192.04, "Disk Time" = 2.86, "Disk Bytes/Sec" = 689,229.22, "Bytes Total/Sec" = 64.13, and "Current Bandwidth" = 9.22 × 1018, the total energy consumption is 107.00 W.

How these features influence the energy consumption? Which feature is related to energy consumption? Which feature is not related to energy consumption? To solve these problems, we make a Principal Component Analysis (PCA) [32] for factors' contribution, and each factors' contribution is listed in Table 3.

As shown in Table 3, the top three features (i.e., Processor Time, Disk Bytes/Sec, and Disk Time) contribute significantly while Page Fault/Sec, "Memory Used," and "Bytes Total/Sec" contribute very little while "Current Bandwidth" does not contribute at all. This is because a CPU-intensive application also called a compute-intensive task requires a lot of processing power as compared to other resources.

The transactional web application is similar to CPU-intensive application regarding the contribution of the features. As shown in Table 3, the top three features (i.e., Processor Time, Disk Bytes/Sec, and Disk Time) contribute significantly while Page Fault/Sec and "Memory Used" contribute very little with and "Bytes Total/Sec" and

Table 1 Configuration of DELL PowerEdge R720

Component name	Parameter
CPU architecture	2 × (intel)Xeon E5-2620 six-core
CPU frequency	12 × 2.0 GHz
Level 1 cache	6 × 32 KB
Level 2 cache	6 × 256 KB
Level 3 cache	15 MB
Memory size	20 GB DDR3
Disk size	2 × 1 TB
NIC	Intel quad-port gigabit network adapter

Table 2 Parameter values and energy consumption under different workload

Processor Time	Memory Used	Page Fault/Sec	Disk Time	Disk Bytes/Sec	Bytes Total/Sec	Current bandwidth	Energy consumption
CPU-intensive application							
4.23	4.47	512.78	0.66	4102.28	562.00	9.22E+18	122.49
5.92	2.85	24,808.87	0.38	4102.59	1785.00	9.22E+18	122.05
6.64	4.48	8374.42	5.26	235,386.71	396.00	9.22E+18	124.19
8.21	4.48	278.10	3.71	213,078.94	4526.00	9.22E+18	127.46
5.60	3.80	50.08	1.20	16,410.60	2574.00	9.22E+18	123.32
3.65 …	2.75 …	41,652.42 …	11.93 …	1,292,362.83 …	567.00 …	9.22E+18 …	99.81 …
Transactional web application							
6.90	4.29	28,192.04	2.86	689,229.22	64.13	9.22E+18	107.00
13.48	4.36	61,633.70	2.86	645,129.15	63.58	9.22E+18	115.18
18.42	4.50	77,579.12	0.90	49,230.76	170.23	9.22E+18	116.05
30.53	4.50	114,919.69	5.26	80,508.48	352.14	9.22E+18	119.46
10.81	4.31	42,296.93	2.86	109,744.25	469.58	9.22E+18	113.23
1.89	4.21	53.09	0.88	3589.75	64.10	9.22E+18	105.05
1.28 …	4.11 …	2047.26 …	0.99 …	4039.30 …	63.12 …	9.22E+18 …	97.77 …
I/O-intensive application							
1.43	3.15	6820.57	33.50	12,965,503.46	96.15	9.22E+18	106.82
2.18	3.15	3568.04	37.17	23,877,620.48	1013.53	9.22E+18	107.17
4.88	3.11	65,473.40	51.19	23,920,602.25	1057.64	9.22E+18	112.85
5.53	2.90	32,459.76	37.59	24,110,357.36	1125.20	9.22E+18	120.78
6.51	2.83	38.06	15.72	17,867,756.48	12,834.79	9.22E+18	121.28
7.10 …	2.95 …	13.02 …	4.32 …	1,932,221.31 …	105.00 …	9.22E+18 …	125.13 …

Table 3 Contributions of various features

Parameter	Application contribution (%)		
	CPU intensive	Transactional web	I/O intensive
Processor Time	62	63	53
Disk Bytes/Sec	19	21	27
Disk Time	14	11	15
Page Fault/Sec	4	3	4
Memory Used	1	1	1
Bytes Total/Sec	0	1	0
Current bandwidth	0	0	0

"Current Bandwidth" contributing nothing. The results for the I/O intensive application are similar to the other two applications with the top three features (i.e., Processor Time, Disk Bytes/Sec, and Disk Time) contribute significantly while Page Fault/Sec and "Memory Used" contribute very little with and "Bytes Total/Sec" and "Current Bandwidth" contributing nothing.

Therefore, we choose the nonzero features (i.e., Processor Time, Disk Bytes/Sec, Disk Time, Page Fault/Sec, and Memory Used) to build the energy consumption model.

3.3 Energy Consumption Modeling

We used the subset features from the previous section and a regression method to construct the energy consumption model. In this section, we use four modeling methods namely a linear regression, a power regression, an exponential regression, and a polynomial regression in combination with the representative parameter (see Sect. 3.2) and EViews 8.0 [36] software to build the energy consumption models. In the following subsection, we use $\boldsymbol{y}, \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4}, \boldsymbol{x_5}$, and $\boldsymbol{x_6}$ to represent the features used for use in building power model as shown in Table 4.

3.3.1 Multivariate Linear Regression Model

The multivariate linear regression model is defined as follows:

Table 4 Parameters used

Parameter	Description
y	Energy consumption
x_1	Processor Time
x_2	Disk Bytes/Sec
x_3	Disk Time
x_4	Page Fault/Sec
x_5	Memory Used
x_6	Bytes Total/Sec

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m + \varepsilon \tag{6}$$

where y is the real energy consumption, β_0, β_1, β_2, …, β_m are the regression coefficients and ε represents a stochastic error. For the CPU2006 benchmark [33], which is CPU-intensive task, the energy consumption model of linear regression is given in Eq. (7).

$$\begin{aligned} y = {} & 102.9169 + 1.967511 \times x_1 - 1.37 \times 10^{-5} \times x_2 - 0.001408 \times x_3 \\ & + 1.29 \times 10^{-5} \times x_4 + 2.528892 \times x_5 \end{aligned} \tag{7}$$

For the transactional web task LoadRunner [34], the energy consumption model based on the linear regression is given in Eq. (8):

$$\begin{aligned} y = {} & -869.7 - 14.18 \times x_1 - 8.68 \times 10^{-5} \times x_2 + 22.92 \times x_3 \\ & + 0.002449 \times x_4 + 234.2339 \times x_5 - 0.067755 \times x_6 \end{aligned} \tag{8}$$

For I/O-intensive task Iozone dataset [35], the energy consumption model based on the linear regression is given in Eq. (9).

$$\begin{aligned} y = {} & 111.5943 + 9.173805 \times x_1 - 1.51 \times 10^{-6} \times x_2 + 2.037900 \times x_3 \\ & - 0.000781 \times x_4 - 19.46270 \times x_5 \end{aligned} \tag{9}$$

3.3.2 Power Regression Model

The power regression model is represented as follows:

$$y = b_0 x_1^{b_1} \cdot x_2^{b_2} \cdot x_3^{b_3} \cdots x_m^{b_m} + \varepsilon \tag{10}$$

where y is the real energy consumption, b_0, b_1, b_2, …, b_m are the regression coefficients and ε represents a stochastic error. The energy consumption model based on power regression approach for CPU2006 benchmark [33] is shown in Eq. (11).

$$y = e^{4.840775} \times (x_1)^{0.219818} \times (x_2)^{-0.056527} \times (x_3)^{0.067893} \times (x_4)^{0.000708} \times (x_5)^{0.096609} \tag{11}$$

For the transactional web task LoadRunner [34], the energy consumption model based on the power regression is given in Eq. (12):

$$y = e^{8.920533} \times (x_1)^{0.198811} \times (x_2)^{-0.008926} \times (x_3)^{-0.028378} \times (x_4)^{-0.016527} \times (x_5)^{-2.920025} \times (x_6)^{-0.014455} \tag{12}$$

For I/O-intensive task Iozone dataset [35], the energy consumption model based on the power regression is given in Eq. (13).

$$y = e^{5.626638} \times (x_1)^{0.038072} \times (x_2)^{-0.000339} \times (x_3)^{-0.054210} \times (x_4)^{0.010081} \times (x_5)^{-0.751834} \tag{13}$$

3.3.3 Exponential Regression Model

The exponential regression model is defined as follows:

$$y = \beta_0 e^{\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m} + \varepsilon \tag{14}$$

where y is the real energy consumption, β_0, β_1, β_2, …, β_m are the regression coefficients and ε represents a stochastic error. The energy consumption model based on exponential regression approach for CPU2006 benchmark [33] is shown in Eq. (15).

$$y = e^{4.641940} \times e^{\begin{pmatrix}0.016 \times x_1 - 1.35 \times 10^{-7} \times x_2 + 0.00085 \times x_3 \\ +1.56 \times 10^{-7} \times x_4 + 0.022 \times x_5\end{pmatrix}} \tag{15}$$

For the transactional web task LoadRunner [34], the energy consumption model based on the exponential regression is given in Eq. (16):

$$y = e^{-4.67} \times e^{\begin{pmatrix}-0.13 \times x_1 - 8.18 \times 10^{-7} \times x_2 + 0.217 \times x_3 \\ +2.30 \times 10^{-5} \times x_4 + 2.23 \times x_5 - 0.0006 \times x_6\end{pmatrix}} \tag{16}$$

For I/O-intensive task Iozone dataset [35], the energy consumption model based on the exponential regression is given in Eq. (17).

$$y = e^{4.741696} \times e^{\begin{pmatrix}0.078 \times x_1 - 1.27 \times 10^{-8} \times x_2 + 0.017 \times x_3 \\ -6.53 \times 10^{-6} \times x_4 - 0.1736 \times x_5\end{pmatrix}} \tag{17}$$

3.3.4 Polynomial Regression Model

As for polynomial regression, the regression model is defined as follows:

$$y = \beta_0 + \beta_1 (x_1)^2 + \beta_2 x_2 + \cdots + \beta_m x_m + \varepsilon \tag{18}$$

where y is the real energy consumption, β_0, β_1, β_2, …, β_m are the regression coefficients and ε represents a stochastic error. The energy consumption model for the CPU2006 benchmark [33] based on the polynomial regression model is shown in Eq. (19).

$$\begin{aligned} y = {} & 111.4598 + 0.151606 \times (x_1)^2 - 1.83 \times 10^{-5} \times x_2 \\ & + 0.420755 \times x_3 + 1.08 \times 10^{-7} \times x_4 + 1.816320 \times x_5 \end{aligned} \tag{19}$$

For the transactional web task LoadRunner [34], the energy consumption model based on the polynomial regression is given in Eq. (20):

$$\begin{aligned} y = {} & -334.1569 - 0.115852 \times (x_1)^2 - 6.70 \times 10^{-5} \times x_2 \\ & + 16.867 \times x_3 + 0.000406 \times x_4 + 102.1 \times x_5 - 0.0797 \times x_6 \end{aligned} \tag{20}$$

For I/O-intensive task Iozone dataset [35], the energy consumption model based on the polynomial regression is given in Eq. (21).

$$\begin{aligned} y = {} & 78.99736 + 1.459156 \times (x_1)^2 - 1.51 \times 10^{-6} \times x_2 \\ & + 2.667544 \times x_3 - 0.000969 \times x_4 - 12.17560 \times x_5 \end{aligned} \tag{21}$$

4 Performance Analysis

In this part, we discuss the experimental analysis of the energy consumption model proposed in this section. All experiments were run on DELL PowerEdge R720 with 2.0 GHz (2 × Six-core), 20 GB RAM, and 2 TB disk storage. The parameters and values for the server configuration are given in Table 1. The benchmark for CPU-intensive task is SPEC CPU2006 [33] (it includes “401.bzip2,” “403.gcc,” “429.mcf,” “453.povray,” and “450.soplex” subsets), and the benchmark for transactional web task is HP LoadRunner [34] and for I/O-intensive task is Iozone [35].

To evaluate the accuracy of the energy consumption model, we define the following metric:

$$\text{Power}_{\text{error}} = \frac{\text{Power}_{\text{predict}} - \text{Power}_{\text{ture}}}{\text{Power}_{\text{true}}} \tag{22}$$

where $\text{Power}_{\text{predict}}$ is the predicted value of the energy consumption by the model, $\text{Power}_{\text{ture}}$ is the true value of the energy consumption, and $\text{Power}_{\text{error}}$ is the relative error of the energy consumption. The true value of the energy consumption is measured using the Power Bay-SSM tool.

We compared the proposed approach with three baseline approaches: the Ramon Model [20], the Linear Model [24], and the Cubic Model [27]. The Ramon Model focuses on CPU and memory, while the Linear Model and the Cubic Model focus on CPU alone.

4.1 *Results and Discussion*

In this section, we discuss the experimental results for the seven models (i.e., linear regression, power regression, exponential regression, and polynomial regression, Ramon Model [20], Linear Model [24], and Cubic Model [27]) under the execution of various applications.

4.1.1 Analysis for CPU-Intensive Task

Figure 2 shows the energy consumption of the seven models while Fig. 3 shows the relative error generated by the seven models for the CPU-intensive task CPU2006 [33].

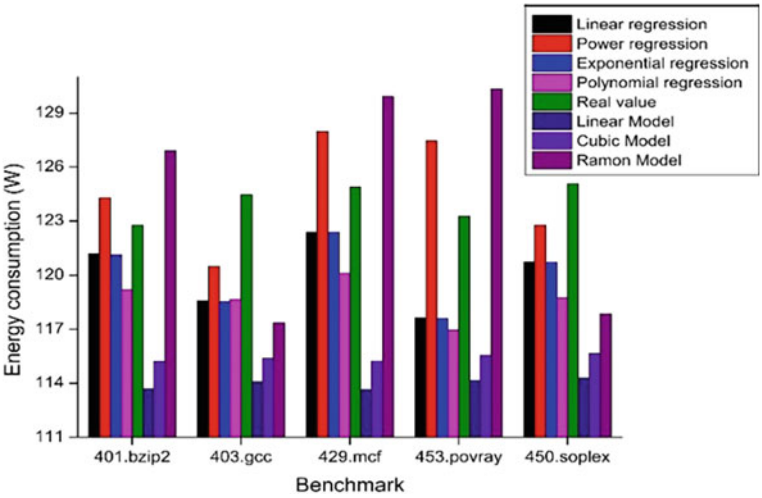


Fig. 2 Energy consumption for the seven models

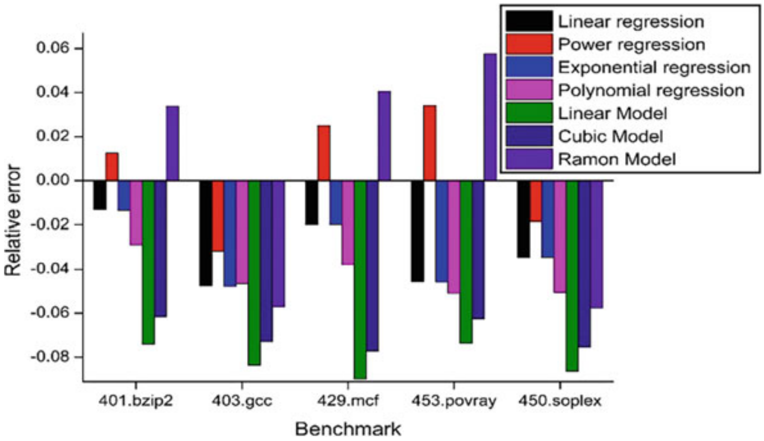


Fig. 3 Relative error for the seven models

Compared to Ramon Model [20], Linear Model [24], and Cubic Model [27], the four modeling methods (linear regression, power regression, exponential regression, and polynomial regression) perform slightly better. The reason is two folds. Firstly, the four modeling methods consider all components related to energy consumption such as CPU, memory, disk, and NIC during the construction of power model, while Ramon Model only takes into account the consumption of the CPU and memory, and Linear Model and Cubic Model only consider the consumption of CPU. Secondly, the four modeling methods leverage the PCA method to improve the accuracy of the power model based on the application characteristics.

4.1.2 Analysis for Transactional Web Task

For the transactional web task LoadRunner [34], Figs. 4 and 5 show the energy consumption and relative error under the transactional web task, respectively.

The four modeling methods (linear regression, power regression, exponential regression, and polynomial regression) perform better than Ramon Model [20], Linear Model [24], and Cubic Model [27]. The reason includes two aspects. On the one hand, the characteristic of the transactional web task determines that this task visits memory and network frequently. Therefore, only considering CPU or memory factor is not enough to build a power model. Conversely, the four modeling methods not only consider CPU and memory factors, but also disk and NIC factors. On the other hand, the four modeling methods utilize the PCA method to improve the accuracy of the power model based on task characteristics. Figures 4 and 5 also illustrate that Ramon Model is better than Linear Model and Cubic Model; this can be explained by the fact that Ramon Model considers both CPU and memory factors, while Linear Model and Cubic Model only consider CPU factor.

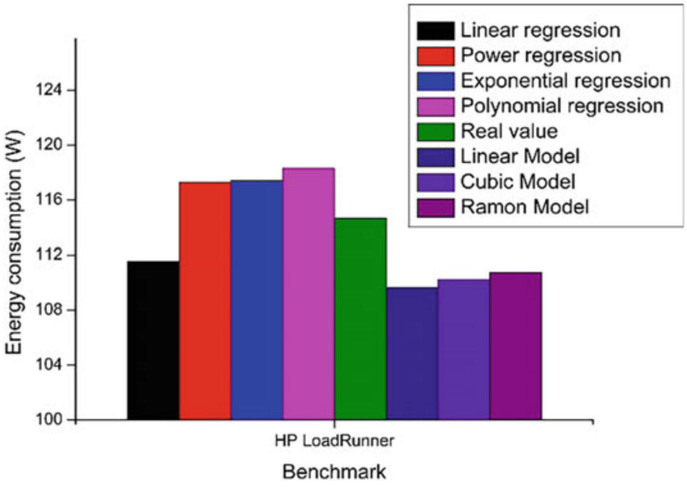


Fig. 4 Energy consumption for the seven models

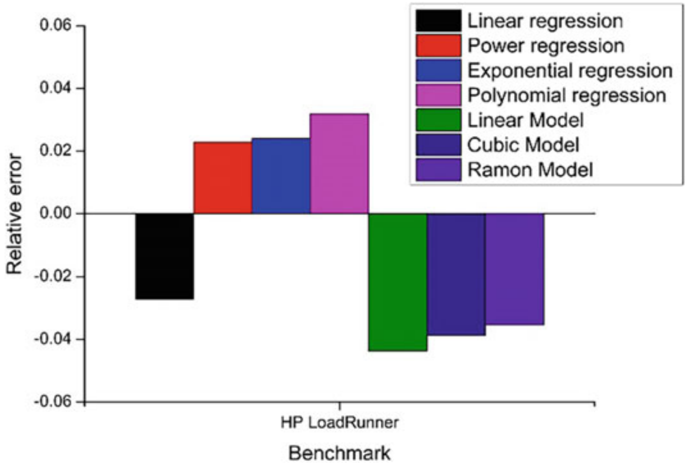


Fig. 5 Relative error for the seven models

4.1.3 Analysis for I/O-Intensive Task

Figures 6 and 7 show the energy consumption and the relative error for the seven models under the I/O-intensive task, respectively.

In comparison with the Ramon Model [20], the Linear Model [24], and the Cubic Model [27], the proposed four models (i.e., linear regression, power regression, exponential regression, and polynomial regression) improve more than 2% accuracy of the energy consumption model. The reason can be explained by the fact that the four modeling methods consider tack characteristics and all factors related to energy

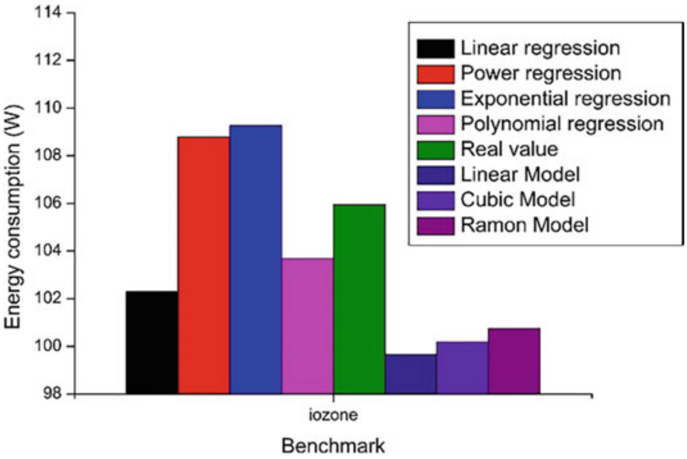


Fig. 6 Energy consumption for the seven models

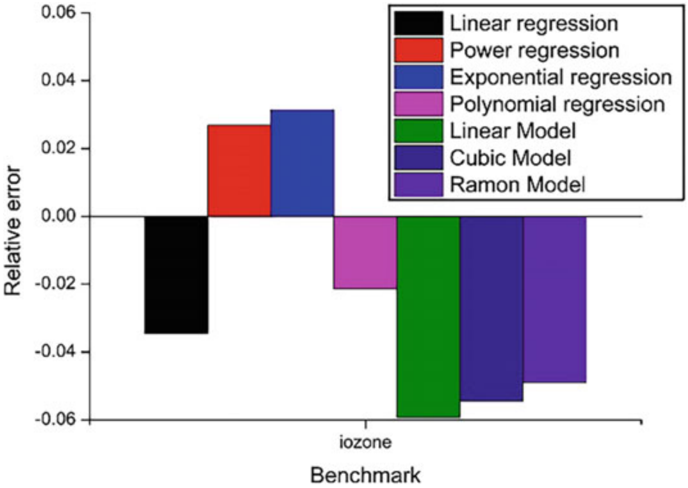


Fig. 7 Relative error for the seven models

consumption (such as CPU, memory, disk, and NIC) during the construction of power modeling. Figures 6 and 7 also show that Ramon Model has better performance than Linear Model and Cubic Model; this reason is that Ramon Model considers both CPU and memory factors, while Linear Model and Cubic Model only consider CPU factor.

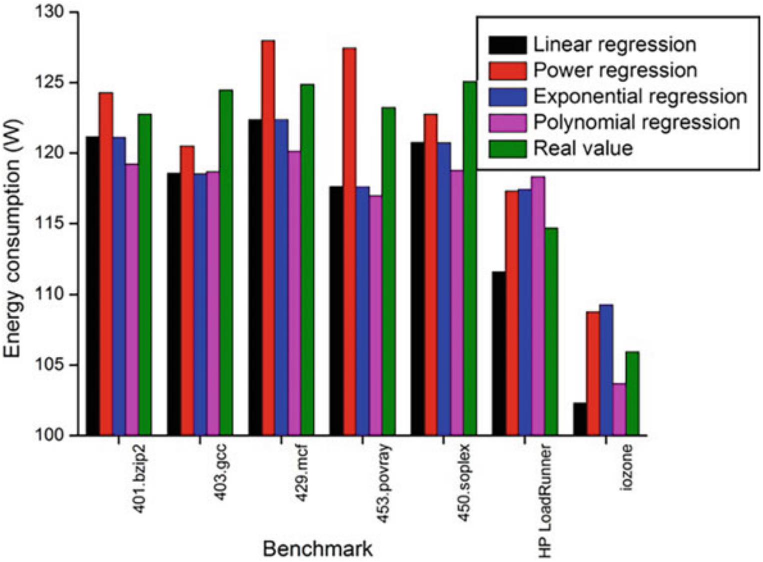


Fig. 8 Energy consumption for the four modeling methods

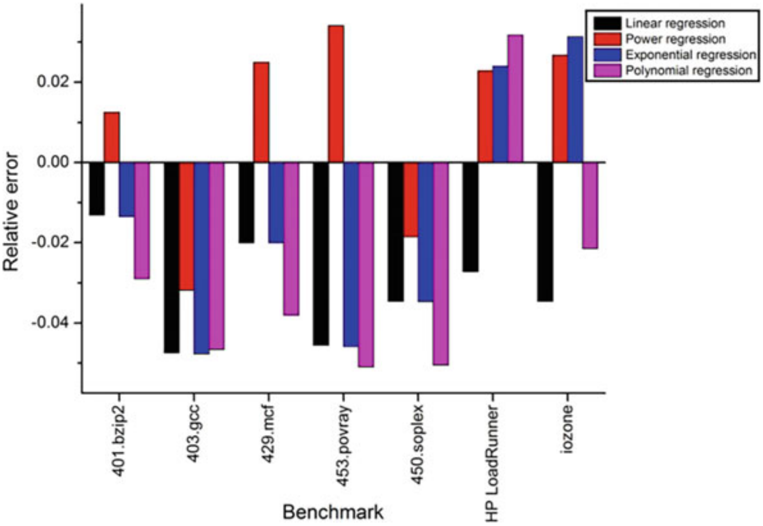


Fig. 9 Relative error for the four modeling methods

4.1.4 The Comparison of the Four Modeling Methods

Figures 8 and 9, respectively, show the comparison for the four modeling methods (linear regression, power regression, exponential regression, and polynomial regression) in terms of energy consumption and relative error. Figures 8 and 9 reveal that, no matter the task belongs to CPU-intensive task or transactional web task or I/O-intensive task, power regression leads to the highest accuracy of power model in most cases. Therefore, we recommend using power regression to build power model in further research.

5 Conclusion

This section proposed an energy consumption model for data centers based on application characteristics (such as CPU-intensive task, transactional web task, and I/O-intensive task) and various subsystems (i.e., CPU, memory, disk, and NIC). Experimental results show that: (1) during the process of building energy consumption model, considering all components related to energy consumption such as CPU, memory, disk, and NIC is more effective than only considering CPU and memory; (2) taking into account, the task characteristics (CPU-intensive task, transactional web task, and I/O-intensive task) provide a better performance than only focusing on CPU-intensive task during the construction of the power model. Moreover, the energy consumption model proposed in this section is more accurate than the existing ones. The proposed energy consumption model can be extended to other servers in data centers, so as to guide the energy-saving algorithm to save energy consumption.

Acknowledgments This work was done while the first author had a visiting position at the School of Information Technology, Deakin University, Australia. The help of Maliha Omar is also sincerely appreciated. This work was supported by the National Natural Science Foundation of China (nos. 61572525, 61373042, 61602525 and 61404213), China Scholarship Council, Deakin University and the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia through the research group project No. RGP-VPP-318.

References

1. Hameed A, Khoshkbarforoushha A, Ranjan R, Jayaraman P, Kolodziej J, Balaji P, Zeadally S, Malluhi Q, Tziritas N, Vishnu A, Khan S, Zomaya A (2016) A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. Computing 98(7):751–774
2. Mastelic T, Oleksiak A, Claussen H, Brandic I, Pierson J, Vasilakos A (2015) Cloud computing: survey on energy efficiency. ACM Comput Surv 47(2):1–36
3. Chen Y, Chang JM (2015) EMaaS: cloud-based energy management service for distributed renewable energy integration. IEEE Trans Smart Grid 6(6):2816–2824

4. Liu Y, Yuen C, Yu R, Zhang Y, Xie S (2016) Queuing-based energy consumption management for heterogeneous residential demands in smart grid. IEEE Trans Smart Grid 7(3):1650–1659
5. Niewiadomska-Szynkiewicz E, Sikora A, Arabas P, Kamola M, Mincer M, Kołodziej J (2014) Dynamic power management in energy-aware computer networks and data intensive computing systems. Future Gener Comput Syst 36(5):284–296
6. Wang L, Khan LU (2013) Review of performance metrics for green data centers: a taxonomy study. J Supercomput 63(3):639–656
7. Ganesh L, Weatherspoon H, Marian T, Birman K (2013) Integrated approach to data center power management. IEEE Trans Comput 62(6):1086–1096
8. Xu Z, Xia Y, Luo S (2013) Energy-aware management in wireless body area network system. KSII Trans Int Inf Syst (TIIS) 7(5):949–966
9. Zhou Z, Hu Z, Li K (2016) Virtual machine placement algorithm for both energy-awareness and SLA violation reduction in cloud data centers. Sci Program 16(1):1–11
10. Zhou Z, Hu ZG, Song T, Yu JY (2015) A novel virtual machine deployment algorithm with energy efficiency in cloud computing. J Cent South Univ 22(3):974–983
11. Buyya R, Beloglazov A, Abawajy J (2010) Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges. In: Proceedings of the 2010 international conference on parallel and distributed processing techniques and applications (PDPTA 2010). CSREA Press, USA, pp 6–17
12. Zhou Z, Abawajy J, Chowdhury M, Hu Z, Li K, Cheng H, Alelaiwi AA, Li F (2017) Minimizing SLA violation and power consumption in cloud data centers using adaptive energy-aware algorithms. Future Gener Comput Syst 86(6):836–850
13. Baccarelli E, Naranjo PGV, Scarpiniti M, Shojafar M, Abawajy JH (2017) Fog of everything: energy-efficient networked computing architectures, research challenges, and a case study. IEEE Access 5(1):9882–9910
14. Khayyam H, Abawajy J, Javadi B, Goscinski A, Stojcevski A, Bab-Hadiashar A (2013) Intelligent battery energy management and control for vehicle-to-grid via cloud computing network. Appl Energy 111:971–981
15. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr Comput Pract Exp (CCPE) 24(13):1397–1420
16. Kansal A, Zhao F, Liu J, Kothari N, Bhattacharya AA (2010) Virtual machine power metering and provisioning. In: proceedings of the 1st ACM symposium on cloud computing (SoCC '10). ACM, New York, pp 39–50
17. Min YL, Porterfield A, Fowler R (2010) SoftPower: fine-grain power estimations using performance counters. In: ACM international symposium on high performance distributed computing (HPDC 2010), pp 308–311
18. Bircher WL, John LK (2012) Complete system power estimation using processor performance events. IEEE Trans Comput 61(4):563–577
19. Bertran R, Gonzalez M, Martorell X, Navarro N, Ayguade E (2010) Decomposable and responsive power models for multicore processors using performance counters. In: International conference on supercomputing, pp 147–158
20. Bertran R, Becerra Y, Carrera D, Beltran V, Gonzàlez M, Marc M, Xavier N, Nacho T, JordiAyguadé E (2012) Energy accounting for shared virtualized environments under DVFS using PMC-based power models. Future Gener Comput Syst 28(2):457–468
21. Singh K, Bhadauria M, McKee SA (2009) Real time power estimation and thread scheduling via performance counters. ACM SIGARCH Comput Archit News 37(2):46–55
22. Contreras G, Martonosi M (2005) Power prediction for intel XScale® processors using performance monitoring unit events. In: Proceedings of the 2005 international symposium on low power electronics and design, pp 221–226
23. Xiao P, Hu Z, Liu D, Yana G, Qu X (2013) Virtual machine power measuring technique with bounded error in cloud environments. J Netw Comput Appl 36(2):818–828
24. Garg SK, Versteeg S, Buyya R (2013) A framework for ranking of cloud computing services. Future Gener Comput Syst 29(4):1012–1023

25. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener Comput Syst 28(5):755–768
26. Hsu CH, Poole SW (2011) Power signature analysis of the SPECpower_ssj2008 benchmark. In: 2011 IEEE international symposium on performance analysis of systems and software (ISPASS), pp 227–236
27. Zhang X, Lu J, Qin X, Zhao X (2013) A high-level energy consumption model for heterogeneous data centers. Simul Model Pract Theory 39(2):41–55
28. Fan X, Weber WD, Barroso LA (2007) Power provisioning for a warehouse-sized computer. In: ACM SIGARCH computer architecture news, pp 13–23
29. Economou D, Rivoire S, Kozyrakis C, Ranganathan P (2006) Full-system power analysis and modeling for server environments. In: Workshop on modeling benchmarking and simulation (MOBS), pp 70–77
30. Castañé GG, Núñez A, Llopis P, Carretero J (2013) E-mc2: a formal framework for energy modelling in cloud computing. Simul Model Pract Theory 39(39):56–75
31. Nguyen TH, Di Francesco M, Yla-Jaaski A (2017) Virtual machine consolidation with multiple usage prediction for energy-efficient cloud data centers. IEEE Trans Serv Comput, 05 Jan 2017. https://doi.org/10.1109/tsc.2017.2648791
32. Wei L, Zhang WX (2013) Principal component analysis. Inf Technol J 16(12):3660–3664
33. The Standard Performance Evaluation Corporation. SPEC cpu2006 User's Guide [EB/OL]. https://www.spec.org/cpu2006/
34. Hewlett-Packard, HP Software Division. HP Load Runner [EB/OL]. http://en.wikipedia.org/wiki/HP_LoadRunner
35. Norcott WD Iozone File system Benchmark [EB/OL]. http://www.iozone.org/
36. Bro R, Smilde AK (2014) Principal component analysis. Anal Methods 9(6):2812–2831