*Research and Development on*
**Information and Communication Technology**
https://ictmag.vn
**Volume 2020, Number 1**
**September**

# TABLE OF CONTENTS

**INVITED ARTICLES**

Computational Personalized Medicine in Cancer Research of the –Omics Data Era ..... ***Duc-Hau Le, Quynh Diep Nguyen*** 1

**REGULAR ARTICLES**

Handling Imbalanced Data in Intrusion Detection Systems using Generative Adversarial Networks ..... ***Ly Vu, Quang Uy Nguyen*** 9

Controlling Contention Window to Ensure QoS for Multimedia Data in Wireless Network ..... ***Ngo Hai Anh, Pham Thanh Giang*** 22

An Efficient ACO+RVND for Solving the Resource-Constrained Deliveryman Problem ..... ***Ha-Bang Ban*** 32

Impact of Inter-Channel Interference on Shallow Underwater Acoustic OFDM Systems ..... ***Do Viet Ha, Nguyen Tien Hoa, Nguyen Van Duc*** 42

# Computational Personalized Medicine in Cancer Research in the -Omics Data Era

Duc-Hau Le, Quynh Diep Nguyen

School of Computer Science and Engineering, Thuyloi University, Hanoi, Vietnam

Correspondence: Duc-Hau Le, hauldhut@gmail.com

Communication: received 29 October 2019, revised 16 April 2020, accepted 6 May 2020

Digital Object Identifier: 10.32913/mic-ict-research.v2020.n1.899

**Abstract: Omics data (e.g., genomics, transcriptomics, proteomics, epigenomics, etc . . . ) generated from high-throughput next-generation sequencers in the big human genome, and cancer genome projects have changed the way to study personalized medicine. In the future, personalized medicine will not be limited to diagnosis and treatment based on a few known disease-associated mutations on some genes, but will rely on whole molecular characteristics of patients by integrating their –omics data. In this study, we draw a big picture of personalized medicine research in cancer research of the –omics data era, including –omics databases, challenges of data fusion to solve two major problems in personalized medicine, i.e., personalized diagnosis and treatment. These problems are approached as patient stratification and drug response prediction based on the –omics data by computational methods.**

**Keywords:** *Personalized medicine, omics data integration, patient stratification, drug response prediction, computational method.*

## I. Introduction

In biomedicine, with high-throughput technologies such as next-generation sequencing (NGS), we are witnessing the exponential growth of biomedical data (e.g., –omics data). To date, it is widely accepted for most diseases that their genetic etiology cannot be explained with one or a few genes/variants, but thousands of whole-genome/variome (i.e., genomics/variomics data). Besides, it is still not enough if the only genome is considered, but also transcriptome (e.g., transcriptomics data), because this data reflects how much change in a genome can affect the expression of a gene in different tissues. More importantly, how the gene expression translated into proteins is a critical issue since protein functions decide how a cell works. Therefore, besides genomics and transcriptomics, proteomics data should also be considered when studying the disease etiologies. It should be noted that the central dogma in biology states protein synthesis as DNA→RNA→Protein. However, only $1 \rightarrow 2\%$ of DNA directly encodes for proteins. The major part ($98 \rightarrow 99\%$ of DNA) are non-coding regions, which do not directly synthesize the proteins but regulate the synthesis process as well as other biological processes and cellular functions. Therefore, epigenome (i.e., epigenomics data) should be taken into account for the disease etiologies. Moreover, cells function under chemical reactions between metabolites; thus, metabolome (i.e., metabolomics data) is also critical. Finally, cellular components such as genes, proteins, and metabolites,. . . do not function alone, but they interoperate in the form of biological networks to make the cells robust against external stimulus or mutations; thus, interactomics data is a final "piece of cake" for a big picture of the –omics data.

In this study, we draw a big picture of personalized medicine in cancer research of the –omics data era. Personalized medicine is a potential approach in biomedicine with the goal is to find the right treatment (i.e., right drug, right dose, and right time) for each patient based on their biological characteristics. Different patients have different biological characteristics in terms of –omics data; thus, the development of the right methods for diagnosis and treatment is a challenging task due to the diversity and complexity of the –omics data. However, to fully understand the biological characteristics of each patient, their –omics data should be combined together in the right ways. Therefore, besides introducing –omics databases, we also discuss how to integrate the –omics data for solving the two main problems (i.e., diagnosis and treatment) in personalized medicine.

## II. The –omics data

With the rapid development of NGS technologies, the cost for sequencing is significantly going down, and the –omics data are growing exponentially. Biomedical research nowadays is not only based on the genomics data, but also on other –omics data such as transcriptomics,

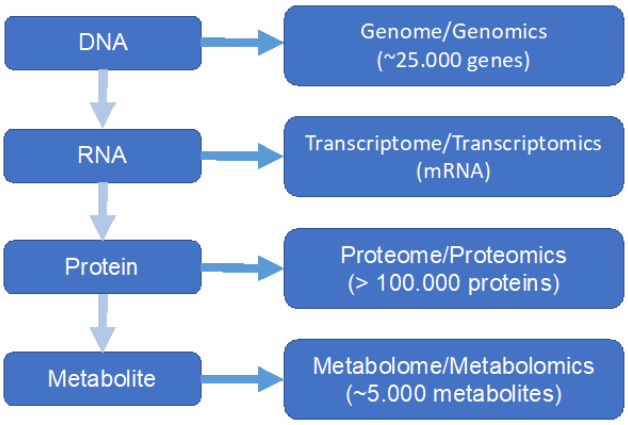


Figure 1. The –omics layers.

proteomics, and metabolomics [1]. Figure 1 shows the central dogma in biology with the corresponding –omics data layers.

A genome is understood as whole genetic information on DNA of a species. The human genome contains about three billion base pairs (bp), which encode for about 25,000 genes. Coding regions occupy approximately $1 \rightarrow 2\%$ of the genome, and the remaining is non-coding regions [2]. Change in a genome (variants) can happen at a single point, such as SNP (Single Nucleotide Polymorphism) or short insertion and deletion (short indel). Those changes can be detected by traditional techniques such as Sanger or modern ones such as microarray and NGS. If the changes happen in protein-coding regions of DNA, they can be synonymous, missense, nonsense, and frameshift. When they occur in the remaining regions, those changes may affect the level of gene expression (e.g., up and down) or RNA splicing process. Besides such small changes, larger changes are also observed, such as structure variants (SV), where the insertions and deletions are found in large regions (100 – 1000 bp). In addition, copy number variant (CNV), inversion, and translocation are also found. For those changes, the microarray or NGS technologies are required for the detection.

A transcriptome includes all RNAs (coding RNAs such as mRNA and non-coding RNAs) in a cell. RNA reflects how much genetic information (i.e., embedded in genes) from DNA is expressed. To measure the level of gene expression, microarray and RNA-seq are widely used. The microarray technology is used to detect known transcripts and isoforms with established and standard pipelines/tools. Meanwhile, the RNA-Seq, as a modern technology, can be used to detect de novo transcripts and isoforms as well as structural variants. However, it also yields big and complex data, thus requires more sophisticated pipelines/tools.

A proteome is all proteins in a cell at a specific time of development. Two main problems that are widely studied for proteins are protein structure and protein interaction network. A total of 20 types of amino acids can be translated by the protein synthesis process. In the structural view, a protein is a sequence of amino acids (aka polypeptide), which can be in different shapes such as linear (level 1) and fold (level 4). The structural features decide the protein's functions. For the second problem, physical interactions between proteins by chemical bonds form the protein interaction networks or protein complexes. These also decide the functions of cells.

A metabolome is mentioned as small chemical compounds such as endogenous and exogenous metabolites.

Figure 1 shows that, for humans, there are about 25,000 genes in the genome, more than 100,000 proteins, and 5,000 metabolites [3]. Those data generated from large-scale biomedical projects such as human genomes, cancer genomes, and cell lines have yielded big–omics databases. In the next section, we are going to introduce some of such projects briefly.

## III. The –omics database

The –omics databases are generated from large-scale projects on the human genome, such as UK's 100,000 Genomes [4] and GenomeAsia 100K [5]. In addition, disease-specific projects such as those for cancer genome have largely contributed their data to the research community. Here are some critical projects which are widely used in cancer research.

The Center for Cancer Genomics of National Cancer Institute in the US had launched a number of big projects on cancers such as TCGA (The Cancer Genome Atlas) [6], CGCI (Cancer Genome Characterization Initiative) and TARGET (Therapeutically Applicable Research to Generate Effective Treatments). These projects are now managed by Genomic Data Commons data portal (GDC, https://portal.gdc.cancer.gov) to foster precise medicine in cancer. The portal has deposited data for more than 40 cancer projects on 61 organs, with 32,555 samples [7]. A number of –omics data has been generated from the projects such as single point somatic/germline mutation, structural somatic mutations, gene/microRNA expression by microarray, gene expression by RNA-Seq, and DNA methylation data. Besides, the Catalogue Of Somatic Mutations In Cancer project (COSMIC) is the largest and most comprehensive database of cancer somatic mutations on human [8]. Major data of COSMIC is collected and curated by experts. It also provides 3D structures of the mutations as well as related genes. In addition, more than 1,000 mutation profiles of cell lines are also accumulated. Finally, the

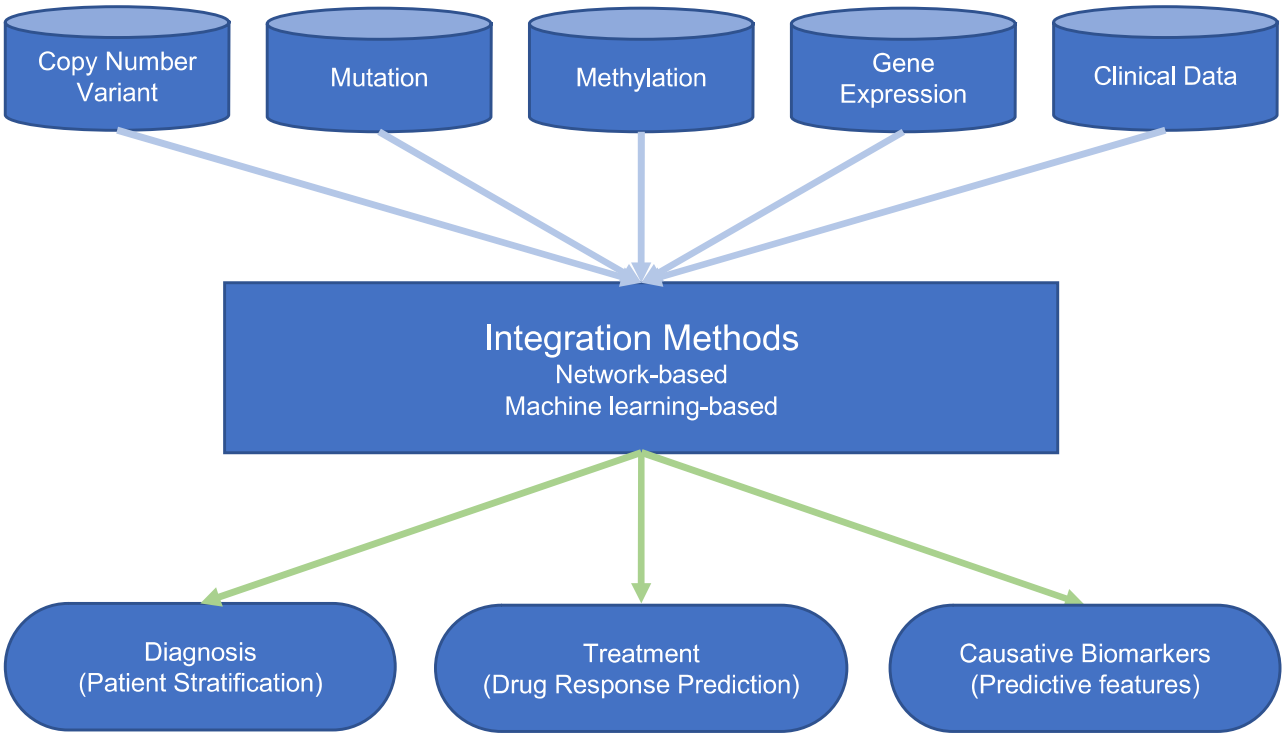


Figure 2. Integration of the –omics data in personalized medicine problems.

International Cancer Genome Consortium (ICGC) aimed to define the genomes of 25,000 primary untreated cancers, as its data portal [9] makes global genomic data sharing for cancer possible, providing the international community with comprehensive genomic data for many cancer types.

In parallel with the abovementioned projects on cancer tumors, projects on testing drugs/compounds on cell lines were also launched. The outcome of those projects are databases of the cell line's biological characteristics and responses of the tested drugs/compounds. These data help foster research on pharmacogenomics, which is the most promising direction in personalized medicine. Two typical projects are Cancer Cell Line Encyclopedia (CCLE) [10] and Genomics of Drug Sensitivity in Cancer (GDSC) [11]. CCLE aims to identify –omics and pharmacological properties of cancer cell lines, then developing analysis tools for predicting drug responses based on the –omics data. A potential application of this is to stratify patients based on their –omics and pharmacological characteristics. CCLE provides the –omics data including single point, indel, and structural mutations and the gene expression of more than 1,000 cancer cell lines as well as the pharmacological properties of 24 drugs. Similarly, GDSC freely published those data for about 1,000 cancer cell lines and 265 drugs/compounds.

## IV. Integration of the –omics data

It is obvious that the future of biomedical research will mainly rely on the –omics data because the biological characteristics of patients are fully considered. Therefore, with the assumption that "More is better", recent biomedical research on the –omics data is trying to combine as many –omics data and other clinical data as possible to solve the two major problems in personalized medicine, i.e., diagnosis (aka patient stratification) and treatment (aka drug response prediction) (see next section) [1, 3, 12]. In addition, the identification of causative biomarkers (aka predictive features) is also addressed (Figure 2).

Data integration is approached by machine learning-based [13] network-based [14] methods. The machine learning-based methods rely on unsupervised and supervised techniques. The typical unsupervised methods, such as matrix factorization, aim at subtyping cell lines using the –omics data with the final goal is stratifying patients. Meanwhile, with supervised methods, labelled data (e.g., patients/cell lines with known drug responses or subgroups) is needed for training prediction models. The trained models are then used to predict for patients/cell lines with unknown drug responses and subgroups. In parallel, the networkbased methods often rely on the functional relationships among biomedical entities in the same or different

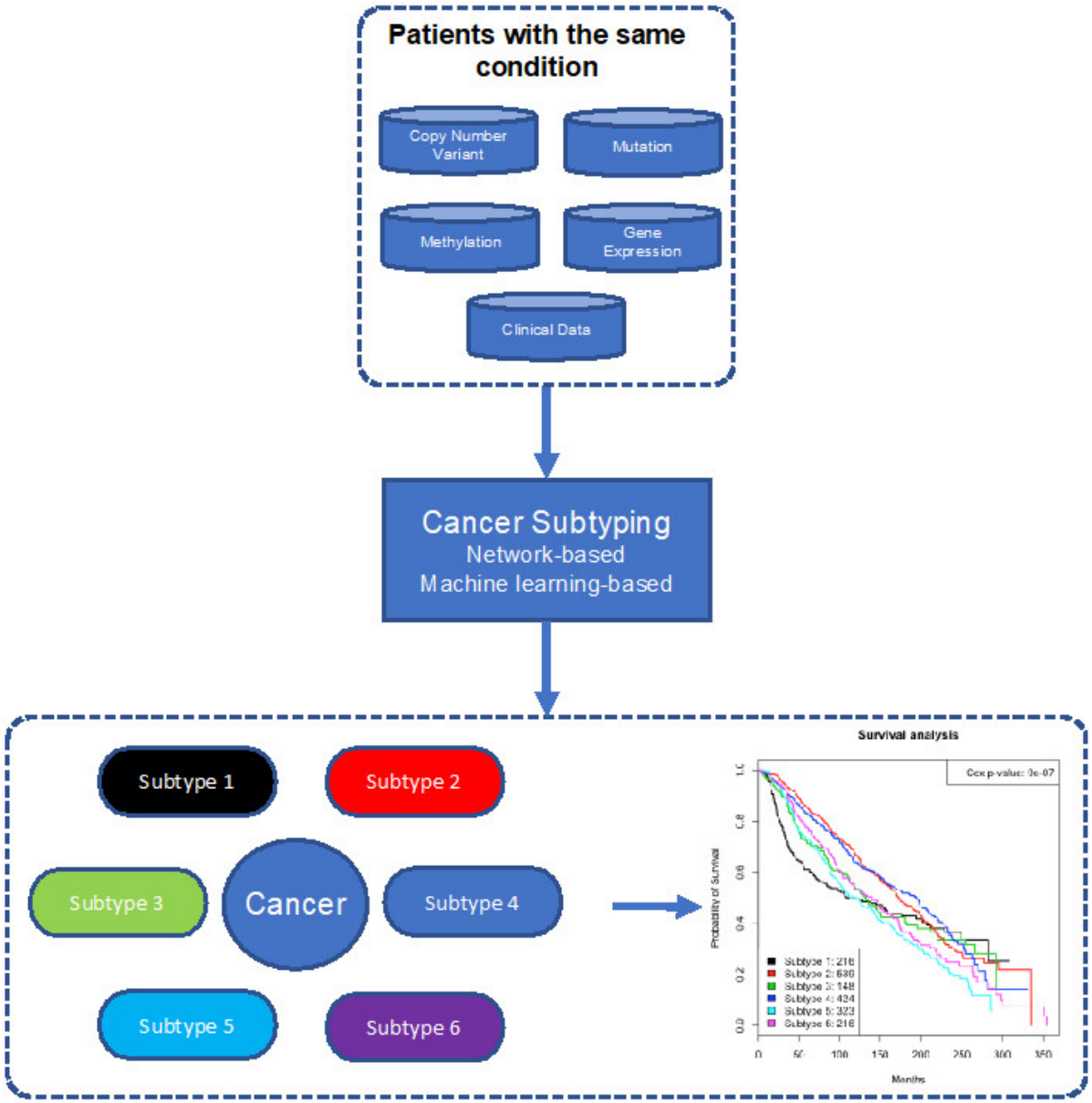


Figure 3. Patient stratification using –omics data.

–omics layers [14].

Along with the advantages of integration, there are also some challenges in integrating the whole genome, transcriptome, and proteome data are also faced with. An obvious problem is "highdimensional data" since thousands to millions of genes/proteins and variants must be investigated. Therefore, techniques in dimension reduction [15] or feature selection [16] have been proposed to select predictive features. Those features are also important biomarkers for diseases. In contrast with the high dimensionality, the number of samples is relatively small (i.e., often thousands). Thus the most challenging problem when dealing with the integration of largescale –omics data is "small $n$, large $p$" (a small number of sample, but high dimension). This raises a lot of problems requiring computational solutions [17].

In summary, the data integration aims to systematically assess the biological characteristics of patients; thus, the application of identifying cancer genes [18] and cancer diagnosis [19] in personalized medicine is approached more precisely. In the next sections, we introduce how the two major problems in computational personalized medicine, i.e., diagnosis (patient stratification) and treatment (drug response prediction), are approached using the –omics data.

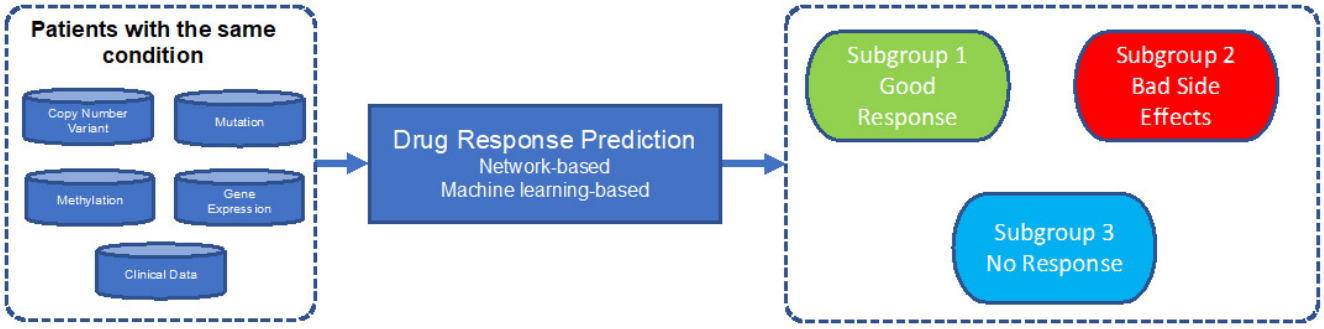


Figure 4. Drug response prediction using –omics data.

## V. Patient Stratification Using –omics Data

In this section, we introduce how to stratify patients using the –omics data. Cancer is selected for study as it is a complex disease. The genetic etiologies are also diversified among different cancers/subtypes of the same cancer. Even cells of the same cancer tumor may also act differently (Figure 3). The patient stratification has been studied based on molecular subtyping for some types of cancer, such as bladder urothelial carcinoma [20] and breast cancer [21].

As the abovementioned of the rapid growth of the –omics data, this fosters the development of computational methods for patient stratification based on molecular profiles and towards clinical application [22]. The problem is generally solved by the following four steps: i) data preprocessing, ii) clustering, iii) supervised classification, and iv) subtype characterization. For the first step, raw –omics data from high-throughput sequencers or genotyper are qualifiedly controlled, standardized, and transformed into the right formats for the next steps. Then, the data clustering is performed based on either genes/proteins or samples. If these two types of data are used simultaneously, then biclustering techniques are used. In addition, triclustering techniques are used if other factors, such as time, are considered. In the third step, supervised classification techniques are used to determine which subtype/subgroup a patient belongs to. Finally, the identified subtypes are characterized by causative biomarkers, involved pathways, and survival time.

Two main approaches have been proposed for this problem, including i) machine learningbased and ii) network-based methods. A common point of the two approaches is that they both rely on the similarity between samples (i.e., the similarity of patient's molecular profiles). The similarity can be represented as similarity matrices in machine learning-based methods or similarity networks in network-based methods. By representing as similarity matrices, clustering techniques on a matrix such as matrix factorization are widely used [23, 24]. For network-based methods, network clustering techniques such as module/community detection are often used. Those techniques can be applied to different types of biological networks such as patient similarity network [25], protein interaction network [26], and gene network [27]. Recent methods have relied on both matrix and network clustering techniques [28] or matrix clustering techniques but guided by a network [23, 29].

Although many methods have been proposed, a number of tools have been developed for the patient stratification problem. However, different methods could lead to different subtypes. This is caused by the complexity of the –omics data (i.e., small $n$, large $p$). In addition, the cancer tumor is heterogeneous; thus, the molecular profiles (i.e., the –omics data) are different for even cells derived from the same cancer types.

## VI. Drug Response Prediction Using –omics Data

Besides the requirement of precisely stratifying patients based on their –omics data, personalized medicine also requires selecting the right drugs for the right treatment for each patient. Patients with the same conditions/cancer types may respond differently to the same drugs due to their different molecular profiles. Thus, the prediction of drug response using the –omics data is an important step for selecting the right drugs (Figure 4).

Similar to the patient stratification problem, many methods have also been proposed for predicting drug response for patients/cell lines [30]. The drug response is measured by dose level to inhibit 50% of the disease's bioactivity (IC50), or they are under the dose-response curve (AUC). They are both continuous values. Thus, the drug response prediction is often approached by regression techniques. However, response values can be categorized into some levels, such as good response, no response, and bad side effects (Figure 4); thus, it can be formulated as a classification problem. The main difference between the two main

problems in personalized medicine is that the patient stratification is usually based on tumor data from tumor/patient-based projects such as TCGA. Meanwhile, the drug response prediction uses cell line and drug response data from drug trial projects such as CCLE and GDSC.

Generally, machine learning- and network-based methods are often proposed for the drug response prediction. Network-based methods are usually based on similarity networks of drugs and cell lines and local [31] or global [32] graph traversal algorithms. In contrast to a few network-based methods, many machine learning-based methods have been proposed for the drug response prediction problem. Indeed, a challenge was organized for research groups over the world [33]. Interestingly, the winner over 44 submissions is a method integrating the –omics data (including single point mutation, structural mutation, gene expression by microarray and RNA-Seq technologies, methylation, and protein data) using multiple kernel learning technique [34]. Other methods also show that response prediction for multiple drugs simultaneously achieve better performance than that for a single drug, because functional and structural similarity among drugs is taken into account [34, 35]. In addition, gene expression data is more dominant than the others [33]. Finally, until now, computational methods for the drug response prediction have been proposed mostly for cancer cell lines. Thus to translate them to clinical application, a recent method has built the prediction model using the data from cell lines in GDSC, then use the built model for predicting drug response for patients in TCGA [36].

## VII. Conclusions

Nowadays, the rapid development of high-throughput technologies and large-scale genome projects have generated a large amount of the –omics data (i.e., the –omics era). This has changed the ways to computationally approach the problems in personalized medicine. To fully understand the biological characteristics of patients, their molecular profiles at the –ome scale has been studied. Thus, the –omics data has been integrated into computational methods to solve the problems in personalized medicine. The two major problems in medicine (i.e., diagnosis and treatment) are formulated as two problems in computational space (i.e., patient stratification and drug response prediction, respectively). Although current studies of the two problems target different objects, i.e., the patient stratification mainly focuses on patient data from the patient/tumor projects; meanwhile, the drug response prediction mostly works with artificial patients/tumors (i.e., cell lines). However, they are both personalized based on molecular profiles of each patient/tumor/cell line. Integration of the –omics data algorithmically faces with the "small $n$, large $p$" problem. The object (i.e., cancer) itself is a complex disease, which is heterogeneous between cancer types and even cells in the same tumor. In addition, unexpected changes in characteristics of cell lines during culture may limit the translation of research results on cell lines to patients. Fortunately, many big human genome and disease genome projects have been launched and freely published the data for the research community. In parallel, state-of-the-art techniques in computational sciences (e.g., artificial intelligence, statistics) have fostered the application of computational methods to study problems in medicine. This could open a brighter future for personalized medicine in cancer research of the –omics data era. Personalized medicine is a broad research area and application. Indeed, besides biological characteristics of the patients, their clinical data, environment, and lifestyles are also important factors in tailoring the individual treatments. In addition, personalized medicine approaches are not only limited to cancers, but also be used to diagnose and treat other disorders such as rare diseases, which are strongly linked to molecular alterations. Furthermore, besides the abovementioned –omics data, metagenomics and metatranscriptomics should also be worthy of studying personalized medicine since there exist interactions between humans and the microbiome.

## Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2017.14.

## References

[1] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton *et al.*, "The sequence of the human genome," *science*, vol. 291, no. 5507, pp. 1304–1351, 2001.

[2] C. Manzoni, D. A. Kia, J. Vandrovcova, J. Hardy, N. W. Wood, P. A. Lewis *et al.*, "Genome, transcriptome and proteome: The rise of omics data and their integration in biomedical sciences," *Briefings in Bioinformatics*, vol. 19, no. 2, pp. 286–302, 2018.

[3] J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski *et al.*, "GENCODE: The reference human genome annotation for the ENCODE project," *Genome Research*, vol. 22, no. 9, pp. 1760–1774, 2012.

[4] R. P. Horgan and L. C. Kenny, "Omic technologies: Genomics, transcriptomics, proteomics and metabolomics," *The Obstetrician & Gynaecologist*, vol. 13, no. 3, pp. 189–195, 2011.

[5] G. N. Samuel and B. Farsides, "The UK's 100,000 Genomes Project: Manifesting policymakers' expectations," *New Genetics and Society*, vol. 36, no. 4, pp. 336–353, 2017.

[6] GenomeAsia100K Consortium *et al.*, "The GenomeAsia 100K Project enables genetic discoveries across Asia," *Nature*, vol. 576, no. 7785, pp. 106–111, 2019.

[7] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott *et al.*, "The cancer genome

atlas pan-cancer analysis project," *Nature Genetics*, vol. 45, no. 10, p. 1113, 2013.

[8] S. Deorowicz, A. Danek, and M. Niemiec, "GDC 2: Compression of large collections of genomes," *Scientific Reports*, vol. 5, p. 11565, 2015.

[9] S. A. Forbes, D. Beare, P. Gunasekaran, K. Leung, N. Bindal, H. Boutselakis *et al.*, "COSMIC: Exploring the world's knowledge of somatic mutations in human cancer," *Nucleic Acids Research*, vol. 43, no. D1, pp. D805–D811, 2015.

[10] J. Zhang, J. Baran, A. Cros, J. M. Guberman, S. Haider, J. Hsu *et al.*, "International Cancer Genome Consortium Data Portal – a one-stop shop for cancer genomics data," *Database*, vol. 2011, 2011.

[11] J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim *et al.*, "The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity," *Nature*, vol. 483, no. 7391, pp. 603–607, 2012.

[12] W. Yang, J. Soares, P. Greninger, E. J. Edelman, H. Lightfoot, S. Forbes *et al.*, "Genomics of Drug Sensitivity in Cancer (GDSC): A resource for therapeutic biomarker discovery in cancer cells," *Nucleic Acids Research*, vol. 41, no. D1, pp. D955–D961, 2012.

[13] S. Huang, K. Chaudhary, and L. X. Garmire, "More is better: Recent progress in multi-omics data integration methods," *Frontiers in Genetics*, vol. 8, p. 84, 2017.

[14] Y. Li, F.-X. Wu, and A. Ngom, "A review on machine learning principles for multi-view biological data integration," *Briefings in Bioinformatics*, vol. 19, no. 2, pp. 325–340, 2018.

[15] J. Yan, S. L. Risacher, L. Shen, and A. J. Saykin, "Network approaches to systems biology analysis of complex disease: Integrative methods for multi-omics data," *Briefings in Bioinformatics*, vol. 19, no. 6, pp. 1370–1381, 2018.

[16] C. Meng, O. A. Zeleznik, G. G. Thallinger, B. Kuster, A. M. Gholami, and A. C. Culhane, "Dimension reduction techniques for the integrative analysis of multi-omics data," *Briefings in Bioinformatics*, vol. 17, no. 4, pp. 628–641, 2016.

[17] F. Rohart, B. Gautier, A. Singh, and K.-A. Lê Cao, "mixOmics: An R package for 'omics feature selection and multiple data integration," *PLoS Computational Biology*, vol. 13, no. 11, p. e1005752, 2017.

[18] M. Bersanelli, E. Mosca, D. Remondini, E. Giampieri, C. Sala, G. Castellani *et al.*, "Methods for the integration of multi-omics data: Mathematical aspects," *BMC Bioinformatics*, vol. 17, no. S2, p. S15, 2016.

[19] C. Dimitrakopoulos, S. K. Hindupur, L. Häfliger, J. Behr, H. Montazeri, M. N. Hall *et al.*, "Network-based integration of multi-omics data for prioritizing cancer genes," *Bioinformatics*, vol. 34, no. 14, pp. 2441–2448, 2018.

[20] Q. Zhao, X. Shi, Y. Xie, J. Huang, B. Shia, and S. Ma, "Combining multidimensional genomic measurements for predicting cancer prognosis: Observations from TCGA," *Briefings in Bioinformatics*, vol. 16, no. 2, pp. 291–303, 2015.

[21] Q. Mo, F. Nikolos, F. Chen, Z. Tramel, Y.-C. Lee, K. Hayashi *et al.*, "Prognostic power of a tumor differentiation gene signature for bladder urothelial carcinomas," *Journal of the National Cancer Institute*, vol. 110, no. 5, pp. 448–459, 2018.

[22] M. Cortet, A. Bertaut, F. Molinié, S. Bara, F. Beltjens, C. Coutant *et al.*, "Trends in molecular subtypes of breast cancer: Description of incidence rates between 2007 and 2012 from three French registries," *BMC Cancer*, vol. 18, no. 1, p. 161, 2018.

[23] L. Zhao, V. H. Lee, M. K. Ng, H. Yan, and M. F. Bijlsma, "Molecular subtyping of cancer: Current status and moving toward clinical applications," *Briefings in Bioinformatics*, vol. 20, no. 2, pp. 572–584, 2019.

[24] M. Hofree, J. P. Shen, H. Carter, A. Gross, and T. Ideker, "Network-based stratification of tumor mutations," *Nature methods*, vol. 10, no. 11, pp. 1108–1115, 2013.

[25] Z. He, J. Zhang, X. Yuan, Z. Liu, B. Liu, S. Tuo *et al.*, "Network based stratification of major cancers by integrating somatic mutation and gene expression data," *PloS One*, vol. 12, no. 5, 2017.

[26] B. Wang, A. M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno *et al.*, "Similarity network fusion for aggregating data types on a genomic scale," *Nature Methods*, vol. 11, no. 3, pp. 333–337, 2014.

[27] F. Zhang, C. Ren, K. K. Lau, Z. Zheng, G. Lu, Z. Yi *et al.*, "A network medicine approach to build a comprehensive atlas for the prognosis of human cancer," *Briefings in Bioinformatics*, vol. 17, no. 6, pp. 1044–1059, 2016.

[28] M. Le Morvan, A. Zinovyev, and J.-P. Vert, "NetNorM: Capturing cancer-relevant information in somatic exome mutation data with gene networks for cancer stratification and prognosis," *PLoS Computational Biology*, vol. 13, no. 6, p. e1005573, 2017.

[29] C. R. Planey and O. Gevaert, "CoINcIDE: A framework for discovery of patient subtypes across multiple datasets," *Genome Medicine*, vol. 8, no. 1, pp. 1–17, 2016.

[30] G. Yu, X. Yu, and J. Wang, "Network-aided Bi-Clustering for discovering cancer subtypes," *Scientific Reports*, vol. 7, no. 1, pp. 1–15, 2017.

[31] F. Azuaje, "Computational models for predicting drug responses in cancer research," *Briefings in Bioinformatics*, vol. 18, no. 5, pp. 820–829, 2017.

[32] N. Zhang, H. Wang, Y. Fang, J. Wang, X. Zheng, and X. S. Liu, "Predicting anticancer drug responses using a dual-layer integrated cell line-drug network model," *PLoS Computational Biology*, vol. 11, no. 9, 2015.

[33] D.-H. Le and V.-H. Pham, "Drug response prediction by globally capturing drug and cell line information in a heterogeneous network," *Journal of Molecular Biology*, vol. 430, no. 18, pp. 2993–3004, 2018.

[34] J. C. Costello, L. M. Heiser, E. Georgii, M. Gönen, M. P. Menden, N. J. Wang *et al.*, "A community effort to assess and improve drug sensitivity prediction algorithms," *Nature Biotechnology*, vol. 32, no. 12, pp. 1202–1212, 2014.

[35] M. Ammad-ud din, S. A. Khan, D. Malani, A. Murumägi, O. Kallioniemi, T. Aittokallio *et al.*, "Drug response prediction by inferring pathway-response associations with kernelized Bayesian matrix factorization," *Bioinformatics*, vol. 32, no. 17, pp. i455–i463, 2016.

[36] D. Le and D. Nguyen-Ngoc, "Multi-task regression learning for prediction of response against a panel of anti-cancer drugs in personalized medicine," in *Proceedings of the International Conference on Multimedia Analysis and Pattern Recognition*, Ho Chi Minh City, Vietnam, Apr. 2018.

**Le Duc Hau** obtained his PhD degree in Bioinformatics from University of Ulsan, Republic of Korea in 2012. He is now leading the Department of Computational Biomedicine, Vingroup Big Data Institute, VietNam. He has been focusing on proposing computational methods for disease- and drug-related problems in personalized medicine, especially on identification of disease-associated biomarkers, prediction of drug targets and response. In parallel, he has been developing bioinformatics tools. So far, he hasmore than fifty papers published in well-recognized journals and conferences, nearly a half of those are in ISI-indexed journals. In addition, he has been a member of program committees and reviewer of several international conferences/journals. Moreover, he is a principal investigator and a key member of some national/ministry-level projects. Specially, he is the principal investigator of the biggest genome project in Vietnam (i.e., building databases of genomic variants for Vietnamese population). Finally, he has been collaborating with some well-recognized international research institutes.

**Quynh Diep Nguyen** obtained her PhD degree in Information Technology from the Institute of Information Technology - The Vietnam Academy of Science and Technology in 2015. She is a lecturer at the School of Computer Science and Engineering, Thuyloi University. She has been focusing on computational methods for reconstructing the metabolic networks. So far, she has more than fifteen papers in journals and conferences published . Moreover, she is a member of some national/ministry-level projects which research on computational methods for uncovering latent knowledge from high-throughput biological data.

# Handling Imbalanced Data in Intrusion Detection Systems using Generative Adversarial Networks

Ly Vu, Quang Uy Nguyen
Le Quy Don Technical University, Hanoi, Vietnam
Correspondence: Ly Vu, vu.ly@lqdtu.edu.vn
Communication: received 24 October 2019, revised 6 February 2020, accepted 30 March 2020
Digital Object Identifier: 10.32913/mic-ict-research.v2020.n1.894

**Abstract: Machine learning-based intrusion detection has become more popular in the research community thanks to its capability in discovering unknown attacks. To develop a good detection model for an intrusion detection system (IDS) using machine learning, a great number of attack and normal data samples are required in the learning process. While normal data can be relatively easy to collect, attack data is much rarer and harder to gather. Subsequently, IDS datasets are often dominated by normal data and machine learning models trained on those imbalanced datasets are ineffective in detecting attacks. In this paper, we propose a novel solution to this problem by using generative adversarial networks to generate synthesized attack data for IDS. The synthesized attacks are merged with the original data to form the augmented dataset. Three popular machine learning techniques are trained on the augmented dataset. The experiments conducted on the three common IDS datasets and one our own dataset show that machine learning algorithms achieve better performance when trained on the augmented dataset of the generative adversarial networks compared to those trained on the original dataset and other sampling techniques. The visualization technique was also used to analyze the properties of the synthesized data of the generative adversarial networks and the others.**

**Keywords:** *Generative adversarial networks, intrusion detection system, synthesized attack, imbalanced dataset, sampling technique.*

## I. Introduction

Communication networks and information systems have become a very important factor in almost every facet of our daily lives [1, 2]. The rapid development of Internet services and communication networks has revolutionized our perspective of the world. Nevertheless, this also resulted in the information systems being very vulnerable to one or more types of cyber attacks. The security of communication networks and information systems is therefore an increasing concern for cyber security officers, network administrators and end-users. Among several approaches to protect information systems, Intrusion Detection System (IDS) plays a crucial role for early detecting of security violation [1].

IDS monitors the network traffic to find any abnormal activity. There are three popular methods for analyzing the network traffic to detect intrusive behaviors: statistical-based, machine learning-based, and knowledge-based methods [3]. Among these, machine learning-based methods have received a great attention and achieved remarkable success recently [1, 4–15] even for encrypted network traffic [16, 17]. To train a good detection model for IDS in the real world, a considerable number of attack and normal data samples are required to be gathered. Compared to normal data, intrusive data is more scarce and more expensive to collect. Thus, the collected network traffic datasets for intrusion detection are often imbalanced. Subsequently, the ability to handle imbalanced data is essential for machine learning algorithms to achieve a good accuracy in intrusion detection.

The imbalance of class samples can reduce the performance of a machine learning based IDS [18]. In this paper, we propose a novel approach to tackle the imbalanced problem of IDS datasets by using generative adversarial networks (GANs) to generate synthesized attack data. The synthesized data is then combined with the original data to form the augmented training data. Three conventional classifiers including decision tree (DT), random forest (RF), and support vector machine (SVM) are trained on the augmented dataset. The experiments were conducted on three popular IDS datasets, i.e., NSL-KDD [19], UNSW-NB15 [20], CICIDS2017 [21], and one our own dataset (i.e., RAWDATA). The results showed that machine learning algorithms achieve better performance when trained on the augmented dataset of GANs compared to those trained on the original dataset and the datasets generated by some

popular sampling techniques.

The main contribution of this paper is the demonstration of the usefulness of GANs in addressing the imbalance problem in IDS datasets. Compared to [22] where we originally proposed this technique, here we present a better version of using GANs for generating synthesized data and examine them in detail on a wide range of benchmarks. The rest of the paper is organized as follows. Section II briefly reviews the previous works in applying machine learning to IDS and the methods for dealing with imbalanced datasets. Section III presents the fundamental background of our paper. The proposed method is then described in Section IV. The tested datasets and experimental settings are provided in Section V. Section VI presents experimental results and the analysis. The conclusions and future work are discussed in Section VII.

## II. Related Work

This section presents a brief review of using machine learning in IDS and the techniques for addressing imbalanced data.

### 1. Machine learning for Intrusion Detection

Machine learning for intrusion detection has received a considerable attention in the research community [1, 4–9, 11, 12, 23, 24]. For a comprehensive review and analysis of different machine learning techniques in IDS, the readers are recommended to see [25]. In this paper, we shortly summarize recent and related research to our topic. Usually, machine learning is applied to intrusion detection to automatically build a detection model based on the training dataset. Machine learning techniques for IDS can be divided into two categories: a single and a hybrid method. The single method attempts to use only one machine learning technique to find the model which is then used to recognize whether the incoming access is a normal access or an attack [1]. The popular single learning algorithms used in the IDS include SVM, K-mean, Artificial Neural Network (ANN), RF, and DT [9–12, 26].

The hybrid method aims to combine two or more learning techniques to enhance the performance of the systems. There are several ways in which different algorithms can be hybridized. The first method is by cascading different classifiers. For example, Hussain et al. [4] proposed a two stage hybrid method using SVM as an anomaly detection in the first stage, and ANN as a misuse detection in the second stage to enhance accuracy of IDS. Aburomman et al. [23] proposed an ensemble classifiers by using the weighted majority algorithm (WMA) approach of particle swarm optimization with the SVM and K-Nearest Neighbor algorithm to enhance the accuracy of IDS. Aburomman et al. [24] also compared the effectiveness of various ensemble techniques (Boosting, Voting, and Stacking) for IDS. Other methods are based on re-sampling data samples and then taking a majority vote of the resulting weak learners [27]. Several other approaches [28, 29] aimed to collect datasets to improve the accuracy of IDS.

Recently, deep learning has been applied to improve the accuracy of IDS [13, 14, 30, 31]. Malaiya et al. [13] used Convolutional Neural Network (CNN) for detecting abnormal behaviours in the network. They used the 1D-feature of the network traffic in IDS datasets as the input of CNN to enhance the accuracy of intrusion detection. Salama et al. [30] proposed a method for the anomaly intrusion detection scheme using Restricted Bolzman Machine (RBM)-based Deep Believe Network (DBN). In their work, DBN is used as a feature reduction method and SVM is utilized as a classifier. Kim et al. [31] showed that Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) were effective for IDS problems. Kwon et al. [14] compared the accuracy of some popular deep learning models including RBN, RNN, AutoEncoder, and DBN in intrusion detection. They also proposed a Fully Connected Model (FCM) for intrusion detection. In FCM, the number of neurons in each layer is equal to the number of data features. Rodda [15] showed that the multi-layer perceptron is better than radial basis function networks for designing network intrusion detection system. Moreover, Benlenko [32] presented generative adversarial artificial neural networks to detect security intrusions in the large-scale networks of cyber Internet of Thing devices.

Overall, deep learning has often been used to extract meaningful features from intrusion data sets. In this paper, we propose a new application of deep learning to IDS. More specifically, we use a generative model (i.e., GAN) to generate synthesized attacks to handle the imbalanced problem of IDS datasets. The detailed description of our method will be presented in Section IV.

### 2. Handling Imbalanced Data in Machine Learning

Imbalanced data typically refers to a problem where the number of observations belonging to one class is significantly lower than those belonging to the other classes [33]. In this situation, the predictive model developed using conventional machine learning algorithms could be biased and inaccurate [33]. The reason is that machine learning algorithms are usually designed to improve the accuracy by reducing the error. Thus, they do not take into account the class distribution/proportion or the balance of classes. In order to improve the accuracy of machine learning in imbalanced datasets, several techniques have been proposed [33].

Perhaps, the most popular technique for dealing with imbalanced data is sampling. Sampling techniques usually consider the ratio of sample classes. Random undersampling [34] aims to downsize the majority classes by removing observations until the dataset is balanced. Random oversampling [34] decreases the level of class imbalance by copying the minority class until the classes have equal samples. The downsize of oversampling is the increase in the risk of overfitting [34]. Synthetic Minority Oversampling Technique (SMOTE) [35] generates the samples of the minor class by extrapolating and interpolating minor samples from the neighborhood ones. An extension of SMOTE is SMOTE-SVM [36] that synthesizes samples located in the borderline between classes by only generating the samples for the support vectors of a SVM model trained on the original dataset.

BalanceCascade [27] combines a sampling technique with a classifier to discover the distribution of the majority and minority class. This method develops an ensemble model of classifiers to systematically select the major samples to remove. A number of sub-datasets of balanced rate between minority and majority are created by undersampling the major class. Next, an ensemble of classifiers are trained on these sub-datasets. Finally, the samples in the major class will be removed if they are classified correctly by the ensemble model.

Some other techniques use the distance between data samples to remove the noisy or borderline samples of each class. Tomek link [37] removes samples from the major class that are close to the minor region in order to return a dataset that presents a better separation between the two classes. TomekSMOTE is a hybrid technique that uses SMOTE to oversample the minor class and apply Tomek to remove the noisy samples generated by SMOTE.

Although TomekSMOTE and SMOTE-SVM have been popularly used and their effectiveness has been well-evidenced [33, 37, 38], the shortcoming of these methods is that the distribution of the original data can be lost. In other words, the generated data samples may have a very different distribution from the original data. In this paper, we propose a method for oversampling data by using a generative deep learning model. We hypothesize that the synthesized data by the generative model will preserve the distribution of the original data better than the traditional sampling techniques. Subsequently, the augmented dataset of deep learning method will help to improve the accuracy of classification algorithms. In [39], the author also synthesized the data samples to improve the quality of IDS datasets by using a variety of Generative Adversarial Network (GAN). Their model is called SynGAN. which uses an extension of GAN, i.e., Wasserstein GAN (WGAN) [40] to synthesize data samples. The difference of SynGAN compared with our model is that SynGAN is operated in an unsupervised manner, and thus, it does not use the label information for training. Conversely, our proposed method, i.e., ACGAN-SVM uses the label information during the training process, thereby improving the accuracy of IDS.

## III. Background

This section describes in detail Generative Adversarial Network and its extension Auxiliary Classifier Generative Adversarial Network.

### 1. Generative Adversarial Network

Generative Adversarial Network (GAN) [41] is a recent deep learning method for generating synthesized data. GAN composes of two neuron networks: a Generator (G) and a Discriminator (D). The input of the generator is noise and it outputs a generated sample. The input of the discriminator includes two sources: a generated sample of the generator and a training data sample. The discriminator attempts to differentiate between a real data sample and a fake sample generated by the generator. These two networks are trained continuously, where the generator learns to generate more realistic samples, and the discriminator aims to separate the generated data from the real data. Usually, the competition between two networks will result in the generated samples being difficult to distinguish from the real samples.

Let $z$ (random noise) be the input of G and its output is a synthesized sample $X_{fake} = G(z)$. D takes as input either a real ($x$) or a fake sample ($G(z)$), and its output is a value that presents the probability of being real of the input sample. In other words, D is trained to increase the probability of the real data and to decrease the probability of the generated data by maximizing (1). Conversely, G is trained to increase the probability of the fake data being rated as the real data by minimizing the second term in this equation.

$$L = E[\log D(x)] + E[\log(1 - D(G(z)))]. \quad (1)$$

One appealing property of GAN is that it is a fully unsupervised approach. Therefore, we can train a GAN in an unsupervised manner and then use its generator and discriminator as feature extractors for supervised tasks. However, GAN is more unstable to train because we have to train two networks in an opposite way from a single back-propagation. Subsequently, the resulting generator may produce nonsensical outputs.

**Algorithm 1:** Algorithm of training G.

**Inputs**: Random noise $z$, Class label $c$ .
**Outputs**: Trained generator $G$.
**begin**
    Set input value of $G$ is $z$ concatenate $c$;
    Train $G$ by maximizing $(L_C - L_S)$;
    **return** $G$.
**end**

**Algorithm 2:** Algorithm of training D.

**Inputs**: Real data sample from original dataset $X$, Class label $c$.
**Outputs**: Trained discriminator $D$.
**begin**
    Generate random noise $z$;
    $X' = G(z, c)$;
    Set input value of $D$ is $X$ or $X'$ concatenate $c$ ;
    Train $D$ by maximizing $(L_C + L_S)$;
    **return** $D$.
**end**

## 2. Auxiliary Classifier Generative Adversarial Network

Auxiliary Classifier Generative Adversarial Network (ACGAN) [42] is an extension of GAN by using the class label in the training process. ACGAN also includes two neural networks operating in a contrary way: a Generator ($G$) and a Discriminator ($D$). The input of $G$ in ACGAN includes a random noise $z$ and a class label $c$ instead of only random noise $z$ as in the GAN model. Therefore, the synthesized sample of $G$ in ACGAN is $X_{fake} = G(c, z)$, instead of $X_{fake} = G(z)$. In other words, ACGAN can generate data samples for a desired class label. The objective function of ACGAN has two parts (in (2) and (3)): the log-likelihood of the correct data, $L_S$, and the log-likelihood of the correct class, $L_C$. $D$ is trained to maximize $L_C + L_S$ and $G$ is trained to maximize $L_C - L_S$.

$$L_S = E[\log D(x)] + E[\log(1 - D(G(z)))]. \quad (2)$$

$$L_C = E[\log D(x, c)] + E[\log(1 - D(G(z, c)))]. \quad (3)$$

Algorithms 1 and 2 describe the process of training $G$ and $D$ networks in the ACGAN model. $G$ is trained to generate fake samples that are similar to real samples. Conversely, the objective of training $D$ is to increase the difference between a real sample and a fake sample of the same class. These two networks are trained simultaneously until achieving a Nash equilibrium [43].

Similar to GAN, the training process in ACGAN using gradient descent algorithm may not be converged. Since two cost functions are updated independently, there is no guarantee to enhance the training error for both neural networks in ACGAN [43]. In this paper, we investigated various structures of $G$ and $D$ to determine the structure that achieve good performance when trained on the network traffic datasets.

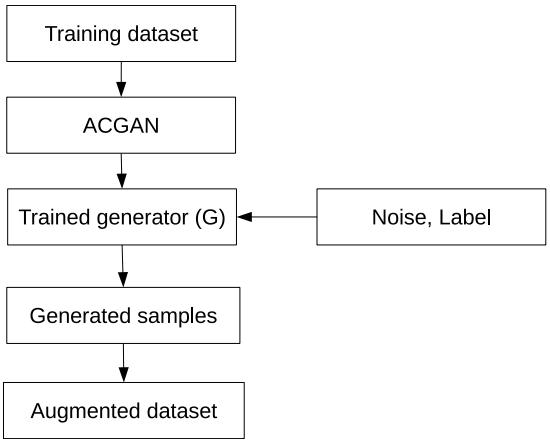


Figure 1. Process of using ACGAN to create the augmented datatset.

# IV. Methods

This section describes two approaches for generating synthesized attacks in IDS. The first approach uses an ACGAN network to generate the synthesized data. The second approach uses a SVM model to remove the noisy samples that is generated when using the ACGAN model.

## 1. Generating Synthesized Attacks using ACGAN

The flow of using ACGAN to create augmented dataset is described in Fig. 1. First, the original dataset is divided into two parts: a training set and a testing set. The ACGAN model is trained on the training set. After training, the generator ($G$) is used to generate synthesized data samples for the minor (attack) classes. The synthesized samples are then combined with the training dataset to form the augmented dataset. Three supervised classification algorithms (i.e., SVM, DT, and RF) are then trained on the augmented dataset. Finally, the supervised classification algorithms (after being trained on the augmented dataset) are tested on the testing set.

## 2. Generating Synthesized Attacks using ACGAN-SVM

The second method for generating artificial data is ACGAN-SVM. ACGAN-SVM attempts to produce samples that are near the borderline area defined by the SVM model. However, unlike SMOTE-SVM that produces data

samples by extrapolating or interpolating from the original samples [36], ACGAN-SVM uses a generative model to generate new samples. We hypothesize that the samples generated by ACGAN-SVM will preserve the underlying distribution of the original dataset better than SMOTE-SVM. Subsequently, the augmented dataset of ACGAN-SVM will improve the performance of classification algorithms.

**Algorithm 3:** ACGAN-SVM algorithm for over-sampling.

**Inputs**: original training set $X$, generated data samples $X'$, number of nearest neighbors $m$, Euclid distance between vector $x$ and vector $y$ d($x$,$y$).
**Outputs**: new sampling set $X_{new}$.
**begin**
  Training ACGAN on $X$ to have trained Generator $G$;
  Set $X'$ contains minority samples generated by $G$ network;
  Train SVM model on $X$ to have the set of support vectors $SVs$;
  **foreach** $sv_i \in SVs$ **do**
    Compute $m$ nearest neighbors in $X$;
    Compute $d_i$ that is average of Euclid distance from $m$ nearest neighbors to $sv_i$;
  **end**
  **foreach** $x_j \in X'$ **do**
    **if** $d(x_j, sv_i) \leq d_i$ **then**
      $X_{new} = X \cup \{x_j\}$;
    **end**
  **end**
  **return** $X_{new}$.
**end**

Algorithm 3 presents the detailed description of using ACGAN-SVM for generating synthesized data. The technique is divided into two main phases, i.e., generation and selection. In the generation phase, the ACGAN network is trained on the training dataset $X$. After that, the generator network ($G$) of ACGAN is used to generate synthesized samples $X'$. In the selection phase, the SVM model is trained on the training dataset $X$ and the set of support vectors of this model is called $SV_s$. For each support vector $sv_i \subset SV_s$, we calculate the average Euclidean distance $d_i$ of $m$ nearest neighbor samples of $sv_i$ in $X$ to $sv_i$. If a generated sample $x_j \subset X'$ has the distance to $sv_i$ smaller than $d_i$, this sample is kept. Conversely, if the distance from $x_i$ to $sv_i$ is greater than $d_i$, the sample is removed. The algorithm will stop when the augmented dataset is balanced for every class. The augmented dataset is then used to train three classification algorithms as in ACGAN.

## V. Experimental Settings

This section presents the datasets used in the experiments and the parameter's setting for the tested algorithms.

### 1. Datasets

In order to test the effectiveness of the proposed method we used three well-known network intrusion detection datasets, i.e., NSL-KDD, UNSW-NB15, CICIDS2017 and one our own dataset, i.e., RAWDATA.

**NSL-KDD** is an IDS dataset [19] which is used to solve some intrinsic drawbacks of the KDD'99 dataset. The NSL-KDD dataset contains 148517 records divided into the training set (125973 data samples) and the testing set (22544 data samples). Each sample has 41 features and is labeled either as a type of attack or normal data. The training set contains 24 attack types, and the testing set includes 14 more types of attack. The simulated attack samples belong to one of four categories: DoS, R2L, U2R, and Probing.

**UNSW-NB15** is created by utilizing the synthetic environment in the Cyber Range Lab of the Australian Center of Cyber Security (ACCS) [20]. The number of records in the training set and the testing set are 175341 and 82332, respectively. The data samples are labeled as normal or one of nine attack types. Nine categories of attacks includes Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. Each data sample has 49 features which were generated by using Argus, Bro-IDS tools and twelve other algorithms to analyze characteristics of network packets.

**CICIDS2017** is an IDS dataset developed by Canadian Institute for Cybersecurity. Data samples in CICIDS2017 include benign samples and some recent common attacks such as Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Port Scan, Infiltration, Botnet and [21]. We prepossessed data by dropping some attributes that do not represent the characteristic of a network flow. The removed attributes include FlowID, Source IP address, Destination IP address and time stamp. The resulting network flow is represented by 78 attributes. We also removed Heartbleed and Infiltration attacks from the dataset since they have too few samples for any algorithm to learn[1]. The final dataset used in our experiments has four categories of attacks: DoS, Brute Force, Web Attack, and Botnet and is randomly divided into the training dataset (266028 records) and the testing dataset (186213 records).

[1]Infiltration and Heartbleed have only 3 and 1 samples, respectively.

**RAWDATA** is an IDS dataset which we collected at our Computer Network Lab at Le Quy Don Technical University. First, we set up a network environment and captured three kinds of traffic including the normal traffic, the Structured Query Language (SQL) injection traffic, and the Cross-Site Scripting (XSS) traffic using the Winpcap library [44]. We used the Sqlmap tool [45] to simulate the SQL injection traffic and the XSS attack traffic was generated manually. Then, the collected packets were extracted by the Scapy library [46] in the Python language. Each packet was extracted by 500[2] raw bytes as features. Finally, we labeled the normal traffic as 0, the SQL injection traffic as 1, and the XSS traffic as 2. The final dataset was randomly divided into the training set (80% number of samples) and a testing set (20% number of samples). The processed dataset is available for download[3].

## 2. Parameters Settings

Since it is often difficult to guarantee a good convergence when training ACGAN, we have conducted a number of experiments to tune the hyper-parameters of ACGAN. The best values of hyper-parameters in ACGAN are calibrated and presented in Table I. For ACGAN-SVM, we used the same structure as ACGAN in Table I. Moreover, the SVM model with the rbf kernel and gama parameter of 0.01 was trained on the original data. For both ACGAN and ACGAN-SVM, we used the ReLu activation function for all hidden layers except the last layer where we used the Sigmoid activation function. The Adam optimization [47] with the learning rate of $10^{-3}$ was used to train $G$ and $D$ networks in ACGAN and ACGAN-SVM.

TABLE I
PARAMETERS SELECTION FOR $G$ AND $D$ NETWORKS IN ACGAN AND ACGAN-SVM.

| Dataset | Number of layers | Number of Neurons | Batch size |
|---|---|---|---|
| NSL-KDD | 5 | 160 | 128 |
| UNSW-NB15 | 5 | 160 | 64 |
| CICIDS2017 | 5 | 160 | 100 |
| RAWDATA | 5 | 160 | 50 |

After using ACGAN and ACGAN-SVM to generate the synthesized data, three popular classification algorithms, i.e., SVM, DT, and RF are trained on the augmented datasets. We used the implementation of these algorithms in a popular machine learning packet in Python, Scikit learn [48]. In order to lessen the impact of experimental parameters to the performance of the classifiers, we used the grid search technique for each algorithm. The range of values for the important parameters tuned by the grid search technique are presented in Table II.

TABLE II
PARAMETER'S RANGE OF THE GRID SEARCH FOR CLASSIFIERS.

| Classifiers | Parameters |
|---|---|
| SVM | $kernel = rbf; gama = 0.001, 0.01, 0.1, 1.0$ |
| DT | $max - depth = 5, 6, 7, 8, 9, 10, 50, 100$ |
| RF | $n - estimators = 20, 40, 80, 150$ |

For each dataset, we conducted two sets of experiments. In the first set, the traffic datasets were adapted to form two-class classification problems. In other words, the class label has two values: "Normal"("Benign") or "Attack". In the second set, the datasets were processed to form multi-class classification problems. In this case, NSL-KDD and CICIDS2017 datasets have five classes, UNSW-NB15 has ten classes, and RAWDATA has three classes. The number of samples in each class for binary and multiclass problems are described in Table III. It can be seen that these datasets are mostly balanced in the binary classification problem. However, in the multiclass classification problem, both datasets are highly imbalanced. The number of samples of some classes such as U2L in NSL-KDD and Worms, Shellcode in UNSW-NB15, and Brute Force, Botnet in CICIDS2017, SQL injection in RAWDATA are much less than the number of samples of normal class and some other attack classes.

We compared the performance of the classification algorithms when they are trained on the augmented datasets of WGAN [40], ACGAN [42], and ACGAN-SVM with the version trained on the original data and the synthesized data of three traditional sampling approaches: SMOTE-SVM [36], BalanceCasscade [27], TomekSMOTE [37]. The source code of all tested methods are available for download[4]. All techniques were implemented in Python, Scikit-learn machine learning library [48] and Tensorflow deep learning framework [49]. Moreover, the same computing platform (Operating system: Ubuntu 16.04 (64 bit), Intel(R) Core(TM) i5-5200U CPU, 2 cores and 4GB RAM memory) was used in every experiment in this paper.

## 3. Evaluation Metrics

Three popular performance metrics in a classification problem were used to measure the effectiveness of our method. The reported metrics include precision score, recall score, and F1 score [50]. Equation (4) and (5) present precision and recall score for one class. The final values of these metrics are the average over all classes.

[2]This number is a popular size of packet for our data chosen by a statistical method.

[3]https://github.com/vuthily/data/rawdata.

[4]https://github.com/vuthily/acgansvm.

TABLE III
NUMBER OF CLASS SAMPLES IN EACH TRAINING DATASET.

| NSL-KDD | | UNSW-NB15 | | CICIDS2017 | | RAWDATA | |
|---|---|---|---|---|---|---|---|
| Classes | Number | Classes | Number | Classes | Number | Classes | Number |
| Normal | 67373 | Normal | 37000 | Benign | 219110 | Benign | 6503 |
| Attack | 58630 | Attack | 45332 | Attack | 46928 | Attack | 1528 |
| DoS | 45927 | Generic | 18871 | DoS | 29447 | SQL injection | 526 |
| U2L | 52 | Exploits | 11132 | Port Scan | 15893 | XSS | 1002 |
| R2L | 995 | Fuzzers | 6062 | Brute Force | 1392 | | |
| Probing | 11656 | DoS | 4089 | Botnet | 196 | | |
| | | Reconna-issance | 3496 | | | | |
| | | Analysis | 677 | | | | |
| | | Backdoor | 583 | | | | |
| | | Shellcode | 378 | | | | |
| | | Worms | 44 | | | | |

TABLE IV
RESULT OF DT, RF, AND SVM ON TWO-CLASS CLASSIFICATION FOR NSL-KDD, UNSW-NB15, CICIDS2017, AND RAWDATA DATASETS.

| **Algorithm** | **Augmented datasets** | **NSL-KDD** | | | **UNSW-NB15** | | | **CICIDS2017** | | | **RAWDATA** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Precision** | **Recall** | **F1-score** | **Precision** | **Recall** | **F1-score** | **Precision** | **Recall** | **F1-score** | **Precision** | **Recall** | **F1-score** |
| **DT** | ORIGINAL | 0.92 | 0.90 | 0.91 | **0.84** | 0.74 | 0.80 | 0.95 | 0.95 | 0.94 | 0.97 | 0.95 | 0.96 |
| | SOMTE-SVM | 0.91 | 0.90 | 0.90 | 0.82 | 0.72 | 0.81 | 0.93 | 0.89 | 0.90 | 0.97 | 0.98 | 0.97 |
| | BalanceCascade | 0.92 | 0.91 | **0.92** | 0.81 | 0.70 | 0.82 | 0.94 | 0.92 | 0.93 | 0.96 | 0.96 | 0.95 |
| | TomekSMOTE | 0.92 | 0.92 | **0.92** | 0.81 | 0.74 | 0.83 | 0.94 | 0.92 | 0.93 | 0.97 | 0.95 | 0.96 |
| | WGAN | 0.92 | 0.90 | 0.90 | 0.80 | 0.78 | 0.76 | 0.93 | 0.92 | 0.92 | 0.97 | 0.98 | 0.97 |
| | ACGAN | 0.91 | 0.91 | 0.91 | 0.81 | 0.81 | 0.82 | 0.95 | 0.96 | 0.95 | 0.98 | 0.98 | 0.98 |
| | ACGAN-SVM | **0.93** | **0.93** | **0.92** | 0.82 | **0.84** | **0.84** | **0.97** | **0.98** | **0.96** | **0.99** | **0.99** | **0.99** |
| **RF** | ORIGINAL | **0.87** | 0.83 | 0.83 | 0.91 | 0.88 | 0.88 | 0.98 | 0.98 | 0.98 | 0.97 | 0.97 | 0.97 |
| | SMOTE-SVM | **0.87** | 0.83 | 0.83 | 0.88 | 0.82 | 0.83 | **0.99** | **0.99** | **0.99** | 0.98 | 0.97 | 0.98 |
| | BalanceCascade | **0.87** | 0.83 | 0.83 | 0.91 | 0.89 | **0.89** | **0.99** | **0.99** | **0.99** | 0.97 | 0.97 | 0.97 |
| | TomekSMOTE | 0.86 | 0.84 | 0.84 | 0.91 | 0.90 | **0.89** | **0.99** | **0.99** | **0.99** | 0.97 | 0.96 | 0.97 |
| | WGAN | 0.85 | 0.86 | 0.84 | 0.89 | 0.87 | 0.88 | 0.97 | 0.95 | 0.95 | 0.98 | 0.98 | 0.98 |
| | ACGAN | **0.87** | 0.84 | 0.84 | 0.91 | 0.88 | **0.89** | **0.99** | **0.99** | 0.98 | **0.99** | 0.97 | 0.98 |
| | ACGAN-SVM | **0.87** | **0.86** | **0.85** | **0.92** | **0.90** | **0.89** | **0.99** | **0.99** | **0.99** | **0.99** | **0.99** | **0.99** |
| **SVM** | ORIGINAL | **0.85** | 0.82 | 0.83 | **0.87** | **0.87** | 0.87 | 0.96 | 0.96 | 0.96 | 0.96 | 0.95 | 0.96 |
| | SMOTE-SVM | **0.85** | **0.83** | 0.83 | 0.84 | 0.76 | 0.77 | 0.93 | 0.91 | 0.91 | 0.97 | 0.97 | 0.97 |
| | BalanceCascade | **0.85** | **0.83** | 0.83 | 0.84 | 0.76 | 0.76 | 0.92 | 0.90 | 0.91 | 0.97 | 0.96 | 0.97 |
| | TomekSMOTE | **0.85** | 0.82 | 0.83 | 0.85 | 0.77 | 0.78 | 0.92 | 0.90 | 0.91 | 0. | 0. | 0. |
| | WGAN | **0.86** | 0.84 | **0.85** | 0.85 | 0.83 | 0.85 | **0.98** | 0.96 | 0.96 | 0.97 | 0.98 | 0.97 |
| | ACGAN | 0.85 | 0.82 | 0.83 | **0.87** | 0.85 | 0.87 | 0.97 | **0.98** | 0.97 | 0.97 | **0.99** | 0.97 |
| | ACGAN-SVM | 0.85 | **0.83** | 0.84 | **0.87** | 0.86 | **0.88** | **0.98** | 0.97 | **0.98** | **0.98** | 0.97 | **0.98** |

$$Precision = \frac{TP}{TP + FP}. \tag{4}$$

$$Recall = \frac{TP}{TP + FN}. \tag{5}$$

In (4) and (5), $TP$ and $FP$ are the number of correct and incorrect predicted samples for class $i$, respectively, and $FN$ is the number of incorrect predicted samples of the rest of the classes. Although, precision and recall are very intuitive and easy to implement, they make no distinction between classes. Therefore, they are not suitable to measure a classifier performance in imbalanced datasets.

F1-score (in (6)) aims to overcome the limitation of precision and recall score. F1-score is calculated as the mean [50] of precision and recall. F1-score is often considered as a reliable metric to evaluate the performance of classification algorithms in imbalanced datasets. Therefore, we will use this metric for comparing various techniques in this paper. Precision and recall are used as reference only.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}. \tag{6}$$

## VI. RESULTS AND DISCUSSION

This section compares the accuracy of SVM, DT, and RF when they are trained on the augmented datasets of WGAN, ACGAN, and ACGAN-SVM with those trained on the original dataset and the datasets generated by three traditional sampling approaches: SMOTE-SVM [36],

BalanceCascade [27], TomekSMOTE [37]. After that, the computational time for generating the augmented dataset of the sampling approaches is analyzed. Finally, we analyze the quality of the synthesized data and visualize the borderline samples of ACGAN, ACGAN-SVM, and the other oversampling techniques.

### 1. Accuracy of classification algorithms with synthesized attacks

The accuracy of classification algorithms on the binary and muticlass classification problems is presented in this subsection. Table IV presents the precision, recall, and F1-score of three classifiers on the binary classification problem. In this table and Table V, the best value in each configuration is printed bold face. It can be observed that all techniques for handling imbalanced data only slightly improve the accuracy of classifiers. The F1-score of SVM, DT and RF when trained on the augmented datasets of ACGAN, ACGAN-SVM and three traditional approaches is only slightly greater than that trained on the original version. The reason could be that, on the binary classification problem, the training data is mostly balanced. Therefore, using techniques for addressing imbalanced data did not help classification algorithms achieve a significant improvement.

Among the tested sampling techniques, we can see that ACGAN-SVM outperforms all others. The F1-score of the classifiers trained on the datasets of ACGAN-SVM is often higher than those trained on the datasets of the others. Moreover, the margin of the improvement of ACGAN-SVM over the original version is always greater than the margin of the improvement of the other sampling techniques. For example, using the augmented dataset of ACGAN-SVM for training, the F1-scores of DT, RF, and SVM are increased from 0.80, 0.88, and 0.87 to 0.84, 0.89, and 0.88 on UNSW-NB15 compared to using the original datatset. Another deep neural network based technique, i.e., WGAN, also provides the augmented dataset with the high accuracy of classifiers. However, it is still lower than those of the ACGAN based models. The reason is that training WGAN does not use the label of data as training the ACGAN models. For three traditional sampling techniques, the table shows that TomekSMOTE is often better than BalanceCascade and SOMTE-SVM. However, the accuracy of the classifiers trained on the datasets of TomekSMOTE is mostly equal to the accuracy of the classifiers trained on the original data.

Table V presents the F1-score, precision and recall of DT, RF, and SVM on the multiclass classification problem. It can be seen that the accuracy of the classifiers is improved considerably when trained on the datasets generated by all sampling techniques. Among three traditional sampling techniques, TomekSMOTE still achieves better results than SMOTE-SVM and BalanceCascade. Specifically, in UNSW-NB15, the F1-score of the classifiers that trained on the datasets of TomekSMOTE is often much better than that trained on the original dataset. Specifically, F1-scores of DT, RF, and SVM are improved from 0.42, 0.41, and 0.41 to 0.60, 0.72, and 0.60 when trained with the augmented datasets of TomekSMOTE compared to those trained on the original dataset. On three other datasets, i.e., NSL-KDD, CICIDS2017, and RAWDATA, the F1 scores of classifiers trained on the augmented dataset generated by TomekSMOTE are mostly equal to those trained on the original data.

The most impressive results in Table V are obtained by our proposed methods. The table shows that ACGAN and ACGAN-SVM are often considerably better than other techniques. Compared to training on the original dataset, the F1-scores of DT, RF, and SVM on UNSW-NB15 with ACGAN are increased from 0.42, 0.41, and 0.41 to 0.63, 0.73, and 0.62, respectively. The F1-score of ACGAN-SVM is often the highest value among all tested techniques. The improvement margin of ACGAN-SVM over the original version is often greater than that of ACGAN and TomekSMOTE. For example, the F1-scores of DT, RF, and SVM on UNSW-NB15 with ACGAN-SVM are increased to 0.67, 0.74, and 0.63 compared to those of the original dataset. On the CICIDS2017 dataset, only ACGAN-SVM achieves better performance than the original data. Additionally, on the RAWDATA, because the imbalance rate is relatively low, the classification accuracy of handling imbalance techniques is slightly improved compared with those of the original data.

The results in Table V give evidence for the benefit of using ACGAN and ACGAN-SVM to improve the machine learning performance in IDSs when the training datasets are highly imbalanced. In other words, using ACGAN and ACGAN-SVM to generate synthesized attacks enhances the quality of the IDS training datasets. This subsequently improves the effectiveness of the machine learning algorithms when they are trained on the augmented datasets of ACGAN and ACGAN-SVM. The reason for the better performance of ACGAN and ACGAN-SVM compared to others could be that the synthesized samples of the traditional sampling techniques may not fully follow the original data distribution and that this problem is mitigated in the generative models (see Subsection VI.3). Moreover, using SVM to remove unimportant samples which do not contribute much to the classification algorithms helps ACGAN-SVM to further improve the performance of classifiers over using ACGAN.

TABLE V
RESULT OF DT, RF, AND SVM ON MULTICLASS CLASSIFICATION FOR NSL-KDD, UNSW-NB15, CICIDS2017, AND RAWDATA DATASETS .

| Algorithm | Augmented datasets | NSL-KDD | | | UNSW-NB15 | | | CICIDS2017 | | | RAWDATA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1-score | Precision | Recall | F1-score | Precision | Recall | F1-score | Precision | Recall | F1-score |
| **DT** | ORIGINAL | 0.87 | 0.83 | 0.82 | 0.51 | 0.50 | 0.42 | 0.97 | 0.97 | 0.97 | 0.97 | 0.95 | 0.95 |
| | SMOTE-SVM | 0.85 | 0.80 | 0.82 | 0.49 | 0.47 | 0.46 | 0.92 | 0.85 | 0.87 | **0.98** | 0.95 | 0.96 |
| | BalanceCascade | 0.80 | 0.81 | 0.79 | 0.63 | 0.61 | 0.58 | 0.95 | 0.94 | 0.94 | 0.97 | 0.95 | 0.96 |
| | TomekSMOTE | 0.86 | 0.82 | 0.82 | 0.65 | 0.62 | 0.60 | 0.94 | 0.93 | 0.93 | 0.95 | **0.97** | 0.96 |
| | WGAN | 0.83 | 0.80 | 0.78 | 0.64 | 0.60 | 0.61 | 0.96 | 0.95 | 0.94 | 0.96 | 0.96 | 0.96 |
| | ACGAN | 0.89 | **0.85** | 0.83 | 0.66 | 0.61 | 0.63 | 0.98 | 0.97 | 0.97 | 0.96 | 0.96 | 0.96 |
| | ACGAN-SVM | **0.90** | 0.84 | **0.84** | **0.67** | **0.63** | **0.67** | **0.98** | **0.98** | **0.98** | 0.97 | **0.97** | **0.97** |
| **RF** | ORIGINAL | 0.81 | 0.76 | 0.72 | 0.46 | 0.54 | 0.41 | **0.99** | 0.99 | 0.98 | 0.97 | 0.96 | 0.96 |
| | SMOTE-SVM | 0.82 | 0.78 | 0.74 | 0.74 | 0.70 | 0.71 | **0.99** | 0.95 | **0.99** | 0.98 | **0.98** | **0.98** |
| | BalanceCascade | 0.83 | 0.79 | 0.77 | 0.82 | 0.67 | 0.70 | 0.98 | 0.98 | 0.98 | 0.97 | **0.98** | 0.97 |
| | TomekSMOTE | 0.88 | 0.80 | 0.77 | **0.84** | 0.69 | 0.72 | **0.99** | **0.99** | **0.99** | 0.97 | **0.98** | 0.97 |
| | WGAN | 0.81 | 0.76 | 0.84 | 0.69 | 0.70 | 0.72 | 0.96 | 0.97 | 0.96 | 0.98 | **0.98** | **0.98** |
| | ACGAN | 0.84 | 0.79 | 0.78 | 0.69 | 0.72 | 0.73 | **0.99** | **0.99** | **0.99** | 0.98 | 0.97 | **0.98** |
| | ACGAN-SVM | **0.84** | **0.82** | **0.80** | 0.83 | **0.76** | **0.74** | **0.99** | **0.99** | **0.99** | **0.99** | **0.98** | **0.98** |
| **SVM** | ORIGINAL | 0.67 | 0.74 | 0.69 | 0.46 | 0.55 | 0.41 | 0.93 | 0.92 | 0.92 | 0.94 | 0.94 | 0.94 |
| | SMOTE-SVM | 0.81 | 0.77 | 0.75 | 0.65 | 0.52 | 0.54 | 0.92 | 0.84 | 0.86 | **0.96** | 0.94 | 0.95 |
| | BalanceCascade | 0.70 | 0.69 | 0.63 | 0.75 | 0.61 | 0.59 | 0.93 | 0.81 | 0.85 | 0.95 | 0.94 | 0.95 |
| | TomekSMOTE | 0.81 | 0.78 | 0.68 | 0.71 | 0.63 | 0.60 | **0.94** | 0.85 | 0.88 | 0.94 | 0.95 | 0.95 |
| | WGAN | **0.83** | **0.80** | **0.84** | 0.75 | 0.63 | 0.62 | **0.94** | 0.93 | 0.92 | **0.96** | 0.95 | **0.96** |
| | ACGAN | 0.81 | 0.74 | 0.76 | 0.76 | 0.65 | 0.62 | **0.94** | 0.94 | 0.93 | **0.96** | **0.96** | **0.96** |
| | ACGAN-SVM | **0.83** | 0.79 | 0.78 | **0.78** | **0.66** | **0.63** | **0.94** | **0.95** | **0.95** | **0.96** | **0.96** | **0.96** |

TABLE VI
COMPUTATIONAL TIME (IN SECONDS) TO SYNTHESIZE DATA OF SAMPLING TECHNIQUES.

| Methods | NSL-KDD | | UNSW-NB15 | | CICIDS2017 | | RAWDATA | |
|---|---|---|---|---|---|---|---|---|
| | b-classes | m-classes | b-classes | m-classes | b-classes | m-classes | b-classes | m-classes |
| SMOTE-SVM | 280 | 2266 | 645 | 4026 | 3310 | 3439 | 312 | 155 |
| BalanceCascade | 256 | 5 | 33 | 3 | 413 | 4 | 232 | 137 |
| TomekSMOTE | 386 | 1651 | 88 | 503 | 1251 | 5946 | 342 | 901 |
| WGAN | 5700 | 5287 | 6230 | 5893 | 5902 | 6101 | 6003 | 6101 |
| ACGAN | 7200 | 6884 | 7508 | 6280 | 6285 | 6348 | 5974 | 6048 |
| ACGAN-SVM | 7935 | 7725 | 8276 | 7198 | 7304 | 7122 | 7502 | 7107 |

## 2. Computational Time of Sampling Approaches

This subsection compares the computational time of the sampling methods. The computational time in seconds of five sampling techniques to generate the augmented datasets is showed in Table VI. It can be seen that the computational time of BalanceCascade and TomekSMOTE are relatively small while these values of others are often much higher. The running time of ACGAN-SVM is always greater than the time of all other approaches. The reason is that this method needs to train both the ACGAN model and the SVM model before it is used to create the synthesize data. However, the sampling phase can be considered as the pre-processing data step. When applying a machine learning model to a real world problem like IDS, the computational time for prediction is often the most desirable and the sampling step does not affect the prediction time of the classifiers.

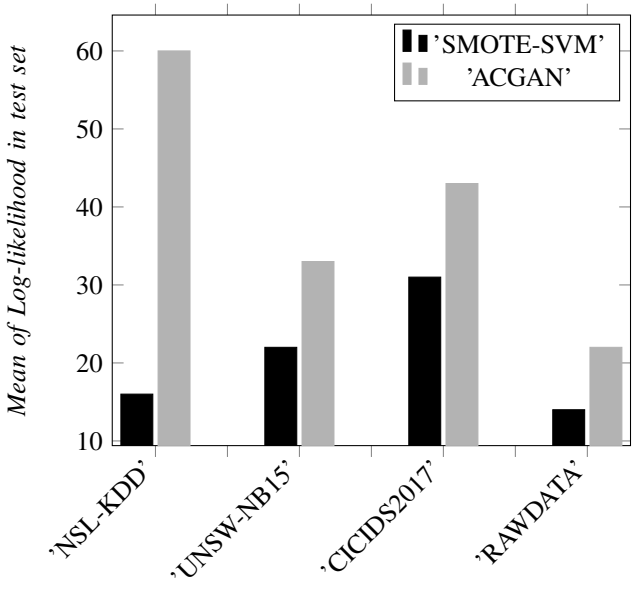


Figure 2. Parzen window log-likelihood of test set.

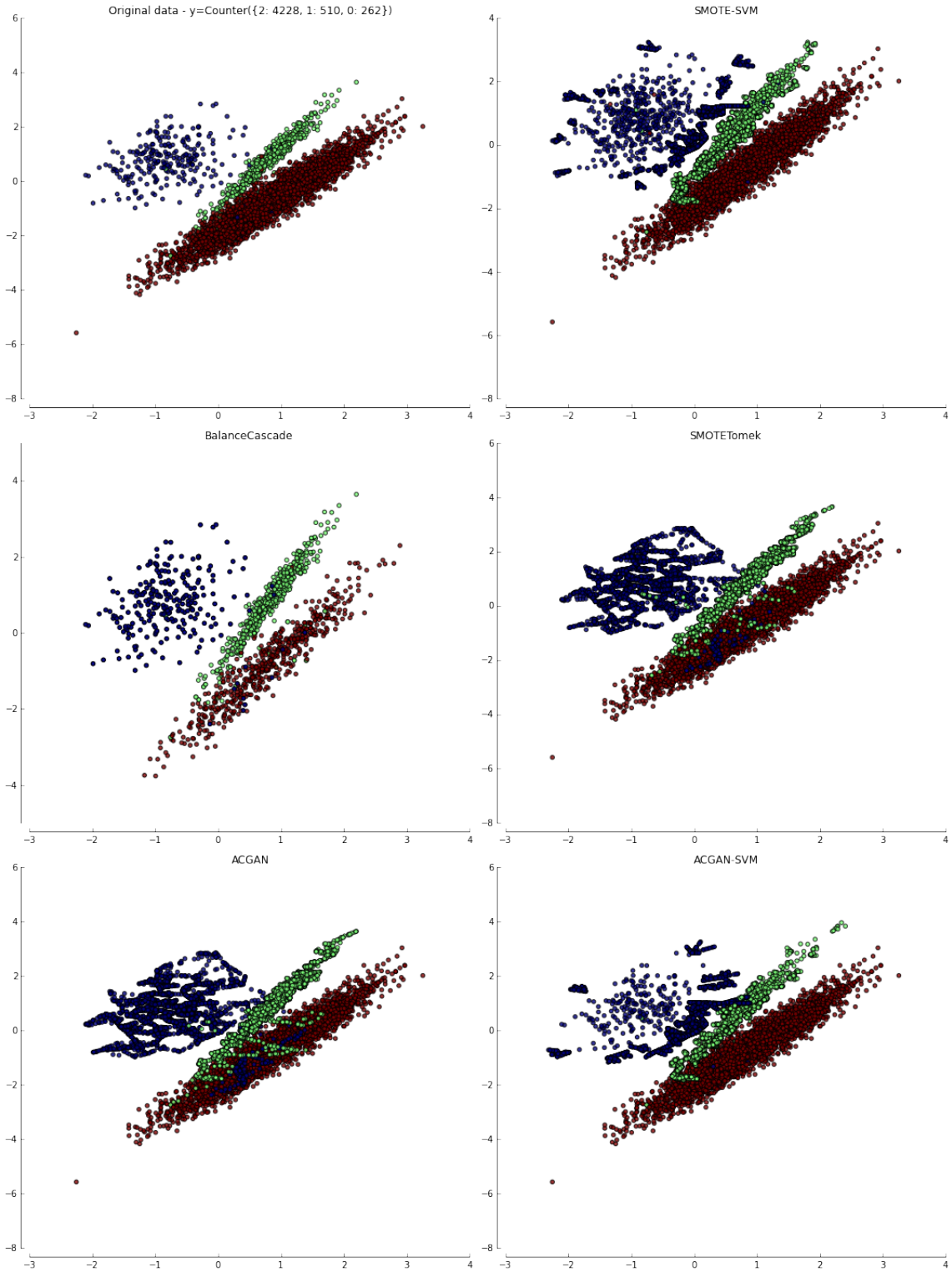


Figure 3. Illustration of oversampling techniques.

## 3. Analysis of Generated Data

This subsection qualitatively evaluates the generated data of SMOTE-SVM and ACGAN by measuring whether this data converges to the true data distribution. This experiment is applied to assess generated data techniques by oversampling techniques. Thus, we just assess two sampling techniques such as SMOTE-SVM and ACGAN. We applied

the procedure of Goodfellow et al. [41] to assess the quality of generative models. For each dataset, we use a Gaussian Parzen window to fit datasets. Then, the log-likelihood of each generative model under the true distribution is reported. Experimental results are shown in Fig. 2 where a greater value presents a better model.

It can be seen that the log-likelihood of ACGAN is always higher than the value of SMOTE-SVM on three tested datasets. The reason is that ACGAN are trained to learn the true distribution of the original data while SMOTE-SVM does not do so. The figure evidences that the generated data of ACGAN correlates more strongly to the original data than the data of SMOTE-SVM. This result supports for the better performance of machine learning algorithms when they are trained on the augmented datasets of ACGAN compared to those trained on the augmented datasets of SMOTE-SVM.

### 4. Borderline Samples Visualization

This subsection visualizes the data synthesized by ACGAN, ACGAN-SVM, and other sampling techniques. We created a random dataset of 5000 samples with two features. The dataset includes three classes: two minor classes and one major class. The ratio of samples in each class is 0.05 : 0.10 : 0.85. This dataset is visualized on the top left of Fig. 3. In this figure, the red dots, blue dots, and green dots are the data samples of the major class, the first and the second minor classes, respectively. Using ACGAN, we generated 3966 samples for the first minor class and 3718 samples for the second minor class. Totally, 7684 samples of two minor classes were generated to form a balanced dataset. Similarly, SMOTE-SVM and ACGAN-SVM were used to generate 7684 samples for two minor classes. BalanceCascade reduced the number of samples of the major class to 524 and oversampled the most minor class to 524 to balance with the middle class. For TomekSMOTE, we oversampled two minor classes to have 4163 and 4158 samples, respectively. After that, 117 samples in the major class is reduced to form the augmented dataset. The generated datasets of all sampling techniques are presented in Fig. 3.

It can be seen from Fig. 3 that BalanceCascade is only the technique that reduces the number of data samples. All other techniques generate synthesized data from the original data. Among four oversampling techniques, we can see that both ACGAN and TomekSMOTE generate many samples that are in the middle area of the minor classes. Conversely, SMOTE-SVM and ACGAN-SVM only generate the samples that are near the borderline between classes. In the context of sampling techniques, the samples that are near the borderline between classes often contribute more significantly to the effectiveness of the classifiers than the samples located in the center area.

Moreover, Fig. 3 also shows that TomekSMOTE and ACGAN sometimes create the samples that are overlapped with the samples of other classes. Subsequently, these samples will be difficult for the classifiers to separate correctly. Conversely, the generated samples of SMOTE-SVM and ACGAN-SVM are not overlapped with other classes. This evidences that using SVM helps to remove the noisy samples generated by ACGAN and SMOTE. This provides partial explanation for the better performance of ACGAN-SVM compared to ACGAN and others. Moreover, the superior performance of ACGAN-SVM and ACGAN to other sampling techniques could be that the generated samples of ACGAN and ACGAN-SVM are better follow the original distribution than the traditional techniques as analyzed in Subsection VI.3.

Overall, the results in this section show that Generative Adversarial Networks can generate the meaningful samples for imbalanced IDS datasets. The classification algorithms that are trained on datasets augmented by ACGAN and particularly ACGAN-SVM are often better than those trained on the original dataset and the datasets obtained by using some popular sampling techniques.

## VII. Summary

In this paper, we proposed a novel approach based on generative adversarial networks for addressing imbalanced datasets in IDS. Specifically, we proposed two techniques based on ACGAN and ACGAN-SVM to generate samples for the attack classes in IDS. The augmented datasets of ACGAN and ACGAN-SVM are then used as the training dataset for three popular classification algorithms, SVM, DT, and RF. The experiments were conducted on three common IDS datasets: NSL-KDD, UNSW-NB15, and CICIDS2017 and one our own dataset, i.e., RAWDATA. The results show that the augmented datasets of ACGAN and ACGAN-SVM help machine learning to enhance accuracy on the imbalanced datasets although the training processes of ACGAN and ACGAN-SVM are often slower than the traditional sampling approaches. We analysed the quality of the generated data of ACGAN and SMOTE-SVM and visualized the borderline synthesized samples of five tested sampling techniques. The visualization partially explains the better performance of ACGAN and particularly ACGAN-SVM compared to others.

There are a number of research areas for future work that arise from this paper. First, we would like to examine the effectiveness of other deep learning generative models such as auto-encoder in generating the synthesized attacks for IDS. Second, the visualization technique has shed some

light on the superior performance of ACGAN-SVM to other sampling techniques. However, this technique did not completely explain why ACGAN is also often better than SMOTE-SVM. We hypothesize that the generated data of ACGAN is better follow the original distribution than SMOTE-SVM. In the future, we will study the method to quantify the distribution of the synthesized samples of these sampling techniques [41]. Last but not least, we want to extend this approach to other problems in security and in other areas such as anomaly detection.

## ACKNOWLEDGEMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2019.05.

## REFERENCES

[1] M. Albayati and B. Issac, "Analysis of intelligent classifiers and enhancing the detection accuracy for intrusion detection system," *Int. J. Comput. Intell. Syst.*, vol. 8, pp. 841–853, 2015.

[2] G. T. Nguyen, B. M. Nguyen, D. Tran, and L. Hluchý, "A heuristics approach to mine behavioural data logs in mobile malware detection system," *Data Knowl. Eng.*, vol. 115, pp. 129–151, 2018.

[3] X. Jing, Z. Yan, and W. Pedrycz, "Security data collection and data analytics in the internet: A survey," *IEEE Communications Surveys & Tutorials*, vol. Accepted Manuscript, 2018.

[4] J. Hussain, S. Lalmuanawma, and L. Chhakchhuak, "A two-stage hybrid classification technique for network intrusion detection system," *Int. J. Comput. Intell. Syst.*, vol. 9, pp. 863–875, 2016.

[5] D. A. Effendy, K. Kusrini, and S. Sudarmawan, "Classification of intrusion detection system (ids) based on computer network," *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 90–94, 2017.

[6] B. Xu, S. Chen, H. Zhang, and T. Wu, "Incremental k-nn svm method in intrusion detection," in *International Conference in Software Engineering and Service Science (ICSESS)*, 2017, pp. 712–717.

[7] A. Hadri, K. Chougdali, and R. Touahni, "Intrusion detection system using pca and fuzzy pca techniques," *2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS)*, pp. 1–7, 2016.

[8] A. R. Syarif and W. Gata, "Intrusion detection system using hybrid binary pso and k-nearest neighborhood algorithm," in *IEEE conference in Information and Communication Technology and System (ICTS)*, 2017, pp. 181–186.

[9] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. C. Atkinson, and X. J. A. Bellekens, "A taxonomy and survey of intrusion detection system design techniques, network threats and datasets," *CoRR*, vol. abs/1806.03517, 2018.

[10] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system," *Expert Syst. Appl.*, vol. 67, pp. 296–303, 2017.

[11] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Syst. Appl.*, vol. 42, pp. 2670–2679, 2015.

[12] B. W. Masduki, K. Ramli, F. A. Saputra, and D. Sugiarto, "Study on implementation of machine learning methods combination for improving attacks detection accuracy on intrusion detection system (ids)," *2015 International Conference on Quality in Research (QiR)*, pp. 56–64, 2015.

[13] R. K. Malaiya, D. Kwon, J. Kim, S. C. Suh, H. Kim, and I. Kim, "An empirical evaluation of deep learning for network anomaly detection," in *2018 International Conference on Computing, Networking and Communications, ICNC*, 2018, pp. 893–898.

[14] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, pp. 1–13, 2017.

[15] S. Rodda, "Network intrusion detection systems using neural networks," in *Information Systems Design and Intelligent Applications*. Springer, 2018, pp. 903–908.

[16] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE communications magazine*, vol. 57, no. 5, pp. 76–81, 2019.

[17] P. Wang, X. Chen, F. Ye, and Z. Sun, "A survey of techniques for mobile service encrypted traffic classification using deep learning," *IEEE Access*, vol. 7, pp. 54 024–54 033, 2019.

[18] S. Rodda and U. S. R. Erothi, "Class imbalance problem in the network intrusion detection systems," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. IEEE, 2016, pp. 2685–2688.

[19] M. Tavallaee, E. Bagheri, W. Lu, , and A. A. Ghorbani, "Nsl-kdd data set for network-based intrusion detection systems," http://nsl.cs.unb.ca/NSL-KDD/, 2009, accessed: 2018-04-10.

[20] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference, MilCIS 2015, Canberra, Australia, November 10-12, 2015*, 2015, pp. 1–6.

[21] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *International Conference on Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116.

[22] L. Vu, B. C. Thanh, and Q. U. Nguyen, "A deep learning based method for handling imbalanced problem in network traffic classification," in *Proceeding of Symposium on Information and Communication Technology*, 2017, pp. 333–339.

[23] A. A. Aburomman and M. B. I. Reaz, "A novel svm-knn-pso ensemble method for intrusion detection system," *Appl. Soft Comput.*, vol. 38, pp. 360–372, 2016.

[24] A. AminAburomman and M. B. IbneReaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," *Computers & Security*, vol. 65, pp. 135–152, 2017.

[25] E. Hodo, X. J. A. Bellekens, A. Hamilton, C. Tachtatzis, and R. C. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," *CoRR*, vol. abs/1701.02145, 2017.

[26] Akashdeep, I. Manzoor, and N. Kumar, "A feature reduced intrusion detection system using ANN classifier," *Expert Syst. Appl.*, vol. 88, pp. 249–257, 2017.

[27] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory under-sampling for class-imbalance learning," *Sixth International Conference on Data Mining (ICDM'06)*, pp. 965–969, 2006.

[28] C. G. Cordero, E. Vasilomanolakis, A. Wainakh, M. Mühlhäuser, and S. Nadjm-Tehrani, "On generating network traffic datasets with synthetic attacks for intrusion detection," *arXiv preprint arXiv:1905.00304*, 2019.

[29] L. Arnaboldi and C. Morisset, "Generating synthetic data for real world detection of dos attacks in the iot," in *Federation of International Conferences on Software Technologies: Ap-*

*plications and Foundations*. Springer, 2018, pp. 130–145.
[30] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, "Hybrid intelligent intrusion detection scheme," *Soft Computer Industry Application*, pp. 193–303, 2011.
[31] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *International Conference on Platform Technology and Service (PlatCon)*, 2016, pp. 1–5.
[32] V. Belenko, V. Chernenko, M. Kalinin, and V. Krundyshev, "Evaluation of gan applicability for intrusion detection in self-organizing networks of cyber physical systems," in *2018 International Russian Automation Conference (RusAutoCon)*. IEEE, 2018, pp. 1–7.
[33] A. D. Pozzolo, O. Caelen, S. Waterschoot, and G. Bontempi, "Racing for unbalanced methods selection," in *Intelligent Data Engineering and Automated Learning - IDEAL 2013*, 2013, pp. 24–31.
[34] C. Drummond and R. Holte, "C4.5 class imbalance, and cost sensitivity: why under-sampling beats over-sampling," *Workshop on Learning from Imbalanced Data Sets II*, vol. 11, pp. 1–8, 2003.
[35] K. W. Bowyer, N. V. Chawla, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
[36] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," *International Journal of Knowledge Engineering and Soft Data Paradigms (IJKESDP)*, vol. 3, no. 1, pp. 4–21, 2011.
[37] A. Namvar, M. Siami, F. Rabhi, and M. Naderpour, "Credit risk prediction in an imbalanced social lending environment," *Int. J. Comput. Intell. Syst.*, vol. 11, no. 1, pp. 925–935, 2018.
[38] Q. Wang, Z. Luo, J. Huang, Y. Feng, and Z. Liu, "A novel ensemble method for imbalanced data learning: Bagging of extrapolation-smote SVM," *Comp. Int. and Neurosc.*, vol. 2017, pp. 1 827 016:1–1 827 016:11, 2017.
[39] J. Charlier, A. Singh, G. Ormazabal, R. State, and H. Schulzrinne, "Syngan: Towards generating synthetic network attacks using gans," *arXiv preprint arXiv:1908.09899*, 2019.
[40] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.
[41] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial networks," in *Conference on Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.
[42] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 2642–2651.
[43] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016, pp. 2226–2234.
[44] Winpcap, "The industry standard windows packet library," https://www.winpcap.org/default.htm, online; accessed 20 December 2019.
[45] SQL map, "Automatic SQL injection and database takeover tool," http://sqlmap.org/, online; accessed 20 December 2019.
[46] Scapy library, "Packet crafting for Python2 and Python3," https://scapy.net/, online; accessed 20 December 2019.
[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[49] G. Research, "Tensorflow tutorial," https://www.tensorflow.org/, 2015, accessed: 2018-04-24.
[50] D. M. W. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine learning Technologies*, vol. 2, pp. 37–63, 2011.

**Ly Vu** received her B.Sc. and M.Sc. degrees in computer science from Le Quy Don Technical University (LQDTU), Vietnam and Inha University, Korea, respectively. She is currently pursuing the Ph.D. degree with LQDTU. She was a Lecturer with Le Quy Don Technical University. Her research interests include data mining, machine learning, deep learning, network security.

**Quang Uy Nguyen** received B.Sc. and M.Sc. degree in computer science from Le Quy Don Technical University (LQDTU)), Vietnam and the PhD degree at University College Dublin, Ireland. Currently, he is a senior lecturer at LQDTU and the director of Machine Learning and Applications research group at LQDTU. His research interest includes Machine Learning, Computer Vision, Information Security, Evolutionary Algorithms and Genetic Programming.

# Controlling Contention Window to Ensure QoS for Multimedia Data in Wireless Network

Ngo Hai Anh, Pham Thanh Giang
Institute of Information Technology, Vietnam Academy of Science and Technology
Correspondence: Ngo Hai Anh. Email: ngohaianh@ioit.ac.vn
Communication: received 8 November 2019, revised 11 August 2020, accepted 1 September 2020
Digital Object Identifier: 10.32913/mic-ict-research.v2020.n1.896

**Abstract: The IEEE-802.11e standard was published with the goal of ensuring quality of service, especially with multimedia data. However, this standard only assigns different priorities for different types of data but does not control the sharing of bandwidth among different data flows. In this paper, we will propose a method of sharing bandwidth for proportional data flows and controlling Contention Window of each priority flow to achieve that ratio.**

**Keywords:** *Ad hoc network, IEEE 802.11e EDCA, service differentiation, traffic differentiation.*

## I. INTRODUCTION

For the purpose of providing QoS for multimedia data, IEEE-802.11e was introduced in 2005 [1] and approved in 2012 [2]. The work was done with the objective of ensuring quality of service for multimedia data. However, this standard has not yet ensured on-demand differentiation between flow's traffics. For example, the ratio between data types such as voice, video, and best-effort is always kept at the highest level for voice, and the lowest level for best-effort. But this fixed ratio does not meet the flexible proportional division, in saturation state, high priority Traffic Class (TC) flows often gets all bandwidth, and low priority TC flows are hard to access channels.

Some studies have shown that although IEEE-802.11e has focused on improving QoS, there are still many issues to be resolved [3, 4]. The research [3] investigates EDCA performance in presence of voice and data integrated traffic, and observes the inefficiencies that arise when access differentiation is performed on the basis of traffic type only. In [4], authors investigate the use of the 802.11e MAC EDCF to address transport layer unfairness in WLAN.

In this paper, we will perform the evaluation by simulation to show that although IEEE 802.11e can provide proportional bandwidth distribution for different types of multimedia data, that division is fixed. For example, voice data is always the largest ratio and background data is the smallest. Therefore, in some cases, for example real–time data, which provides service differentiation for best effort and variable-rate real-time traffic, 802.11e cannot provide adequate QoS [5]. And hence it is necessary to have a more flexible division mechanism. We will solve the problem by measuring actual data each node received over a period of time, and then comparing it to theoretical data to determine whether to increase or decrease the Contention Window (CW) value. Our proposed algorithm will control the increase or decrease of this CW value to achieve flexible ratio to adapt user's on-demand with various types of data such as voice, video and background.

Next parts of this paper include the following sections: Section 2 summarizes some related research around the issue we are going to solve; In Section 3, we analyze the lack of 802.11e in adapting different throughput ratio for different data types; Section 4 and 5 will describe our proposed solution, the methods used to accomplish our goal and the simulation scenario evaluating the results achieved according to the outlined criteria.

## II. RELATED WORKS

Some research papers focus on designing a Medium Access Control (MAC) and channel usage algorithm in multi-rate Wireless Local Area Networks (WLANs) for improving efficiency and sharing fairly channel resources among contending nodes [6]. Aiming to address problems that the high collision is often caused by Binary Exponential Backoff (BEB) mechanism in the legacy IEEE-802.11 and the shared channel can be overused by low bit-rate nodes, authors proposed a Differentiated Reservation (DR) algorithm to reduce the collision among contending nodes by setting their back-off counter as a deterministic value once successfully accessing the channel. Furthermore, to eliminate the performance anomaly, some nodes are permitted to send multiple packets in one transmission opportunity according to their feature. Moreover, the al-

gorithm presented the implementation of the DR algorithm that is readily applied to both the existing IEEE 802.11 DCF and 802.11e EDCA networks with the minimum modification. In addition, we also investigate the limitation of DR algorithm and propose a Group-Based Differentiated Reservation (GDR) algorithm applied to high density scenarios. The results of theoretical analysis and simulation validate that the proposed algorithms (DR and GDR) can obtain high throughput, good airtime fairness and low collision rate.

In the research about a fair access mechanism in IEEE-802.11e networks [7], some proposed mechanisms are evaluated. According to their evaluated strengths and weaknesses, authors proposed a new mechanism for TXOP determination. This algorithm considers data rate, channel error rate and data packet lengths to calculate adaptive TXOPs for the stations. The simulation results show that the proposed algorithm leads to better fairness. It also achieves higher throughput and lower delays in the network.

Some previous paper on IEEE-802.11 QoS has proved that 802.11 parameters can be adjusted to get better network performance by maximizing throughput based on current network situations [8] [9]. 802.11 also allows parameter tuning such that some services can be accommodated and provided some QoS guarantees [10]. But these approaches do not provide different priority levels then lead to the development of 802.11e which has been done to dynamically adjust 802.11e parameters to provide better performance and QoS.

In research [11] the effects of different $CW$ sizes are explored. This work shows that $CW_{min}$ default in 802.11 leads to worse network performance, and finds that in situations with small numbers of stations, $CW_{min}$ with small values greatly increased the collision probability, reduced overall throughput, and increased delays. However, when the network is fully utilized, the probability of transmission when the medium was free of a station also increases because the number of stations was increased. This research also finds out that when a single station is transmitting, throughput is greatly increased when $CW_{min}$ values are small since a physical transmission medium is less idle time.

The effect of controlling dynamically to QoS prioritization was presented in [12]. The authors show that IEEE-802.11e designation increases the possibility of collisions and increases delay by adding an extra contention layer. In normal IEEE-802.11 networks the only collisions which can occur are those between stations as they try to gain access to the medium to transmit the packet at the head of the single send queue. In contrast with 802.11, there is only one type of data, because 802.11e divides into multiple classes of data and each class has its own queue, then virtual collisions can occur when each Traffic Class (TC) queue must contend for access within the station's queue manager, while real collisions occur when the winning queue is allowed to transmit. The results in [12] show that throughput in 802.11e is decreased and latency is increased when compared to 802.11a networks due to this extra contention. Hence, dynamic and aggressive tuning of the network parameters is required to achieve benefit from prioritization.

In the paper [13], authors extend the approach for the IEEE-802.11e network, where different QoS classes are defined and show how to find the class-specific optimal Contention Window sizes that yield the maximum aggregate throughput while maintaining the target throughput difference between classes. However, this approach is mainly based on mathematical models with assumptions that are difficult to realize by simulation or in practice.

Another method uses a two-level protection and guarantees a mechanism for voice and video traffic in IEEE 802.11e wireless LANs [14]. In first-level protection, existing voice and video flows are protected from new and other existing voice and video flows. In second-level protection, the voice and video flows are protected from best-effort data traffic. For each protection level, a couple of protection mechanisms are proposed. Extensive simulation results show that the proposed two-level protection and guaranteed mechanism are very effective in terms of protecting and guaranteeing existing voice and video flows as well as fully utilizing the channel capacity

IEEE-802.11e parameters can be tuned based on network conditions to allow better performance than a single setting [15]. Although these settings are not changed dynamically in this study, they do show that changing network conditions require changing parameters to use the channel efficiently. These optimizations are evaluation in a testbed under realistic conditions. The authors investigate two methods of choosing $CW_{min}$ in IEEE-802.11e networks based on proportional fairness and time-based fairness. They conclude that proportional fairness in a network based on weights provides higher throughput than time based fairness. Their work shows that $CW_{min}$ is an important parameter and can be tailored to a network in the case of nodes with or without ensuring fairness or traffic differentiation.

From the above research studied, we find that sharing of traffic by different data types without permanently assigning QoS parameters will be more effective for multimedia data, which is the type of data that always varies according to users' needs. The next sections of the paper will present our approach to achieving that goal.

## III. ANALYZING OF EDCA THROUGHPUT

TABLE I
USER PRIORITY AND ACCESS CATEGORY [2].

| Priority | UP | AC | Designation |
|---|---|---|---|
| lowest | 1 | AC_BK | Background |
| - | 2 | AC_BK | Background |
| - | 0 | AC_BE | Best effort |
| - | 3 | AC_BE | Best effort |
| - | 4 | AC_VI | Video |
| - | 5 | AC_VI | Video |
| - | 6 | AC_VO | Voice |
| highest | 7 | AC_VO | Voice |

EDCA parameters for each AC are shown in the Table III.

TABLE II
CALCULATION OF CONTENTION WINDOW BOUNDARIES.

| AC | CWmin | CWmax | AIFSN | TXOP limit (ms) |
|---|---|---|---|---|
| AC_BK | 15 | 1023 | 7 | 0 |
| AC_BE | 15 | 1023 | 3 | 0 |
| AC_VI | 7 | 15 | 2 | 3.008 |
| AC_VO | 3 | 7 | 2 | 1.504 |

IEEE 802.11e introduced some enhancements to existing DCF and PCF of IEEE 802.11 with Hybrid Coordination Function (HCF) [1] [2]. The HCF includes two mechanism: Enhanced Distributed Channel Access (EDCA), which is an enhanced DCF, and HCF Controlled Channel Access (HCCA), which is an enhanced PCF. Both EDCA and HCCA could work separately or together, in contract of 802.11, where DCF is mandatory and PCF is optional. While the fundamentals of the original functions were not changed, augmented information allows HCF to provide QoS to specific flows and/or stations. In EDCA, MAC-layer parameters provide priority to each *Traffic Class* (TC) in a contention access manner similar to DCF. The parameters that can be manipulated are the *Arbitration Inter-Frame Space* (AIFS), the *Transmission Opportunity* (TXOP), the $CWmin$, and the $CWmax$. These parameters are given default values at each station for each TC, or they can be overridden by an Access Point using special coordination frames.

EDCA uses differentiated, distributed and contention-based medium access mechanism to enhance the original DCF by providing prioritized QoS for each type of data flow. EDCA mechanism defines four *Access Categories* (ACs) to provide support for delivery of traffic with *User Priorities* (UPs) at the stations. There are four traffic types corresponding to the four ACs: AC_BK, AC_BE, AC_VI and AC_VO stand for background, best effort, video, and voice traffic, respectively. For example, voice data should be sent with highest priority, so AC_VO is selected for this type of data. Default values of ACs are described in Table III. Note that although in this Table III, 802.11e EDCA defined eight default priorities for different data types [2], however, eight is not fixed and 802.11e EDCA allows less or more than the number of eight.

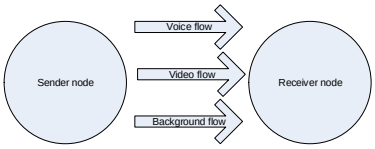


Figure 1. Two nodes with three data flows scenario.

The simulation parameters was shown in Table III.

TABLE III
SIMULATION PARAMETERS.

| Parameter | Value |
|---|---|
| Channel data rate | 11 Mbps |
| Antenna type | Omni direction |
| Radio Propagation | Two-ray ground |
| Transmission range | 250 m |
| Carrier Sensing range | 550 m |
| MAC protocol | IEEE 802.11e (EDCA) |
| Connection type | UDP/CBR |
| Packet size | 1024 bytes |
| Send rate | incremental from 1 to 1000 (packets/second) |
| Simulation time | 400 s |

With proper tuning of EDCA parameters set, traffic performance from different access categories can be optimized and prioritization of traffic can be achieved. It means that, these parameters are simply intended to define a fixed priority for each type of data: voice is always of the highest priority, then lower for video, and background.

IEEE 802.11e EDCA uses different parameter sets leading to different priorities for data flows [2]. Therefore, although there is no contention of accessing channel, between the flows there will be a contention of using channel. This leads to different throughput of each flow. However, multimedia data does not always have a fixed priority. In many cases, we may need to change the priority flexibly, so assigning fixed EDCA parameters to each data type will not solve this flexible control priority problem. To clarify this issue, we will focus on analyzing the effect of Contention Window sizes to throughput in 802.11e EDCA.

We evaluate the performance of 802.11e by using simlulation tool. There is a lot of software which is used to simulate wireless networks but is limited to the original 802.11. NS2 [16] is very popular for network simulations but it does not support 802.11e. Therefore, we have applied

the extension patch for 802.11e from [17] which was integrated with NS2 to simulate and evaluate related 802.11e problems in this paper.

We consider a simple two-node model with the flows of the three data types (voice, video, best effort) as shown in Figure 1 to investigate the 802.11e parameters set for contention only between data flows with different priorities, there is no contention in accessing channel. This model will be used to evaluate the influence of Contention Window values to the throughput of different data types.

Multimedia data can use TCP or UDP, but in this paper, we perform simulations with UDP data only as shown in Table III because many video and multimedia applications, such as VoIP use UDP. These applications can tolerate some data loss with little or no noticeable effect. The reliability mechanisms of TCP is not suitable for real-time applications as the re-transmissions can lead to high delay and cause delay jitter, which significantly degrades the quality.

Figure 2 show the throughput of each Traffic Class according to offered load, with default IEEE 802.11e parameters, the offered load is increased with high priority.

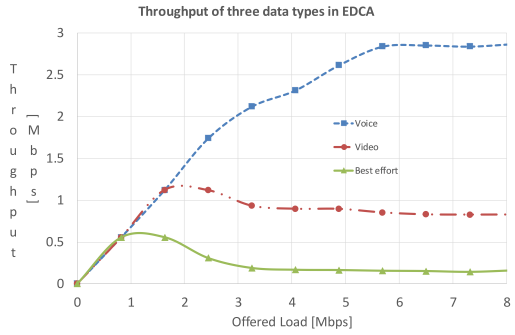


Figure 2. Simulation result of two nodes with three data flows scenario.

We demonstrate the relationship between Contention Window size of three data types (Voice, Video, and Background). To do that, we fixed two CW-size of the highest (voice data) and the lowest (background data) priorities and step-by-step changed the CW-size of video data. The range of CW values varies in many cases , for example {3 — 15} in default EDCA parameter as shown in Table III, or {7 – 31} with QoS Access Point/Basic Service Sets (BSS) [18], or {17 – 1023} for maximal fair throughput allocation with a toy WLAN scenario consisting of one Access Point and two associated stations [19]. In order to show that our proposed method can adapt to variable CW values, we will take CW values of three types of data which also change from 17 to 33, and CW size of the video data which will change in that range (17 to 33) to give an observation of how the throughput ratio of three data types changes corresponding to CW changes.

Looking at the simulation results in Figure 3. Here, the throughput of background data is considered as the basis (*Th BE*), the throughput ratios of Voice and Video compared to Background (*Th VO:BE* and *Th VI:BE*) are used to observe the effect of changing CW values to throughput of three data types. When CW increases, throughput decreases, whereas reduced CW increases throughput. As we see, the values of CW corresponding ratio *3:2:1* is *17:20:32*. We can see the change of throughput over different CW values, so in order to have flexible change ratio as user's demand, we will need many sets of CW values. It means that when assigning fixed CW values for data types, changing throughput ratio in 802.11e is very difficult to achieve.

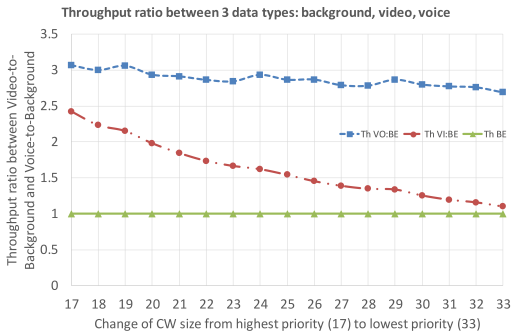


Figure 3. Estimation throughput by priority level ($CW_{min}$) between Voice and Video with Background data.

In the next section, we will propose a method to achieve this ratio with dynamic controlling CW size.

## IV. PROPOSED METHOD TO ACHIEVE THROUGHPUT BY RATIO

### 1. Define the ratio share between traffic class

IEEE-802.11e provides priority level for each traffic class, but it cannot guarantee throughput ratio suitable for each traffic class [20]. We propose the weighted number for each priority different from defined settings of IEEE-802.11e, these numbers will correspond to throughput ratio in our proposed solution. It means that, with a proposed numbers *3:2:1:1* shown in Table IV.1, our proposed method will get the ratio shared of throughput [of 2 Mbps total] in Figure 5.

Our previous work shows that the fairness between different data flows in 802.11e can be achieved via changing *CW* value with simple simulation scenario with two nodes sender–receiver [21], and the achieved result of fairness

mainly limited aspects allocate resources between flows in a traffic class (the same services).

In this paper, we improve the mechanism by flexibly assigning *CW* values to different data flows to achieve flexible ratio with more complex simulation scenarios. It means that users can simultaneously use different types of data and still achieve a fair share ratio of all traffic classes.

TABLE IV
NUMBERS FOR THROUGHPUT RATIO WITH DIFFERENT DATA TYPES.

| Priority | UP | AC | Data type | Weighted number |
|---|---|---|---|---|
| lowest | 1 | AC_BK | Background | 1 |
| - | 2 | AC_BK | Background | 1 |
| - | 0 | AC_BE | Best-effort | 1 |
| - | 3 | AC_BE | Best-effort | 1 |
| - | 4 | AC_VI | Video | 2 |
| - | 5 | AC_VI | Video | 2 |
| - | 6 | AC_VO | Voice | 3 |
| highest | 7 | AC_VO | Voice | 3 |

Our proposed method will control *CW* size to adapt to different throughput ratios of {*background*, *best effort*, *video*, *voice*} such as {10:10:30:40} or {15:15:30:40} percentage (%) of total bandwidth.

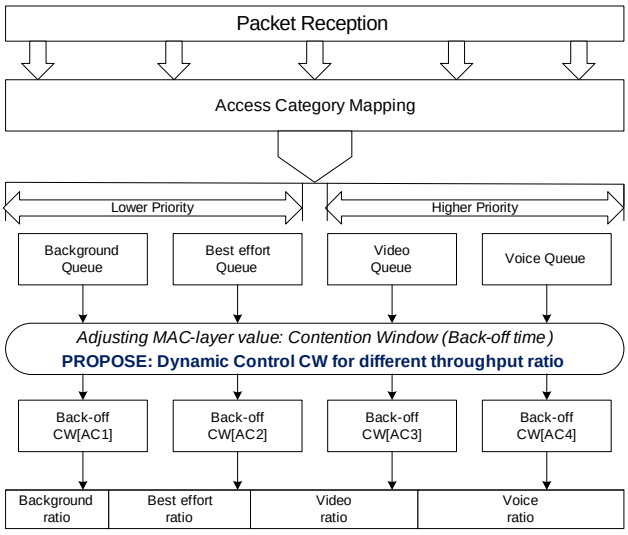


Figure 4. Controlling CW size to adaptive different throughput ratio.

With the proposed mechanism as shown in the processing module in Figure 4, after receiving input packets, the Access Category Mapping will put these packets into different priority queues, for example Background and Best Effort data onto low priority queues, Voice and Video data into high priority queues.

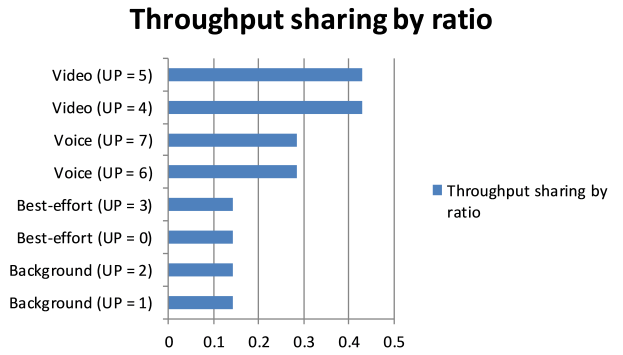


Figure 5. Expected throughput sharing by ratio with weighted numbers.

Then the control mechanism at the MAC sub-layer is proposed to control the corresponding Contention Window value (*CW*) for the throughput ratios corresponding to the different Access Types (*AC*) according to the parameters $CW[AC]$, since a large/small *CW* value will result in a short/long timeout (backoff) and consequently the corresponding larger/smaller throughput ratios, but will still not exceed network bandwidth. An example of the result would look like Figure 5, where bandwidth 2 [Mbps] is divided among different data types (Video:Voice:Best-effort:Background) in a ratio of 4:2:1:1).

## 2. Control CW to achieve the ratio share between traffic classes

The basic feature of IEEE-802.11e is to ensure that fixed-level priorities i.e. voice data are always assigned higher priority than background data, which leads to low-level of fairness in network. In order to achieve a more flexible throughput sharing, our proposed mechanism will adjust Contention Window (*CW*) size, because the *CW* size is related to the back-off time, and the longer or shorter back-off time leads to the opposite of access time being smaller or larger. It means that network traffic will vary with *CW* sizes.

Based on the cross-layer scheme to ensure fairness in IEEE 802.11 [22], we propose a number of MAC layer improvements in IEEE-802.11e in order to achieve fairness between different data flows (video, audio, text, for example). In general, data flows should have different priorities of bandwidth. For example, video data will need more bandwidth than voice data, or when calling phone over Internet (VoIP), voice data will be more prioritized than video data. Therefore, we need to have the corresponding way to assign bandwidth usage for data flows. We did this work based on two modules named *TX Flow Estimation* and *Utilization Estimation*. In fact, it is very difficult to have any method to ensure fairness in different layers (application,

network, MAC, etc.). Our proposed method will try to focus on ensuring fairness so that high-priority flows will not take all the bandwidth of low-priority flows.

Module *TX Flow Estimation* works in MAC layer to count the number of flows in the transmission range. We call these flows *TX flows*. A *TX flow* is determined by MAC addresses, IP addresses of source and destination and flow type (e.g. Voice, Video, BestEffort), by decoding the header of the packet. We define *TX flows* as $n_{TX[i]}$ where $i$ stands for $\{Voice, Video, BestEffort\}$. Note that in this case, *TX flow* is end-to-end flows data type, and it can include some process-to-process flows.

Assume that, there are $n$ data flows with $k_i$ as the weighted number of the data types defined in IEEE-802.11e. E.g. throughput ratio of Voice, Video and Best Effort is 3:2:1, we have $k_{Voice} = 3$, $k_{Video} = 2$, $k_{BestEffort} = 1$.

We calculate the total flow $n_{total}$ in $TXFlowEstimation$ module by the following formula:

$$n_{total} = \sum_{i=1}^{n} k_i \times n_{TX[i]} \tag{1}$$

Next we define the fair ratio share of the bandwidth for each flow by the formula:

$$Fair\_Share\_Ratio[i] = \frac{k_i}{\sum_{i=1}^{n} k_i \times n_{TX[i]}} \tag{2}$$

*Utilization Estimation* module evaluates real link utilization of the flow. The link utilization is determined by analyzing period $Active\_Time[i]$ of the flow in a predefined estimation period $EP$. The $Active\_Time[i]$ of the flow is defined as time used to transmit packets in $flowi$. The algorithm IV.1 is used to estimate the value of $Active\_Time[i]$ of the data flow $i$ by sensing packets. We compute real time for sending data in given EP time by 80 percent of current sending time plus 20 percent of the past calculated time (before) as shown in Algorithm IV.1 below.

**Algorithm IV.1:** $(Active\_Time[i])$

**Initialization:**
$Active\_Time[i] = 0$
$T_{Active[i]} = 0$
**for each** each interval time EP
$Active\_Time[i] = 0.8 \times Active\_Time[i] + 0.2 \times T_{Active[i]}$
$T_{Active[i]} = 0$
**for each** each packet p
**if** $p \rightarrow destID == localID$
**if** $p \rightarrow Type == CTS$
$T_{Active[i]} = T_{Active[i]} + T_{RTS} + T_{CTS}$
**if** $p \rightarrow Type == ACK$
$T_{Active[i]} = T_{Active[i]} + T_{DATA} + T_{ACK} + T_{Backoff}$

The $Real\_Share\_Ratio[i]$ is defined as the ratio of the $Active\_Time[i]$ to the estimation period $EP$ as the below equation:

$$Real\_Share\_Ratio[i] = \frac{Active\_Time[i]}{EP} \tag{3}$$

In this equation 3, estimation period ($EP$) time is a given observation period, chosen so that it is neither too short to ensure evaluation process, nor too long because it will distort the simulation. In the simulation done in the next part of this paper, the $EP$ value which we selected is two seconds. $CW$ will be adjusted, and the throughput will be evaluated during this $EP$ value.

Based on the $CW$ size which has been defined in Table 2, adjusted value of $CW$ will be determined by the formula:

$$CW'[i] = \frac{Real\_Share\_Ratio[i]}{Fair\_Share\_Ratio[i]} \times CW[i] \tag{4}$$

In the above formula, the fairness value $Fair\_Share\_Ratio[i]$ is used as a threshold of priority for accessing channel. If flows realized that the real value $Real\_Share\_Ratio[i]$ is less than its $Fair\_Share\_Ratio[i]$, it will use the $CW$ size which is smaller than in the back-off state. Thus, the flow can increase its chance to access channels and bandwidth allocation. Vice versa, if flow realized that $Real\_Share\_Ratio[i]$ is greater than its $Fair\_Share\_Ratio[i]$, it will use a greater value than $CW$ in back-off state. Therefore, it will reduce the chance to access channel, leading to other disadvantaged flows to have more opportunities to access the channel. In case some of the flows only have a small offered load, they will access channel more easily, and the remaining bandwidth will be shared by other flows. Thus, it allows

more efficiently use of channel bandwidth and ensure fair bandwidth allocation among flows.

## V. ANALYSIS OF SIMULATION RESULTS

### 1. The throughput by ratio and measurement to evaluate

For priorities shown in Table III, we will evaluate and simulate the weighted numbers *3:2:1* as shown in Table V.1.

TABLE V
THE WEIGHTED NUMBERS OF THREE DATA TYPES (VOICE, VIDEO, BEST-EFFORT) FOR SIMULATION.

| Priority | UP | AC | Data type | Weighted number |
|---|---|---|---|---|
| lowest | 1 | AC_BE | Best-effort | 1 |
| - | 2 | AC_VI | Video | 2 |
| highest | 3 | AC_VO | Voice | 3 |

To evaluate the effect of throughput sharing by ratio according to our proposed method, we use the *fairness index*, which is defined by R. Jain [23]. In this paper, the Fairness Index is evaluated based on the goodput at the destination station. We propose the equation for calculating fairness by priorities as follows:

$$FairnessIndex = \frac{(\sum_{i=1}^{n} \frac{x_i}{k_i})^2}{n \times \sum_{i=1}^{n} (\frac{x_i}{k_i})^2} \quad (5)$$

Here, $n$ is the number of flows, $x_i$ is the end-to-end throughput of *flow i*, and $k_i$ is the weighted number with the corresponding types of data. The result ranges from $1/n$ (worst case) to 1 (best case), and it is maximum when all flows receive the same allocation. This *fairness* value will be used to evaluate the sharing ratio of different bandwidth data types, meaning that the closer the value is to one (1), the more likely the ratio will be achieved.

### 2. Simulation models

We evaluate proposed method with different simulation models. The simulation parameters are described in Table III with NS-2 simulator. The EDCA *CW* used values of three data {*Voice, Video, Background*} are {17, 20, 32} as shown in Fig. 3 will ensure the ratio *Voice:Video:Background* = 3:2:1. For more complex models, we use the simulations below, this ratio is only achieved with our proposed mechanism but not with IEEE-802.11e, as in sections: V.2.a(2 senders), V.2.b (3 senders), and V.2.c (multiple senders with random model).

#### *a) Three nodes model*

The first model includes a chain of three nodes with six flows corresponding with three types of data (Voice, Video, Background) in Fig. 6. This model was known with *large-EIFS* problem [24]. According to IEEE 802.11 specifications, before a station starts the transmission, it should defer for a distributed IFS (DIFS) or an EIFS delay time and then selects a random contention window (*CW*) counter for backoff. Once the backoff counter decreases to zero, the station starts its transmission. The choice between DIFS and EIFS depends on the event of the last transmission. If the last event is an unsuccessful transmission (e.g., a collision), the station waits for an EIFS, otherwise it waits for a DIFS. The station which waits for an EIFS is called a retry station. The EIFS in the IEEE 802.11 standard is the largest Inter-Frame Space (IFS) and is used to protect ongoing frame, especially acknowledge frame (ACK) from collisions. When stations detect a transmission but cannot decode it, they set their Network Allocation Vectors (NAVs) for the EIFS duration. However sometimes the EIFS duration is larger/smaller than ongoing frame duration, leading to considerable unfairness and throughput degradation [25].

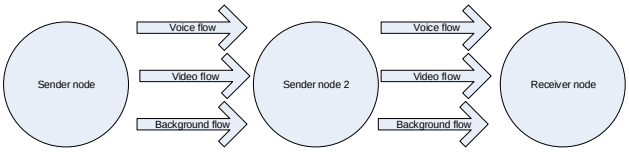


Figure 6. *Three nodes model.*

We examine network performance in this model by letting sender nodes 1 and 2 generate traffic at the same offered load to receiver node. The performance metrics Fairness Index and Total Throughput are evaluated with offered load.

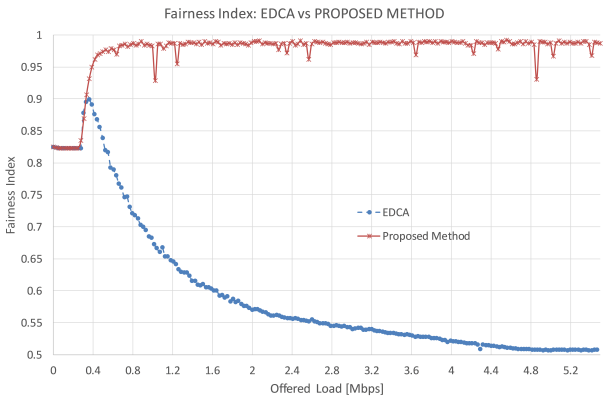


Figure 7. *Fairness Index in Three nodes model.*

In Fig. 7 and 8, "EDCA" means the result by using IEEE 802.11e and "Proposed Method" means our CW size adjustment shown in the paper (Section IV). Fairness Indexes are shown in Fig. 7. When offered load is small, all flows get its requirement. When offered load becomes larger, in EDCA, fixed priorities (corresponding fixed CW size) are assigned to different data types and higher priority flows can get more opportunity to transmit data, then Fairness Index might be worse. In our method, flexible CW size is controlled to different data types, abd the throughput of each flow becomes fairer and the bandwidth allocation at the MAC layer is also improved. Thus, our method achieves good Fairness Index.

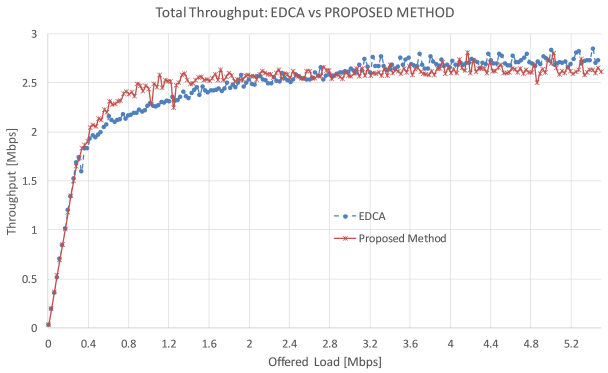


Figure 8. *Total throughput in Three nodes model.*

The Total Throughput of all flows is shown in Fig. 8. When the offered load is small, Total Throughput of all methods is similar. When offered load becomes large, in EDCA, bandwidth utilization is less efficient than others because EDCA gives some advantages to traffics of higher priority flows. In our method, by changing the flexibility of CW size, the lower priority flows chance to access channel can be increased by reducing CW size (back-off time). Therefore, the throughput of different data flows remains at a stable level for most of simulation time. It means, the Fairness and Throughput are often opposite. Figures 7 and 8 show that our proposed method achieves better fairness and there is still has a slight difference in throughput compared with EDCA.

*b) Three pairs model*

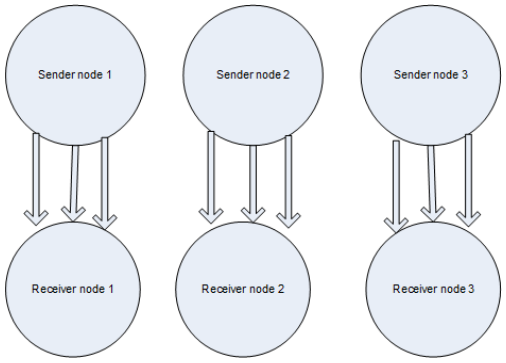


Figure 9. *Three pairs model.*

Figure 9 shows three pairs of sender and receiver nodes. The problem in this scenario is also known as *three-pair* problem which was first investigated in [26]. In this scenario, sender nodes 1-2 and 2-3 are out of the transmission range but in the carrier sensing range. Senders nodes 1 and 3 are out of the carrier sensing range, hence the two external pairs Sender 1 – Receiver 1 and Sender 3 – Receiver 3 are completely independent, i.e., they can send packets simultaneously without interference with each other. Thus, the two external pairs contend bandwidth only with the central pair Sender 2 – Receiver 2, while the central pair contends with both external pairs. In this topology, the central pair cannot access the medium in the saturated state in the original IEEE 802.11 [26].

We examine the network performance in this model by letting the sender nodes 1, 2 and 3 generate traffic at the same offered load to receiver nodes. The performance metrics Fairness Index and Total Throughput are evaluated with offered load.

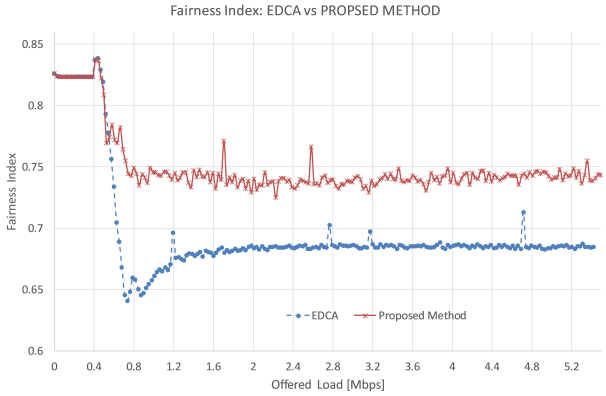


Figure 10. *Fairness Index in Three pairs model.*

Fairness Index is shown in Fig. 10. When offered load becomes large, EDCA cannot help the central pair access

the medium, because scheduling queue in EDCA only works at the link layer, so it does not have information of flows out of the transmission range. Therefore, it cannot improve MAC layer fairness. In our Proposed Method, the central pair finds out that its bandwidth is less than fair bandwidth allocation. Then Proposed Method tries to improve its chance to access channel by decreasing CW Size and therefore reducing the back-off time of station Sender 2. Thus, Proposed Method can achieve a better fairness than EDCA.

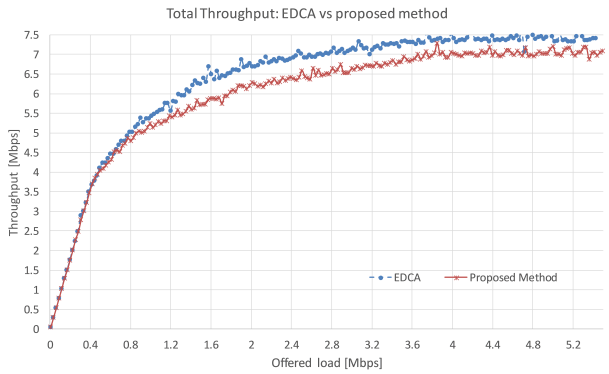


Figure 11. *Total throughput in Three pairs model.*

The Total Throughput of all flows is shown in Fig. 11. When offered load becomes large, EDCA achieves larger Total Throughput than our methods. This phenomenon is explained as follows. In EDCA, higher priority flows such as voice flow and video flow can use most of channel bandwidth with its CW value, so the throughput of the left and right pairs (numbered one and three) will take up most of the bandwidth, and the middle pair (numbered two) has very little bandwidth. In our method, with flexible CW size, we can give more opportunity for lower priority flows, so the throughput of the middle pair will increase and the throughput of the left and right pairs will decrease, leading to the total throughput of our method being possibly lower than EDCA.

*c) Random model*

The third model is a random topology. We make a topology with 50 stations at random positions in $1000[m] \times 1000[m]$ area. Among those 50 stations, $m$ stations are chosen randomly and these $n$ stations generate UDP traffic flows to one destination station. Total offered load of source stations is set equal to channel data rate 11[Mbps]. The average of Fairness Index and total end-to-end throughput are used as metrics to compare the throughput and fairness, respectively. These terms of network performance are examined versus the number of flows. Each data point is the average of over 50 simulations.

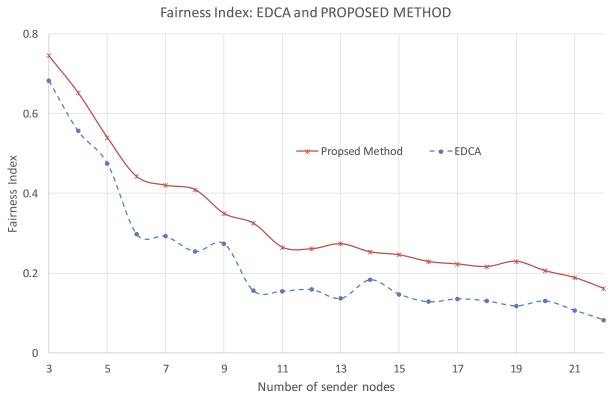


Figure 12. *Fairness Index in Random model.*

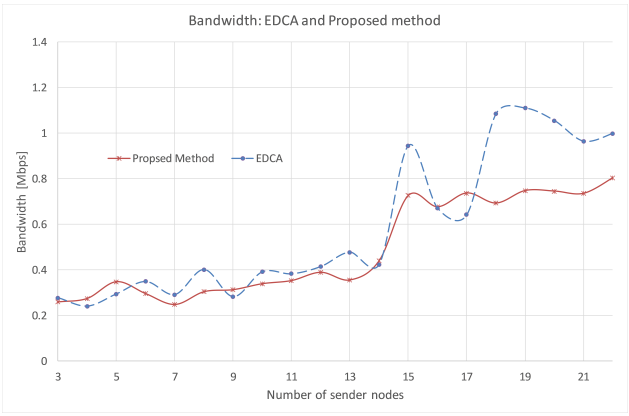


Figure 13. *Total throughput in Random model.*

These terms of network performance are examined versus the number of flows. The simulation results also prove that our proposed method achieves good fairness performance as in Fig. 12. Our throughput performance is slightly reduced by the trade-off with fairness performance as described in Fig. 13.

## VI. CONCLUSION

Inheriting from IEEE 802.11, IEEE 802.11e standard was developed and published, partly satisfying QoS for multimedia data, but in terms of sharing fairness, this standard is still limited because it gives relatively fixed values to the QoS parameters. Our research has proposed a mechanism to improve the fair sharing of bandwidth between flows by adapting CW in a more reasonable way. Our algorithm is simple to implement, and it provides flexible CW value, corresponding to the value of priority of throughput for different data types to achieve a reasonable level of sharing between multimedia flows. The evaluated and simulated results by NS-2 have verified that our method is better than original IEEE-802.11e in some simulation

models. Our future research is aimed at evaluating mixed data UDP and TCP based on the wireless testbed system.

## ACKNOWLEDGMENT

This work was financially supported by Institute of Information Technology (IOIT), Vietnam Academy of Science and Technology (VAST).

## REFERENCES

[1] *IEEE 802.11e*, https://standards.ieee.org/standard/802_11e-2005.html.
[2] *IEEE 802.11-2012*, https://standards.ieee.org/standard/802_11-2012.html.
[3] C. Casetti and C. Chiasserini, "Improving fairness and throughput for voice traffic in 802.11e EDCA," in *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, vol. 1, September 2004, pp. 525–530.
[4] D. J. Leith and P. Clifford, "Using the 802.11 e edcf to achieve tcp upload fairness over wlan links," in *Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'05)*. IEEE, 2005, pp. 109–118.
[5] E. Park, N. Kwak, S. K. Lee, J. Kim, and H. Kim, "Provisioning QoS for WiFi-enabled Portable Devices in Home Networks," *TIIS*, vol. 5, no. 4, pp. 720–740, 2011. [Online]. Available: https://doi.org/10.3837/tiis.2011.04.006
[6] J. Lei, J. Huang, J. Tao, and Y. Xia, "A Differentiated Reservation MAC Protocol for Achieving Fairness and Efficiency in Multi-rate IEEE 802.11 WLANs," *IEEE Access*, vol. PP, pp. 1–1, 01 2019.
[7] M. Yazdani, M. Kamali, N. Moghim, and M. Ghazvini, "A fair access mechanism based on txop in ieee 802.11 e wireless networks," *International Journal of Communication Networks and Information Security*, vol. 8, no. 1, p. 11, 2016.
[8] F. Cali, C. M, and G. E, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Transactions On Networking*, vol. 8, no. 6, pp. 785–799, December 2000.
[9] F.Cali, C. M, and G. E, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, September 2000.
[10] D. Chen, S. Garg, M. Kappes, and K. Trivedi, "Supporting VBR VoIP Traffic in IEEE 802.11 WLAN in PCF Mode," *IEEE Transactions on Wireless Communications*, 2002.
[11] S. Garg, M. Kappes, and A. Krishnakumar, "On the Effect of Contention-Window Sizes in IEEE 802.11b Networks," *Avaya Labs Research*, June 2002.
[12] D. He and C. Shen, "Simulation Study of IEEE 802.11e EDCA," in *Vehicular Technology Conference*, vol. 1, Spring 2003, pp. 685–689.
[13] J. Yoon, S. Yun, H. Kim, and S. Bahk, "Maximizing Differentiated Throughput in IEEE 802.11e Wireless LANs," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, Nov 2006, pp. 411–417.
[14] Y. Xiao, H. Li, and S. Choi, "Protection and guarantee for voice and video traffic in IEEE 802.11e wireless LANs," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 3, March 2004, pp. 2152–2162.
[15] V. A. Siris and G. Stamatakis, "Optimal CWmin selection for achieving proportional fairness in multi-rate 802.11e WLANs: testbed implementation and evaluation," in *WiNTECH '06 Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization*, 2006, pp. 41–48.
[16] *The Network Simulator: ns-2*, http://www.isi.edu/nsnam/ns/.
[17] "IEEE 802.11e EDCA Simulation Model for ns-2 (TU Berlin)," http://www2.tkn.tu-berlin.de/research/802.11e_ns2/, last accessed on 02/11/18.
[18] "Wi-Fi Quality of Service," http://www.revolutionwifi.net/revolutionwifi/2010/08/wireless-qos-part-5-contention-window.html, last accessed on 12/10/18.
[19] A. Garcia-Saavedra, P. Serrano, A. Banchs, and M. Hollick, "Energy-efficient fair channel access for IEEE 802.11 WLANs," in *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, June 2011, pp. 1–9.
[20] M. K. Alam, S. A. Latif, M. Akter, F. Anwar, and M. K. Hasan, "Enhancements of the dynamic txop limit in edca through a high-speed wireless campus network," *Wireless Personal Communications*, vol. 90, no. 4, pp. 1647–1672, Oct 2016. [Online]. Available: https://doi.org/10.1007/s11277-016-3416-4
[21] N. H. Anh and P. T. Giang, "An enhanced mac-layer improving to support qos for multimedia data in wireless networks," *Indian Journal of Science and Technology*, vol. 9, no. 20, 2016. [Online]. Available: http://www.indjst.org/index.php/indjst/article/view/92732
[22] P. Giang and K. Nakagawa, "Cross-Layer Scheme to control Contention Window for per-flow Fairness in Asymmetric Multi-hop Networks," *IEICE TRANSACTIONS on Communications*, vol. E93-B, no. 9, pp. 2326–2335, 2010.
[23] R. Jain, D. Chiu, , and W. Hawe, "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems," DEC Research Report TR-301, Tech. Rep., September 1984.
[24] Z. Li, S. Nandi, and A. K. Gupta, "Ecs: An enhanced carrier sensing mechanism for wireless ad hoc networks," *Computer Communications*, vol. 28, no. 17, pp. 1970 – 1984, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366405001416
[25] F. Khan, "Fairness and throughput improvement in multihop wireless ad hoc networks," in *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2014, pp. 1–6.
[26] C. Chaudet, I. G. Lassous, E. Thierry, and B. Gaujal, "Study of the impact of asymmetry and carrier sense mechanism in IEEE 802.11 multi-hops networks through a basic case," in *PE-WASUN '04: Proceedings of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. New York, NY, USA: ACM Press, 2004, pp. 1–7.

**Ngo Hai Anh** received his M.Tech degrees from Vietnam National University and is a PhD student at Graduate Univeristy of Science and Technology, Vietnam Academy of Science and Technology. He is currently a researcher at Telemactis Department of Institute of Information Technology, Vietnam Academy of Science and Technology. His current research interests include wireless network, network performance and network management.

**Pham Thanh Giang** received his B.E. degree from the Hanoi University of Technology, Vietnam, in 2002. He achieved M.E. and D.E. degrees from the Nagaoka University of Technology, Japan, in 2007 and 2010, respectively. He is currently a researcher at Institute of Information Technology, Vietnam Academy of Science and Technology. His interests are security, mobile ad hoc networks, the Internet architecture in mobile environments and Internet traffic measurement.

# An Efficient ACO+RVND for Solving the Resource-Constrained Deliveryman Problem

Ha-Bang Ban
School of Information and Communication Technology
Hanoi University of Science and Technology
Email: BangBH@soict.hust.edu.vn; Bang.BanHa@hust.edu.vn

Communication: received 20 December 2019, revised 21 February 2020, accepted 17 April 2020
Digital Object Identifier: 10.32913/mic-ict-research.v2020.n1.911

**Abstract: In this paper, Resource-Constrained Deliveryman Problem (RCDMP) is introduced. The RCDMP problem deals with finding a tour with minimum waiting time sum so that it consumes not more than $R_{max}$ units of resources, where $R_{max}$ is some constant. Recently, an algorithm developed in a trajectory-based metaheuristic has been proposed. Since the search space of the problem is a combinatorial explosion, the trajectory-based sequential can only explore a subset of the search space, therefore, they easily fall into local optimal in some cases. To overcome the drawback of the current algorithms, we propose a population-based algorithm that combines an Ant Colony Algorithm (ACO), and Random Variable Neighborhood Descent (RVND). In the algorithm, the ACO explores the promising solution areas while the RVND exploits them with the hope of improving a solution. Extensive numerical experiments and comparisons with the state-of-the-art metaheuristic algorithms in the literature show that the proposed algorithm reaches better solutions in many cases.**

**Keywords:** *RCDMP, ACO, GA, metaheuristic.*

## I. INTRODUCTION

The Resource-Constrained Deliveryman Problem (RCDMP) has not been studied much in literature. Informally, the RCDMP aims to find a tour with minimum waiting time sum so that it consumes no more than $R_{max}$ units of resources, where $R_{max}$ is some constant. The resource consumption of a tour is the sum of the resource consumptions of its edges. In the special case, since resource-constraint is ignored, the RCDMP becomes the Deliveryman Problem (DMP) or Traveling Repairman Problem (TRP). The problem has applications in reality, e.g., disk head scheduling [1, 5, 6, 8, 17, 19].

The RCDMP is at least as hard as DMP, therefore it is a NP-hard problem. The RCDMP is formulated as followings: Given a complete graph $K_n = (V, E)$ where $V = 1, 2, \ldots n$ is a set of vertices, and $E$ is the set of edges. For each edge $(v_i, v_j) \in E$, which connects the two vertices $v_i$ and $v_j$, there exists a cost $c(v_i, v_j)$ and resource consumption $r(v_i, v_j)$. Suppose that $T = (v_1, \ldots, v_k, \ldots, v_n)$ is a tour in $K_n$. Denote $P(v_1, v_k)$ is the path from $v_1$ to $v_k$ on the tour $T$ and $l(P(v_1, v_k))$ is its length. The waiting time of a vertex $v_k (1 < k \leq n)$ on $T$ is the length of the path from starting vertex $v_1$ to $v_k$:

$$w(v_k) = \sum_{i=1}^{k-1} c(v_i, v_{i+1}).$$

The total waiting time of tour $T$ is defined as the sum of waiting time of all vertices:

$$L(T) = \sum_{k=2}^{n} w(v_k).$$

The resource consumption of a tour is the sum of the resource consumption of its edges. Tour $T$ must satisfy the following resource constraint:

$$\sum_{i=1}^{n-1} r(v_i, v_{i+1}) \leq R_{max}. \quad (1)$$

The RCDMP asks for a tour with minimum waiting time sum so that it consumes not more than $R_{max}$ units of resource.

The RCDMP is a NP-hard problem; thus, a metaheuristic algorithm is a suitable approach for it in short computation time. Previously, the only algorithm was developed in a trajectory-based metaheuristic approach to solve the RCDMP [4]. This algorithm starts with a single initial solution and, at each step of the search, the current solution is replaced by another solution found in its neighborhood. However, this algorithm may implement a strong intensification strategy. However, its diversification may not be good enough. As a result, it gets stuck in local optima in some cases. Another approach is population-based which performs a search with multiple promising solutions, thus, its exploring search space is extended. As a result, the chance to obtain better solutions is higher.

In this paper, we propose a population-based algorithm that combines an Ant Colony algorithm (ACO) [7], and Random Variable Neighborhood Descent (RVND) [16]. The two important features in our algorithm are that:

- Two kinds of ants coexist in the ACO. In a computational experiment, the researchers [10] performed the feeding behavior by using intelligent ants, which can trail the pheromone exactly, and dull ants which cannot trail the pheromone. From results, the ant group that includes the dull ants can obtain more foods than the group consisting of only the intelligent ants. The dull ants are regarded as having a task that finds the new food sources by dawdle. It means that the coexistence of the intelligent and dull ant improves the effectiveness of the feeding behavior.
- To be successful, a search algorithm needs to establish a good ratio between exploration and exploitation. In our algorithm, the ACO explores the promising solution areas while the RVND exploits them with the hope of improving a solution. Extensive numerical experiments and comparisons with the state-of-the-art metaheuristic algorithms on 320 instances show that the proposed algorithm reaches better solutions in 303 cases.

The rest of this paper is organized as follows. Section 2 presents the proposed algorithm. Computational evaluations are reported in section 3. Sections 4, 5 discuss and conclude the paper, respectively.

## II. THE PROPOSED ALGORITHM

In this paper, we propose an algorithm which utilizes the advantages of both the ACO [7], and RVND [16]. First, in the ACO algorithm [7], multiple solutions called “ants" coexist, and the ants drop pheromone on the path connecting the locations. Pheromone trails are updated depending on the behavior of the ants. The ants find a food source through paths having strong pheromone. By communicating with other ants according to the pheromone strength, the algorithm tries to find the optimal solution. Lastly, the RVND [16] is based on a simple principle of systematic switches between different neighborhoods.

A pseudocode of our algorithm is given in Algorithm 1. Our algorithm is repeated a number of times, and the best solution found is reported. In Step 1, the ACO is used to generate an initial ant population that is an input for the RVND in Step 2. In Step 2, to exploit promising solution spaces, the RVND is implemented with the best solution from the ACO. Our algorithm stops after $m$ iterations. If the best solution has not been improved. In the remaining of this section, more details about the three steps of our algorithm are given in Algorithm 1.

**Algorithm 1** ACO-RVND

**Input:** $K_n, Sp$, and $\tau_0$ are the graph, the cost matrix, the size of ant population, and the initial amount of pheromone deposited, respectively.
**Output:** The best solution $T^*$.

```
while (The termination criterion of the ACO-RNVD is not satisfied) do
  {Step 1: The ACO step}
  Initiate pheromone;
  while (|P| < Sp) do
    {Choose the type of ant. If the value of rd is less than 80, the ant depends on the intelligent type. Otherwise, it fails into dull type}
    rd = random(100);
    T_k = T_k ∪ v_1;
    v = v_1; {v is a next vertex}
    while | T_k |≤ n do
      N_k = {v_j | v_j ∉ T_k and v_j ∈ K_n};
      v = Pick-Forward-Vertex(v, rd, N_k);
      T_k = T_k ∪ v;
    end while
    Update phernomone τ_ij according to (4), (5);
    P = P ∪ T_k; {Add T_k to the ant population P}
  end while
  {Step 2: The RVND step}
  for T ∈ P do
    T' = RVND(T);
    if (L(T') < L(T*)) and (T' is feasible)) then
      T* = T';
      improved=True;
    end if
  end for
  {Step 3: Update pheronome}
  if (improved==True) then
    Update pheronome information by (6), and (7);
  end if
end while
return T*;
```

### 1. Penalty on infeasible solution

Our search is not constrained to feasible solutions. During the search we penalize infeasible solutions by incorporating a penalty term. For each solution $T$, let $L(T)$ denote the total latency and let $V(T)$ denote the total violation of the constraint. The total latency violation $V(T)$ is computed on a tour basis with respect to the value $R_{max}$. Specifically, it is equal to

$$\max\{LR - R_{max}, 0\},$$

where $LR$ is the resource consumption value in the current solution (feasible or infeasible).

**Algorithm 2** Pick-Forward-Vertex($v_i, rd, N_k$)

**Input:** $v_i, rd, N_k$ are the current vertex, a random number, and a set of unvisited vertices, respectively.
**Output:** A next vertex $v$.

```
proSum = 0;
if (rd ≤ ant_type) then
  {The second only uses the amount of phernomone as (2)}
  for (v_j ∈ N_k) do
    proSum = proSum + pow(τ[i,j], β_τ) × pow(β[i,j], β_μ);
  end for
  for (v_j ∈ N_k) do
    p[i,j] = pow(τ[i,j],β_τ)×pow(η[i,j],β_μ) / proSum;
  end for
  rd = random(proSum);
  for (v_j ∈ N_k ) do
    wheelSum = wheelSum + p[i,j];
    if (wheelSum > rd) then
      break;
    end if
    v = v_j;
  end for
end if
{The third cannot trail the pheromone as (3)}
if (rd > ant_type) then
  Select randomly vertex v ∈ N_k;
end if
```

**Algorithm 3 RVND**($T$)

**Input:** $T$ is a tour.
**Output:** A new solution $T$.

```
Initialize the Neighborhood List NL;
while NL ≠ 0 do
  Choose a neighborhood N in NL at random
  T' ← arg min N(T); {Neighborhood search}
  if ((L(T') < L(T)) then
    T ← T'
    Update NL;
  else
    Remove N from the NL;
  end if
end while
```

Solutions are then evaluated according to the weighted fitness function $L' = L + PF \times V(T)$, where $PF$ is the penalty factor. If the obtained solution is feasible then $LR \leq R_{max}$ and $L' = L$.

## 2. Step 1: ACO

Step 1 includes some small steps as follows:
- **Encoding for ants**: We simply use the natural encoding, in which an ant individual is represented as a list of $n$ vertices $(v_1, v_2, ..., v_k, ..., v_n)$, where $v_k$ is the $k-$th vertex to be visited. Each individual represents one possible tour.
- **Initializing parameters for the ACO**: Let $\tau_{k-1ij}$ be the amount of pheromone on the edge $(v_i, v_j)$ so that the $k-$th ant uses to trail $(k = 1, 2, ..., Sp)$. Initially, $\tau_{0ij}$ is set to $\tau_0$, respectively.
- **Selecting for the next vertex**: Assume that, $k-$th ant needs to be chosen to go from vertex $v_i$ to a next vertex in an unvisited vertex set $N_k$. Our algorithm maintains two types of ant:
The first uses the amount of phernomone thus, its probability equation is as follows:

$$p^2_{kij} = \frac{[\tau_{k-1ij}]^{\beta_r}[\eta_{k-1ij}]^{\beta_\eta}}{\sum_{v_j \in N_k}[\tau_{k-1ij}]^{\beta_r}[\eta_{k-1ij}]^{\beta_\eta}} \quad (2)$$

The last does not use either the phernomone or genetic information, therefore its probability equation is as follows:

$$p^3_{kij} = \frac{1}{|N_k|} \quad (3)$$

where $\beta_r$, $\beta_\eta$, and $\beta_g$ are parameters, $\eta_{kij} = \frac{1}{\rho \times c_{ij}}$ ($\rho$, and $|N_k|$ are the position of edge $(v_i, v_j)$, and number of vertices in the tour, respectively). The selection of $v_j$ is done by the roulette wheel selection [9]. The ants repeat choosing a next vertex until all vertices are visited. The detail in this step is given in Algorithm 2.
- **Updating Pheromone of $k-$ant**: After $k-$ant has completed the tour $T_k$, its deposited pheromone on $T_k$ is updated. We should note that the dull ants can deposit the pheromone although they cannot trail the pheromone. The mount of pheromone which an ant deposits on $T_k$ is calculated as follows:

$$\Delta\tau_{kij} = \begin{cases} \frac{\sigma}{L(T_k)}, & \text{if } (v_i, v_j) \in T_k \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $\sigma$ is a parameter. Since the $k$-ant finishes its tours, we update $\tau_{ij}$ of each edge $(v_i, v_j)$ depending on $\Delta\tau_{kij}$ as follows:

$$\tau_{kij} = (1 - p) \times \tau_{k-1ij} + \Delta\tau_{kij} \quad (5)$$

where $p \in (0, 1)$ is the pheromone trail decay coefficient.

## 3. Step 2: RVND

To be successful, a search algorithm needs to establish a good ratio between exploration and exploitation. In the proposed algorithm, the ACO is also combined with the RVND to keep a balance between exploration and

exploitation. In this combination, the ACO explores the promising solution spaces while the RVND bases on a simple principle of systematically switches between different neighborhoods to exploit these spaces. In the RVND step, several neighborhoods are used such as remove-insert, reinsertion, swap-adjacent, swap, 2-opt, and or-opt in [16]. In THE RCDMP, by using the known cost of the current solution, this operation can be done in constant time [4]. Therefore, the running time of exploring the neighborhoods can be sped up. Six neighborhoods used in the step are described as followings: 1) **Swap-adjacent**: It swaps each pair of adjacent vertices. The complexity of exploring the neighborhood is $O(n)$; 2) **Remove-insert** moves each vertex in the solution at the end of it. The complexity of exploring the neighborhood is $O(n)$; 3) **Reinsertion**: One vertex is relocated to another position of the tour. The complexity of exploring the neighborhood is $O(n)$; 4) **Swap**: It tries to swap the positions of each pair of vertices in the single tour $T$. The complexity of exploring the neighborhood is $O(n^2)$; 5) **2-opt**: It removes each pair of edges from the tour and reconnects them. The complexity of exploring the neighborhood is $O(n^2)$; 6) **Or-opt**: Three adjacent vertices are reallocated to another position of the tour. The complexity of exploring the neighborhood is $O(n^2)$. A pseudocode of the RVND algorithm is given in Algorithm 3.

### 4. Step 3: Update pheromone

If the new best solution is found, we use it to update pheromone value. The pheromone information $\Delta\tau^*_{ij}$ bequeathed to the ACO in the next iteration is calculated as follows:

$$\Delta\tau^*_{ij} = \begin{cases} \dfrac{\sigma}{L(T^*)}, & \text{if } (v_i, v_j) \in T^* \\ 0, & \text{otherwise;} \end{cases} \quad (6)$$

where $\sigma$, $L(T^*)$ are parameter, and the cost of $T^*$, respectively. Therefore, the pheromone information on each edge $(v_i, v_j)$ is updated by $\Delta\tau^*_{ij}$ as follows:

$$\tau_{ij} = \tau_{ij} + \Delta\tau^*_{ij} \quad (7)$$

## III. COMPUTATIONAL RESULTS

The experiments are conducted on a personal computer, which is equipped with an Intel Pentium core i7 duo 2.10 Ghz CPU and 8 GB bytes RAM memory.

### 1. Benchmark Instances

The numerical analysis was performed on a set of benchmarks described in [4, 19, 23]. In our experiments, two types of dataset are selected as followings:

- The RCDMP dataset in [4]: Cost matrix elements, $c_{ij}$, are independent and uniformly chosen random integers in the range [0, 200]. Resource matrix elements, $r_{ij}$, are independent and uniformly chosen integers in the range [0, 200 - $c_{ij}$]. The cost matrix $C = [c_{ij}]$ and $R = [r_{ij}]$ are generally symmetric and metric. Maximum resource usage, $R_{max}$, is computed using the following formula:

$$R_{max} = \lceil \alpha \times \sum_{i \in V} \sum_{j \in V} r_{ij} x^c_{ij} + (1 - \alpha) \times \sum_{i \in V} \sum_{j \in V} r_{ij} x^r_{ij} \rceil$$

  In the above formula, $x^c_{ij}$ represent the optimal solution of the unconstrained DMP with the cost matrix $c_{ij}$. Similarly, $x^r_{ij}$ represent the optimal solution of the problem

$$\sum_{i=1}^{n-1} r(v_i, v_{i+1}) \rightarrow \min,$$

  where the cost matrix is defined by the matrix $r_{ij}$. For the problem, we choose the Concorde tool [1] to solve exactly the problem. In the equation, $\alpha$ is a parameter, that provides means for controlling tightness of the resource constraint. When $\alpha = 0$, the resource constraint is very tight. On the other hand, when $\alpha = 1$, the resource constraint has no effect on the optimal solution. We choose $\alpha = 0, 0.5, 0.75$ and 1 and vary the size of the instances between 30 to 100 vertices with the different values of $\alpha$ to create 320 instances. For unconstrained DMP with a small number of instances ($n = 30, 40, 50$), we use the exact algorithm in [1, 2] to solve. Currently, there is no exact algorithm in the literature that can solve the problem with up to 100 vertices. Therefore, we use the metaheuristic in [3] to obtain near-optimal solutions. All instances are available in the online supplement to this paper at [2].
- Salehipour et al.'s dataset in [19]: Five of these sets are generated by Salehipour et al., where each of them is composed of 20 instances with 10, 20, 50, 100, and 200 customers, respectively. In total, there are 100 instances.
- TSPLIB in [23]: Some instances, that are from 70 to 150 vertices are selected. The optimal solutions to them are extracted from [1]. In total, there are 11 instances.

The selection of pheromone-related parameters are conducted in preliminary experiments. The parameter *ant_type* is set to 80 because it has been reported that about 20 percent of the dull ant and 80 percent of the intelligent ant live in the real ant's world in [10]. For the

[1] http://www.math.princeton.edu/tsp/concorde.html
[2] https://drive.google.com/drive/u/2/folders/1M_zAeuJXpOuVlnmdmwg6pjDSn4ctA9X3?ths=true

Table 2. The experiment results for the RCDMP-instances ($n$ = 30)

| Instances | α = 0 | | | | | α = 0.5 | | | | | α = 0.75 | | | | | α = 1 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TS | Our algorithm | | | | TS | Our algorithm | | | | TS | Our algorithm | | | | TS | Our algorithm | | | |
| | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* |
| test-1 | 6654 | **6259** | 6259 | -5.94 | 0.13 | 6611 | **6470** | 6470 | -2.13 | 0.13 | 6391 | **6351** | 6351 | -0.63 | 0.12 | 6391 | **6331** | 6331 | -0.94 | 0.10 |
| test-2 | 7130 | **6854** | 6854 | -3.87 | 0.11 | 7289 | **6983** | 6983 | -4.20 | 0.12 | 7050 | **6710** | 6710 | -4.82 | 0.15 | 6654 | 6903 | 6903 | 3.74 | 0.11 |
| test-3 | 6546 | **6042** | 6042 | -7.70 | 0.14 | 6441 | **5939** | 5939 | -7.79 | 0.14 | 6430 | **5958** | 5958 | -7.34 | 0.12 | 6381 | **6060** | 6060 | -5.03 | 0.14 |
| test-4 | 6866 | **6338** | 6338 | -7.69 | 0.11 | 6879 | **6260** | 6260 | -9.00 | 0.13 | 6715 | **6407** | 6407 | -4.59 | 0.11 | 6654 | **6336** | 6336 | -4.78 | 0.10 |
| test-5 | 6619 | **6124** | 6124 | -7.48 | 0.13 | 6551 | **6111** | 6111 | -6.72 | 0.12 | 6674 | **6021** | 6021 | -9.78 | 0.15 | 6215 | **6246** | 6246 | 0.50 | 0.15 |
| test-6 | 6652 | **6525** | 6525 | -1.91 | 0.11 | 6780 | **6448** | 6448 | -4.90 | 0.15 | 6690 | **6520** | 6520 | -2.54 | 0.15 | 6654 | **6435** | 6435 | -3.29 | 0.14 |
| test-7 | 7189 | **6804** | 6804 | -5.36 | 0.12 | 7269 | **6918** | 6918 | -4.83 | 0.14 | 7457 | **6710** | 6710 | -10.02 | 0.12 | 6654 | 6815 | 6815 | 2.42 | 0.12 |
| test-8 | 5888 | **5569** | 5569 | -5.42 | 0.13 | 5956 | **5713** | 5713 | -4.08 | 0.13 | 5830 | **5720** | 5720 | -1.89 | 0.11 | 5830 | **5638** | 5638 | -3.29 | 0.13 |
| test-9 | 6063 | **5862** | 5862 | -3.32 | 0.14 | 6228 | **5718** | 5718 | -8.19 | 0.13 | 6051 | **5574** | 5574 | -7.88 | 0.11 | 5878 | **5867** | 5867 | -0.19 | 0.11 |
| test-10 | 6475 | **6047** | 6047 | -6.61 | 0.10 | 6263 | **6080** | 6080 | -2.92 | 0.13 | 6257 | **6201** | 6201 | -0.89 | 0.12 | 6257 | **6207** | 6207 | -0.80 | 0.12 |
| test-11 | 7424 | **6952** | 6952 | -6.36 | 0.15 | 7499 | **6871** | 6871 | -8.37 | 0.11 | 7590 | **7116** | 7116 | -6.25 | 0.13 | 6654 | **6769** | 6769 | 1.73 | 0.15 |
| test-12 | 6371 | **5796** | 5796 | -9.03 | 0.14 | 6110 | **5807** | 5807 | -4.96 | 0.12 | 6357 | **5936** | 5936 | -6.62 | 0.11 | 6110 | **5909** | 5909 | -3.29 | 0.13 |
| test-13 | 6029 | **5671** | 5671 | -5.94 | 0.12 | 6083 | **5736** | 5736 | -5.70 | 0.12 | 6134 | **5680** | 5680 | -7.40 | 0.13 | 5885 | **5739** | 5739 | -2.48 | 0.13 |
| test-14 | 6160 | **5613** | 5613 | -8.88 | 0.12 | 5545 | **5678** | 5678 | 2.40 | 0.11 | 5960 | **5613** | 5613 | -5.82 | 0.14 | 5545 | **5606** | 5606 | 1.10 | 0.11 |
| test-15 | 6387 | **5810** | 5810 | -9.03 | 0.12 | 6237 | **5832** | 5832 | -6.49 | 0.14 | 6013 | **5993** | 5993 | -0.33 | 0.11 | 6013 | 6040 | 6040 | 0.45 | 0.12 |
| test-16 | 6772 | **6532** | 6532 | -3.54 | 0.12 | 6818 | **6409** | 6409 | -6.00 | 0.11 | 6794 | **6468** | 6468 | -4.80 | 0.11 | 6654 | **6348** | 6348 | -4.60 | 0.13 |
| test-17 | 6591 | **6509** | 6509 | -1.24 | 0.13 | 7069 | **6546** | 6546 | -7.40 | 0.11 | 6988 | **6699** | 6699 | -4.14 | 0.11 | 6654 | 6804 | 6804 | 2.25 | 0.13 |
| test-18 | 6943 | **6460** | 6460 | -6.96 | 0.13 | 6856 | **6657** | 6657 | -2.90 | 0.11 | 6736 | **6520** | 6520 | -3.21 | 0.12 | 6654 | **6576** | 6576 | -1.17 | 0.12 |
| test-19 | 7622 | **7258** | 7258 | -4.78 | 0.14 | 7289 | **6942** | 6942 | -4.76 | 0.11 | 7354 | **7194** | 7194 | -2.18 | 0.12 | 6654 | **7076** | 7076 | 6.34 | 0.12 |
| test-20 | 6472 | **5848** | 5848 | -9.64 | 0.14 | 6352 | **5985** | 5985 | -5.78 | 0.12 | 6409 | **5947** | 5947 | -7.21 | 0.13 | 6352 | **5938** | 5938 | -6.52 | 0.15 |
| Aver | | | | -6.03 | 0.13 | | | | -5.24 | 0.12 | | | | -4.92 | 0.12 | | | | -0.89 | 0.13 |

Improved solutions are highlighted in boldface

Table 3. The experiment results for the RCDMP-instances ($n$ = 40)

| Instances | α = 0 | | | | | α = 0.5 | | | | | α = 0.75 | | | | | α = 1 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TS | Our algorithm | | | | TS | Our algorithm | | | | TS | Our algorithm | | | | TS | Our algorithm | | | |
| | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* |
| test-1 | 10745 | **10606** | 10606 | -1.29 | 0.25 | 10764 | **10622** | 10622 | -1.32 | 0.20 | 10851 | **10525** | 10525 | -3.00 | 0.21 | 10745 | **10508** | 10508 | -2.21 | 0.22 |
| test-2 | 9934 | **9539** | 9539 | -3.98 | 0.25 | 9925 | **9324** | 9324 | -6.06 | 0.23 | 9867 | **9491** | 9491 | -3.81 | 0.21 | 9605 | **9252** | 9252 | -3.68 | 0.20 |
| test-3 | 9800 | **9400** | 9400 | -4.08 | 0.20 | 10226 | **9572** | 9572 | -6.40 | 0.24 | 9715 | **9352** | 9352 | -3.74 | 0.23 | 9715 | **9238** | 9238 | -4.91 | 0.23 |
| test-4 | 11222 | **10317** | 10317 | -8.06 | 0.24 | 11054 | **10167** | 10167 | -8.02 | 0.23 | 11002 | **9886** | 9886 | -10.14 | 0.21 | 10745 | **10248** | 10248 | -4.63 | 0.22 |
| test-5 | 10209 | **9413** | 9413 | -7.80 | 0.21 | 9743 | **9423** | 9423 | -3.28 | 0.21 | 10118 | **9310** | 9310 | -7.99 | 0.24 | 9743 | **9607** | 9607 | -1.40 | 0.23 |
| test-6 | 10117 | **9272** | 9272 | -8.35 | 0.22 | 9684 | **9412** | 9412 | -2.81 | 0.22 | 9892 | **9243** | 9243 | -6.56 | 0.25 | 9684 | **9276** | 9276 | -4.21 | 0.23 |
| test-7 | 10182 | **8824** | 8824 | -13.34 | 0.23 | 9899 | **9086** | 9086 | -8.21 | 0.22 | 10333 | **9219** | 9219 | -10.78 | 0.24 | 9629 | **9262** | 9262 | -3.81 | 0.23 |
| test-8 | 9456 | **8787** | 8787 | -7.07 | 0.25 | 9443 | **8874** | 8874 | -6.03 | 0.25 | 9267 | **9034** | 9034 | -2.51 | 0.22 | 9267 | **8822** | 8822 | -4.80 | 0.20 |
| test-9 | 10978 | **10470** | 10470 | -4.63 | 0.22 | 10980 | **10491** | 10491 | -4.45 | 0.21 | 10948 | **10491** | 10491 | -4.17 | 0.23 | 10745 | **10317** | 10317 | -3.98 | 0.20 |
| test-10 | 9568 | **8884** | 8884 | -7.15 | 0.25 | 9809 | **8847** | 8847 | -9.81 | 0.24 | 9553 | **9043** | 9043 | -5.34 | 0.21 | 9553 | **8780** | 8780 | -8.09 | 0.22 |
| test-11 | 10456 | **9773** | 9773 | -6.53 | 0.22 | 10207 | **9811** | 9811 | -3.88 | 0.23 | 10802 | **9478** | 9478 | -12.26 | 0.25 | 9621 | 9610 | 9610 | -0.11 | 0.23 |
| test-12 | 9728 | **9123** | 9123 | -6.22 | 0.24 | 9932 | **9186** | 9186 | -7.51 | 0.22 | 9860 | **9245** | 9245 | -6.24 | 0.24 | 9735 | **9361** | 9361 | -3.84 | 0.23 |
| test-13 | 10829 | **10246** | 10246 | -5.38 | 0.23 | 10407 | **10234** | 10234 | -1.66 | 0.21 | 10232 | **10190** | 10190 | -0.41 | 0.24 | 10232 | **9948** | 9948 | -2.78 | 0.22 |
| test-14 | 10527 | **9779** | 9779 | -7.11 | 0.23 | 10368 | **9882** | 9882 | -4.69 | 0.22 | 10203 | **10057** | 10057 | -1.43 | 0.21 | 10111 | 10165 | 10165 | 0.53 | 0.24 |
| test-15 | 10146 | **9513** | 9513 | -6.24 | 0.23 | 10288 | **9668** | 9668 | -6.03 | 0.22 | 10310 | **9434** | 9434 | -8.50 | 0.23 | 10120 | **9288** | 9288 | -8.22 | 0.24 |
| test-16 | 10624 | **10440** | 10440 | -1.73 | 0.23 | 10872 | **10305** | 10305 | -5.22 | 0.21 | 10704 | **10109** | 10109 | -5.56 | 0.20 | 10704 | **10201** | 10201 | -4.70 | 0.25 |
| test-17 | 10653 | **10005** | 10005 | -6.08 | 0.21 | 10102 | **9487** | 9487 | -6.09 | 0.23 | 10654 | **10075** | 10075 | -5.43 | 0.22 | 10102 | **9910** | 9910 | -1.90 | 0.23 |
| test-18 | 10140 | **9546** | 9546 | -5.86 | 0.21 | 9651 | **9174** | 9174 | -4.94 | 0.21 | 10047 | **9210** | 9210 | -8.33 | 0.22 | 9651 | **9419** | 9419 | -2.40 | 0.22 |
| test-19 | 9490 | **8956** | 8956 | -5.63 | 0.25 | 10009 | **9354** | 9354 | -6.54 | 0.22 | 9714 | **9410** | 9410 | -3.13 | 0.21 | 9714 | **9339** | 9339 | -3.86 | 0.21 |
| test-20 | 9500 | **8578** | 8578 | -9.71 | 0.21 | 9415 | **8589** | 8589 | -8.77 | 0.23 | 9183 | **8478** | 8478 | -7.68 | 0.21 | 9183 | **8489** | 8489 | -7.56 | 0.23 |
| Aver | | | | -6.31 | 0.23 | | | | -5.59 | 0.22 | | | | -5.85 | 0.22 | | | | -3.83 | 0.22 |

Improved solutions are highlighted in boldface

other parameters, the best configuration is presented as follows: As finding the best configuration by running all instances would be computationally too expensive, we implement our numerical analysis on some selected instances.

Table 4. The experiment results for the RCDMP-instances ($n$ = 50)

| Instances | α = 0 | | | | | α = 0.5 | | | | | α = 0.75 | | | | | α = 1 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TS | Our algorithm | | | | TS | Our algorithm | | | | TS | Our algorithm | | | | TS | Our algorithm | | | |
| | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* |
| test-1 | 13691 | **13237** | 13350.1 | -3.32 | 1.13 | 14069 | **13232** | 13453.8 | -5.95 | 0.98 | 14055 | **13392** | 13534.8 | -4.72 | 0.95 | 13691 | **13107** | 13364.9 | -4.27 | 1.09 |
| test-2 | 13419 | **11881** | 12296.6 | -11.46 | 1.03 | 13369 | **12292** | 12509.3 | -8.06 | 0.97 | 12785 | **12577** | 12688.1 | -1.63 | 1.12 | 12785 | **12285** | 12654.3 | -3.91 | 1.10 |
| test-3 | 13740 | **12911** | 13102.2 | -6.03 | 0.93 | 14210 | **13020** | 13230.6 | -8.37 | 1.10 | 13521 | **12781** | 13084.4 | -5.47 | 1.04 | 13521 | **13032** | 13161.5 | -3.62 | 1.09 |
| test-4 | 14848 | **14529** | 14616.9 | -2.15 | 0.98 | 15178 | **14016** | 14510.3 | -7.66 | 1.15 | 15261 | **14380** | 14585.3 | -5.77 | 0.95 | 13691 | **13691** | 13691 | 0.00 | 1.18 |
| test-5 | 15539 | **14227** | 14503.1 | -8.44 | 0.95 | 14798 | **13928** | 14319.8 | -5.88 | 1.00 | 14770 | **14466** | 14591.5 | -2.06 | 1.00 | 13691 | 14243 | 14372 | 4.03 | 0.96 |
| test-6 | 14925 | **13778** | 13959.1 | -7.69 | 0.98 | 14496 | **13956** | 14132.7 | -3.73 | 1.13 | 14776 | **14069** | 14262.7 | -4.78 | 1.08 | 13691 | **13691** | 13691 | 0.00 | 1.11 |
| test-7 | 15657 | **13425** | 14141.6 | -14.26 | 1.03 | 15658 | **14144** | 14480.3 | -9.67 | 1.10 | 15087 | **13917** | 14113.4 | -7.76 | 0.96 | 13691 | **13691** | 13691 | 0.00 | 0.97 |
| test-8 | 15966 | **14819** | 15025.2 | -7.18 | 1.06 | 15555 | **14758** | 14873.9 | -5.12 | 0.90 | 15902 | **14646** | 14829.3 | -7.90 | 1.12 | 13691 | **14765** | 14992.1 | 7.84 | 0.94 |
| test-9 | 15815 | **15446** | 15599.2 | -2.33 | 1.04 | 16105 | **14974** | 15187.9 | -7.02 | 1.08 | 16176 | **15003** | 15159.4 | -7.25 | 0.97 | 13691 | **13691** | 13691 | 0.00 | 1.08 |
| test-10 | 12819 | **11956** | 12120.8 | -6.73 | 1.16 | 12686 | **12079** | 12283.5 | -4.78 | 1.02 | 12800 | **11865** | 12232.4 | -7.30 | 1.18 | 12686 | **11961** | 12177.1 | -5.71 | 1.04 |
| test-11 | 15237 | **13977** | 14221.7 | -8.27 | 1.06 | 14370 | **14125** | 14321.8 | -1.70 | 1.17 | 14670 | **13780** | 14137.8 | -6.07 | 0.98 | 13691 | **13620** | 13941.3 | -0.52 | 1.04 |
| test-12 | 15204 | **14241** | 14517.8 | -6.33 | 1.18 | 15275 | **14134** | 14464.5 | -7.47 | 0.90 | 15399 | **14093** | 14598.6 | -8.48 | 1.13 | 13691 | 14417 | 14705.4 | 5.30 | 1.10 |
| test-13 | 14119 | **12908** | 13134.9 | -8.58 | 1.09 | 14347 | **12900** | 13084.7 | -10.09 | 1.04 | 14406 | **13025** | 13195.3 | -9.59 | 0.96 | 13691 | **13113** | 13234.6 | -4.22 | 1.13 |
| test-14 | 14551 | **13363** | 13659.7 | -8.16 | 1.19 | 14939 | **13070** | 13566.8 | -12.51 | 1.03 | 14721 | **13750** | 13989.1 | -6.60 | 0.99 | 13691 | **13358** | 13520.7 | -2.43 | 1.01 |
| test-15 | 14657 | **13939** | 14135.1 | -4.90 | 0.97 | 14845 | **13964** | 14136.4 | -5.93 | 1.04 | 14080 | **13706** | 14172.4 | -2.66 | 0.93 | 13691 | 13748 | 14181 | 0.42 | 1.10 |
| test-16 | 15053 | **14065** | 14267.6 | -6.56 | 1.10 | 15165 | **14186** | 14288.4 | -6.46 | 1.13 | 15312 | **14257** | 14360.6 | -6.89 | 1.07 | 13691 | 14008 | 14220.2 | 2.32 | 1.02 |
| test-17 | 14587 | **13213** | 13503.4 | -9.42 | 0.99 | 14208 | **13072** | 13580.2 | -8.00 | 1.00 | 14413 | **13561** | 13745.3 | -5.91 | 1.11 | 13691 | **13247** | 13558.4 | -3.24 | 1.15 |
| test-18 | 13652 | **12727** | 12925.8 | -6.78 | 1.10 | 13844 | **13035** | 13187.9 | -5.84 | 1.14 | 14403 | **12862** | 13044.3 | -10.70 | 1.06 | 13691 | **12852** | 13060 | -6.13 | 1.15 |
| test-19 | 15883 | **14600** | 14744.2 | -8.08 | 1.11 | 15521 | **14490** | 14840.5 | -6.64 | 1.04 | 15902 | **14808** | 15000.2 | -6.88 | 1.03 | 13691 | 14905 | 14995.9 | 8.87 | 0.98 |
| test-20 | 14771 | **13700** | 13918.6 | -7.25 | 0.92 | 15259 | **13815** | 14111.2 | -9.46 | 0.91 | 14649 | **13885** | 14134.4 | -5.22 | 1.09 | 13691 | 13939 | 14109.7 | 1.81 | 1.08 |
| Aver | | | | -7.20 | 1.05 | | | | -7.02 | 1.04 | | | | -6.18 | 1.04 | | | | -0.17 | 1.07 |

Improved solutions are highlighted in boldface

Table 5. The experiment results for the RCDMP-instances ($n$ = 100)

| Instances | α = 0 | | | | | α = 0.5 | | | | | α = 0.75 | | | | | α = 1 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TS | Our algorithm | | | | TS | Our algorithm | | | | TS | Our algorithm | | | | TS | Our algorithm | | | |
| | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* | *Best.Sol* | *Best.Sol* | *Aver.Sol* | $gap_2$ | *T* |
| test-1 | 40496 | **39988** | 40567.1 | -1.25 | 11.75 | 39729 | **39585** | 39950.3 | -0.36 | 11.21 | 40067 | **39781** | 40245.2 | -0.71 | 11.04 | 38855 | **38668** | 39105.9 | -0.48 | 12.47 |
| test-2 | 39441 | **39207** | 38988.3 | -0.59 | 11.62 | 38249 | **38127** | 38988.3 | -0.32 | 11.35 | 39585 | **39263** | 38988.3 | -0.81 | 10.45 | 38745 | **38686** | 38988.3 | -0.15 | 11.29 |
| test-3 | 40045 | **40029** | 40339.6 | -0.04 | 12.61 | 39804 | **39690** | 40132.8 | -0.29 | 11.10 | 40134 | **40037** | 40310.3 | -0.24 | 11.76 | 40323 | **40239** | 40451.8 | -0.21 | 12.66 |
| test-4 | 38882 | **38762** | 39149.6 | -0.31 | 10.79 | 38402 | **38395** | 38879.7 | -0.02 | 12.29 | 38067 | **37977** | 38470.2 | -0.24 | 10.79 | 39190 | **39182** | 39490.1 | -0.02 | 11.17 |
| test-5 | 40414 | **40404** | 40552.2 | -0.02 | 10.95 | 40752 | **40726** | 41003.9 | -0.06 | 11.88 | 40885 | **40741** | 41171.2 | -0.35 | 10.13 | 40212 | **40142** | 40408.9 | -0.17 | 12.31 |
| test-6 | 38338 | **38271** | 38543.2 | -0.17 | 10.36 | 39100 | **38744** | 39165.2 | -0.91 | 12.32 | 38954 | **38916** | 39298.9 | -0.10 | 12.26 | 39257 | **39203** | 39332.6 | -0.14 | 11.19 |
| test-7 | 38729 | **38243** | 38954.3 | -1.25 | 12.82 | 38005 | **37984** | 38536.2 | -0.06 | 12.80 | 37812 | **37729** | 37937.1 | -0.22 | 10.73 | 38567 | **38492** | 38775.9 | -0.19 | 12.43 |
| test-8 | 41141 | **41056** | 41281 | -0.21 | 11.94 | 41087 | **41076** | 41244.2 | -0.03 | 12.92 | 40663 | **40627** | 40985.1 | -0.09 | 11.33 | 39661 | **39585** | 39837.9 | -0.19 | 12.27 |
| test-9 | 38705 | **38311** | 39190.3 | -1.02 | 11.44 | 38308 | **38106** | 38804 | -0.53 | 10.58 | 39824 | **39602** | 39909.4 | -0.56 | 12.06 | 38982 | **38954** | 39303.2 | -0.07 | 11.13 |
| test-10 | 38219 | **38205** | 38519.9 | -0.04 | 11.92 | 38888 | **38883** | 39614.1 | -0.01 | 10.42 | 39188 | **39173** | 39280.5 | -0.04 | 11.08 | 38318 | **38226** | 38453.4 | -0.24 | 10.65 |
| test-11 | 39112 | **39009** | 39234.5 | -0.26 | 11.63 | 39392 | **39366** | 39860.7 | -0.07 | 12.09 | 39299 | **39279** | 39536.5 | -0.05 | 12.21 | 38921 | **38763** | 39189 | -0.41 | 12.37 |
| test-12 | 38519 | **38447** | 38768 | -0.19 | 11.94 | 38095 | **37977** | 38160.3 | -0.31 | 10.28 | 38102 | **38007** | 38279.7 | -0.25 | 11.18 | 38076 | **38038** | 38196.4 | -0.10 | 12.85 |
| test-13 | 40809 | **40765** | 41150.4 | -0.11 | 11.63 | 39986 | **39738** | 40480.1 | -0.62 | 11.58 | 40233 | **40219** | 40738.8 | -0.03 | 12.05 | 40670 | **40648** | 40888.2 | -0.05 | 10.98 |
| test-14 | 40199 | **39993** | 40849 | -0.51 | 12.16 | 40745 | **40727** | 41192.1 | -0.04 | 11.59 | 39386 | **39330** | 40120.7 | -0.14 | 12.11 | 40294 | **40157** | 40518.6 | -0.34 | 12.01 |
| test-15 | 39754 | **39728** | 40062.3 | -0.07 | 11.57 | 39912 | **39876** | 39990.7 | -0.09 | 12.58 | 39304 | **39288** | 39692.7 | -0.04 | 11.33 | 39026 | **38987** | 39362.1 | -0.10 | 11.32 |
| test-16 | 40166 | **40061** | 40354.5 | -0.26 | 12.98 | 40352 | **40211** | 40609.3 | -0.35 | 11.45 | 39969 | **39911** | 40193.4 | -0.15 | 10.06 | 40932 | **40897** | 41023.9 | -0.09 | 12.50 |
| test-17 | 39363 | **39289** | 39682.5 | -0.19 | 10.66 | 38991 | **38885** | 39433.9 | -0.27 | 11.18 | 39531 | **39505** | 39932.4 | -0.07 | 10.99 | 40096 | **40017** | 40178.3 | -0.20 | 12.31 |
| test-18 | 40212 | **40169** | 40362.5 | -0.11 | 10.32 | 39018 | **38895** | 39129.3 | -0.32 | 12.01 | 40031 | **39832** | 40099.3 | -0.50 | 11.27 | 40089 | **40079** | 40532 | -0.02 | 10.50 |
| test-19 | 39875 | **39861** | 40111.9 | -0.04 | 10.33 | 39420 | **39137** | 39958.2 | -0.72 | 12.22 | 39675 | **39349** | 39914 | -0.82 | 10.81 | 39091 | **39067** | 39462.2 | -0.06 | 12.59 |
| test-20 | 38811 | **38810** | 38910.7 | 0.00 | 10.19 | 38204 | **38086** | 38291.2 | -0.31 | 11.56 | 39137 | **39126** | 39230.6 | -0.03 | 10.59 | 37869 | **37864** | 38342 | -0.01 | 12.97 |
| Aver | | | | -0.33 | 11.48 | | | | -0.28 | 11.67 | | | | -0.27 | 11.21 | | | | -0.16 | 11.90 |

Improved solutions are highlighted in boldface

Table 6. The average experimental results for RCDMP-instances

| Instance | α = 0 | | α = 0.5 | | α = 0.75 | | α = 1 | |
|---|---|---|---|---|---|---|---|---|
| | $gap_2$ [%] | $T$ | $gap_2$ [%] | $T$ | $gap_2$ [%] | $T$ | $gap_2$ [%] | $T$ |
| 30 | -6.03 | 0.13 | -5.24 | 0.12 | -4.92 | 0.12 | -0.89 | 0.13 |
| 40 | -6.31 | 0.23 | -5.59 | 0.22 | -5.85 | 0.22 | -3.83 | 0.22 |
| 50 | -7.20 | 1.05 | -7.02 | 1.04 | -6.18 | 1.04 | -0.17 | 1.07 |
| 100 | -0.33 | 11.48 | -0.28 | 11.67 | -0.27 | 11.21 | -0.16 | 11.90 |

Table 7. The experimental results for DMP-instances

| Instances | GRASP– VND | | GRASP–VNS | | GVNS-1 | | GVNS-2 | | TS-RVNS | | Our algorithm | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $gap_1$ [%] | $T$ | $gap_1$ [%] | $T$ | $gap_1$ [%] | $T$ | $gap_1$ [%] | $T$ | $gap_1$ [%] | $T$ | $gap_1$ [%] | $T$ |
| 10 | 33.04 | 0.00 | 33.04 | 0.00 | 33.04 | 0.07 | 33.04 | 0.23 | 33.04 | 0.00 | **33.04** | 0.10 |
| 20 | 39.63 | 0.00 | 40.34 | 0.04 | 39.21 | 0.41 | 39.29 | 0.87 | 39.29 | 0.07 | **39.29** | 0.21 |
| 50 | 45.30 | 0.00 | 47.20 | 3.54 | 43.90 | 2.41 | 43.97 | 4.36 | 43.97 | 0.65 | **43.97** | 1.12 |
| 100 | 43.26 | 1.11 | 44.28 | 103.92 | 40.79 | 25.8 | 40.82 | 38.4 | 40.82 | 7.85 | **40.82** | 13.52 |
| 200 | 43.16 | 3.75 | 38.77 | 3995.0 | 37.95 | 124.3 | 38.14 | 93.8 | 38.14 | 78.08 | **38.14** | 121.21 |
| 500 | 46.11 | 604.89 | 49.50 | 1524.09 | 39.41 | 414.05 | 39.54 | 421.72 | - | - | 41.94 | 207.94 |
| 1000 | 45.99 | 3876.51 | 42.35 | 9311.21 | 41.51 | 942.47 | 41.28 | 921.52 | - | - | **41.28** | 2079.67 |

Improved gap is highlighted in boldface

Table 8. The experimental results for the instances in TSPLIB

| Instances | *UB* | Our algorithm | | |
|---|---|---|---|---|
| | | *Best.Sol* | *Aver.Sol* | *T* |
| dantzig42 | 12528* | **12528** | 12528 | 1.16 |
| att48 | 209320* | **209320** | 209320 | 1.40 |
| eil51 | 10178* | **10178** | 10178 | 1.70 |
| berlin52 | 143721* | **143721** | 143721 | 1.20 |
| st70 | 20557* | **20557** | 20557 | 4.21 |
| KroA100 | 983128* | **983128** | 983128 | 14.52 |
| KroC100 | 961324* | **961324** | 961324 | 12.41 |
| KroD100 | 976965* | **976965** | 976965 | 13.31 |
| Eil101 | 27513 | **27513** | 27513 | 12.41 |
| Pr124 | 3154346 | **3154346** | 3154346 | 16.31 |
| Lin105 | 603910 | **603910** | 603910 | 15.22 |

* the optimal values

Table 9. Comparison of the optimal solution for the DMP with the best-found CRDMP-solution using the CRDMP objective function

| instances | α = 0 | | α = 0.5 | | α = 0.75 | | α = 1 | |
|---|---|---|---|---|---|---|---|---|
| | %diff | feasibility | %diff | feasibility | %diff | feasibility | %diff | feasibility |
| test-x-30 | 4.78 | No | 4.79 | No | 4.98 | No | 5.36 | No |
| test-x-40 | 6.24 | No | 6.18 | No | 6.17 | No | 6.10 | No |

This configuration determined in multiple combinations are tested and the one who has presented the best solution is chosen. In Table 1, we define a range for each of the five parameters that yields 1200 different parameter combinations and run the algorithm for some selected instances of these combinations. Each run is limited to 1000 iterations but was stopped when the best-known solution is found. This exhaustive search for the best parameter combinations is useful as a benchmark for evaluating the algorithm. Looking at the parameter combinations, we find the following settings so that our algorithm obtains the best solutions: $\beta_r = 5, \beta_\eta = 5, \alpha = 1$, $p = 1$, and $\tau_0 = 10$. Lastly, the last aspect to discuss is the stop criterium of the algorithm. The parameter m is set to 10 to balance between

TABLE I
THE VARIABLE PARAMETERS

| Parameter | Value Range |
|---|---|
| $\beta_r$ | $0 \leq \beta_r \leq 10$, incremented by 0.25 |
| $\alpha$ | $0.5 \leq \alpha \leq 1.5$, incremented by 0.25 |
| $\beta_\eta$ | $0 \leq \beta_\eta \leq 10$, incremented by 0.25 |
| $\tau_0$ | $0 \leq \tau_0 \leq 20$, incremented by 5 |
| $p$ | $0.3 \leq p \leq 1$, incremented by 0.2 |

computation time and efficiency. This parameter setting has thus been used in the following experiments.

## 2. Bounds

To evaluate the efficiency of the metaheuristic algorithm, its solution can be compared to 1) the optimal solution (*OPT*); and 2) a good upper bound (*UB*). We define the improvement of our algorithm with respect to *Best.Sol* (*Best.Sol* is the best solution found by our algorithm) in comparison with the optimal solution ($gap_1[\%]$), and upper bound ($gap_2[\%]$) in percent respectively as follows:

$$gap_1[\%] = \frac{Best.Sol - LB}{LB} \times 100\% \quad (8)$$

$$gap_2[\%] = \frac{Best.Sol - UB}{UB} \times 100\% \quad (9)$$

We choose several state-of-the-art metaheuristic algorithms for RCDMP, and DMP [4, 17, 19] as a baseline in our research. Note that: The value of *LB* is described in [17] while the value of *UB* is given by the algorithm in [4].

## 3. Results and Comparative Analysis

The tables present *Best.Sol*, *Aver.Sol*, and *T*, respectively. Tables from 2 to 5 compare the results of our algorithm with the solutions obtained by Ban et al. [4] for the RCDMP-instances. Column TS in Tables 2 to 5 corresponds to the best solution of Tabu Search [4]. Column $\alpha$ is corresponding to the results with different resource constraint values. The results in Table 6, which are the average values calculated from Table 2 to 5. Tables 7 to 8 compare our algorithm with the state-of-the-art metaheusitics in terms of $gap_2$ for DMP-instances. Table 9 compares the optimal solutions with the best-found CRDMP-solutions using the CRDMP objective function. For DMP-instances, the optimal solutions are extracted from the exact algorithm in [2].

Tables 2 to 5 compare the results obtained by Ban et al. [4] to our algorithm for 320 instances. The results show that our algorithm reaches better solutions at a reasonable amount of time. Regarding the average $gap_2$ in Table 6, our algorithm quality improves 7.20% compared with Ban et al.'s algorithm [4]. The improvement is significant since it can be observed that our algorithm is capable of finding the new best solutions for the 303 RCDMP-instances. Note that: the found-best solutions is highlighted in bold line.

Table 7 compares the results of our algorithm with the lower bounds obtained from [19] with 100, 200, 500, and 1000 vertices for DMP. Our algorithm gives much better solutions than those of GVNS-1 [17], GRASP-VND and GRASP-VNS [19] for all instances. In comparison with GVNS-2 [17], and TS [3], for most cases, our algorithm gives the same results.

Table 8 presents the average results of our algorithm on instances with up to 150 vertices selected from the TSPLIB [23]. These solutions are compared with optimal values, found by Abeledo et al. [1]. It is noteworthy that our algorithm obtains the optimal solutions with up to 100 vertices such as KroA100, KroB100, KroC100, and KroD100 at reasonable amount of time. The experimental results show that the proposed algorithm can be applied well to the DMP (note that: the DMP is another variant of the RCDMP when resource limit is ignored), though it is not designed to solve it. Therefore, our algorithm can easily adapt to similar variants of the problem.

Table 9 shows that the optimal solutions for the DMP are generally not good solutions to the CRDMP on the same instance. Specifically, our algorithm is better than the optimal solution for the DMP using the CRDMP objective function since the difference between their objective values is up to 6.24%. Although the difference is not large, all these optimal solutions for the DMP are not feasible solutions for the CRDMP. Therefore, the methods designed for the DMP may not be adapted easily to solve the CRDMP. Developing an algorithm for the CRDMP is very necessary.

The average scaled running time of our algorithm is better than those of A. Salehipour et al.'s [19], and grow quite moderate with the one of Mladenovic et al.'s [17] and Ban et al.'s algorithm [4].

## IV. DICUSSIONS

In general, metaheuristic is a suitable approach for solving the NP-hard problems in short computation time. Recently, a metaheuristic for solving the problem has been proposed. The algorithm is developed in a sequential heuristic approach which starts with a single initial solution, and at each step of the search, the current solution is replaced by another solution found in its neighborhood. Since the search space of the problem is a combinatorial explosion, these algorithms only explore a subset of the search space. Therefore, they can easily fall into the local optimal in some cases. Our hybrid algorithm that brings together the components of the ACO, and RVND can overcome this

drawback by searching from multiple promising solution areas, thus, its exploring search space is extended. As a result, the chances of finding better solutions are higher. To summarize, our scheme includes new features as follows:

- Two kinds of ants coexist in the ACO. The ant group that includes the dull ants can obtain more foods than the group consisting of only the intelligent ants. It is suitable in the real ant's world.
- Every search algorithm needs to address the exploration and exploitation of a search space. Exploration is the process of visiting entirely new regions of a search space, whilst exploitation is the process of visiting those regions of a search space within the neighborhood of previously visited points. To be successful, a search algorithm needs to establish a good ratio between exploration and exploitation. The ACO is also combined with the RVND to keep a balance between exploration and exploitation. In this combination, the ACO allows us to explore the promising solution spaces while the RVND based on a simple principle of systematic switches between different neighborhoods to exploit in these spaces.

Our experiments not only prove these expectations but also indicate that this approach finds the new best known solutions in the 303 instances in comparison with the-state-of-art metaheuristic. Moreover, our experiments show that the methods designed for the DMP [3, 17, 19] may not be adapted easily to solve the CRDMP since the optimal solutions for the DMP are infeasible solutions for the CRDMP. Therefore, it is important for developing a metaheuristic for the CRDMP .

We discuss more about some optimization techniques based on swarm intelligence. There are many optimization techniques such as Genetic Algorithm [9], Particle Swarm Optimization (PSO) [12], Artificial Bee Colony (ABC) [11], and Cuckoo Search (CS) [20], etc. In fact, no optimization technique is better than the others in all cases. However, in the paper, the ACO algorithm is used to solve the problem because: 1) the field of ACO has attracted a large number of researchers, and nowadays a large number of research results of both experimental and theoretical nature exists; 2) Its efficiency is demonstrated through solving the other variants of problem (Traveling Salesman Problem (TSP) [22], Vehicle Routing Problem (VRP) [18], ...); 3) it is very understandable and fast to implement.

## V. CONCLUSIONS

In this paper, we propose the metaheuristic algorithm which is mainly based on the principles of the ACO, and RVND to solve the problem. Extensive numerical experiments on benchmark instances show that on average, our algorithm finds the new best solutions in 303 instances. For close variants of the problem, our algorithm can reach the optimal solutions for the problems with 100 vertices in a reasonable amount of time. When the problem size is too large, it takes longer time to operate the whole algorithm, lowering the efficiency of finding the optimum. In these cases, to reduce the running time, parallelizing ACO [14] and divide and conquer ACO [13] can be applied to large-scale instances. This is a topic for future research.

## ACKNOWLEDGEMENT

This research is funded by Hanoi University of Science and Technology (HUST) under grant number T2017-PC-082.

## REFERENCES

[1] H.G. Abeledo and G. Fukasawa and R. Pessoa and A. Uchoa, "The Time dependent Traveling Salesman Problem: Polyhedra and Branch-cut-and price Algorithm", *J. Math. Prog. Comp.*, Vol. 5, No. 1, 2013, pp. 27-55.

[2] H.B. Ban, K. Nguyen, M.C. Ngo, and D.N. Nguyen, "An Efficient Exact Algorithm for Minimum Latency Problem", *J. PI*, No. 10, 2013, pp. 1-8.

[3] H.B. Ban, and D.N. Nguyen, "A Meta-Heuristic Algorithm Combining between Tabu and Variable Neighborhood Search for the Minimum Latency Problem", *J. FI*, Vol. 156, No. 1, 2017, pp. 21-41

[4] H.B. Ban, "Using Metaheuristic for Solving the Resource-Constrained Deliveryman Problem", *Proc. SOICT*, 2019, pp. 917-935.

[5] A. Lucena, "Time-dependent Traveling Salesman Problem-the Deliveryman case", *J. Networks*, Vol. 20, No. 6, 1990, pp. 753-763.

[6] A. Blum, P. Chalasani, D. Coppersmith, W. Pulleyblank, P. Raghavan, and M. Sudan, "The Minimum Latency Problem", *Proc. STOC*, 1994, pp. 163-171.

[7] M. Dorigo, and T. Stutzle, "Ant Colony Optimization", *Bradford Books*, London, 2004.

[8] M. Fischetti, G. Laporte, S. Martello, "The Delivery-man Problem and Cumulative Matroids", *J. Oper Res*, Vol. 41, No. 6, 1993, pp. 1055–1064.

[9] D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", *Addison-Wesley, Reading, Massachusetts*, 1989.

[10] H. Hasegawa, "Optimization of GROUP behavior, Japan Ethological Society Newsletter", Vol. 43, 2004, pp. 22-23.

[11] D. Karaboga, “An Idea Based on Honeybee Swarm for Numerica Optimization”, *Technical Report TR06, Erciyes University*, 2005.

[12] J. Kennedy, R. Eberhart, “Particle Swarm Optimization”, *Conf. Neural Networks*, 1995, pp. 1942–1948.

[13] X. Li, J. Liao, and M. Cai, “Ant Colony Algorithm for Large Scale TSP”, *Proc. ICECING*, 2011, pp.573-576.

[14] M. Manfrin, M. Birattari, T. Stützle, and Marco Dorigo, “Parallel Ant Colony Optimization for the Traveling Salesman Problem”, *Proc. ANTS*, 2006, pp 224-234.

[15] I. Mendez-Diaz, P. Zabala, A. Lucena, “A New Formulation for the Traveling Deliveryman Problem", *J. Discret Appl Math*, No. 156, 2008, pp. 3223-3237.

[16] N. Mladenovic, P. Hansen, “Variable Neighborhood Search", *J. Operations Research*, Vol. 24, 1997, pp. 1097-1100.

[17] N. Mladenovic, D. Urosevi, and S. Hanafi, “Variable Neighborhood Search for the Travelling Deliveryman Problem", *J. 4OR*, Vol. 11, 2012, pp. 1-17.

[18] M. Reimann, K. Doerner, and R.F. Hartl, “Analyzing a Unified Ant System for the VRP and Some of Its Variants”. *S. Cagnoni et al. (Eds.): EvoWorkshops, LNCS*, pp. 300-310, 2003.

[19] A. Salehipour, K. Sorensen, P. Goos, and O.Braysy, “Efficient GRASP+VND and GRASP+VNS metaheuristics for the Traveling Repairman Problem", *J. Operations Research*, Vol. 9, No. 2, 2011, pp. 189-209.

[20] XS. Yang, S. Deb, “Cuckoo Search Via Levy Flights”, *Proc. NaBIC*, 2009, pp. 210–214.

[21] D. S. Johnson, and L. A. McGeoch, “The Traveling Salesman Problem: A Case Study in Local Optimization in Local Search in Combinatorial Optimization", *E. Aarts and J. K. Lenstra, eds.*, 1997, pp. 215-310.

[22] H. X. Huan, N. Linh-Trung, D. Duc-Dong, and T. Huynh, “Solving the Traveling Salesman Problem with Ant Colony Optimization: A Revisit and New Efficient Algorithms”, *J. REV*, Vol. 2, No. 3–4, 2012, pp. 121-129.

[23] http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html.

**Ha-Bang Ban** Ha-Bang Ban received his B.E. in Information Technology, and Ph.D in Computer Science at Hanoi University of Science and Technology (HUST), Vietnam in 2006, 2015, respectively. He is currently a lecturer at the School of Information and Communication Technology (SOICT), HUST, Vietnam. Research interests of Dr. Ban include algorithms, graphs, optimization, and logistics, etc. He has published many publications in peer-reviewed international journals and conferences.

# Impact of Inter-Channel Interference on Shallow Underwater Acoustic OFDM Systems

Do Viet Ha[1], Nguyen Tien Hoa[2], Nguyen Van Duc[2]

[1] Faculty of Electrical and Electronic Engineering, University of Transport and Communications, Hanoi, Vietnam

[2] School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam

Correspondence: Nguyen Van Duc, duc.nguyenvan1@hust.edu.vn

Communication: received 27 July 2020, revised 27 August 2020, accepted 30 September 2020

Digital Object Identifier: 10.32913/mic-ict-research.v2020.n1.929

**Abstract: This paper investigates the impacts of Inter-Channel Interference (ICI) effects on a shallow underwater acoustic (UWA) orthogonal frequency-division multiplexing (OFDM) communication system. Considering both the turbulence of the water surface and the roughness of the bottom, a stochastic geometry-based channel model utilized for a wide-band transmission scenario has been exploited to derive a simulation model. Since the system bandwidth and the sub-carrier spacing is very limited in the range of a few kHz, the channel capacity of a UWA system is severely suffered by the ICI effect. For further investigation, we construct the signal-to-noise-plus-interference ratio (SINR) based on the simulation model, then evaluate the channel capacity. Numerical results show that the various factors of a UWA-OFDM system as subcarriers, bandwidth, and OFDM symbols affect the channel capacity under the different Doppler frequencies. Those observations give hints to select good parameters for UWA-OFDM systems.**

**Keywords:** *Underwater acoustic (UWA), geometry-based channel modeling, underwater OFDM systems*

## I. Introduction

Underwater information systems have many applications in commerce and military [1]. Nowadays, the development and improvement of the service quality of the system is facing many challenges such as frequency spectrum limitation, time and frequency-dependent channel characteristics [2, 3]. It is a fact that the velocity of water sound waves is much lower than the velocity of electromagnetic waves. Therefore, the prominent features of the UWA channels are a large delay spread and strong Doppler effects [4]. The OFDM technique is widely used thanks to its ability to eliminate the inter-symbol interference (ISI) which is caused by a large delay spread [5, 6]. However, OFDM systems are very easily influenced by the Doppler shifts that result in the so-called inter-carrier interference (ICI) and degrade the system performance dramatically [7].

In shallow water environments, acoustic communication systems are suffered from the Doppler shifts with two main sources, which are the relative movement between the transmitter/the receiver and the disturbance from the water surface [8, 9]. In particular, a surface displacement process usually varies very fast over time, thus an unpredictable Doppler frequency shift creates many difficulties in identifying and compensating the Doppler shifts in the receiver [10, 11]. Therefore, system performance should be evaluated under the Doppler effects. In this paper, we focus on analyzing the system capacity because it is an important factor to design components in a communication system.

UWA-OFDM system capacity studies are still an open area up to now as a consequence of the lack of standard channel models [12, 13]. Besides, many studies have ignored the ICI effect in the channel capacity formulation [14–18]. For the sake of simplicity, the time-invariant UWA channel model in which the ICI effect does not occurred has been used in [14–16] to analyze system capacity. The authors in [17, 18] have considered time-variant channels to analyze the channel capacity versus the SNR, which means that the ICI effect has not been taken into account. Furthermore, ICI is also computed by the time correlation function converted from the Doppler spectrum form of the UWA channel, which is assumed to be Jack, uniform or two-path spectrum [11, 19, 20]. These assumptions may not be applicable to UWA channels due to the complicated time varying characteristics of the shallow UWA propagation environments. In addition, the UWA channels are considered as non stationary channels such that the conversion between the Doppler spectrum and the time correlation function should be accompanied by certain conditions. In [21], the capacity of UWA-OFDM has been derived from the SINR which is considered to be the same for all subcarriers. However, the UWA-OFDM system is inherently a wideband

system [16, 22], the Doppler shifts over subcarriers are thus different from each other. Consequently, it is nesseary to investigate the channel capacity of UWA systems under the ICI impacts over every subcarriers.

Over the past few decades, although a large variety of UWA channel models have been proposed, there is still no typical model that can be applied for all UWA channels because of differences in geographical areas, weather conditions, and seasonal cycles [17]. The statistical characteristics of shallow UWA channels are greatly affected by the distribution of scatterers on the surface and the bottom, which results in the different delays and Doppler frequency shifts of the transmit signals. The geometry-based UWA channel model in [23] has been derived from computing the number of scatterers and their positions using the wave-guide geometry, which does not represent the surface displacement. In [22, 24, 25], the proposed UWA channel models concentrate on analyzing the path loss and the multipath propagation whereas the Doppler effect has not been integrated with the models.

In this paper, we use the geometry-based channel model for shallow water with rough surface conditions [26] to investigate the quality of the UWA-OFDM system. Particularly, the scattering points are assumed to be uniformly distributed between the transmitter and the receiver. From the assumption, the pulse response time variation of the channel pattern is obtained. Using this channel impulse response, the SINR of each sub-carrier is derived for the considered system. As aforementioned, the SINR of shallow UWA channels with rough surface transmission system is strongly dependent on ICI effects which is caused by Doppler shifts. We have, in contrast to other works, derived SINR expression of each sub-carrier with considering of the time-variant UWA channel model. We have also used the geometrical scattering model for representing the characteristics of shallow water with rough surface conditions to evaluate the channel capacity. The channel capacity is further formulated as the superposition of all the single channel capacity from each sub-channel. By analyzing the numerical results, a set of suitable parameters for the considered UWA-OFDM system is found, which includes the number of subcarriers, signal bandwidth and the length of the OFDM symbol. It should be noticed that these are important parameters and they need to be determined appropriately in order to eliminate ISI while limiting the ICI effects.

The rest of the paper is organized as follows: Section II presents the UWA geometry-based channel model which is used to derive the time-variant channel impulse response and the channel transfer function. The calculations of the SINR expression and channel capacity for the considered UWA-OFDM system using this channel model are presented in detail in Section III. Section IV shows the simulation results together with discussions. Finally, Section V draws the main conclusions of this paper.

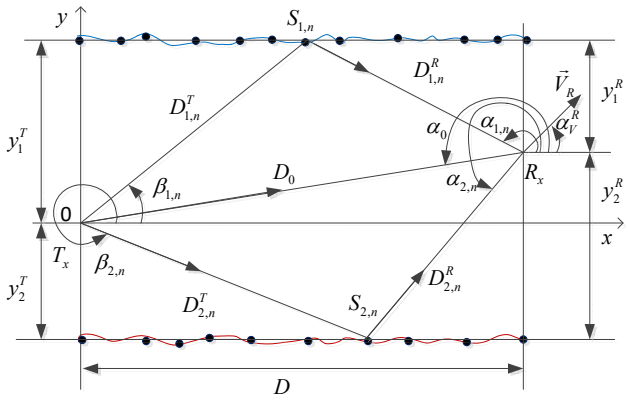


Figure 1. The Geometry-Based channel model for a shallow underwater acoustic channel.

## II. THE UNDERWATER ACOUSTIC GEOMETRY-BASED CHANEL MODEL

The geometry-based channel model [26] is illustrated in Fig. 1, where $D$ denotes the distance between the transmitter (Tx) and receiver (Rx). The scatterers $S_{i,n}(n = 1, 2, \ldots, N_i; i = 1, 2)$ are assumed to be randomly distributed on the surface ($i = 1$) and the bottom ($i = 2$) of a shallow-water environment. The symbols $\alpha_{i,n}, \beta_{i,n}$, $(\beta_{i,n} \neq 0; \alpha_{i,n} \neq \pi)$ are the angle-of-departure (AOD) and the angle-of-arrival (AOA) of the $n$-th path, respectively. In the below subsections, the UWA channel parameters are derived with reference to [26].

### 1. The Channel Impulse Response

The time-variant channel impulse response (TVCIR) $h(\tau, t)$ of the shallow UWA environment is composed of three components, which is given by [26]

$$h(\tau, t) = \sum_{i=0}^{2} h_i(\tau, t). \tag{1}$$

In Eq. (1), the line-of-sight (LOS) component is denoted by $h_0(\tau, t)$, whereas $h_1(\tau, t)$ and $h_2(\tau, t)$ stand for the scattered components from the surface and the bottom, respectively. The LOS part $h_0(\tau, t)$ is specified by

$$h_0(\tau, t) = \sqrt{\frac{c_R}{1 + c_R}} A_s(D_0) A_a(D_0)\ e^{j(2\pi f_0 t + \theta_0)} \delta(\tau - \tau_0), \tag{2}$$

in which $c_R$, $\tau_0$, $f_0$ and $\theta_0$ denote the Rice factor, the propagation delay, the Doppler frequency, and the phase shift of the LOS path, respectively. The function $A_s(D_0)$ is

the propagation loss coefficient due to spherical spreading, which can be obtained by

$$A_s(D) = \frac{1}{D}, \tag{3}$$

where $D$ denotes the total propagation distance in meter. For the LOS path, the distance is defined by

$$D_0 = \sqrt{D^2 + (y_1^T - y_1^R)^2}. \tag{4}$$

The absorption loss coefficient $A_a(D)$ is given by

$$A_a(D) = 10^{-\frac{D\beta}{2000}}. \tag{5}$$

The parameter $\beta$ in Eq. (5) is computed as

$$\beta = 8.68 \times 10^3 \left( \frac{S_a f_T f_c^2 A}{f_T^2 + f_c^2} + \frac{B f_c^2}{f_T} \right) \times \times (1 - 6.54 \times 10^{-4} P)[dB/km], \tag{6}$$

where $A = 2.34 \times 10^{-6}$ and $B = 3.38 \times 10^{-6}$, $S_a$ is salinity (in ppt), $f_c$ is the carrier frequency (in kHz), $f_T$ is the relaxation frequency (in kHz) and $T$ is temperature (in °C). The symbol $P$ denotes the hydro-static pressure (in kg/cm$^2$), which is determined by $P = 1.01(1+0.1h)$, where $h$ is the water depth (in meter). The scattered components $h_1(\tau, t)$ and $h_2(\tau, t)$ of the TVCIR $h(\tau, t)$ are computed by

$$h_i(\tau, t) = \frac{1}{\sqrt{2N_i(1 + c_R)}} \sum_{n=1}^{N_i} A_s(D_{i,n}) A_a(D_{i,n}) \times e^{j(2\pi f_{i,n} t + \theta_{i,n})} \delta(\tau - \tau_{i,n}), \tag{7}$$

in which $f_{i,n}$, $\tau_{i,n}$, and $\theta_{i,n}$ denote the Rice factor, the propagation delay, the Doppler frequency, and the phase shift of the scattered path, respectively. With reference to Fig. 1, the total propagation distance $D_{i,n}$ can be computed as

$$D_{i,n} = \frac{y_i^T}{\sin(\beta_{i,n})} + \frac{y_i^R}{\sin(\alpha_{i,n})}. \tag{8}$$

The parameters of the UWA channel are initiated by using optimum values of $x_{i,n}$

$$x_{i,n}^{\text{opt}} = \frac{D}{N_i} \left( n - \frac{1}{2} \right). \tag{9}$$

Using $x_{i,n}^{\text{opt}}$, the other values AOA $\alpha_{i,n}$ and AOD $\beta_{i,n}$ can be computed as follow

$$\alpha_{i,n} = \begin{cases} \frac{\pi}{2} + \arctan\left(\frac{D - x_{i,n}}{y_1^R}\right), & \text{if } 0 \le x_{i,n} \le D. \\ \pi + \arctan\left(\frac{y_2^R}{D - x}\right), & \text{if } 0 \le x_{i,n} \le D. \end{cases} \tag{10}$$

$$\beta_{i,n} = \begin{cases} \arctan\left(\frac{y_1^T}{x_{i,n}}\right), & \text{if } 0 \le x_{i,n} \le D. \\ \frac{3\pi}{2} + \arctan\left(\frac{x_{i,n}}{y_2^T}\right), & \text{if } 0 \le x_{i,n} \le D. \end{cases} \tag{11}$$

Substituting $\alpha_{i,n}$ and $\beta_{i,n}$ in Eq. (8) allows us to compute $D_{i,n}$. The Doppler frequencies are computed by $f_{i,n} = f_{D,\max} \cos(\alpha_{i,n} - \alpha_v^R)$, where $f_{D,\max}$ stands for the maximum Doppler frequency. The propagation delays can be obtained by $\tau_{i,n} = D_{i,n}/c_s$. The phase shift $\theta_{i,n}$ is assumed to be uniformly distributed over the range $(-\pi, \pi]$. Finally, the scattered components $h_1(\tau, t)$ and $h_2(\tau, t)$ of the TVCIR $h(\tau, t)$ in Eq. (7) are derived.

## 2. Channel Transfer Function

For further analysis of the UWA-OFDM system, the time-variant channel transfer function (TVCTF) $H(f, t)$ needs to be derived by taking the Fourier transform of the TVCIR $h(\tau, t)$, which can be expressed by

$$H(f, t) = \sum_{i=0}^{2} H_i(f, t), \tag{12}$$

where

$$H_0(f, t) = \sqrt{\frac{c_R}{1 + c_R}} A_s(D_0) A_a(D_0) \times e^{j[2\pi(f_0 t - f\tau_0)]}, \tag{13}$$

and

$$H_i(f, t) = \frac{1}{\sqrt{2N_i(1 + c_R)}} \sum_{n=1}^{N_i} A_s(D_{i,n}) A_a(D_{i,n}) \times e^{j[2\pi(f_{i,n} t - f\tau_{i,n}) + \theta_{i,n}]} \tag{14}$$

for $i = 1, 2$.

# III. CHANNEL CAPACITY ANALYSIS UNDER INTERFENCE EFFECTS

This section uses the geometry-based UWA channel simulation model to analyze the ICI effect in the UWA-OFDM system. The SINR of each sub-carrier has been formulated in terms of the time-variant channel transfer function (TVCTF) $H(f, t)$ of the UWA channel model. The capacity estimation of UWA-OFDM system has been analyzed using the results of SINRs.

## 1. SINR Computation

The OFDM base-band signal is given as

$$x[t_n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[f_k] e^{j2\pi nk/N}, \tag{15}$$

where $N$ is the number of sub-carriers, $X_k$ denotes the $k^{\text{th}}$ data-modulated sub-carrier in the OFDM symbol. The OFDM signal at the receiver side is represented as

$$\widehat{x}[t_n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} H[f_k, t_n] X[f_k] e^{i2\pi nk/N} + w[t_n], \tag{16}$$

where $H[f_k, t_n]$ stands for the TVCTF of the UWA channel and $w[t_n]$ denotes the ambient noise in the UWA communication system. After FFT at the receiver, the signal $\widehat{X}[f_k]$ in frequency domain can be expressed as

$$\widehat{X}[f_k] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \widehat{x}[t_n] e^{-i2\pi nk/N}. \tag{17}$$

By using $\widehat{x}[t_n]$ from Eq. (16) in Eq. (17), the $\widehat{X}[f_k]$ is given as in Eq. (25), where $S[f_k]$, $I[f_k]$, $W[f_k]$ denote the desired signal, the interference signal, and the frequency domain of ambient noise $w[t_n]$, respectively, which are computed by

$$S[f_k] = \frac{1}{N} \sum_{n=0}^{N-1} H[f_k, t_n] X[f_k], \tag{18}$$

$$I[f_k] = \frac{1}{N} \sum_{\substack{m=0 \\ m \neq k}}^{N-1} \sum_{n=0}^{N-1} H[f_m, t_n] e^{j2\pi(m-k)n/N} X[f_m], \tag{19}$$

and

$$W[f_k] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} w[t_n] e^{-j2\pi nk/N}. \tag{20}$$

Assuming the Signal to Noise Ratio of the $k^{\text{th}}$ subcarrier in the absence of ICI is denoted by the $\text{SNR}[f_k]$, which is given by

$$\text{SNR}[f_k] = \frac{P_{\text{S}}[f_k]}{P_{\text{N}}[f_k]}, \tag{21}$$

where $P_{\text{S}}[f_k]$ is the receiver power and $P_{\text{N}}[f_k]$ stands for the noise power at the $k^{\text{th}}$ subcarrier. Using (18) and (19), the desired signal power $P_{\text{D}}[f_k]$ and the ICI power $P_{\text{I}}[f_k]$ of the $k^{\text{th}}$ subcarrier can be obtained by

$$P_{\text{D}}[f_k] = \frac{P_{\text{S}}[f_k]}{N^2} \left( \sum_{n=0}^{N-1} H[f_k, t_n] \right)^2, \tag{22}$$

and

$$P_{\text{I}}[f_k] = \frac{P_{\text{S}}[f_k]}{N^2} \left( \sum_{\substack{m=0 \\ m \neq k}}^{N-1} \sum_{n=0}^{N-1} H[f_m, t_n] e^{j2\pi(m-k)n/N} \right)^2. \tag{23}$$

The SINR of the kth sub-carrier is given by

$$\text{SINR}[f_k] = \frac{P_{\text{D}}[f_k]}{P_{\text{I}}[f_k] + P_{\text{N}}[f_k]}. \tag{24}$$

Using Eq. (21), Eq. (22), and Eq. (23), the SINR of the $k^{\text{th}}$ subcarriers can be obtained as in Eq. (26).

## 2. Capacity Analysis

The channel capacity of $k^{\text{th}}$ sub-carrier of an UW-OFDM system is computed from the $\text{SINR}[f_k]$ as follows

$$C_k = \Delta f \log_2(1 + \text{SINR}[f_k]). \tag{27}$$

Suppose that each sub-carrier carries data, the total channel capacity of UWA-OFDM system is calculated by

$$C_{\text{SINR}} = \frac{T_S}{T_S + T_G} \Delta f \sum_{k=0}^{N-1} \log_2(1 + \text{SINR}[f_k]), \tag{28}$$

where $T_S, T_G$ are the symbol duration and the guard length of the UWA-OFDM system, respectively.

In UWA communication systems, the spectral efficiency $C/B$ is a vital parameter in system performance evaluation because of the limited bandwidth. Substituting $\Delta f = B/N$ into Eq. (28), we obtain the spectral efficiency $C/B$ as below

$$\frac{C}{B} = \frac{T_S}{T_S + T_G} \times \frac{1}{N} \times \sum_{k=0}^{N-1} \log_2(1 + \text{SINR}[f_k]). \tag{29}$$

Observing Eq. (29), several constraints need to be considered in determining OFDM transmission parameters to optimize the spectral efficiency. Assuming that the guard length $T_G$ is chosen to be equal to the maximal delay spread $\tau_{\max}$ to remove the ISI noise. With a given number of sub-carriers $N_c$, the bandwidth efficiency coefficient $\beta = T_S/(T_S + T_G)$ is increased by a larger symbol duration $T_S$. However, the larger $T_S = N/B$ makes the sub-carrier spacing $\Delta f = B/N$ decrease. Consequently, the ICI effect will be stronger, which results in degrading SINR. Therefore, the set of parameters $T_S, N_c, B$ should be considered thoughtfully to meet the quality requirements of the system.

The numerical results of SINRs and system capacity in section IV will evaluate the appropriate system parameters, including thoes of $T_S$, $B$, and $N_c$ to achieve optimal spectral efficiency for the UWA-OFDM system with ICI effects.

# IV. RESULTS AND DISCUSSIONS

## 1. Simulation Setting

The results of surveying system performance including SINR and Capacity are averaged over 10 simulations, each running with a time of 10000 OFDM symbols. The main parameters running in the simulation are taken with central carrier frequency $f_c$ = 30 kHz, SNR = 20 dB at receiver, the number of subcarriers $N_c$ = 512, 1024 and 2048, and the signal bandwidth varies between 1 kHz and 30 kHz.

$$\widehat{X}[f_k] = \frac{1}{N}\sum_{m=0}^{N-1}\left[\left(\sum_{n=0}^{N-1} H[f_m, t_n]e^{j2\pi(m-k)}\right)X[f_m]\right] + W[f_k] = S[f_k] + I[f_k] + W[f_k]. \quad (25)$$

$$\mathrm{SINR}[f_k] = \frac{\left|\frac{1}{N}\sum_{n=0}^{N-1} H[f_k, t_n]\right|^2}{\left|\frac{1}{N}\sum_{m=0,m\neq k}^{N-1}\sum_{n=0}^{N-1} H[f_m, t_n]e^{j2\pi(m-k)n/N}\right|^2 + \frac{1}{\mathrm{SNR}[f_k]}}. \quad (26)$$

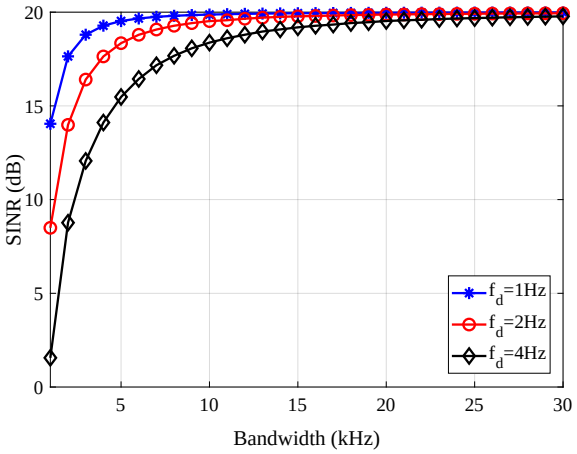


Figure 2. SINRs versus signal bandwidth for different Doppler frequencies.

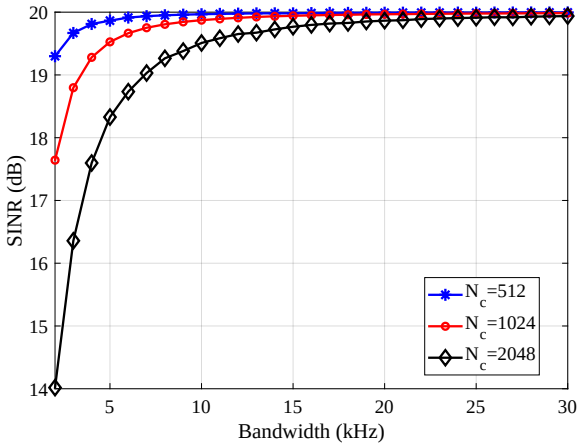


Figure 3. SINR versus signal bandwidth for different numbers of subcarriers ($f_d$= 1Hz).

## 2. SINR Results

Figure 2 shows the SINR results of the UWA-OFDM system with the number of subcarriers $N_c$ = 1024. It is noted that the SINR is calculated for each subcarrier and the results in Fig. 2 is the average of all subcarriers. We see the strong Doppler effect on the SINR when the signal bandwidth is small. With a given number of subcarriers, the smaller signal bandwidth is, the subcarrier spacing $\Delta f = N/B$ is narrower that causes more serious ICI effect and decreasing the SINR. On the contrary, when increasing the signal bandwidth, the larger carrier spacing mitigates the ICI effect. If bandwidth $B$ is large enough, the ICI effect can be neglected and the SINR results approach to the SNR=20 dB.

From the SINR results in Fig. 2, it is observed that with a bandwidth greater than 10 kHz, the ICI effect for the Doppler frequencies of $f_d$ = 1 Hz and 2 Hz is negligible. For the case of $f_d$ = 4 Hz, a bandwidth greater than 20 kHz is required to significantly reduce the ICI effect. Therefore, depending on the hardware capabilities of the system, the impact of ICI on system performance can be significantly reduced if a wide bandwidth is chosen.

Another aspect which should be considered in UWA-OFDM system design is the number of subcarriers $N_c$. For a given bandwidth, the subcarrier spacing is narrower with a larger $N_c$. Consequently, the ICI effect on the OFDM system is more severe, and then the SINR decreases. As the results show in Fig. 3, the SINR is lower in the case of larger number of subcarriers $N_c$. Using these results, the appropriate values of bandwidth $B$ and the number of sub-carrier $N_c$ can be determined to achieve a required SINR of the UWA-OFDM system. For even a very small value of Doppler shift $f_d$ = 1 Hz, the maximum number of subcarriers of $N_c$ = 1024 and the minimum bandwidth of 10 kHz should be selected to avoid the ICI effect. For $N_c$ = 2048, the minimum bandwidth is required to be greater than 20 kHz. However, on one hand, a small number of subcarriers results in mitigating the Doppler effect; on the other hand, it makes the efficiency of the spectrum and the system capacity decrease. The next section will evaluate the capacity to determine the appropriate number of subcarriers for the UWA-OFDM system.

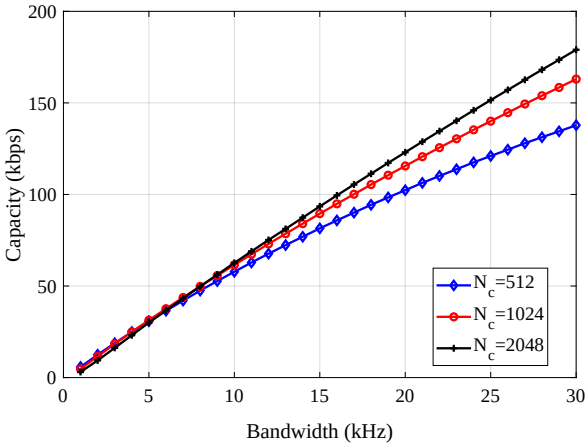


Figure 4. System capacity $C$ (Kbps) versus bandwidth for different numbers of subcarriers ($f_d$ = 1Hz).

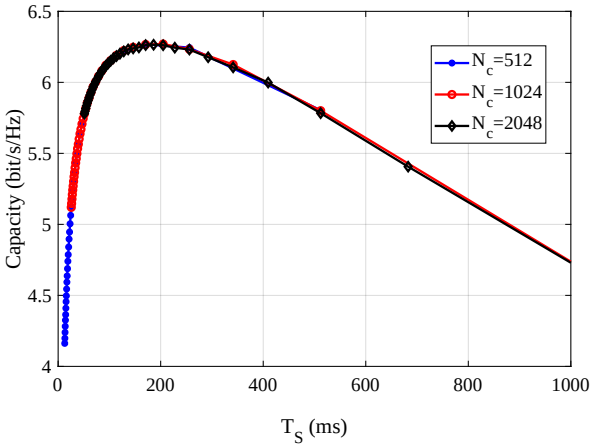


Figure 5. Spectral efficiency $C/B$ (b/s/Hz) versus symbol length $T_S$ for different numbers of subcarriers.

## 3. Capacity Results

The system capacity of the UWA-OFDM system versus the signal bandwidth for different numbers of subcarriers (with $f_d$ = 1Hz) is shown in Fig. 4. For the bandwidth range of $B < 10$ kHz, the system capacity is almost the same value for different numbers of subcarriers of 512, 1024, and 2048. In this case, the lowest number of subcarriers $N_c = 512$ should be chosen to reduce complexity of receiver. The reason can be explained by using Eq. (28). For the narrow bandwidth (i.e. less than 10 kHz for the considered case), the increase in the number of sub-carriers leads to the decrease in the SINR as shown in Fig. 3. However, that makes the OFDM symbol $T_S = N/B$ larger. As a result, the bandwidth efficiency $\beta = T_S/(T_S + T_G)$, in which $T_G = \tau_{max}$ is fixed, will be increased. Therefore, if $N_c$ increases, the bandwidth efficiency will increases, while the SINR decreases. From this argument along with Eq. (28), we also observed that the capacity remains unchanged for the larger numbers of subcarriers as shown in Fig. 4 with the bandwidth range $B < 10$ kHz. In other words, with the limited bandwidth of UWA channels, it may be impossible to increase the capacity by simply increasing the number of sub-carriers.

In a similar way, for the larger bandwidth of range from 10 kHz to 15 kHz, one should choose $N_c = 1024$ to ensure that the system capacity is not significantly reduced, but to limit the ICI effect. For a bandwidth of more than 20 kHz, $N_c = 2048$ is suitable.

Besides the system capacity $C$ (Kbps), the spectrum efficiency $C/B$ (b/s/Hz) is also an important factor in the OFDM system design due to the limited bandwidth of UWA channels. Figure 5 shows the results of the spectrum efficiency $C/B$ (b/s/Hz) versus the symbol duration $T_S$ for

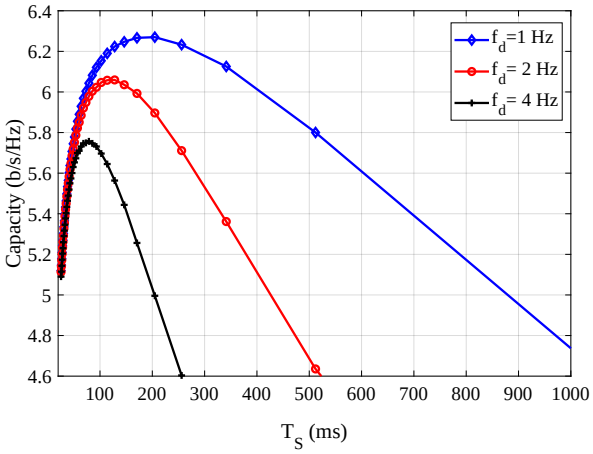


Figure 6. Spectral efficiency $C/B$ (b/s/Hz) versus symbol length $T_S$ for different Doppler frequencies.

different numbers of subcarriers for the case of $f_d = 1$ Hz. For the small symbol duration range of $T_S < 204.8$ms, a larger $T_S$ results in a higher spectral efficiency $C/B$. This is because a larger $T_S$ makes a higher bandwidth efficiency coefficient $\beta$. Consequently, the spectral efficiency $C/B$ will be increased as shown in Eq. (29). On the contrary, an increase in $T_S$ will get a decrease in the spectral efficiency $C/B$ for the larger symbol range $T_S > 204.8$ ms. The reason is that the larger $T_S$ (i.e. the smaller subcarrier spacing $\Delta_f$) makes the ICI effect more serious on the UWA channel. Hence, $T_S > 204.8$ ms also results in the decrease of both SINR and the spectral efficiency $C/B$ as shown in Eq. (29). As observed in Fig. 5, the spectral efficiency all achieves the maximal value $C/B_{\max} = 6.265$ (b/s/Hz) at $T_S = 204.8$ ms. Using this result, we can determine the optimal bandwidth for each different number of subcarriers $N_c$.

TABLE I
THE UWA-OFDM SYSTEM PARAMETERS FOR DIFFERENT DOPPLER FREQUENCIES.

| $f_d$ | $(C/B)_{\max}$ (b/s/Hz) | $T_S$(ms) | $B$ (kHz) | | |
|---|---|---|---|---|---|
| | | | N=512 | N= 1024 | N =2048 |
| 1 Hz | 6.265 | 204.8 | 2.5 | 5.0 | 10.0 |
| 2 Hz | 6.059 | 128.0 | 4.1 | 8.2 | 16.4 |
| 4 Hz | 5.757 | 78.8 | 6.5 | 13.0 | 26.0 |

Moreover, depending on the Doppler shift $f_d$, an appropriate $T_S$ should be chosen to achieve the maximum $C/B$. Figure 6 depicts the $C/B$ results versus $T_S$ for different Doppler shifts for the case of $N_c = 1024$. When the Doppler frequency increases, the optimal value of $T_S$ to achieve maximum $C/B$ is decreased. The optimal $T_S$ and $B$ values for different Doppler frequencies are shown in Table I. Based on these results, one can determined the parameters $T_S, B, N_c$, for UWA-OFDM systems to not only achieve maximum spectral efficiency and channel capacity but also limit the ICI effect and minimize complexity at the receiver under different transmission conditions.

## V. CONCLUSIONS

This paper has investigated the capacity of the UWA-OFDM system under the impact of ICI effect. Using the time variant geometry-based channel model for the shallow water environment, the SINR of each sub-carrier has been derived. The channel capacity is computed from the SINR, which takes the Doppler effect into account. The numerical results of SINR, channel capacity, and spectral efficiency give us guidance in UWA-OFDM system design, specifically in choosing the transmission parameters including the number of subcarriers, the symbol length, and the signal bandwidth for the different Doppler shifts.

## ACKNOWLEDGEMENT

This study was funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under the project number 102.04-2018.12.

## REFERENCES

[1] G. Marani, S. K. Choi, and J. Yuh, "Underwater Autonomous Manipulation for Intervention Missions AUVs," *Ocean Engineering*, vol. 36, no. 1, pp. 15–23, 2009.

[2] D. W. Kim, "Tracking of REMUS Autonomous Underwater Vehicles with Actuator Saturations," *Automatica*, vol. 58, no. C, pp. 15–21, 2015.

[3] N. T. Hoa, N. T. Hieu, N. Van Duc, G. Gelle, and H. Choo, "Second Order Suboptimal Power Allocation for OFDM-Based Cognitive Radio Systems," in *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, ICUIMC '13, (New York, NY, USA), Association for Computing Machinery, 2013.

[4] T. Ebihara and K. Mizutani, "Experimental Study of Doppler Effect for Underwater Acoustic Communication Using Orthogonal Signal Division Multiplexing," *Japanese Journal of Applied Physics*, vol. 51, p. 07GG04, jul 2012.

[5] J. Tao, "DFT-Precoded MIMO OFDM Underwater Acoustic Communications," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 3, pp. 805–819, 2018.

[6] L. Zhang, T. Kang, H. C. Song, W. S. Hodgkiss, and X. Xu, "MIMO-OFDM Acoustic Communication in Shallow Water," in *2013 OCEANS - San Diego*, pp. 1–4, 2013.

[7] H. Tran Minh, S. Rie, S. Taisuki, and T. Wada, "A Transceiver Architecture for Ultrasonic OFDM with Adaptive Doppler Compensation," in *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–6, 2015.

[8] S. Yoshizawa, H. Tanimoto, and T. Saito, "Experimental results of OFDM rake reception for shallow water acoustic communication," in *2016 Techno-Ocean (Techno-Ocean)*, pp. 185–188, 2016.

[9] H. Do Viet, T. Chien, and V. Nguyen, "Proposals of Multipath Time-Variant Channel and Additive Coloured Noise Modelling for Underwater Acoustic OFDM-Based Systems," *International Journal of Wireless and Mobile Computing*, vol. 11, pp. 329–338, Jan. 2016.

[10] G. Qiao, Z. Babar, L. Ma, L. Wan, X. Qing, X. Li, and M. Bilal, "Shallow water acoustic channel modeling and mimo-ofdm simulations," in *2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 709–715, 2018.

[11] D. Nguyen, H. Nguyen, and H. Ho, "Methods to Estimate the Channel Delay Profile and Doppler Spectrum of Shallow Underwater Acoustic Channels," *Archives of Acoustics*, vol. 44, pp. 375–383, Jan. 2019.

[12] X. Wang, J. Wang, L. He, and J. Song, "Doubly Selective Underwater Acoustic Channel Estimation with Basis Expansion Model," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2017.

[13] M. Naderi, D. V. Ha, V. D. Nguyen, and M. Patzold, "Modelling The Doppler Power Spectrum of Non-Stationary Underwater Acoustic Channels Based on Doppler Measurements," in *OCEANS 2017 - Aberdeen*, pp. 1–6, 2017.

[14] H. Nouri, M. Uysal, E. Panayirci, and H. Senol, "Information Theoretical Performance Limits of Single-carrier Underwater Acoustic Systems," *IET Commu-*

*nications*, vol. 8, no. 15, pp. 2599–2610, 2014.

[15] P. Bouvet and A. Loussert, "Capacity Analysis of Underwater Acoustic MIMO Communications," in *OCEANS'10 IEEE SYDNEY*, pp. 1–8, 2010.

[16] D. E. Lucani, M. Stojanovic, and M. Medard, "On the Relationship between Transmission Power and Capacity of an Underwater Acoustic Communication Channel," pp. 1–6, 2008.

[17] Y. M. Aval, S. K. Wilson, and M. Stojanovic, "On the Achievable Rate of A Class Of Acoustic Channels and Practical Power Allocation Strategies for OFDM Systems," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 4, pp. 785–795, 2015.

[18] A. Radosevic, J. G. Proakis, and M. Stojanovic, "Statistical Characterization and Capacity of Shallow Water Acoustic Channels," in *OCEANS 2009-EUROPE*, pp. 1–8, 2009.

[19] Ye Li and L. J. Cimini, "Bounds on The Interchannel Interference Of OFDM in Time-Varying Impairments," *IEEE Transactions on Communications*, vol. 49, no. 3, pp. 401–404, 2001.

[20] A. S. I. R. Capoglu, Y. Li, "Effect of Doppler spread in OFDM-based UWB systems," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2559–2567, 2005.

[21] P. J. Bouvet and Y. Auffret, "On the Achievable Rate of Multiple-Input–Multiple-Output Underwater Acoustic Communications," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 3, pp. 1126–1137, 2020.

[22] D. V. Ha, V. D. Nguyen, and Q. K. Nguyen, "Modeling of Doppler power spectrum for underwater acoustic channels," *Journal of Communications and Networks*, vol. 19, no. 3, pp. 270–281, 2017.

[23] A. G. Zajic, "Statistical Modeling of MIMO Mobile-to-Mobile Underwater Channels," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1337–1351, 2011.

[24] Y. Widiarti, Suwadi, Wirawan, and T. Suryani, "A Geometry-Based Underwater Acoustic Channel Model for Time Reversal Acoustic Communication," in *2018 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pp. 345–350, 2018.

[25] A. Withamana and H. Kondo, "Wideband Time-Varying Underwater Acoustic Channel Emulator," in *2017 IEEE Underwater Technology (UT)*, pp. 1–5, 2017.

[26] M. Naderi, M. Pätzold, and A. G. Zajic, "A Geometry-Based Channel Model for Shallow Underwater Acoustic Channels Under Rough Surface And Bottom Scattering Conditions," in *2014 IEEE Fifth International Conference on Communications and Electronics (ICCE)*, pp. 112–117, 2014.

**Do Viet Ha** received her B.S, M.Sc., and Ph.D. degrees in Electronics and Telecommunications Engineering from Hanoi University of Science and Technology (HUST), Hanoi, Vietnam, in 2001, 2007, and 2017, respectively. She is currently working with the Department of Electronics Engineering, University of Transport and Communications, Hanoi, Viet Nam, as a lecturer. Her main areas of research interest are mobile channel modeling, especially underwater acoustic channels, underwater acoustic OFDM systems, and mobile-to-mobile communications

**Nguyen Tien Hoa** graduated with a Dipl.-Ing. in Electronics and Communication Engineering from Hanover University. He has worked in the R&D department of image processing and in the development of SDR-based drivers in Bosch, Germany. He devoted three years ot experiment with MIMOon's R&D team to develop embedded signal processing and radio modules for LTE-A/4G. He worked as a senior expert at Viettel IC Design Center (VIC) and VinSmart for development of advanced solutions for aggregating and splitting/steering traffic at the PDCP layer to provide robust and QoS/QoE guaranteeing integration between heterogeneous link types, as well as Hybrid Beamforming for Millimetre-Wave 5G systems. Currently, he is a lecturer at the School of Electronics and Telecommunications, Hanoi University of Science and Technology. His research interests are resource allocation in B5G, and vehicular communication systems.

**Nguyen Van Duc** received his B.S and M.Sc degrees in Electronics and Telecommunications engineering from Hanoi University of Science and Technology (HUST), Hanoi, Vietnam, in 1995 and 1997, respectively. In 2003, he received Ph.D. in Electronics and Telecommunications Engineering from Leibniz University Hannover, Germany. Since 2006, he has been with the Department of Communications Engineering, HUST, where he is currently an Associate Professor. His current research interests include underwater communications, intelligent transport systems, and cognitive radio networks.