Performance Optimization and Modelling
of Complex Wireless Networks Using Surrogate Models

Michael Mehari

Promotoren: prof. dr. ir. E. De Poorter, prof. dr. ir. I. Moerman
Proefschrift ingediend tot het behalen van de graad van
Doctor in de ingenieurswetenschappen: elektrotechniek

UNIVERSITEIT
GENT

Vakgroep Informatietechnologie
Voorzitter: prof. dr. ir. B. Dhoedt
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2017 - 2018

ISBN 978-94-6355-125-0
NUR 950
Wettelijk depot: D/2018/10.500/43

Ghent University
Faculty of Engineering and Architecture
Department of Information Technology

Promotors: prof. dr. ir. Eli De Poorter
prof. dr. ir. Ingrid Moerman

Jury members: em. prof. dr. ir. Hendrik Van Landeghem, Ghent University (chair)
prof. dr. ir. Eli De Poorter, Ghent University (promotor)
prof. dr. ir. Ingrid Moerman, Ghent University (promotor)
prof. dr. mult. Dirk Deschrijver, Ghent University (secretary)
dr. ir. Toon De Pessemier, Ghent University
prof. dr. ir. Ann Nowé, Vrije Universiteit Brussel
prof. dr. ir. Violet R. Syrotiuk, Arizona State University

Ghent University
Faculty of Engineering and Architecture

Department of Information Technology
iGent Tower, Technologiepark 15, B-9052 Gent, België

Tel.: +32-9-331.49.00
Fax.: +32-9-331.48.99

Dissertation to obtain the degree of
Doctor of Electrical Engineering
Academic year 2017-2018

Acknowledgments

“ነገርን ሁሉ በጊዜዉ ዉብ አድርጎ ሠራዉ፤ እግዚአብሔርም ከጥንት ጀምሮ እስከ ፍጻሜ ድረስ የሰራዉን ስራ ሰዉ መርምሮ እንዳያገኝ ዘላለምነትን በልቡ ሰጠዉ ።”

-- መጽሐፈ መክብብ ፫፡፲፩

“He hath made every thing beautiful in his time: also he hath set the world in their heart, so that no man can find out the work that God maketh from the beginning to the end.”

– Ecclesiastes 3:11

Since I was a little kid, I had the luxury to be properly taught and guided by my father, Tetemke Mehari. In every step of the way, he was always there to guide me and for that, I have an everlasting love and respect. “Your guide has also passed on to the way I am raising my daughter and you should be proud of yourself.” My mother, Hanna Belete, is the most hardworking person I know. Her presence during my early morning studies, the delicious foods after school hours, and the reminding phone calls during evening hours are constant reminders of her love and dedication. “Your effort has paid off for this day and you should also be proud of yourself.” I also had the luxury to be raised alongside two of my brothers, Yisihak Tetemke and Fitsumbirhan Tetemke. I have seen all that is to be seen (the loves and the fights) and I wouldn’t have asked for more. “You are wonderful in every way possible and I am very much thankful that you are part of my life.” As a second mother, I have great respect and love for my grandmother, Tiruwork Gebremichael. I used to spend most of my summer vacation with her and after I finished my undergraduate studies, I moved to the city where she lives. I also have great respect for my uncle, Eshetu Belete, who not only raised me during my childhood, when my father was abroad, but also was an inspirational figure of a strong will and dedication.

My higher education career started in the state’s capital University, Addis Ababa University, Ethiopia. The knowledge and hardworking habits I had acquired with my family, is further cemented during my undergraduate study. Then after, I got a scholarship grant for studying a graduate program in KU Leuven (GroepT campus), Belgium. This was where I met my professor and thesis promotor, Prof. Dr. Luc Bienstman. He opened the path to my future career when he introduced me to Prof. Dr. Ingrid Moerman, my Ph.D. promotor. We have spent enjoyable and fruitful two years together, and I am very much thankful for all the things that he did for me.

Leuven is not only the place where I started my Ph.D. career, but it is also the place where I met my wife, Eden Tesfasselassie. Eden is a very strong woman, passionate about life and in particular for study. Together, we brought our beautiful daughter, Rebekka Michael. We are so lucky to have her in our lives and to see her grow every single day. I must say that if it wasn't for them, my life could have been boring and they gave meaning to it. "I am very lucky to have you both in my life and it is my wish to see you succeed in life."

In my six years of Ph.D. research, I came across a number of helpful people both internal and external to my research group. My sincere appreciation and respect go to my promotors Prof. Dr. Eli De Poorter and Prof. Dr. Ingrid Moerman. "You have helped me to see the best of myself and for that, I am greatly thankful." As an experiment-driven researcher, I have worked on the IMEC w-iLab.t testbed for almost 6 years and without the help of my colleagues from the w-iLab.t team (Bart Jooris, Vincent Sercu, and Pieter Becue), the research could have taken its toll on me. I also like to thank Dr. Adnan Shahid and Jan Bauwens for their advise and help in one of my research work. Collaboration was a core part of my Ph.D. and I would like to thank Prof. Dr. Dirk Deschrijver, Dr. Ivo Couckuyt and Tom Van Steenkiste from the surrogate modeling research group, Prof. Dr. David Plets, Dr. Günter Vermeeren and Prof. Dr. Wout Joseph from the WAVES research group and Prof. Dr. Violet R. Syrotiuk, Prof. Dr. Charles J. Colbourn, RANDY Compton and Stephen Seidel from the Arizona State University. Last but not least, I want to thank all people (Dr. Liu Wei, Dr. Adnan Shahid, Jen Rossey and Mathias Baert) who have helped me by correcting, translating and sharing ideas while I was finalizing this Ph.D. dissertation.

Besides the Ph.D. research, I have worked in multiple European and National projects, collaborated with a number of International and Belgian researchers and I want to thank all people that were part of this amazing experience. The first three years of my Ph.D., I was involved in the CREW project and collaborated with multiple European partners. Writing deliverable, participating in plenary and review meetings and preparing demonstrations were great experiences. As part of the CREW project, I would like to thank Dr. Stefan Bouckaert, Dr. Liu Wei, Dr. Daan Pareit, Jono Vanhie-Van Gerwen and Dries Naudts for their support and a great team effort we put together. There is a saying that goes, an engineer couldn't be a real engineer unless he/she is exposed to the industry, and it was until I was involved in the FORWARD project. Within the FORWARD project, I have gained industrial experience through numerous hurdles; i.e. challenging wireless environments, critical application requirements, secure industrial networks and the list goes on. And as part of this great experience, I would like to thank Prof. Dr. Jeroen Hoebeke for his great effort by collaborating and leading our team until the end of the project. I also worked in close contact with Dr. Liu Wei and Jetmir Haxhibeqiri, and I had a great experience. The last two projects I was involved were European projects (WISHFUL and eWINE), and they were continuation of the CREW project. For the most part, the research partners were the same as to the CREW project and it felt like working within the same research group. It was also at this moment that I shifted my research direction towards wireless sensor

networks and I would like to thank Jan Bauwens, Peter Ruckebusch and Dr. Spilios Giannoulis for helping me during this transition period. Currently, I have started working on an international project (Spectrum Collaboration Challenge) hosted by the US Defense Advanced Research Projects Agency (DARPA). This is the second year of the project and our team has already accomplished amazing results. I had great support from Dr. Xianjun Jiao, FelipeAugusto PereiradeFigueiredo, and Dr. Spilios Giannoulis when they first introduced me to the project. Besides academic and industrial projects, I also have tutored a number of practicum sessions in the department of information technology and I like to acknowledge Andy Van Maele for his help while preparing the sessions.

Office life was also a great experience, to work alongside Belgian and International researchers. I came to experience European and Asian cultures during lunch breaks, at sporting events, in IDLab days, among others. The cultural food gatherings were a unique experience. Thank you, Merima, Tarik, Floris, Cristian, Elnaz, Wei, Jen, Girum, Matteo, Bart M., Felipe, Mathias, Amina, Jaron, Abdulkadir, Nicola, Irfan, Vasileios, Martijn, Aslam, ... "You guys were amazing." The afternoon carpool with Enri and Jetmir was also memorable. As an Ethiopian, I feel running is part of me and I have enjoyed every moment of the different races I have participated. I will never forget the neck-and-neck race finish I had with Floris, though some have not finished. The running spirit is still alive in our group and thanks to Subho, Jaron, and Mathias for making it memorable. Our administrative support people are also one of a kind, and I like to thank the sysadmin team (Joeri Casteels, Simon Roberts, Vicent Javier Borja-Torres, Brecht Vermeulen, and Jonathan Moreel) and the secretary office (Martine Buysse, Davinia Stevens, and Bernadette Becue) for their daily support.

Finally, I would like to acknowledge all people that were involved in my Ph.D. research in one way or the other (Prof. Mogessie Ashenafi, Dr. Gebreselassie Baraki, Nardos Tesfaselassie, Tim Vanden Bempt, Daniel Abebe, Girma Abegaz, Etsegenet Belete, Firehiwot Kedir, Hewan Yilma, Leuven running group, ...).

Ghent, June 2018
Michael Tetemke Mehari

Table of Contents

List of Figures

List of Tables

List of Acronyms

A

ACO	Ant Colony Optimization
AGV	Automated Guided Vehicle
AIS	Artificial Immune System
AM	Amplitude Modulation
AMM	Adaptive Multiresolution Modulation
ANOVA	ANalysis Of VAriance
APF	Approximate Pareto Front

B

BESTCOM	BElgian network on STochastic modelling, analysis, design and optimization of COMmunication systems
BL	Bayesian Learning
BLE	Bluetooth Low Energy
BT-OMP	BackTracking Orthogonal Matching Pursuit

C

CCA	Clear Channel Assessment
CCTV	Closed-Circuit Television
CN	Cognitive Networks
CoS	Cost of Service

COT Channel Occupation Time
CREW Cognitive Radio Experimentation World
CSMA/CA Carrier Sense Multiple Access with Collision Avoidance
C-THLD Conditional THreshoLD

D

DE Differential Evolution
DSSS Direct Sequence Spread Spectrum
DVB-T Digital Video Broadcasting - Terrestrial

E

EA Evolutionary Algorithms
ED Epsilon Dominance
EC Experiment Controller
EGP Evidence Gathering Process
EGO Efficient Global Optimization
EI Expected Improvement
EI Exposure Index
EMO Efficient Multi-objective Optimization
ETSI European Telecommunications Standards Institute
eWINE elastic WIreless Networking Experimentation

F

FI Fixed Iteration
FM Frequency Modulation
FWO Fonds Wetenschappelijk Onderzoek
FSPL Free Space Path Loss

G

GA Genetic Algorithm
GMCO Global Maximum Combined Objective
GP Gaussian Process

H

HSS Hammersley Sequence Sampling
HV-PoI HyperVolume based Probability of Improvement

I

IAP Interuniversity Attraction Poles
IDW Inverse Distance Weighting
IEEE Institute of Electrical and Electronics Engineers
IETF Internet Engineering Task Force
IoT Internet of Things
ISM Industrial Science and Medical
ISO International Organization for Standardization
ITU International Telecommunication Union

K

KF Kalman Filtering
KNN K-Nearest Neighbor

L

LA Locating Array
LEXNET Low emf EXposure NETworks
LHD Latin Hypercube Designer
LHS Latin Hypercube Sampling
LR Linear Regression
LTE Long Term Evolution

M

MA Moving Average
MAC Medium Access Control
MAD Median Absolute Deviation
MANET Mobile Adhoc NETwork
Ma.HMM Manifold alignment with Hidden Markov Model
MDR Mutual Domination Rate
MOEA Multi-Objective Evolutionary Algorithms
MOPSO Multi-Objective Particle Swarm Optimizer
MOS Mean Opinion Score
MOSBO Multi-Objective Surrogate-Based Optimization

N

NN Neural Networks
NSGA Non-dominated Sorting Genetic Algorithm

O

OFDM Orthogonal Frequency Division Multiplexing

OFI Objective Function Improvement
OMF Orbit Management Framework
OMP Orthogonal Matching Pursuit
OPF Optimal Pareto Front

P

PCA Principal Component Analysis
PDF Probability Density Function
PDR Population Domination Rate
PESQ Perceptual Evaluation of Speech Quality
PG Performance Gap
PI Progress Indicator
PLC Programmable Logic Controller
PoI Probability of Improvement
POLQA Perceptual Objective Listening Quality Assessment
PSO Particle Swarm Optimization

Q

QoE Quality of Experience
QoS Quality of Service

R

RAT Radio Access Technology
RC Resource Controller
RL Reinforcement Learning
RSA Response Surface Approximation
RRSE Root Relative Square Error

RSSI Received Signal Strength Indicator
RTP Real-time Transport Protocol
RTCP RTP Control Protocol

S

SA Simulated Annealing
SAMURAI Software Architecture and Modules for Unified RAdIo control
SAR Specific Absorption Rate
SBO Surrogate Based Optimization
SD Stopping Decision
SF Spreading Factor
SI Swarm Intelligence
SL Supervised Learning
SPEA Strength Pareto Evolutionary Algorithm
SMS-EMOA S-Metric Selection Evolutionary MultiObjective Algorithm
SNR Signal-to-Noise Ratio
SSD SuperSaturated Design
STD STandard Deviation
STD-THLD STandard Deviation THreshoLD
STD-WIDTH STandard Deviation WIDTH
SUMO SUrrogate MOdeling
SuF Speed up Factor
SUT System Under Test
SVM Support Vector Machine
SVR Support Vector Regression

T

TAISC Time Annotated Instruction Set Computer
TIA Telecommunications Industry Association

U

UL	Unsupervised Learning
USRP	Universal Software Radio Platform

W

WARP	Wireless open Access Research Platform
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide interoperability for Microwave Access
WISHFUL	WIreless Software and Hardware platforms for Flexible and Unified radio and network controL
WLAN	Wireless Local Area Networks
WSN	Wireless Sensor Network

Samenvatting
– Summary in Dutch –

Draadloze systemen worden beschouwd als één van de grootste uitvindingen uit de hedendaagse tijd. Door de jaren heen is het aantal types en variteiten van dit soort netwerken steeds blijven groeien, alsook de complexiteit ervan. Huidige draadloze systemen trachten hun performantie te *optimaliseren*, met als doel de Quality of Service (QoS) te verhogen of de Cost of Service (CoS) te verlagen. Optimaliseren van de performantie is applicatie-afhankelijk en kan ofwel één doel (bv. verhogen van de kwaliteit van een draadloos audio systeem) of meerdere doelen (bv. verhogen van de levensduur van een draadloos sensornetwerk en terwijl een zo hoog mogelijk QoS blijven aanbieden) voor ogen hebben. Soms kan de optimale performantie niet bereikt worden zonder een diepgaander begrip van de werking van het draadloze systeem. In zón geval, wordt het draadloos systeem *gekarakteriseerd* en wordt er een representatief model voor gecreëerd. Later wordt dat model gebruikt in opeenvolgende optimaliseringsoperaties. Een ander probleem dat meer aandacht verkregen heeft in de afgelopen jaren, is het beoordelen en correct instellen van de parameters van draadloze netwerken. Door de toenemende complexiteit van recente draadloze netwerken, worden er ook steeds meer configureerbare parameters gebruikt in draadloze ontwerpen. Desalniettemin, state-of-the-art oplossingen kunnen dit soort complexiteiten niet aan, waardoor die eerst de parameters beoordelen. Dit met de intentie om de belangrijkste parameters eruit te selecteren. Daarna gaat het optimaliseren of modeleren verder.

Het ontwerpen en optimaliseren van een nieuwe generatie draadloze netwerken vormt een uitdaging, dit om diverse redenen, waaronder de samenwerking tussen heterogene technologien, het tijdrovende ontwerpproces, uitgebreide applicatienoden, big data analyse ... Het is onmogelijk om al deze uitdagingen aan te pakken in n doctoraat. *Dit proefschrift is gericht op het oplossen van het langdurige proces van het zoeken naar de optimale configuratie instellingen in complexe, draadloze netwerken*. Specifiek voor real-time applicaties is dit een probleem, omdat de oplossing binnen een beperkte tijdsperiode dient te worden geëvalueerd. In het verleden was het *grondig doorzoeken* van de oplossingsruimte een populaire methode, omdat het een accurate oplossing zal opleveren, binnen een redelijke tijdsperiode en voor een gelimiteerd aantal opties. Wanneer de complexiteit van het draadloze netwerk echter toeneemt, is deze methode niet mogelijk en wordt er overgegaan tot *heuristieken*. Heuristieken evalueren benaderende oplossingen van een draadloos probleem en komen tot een consensus door het ontdekken van hun

omgeving en erover te leren. In tegenstelling tot de grondig doorzoekende methode, garanderen heuristieken geen globale oplossing voor een probleem, maar bekomen ze een goede oplossing binnen een redelijke tijdsperiode. Hun belangrijkste voordeel is het gebruik van reeds geëvalueerde oplossingen in het zoeken naar betere oplossingen. Dit laat toe om het langdurige optimalisatie proces te verkorten. De complexiteit van de draadloze netwerken van vandaag heeft ook het potentieel van heuristieken overtroffen, waardoor een betere aanpak zich opdringt. Een alternatief is het gebruik van surrogaat modellen in het optimalisatie proces.

Een *surrogaat model* (SUMO) is een efficiënte wiskundige representatie van een *black-box* systeem (bijvoorbeeld een complex draadloos netwerk). Een black-box aanpak behandelt alle netwerkparameters gelijkaardig waardoor de hindernis van de verschillende protocollagen van draadloze systemen wegvalt. Op deze manier wordt SUMO niet alleen gebruikt als globale verbeteraar, maar ook om optimalisatieproblemen, verspreid over meerdere lagen, aan te pakken. Intern past SUMO verschillende modeleringsmethodes toe (Kriging, Gaussiaanse processen, neurale netwerken, support vector regressie) die gebruikt worden in verschillende applicaties. Dit proefschrift past het *Kriging model* toe omdat het reeds effectief is bevonden in meerdere engineering toepassingen (bv. aerodynamische en elektromagnetische problemen). Een Kriging model presteert het beste wanneer performantiemetrieken graduale reacties aantonen op de configuratieruimte. Bijvoorbeeld, het plotten van de levensduur van een draadloos sensornetwerk (WSN) in functie van zendvermogen zal een graduale relatie aantonen. Doordat draadloze netwerken dit soort gedrag ook ervaren, zal SUMO daarbij naar verwachting zeer goed presteren. Wanneer dit niet het geval is, zoals bij categorische parameters (bv. zendcapaciteit in functie van MAC protocollen), is het gebruik van verschillende surrogaat modellen (bv. neurale netwerken) nodig.

In dit proefschrift wordt SUMO uitgebreid gebruikt om het langdurige proces van het zoeken naar optimale configuratie instellingen op te lossen. Als een representatieve use case, namen we een draadloos audioconferentiesysteem met als de doel om optimale configuratie instellingen te vinden, die een verhoogde *audio kwaliteit* en een verlaagd *elektromagnetische blootstelling* aan het draadloze systeem aanbieden. Er worden vier parameters geconfigureerd op het draadloze systeem en twee conflicterende doelen worden pareto geoptimaliseerd. Voor dit probleem, vindt SUMO een pareto optimale oplossing 70x sneller dan een grondige doorzoekende methode (6528 configuratiepunten) en bereikt het een oplossingsnauwkeurigheid van 97 %. Het onderzoekswerk wordt nog verder verbeterd door over te gaan naar een dynamische, draadloze omgeving, wat het onderzoek ook dichter bij de realiteit brengt. Het genomen scenario bestaat uit een single-hop WSN, dat beïnvloed wordt door een dynamische omgeving. De innovatie van dit onderzoek is het gebruik van een *cloud repository* om er meerdere performantiemodellen (SUMO modellen) van het WSN in op te slaan, waarbij elk model een statische instantie van de dynamische omgeving weergeeft. Door het gebruik van een cloud repository, wordt het WSN geoptimaliseerd via een representatief model van de huidige omgeving. Als er een representatief model wordt gevonden in de cloud repository, wordt dat model geselecteerd om de performantie te

optimaliseren. Indien niet, dan worden verschillende modellen vanuit de cloud repository samengevoegd om de unieke, huidige omgeving weer te geven en de performantie via dat samengevoegd model te optimaliseren. Op deze manier verliest het WSN geen tijd met het modelleren van de performantie, elke keer dat de omgeving wijzigt, maar kan de performantie rechtstreeks geoptimaliseerd worden via opgeslagen modellen. Specifiek voor het genomen single-hop WSN, zijn er slechts 10 experimenten nodig om de omgeving te karakteriseren en een representatief model te selecteren of samen te voegen. Dit in tegenstelling tot state-of-the-art SUMO optimalisatie, waarbij 125 experimenten nodig zijn, of grondig doorzoekende optimalisaties, waarbij 4800 experimenten nodig zijn. Tenslotte wordt parameter screening onderzocht, om het probleem van het groot aantal configureerbare parameters in complexe draadloze systemen, aan te pakken. Daartoe wordt een combinatorische ontwerpmethode genaamd *locating array* (LA) toegepast om een screening experiment te ontwerpen. Daarna worden alle testen uitgevoerd en wordt het resultaat geanalyseerd via een *backtracking orthogonal matching pursuit* (BT-OMP) methode. De finale output van het screening experiment (gevoelig parameters) wordt dan gebruikt in opeenvolgende SUMO optimalisaties en modeleringsprocessen. Ter illustratie wordt een draadloos audio conferentiesysteem genomen, dat gebruik maakt van 24 parameters. In totaal heeft het systeem $\approx 10^{13}$ configuratie instellingen, maar door het ontwerpen van 109 LA tests, is het mogelijk om de meest gevoelige parameters te screenen.

Het uitgevoerde onderzoek in dit proefschrift, is hoofdzakelijk gedreven vanuit experimenten, waarbij een draadloos testbed wordt gebruikt om complexe draadloze netwerken te evalueren. Deze aanpak heeft als voordeel dat het realistische interacties weergeeft tussen hard- en software, op meerdere protocollagen. Alle experimenten zijn uitgevoerd in het *IMEC w-iLab.t* draadloos testbed, wat *pseudo beschermd* is tegen interferentie van buitenaf. Interferentie is wel mogelijk binnenin het draadloos testbed (bv. door parallel lopende experimenten) en een monitoringsoplossing voor experimenten wordt toegepast als gevolg. De andere uitdaging in vanuit experimenten gedreven onderzoek, is *experiment orkestratie*. Om een experiment uit te voeren, geeft een *Experiment Controller* (EC) een serie van commandos aan een aantal *Resource Controllers* (RCs). Wanneer meerdere nodes worden gebruikt in het experiment, duurt het langer om elk commando vanuit de EC naar de RCs te orkestreren. Dit is ook problematisch voor tijdsgebonden applicaties en experimenten. Er worden twee experiment orchestratie tools toegepast in dit proefschrift: (i) *Orbit Management Framework* (OMF) en (ii) *Bash Experiment Orchestration Framework* (BEOF). Het tweede framework is een in-house oplossing en wordt gebruikt in de meerderheid van de gevallen.

Samengevat, is het onderzoeksavontuur van het gebruik van SUMO om het langdurige proces van het optimaliseren van complexe, draadloze netwerken, uitgedraait op een succesverhaal. Als resultaat, werden er meerdere internationale journals en conference papers geproduceerd. Maar het beste van alles is, dat we een tastbare aanwinst hebben bijgedragen aan de draadloze onderzoeksgemeenschap en we hopen dat dit werk ruimte zal vrijmaken voor verder onderzoek en opvolgende resultaten in de nabije toekomst.

Summary

Wireless Networks are among the greatest inventions of human beings. Over the years, the types and varieties of wireless networks have grown tremendously, and the complexity is growing more than ever. Looking into currently existing wireless networks, most of them *optimize* system performance with the aim of improving the Quality of Service (QoS) or reducing the Cost of Service (CoS). Performance optimization is application dependent and it can either be a single objective (e.g. increasing the quality of a wireless audio system) or multiple objectives (e.g. improving the lifetime of a wireless sensor network while providing the highest QoS). Sometimes, optimum performance cannot be achieved without a deeper understanding of the wireless system. In this case, the wireless system is *characterized* and a representative model is created. Later, the representative model is used in consecutive optimization operations. *Screening* the parameters of complex wireless networks is another problem that has gained attention in recent years. Due to the growing complexity of recent wireless networks, a large number of configuration parameters are used in wireless designs. However, state-of-the-art solutions cannot handle such complexities and screening is the first operation carried out, with the intention of selecting the most important parameters. Afterward, optimization or modeling operation proceeds.

Designing and optimizing next-generation wireless networks is challenging due to several reasons. These include coexistence of heterogeneous technologies, time-consuming design process, extended application requirements, big data analysis, and others. While it is impossible to tackle all challenges in one Ph.D., *this dissertation is focused on solving the time-consuming process of configuration parameter optimization in complex wireless networks*. Specific to real-time applications, this becomes problematic because the solution must be evaluated within a limited time period. In the past, *exhaustively searching* the solution space was a popular method because it will provide an accurate solution from a limited set of options and within a reasonable time frame. When the complexity of wireless networks starts to grow, exhaustive searching methods were not possible anymore and *heuristic approaches* become popular. Heuristic approaches evaluate approximate solutions of a wireless problem and they reach a consensus by learning and discovering their environment. Unlike the exhaustive searching method, heuristic approaches don't guarantee a global solution but they arrive at a good solution in a short period of time. Their main advantage is in the use of previously evaluated solutions for searching better solutions, and this allows them to cut down the lengthy optimization process. Once again, the complexity of today's wireless net-

works have surpassed the potential of heuristic algorithms and better approaches are needed. One alternative is to use surrogate models in the optimization process.

A *surrogate model* (SUMO) is an efficient mathematical representation of a *black-box* system (e.g. complex wireless network). A black-box approach treats all network parameters similarly and the *protocol layering* principle of wireless networks is removed as a result. This way, SUMO is not only used as a global optimizer, but it can also address optimization problems spanning multiple protocol layers. Internally, SUMO applies different modeling methods (i.e. Kriging, Gaussian process, neural networks, support vector regression) that are used in different applications. This dissertation applies *Kriging model* because it is found effective in multiple engineering use cases (e.g. aerodynamic and electromagnetic problems). A Kriging model performs best when performance metrics show gradual responses on the configuration space. For example, plotting Wireless Sensor Network (WSN) lifetime as a function of transmission power will exhibit a smooth relationship. Therefore, SUMO is expected to perform very well in most wireless network problems since they experience such a behavior. When this is not the case, such as in categorical parameters (e.g. throughput as a function of MAC protocol), different types of surrogate models (e.g. neural networks) are used.

In this dissertation, SUMO is used extensively to solve the time-consuming process of searching optimum configuration settings. As a representative use case, a wireless audio conferencing system is considered, and the goal is to find optimum configuration settings that will bring improved *audio quality* and reduced *electromagnetic exposure* to the wireless system. Four parameters are configured on the wireless system and two conflicting objectives are Pareto optimized. For this problem, SUMO finds a Pareto optimum solution 70x faster than an exhaustive searching method (6528 configuration points) and achieves a solution accuracy of 97%. Next, the research work is further improved by considering a dynamic wireless environment such that a wireless network drops performance whenever the environment changes and the system is re-calibrated to regain the optimum performance. To this end, a single-hop WSN under the influence of a slowly changing dynamic environment is considered. The innovation of this research is in the use of a *cloud repository* for storing multiple performance models (SUMO models) of the WSN, each representing a static instance of the dynamic environment. By using a cloud repository, the WSN optimizes its performance using a representative model of the current environment. If a representative model is found in the cloud repository, a model is selected to optimize the performance. However if not, different models from the cloud repository are merged together to represent the unique environment. This way, the WSN does not waste time to model the performance every time the environment changes and directly applies stored models to optimize the performance. Specific to the single-hop WSN set up, only 10 experiments are needed to characterize the environment and select/merge a representative model, as opposed to 125 experiments for state-of-the-art SUMO optimization or 4800 experiments for an exhaustive search based optimization. Finally, parameter screening is researched to tackle the problem of large configuration parameters in complex wireless networks. To this end, a combinatorial design method called

locating array (LA) is used to design a screening experiment. Afterward, all tests are executed and the result is analyzed using a *backtracking orthogonal matching pursuit* (BT-OMP) method. The most sensitive parameters from the screening experiment are then used in consecutive SUMO optimization and modeling operations. As a showcase, a wireless audio conferencing system is set up by using 24 parameters collected from multiple protocol layers. In total, the system has $\approx 10^{13}$ configuration settings, but by only designing 109 LA tests, the most sensitive 7 parameters are screened.

The research performed in this dissertation is mainly experiment-driven, whereby a wireless testbed is used to evaluate complex wireless networks. This approach has the advantage of giving realistic interactions between hardware and software at multiple protocol layers. To this end, all experiments are conducted in the *IMEC w-iLab.t* wireless testbed which is *pseudo shielded* from outside interference. However, interference is possible within the wireless testbed (e.g. parallel running experiments) and an experiment monitoring solution is applied as a result. The other challenge in experiment-driven research is *orchestration*. In order to execute an experiment, an *Experiment Controller* (EC) exchanges a series of commands to a number of *Resource Controllers* (RCs). When a large number of nodes are used in the experiment, it takes more time to orchestrate all commands. This is also problematic for time-bounded applications and experiments. In summary, two experiment orchestration tools are used in this dissertation, (i) *Orbit Management Framework* (OMF) and (ii) *Bash Experiment Orchestration Framework* (BEOF). The latter is an in-house built solution and is used in the majority of use cases.

In summary, the research adventures of SUMO to reduce the time-consuming process of complex wireless network optimization is a success story. Multiple international journals and conference papers are produced as a result. Best of all, we have contributed a tangible asset to the wireless research community, and we hope it will open room for further research and result in outputs in the near future.

1 Introduction

"It turns out that if you optimize the performance of a car and of an airplane, they are very far away in terms of mechanical features. So you can make a flying car. But they are not very good planes, and they are not very good cars."

– Gregory Benford (1941 -)

The first wireless network was established in 1880 when Alexander Graham Bell and Charles Sumner Tainter made a successful wireless telephone conversation. Since then, the number of wireless networks has grown quite rapidly. To name a few, cellular network (2G, 3G, 4G), public broadcast (AM, FM, DVB-T), satellite communication, home automation (Wi-Fi, baby phones, thermostat), environment monitoring (fire and habitat protection), surveillance (CCTV) and industrial automation (wireless PLC, AGV) are typical wireless networks used in our daily lives.

This large number of wireless networks have one thing in common: they fulfill specific needs of the society. However, societal needs have never stopped growing and as a result, wireless networks become more and more complex.

1.1 Complex Wireless Networks

A wireless network can be complex for a number of reasons. It can be due to a large number of nodes available in the network, a huge configuration space from which input parameters are drawn, the coexistence of heterogeneous access technologies,

a wide range of capabilities supported by networks, meeting stringent application requirements in harsh environments, among others.

– *Internet of Things* (IoT), for example, is a good showcase where large a number of nodes are interconnected to exchange data. Even though the technologies used in most IoT devices are not groundbreaking, the idea of **inter-connecting each and every device together makes the wireless network massively complex**. Based on the analysis from Cisco, there will be around 50 billion IoT connected devices by the end of 2020 [1]. The challenge with many interconnected devices is the generation of huge amount of data and the need for a large amount of storage space. Next, these mass of datasets are analyzed for a meaningful semantics such that *big data analysis* becomes a potential candidate by using *machine learning* tools and *parallel computing* architectures.

– By the same token, the 5^{th} generation (5G) of wireless communication systems is another showcase for complex wireless networks. The demand for a higher quality of service has led cellular networks to progress through multiple generations (2G, 3G, 4G), and the recently introduced 5G wireless network is planned to support a wide range of capabilities (i.e. huge data volumes, large number of connected devices, low latency applications, higher data rates and enhanced lifespan of low power devices) [2]. In the past, objective requirements were not strict and wireless networks only consider a small set of objectives. However, **when more and more objectives are involved, they all need to be met simultaneously and this requires huge computation time**. Furthermore, it is a major issue in *real time wireless applications* operating under hostile environments. Wireless networks operating in a hostile environment, experience rapid changes in the system performance and it becomes challenging to meet all the objectives in real time.

– **Looking into the configuration space of wireless networks, the larger it gets in size (more design points), the more difficult it will become to find optimum configuration settings**, thus making the wireless network more complex. In [3], 75 design parameters of a mobile wireless network are optimized for a maximum TCP throughput performance. In total, the system has a configuration space of more than 10^{43} design points and performing optimization is not feasible at this level of complexity prior to a screening operation. *Heuristic algorithms* are normally applied to solve optimization problems in wireless networks, especially when the objective function cannot be expressed in closed-form expression. However, they inherently require a large number of iterations before evaluating the optimum solution and this makes them not suitable for complex wireless networks. *Machine learning tools* are other methods that are often used in performance optimization and system modeling problems. They

too have the limitation of using a large amount of training data for creating regression, classification and learning models.

– *Heterogeneous wireless networks* are also becoming a source of complexity in recent years. Future wireless networks are becoming heterogeneous in order to support a multitude of services and to better interact with their environment. When wireless networks were designed in the past, it was assumed that they operate on the assigned spectrum alone. This was reasonable because part of the spectrum was licensed (i.e. for telecommunication and satellite services), but when the same idea is brought to unlicensed bands (i.e. Industrial, Scientific and Medical (ISM)), **multiple technologies start to share the same spectrum and coexistence becomes an issue**. At 2.4GHz ISM band, for example, Wi-Fi, Zigbee, Bluetooth, Bluetooth Low Energy (BLE) and LTE-Unlicensed are typical technologies that experience the coexistence problem. The main problem of coexistence is interference and it is due to the collocation of heterogeneous wireless technologies (e.g. Wi-Fi and Zigbee networks operating in a warehouse). *Cognitive Networks* (CN) are designed to solve this issue such that a wireless network under consideration constantly monitors the spectrum and whenever an enduring traffic is detected, the wireless network defers to other free channels. This approach only works for loosely populated networks and it becomes problematic when more networks are added to the area because all of them end up switching channels rather than doing useful work. One way to solve the coexistence problem is through collaboration among different networks, which is a hot topic in the *Spectrum Collaboration Challenge* (SC2) hosted by the Defense Advanced Research Projects Agency (DARPA), US. Another approach is to support multiple *Radio Access Technologies* (multi-RAT) within the network, which was a core objective in the *Flex5Gware* Horizon 2020 European project [4].

– The introduction of wireless networks in industrial premises was a convenient breakthrough to reduce the clutter of cables and to scale industrial services conveniently. However, **stringent requirements of industrial applications (i.e. low latency and reliability) combined with harsh environments are difficult to be met with wireless networks**. As a result, wireless networks become complex in such scenarios. One example is the Sprouts WSN platform [5], which supports a number of features specifically designed for harsh industrial environments such as modularity, energy harvesting, remote access, ruggedness and fault tolerance.

– **Wireless networks can be complex not only during operation but also in the design process**. In the past, network simulators have made a lot of contributions for designing next-generation wireless networks. New emerging standards and

technologies are usually impossible to test on a real system and network simulators are the first tools researchers will lay their hands on. However, with the recent trend in *experiment driven research*, designing wireless networks becomes challenging because i) physical hardware is usually not available, ii) experimentation effort increases rapidly with the size of the network, and iii) constructing large-scale wireless testing facilities (testbeds) is expensive.

1.2 Coping with Network Complexity

In the previous section, we have seen a number of reasons for a wireless network to become complex. In this section, we will deal with the different methodologies and state-of-the-art solutions that are applied to cope with the network complexity. Specifically, complexity due to huge configuration space, large performance objectives, and harsh wireless environments are addressed.

1.2.1 Methodologies

When considering possible solutions to wireless network problems, different methodologies are applied. These are (i) performance optimization, (ii) system characterization and (iii) parameter screening. Figure 1.1 shows an overview.

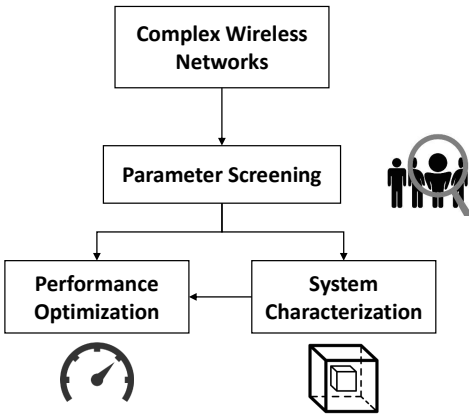


Figure 1.1: An overview of the different methodologies applied to cope with wireless network complexity.

1.2.1.1 Performance optimization

Performance optimization refers to a systematic approach of finding an optimum solution to a wireless network problem, in the shortest time possible. Performance

optimization is application dependent and it can either be single objective (e.g. increasing the quality of a wireless audio system) or multiple objectives (e.g. improving the lifetime of a wireless sensor network while providing the highest quality of service). Mobile Ad-hoc NETwork (MANET) is a typical example optimized in wireless networks. MANET is a continuous self-organizing and infrastructureless network of wirelessly connected mobile devices. Because of inherent mobility, MANET is usually optimized to prolong the network lifespan while keeping the packet latency within a limit. This could, for example, be used in military applications where real-time traffic and network lifespan are the two most important objectives. Figure 1.2 shows a hypothetical example of a MANET problem.

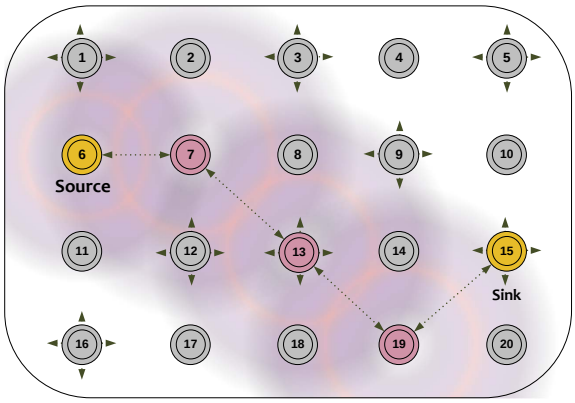


Figure 1.2: MANET composed of a source, a sink and relaying nodes. The goal is to select relaying nodes between the source and sink nodes and their transmission power levels, that will prolong network lifespan and reduce communication latency.

One way to solve the MANET multi-objective problem is by using Genetic Algorithms (GAs) [6]. GAs mimic the behavior of natural selection and initialize a population of candidate solutions that satisfy the objective function. Later on, subsequent generations are evolved through selection, crossover and mutation operations and finally converge to a global optimum solution. In the MANET problem, there are multiple paths for a packet to traverse from the source node to the sink node. Moreover, every node individually updates its transmission power and this exponentially increases possible paths and power level combinations. An initial population of paths and power levels is randomly selected and their performance is evaluated on the objective function. The selection procedure chooses paths and power levels resulting in the least energy consumption and within a given latency bound. Mutation and crossover operations also add unique solutions to the generation in order to explore and exploit the design space respectively. After a number of generations, the GA provides the global optimum path that has the least energy consumption and within a given latency bound.

1.2.1.2 System Characterization

System characterization, on the other hand, deals with the global accuracy of a wireless system, whereby it builds a predictive performance model overall points in the configuration space. For this reason, system characterization is used in design applications where the system is modeled once and used in consecutive design operations. By doing so, it saves experimentation/simulation time which otherwise has to be carried out on the real system over and over again. Considering solutions of a wireless system as inputs to a *black-box* fitness function, a point [solution, fitness] is drawn on the *black-box* function every time a solution is evaluated by an optimizer. Therefore, system characterization is about selecting and evaluating as few sample points as possible, to create a representative and accurate model of the *black-box* function.

Microstrip patch antenna, shown in Figure 1.3, is a typical example where system modeling is applied [7]. Since a high fidelity simulation of the antenna gain is an expensive operation, it takes a considerable amount of time to evaluate the optimum design parameters (patch and substrate dimensions, substrate permittivity and loss tangent, ground conductivity). Instead, an accurate system model of the microstrip antenna is characterized and is used in consecutive optimization operations, such as when multiple microstrip antennas are needed with unique propagation characteristics.

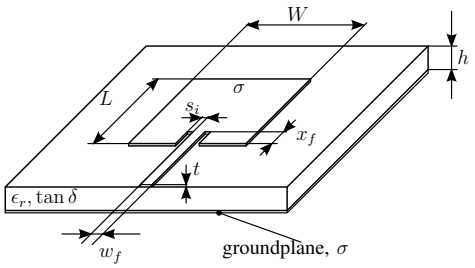


Figure 1.3: Design parameters of a microstrip patch antenna.

1.2.1.3 Parameter Screening

Complex wireless networks involve a large number of configuration parameters and this makes it difficult to optimize the performance or even model the underlying system. For this reason, screening is the first operation that is usually carried out to select the most important configuration parameters. When screening configuration parameters, main effects (influences from individual parameters) are usually considered such as in supersaturated designs (SSDs) [8]. However, complex wireless networks experience parameter interactions at different protocol lay-

ers [9], and it is possible for a parameter to have a minor influence individually but a major influence when interacting with other parameters. Furthermore, interactions can occur at higher orders between multiple parameters. However in this dissertation, the *sparsity-of-effects* principle [10] is assumed, where a system is dominated by main effects and 2^{nd} order interactions.

1.2.2 State-of-the-art Solutions

The different methodologies presented in Section 1.2 are implemented using multiple state-of-the-art solutions. An *exhaustive search* design [11], one of the simplest approaches, investigates all possible input settings and selects the optimum configuration. Except for a guaranteed performance, it requires the most amount of time and this makes it infeasible for the complex wireless networks that we have today. On the other hand, (i) nature-inspired algorithms and (ii) machine learning tools are commonly applied state-of-the-art solutions.

1.2.2.1 Nature-Inspired Algorithms

Nature-inspired algorithms are class of heuristic algorithms, that find approximate solutions of a problem through learning and discovering the environment. They are attractive for solving complex wireless problems because i) they have fast convergence speeds and ii) their proposed solutions are acceptable in most cases. Nature-inspired optimizers for complex wireless networks can be broadly classified into evolutionary algorithms, swarm intelligence algorithms and simulated annealing algorithms.

– *Evolutionary Algorithms* (EA) mimic the process of natural selection whereby the population evolves by crossing over well-fitted traits and by mutating random traits to add new variations. Examples of EA are Genetic Algorithms (GA) [6], Differential Evolution (DE) [12] and Artificial Immune System (AIS) [13].

– *Swarm Intelligence* (SI) algorithms, on the other hand, exploit the collective behavior of self-organized and decentralized natural systems such as ant colonies, bird flocks, and fish schools. Examples of SI include Ant Colony Optimization (ACO) [14] and Particle Swarm Optimization (PSO) [15].

– *Simulated Annealing* (SA) is inspired by metallurgy and freezing liquids when forming crystalline structures, such that with sufficient time the structure acquires a minimum energy state [16].

The general principle of nature-inspired algorithms, shown in Figure 1.4, starts by initializing a population of candidate solutions to the problem at hand. This is done by randomly selecting candidates from the entire population space. Next, the

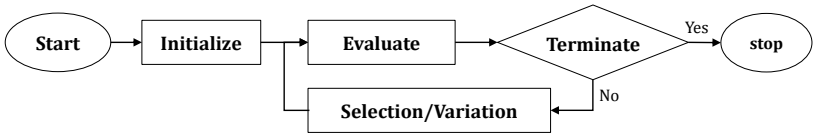


Figure 1.4: Operational flow-chart of nature-inspired algorithms.

fitness of candidate solutions is calculated. Afterward, the criteria for a stopping decision is evaluated to end the iterative process. If more iterations are needed, a new generation of candidates is selected by crossing-over already existing solutions or by using new candidate solutions. This way nature-inspired algorithms explore and exploit the solution space, to locate the global optimum solution.

1.2.2.2 Machine Learning Tools

A different approach to solving wireless network problems is by using *machine learning* tools. Machine learning is a powerful technique which allows computers to learn their environment without being explicitly programmed, thus making it applicable for a wide range of engineering problems, including wireless networks. Machine learning tools for complex wireless networks can be broadly classified as i) supervised learning, ii) unsupervised learning and iii) reinforcement learning.

– *Supervised learning* is a learning process under the guidance of a supervisor. This is usually applicable in classification problems where a supervisor collects a training data categorized into classes and a new performance data is classified into one of the categories. On the other hand, supervised learning can also be used in regression problems for modeling and predicting system performance. Typical algorithms used in supervised learning are K-Nearest Neighbor (KNN) [17], Bayesian Learning (BL) [18], Convolutional Neural Networks (CNN) [19], and Support Vector Machine (SVM) [20].

– *Unsupervised learning*, unlike supervised learning, is neither guided by a supervisor nor does it use labeled dataset to learn the behavior of the system. The goal is to look for hidden patterns from the unlabeled dataset and analyze the patterns. In the scope of wireless networks, unsupervised learning is mainly used for node clustering and data aggregation such as K-Means clustering [21] and Principal Component Analysis (PCA) [22].

– *Reinforcement learning* is a hybrid of supervised and unsupervised learning, such that it interacts with the surrounding by taking actions and learning from the rewards of the actions taken. Typical use cases of reinforcement learning are routing optimization, coverage and capacity optimization of wireless networks. In terms of implementation, Q-Learning is the most widely used algorithm [23].

Table 1.1: List of state-of-the-art solutions and applied methods in complex wireless networks.

	Performance Optimization	**System Characterization**	**Parameter Screening**
Nature-Inspired Algorithms	[6], [12], [13], [14], [15], [16]		
Machine Learning	[17], [18], [20], [21], [22], [23]	[18], [19], [20], [23]	
Generic Tools	[24]	[25]	[3]

1.2.2.3 Generic Tools

Nature-inspired algorithms and machine learning tools have historical backgrounds, and therefore they have well-defined methodologies. However, there are other methods which cannot be grouped into any specific category and in this dissertation, these methods are referred as "Generic Tools". Some examples include methods for creating statistical models, Markov models, analytical models, screening models, etc.

Table 1.1 shows a list of state-of-the-art solutions that are applied to different wireless network methodologies. Performance optimization is the most widely applied method followed by system characterization and parameter screening. Looking from a different perspective, nature-inspired algorithms are almost always applied in performance optimization problems, whereas machine learning tools are used in both characterization and performance optimization problems. In fact, most characterization problems apply some form of performance optimization. In the case of generic tools, solutions are evenly spread over all problem types.

1.3 Alternative Approach

The previous section has discussed different methodologies and state-of-the-art solutions that are used to cope with network complexity. In recent years, *surrogate model* (SUMO) assisted optimization and system characterization are gaining momentum due to the good performance trade-off they possess between accuracy and computational complexity [26]. For this reason, SUMO is often applied in complex engineered systems, such as in aerodynamic structure designs, electromagnetic and antenna designs, groundwater exploitation, and chemical processing. On the contrary, the use of SUMO in wireless networking is limited and is mainly used to augment the well-established state-of-the-art solutions (i.e. nature-inspired algorithms and machine learning tools) [27, 28]. This dissertation, however, takes a different approach, whereby SUMO is researched as a potential candidate for solving wireless network problems.

SUMO is an efficient mathematical representation of a *black box* system, which in this case is the complex wireless network. By considering wireless networks as black-box systems, the *protocol layering* principle is removed and optimization algorithms operate on the entire protocol stack to bring system-wide solutions.

SUMO has multiple variants (i.e. Kriging, Gaussian process, support vector regression, neural networks, among others) that are used in different applications. In this research work, the *Kriging surrogate model* is used because it is found effective in a number of engineering problems [26]. Kriging was originally used in Geostatistics to estimate and predict spatial resources according to a given distribution. From a given sampled resource distribution, Kriging predicts any spatial resource by computing a weighted average of the known values, in the neighborhood of an unknown point. Mathematically, it is similar to *Inverse Distance Weighting* (IDW) [29], except it considers the spatial auto-correlation of data points. Therefore, Kriging provides some measure of certainty or accuracy of the predictions.

Coming back to wireless network methodologies (discussed in Section 1.2.1), SUMO was the cornerstone in solving performance optimization and system characterization problems. For parameter screening problems, however, SUMO is applied at a later stage to optimize or characterize the system performance after the complexity of the network is reduced. Screening the parameters of wireless networks was not a common practice in the past, because a large number of configuration parameters were rarely considered in the design process. However, as wireless networks are becoming more and more complex, multiple parameters are getting equally important eventually requiring a large number of configuration parameters in the design process.

1.4 Proposed Solution

In the previous section, a number of challenges have been described for next-generation wireless networks and this section proposes a solution by following two approaches. The first solution applies a *sequential* design method and the second solution applies a *one-shot* design method. Both approaches are further explained in the coming sections with the help of Figure 1.5.

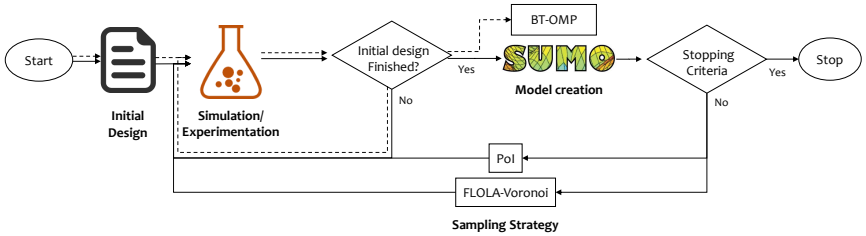


Figure 1.5: Proposed solution using sequential design approach (solid path) and one-shot design approach (broken path).

1.4.1 Sequential Design

A *sequential design* approach updates the system model recursively following a sampling strategy. It starts with an initial design set and creates the first model after executing all tests. The sampling strategy then produces a new design point which is later evaluated to update the previous model. In turn, the sampling strategy uses the updated model to produce a second design point and the iteration continues until a stopping criterion is met. Looking into the specifics, a sequential design approach is applied to solve performance optimization and system characterization problems in complex wireless networks. The major steps involved in a sequential design are i) initial design, ii) model creation, iii) stopping criteria, and iv) sampling strategy.

1.4.1.1 Initial Design

The initial design step is responsible for creating the first model that optimization and system characterization steps will rely on. A better initial model is crucial because it will save valuable time during the sequential phase. To this end, designing a large number of initial points could be beneficial. However, it also increases the experimentation time which is against the goal. Therefore, there is a trade-off in the size of the initial design points and this research applies the *latin hypercube design* (LHD) [31] method. LHD is a stratified sampling method that selects sample points evenly along the configuration space while ensuring proportional representation of design variables.

1.4.1.2 Model Creation

After the initial samples are designed and measurements are taken, the first Kriging model is created to be used in optimization or system characterization problems. It was also described in Section 1.3 that SUMO predicts unknown data points by weighting and correlating known data points. A graphical illustration of the Kriging model process is shown in Figure 1.6.

Starting from data points and performances (circles), a Kriging surrogate model (dashed line) is constructed. Kriging assumes that data points are spatially correlated, such that points that are closer will have comparable performances and points that are farther away will have distinct performances. Another way of saying this is, the performance difference between two data points is smaller if they are closer to each other than far away points. On the other hand, two pairs of data points which are equidistant from each other will not have identical *performance difference* (e.g. between data points 2^{nd} - 3^{rd} and 3^{rd} - 7^{th}). There will always be a discrepancy and Kriging treats it as a random variable with a *Gaussian probability density function* (PDF). If we want to predict the performance of another equidistant point from any known data point, the prediction will have variation

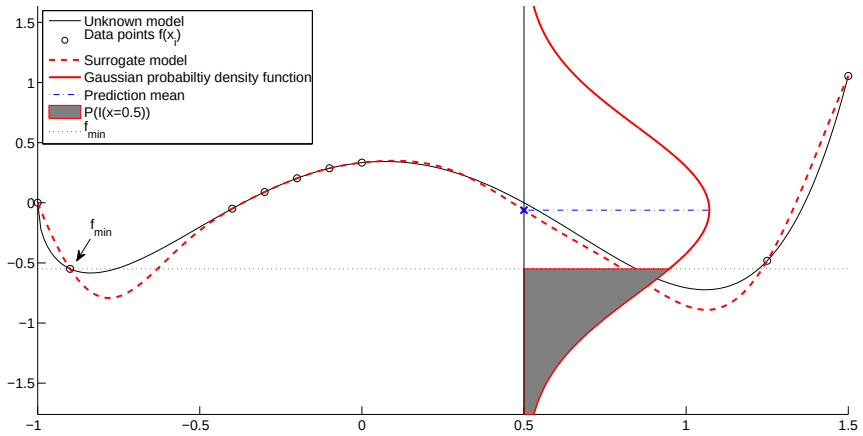


Figure 1.6: Graphical illustration of a Kriging model.

over the expected mean. An example is shown at $x = 0.5$, where the PDF curve estimates the performance probability over a continuous range. Furthermore, if we want to know the likelihood that a prediction is smaller than the currently existing minimum, we have to calculate the area of the PDF under the line f_{min}. This is shown as a shaded area in Figure 1.6 and it is termed as *probability of improvement* (will be further discussed in Section 1.4.1.4). A Kriging model is thus *a mathematical formula that estimates the performance mean and variance of unknown data points from a given set of known data points and performances*.

1.4.1.3 Stopping Criteria

To know whether a performance is a good approximation of the global optimum or the system model is accurate enough for making predictions, stopping criteria are applied.

– In performance optimization, the accuracy of an *approximate Pareto front* (APF) of the objectives is continuously checked and compared to a set of rules to stop the optimization process. In particular, the stopping criteria is composed of i) *progress indicators*: to estimate the current performance improvement by comparing previous iterations, ii) *evidence gathering process*: to perform a statistical analysis on the progress indicators and calculate the changes over time, and iii) *stopping decision*: to compare the results of the evidence gathering process against a predefined stopping decision.

– Whereas in system characterization, the accuracy of the model is verified continuously by using a *cross-validation* method. Cross-validation is a technique

where a given dataset is divided into k-1 training and 1 testing sets and the predicted response of the model, built from the training set, is compared against the testing set. Later, the process is repeated k times by shuffling the training and testing sets to add robustness to the prediction. Prediction accuracy is calculated by using error measurement metrics (i.e. Root Mean Square Error, Root Relative Square Error, Bayesian Estimation Error Quotient, R^2) and a lower value usually indicates a good prediction accuracy. In this research work, the *Root Relative Square Error* (RRSE) measure is applied which is defined as

$$RRSE(y, \tilde{y}) = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1.1}$$

where n is the number of samples, y is the measured value, $\tilde{y}$ is the predicted value and $\bar{y}$ is the mean of the measured values.

In all the cases (performance optimization and system characterization), the stopping criteria are upper limited by to a maximum number of iterations and it will stop execution once it passes a specified limit.

1.4.1.4 Sampling Strategy

The computation complexity of wireless networks during performance optimization and system modeling was the main reason for having a sequential design approach and the sampling strategy has a key role in this process.

– In performance optimization, the goal is to evaluate the optimum settings in a short period of time. In order to do so, the *probability of improvement* (PoI) [7] of every untested point, from the Kriging model, is compared and the one having the highest PoI is selected for the next round test. PoI sampling strategy estimates the chance that an untested point is better than the current optimum performance. If the Kriging prediction is uncertain, the estimated random variable will have high variance and samples are taken from this region. This is called *exploration* and it searches for unvisited regions. On the other hand, adequate sampling reduces the prediction variance and optimum regions are *exploited* by using currently available information. This way, the PoI sampling strategy balances both exploration and exploitation and guides the optimization process to a global optimum solution.

– For system characterization, RRSE of the Kriging model is checked at every iteration and a new sample point is selected if the model accuracy is not satisfactory. Recall that the initial model is designed to provide a good starting point by *exploring* the configuration space of the system. However, it has to be further explored since limited design points are used. Even worse, the initial

model has not *exploited* interesting regions of the system yet (i.e. discontinuities, non-linear regions and local optima), and this has to be included in the sampling strategy as well. To this end, a novel sampling strategy called *FLOLA-Voronoi* [32] is used to guide the modeling process through balancing the exploration and exploitation processes. The *FLOLA* part, a Fuzzy implementation of a Local Linear Approximation-based (LOLA) sequential design strategy [33], is responsible for exploiting non-linear regions of the complex system. It uses local linear approximations to measure the linearity of the region. The main idea behind this is that, non-linear regions are difficult to model compared to linear regions and thus more sample points are required in the modeling process. On the other hand, the *Voronoi* part explores the configuration space by searching sparsely sampled regions. It draws a Voronoi tessellation diagram from tested points and calculates the area of each Voronoi cells. A smaller area indicate the presence of nearby points thus a tightly packed region and a wider area indicate the absence of nearby points thus a sparsely packed region. Finally, the scores from FLOLA and Voronoi calculations are combined together to decide the next sample point, for improving the system model. Figure 1.7 explains the FLOLA-Voronoi sampling strategy graphically.

1.4.2 One-shot Design

Unlike the sequential approach, a *one-shot* approach (i) designs an experiment upfront and (ii) analyzes the result after all tests are executed. In particular, it is used to screen sensitive parameters of complex wireless networks.

1.4.2.1 Initial Design

To design the initial sample points of a screening experiment, a combinatorial design method called *locating array* (LA) [30] is used. LA is designed to screen main effects and interactions of a complex engineered system, such as wireless networks. It starts by designing a *covering array* (CA), to consider the response of the system on main effects and interactions. Later, a locating property is added on top of the covering array and the contributions from main effects and interactions are uniquely identified. By doing so, LA loses balance and traditional analysis methods cannot be used in the screening process. Balance is a measure of the symmetry of the covering property, and LA doesn't preserve symmetry, unlike CA. On the other hand, the size of LA grows logarithmically with the number of parameters and this makes it suitable for complex wireless networks.

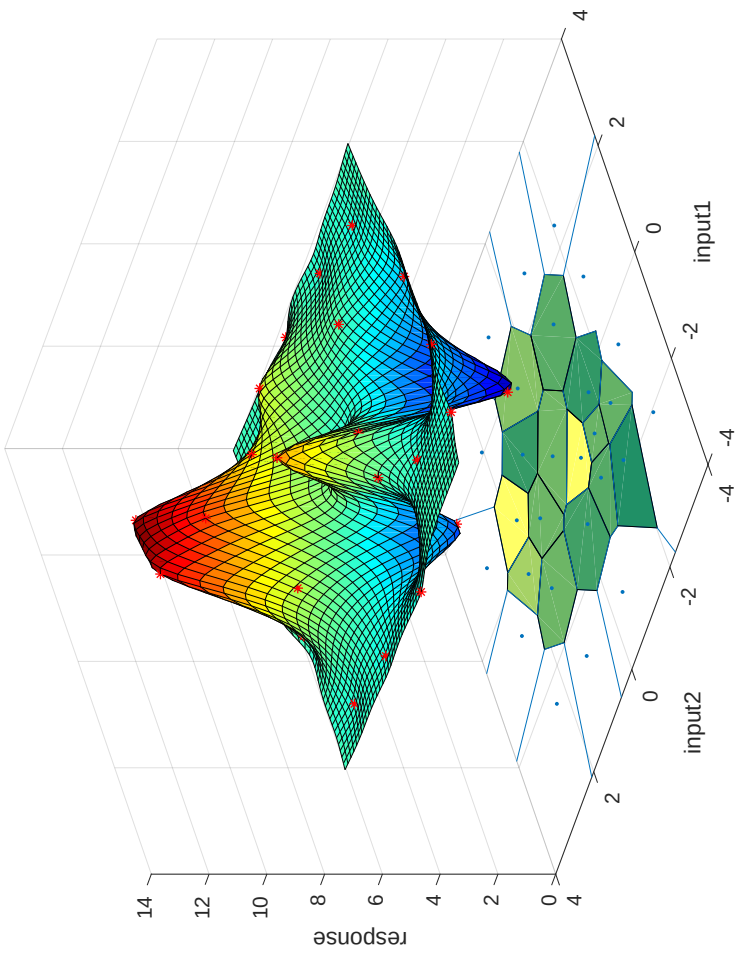


Figure 1.7: Operation of FLOLA-Voronoi sampling strategy using a modified peaks function. The Voronoi part, responsible for exploring the configuration space, is estimated by the area of the Voronoi cells. Larger cells result in better scores because they represent sparsely packed regions. The FLOLA part, on the other hand, exploits non-linear regions and it is indicated by a 'summer' color map on the Voronoi diagram. Drier cells indicate non-linear regions and have a better score compared to greener cells.

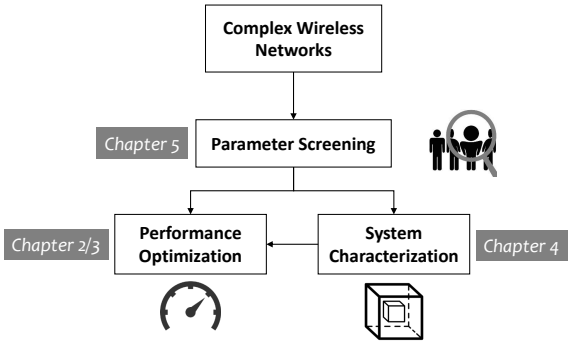


Figure 1.8: Mapping of different chapters and research methodologies.

1.4.2.2 Analysis

After all LA tests are executed, the dataset is analyzed using a *backtracking orthogonal matching pursuit* (BT-OMP) method. BT-OMP is a variation of the orthogonal matching pursuit (OMP) method, customized to handle the unbalanced nature of LA. A candidate model is evaluated incrementally by adding terms on every iteration. While adding terms, the dot product of model residuals and term's *compressing sensing matrix* is calculated and a term with the highest dot product value is selected. Later, BT-OMP evaluates candidate models with a given *coefficient of determination* (R^2) score. Each evaluated model is an approximate solution of the underlying complex system, and promising candidate models are combined together to screen the most sensitive parameters.

1.5 Outline

Until now, we have talked about the different challenges of complex wireless networks and the proposed solution to solve them. In this section, we look at the detailed outline of the research conducted. Figure 1.8 maps the different book chapters to the research methodologies applied.

Chapter 2 optimizes the performance of a complex wireless conferencing system. It starts discussing the *curse of dimensionality* [34], where an increase in the number of design parameters results in an exponential growth of the configuration space, thus requiring a huge amount of time to find the *optimum operating points*. The chapter also describes different techniques to solve wireless network problems, such as heuristic evolutionary algorithms. These techniques, however, are mostly bound to simulation environments because they inherently require a lot of iterations before evaluating the optimum solution. Moreover, the underlying propagation models are usually inadequate to mimic the behavior of the wireless

physical layer, thus shifting to an *experiment driven research*. In this direction, the chapter also adds its limitations (i.e. start-up overheads and orchestration delays) thus justifying the use of *Efficient Global Optimization (EGO)* methods to minimize the overall experimentation effort. Afterward, the chapter describes the wireless conferencing scenario and explains the different steps involved in the optimization process. Being an experiment-driven research, there are two important points that need consideration, i) repeatability and ii) experiment outliers. Repeatability, a condition an experiment measures a similar performance at different time instances, is a basic requirement for comparing results of different experiments. Experiment outliers are also common in wireless experiments due to possible interference or hardware/software malfunctions, thus needing to conduct multiple identical experiments and removing the outliers. In the optimization process, initial designs bootstrap a surrogate optimizer (a *surrogate modeling* tool + a *Probability of Improvement* sampling strategy) and stopping criteria is continuously checked to stop the experiment execution. While initial designs (*Latin Hypercube sampling*, *Orthogonal sampling*, *Random sampling*, *Hammersley Sequence sampling*) affect the optimization performance, the sample size of the initial design also plays a crucial part in the balance between *exploration* vs *exploitation* tradeoff. The surrogate modeling tool optimizes a combined objective of increased audio quality and reduced transmission exposure, which are conflicting towards each other. Depending on application requirements, different weights can be applied to combine the objectives and unique optimum settings are calculated as a result. As stopping criteria, the progress of the combined objective is continuously checked and when it gets sufficiently stable or when the iteration exceeds a threshold, the experiment is stopped. Finally, the performance of multiple optimization experiments (using different sampling methods and scenarios) is compared to an exhaustively searched experiment, using *duration gain* and *performance gain* metrics.

Similar to Chapter 2, Chapter 3 also optimizes the performance of a complex wireless conferencing system. However, Chapter 3 goes beyond a global optimization solution by pushing the limits of different complexity metrics. Following is the list of changes made in Chapter 3 when compared to Chapter 2.

– The size of the wireless audio conferencing scenario is increased by adding more listener nodes.

– More design parameters are used and the configuration space has increased as a result.

– Advanced performance objectives are used.

– Advanced stopping criteria are used.

– Performance objectives are treated individually and therefore *Pareto Front* is used to analyze their behavior.

Chapter 3 starts with different multi-objective optimization approaches in wireless networks, whereby multi-objective versions of the evolutionary algorithms are the most widely used methods. This brought us back to the same problem we had in the previous chapter, where a large number of iterations are inherently required before evaluating the optimum solution. The solution considered is also a multi-objective version of the surrogate optimizer which includes a *surrogate modeling* tool and a *Hyper volume based Probability of Improvement (HV-PoI)* sampling strategy. Regarding the performance objectives, the audio quality used in the previous chapter only considered degradation over the wireless medium, which is a function of *packet error rate*, *jitter* and *latency* metrics. In this chapter, the quality impact due to audio codecs is also included. When looking into stopping criteria, multiple decisions are combined together using *Progress Indicators* (PIs), *Evidence Gathering Processes* (EGPs) and *Stopping Decisions* (SDs). PI measures the improvement of current iteration from the previous iteration. Since multiple objectives are used, a *Pareto Front* (PF) measurement comparison is performed. Next, EGP is performed on the PIs using statistical measures (i.e. moving averages, standard deviation, linear regression and Kalman filtering). Finally, the result from EGPs are compared against a predefined SD and the experiment execution is stopped once conditions are met. Now looking into performance evaluation, *Speed up Factor* (SuF) and *Population Domination Rate* (PDR) metrics are used. While SuF compares the number of iterations, PDR measures the PF fitness between optimization experiments and an exhaustive search experiment. Finally, Chapter 3 concludes the discussion by analyzing the sensitivity of different initial sample sizes on the system performance and estimating the computational complexity of the optimization process.

Chapters 2 and 3 have considered performance optimization with the aim of finding an optimum configuration settings. This finds use in bootstrapping a wireless network by searching a good starting point. Chapter 4, however, is about the online operation of a wireless network, therefore characterizing the system performance for different environmental conditions. The wireless system under consideration is a *Wireless Sensor Network* (WSN) and the goal is to optimize its performance in the presence of a dynamically changing environment. Because of the environment dynamics, an optimally working WSN can become sub-optimal and the design parameters need to be re-calibrated to bring the system back to an optimum state. To this end, a *cloud repository* is used to store multiple models of the WSN performance, each representing a static instance of the dynamic environment. Therefore, by using all models stored in the cloud repository, it should be possible to reconstruct a large portion of the dynamic environment. We term this procedure as *model merging* since multiple models are merged together to repre-

sent the WSN under the influence of a dynamic environment. On another note, the chapter also discusses the different types of dynamic environments (i.e. *slow changing environment* and *fast changing environment*). We considered a slowly changing environment in this work because the time required to apply corrective actions is assumed to be lower than the static period of the dynamic environment. Moreover, the advantage of using a cloud repository is further supported by the fact that most WSN infrastructures (i.e. suburbs and apartment buildings) have similar construction and propagation models, thus facilitating scalability. Finally, the chapter validates the model selection and merging procedures by using different test environments and comparing them to real models, which are made only for validation purposes.

The last chapter of this dissertation is focused on parameter screening in complex wireless networks. It starts by explaining the impracticality of a large number of parameters in designed experiments. Thus traditional methods are infeasible to screen such systems. Instead, *supersaturated designs* (SSDs) are commonly used but they are limited to identifying only main effects. The chapter further emphasizes the need for screening interactions in complex wireless networks and proposes a new combinatorial design method called *locating array* (LA). LA is based on *covering arrays* and is able to cover main effects and interactions with a locating property added on top. As good as it is, LA also brings a challenge to the screening analysis because designed experiments are unbalanced. Balance is a measure of the symmetry of the covering property in the LA tests. As a solution, a modified version of *orthogonal matching pursuit* (OMP) called *backtracking OMP* (BT-OMP) is used. Furthermore, the chapter explains the proposed *tree-based search* strategy to analyze the results of multiple screening iterations. Since there are multiple candidate models from a given screening problem (with a given coefficient of determination R^2 score), promising candidate models are combined together to screen the most sensitive parameters.

1.6 Research Contributions

The research work on performance optimization, system characterization and parameter screening of complex wireless networks has resulted in a number of contributions. Following is a list of major contributions on a per chapter basis.

- Single-objective SUMO optimization of a wireless audio conferencing system (Ch. 2).
  - Integration of the SUMO toolbox into a wireless testbed.
  - Mean opinion score (MOS) calculation using latency, jitter and packet loss metrics.

  - Repeatability analysis of Wi-Fi experiments in the wireless testbed.
  - Experiment outlier detection using PRE and POST experiment monitoring.

- Multi-objective SUMO optimization of a wireless audio conferencing system (Ch. 3)
  - Advanced mean opinion score (MOS) calculation by quantifying impairments due to audio compression and wireless transmission.
  - Measurement of electromagnetic exposure from a Wi-Fi traffic using specific absorption rate (SAR) metric.
  - Advanced stopping criteria definition using PIs, EGPs, and SDs.
  - Pareto Front analysis of audio quality and transmission exposure performance objectives.

- SUMO based system characterization and performance optimization of a wireless sensor network in dynamic environments (Ch. 4)
  - The use of a cloud repository solution to optimize WSN performance in dynamic environments.
  - Characterization of a dynamic environment using a set of specially designed configuration parameters.
  - Merging of multiple reference models from the cloud repository to characterize an unknown environment.
  - Transferring the knowledge of existing cloud repository solutions, to speed up the optimization of later WSN deployments under similar conditions.

- Screening sensitive parameters of a complex wireless audio conferencing system (Ch. 5)
  - Design of LAs to screen main effects and interactions of complex wireless network parameters.
  - Analysis of LA measurements using OMP tree-based search strategy.
  - The use of backtracking search algorithm (i.e. BT-OMP) to combat noisy measurements in the screening process.
  - Validation of screening results using fractional-factorial experiment design.

1.7 Publications

1.7.1 Publications in international journals (listed in the ISI Web of Science [1])

1. **Michael Tetemke Mehari**, Eli De Poorter, Ivo Couckuyt, Dirk Deschrijver, Jono Vanhie-Van Gerwen, Daan Pareit, Tom Dhaene and Ingrid Moerman. *Efficient Global Optimization of Multi-Parameter Network Problems on Wireless Testbeds.* Published in Ad Hoc Networks, Volume 29, p.15–31, 2015.

2. **Michael Tetemke Mehari**, Eli De Poorter, Ivo Couckuyt, Dirk Deschrijver, Gunter Vermeeren, David Plets, Wout, Joseph, Luc Martens, Tom Dhaene and Ingrid Moerman. *Efficient Identification of a Multi-Objective Pareto Front on a Wireless Experimentation Facility.* Published in IEEE Transactions on Wireless Communications, Volume 15, Issue 10, p.6662–6675, 2016.

3. Wei Liu, Eli De Poorter, Jeroen Hoebeke, Emmeric Tanghe, Wout Joseph, Pieter Willemen, **Michael Tetemke Mehari**, Xianjun Jiao and Ingrid Moerman. *Assessing the Coexistence of Heterogeneous Wireless Technologies With an SDR-Based Signal Emulator: A Case Study of Wi-Fi and Bluetooth.* Published in IEEE Transactions on Wireless Communications, Volume 16, Issue 3, p.1755–1766, 2017.

4. **Michael Tetemke Mehari**, Adnan Shahid, Tom Van Steenkiste, Jan Bauwens, Ivo Couckuyt, Violet R. Syrotiuk, Dirk Deschrijver, Tom Dhaene, Ingrid Moerman, Eli De Poorter. *Metamodel Based WSN MAC Optimization in Dynamic Environments using Cloud Repositories.* Submitted to IEEE/ACM Transactions on Networking 2018.

5. Randy Compton, **Michael Tetemke Mehari**, Charles J. Colbourn, Eli De Poorter, Ingrid Moerman, Violet R. Syrotiuk. *An Efficient Screening Method for Identifying Parameters and Interactions that Impact Wireless Network Performance.* Submitted to IEEE Transactions on Wireless Communications 2017.

1.7.2 Publications in other international journals

1. David Plets, Krishnan Chemmangat, Dirk Deschrijver, **Michael Tetemke Mehari**, Selvakumar Ulaganathan, Mostafa Pakpar, Tom Dhaene, Jeroen

[1] The publications listed are recognized as 'A1 publications', according to the following definition used by Ghent University: A1 publications are articles listed in the Science Citation Index Expanded, the Social Science Citation Index or the Arts and Humanities Citation Index of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper.

Hoebeke, Ingrid Moerman, Emmeric Tanghe. *Surrogate modeling based cognitive decision engine for optimization of WLAN performance.* Published in the Journal of Mobile Communication, Computation and Information, Volume 23, issue 8, pp. 2347–2359, 2017

1.7.3 Publications in international conferences (listed in the ISI Web of Science [2])

1. Wei Liu, S Keranidis, **Michael Tetemke Mehari**, Jono Vanhie-Van Gerwen, Stefan Bouckaert, Opher Yaron and Ingrid Moerman. *Various Detection Techniques and Platforms for Monitoring Interference Condition in a Wireless Testbed.* Published in European Workshop of EU 7th Framework Programme (FP7) on Measurement Methodology and Tools, p.43–60, 09 May 2012, Aalborg, Denmark.

2. **Michael Tetemke Mehari**, Eli De Poorter, Ivo Couckuyt, Dirk Deschrijver, Jono Vanhie-Van Gerwen, Tom Dhaene and Ingrid Moerman. *Efficient Multi-Objective Optimization of Wireless Network Problems on Wireless Testbeds.* Published in international Conference on Network and Service Management (CNSM 2014), p.212–217, 17–21 Nov 2014, Rio De Janeiro, Brazil.

3. Mostafa Pakparvar, Krishnan Chemmangat Manakkal Cheriya, Dirk Deschrijver, **Michael Tetemke Mehari**, David Plets, Tom Dhaene, Jeroen Hoebeke, Ingrid Moerman, Luc Martens and Wout Joseph. *Throughput Optimization of Wireless LANs by Surrogate Model Based Cognitive Decision Making.* Published in IEEE Wireless Communications and Networking Conference Workshops (WCNCW 2015), p.188–193, 09–12 Mar 2015, New Orleans, LA, USA.

4. Mostafa Pakparvar, David Plets, Jeroen Hoebeke, Dirk Deschrijver, **Michael Tetemke Mehari**, Tom Dhaene, Ingrid Moerman, Luc Martens and Wout Joseph. *Throughput Optimization Strategies for Large-Scale Wireless LANs.* Published in IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB 2015), 17–19 Jun 2015, Ghent Univ, Ghent, Belgium.

5. Randy Compton, **Michael Tetemke Mehari**, Charles J Colbourn, Eli De Poorter, Ingrid Moerman and Violet R. Syrotiuk. *Screening Interacting Fac-*

[2]The publications listed are recognized as 'P1 publications', according to the following definition used by Ghent University: P1 publications are proceedings listed in the Conference Proceedings Citation Index - Science or Conference Proceedings Citation Index - Social Science and Humanities of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper, except for publications that are classified as A1.

tors in a Wireless Network Testbed Using Locating Arrays. Published in IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS 2016), 10–14 Apr 2016, San Francisco, CA, USA.

6. Jetmir Haxhibeqiri, **Michael Tetemke Mehari**, Wei Liu, Eli De Poorter, Wout Joseph, Ingrid Moerman and Jeroen Hoebeke. *Wireless Handover Performance in Industrial Environments: a Case Study.* Published in IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2016), 06–09 Sep 2016, OWL Univ Appl Sci, Fraunhofer IOSB INA, Berlin, Germany.

1.7.4 Publications in other international conferences

1. Stephen Seidel, **Michael Tetemke Mehari**, Charles J Colbourn, Eli De Poorter, Ingrid Moerman and Violet R. Syrotiuk. *Analysis of Large-Scale Experimental Data from Wireless Networks.* Published in IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS 2018), 15–19 April 2018, Honolulu, HI, USA.

2. Le Tian, **Michael Tetemke Mehari**, Serena Santi, Steven Latr. *IEEE 802.11ah Restricted Access Window Surrogate Model for Real-Time Station Grouping.* Published in IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), 12–15 June 2018, CHANIA, GREECE.

References

[1] D. Evans. *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything*. Technical report, CISCO, April 2011.

[2] Ericsson. *5G: what is it?* Technical report, Ericsson Telephone Co. Ltd, Oct 2014. Available from: https://www.ericsson.com/assets/local/news/2014/10/5g-what-is-it.pdf.

[3] A. N. Aldaco, C. J. Colbourn, and V. R. Syrotiuk. *Locating Arrays: A New Experimental Design for Screening Complex Engineered Systems*. SIGOPS Oper. Syst. Rev., 49(1):31–40, January 2015. Available from: http://doi.acm.org/10.1145/2723872.2723878, doi:10.1145/2723872.2723878.

[4] M. Payaró, M. Färber, P. Vlacheas, N. Bartzoudis, F. Tillman, D. Ferling, V. Berg, T. Rautio, P. Serrano, and D. Sabella. *Flex5Gware: Flexible and Efficient Hardware and Software Platforms for 5G Network Elements and Devices*. Trans. Emerg. Telecommun. Technol., 27(9):1242–1249, September 2016. Available from: https://doi.org/10.1002/ett.3070, doi:10.1002/ett.3070.

[5] A. E. Kouche. *Towards a wireless sensor network platform for the Internet of Things: Sprouts WSN platform*. In 2012 IEEE International Conference on Communications (ICC), pages 632–636, June 2012. doi:10.1109/ICC.2012.6364196.

[6] Y. Qu and S. Georgakopoulos. *Relocation of wireless sensor network nodes using a genetic algorithm*. In Wireless and Microwave Technology Conference (WAMICON), 2011 IEEE 12th Annual, pages 1–5, 2011. doi:10.1109/WAMICON.2011.5872882.

[7] I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, and L. Knockaert. *Surrogate-based infill optimization applied to electromagnetic problems*. INTERNATIONAL JOURNAL OF RF AND MICROWAVE COMPUTER-AIDED ENGINEERING, 20(5):492–501, 2010. Available from: http://dx.doi.org/10.1002/mmce.20455.

[8] S. Gilmour. *Factor Screening via Supersaturated Designs*. In A. Dean and S. Lewis, editors, Screening, pages 169–190. Springer New York, 2006.

[9] V. Srivastava and M. Motani. *Cross-Layer Design and Optimization in Wireless Networks*, pages 121–146. John Wiley & Sons, Ltd, 2007. Available from: http://dx.doi.org/10.1002/9780470515143.ch6, doi:10.1002/9780470515143.ch6.

[10] G. E. P. Box, J. S. Hunter, and W. G. Hunter. *Statistics for experimenters : design, innovation, and discovery*. Wiley-Interscience, 2005. Available from: http://www.worldcat.org/isbn/9780471718130.

[11] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley and Sons, Inc., 7th edition, 2009.

[12] X. Gao and Y. Gao. *TDMA Grouping Based RFID Network Planning Using Hybrid Differential Evolution Algorithm.* In F. L. Wang, H. Deng, Y. G. 0001, and J. Lei, editors, AICI (2), volume 6320 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 2010. Available from: http://dblp.uni-trier.de/db/conf/aici/aici2010-2.html#GaoG10.

[13] C. A. C. Coello and N. C. Cortés. *Solving Multiobjective Optimization Problems Using an Artificial Immune System*. Genetic Programming and Evolvable Machines, 6(2):163–190, Jun 2005. Available from: https://doi.org/10.1007/s10710-005-6164-x, doi:10.1007/s10710-005-6164-x.

[14] J. F. Yan, Y. Gao, and L. Yang. *Ant colony optimization for wireless sensor networks routing*. In 2011 International Conference on Machine Learning and Cybernetics, volume 1, pages 400–403, July 2011. doi:10.1109/ICMLC.2011.6016670.

[15] R. Kulkarni and G. Venayagamoorthy. *Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey*. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 41(2):262–267, 2011. doi:10.1109/TSMCC.2010.2054080.

[16] K. Kaur, M. Rattan, and M. S. Patterh. *Optimization of Cognitive Radio System Using Simulated Annealing*. Wirel. Pers. Commun., 71(2):1283–1296, July 2013. Available from: http://dx.doi.org/10.1007/s11277-012-0874-1, doi:10.1007/s11277-012-0874-1.

[17] B. K. Donohoo, C. Ohlsen, S. Pasricha, Y. Xiang, and C. Anderson. *Context-Aware Energy Enhancements for Smart Mobile Devices*. IEEE Transactions on Mobile Computing, 13(8):1720–1732, Aug 2014. doi:10.1109/TMC.2013.94.

[18] C. K. Wen, S. Jin, K. K. Wong, J. C. Chen, and P. Ting. *Channel Estimation for Massive MIMO Using Gaussian-Mixture Bayesian Learning*. IEEE Transactions on Wireless Communications, 14(3):1356–1368, March 2015. doi:10.1109/TWC.2014.2365813.

[19] T. OShea and J. Hoydis. *An Introduction to Deep Learning for the Physical Layer*. IEEE Transactions on Cognitive Communications and Networking, 3(4):563–575, Dec 2017. doi:10.1109/TCCN.2017.2758370.

[20] W. Kim, J. Park, J. Yoo, H. J. Kim, and C. G. Park. *Target Localization Using Ensemble Support Vector Regression in Wireless Sensor Networks*. IEEE Transactions on Cybernetics, 43(4):1189–1198, Aug 2013. doi:10.1109/TSMCB.2012.2226151.

[21] J. Zhou, Y. Zhang, Y. Jiang, C. L. P. Chen, and L. Chen. *A distributed K-means clustering algorithm in wireless sensor networks*. In 2015 International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS), pages 26–30, Aug 2015. doi:10.1109/ICCSS.2015.7281143.

[22] A. Rooshenas, H. R. Rabiee, A. Movaghar, and M. Y. Naderi. *Reducing the data transmission in Wireless Sensor Networks using the Principal Component Analysis*. In 2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pages 133–138, Dec 2010. doi:10.1109/ISSNIP.2010.5706781.

[23] Y. Xu, J. Chen, L. Ma, and G. Lang. *Q-Learning Based Network Selection for WCDMA/WLAN Heterogeneous Wireless Networks*. In 2014 IEEE 79th Vehicular Technology Conference (VTC Spring), pages 1–5, May 2014. doi:10.1109/VTCSpring.2014.7023063.

[24] M. Holland, T. Wang, B. Tavli, A. Seyedi, and W. Heinzelman. *Optimizing Physical-layer Parameters for Wireless Sensor Networks*. ACM Trans. Sen. Netw., 7(4):28:1–28:20, February 2011. Available from: http://doi.acm.org/10.1145/1921621.1921622, doi:10.1145/1921621.1921622.

[25] A. Michaloliakos, R. Rogalin, Y. Zhang, K. Psounis, and G. Caire. *Performance Modeling of Next-Generation Wireless Networks*. CoRR, abs/1405.0089, 2014. Available from: http://arxiv.org/abs/1405.0089, arXiv:1405.0089.

[26] A. Forrester, A. Sobester, and A. Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 07 2008.

[27] A. D&#-19;az Manr&#-19;quez, G. Toscano, J. H. Barron-Zambrano, and E. Tello-Leal. *A Review of Surrogate Assisted Multiobjective Evolutionary Algorithms*. Intell. Neuroscience, 2016:19–, June 2016. Available from: https://doi.org/10.1155/2016/9420460, doi:10.1155/2016/9420460.

[28] C. Sun, Y. Jin, J. Zeng, and Y. Yu. *A two-layer surrogate-assisted particle swarm optimization algorithm*. Soft Computing, 19(6):1461–1475, Jun 2015. Available from: https://doi.org/10.1007/s00500-014-1283-z, doi:10.1007/s00500-014-1283-z.

[29] D. Shepard. *A Two-dimensional Interpolation Function for Irregularly-spaced Data*. In Proceedings of the 1968 23rd ACM National Conference, ACM '68, pages 517–524, New York, NY, USA, 1968. ACM. Available from: http://doi.acm.org/10.1145/800186.810616, doi:10.1145/800186.810616.

[30] C. J. Colbourn and D. W. McClary. *Locating and detecting arrays for interaction faults.* J. Comb. Optim., 15(1):17–48, 2008.

[31] F. A. C. Viana. *Things you wanted to know about the Latin hypercube design and were afraid to ask*. 10th World Congress on Structural and Multidisciplinary Optimization, May 19 -24 2013.

[32] J. van der Herten, I. Couckuyt, D. Deschrijver, and T. Dhaene. *A fuzzy hybrid sequential design strategy for global surrogate modeling of high-dimensional computer experiments*. SIAM Journal on Scientific Computing, 37(2):A1020–A1039, 2015.

[33] K. Crombecq, D. Gorissen, D. Deschrijver, and T. Dhaene. *A novel sequential design strategy for global surrogate modeling*. SIAM Journal on Scientific Computing, 33(4):1948–1974, 2011.

[34] R. E. Bellman. *Dynamic Programming*. Dover Publications, Incorporated, 2003.

2

Efficient Global Optimization of Multi-Parameter Network Problems on Wireless Testbeds

In the introductory chapter, it was emphasized that performance optimization is a major issue in complex wireless networks because exhaustively trying out all different parameter combinations is a time-consuming process. Thus, this section tackles the performance optimization problem in complex wireless networks. To this end, the following innovations are realized: (i) 'surrogate modeling tools' are used as 'black-box' optimizers and (ii) a combined performance objective (improved audio quality and reduced electromagnetic exposure) is optimized in a wireless audio conferencing system.

⋆ ⋆ ⋆

Michael Tetemke Mehari, Eli De Poorter, Ivo Couckuyt, Dirk Deschrijver, Jono Vanhie-Van Gerwen, Daan Pareit, Tom Dhaene, Ingrid Moerman

Published in Ad Hoc Networks Volume 29 Pages 15-31, June 2015.

Abstract A large amount of research focuses on experimentally optimizing the performance of wireless solutions. Finding the optimal performance settings typically requires investigating all possible combinations of design parameters, while

the number of required experiments increases exponentially for each considered design parameter. The aim of this paper is to analyze the applicability of global optimization techniques to reduce the optimization time of wireless experimentation. In particular, the paper applies the Efficient Global Optimization (EGO) algorithm implemented in the SUrrogate MOdeling (SUMO) toolbox inside a wireless testbed. Moreover, to cope with the unpredictable nature of wireless testbeds, the paper applies an experiment outlier detection which monitors outside interference and verifies the validity of conducted experiments. The proposed techniques are implemented and evaluated in a wireless testbed using a realistic wireless conferencing scenario. The performance gain and experimentation time of a SUMO optimized experiment is compared against an exhaustively searched experiment. In our proof of concept, it is shown that the proposed SUMO optimizer reaches 99.79% of the global optimum performance while requiring 8.67 times less experiments compared to the exhaustive search experiment.

2.1 Introduction

Wireless networks are utilized in many application domains. For example, if a home user is wirelessly connected, he can move around with his laptop or mobile device, while staying connected to his peers. Wireless sensor networks can be used in applications as diverse as early-warning systems for forest fire and home automation. Body area networks attached to a patient for heath-monitoring purposes make the patient-doctor interaction more productive. These wireless innovations trigger the wireless research community to continuously introduce and validate novel wireless concepts. Such research problems often have several design parameters that can be changed. For example, Wi-Fi networks have parameters that can be tweaked at the physical layer (e.g. transmit power, channel, modulation), MAC layer (e.g. inter frame spacing, contention window), network layer (e.g. routing protocol, mobility, topology) and application layer (e.g. throughput, server configurations). Optimizing all or a subset of these parameters (a.k.a. multi-parameter optimization) in order to find the *optimum operating point* is time consuming since the design space grows exponentially for every investigated design parameter.

Often, these wireless networks are optimized using wireless network simulations. These simulators generate a number of interference and traffic patterns, create a propagation model of the wireless medium, execute the optimization algorithms and analyze a set of performance metrics. However, wireless network simulators also have a number of disadvantages. Results can be very different when executing identical experiments on multiple wireless network simulators. In [1], the accuracy of Opnet, ns-2, and GloMoSim simulators indicate significant differences when evaluating a single protocol problem. Another limitation of a wireless network simulator is its incapability to accurately model the underlying

wireless transmission properties such as channel characteristics and antenna diversity. It is also very hard to model the hardware's imperfections and dissimilarities between devices of the same type [2], which often have a considerable impact on the overall network performance.

As a result, experimentally driven research is necessary to complement simulations [2]. Measurements and performance evaluations on a real-life testbed are gaining more attention as they account for hardware imperfections and dissimilarities. However, wireless testbeds also have limitations. They require more set-up overhead compared to their simulator counterparts before, during and after experimentation. Typical examples are resource management, turning on radio interfaces, message orchestration and output post processing. For example, when using the Orbit Management Framework (OMF) for experimentation control, an experiment having N wireless devices adds an average delay of 5.17*N ms on a single message orchestration [3]. In addition, experiments on real-life testbeds can not be artificially speed up, which is possible when using simulations. In order to mitigate the time overhead, efficient optimization algorithms can be used that are best fitted to wireless testbeds. Two of their most widely used approaches are selective sampling of the design space and sensitivity analysis on the design parameters. In this paper, we investigate the selective sampling approach of Efficient Global Optimization (EGO) [4] implemented in the SUrrogate MOdeling (SUMO) toolbox [5]. EGO uses Kriging approximations to find optimal operation point(s) of a complex problem while minimizing the number of experiments needed. This way, the overall experimentation time is kept to a minimum [6]. In a nutshell, this paper examines the strengths of the SUMO optimizer by applying it to a network problem in a wireless testbed having multiple design parameters.

This paper presents the following novel contributions.

– Integration of the SUMO toolbox in a wireless testbed.

– Definition of a wireless conferencing scenario which involves multiple design parameters and performance objectives.

– A simple mechanism for detecting outliers during Wi-Fi experiments.

– Repeatability analysis of Wi-Fi experiments.

– Sensitivity analysis of global optimization to the choice of the initial sample experiments.

– A generic stopping criteria that can be used in a variety of optimization problems.

The remainder of this paper is organized as follows. Section 2.2 explores the related work on multi-parameter optimization in wireless networks. The principles

of SUMO optimization and modifications to the SUMO toolbox are explained in section 2.3. In section 2.4, a wireless conferencing system is experimentally set-up and optimized using SUMO. The results of the experiment optimization process are presented and analyzed in section 2.5. Finally Section 2.6 concludes the paper.

2.2 Related work

Solutions of wireless network problems often involve multi-objective optimizers in order to optimize multiple design parameters. In literature, a wide range of multi-objective optimization algorithms exist. The effectiveness of such algorithms greatly depends on the methodology behind their implementation as measured by time, processing power, memory and performance. During the optimization process, optimizers carefully investigate two aspects. These are exploration and exploitation [7]. Exploration refers to the phase in which an optimizer understands the dynamics of a problem by selecting as few random sample points as possible. These random sample points have to be selected carefully in order not to waste valuable experimentation time. On the other hand, the exploitation phase locates local optimums starting from the explored design space. If the problem has been explored very well, the exploitation phase guarantees to locate global optimums. Therefore, the question of predicting global optimums in a short period of time creates the *exploration vs exploitation trade off* [8] which all multi-objective optimizers target.

Exhaustive search approaches evaluate all operating points of a solution to select optimum settings from the design space. A generic numerical calculation approach using MATLAB is presented in [9]. This algorithm exhaustively searches the design space and determines the optimum point to give the highest performance objective.

Genetic Algorithms (GA) [10] are heuristic algorithms that mimic the process of natural selection. Starting from an initial population (that consists of so-called chromosomes), new generations are produced, which hopefully contain better (i.e. fitter) chromosomes than the previous generation. The optimization process selects new offsprings according to a fitness function and the evolutionary iterations continue until a predefined stopping criterion is met.

A Particle Swarm Optimization (PSO) [11] algorithm optimizes a problem by exchanging information with neighboring particles such that a single particle with given position and velocity parameters searches an optimum setting. PSO works based on a mathematical formula optimizing a population of solutions (i.e. particles). Finally the optimization process stops when the improvement is below a given limit.

Differential Evolution (DE) [12], similar to GA, starts from a given population and a fixed number of randomly initialized vectors. In every iteration, a newer

Table 2.1: Design parameters, performance objective, execution method of different optimization algorithms, applied to a variety of complex wireless network problems

Algorithm	**Problem definition**	**Design parameters**	**Performance objectives**	**Execution method**	**Refer.**
Numerical calculation	Tuning of physical layer parameters in Wireless Sensor Network	Node hop distance, Transmit energy, Modulation schemes	Energy per Successful received Bit↓	simulation	[9]
GA	Maximizing sensing converge of wireless sensor network	Sensor positions	Relocation energy↓	simulation	[10]
PSO	Wireless Sensor Network deployment, Node localization, Node clustering and Data aggregation	Node positions, Transmit power, Sensor configuration	Quality of Service↑ Network lifetime↑ Localization error↓ Transmit power↓ Reliability↑	simulation	[11]
DE	Radio Frequency Identifier network planning	Position, Angle, Transmit power	Coverage↑ Interference↓ Cost↓	simulation	[12]
SA	Cognitive Radio system optimization	Transmit power, Modulation type	Power consumption↓ Bit Error Rate↓ Throughput↑	simulation	[13]

generation is produced by randomly combining the vectors in order to create a mutation. The newer generation mixed with the target vector is evaluated against an objective function and the selector decides whether or not it should be accepted to compose the next generation.

Simulated Annealing (SA) [13] algorithm is based on the analogous principle of freezing liquid when forming a crystalline structure such that with sufficient time the structure acquires a minimum energy state. In each iteration step, the newly generated point is checked against the current point based on a probability distribution scale proportional to the problem's analogous temperature. Such points are accepted when the total objective function decreases and the iteration continue until the stopping criteria are met.

Table 2.1 compares the different multi-parameter optimization algorithms that have been applied to wireless network problems. All multi-parameter optimization algorithms applied on the wireless network problems made use of simulation as an execution method which has several disadvantages, as outlined in the introductory section. On the other hand, this paper investigates the SUMO toolbox to evaluate its suitability for wireless network optimization. The SUMO optimization toolbox is often used in electromagnetic [14] and aerodynamic [15] optimization problems. Even though we are applying the SUMO toolbox in a wireless testbed for the first time, previous comparisons on multi-objective optimizers [14] [15] favours the SUMO variants which our preference is based upon. Therefore this paper goes beyond the state-of-the-art by (i) evaluating the suitability of the SUMO optimizer for wireless problems and (ii) evaluating the feasibility of multi-objective optimization algorithms in real-life experimentation rather than simulation.

2.3 SUMO

2.3.1 Optimizer principles

The SUMO optimizer is an efficient implementation of the well-known Expected Improvement (EI) criterion, popularized by Jones et al. in [6]. In this work, the optimizer is applied to accelerate the optimization of wireless network problems, because the execution of multiple experiments on a wireless testbed is often a time-consuming procedure. A typical optimization problem comprises a set of N network parameters, denoted by a vector **x**=$\{x_i\}$ for $i = 1, ..., N$, which may consist of either discrete numerical or continuous variables. These parameters need to be tuned within the bounds of a pre-specified parameter range of interest [L_i, B_i] in order to reach an optimal network performance. The aim of the overall procedure is to optimize a given objective function $f(.)$ (for example, to maximize the Quality of Service/Quality of Experience (QoS/QoE), or to minimize a certain cost such as energy consumption, etc.). The optimization algorithm starts from a well-chosen initial experimental design, and a global (but only locally accurate) Kriging surrogate model of the objective function is computed. Such Kriging models are part of a broader class of approximation methods, called the Gaussian Processes (GP), and have some interesting properties that can be exploited by the optimizer. Whereas the standard approximation methods predict only a single function value, GP methods can predict the uncertainty of a function value as the realization of a normally distributed random variable $Y(\mathbf{x}) \sim N(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$, where $\mu(\mathbf{x})$ represents the predicted value for $f(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ the prediction variance at an arbitrary point $\mathbf{x}$ in the parameter space. Based on this random variable $Y(\mathbf{x})$, different statistical criteria (such as the Probability of Improvement (PoI) or Expected Improvement (EI)) can be computed to quantify how interesting a new point in the design space is. In this work, we adopt the EI criterion which simultaneously balances exploration and exploitation [8] of the parameter space. This corresponds to the improvement that is expected to occur when compared to the optimum value obtained so far (i.e., f_{min} or f_{max}). By picking additional points with the highest EI value in the parameter space, the optimization process is directed towards a configuration with optimal performance. For example, in the case of a minimization problem, it can be written in the form of an integral as in [14] where $\varphi(.)$ represents the probability density function of a random variable and $I(x)$ is the improvement function.

$$E[I(x)] = \int_{-\infty}^{f_{min}} I(x)\varphi(Y(x))dY$$

where the improvement function is defined as

$$I(x) = max(f_{min} - Y(x), 0)$$

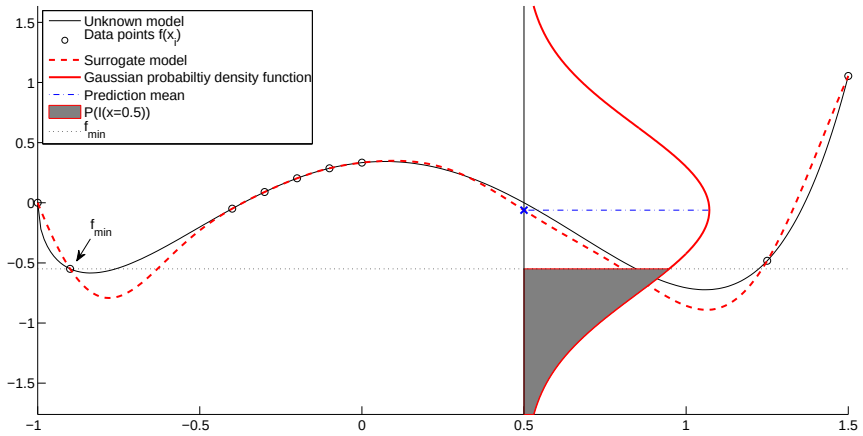


Figure 2.1: Graphical illustration of a Kriging model and the expected improvement criterion. A surrogate model (dashed line) is constructed based on a set of data points (circles). For each point the surrogate model predicts a Gaussian probability density function (PDF). An example of such a PDF is drawn at x = 0.5. The volume of the shaded area is the probability of improvement (PoI) and the first moment of this area is the expected improvement

$E[I(x)]$ corresponds to the improvement that is expected to occur when compared to the optimal value of the objective obtained so far. A graphical illustration of this criterion is shown in Figure 2.1.

A more detailed explanation can be found in Section II-B of [14]. Note that this EI criterion can also be expressed and evaluated in a closed-form, and it is optimized over the parameter space. The selection of new points corresponds to the execution of new experiments on the testbed and the outcome of these results is used to update the Kriging surrogate model. The process of performing experiments and subsequently updating the model to optimize the objective function is iterated until a stopping criterion is met. Typically, Kriging and the EI criterion are used to solve continuous optimization problems though it can be easily applied to discrete optimization problems too. The optimization of the EI criterion for discrete problems can simply be done by traditional discrete optimizers (such as a discrete pattern search or the discrete version of the CMA-ES algorithm [16]. However, regarding the dimensionality of the problem in this paper it is chosen to evaluate the EI criterion over the complete (and limited) set of discrete possibilities and the best point is chosen. The discrete variables of this work are ordinal of nature, which means that standard Kriging can be directly applied (the standard continuous correlation functions can be used). Though, nominal or categorical variables can also be easily handled by Kriging by using an appropriate function

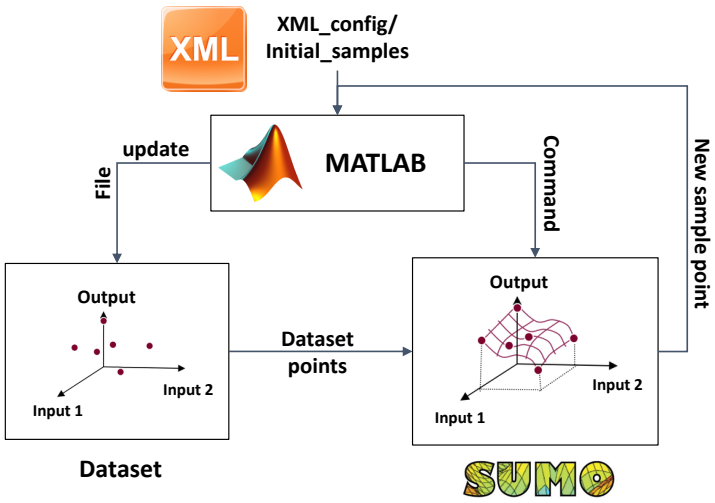


Figure 2.2: Overview of generic SUMO toolbox

to describe the correlation between the discrete data points [17].

In short, the SUMO optimizer is an effective approach to optimize network performance on a real-life testbed. The experimental results confirm its effectiveness and robustness.

2.3.2 Toolbox modification

Out of the box, the SUMO toolbox is used as a complete multi-parameter optimizer. It has a controller unit sitting at the highest level and manages the optimization process. Figure 2.2 describes the SUMO toolbox highlighting the control and optimization functions.

The controller manages the optimization process starting from a given initial dataset (i.e. initial sample points + outputs) and generates a surrogate model. The surrogate model approximates the dataset over the continuous design space range and is used by the optimizer instead of the dataset. Next, the controller predicts the next design space element from the constructed surrogate model at locations where the expected improvement is the largest, with the aim of further meeting the optimization's objective. The optimization process iterates until stopping conditions are met.

Normally, the SUMO toolbox works as a complete optimization solution when used out of the box. However, in the context of wireless testbeds, the SUMO toolbox has to be controlled by the underlying testbed management framework. This means the controller part of SUMO toolbox has to be replaced by the testbed's management framework and Figure 2.3 shows the modification and integration

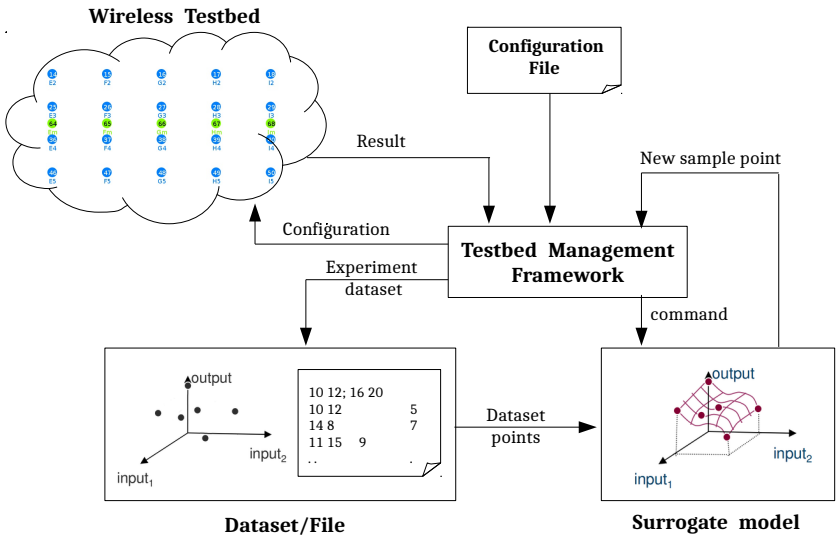


Figure 2.3: Integration of the modified SUMO toolbox in the wireless testbed

work.

This testbed management framework performs similar tasks as the original SUMO controller except for the addition of a number of tasks like experimentation on the wireless testbed, storing the dataset on a separate file, and reading the experiment configuration from a file.

2.4 Experimental Set-up

This section verifies, by integrating theoretical solutions from the previous section, the use of SUMO optimization toolbox in a wireless conference network problem using a wireless testbed. First we give a description of the wireless testbed where experimental set-up is carried out. Next the experiment scenario and the optimization processes are presented. Finally, we look in more detail at the performance objectives and discuss how conflicting performance objectives can be combined into a single objective.

2.4.1 Wireless testbed

The wireless *IMEC w-iLab.t* testbed, located at Zwijnaarde (Ghent, Belgium) [18], is shown in Figure 2.4. The testbed, equipped with heterogeneous wireless devices, is mainly used for wireless experimentation. It has 60 nodes each consisting

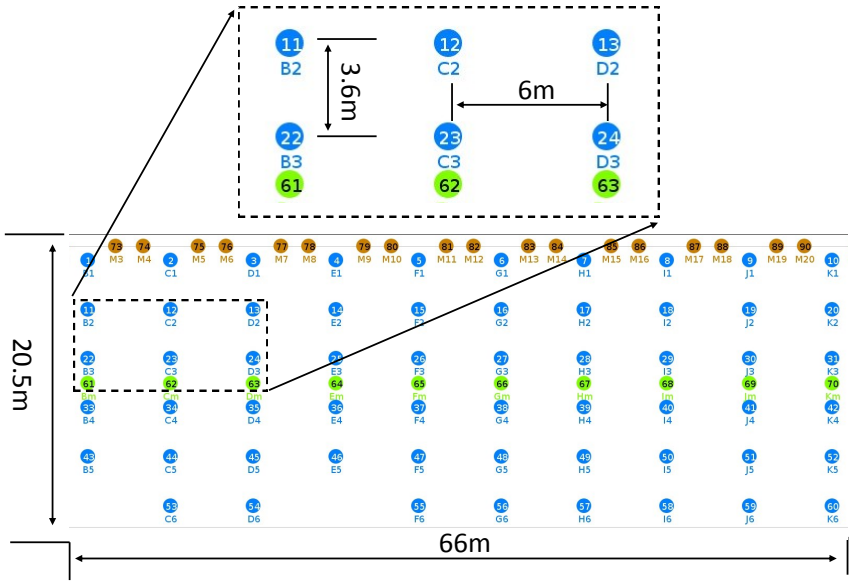


Figure 2.4: Top view of IMEC w-iLab.t wireless testbed

of an embedded Zotac PC having two Wi-Fi interfaces, a sensor node, a Bluetooth dongle and a wired control interface connected to the testbed management framework. Furthermore, the testbed is equipped with advanced spectrum sensing devices. These include Universal Software Radio Platform (USRP), IMEC Sensing Engines, and Wireless open Access Research Platform (WARP) boards. The wireless testbed is also equipped with mobile nodes mounted on Roomba robots allowing mobility experiments.

2.4.2 Experiment scenario

The SUMO optimizer is used to optimize a wireless conferencing scenario experimentally. Figure 2.5 shows the wireless conferencing scenario that comprises a wireless speaker broadcasting a speaker's voice over the air and multiple wireless microphones receiving the audio at the listener end. This type of wireless network is used in a multi-lingual conference room where the speaker's voice is translated into different languages and multiplexed into a single stream. Next, the stream is broadcasted to all listeners and each listener selects their preferred language.

Often, the speaker's audio quality is reduced by external interference and the surrounding environment is impacted by external interference. Thus, the main objective of the wireless conferencing scenario is to improve the received audio quality while keeping the transmission exposure at a minimum. To this end, the

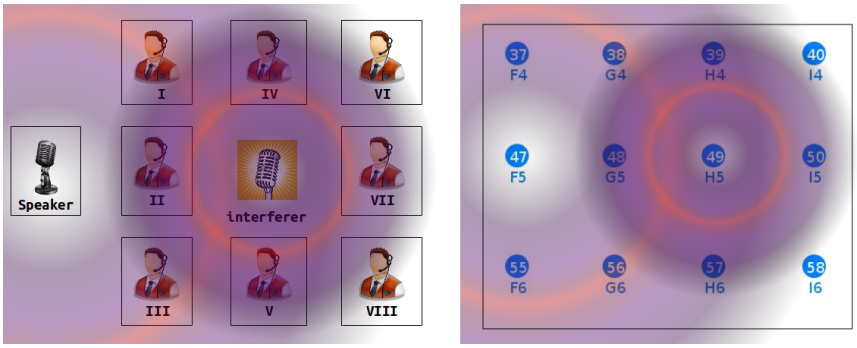


Figure 2.5: Left: wireless conferencing scenario consisting of 8 listeners, 1 speaker, and 1 interferer. Right: mapping of the conferencing scenario to the testbed nodes. The transmission range of the speaker and interferer is indicated.

conferencing operator has the possibility to adapt the speaker's channel and transmission power parameters.

The experiment is composed of 1 interferer creating background interference and a System Under Test (SUT) having 1 speaker and 8 listeners. The speaker broadcasts a 10s audio stream, obtained from ITU-T Test Signals for Telecommunication Systems [19], and each listener calculates the average audio quality within the time frame. The audio stream is encoded using A-Law encoding format at 64 Kbps bitrate. Moreover, the interferer transmits a 10 Mbps continuous UDP stream on dual channels (i.e. 1 and 13) generated using the iperf [20] application. The speaker, listeners and interference generator are shown in Figure 2.5.

On the left hand side of Figure 2.5, the realistic wireless conferencing scenario is shown, where as on the right hand side, the experimentation scenario is mapped on the IMEC w-iLab.t testbed. All listener nodes (i.e. 38, 39, 40, 48, 50, 56, 57, and 58) are associated to the speaker access point (i.e. node 47). Background interference is created by the access point (i.e. node 49) using two separate Wi-Fi cards. The Wi-Fi card and driver used for this experiment are "Atheros Sparklan WPEA-110N/E/11n mini PCI 2T2R" and "Atheros ath9k" respectively. The SUMO algorithms run on a dedicated PC that can communicate with all nodes of the experiment.

So far, we have investigated a simple scenario which is a speaker broadcasting 1 language stream to 8 listeners. Next, we looked two more scenarios to analyze the performance of SUMO optimization under loaded network condition and varied network topology. A network load can be induced by sending more traffic over the wireless medium (i.e. scenario II) and for this, the number of language streams is increased from 1 to 8 assuming multi-lingual listeners. A varied network topology, on the other hand, is created by changing the number of listeners from 8 to 16

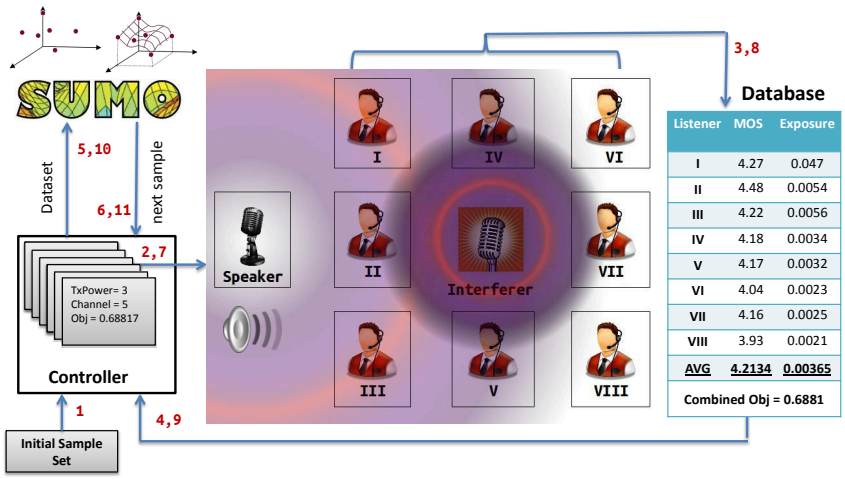


Figure 2.6: The process of SUMO optimization in the wireless conference network problem. The different sequential steps are numbered from 1 to 11.

using 1 language stream (i.e. scenario III). The physical data rate used in all cases is 1 Mbps using the Direct Sequence Spread Spectrum (DSSS) 802.11 modulation technique.

2.4.3 Optimization process

The optimization process is illustrated step by step in Figure 2.6. At (1) the controller is given a list of settings of the first experiments that needs to be configured on the wireless testbed. (2) Experiments are deployed on the wireless testbed using the requested settings, thus resulting in an initial sample set. (3) At the end of each experiment, the controller retrieves the evaluation criteria of the experiment. For the conferencing scenario, the evaluation criteria are the audio quality and exposure performances from all listeners. (4) An objective function is created by processing the evaluation criteria (see Section 2.4.4).(5) When the initial sample set is finished been experimented, the SUMO optimizer generates a surrogate model. (6) SUMO predicts the next sample point with the highest expected improvement. (7) The controller starts the next optimization experiment using the new design parameters. (8) Again, the evaluation criteria are retrieved and (9) the objective function is calculated for the new design parameters. (10) Based on the current dataset, extended by one record, the surrogate model is updated and (11) a new sample is predicted. The optimization process continues until stopping conditions are met.

2.4.4 Performance objectives

Dual objectives are applied in the wireless conference network problem. The first objective is maximizing the received audio quality which is measured using the Mean Opinion Score (MOS). MOS is a subjective audio quality measure represented on a 1 to 5 scale (i.e. 1 being the worst quality and 5 being the best quality). To calculate the MOS score, the experiment described in Section 2.5 uses the ITU-T Perceptual Evaluation of Speech Quality (PESQ) P.862 standard. It calculates the PESQ score out of packet loss, jitter and latency network parameters and maps it onto a MOS scale [21].

The second objective is minimizing transmission exposure. In [22] an in depth calculation of transmission exposure is presented. The exposure at a certain location is a combined measure of received power and transmit frequency. Transmission exposure is an important evaluation metric related to potential health issues, leading the regulatory bodies to set limits on maximum allowable radiation levels.

As maximizing the combined objective is the goal, the weight of performance metrics needs to be defined depending on the problem type. For example, a person who wants to install a wireless conferencing system in urban areas applies tighter exposure requirement than in rural areas. We would also apply high audio quality requirement in parliament auditoriums compared to office meeting rooms. However in our case, the aim is to apply the SUMO toolbox in a wireless system and we combined both metrics first by normalizing them to a [0 1] scale, followed by subtracting the two metrics into a [-1 1] scale and finally renormalizing the combined metric back to a [0 1] scale. To this end, the exhaustive search experiment is used to reference the normalization. Moreover, the combined metric is evaluated in each listener nodes and a representative combined metric is calculated first by averaging the combined metrics and next by selecting the one closest to the average.

2.5 Result and discussion

This section will analyze the viability and efficiency of using the SUMO global optimization technique for wireless experimentation. A methodology for detecting the validity of experiments by detecting outliers is described in section 2.5.1. The overhead of performing an exhaustive search is given in section 2.5.2, the outcome of which is used as a reference for experiment comparison. Experiment repeatability is discussed in section 2.5.3. The sensitivity of experiments to the choice of the initial sample size is discussed in section 2.5.4. Potential stopping criteria are analyzed in section 2.5.5. Finally, the SUMO optimized experiment is compared against the exhaustive search model in Section 2.5.6.

Figure 2.7: PRE and POST experiment monitoring.

2.5.1 Experiment outlier detection

Each experiment iteration has a chance of returning invalid measurement data. An experiment conducted at a certain time can show a different result when repeated at a later time. The most recurring reasons for this anomalous behavior are node malfunctioning and external interference. Since node malfunctions can be detected by most experimental testbeds, this section only focuses on methods to observe and handle experiment outliers due to external interference. This however does not mean interference generated inside SUT since it is part of the experiment.

In any wireless experiment, external interference (i.e. from an external device not participating within the experiment) competes for the scarce wireless medium and as such affects the behavior of the SUT. Ideally, experiments are performed in a controlled environment where unwanted external interference is blocked. However, this cannot be easily achieved, as shielding an experimental environment requires a great deal of money. As a result, many experimental facilities are currently installed in readily-accessible environments that serve multiple purposes, such as office buildings. For these areas, an alternative approach is to measure the effect of interference on an experiment. This approach does not block the interference but gives an interference score for each experiment conducted. Depending on the score, the experimenter decides whether or not to discard the experiment. One way of measuring external interference is by doing correlation measures on the experiment outcome [23]. Such measures correlate the output of identical experiment runs and discard those having lower correlation scores. The disadvantage of this method is that at least three experiment runs are required before getting a tangible result. A second approach for experiment outlier detection, shown in the Figure 2.7, is by doing PRE and POST experiment monitoring [24].

In this approach, the interference level is measured before the experiment starts (PRE) and only if the environment is clean that the experiment execution is triggered. After the experiment ends, the interference level is measured again (POST) to estimate a possible interference on the executed experiment. The main idea of this approach is that an experiment has a higher chance of being interfered by external interference if the environment is not clean during the PRE or the POST experiment.

The above two approaches give indirect indication of experiment interference levels. In contrast, a direct approach sniffs the wireless medium for external interference during experimentation. The level of difficulty imposed varies with the

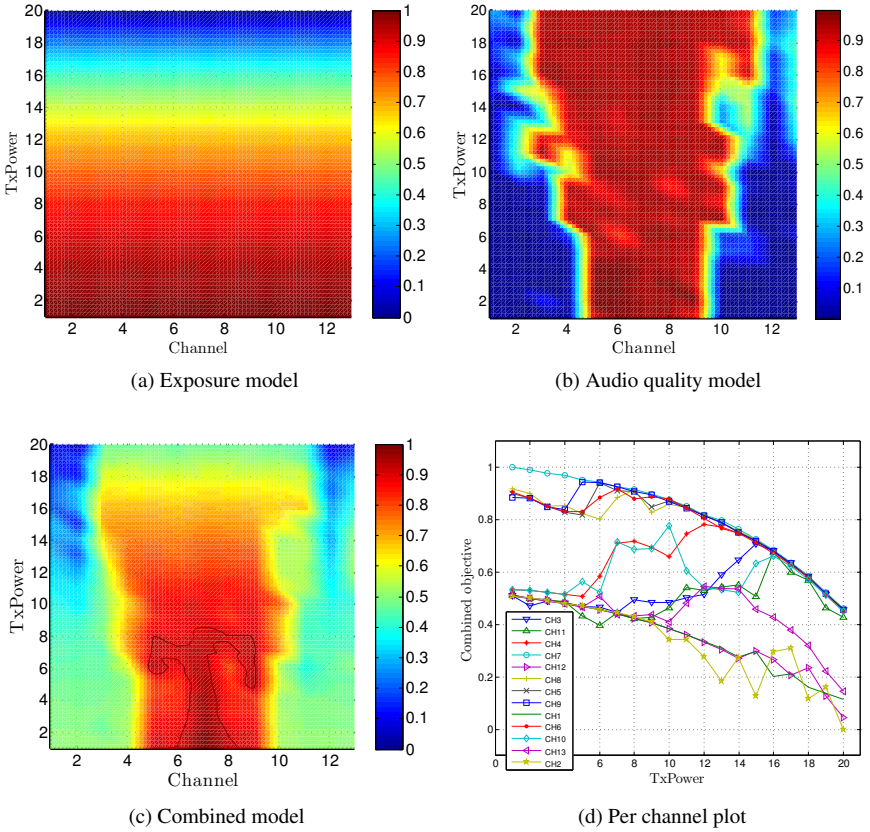
types of interference considered [25]. As the Industrial Science and Medical (ISM) band is a non-licensed band, a number of technologies may coexist together and the impact of external interference from such devices is difficult to characterize. As such, all these technologies need to be considered, thereby requiring a multi-layer (i.e. feature and energy detection) and multi-technology (i.e. Wi-Fi, Zigbee, Bluetooth, ...) distributed sensing solution [26].

Implementation of a distributed sensing solution was under way by the time of writing this paper. Hence we revert to a different approach by taking advantage of the pseudo-shielded nature of the IMEC w-iLab.t wireless testbed. The wireless environment is pseudo-shielded mounted on top of a clean room and experiments are guaranteed with clean environment from external (outside the wireless testbed) interference. Therefore, the PRE and POST experiment monitoring along with the pseudo-shielded wireless environment are used for the experiment outlier detection.

2.5.2 Exhaustive search model

In this section, we describe a reference experiment that was performed to generate an exhaustive search model of the wireless conference network problem. Neither SUMO nor any optimization algorithm is used to generate the model. The exhaustive search model evaluates all possible combinations of settings and will be used as a reference model for comparing SUMO optimization experiments. In total, 260 experiments (i.e. 13 Channels $\times$ 20 Transmit Power) were executed during which an interference is created continuously on dual channels (i.e.1 and 13). We start by making a complete analysis using the first scenario and later present the models of the remaining two scenarios (see Section 2.4.2).

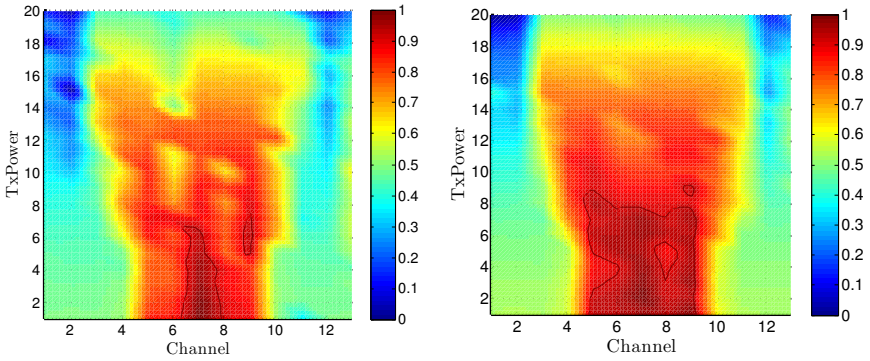
Figure 2.8 shows the outcomes of the exhaustive search model for the first scenario. The exposure model of Figure 2.8(a) only considers the exposure from the speaker but not from background interference, since the goal of the SUT is to reduce its own exposure. Moreover, exposure depends on the distance of the speaker and the average exposure over all listener nodes is calculated at the medium exposure point (Section 2.4.4). For the first scenario, the range of exposure values at the medium location is shown on the color bar of Figure 2.8(a). The exposure objective degrades with increased transmission power independent of the used channel. In contrast, the audio quality objective increases with increased transmission power and the influence of interference can be noted on multiple channels. There is an area on the non-interfered channels (i.e. 6 to 8) where adequate performance is observed also for lower transmit Power (i.e. 1dBm to 6dBm). This area is of interest because it represents a region where exposure is low. On the other hand, the worst performance from the audio quality model is shown between channels 2 to 4, 10 to 12 and transmit power 1 dBm to 7 dBm. Interestingly, this region



(a) Exposure model (b) Audio quality model

(c) Combined model (d) Per channel plot

Figure 2.8: Exhaustive search model. Background interference at channels 1 and 13.



(a) Scenario II (b) Scenario III

Figure 2.9: Exhaustive search model. Area inside the black contour is the Optimum region.

is not located on channels where background interference is applied on but on the neighboring channels. This is due to the fact that the speaker and interferer nodes apply CSMA/CA medium access on identical channels but to a limited scale on neighboring channels which results in degraded performance [26].

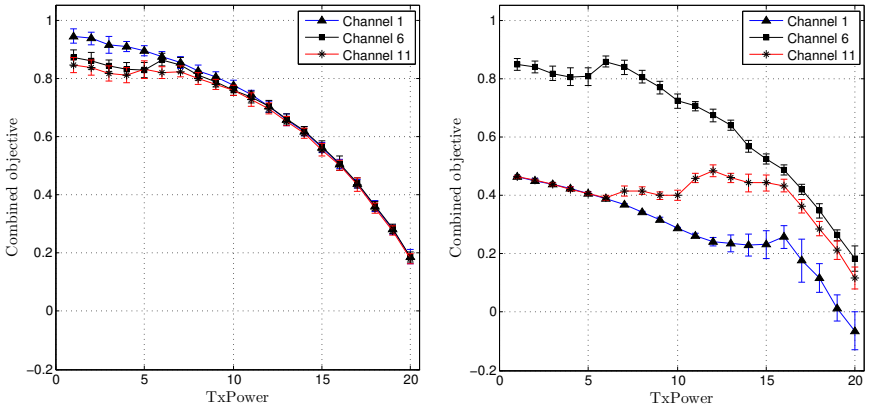
The combined objective model from Figure 2.8(c) is a combination of the exposure model and the audio quality model from Figure 2.8(a) and 2.8(b) respectively. As expected, the non-interfered regions with low transmission power have the highest values for the combined objective function. Figure 2.8(d) shows a different view of Figure 2.8(c) where the combined objective model is plotted for different transmission channels. Color bar shows the strength of the objectives in their respective limits where red indicates the best performance and blue indicates the worst performance.

Figure 2.9 shows the exhaustive models of scenarios II and III. Scenario II used the wireless medium intensely bringing a lot of agitation to the system and thus a smaller optimum region compared to scenarios I and III. On the other hand, scenario III used the same wireless medium as to scenario I but increased the number of listeners which stabilizes the average performance over the listener nodes and a larger optimum region as a result.

2.5.3 Experiment repeatability

A basic criterion for comparing wireless experiments is the requirement that experiments are repeatable. Identical experiments conducted in different time frames should show similar performance. One way of checking repeatability is by calculating the STandard Deviation (STD) of identical experiments and compare it with a threshold. Using scenario II (see Section 2.4.2), two sets of experiments were performed each on three different channels (i.e. 1, 6, and 11) and 35 repeated experiment runs were conducted. The first set considered a clean environment without background interference whereas the second set considered background interference at channels 1 and 13. Once again, a 10 second audio stream is transmitted by the speaker at different transmission power levels and listeners calculate the combined objective.

Figure 2.10 shows the experiment outcomes, using error bars to show the STD of the objective function on top of the average value. The lower the STD on the error bar, the higher the experiment repeatability. From all tests, the experiment of Figure 2.10(b) at channel 1 shows the worst repeatability with a decreasing trend as we increase the transmit power. A physical layer investigation between the speaker and interferer packets reveals that at lower transmit power, the interferer does not see packets coming from the speaker thus jamming the environment constantly assuming it is clean. Whereas for the speaker, it applies the CSMA/CA medium access method and it avoids the medium for most of the time or collides



(a) without background interference (b) with background interference

Figure 2.10: Repeatability test at Wi-Fi channels 1, 6, and 11

with the interferer in case it transmits. This increases the number of lost packets and eventually to a very low audio quality for the listener nodes. Because of the loss in audio quality and the same exposure at fixed power levels, the combined objective at a listener node does not show variation with repeated experiments. However, when we increase the transmit power, the interferer feels the presence of the speaker and it starts applying the CSMA/CA medium access method before transmitting its packets. This lets the speaker to transmit without being interfered and the listeners to receive a higher audio quality. Since the medium is now shared by the two transmitters, the audio quality starts fluctuating depending on the time share the speaker has possessed during the experiment. This creates the difference in the combined objective and reduces repeatability between repeated experiments. On the other hand, experiments on channels 1 and 11 from Figure 2.10(b) show similar repeatability trend except on a reduced scale at channel 11. On channel 11, the interferer overlaps part of the spectrum with the speaker. Thus the same principle applies to the experiment on channel 1 and shows decreasing repeatability as transmit power increases. However, due to the fact that CSMA/CA is used on a limited scale on neighboring channels [26], repeatability at channel 11 is slightly better than at channel 1. On the other hand, the experiment of Figure 2.10(b) at channel 6 has minimal background interference, hence the experiment behaves similar to the case without interference.

So far we have analyzed the repeatability test with and without interference. But the question we need to answer should be, is the wireless environment repeatable? Since any wireless environment can not be 100% repeatable, we have to leave a certain margin depending on the problem type. This margin depends on

the dynamics of the wireless environment but addressing it requires state-of-the-art ideas and tools. Instead we revert to a different approach by comparing repeatability at its worst condition (i.e. with a highest co-channel interference on SUT) and at its best condition (i.e. with no interference on SUT). This comes down to comparing the repeatability variation of Figure 2.10(a) and 2.10(b) and the best case variation (i.e. STD = 0.0301) is smaller than the worst case variation (i.e. STD = 0.0747) and that proves the repeatability test.

2.5.4 Initial sample size sensitivity

As explained in Section 2.4.3, a surrogate model predicts the next experiment input parameters with a highest expected output performance. However, the initial model requires a set of initial sample points from the design space and performance outputs. This section investigates how many initial samples are required before a usable surrogate model can be created.

The initial sample points for any problem have to be selected carefully such that the optimization process quickly converges to the optimum. If the number of initial sample points is large, the optimizer spends too much time during exploration work. On the other hand, considering few initial sample points leads to the risk of missing global optimums and thus exploiting local optimums instead. One way to address the trade-off between exploration and exploitation during optimization is by selecting an appropriate initial sample size. Usually this depends on the complexity of a problem's global model. The more complex a problem's global model is, the larger the initial sample size needed to have good surrogate model approximation and vice versa. It was indicated in [27] that extreme points of a surface can be used to measure the complexity of a problem. These are the minimums, maximums and saddle points of a problem's global model. Moreover, it is also indicated that by setting the initial sample size to the number of extreme points, an optimizer has a higher chance to arrive at the global optimum in short amount of time. This assumption only works if the problem's extreme points are known beforehand. Most of the time this is not the case as we generally optimize unknown problems. Moreover, initial sample size selection depends on the problem type [27]. For our specific problem, setting the initial sample size to 8 points is found a good choice. The 8 initial sample points together with corner points, sums up to 12 initial points in total.

In the following sections, we will each time analyze four different sampling methods to pick the 12 initial sample points from the design space. These are

- Latin Hypercube Sampling (LHS) [28], which is a stratified sampling method that selects sample points evenly along the design space while ensuring proportional representation of design variables.

- Orthogonal sampling, which divides the design space into a number of sub-spaces and LHS is applied in each sub-space.
- Random sampling, which selects points randomly over the design space.
- Hammersley Sequence Sampling (HSS) [29], which is a low-discrepancy quasi-random sampling method providing better uniformity properties and uniform distribution of points in space.

2.5.5 Stopping criteria

The main goal of the global optimization is to reduce the number of required experiments. This section investigates the effect that different stopping criteria have on the problem's optimum value. In this paper, we look at two stopping criteria namely Fixed Iteration (FI) and Objective Function Improvement (OFI). With the FI stopping criterion, a fixed number of iterations are conducted and the optimum value from the output is selected. On the other hand, the OFI stopping criterion looks at the relative difference in performance and stops the iteration when the STD of the top sorted N iterations falls below a given threshold. The idea behind the OFI stopping criterion is that, the sorted objective function of a list of experiments ideally approaches a flat curve as the number of experiments increases.

To perform sensitivity analysis, experiments are conducted using the SUMO toolbox until the stopping criterion is met. Using scenario I (Section 2.4.2), a plot of normalized combined objectives as a function of iteration count is shown on Figure 2.11. Among the different design space sampling methods applied, the LHS method reaches the Global Maximum Combined Objective (GMCO) first (after iteration 9) whereas the Random sampling method arrives last (iteration 25). This, however, does not mean LHS is preferable for all problem types but for the current problem, it approximates the global model better than any other sampling method. On the other hand, we see the plots of the three experiments (i.e. ORTH, HSS, RAND) not reaching the GMCO, and this is due to a small repeatability variation we saw on the Figure 2.10(b) at channel 6.

FI stopping criterion sets one parameter which is the number of iterations an experiment needs to execute. It is clear from Figure 2.11 that it is difficult to draw a conclusion about the number of iterations since the iteration count of the different sampling methods to reach the GMCO is highly variable. As such, the FI stopping criterion is mainly useful for time-constrained testbeds where experimenters can only reserve resources for a limited time. It can be used as an upper limit in case all other stopping criteria fail to satisfy.

On the other hand, the OFI stopping criterion, as stated previously, considers the relative performance difference between experiments and stops the iteration when the STD of the top sorted N iterations falls below a given threshold. The OFI

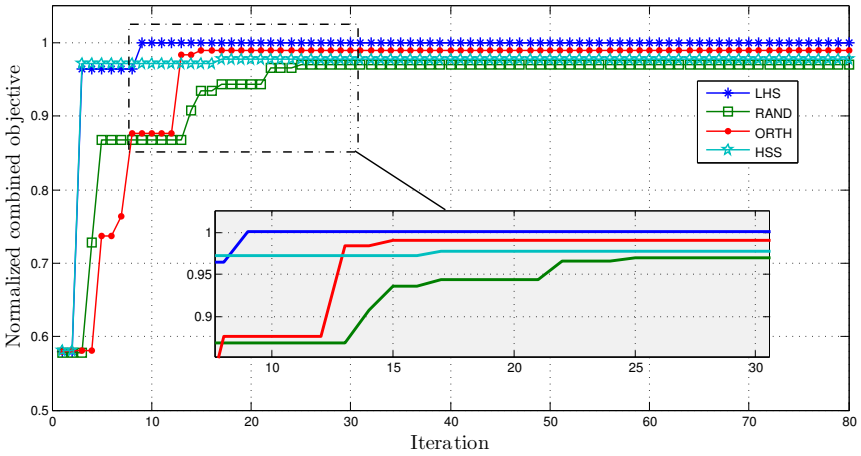
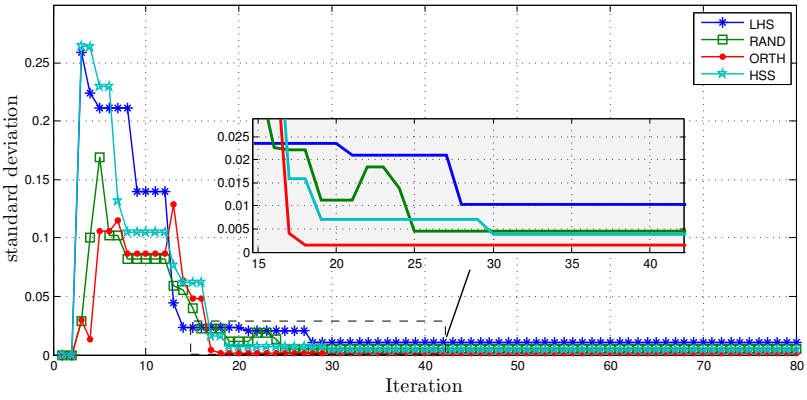


Figure 2.11: Scenario I normalized combined objective as a function of experiment iteration

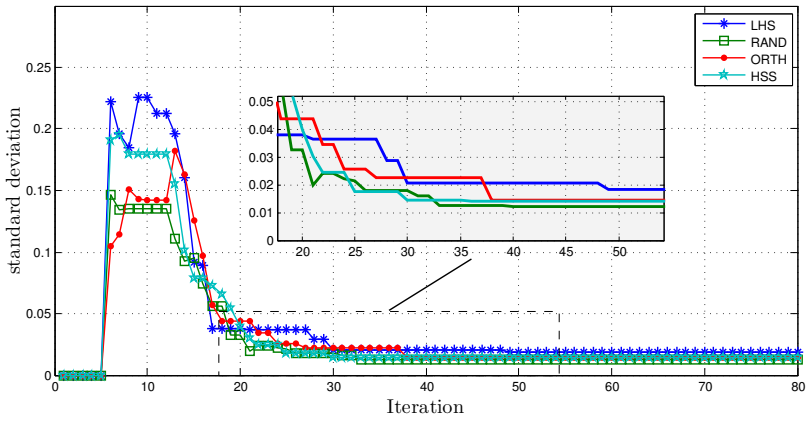
stopping criterion has two parameters to set. These are the STandard Deviation WIDTH (STD-WIDTH) which sets the number of objective performance values in the STD calculation and the STandard Deviation THreshoLD (STD-THLD) which is used as a lower limit for the stopping criterion. Figure 2.12 shows the STD curve of scenario I (section 2.4.2) as a function of iteration count for STD-WIDTH 3, 6 and 10. These numbers are wide enough to show the behavior and variation of different STD curves. Calculation starts after the iteration count reaches STD-WIDTH.

As stated previously, the output of the plots for each STD width approaches a flat curve when the optimization reaches the optimum. On the other hand, the randomness of the curves gradually decreases as the STD-WIDTH increases. This also increases the settling time until the lowest STD value is reached. For example, looking the LHS experiment from Figure 2.12, the settling times for the three STD-WIDTH parameters 3, 6 and 10 are 28, 49 and 62 iterations respectively. Also note the benefit of the SUMO optimization with a sharp declining curve after the 12 initial experiments. As the optimization continues, the STD curve starts decreasing and converges to a stable value.

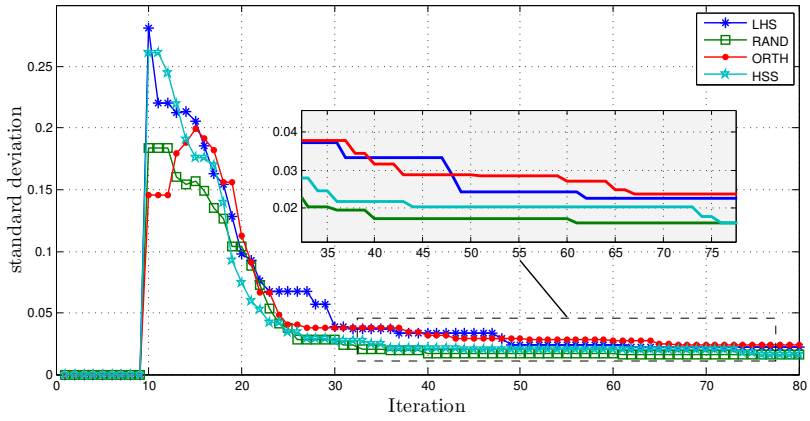
There are two things we want the STD curve to achieve. First, we want the curve to reach a stable value as fast as possible. This depends on the size of the optimum region in the problem's global model. The optimum regions of all scenarios are shown as a black contour on top of the exhaustive search model (i.e. Figures 2.8(c), 2.9(a) and 2.9(b)). The larger this area, the sooner the optimization locates the optimum and the STD curve converges to a stable value and vice versa (see next section). However, the size of a problem's optimum region is not known



(a) STD-WIDTH=3

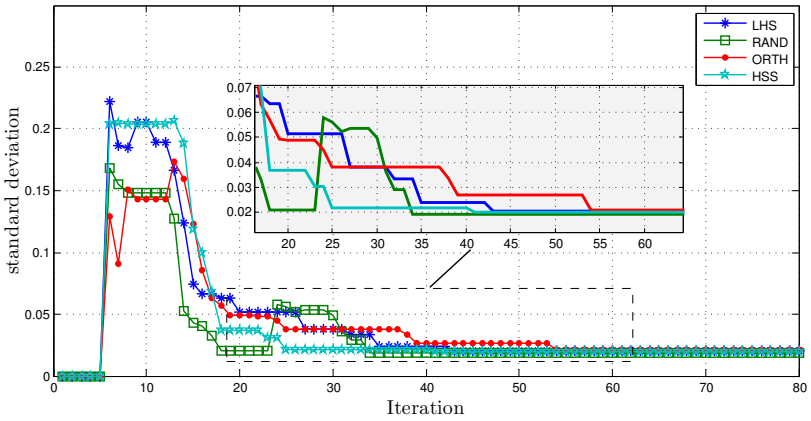


(b) STD-WIDTH=6

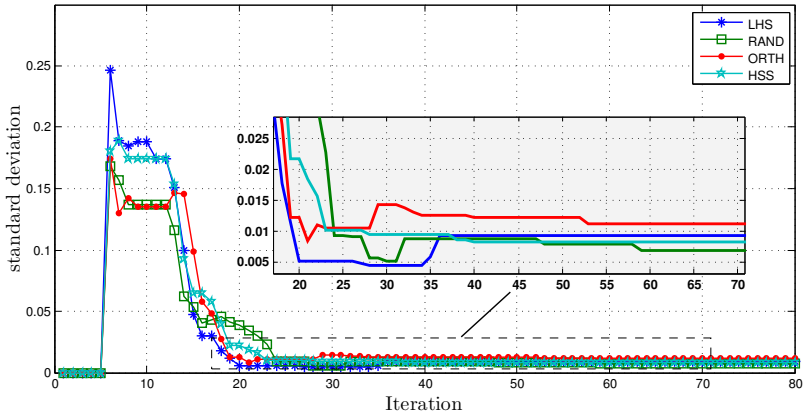


(c) STD-WIDTH=10

Figure 2.12: Scenario I standard deviation as a function of experiment iteration



(a) STD-WIDTH=6



(b) STD-WIDTH=6

Figure 2.13: Scenario II and III standard deviation as a function of experiment iteration

beforehand and a good value of STD-WIDTH, in such cases, is half the elements of the initial sample size. In our case, STD-WIDTH will be 6 (i.e. 12/2 = 6).

Second, we want the curve to reach a very small stable value. Again, this value never approaches to zero as the wireless medium shows a small variation. Since the data points used in the STD calculation after it gets stable are inside the optimum region, STD-THLD can assume the maximum STD where repeated experiments show over the optimum region. Again, the optimum region is not known before hand and the work around is to perform repeatability tests without background interference (i.e. Figure 2.10(a)) and select the maximum value from the list (i.e. scenario I = 0.02418, II = 0.02865 and III = 0.02067).

The STD performance for scenarios II and III are shown in Figure 2.13.

2.5.6 Performance comparison

Now the stopping criteria and initial sample size are selected and experiment repeatability is verified. Next, we compare the SUMO approach to the traditional experimentation that exhaustively searches all parameters. For the comparison, we have defined the parameters of the OFI stopping criterion to the following: STD-WIDTH = 6, STD-THLD1 = 0.02418, STD-THLD2 = 0.02865 and STD-THLD3 = 0.02067. Table 2.2 shows the performance metrics of each conducted experiment for the three scenarios when these parameters are applied. The four different sampling methods from Section 2.5.4 are also included. The required *number of iterations*, before the stopping conditions are met, are shown for each of the sampling methods. The *Duration Gain* metric calculates the rate by which SUMO experiment duration is reduced compared to the exhaustive search experiment that took 260 experiments. The *Performance Gain* metric evaluates how close the optimum solution of the SUMO experiment is to the GMCO value.

When comparing all scenarios from Table 2.2, LHS is found the best sampling method and RAND is the worst sampling method in terms of performance gain. The LHS sampling method almost all the time achieved the GMCO value of the problem but for the RAND sampling method, depending on the scenario, it shows a wide performance variation. On the other hand, RAND sampling method converges the quickest in the first two scenarios but at the expense of a lower performance gain (scenario I = 89.15% and II = 78.76%). This is because of poor initial samples are explored and it leads to a local optimum instead of a global optimum. Looking the RAND curve of Figure 2.13(a) in particular, reveals this finding by having bumps along the curve (iteration 2334), had we continue the optimization. The other finding discerned from Table 2.2 is that the number of iterations generally decreases with an increase in the problem's optimum region. Scenario III, having the largest optimum region, has the smallest number of iterations followed by scenario I and II. On the other hand, when looking at the different sampling

Table 2.2: Duration Gain and Performance Gain of SUMO optimized experiments using 4 sampling methods

(a) Scenario I

Sampling Method	No. of iterations	Duration Gain	Performance Gain
LHS	30	260/30=8.67	3.9398/3.9480=99.79%
RAND	21	260/21=12.38	3.5198/3.9480=89.15%
ORTH	27	260/27=9.63	3.8663/3.9480=97.93%
HSS	25	260/25=10.4	3.7708/3.9480=95.51%

(b) Scenario II

Sampling Method	No. of iterations	Duration Gain	Performance Gain
LHS	35	260/35=7.42	3.9139/3.9192=99.86%
RAND	18	260/18=14.44	3.0869/3.9192=78.76%
ORTH	39	260/39=6.67	3.9102/3.9192=99.77%
HSS	25	260/25=10.4	3.7749/3.9192=96.32%

(c) Scenario III

Sampling Method	No. of iterations	Duration Gain	Performance Gain
LHS	18	260/18=14.44	3.5419/3.6803=96.24%
RAND	24	260/24=10.834	3.4565/3.6803=93.92%
ORTH	19	260/19=13.68	3.3069/3.6803=89.85%
HSS	21	260/21=12.38	3.4926/3.6803=94.89%

method performances, LHS and ORTH show similar performance in both metrics and in all scenarios. This is because both are using latin hypercube sampling and so does their similarity on performance. In addition, both LHS and ORTH are the most sensitive sampling methods to a change in the optimum region.

2.6 Conclusion

This paper investigated the feasibility of the SUMO optimizer when used in experimental optimization of wireless solutions. In particular, a wireless conferencing scenario is considered. This paper also described the integration work of the SUMO optimizer in the IMEC w-iLab.t wireless testbed.

To compare the efficiency of SUMO optimized experiments, an exhaustively searched experiment is first conducted which leads to an accurate model of the problem to be optimized. However experiment repeatability needs to be guaranteed before comparison. To this end, identical experiments both with and without interference are conducted to validate this property. Moreover, experiments might become invalid due to external interference and an experiment outlier detection is applied to check validity of each conducted experiment.

SUMO is a powerful optimizer but a number of configurable parameters affect its efficiency. The sensitivity to initial sample size and the effect of stopping criteria are investigated in this paper. The initial sample size sensitivity exploits the exploration and exploitation balance of an optimization problem such that with few initial samples, an optimizer locates the optimum in a short period of time. Next, the Fixed Iteration (FI) and Objective Function Improvement (OFI) stopping criteria are considered. The FI stopping criterion is found not suited for our problem as it shows a wide variation in iteration count to reach the Global Maximum Combined Objective (GMCO). On the other hand, the OFI stopping criterion is well suited since it considers a relative difference in combined objective performance. Four sampling methods (Latin Hypercube Sampling, Random sampling, Orthogonal sampling and Hammersley Sequence Sampling) were combined with the SUMO toolbox to optimize the experiment until the OFI stopping criteria is met. The experiment is grouped into three scenarios. Scenario I realized the simplest wireless conferencing system where a speaker streamed one language to 8 listener nodes. Scenario II utilized the wireless spectrum intensely by increasing the language count from 1 to 8. And scenario III varied the network topology by increasing the listeners count from 8 to 16. In our proof of concept, the SUMO experiment with LHS sampling method arrived the GMCO value at least 7.42 times faster than the exhaustive search experiments in all scenarios.

Acknowledgment

The research leading to these results has received funding from the European Union's Seventh Framework Programme FP7/2007-2013 under Grant agreement No. 258301 (CREW Project www.crew-project.eu) and 318273 (LEXNET Project www.lexnet-project.eu). This work was also supported by the Interuniversity Attraction Poles Programme BESTCOM initiated by the Belgian Science Policy Office.

References

[1] D. Cavin, Y. Sasson, and A. Schiper. *On the Accuracy of MANET Simulators*. In Proceedings of the Second ACM International Workshop on Principles of Mobile Computing, POMC '02, pages 38–43, New York, NY, USA, 2002. ACM. Available from: http://doi.acm.org/10.1145/584490.584499, doi:10.1145/584490.584499.

[2] K. Tan, D. Wu, A. Chan, and P. Mohapatra. *Comparing simulation tools and experimental testbeds for wireless mesh networks*. In World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a, pages 1–9, 2010. doi:10.1109/WOWMOM.2010.5534917.

[3] T. Rakotoarivelo, G. Jourjon, and M. Ott. *Technical Report: Designing and Orchestrating Reproducible Experiments on Federated Networking Testbeds*. Technical report, NICTA, Sydney, Australia, July 2012.

[4] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq. *A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design*. J. Mach. Learn. Res., 11:2051–2055, August 2010. Available from: http://dl.acm.org/citation.cfm?id=1756006.1859919.

[5] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq. *A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design*. J. Mach. Learn. Res., 11:2051–2055, August 2010. Available from: http://dl.acm.org/citation.cfm?id=1756006.1859919.

[6] D. Jones, M. Schonlau, and W. Welch. *Efficient Global Optimization of Expensive Black-Box Functions*. Journal of Global Optimization, 13(4):455–492, 1998. Available from: http://dx.doi.org/10.1023/A%3A1008306431147, doi:10.1023/A:1008306431147.

[7] C. E. S. ANIL K. GUPTA, KEN G. SMITH. *THE INTERPLAY BETWEEN EXPLORATION AND EXPLOITATION*. Academy of Management Journal, 49:693–706, 2006.

[8] G. Hawe and J. Sykulski. *Balancing Exploration Exploitation using Kriging Surrogate Models in Electromagnetic Design Optimization*. In Electromagnetic Field Computation, 2006 12th Biennial IEEE Conference on, pages 229–229, 2006. doi:10.1109/CEFC-06.2006.1633019.

[9] M. Holland, T. Wang, B. Tavli, A. Seyedi, and W. Heinzelman. *Optimizing Physical-layer Parameters for Wireless Sensor Networks*. ACM Trans. Sen. Netw., 7(4):28:1–28:20, February 2011. Available from: http://doi.acm.org/10.1145/1921621.1921622, doi:10.1145/1921621.1921622.

[10] Y. Qu and S. Georgakopoulos. *Relocation of wireless sensor network nodes using a genetic algorithm*. In Wireless and Microwave Technology Conference (WAMICON), 2011 IEEE 12th Annual, pages 1–5, 2011. doi:10.1109/WAMICON.2011.5872882.

[11] R. Kulkarni and G. Venayagamoorthy. *Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey*. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 41(2):262–267, 2011. doi:10.1109/TSMCC.2010.2054080.

[12] X. Gao and Y. Gao. *TDMA Grouping Based RFID Network Planning Using Hybrid Differential Evolution Algorithm.* In F. L. Wang, H. Deng, Y. G. 0001, and J. Lei, editors, AICI (2), volume 6320 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 2010. Available from: http://dblp.uni-trier.de/db/conf/aici/aici2010-2.html#GaoG10.

[13] K. Kaur, M. Rattan, and M. S. Patterh. *Optimization of Cognitive Radio System Using Simulated Annealing*. Wirel. Pers. Commun., 71(2):1283–1296, July 2013. Available from: http://dx.doi.org/10.1007/s11277-012-0874-1, doi:10.1007/s11277-012-0874-1.

[14] I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, and L. Knockaert. *Surrogate-based infill optimization applied to electromagnetic problems*. INTERNATIONAL JOURNAL OF RF AND MICROWAVE COMPUTER-AIDED ENGINEERING, 20(5):492–501, 2010. Available from: http://dx.doi.org/10.1002/mmce.20455.

[15] P. H. Reisenthel and D. J. Lesieutre. *Statistical Benchmarking of Surrogate-Based and Other Optimization Methods Constrained by Fixed Computational Budget*. In 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 18th, 51st Structural Dynamics, and Materials Conference, 12 - 15 April 2010, Orlando, Florida.

[16] N. Hansen and A. Ostermeier. *Completely Derandomized Self-Adaptation in Evolution Strategies*. Evol. Comput., 9(2):159–195, June 2001. Available from: http://dx.doi.org/10.1162/106365601750190398, doi:10.1162/106365601750190398.

[17] Q. Zhou, P. Z. G. Qian, and S. Zhou. *A Simple Approach to Emulation for Computer Models With Qualitative and Quantitative Factors*. Technometrics, 53(3):266–273, 2011. Available from: http://amstat.tandfonline.com/doi/abs/10.1198/TECH.2011.10025, arXiv:http://amstat.tandfonline.com/doi/pdf/10.1198/TECH.2011.10025, doi:10.1198/TECH.2011.10025.

[18] S. Bouckaert, P. Becue, B. Vermeulen, B. Jooris, I. Moerman, and P. Demeester. *Federating Wired and Wireless Test Facilities through Emulab and OMF: The iLab.t Use Case*. In T. Korakis, M. Zink, and M. Ott, editors, Testbeds and Research Infrastructure. Development of Networks and Communities, volume 44 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 305–320. Springer Berlin Heidelberg, 2012. Available from: http://dx.doi.org/10.1007/978-3-642-35576-9_25, doi:10.1007/978-3-642-35576-9_25.

[19] I. telecommunication. *ITU-T Test Signals for Telecommunication Systems*. http://www.itu.int/net/itu-t/sigdb/genaudio/AudioForm-g.aspx?val=1000050. Last accessed Feb 27, 2014. Available from: http://www.itu.int/net/itu-t/sigdb/genaudio/AudioForm-g.aspx?val=1000050.

[20] NLANR/DAST. *iperf: TCP and UDP bandwidth performance measurement tool*. https://code.google.com/p/iperf/, May 2009. Last accessed Feb 27, 2014.

[21] N. K. Base. *How is MOS calculated in PingPlotter Pro?* http://www.nessoft.com/kb/50, November 2005. last accessed Feb 27, 2014.

[22] D. Plets, W. Joseph, K. Vanhecke, and L. Martens. *Exposure optimization in indoor wireless networks by heuristic network planning*. Progress In Electromagnetics Research, 139:445–478, 2013.

[23] S. Keranidis, W. Liu, M. Mehari, P. Becue, S. Bouckaert, I. Moerman, T. Korakis, I. Koutsopoulos, and L. Tassiulas. *Concrete: A benchmarking framework to control and classify repeatable testbed experiments*. In FIRE Engineering Workshop, Abstracts, 2012.

[24] S. Bouckaert, M. Mehari, W. Liu, I. Moerman, P. V. Wesemael, D. Finn, M. Chwalisz, J. Hauer, M. Doering, N. Michailow, D. Depierre, C.Heller, M. Smolnikar, Z. Padrah, and M. Vucnik. *Methodology for performance evaluation*. www.crew-project.eu/sites/default/files/CREW_D4.2_-IBBT_R_PU_2012-09-30_final.pdf, sep 2010. section 3.3. Available from: http://www.crew-project.eu/sites/default/files/CREW_D4.2_IBBT_R_PU_2012-09-30_final.pdf.

[25] Wikipedia. *Electromagnetic interference at 2.4 GHz*. http://en.wikipedia.org/wiki/ Electromagnetic_interference_at_2.4_-GHz, August 2008. Last accessed Feb 27, 2014. Available from: http://en.wikipedia.org/wiki/Electromagnetic_interference_at_2.4_GHz.

[26] W. Liu, S. Keranidis, M. Mehari, J. Vanhie-Van Gerwen, S. Bouckaert, O. Yaron, and I. Moerman. *Various Detection Techniques and Platforms for Monitoring Interference Condition in a Wireless Testbed.* In L. Fbrega, P. Vil, D. Careglio, and D. Papadimitriou, editors, Measurement Methodology and Tools, volume 7586 of *Lecture Notes in Computer Science*, pages 43–60. Springer Berlin Heidelberg, 2013. Available from: http://dx.doi.org/10.1007/978-3-642-41296-7_4, doi:10.1007/978-3-642-41296-7_4.

[27] Z. Zhou. *Two-phase IMSE-optimal Latin hypercube design for computer experiments*. PhD thesis, University of Southern California, 2006.

[28] F. A. C. Viana. *Things you wanted to know about the Latin hypercube design and were afraid to ask*. 10th World Congress on Structural and Multidisciplinary Optimization, May 19 -24 2013.

[29] W.-S. L. Tien-Tsin Wong and P.-A. Heng. *Sampling with Hammersley and Halton Points*. Journal of Graphics Tools, 2(2):9–24, 1997.

3

Efficient Identification of a Multi-Objective Pareto Front on a Wireless Experimentation Facility

In this chapter, we further improve upon the surrogate modeling based optimization strategy discussed in chapter 2 and demonstrate that the proposed approach also scales towards more complex networks. To this end, we consider (i) a more complex scenario, (ii) more configuration parameters, (iii) advanced performance objectives, (iv) multi-objective optimization, and (v) pareto front analysis.

⋆ ⋆ ⋆

Michael Tetemke Mehari, Eli De Poorter, Ivo Couckuyt, Dirk Deschrijver, Gunter Vermeeren, David Plets, Wout Joseph, Luc Martens, Tom Dhaene, Ingrid Moerman

Published in IEEE Transactions on Wireless Communications Volume 15 Issue 10, Oct 2016.

Abstract Wireless systems often need to optimize multiple conflicting objectives (low delay, high reliability, low cost) which are difficult to fulfill simultaneously. In such cases, the wireless system exhibits multiple optimal operation points, referred to as the Optimal Pareto Front (OPF). However, due to the large number

of parameter settings to be evaluated and the time-consuming nature of performing wireless experiments, it is typically not possible to identify the OPF by exhaustively evaluating all possible settings. Instead, for many use cases an approximation is good enough. To this end, this paper applies a Multi Objective Surrogate-Based Optimization (MOSBO) toolbox to efficiently optimize wireless systems and approximate the OPF using a limited number of iterations. Moreover, a real Wi-Fi conferencing scenario is optimized that has two conflicting objectives (exposure and audio quality) and 4 configurable parameters (Tx-Power, Tx-Rate, Codec Bit-Rate, Codec Frame-Length). The benefits of using the MOSBO approach for such a network problem is demonstrated by approximating the OPF using 94 iterations instead of requiring the exploration of 6528 different parameter combinations, while still dominating 96.58% of the complete design space.

3.1 Introduction

The introduction of wireless systems, replacing the legacy wired systems, created a wide range of opportunities. Nowadays, industrial environments are equipped with wireless sensors in places where it is difficult to put wired connections for temperature and pressure readings. It is also very cheap to deploy cellular networks these days rather than installing expensive copper wirings at customers' premises. Such application areas are pushing the need for wireless systems which otherwise are difficult to be realized by wired systems or impractical in some areas.

On the other hand, the opportunities that were envisioned in wireless systems are challenged by the need to optimize multiple conflicting objectives. Wireless surveillance systems, for example, target the highest video quality whilst utilizing a minimal bandwidth. Wireless conferencing systems also strive for realizing the best audio quality but simultaneously aim to limit their wireless exposure. Another example is Automated Guided Vehicles (AGVs) working in a large factory hallway. While robustness is typically a first priority, communication between AGVs needs to be secure which also demands higher network utilization during frequent roaming between access points. Such wireless systems, showing conflicting objectives, exhibit multiple non-dominated operating points (i.e. parameter settings) which in literature is referred to as the Optimum Pareto Front (OPF) [1], [2]. The OPF is a set of performance objectives which cannot be further improved by any other parameter combinations without affecting at least one objective. For designers of wireless systems, it is typically not possible to identify the OPF because it requires an exhaustive search of the parameter space which tends to be time intensive and sometimes impossible. Instead, an approximation is good enough. To this end, an Approximate Pareto Front (APF) [2] is calculated using a number of multi-objective optimizers such as the Non-dominated Sorting Genetic Algorithm II (NSGA-II [3]), the S-Metric Selection Evolutionary MultiObjective Algorithm

(SMS-EMOA [4]) and the Strength Pareto Evolutionary Algorithm 2 (SPEA2 [5]).

Even though all of these are targeted at multi-objective optimization, they are typically not designed to minimize the number of iterations. In fact, most evolutionary based multi-objective variants (i.e. NSGA-II, SPEA2, SMS-EMOA, ...) inherently require a large number of iterations to identify the APF. On the other hand, evaluating real-life wireless systems is time-intensive since each experiment requires resource deployment, configuration, execution and evaluation of the wireless experiment. For example, when using the Orbit Management Framework (OMF) for experimentation control, an experiment having N wireless nodes adds an average delay of 5.17*N ms on a single message orchestration [6]. As such, most designers of wireless systems aim to approximate the OPF using a limited number of experiments, specially when relying on real-life deployments rather than simulations. As a solution, this paper considers a Multi Objective Surrogate-Based Optimization (MOSBO) approach [7]. MOSBO makes use of Kriging models and Hypervolume based Probability of Improvement (HV-PoI) to economize experiment runs and approximate the OPF using a limited number of experiments.

The paper introduces the following novel contributions

– An architecture for integrating a MOSBO optimizer in realistic conditions for approximating the OPF of complex wireless systems using a limited number of experiments

– The introduction of advanced optimization objectives, including (i) an advanced calculation of end-to-end audio quality Mean Opinion Score (MOS) and (ii) an exposure Specific Absorption Rate (SAR) metric for Wi-Fi traces.

– An analysis of the influence of different MOSBO design criteria, such as the stopping criteria and initial sample size, on the overall performance of the system.

– An experimental validation of the overall multi-objective optimization through a large-scale Wi-Fi conferencing system using a wireless testbed facility.

The remainder of this paper is structured as follows. Section 3.2 presents a literature survey of multi-objective optimization solutions in wireless systems. Next, the MOSBO optimizer is presented in Section 3.3. Section 3.4 introduces a Wi-Fi conferencing scenario which is used to test the MOSBO optimizer. The results from the Wi-Fi conferencing experiments are discussed in Section 3.5. Finally, conclusion and future work are presented in Section 3.6.

3.2 Related work

In Section 3.4, a Wi-Fi conferencing scenario will be discussed in order to optimize two conflicting objectives: improving audio quality (e.g. using MOS score) and reducing the transmission exposure (expressed as SAR). This section will discuss aspects related to the conflicting objectives that will be investigated as well as state-of-the-art in multi-objective optimization of wireless networks.

3.2.1 Electromagnetic exposure

Due to the increased use of wireless technologies, an increasing amount of attention is given to the impact of electromagnetic radiation on the human body. In exposure assessment, one distinguishes between (i) compliance testing and (ii) realistic exposure assessment. The former evaluates if worst-case exposure situations comply with exposure limits and is either uplink (transmission from a wireless end device to a base station) or downlink (transmission from a base station down to a wireless end device) focused. The exposure metrics used in compliance testing are incident electric and magnetic field, incident power density, SAR, etc. The latter evaluates the exposure of a person under realistic exposure conditions which is of interest in epidemiological studies and often combines both uplink and downlink exposures together. For analyzing and optimizing wireless deployments, the realistic exposure assessment is important [8], especially when co-optimizing the exposure with other performance criteria. In [9], the authors address this concern by creating a network planning tool to jointly optimize transmission exposure and coverage. Furthermore, the author in [10] decomposes exposure into uplink and downlink and applied a joint minimization to lower the total human exposure dose. A more realistic approach by using a wireless testbed is proposed by [8]. The paper evaluated the Exposure Index of an LTE data scenario implemented in a real urban area (part of the 7th district of Paris) and quantifies the total exposure of a population in the area.

3.2.2 Audio quality

In contrast to methods for calculating the exposure in Wi-Fi networks, the field of calculating audio quality objective is rather mature and sufficient research has already been carried out. One way of improving the audio quality in streaming applications is by allowing dynamic source rate adaptation. The authors in [11] presented this concept such that by using RTCP receiver reports to understand the Network condition, the bandwidth of the source audio is adjusted and the quality of the receiver is improved with continuous delivery and lower packet loss. On another level, a method for optimizing the audio quality of a VOIP application (using MOS score) in a WiMAX network is presented in [12]. Furthermore, the author

in [13] extended this work towards multiple wireless technologies by including Wi-Fi and LTE along with WiMAX and optimized three different audio codecs while searching for the best audio quality. In [14], Quality of Experience (QoE) of an audio in wireless networks is guaranteed by jointly optimizing application layer and lower layer networking parameters. Looking into handover performance and the influence on audio quality, the authors in [15] have proposed the use of a MOS-based handover scheme over the traditional Received Signal Strength (RSS) based handover scheme. The numerical results show that the MOS-based handover scheme maintains high call quality and reduce the probabilities for both handover dropping and call dropping.

3.2.3 Multi-objective optimization in wireless networks

As discussed in the introduction, wireless networks typically exhibit a wide range of conflicting objectives. In the field of antenna design, for example, there exists a significant amount of work on Pareto front analysis and multi-objective optimization in relation to field/electrical and geometrical properties. The authors in [16] and [17] have used Response Surface Approximation (RSA) models and sequential domain patching to optimize the geometry of a compact DRA antenna and a planar monopole antenna respectively. Surrogate modelling tools are also applied in antenna design to cut down the time intensive operation [18], [19]. In wireless protocol stacks, however, the use of Pareto front analysis and multi-objective optimization is quite limited. As such, this section gives an overview of multi-objective optimization approaches that have previously been applied to wireless protocols.

Throughput and outage probability, for example, are two conflicting objectives described in [20]. Maximizing the throughput of the first network counteracts the outage probability of the second network, which depends on the received interference due to the first network activity. In their work, an analytical framework is used to simulate and find the OPF of the two wireless networks by applying a channel coding scheme. Another Pareto front optimization on intrusion detection accuracy of wireless sensor networks and memory consumption is considered by Martin et al. [21]. NSGA-II and SPEA2 optimizers were used to locate the APF of the design space composed of 25 million parameters. The inherent large design space demands the use of evolutionary algorithms since they work in sample batches of the population and their computational demand is relatively simple. Looking into a multi-user Orthogonal Frequency Division Multiplexing (OFDM) system, a Pareto front of carrier capacity and power consumption is constructed by using Multi-Objective Particle Swarm Optimizer (MOPSO) [22]. Even though they compare the performance of a modified MOPSO with NSGA-II, the exceptionally low computation requirement of MOPSO limits its usability for complex

wireless networks. Looking into cellular networks, the authors in [23] optimized the throughput sum power of base and relay stations while guaranteeing an end to end Signal-to-Noise Ratio (SNR) metric. A more recent work on multi-layer parameter optimization of Wireless Sensor Networks is presented by [24]. The authors have collected an extensive list of performance metrics (i.e. energy, throughput, delay and loss) out of 7 multi-layer parameters (i.e. packet inter-arrival time and payload size from application layer, maximum queue size, number of transmission and retry delay from MAC layer, transmission power level and distance between nodes from PHY layer). In total, around 50 thousand parameter configurations were experimented within 6 month duration which shows how exhaustive searching can be a tedious and time consuming task.

In general, most of the wireless problems considered in literature use simulations with low computational complexity for approximating the OPF. However, for experimental and time intensive problems, the above solutions become impractical. To the best of our knowledge, this paper is the first to experimentally optimize and analyse the Pareto front of exposure and audio quality objectives from an experimental Wi-Fi conferencing set-up.

3.3 MOSBO

Surrogate-based Optimization (SBO) methods have proven themselves to be effective in solving complex optimization problems, and are increasingly being used in different fields [18, 25–27]. Unlike multi-objective evolutionary algorithms such as NSGA-II [3], SMS-EMOA [4] and SPEA2 [5], surrogate-based methods typically require very few experiment iterations to converge. This feature makes surrogate-based methods very attractive for solving optimization problems that require time-consuming wireless experimentation.

3.3.1 Efficient Multi-objective Optimization (EMO)

The expected improvement and Probability of Improvement (PoI) criteria are widely used for single-objective optimization such as electromagnetic and aerodynamic problems [19, 28]. Recently, multi-objective versions of these criteria are increasingly being used to solve complex multi-objective problems [7, 29]. While they have been used in SBO schemes, due to the computational requirements, their applicability in practice has been limited to problems of 2 objectives. The recently introduced EMO algorithm [7] provides an efficient computation method and can be applied to problems up to 7 objectives.

A flowchart of the EMO algorithm is shown in Figure 3.1. The algorithm begins with the generation of an initial set of experiments X corresponding to different settings $\mathbf{x}$ of the network parameters. These initial configurations are ex-

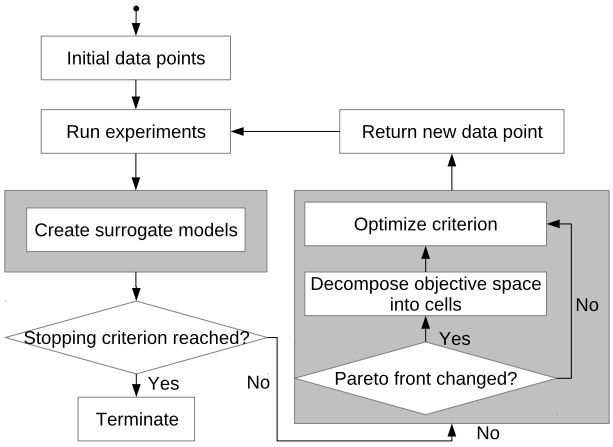


Figure 3.1: Flow chart of the Efficient multi-objective Optimization (EMO) algorithm [7].

ecuted on the testbed, in order to evaluate corresponding values of the objectives $f_j(\mathbf{x})$, for $j = 1 \ldots m$. Each objective function $f_j(\mathbf{x})$ quantifies a QoS performance characteristic and is approximated by a Kriging surrogate model. Based on the models, useful criteria can be constructed to identify new configurations of network parameters that likely improve the currently identified Pareto set $\mathcal{P}$ of Pareto-optimal solutions. As such, these criteria are used to define a new experiment (i.e., a point) in the parameter space, which is again executed on the testbed to evaluate the expensive QoS objective functions $f_j(\mathbf{x})$. The models are then updated with this new information and this process is repeated in an iterative manner until a predefined stopping criterion is met.

This paper adopts the hypervolume-based PoI criterion. It is important to note that the computation of these criteria requires a prediction of the modeling uncertainty. Hence, the choice of surrogate model is limited to those which can provide the uncertainty of the prediction (such as e.g. Kriging).

3.3.2 Kriging

Kriging models are very popular in the optimization of complex systems [30]. This is partly due to the fact that Kriging models provide the mean and prediction variance which can be exploited by statistical sampling criteria. Their popularity also stems from the fact that many implementations are widely available [31–33].

Assume that a set of n samples $X = (\mathbf{x}_1, ..., \mathbf{x}_n)'$ in d dimensions having the target values $\mathbf{y} = (y_1, ..., y_n)'$ is given. The prediction mean and prediction variance of Kriging are then derived, respectively, as,

$$\hat{y}(\mathbf{x}) = \alpha + r(\mathbf{x}) \cdot \Psi^{-1} \cdot (\mathbf{y} - \mathbf{1}\alpha) \tag{3.1}$$

$$s^2(\mathbf{x}) = \sigma^2 \left(1 - r(\mathbf{x})\Psi^{-1} r(\mathbf{x})^\top + \frac{(1 - \mathbf{1}^\top \Psi^{-1} r(\mathbf{x})^\top)}{\mathbf{1}^\top \Psi^{-1} \mathbf{1}}\right) \tag{3.2}$$

where $\mathbf{1}$ is a vector of ones, α is the coefficient of the constant regression function, determined by Generalized Least Squares (GLS), $r(\mathbf{x})$ is a $1 \times n$ vector of correlations between the point $\mathbf{x}$ and the samples X, and $\sigma^2 = \frac{1}{n}(\mathbf{y} - \mathbf{1}\alpha)^\top \Psi^{-1}(\mathbf{y} - \mathbf{1}\alpha)$ is the variance.

Ψ is a $n \times n$ correlation matrix of the samples X,

$$\Psi = \begin{pmatrix} \psi(\mathbf{x}_1, \mathbf{x}_1) & \dots & \psi(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ \psi(\mathbf{x}_n, \mathbf{x}_1) & \dots & \psi(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix},$$

with ψ being the correlation function. The correlation function greatly affects the accuracy of the Kriging model and in this paper the Matérn correlation function [34] with $\nu = \frac{3}{2}$ is used,

$$\psi(\mathbf{x}_a, \mathbf{x}_b)^{Matérn}_{\nu=\frac{3}{2}} = \left(1 + \sqrt{3}l\right) \exp\left(-\sqrt{3}l\right),$$

with $l = \sqrt{\sum_{i=1}^{d} \theta_i (x_a^i - x_b^i)^2}$. The hyperparameters θ are identified using Maximum Likelihood Estimation (MLE).

3.3.2.1 Hypervolume-based probability of improvement

In a multi-objective setting the improvement I over the current Pareto set $\mathcal{P}$ can be defined in several ways. The hypervolume metric (or $\mathcal{S}$-metric) [35] is often used to evaluate the goodness of the Pareto set. The hypervolume indicator $\mathcal{H}(\mathcal{P})$ denotes the volume of the region in the objective space dominated by the Pareto set $\mathcal{P}$, bounded by a reference point $f^{max} + \varepsilon$, where f^{max} denotes the anti-ideal point.

A better Pareto set has a higher corresponding hypervolume $\mathcal{H}(\mathcal{P})$. The contributing hypervolume $\mathcal{H}_{\text{contr}}(\mathbf{p}, \mathcal{P})$ of a Pareto set $\mathcal{P}$ relative to a point $\mathbf{p}$ (see Fig. 3.2) is defined as,

$$\mathcal{H}_{\text{contr}}(\mathbf{p}, \mathcal{P}) = \mathcal{H}(\mathcal{P} \cup \mathbf{p}) - \mathcal{H}(\mathcal{P}), \tag{3.3}$$

$\mathcal{H}_{\text{contr}}$ measures the contribution (or improvement) offered by the point $\mathbf{p}$ over the Pareto set $\mathcal{P}$ and can be used to define a scalar improvement function I as,

$$I(\mathbf{p}, \mathcal{P}) = \begin{cases} \mathcal{H}_{\text{contr}}(\mathbf{p}, \mathcal{P}) & : \mathbf{p} \text{ is not dominated by } \mathcal{P} \\ 0 & : \text{otherwise.} \end{cases} \tag{3.4}$$

Let $y_j = f_j(\mathbf{x})$, $\hat{y}_j(\mathbf{x})$ be the prediction mean, and $s_j^2(\mathbf{x})$ be the prediction variance of a given surrogate model associated with the j^{th} objective, then a Gaussian probability density function ϕ_j with mean $\hat{y}_j(\mathbf{x})$ and variance $s_j^2(\mathbf{x})$ is defined as

$$\phi_j[y_j] \triangleq \phi_j\big[y_j; \hat{y}_j(\mathbf{x}), s_j^2(\mathbf{x})\big], \tag{3.5}$$

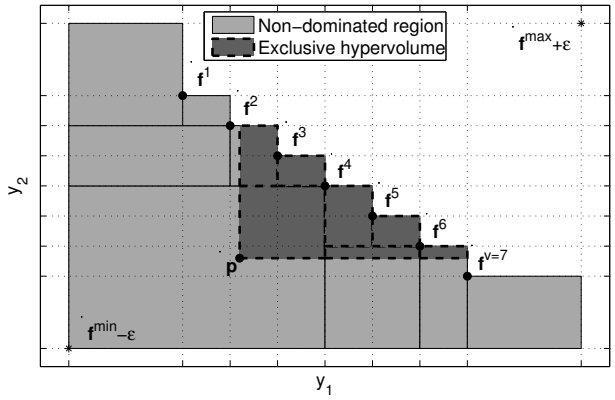


Figure 3.2: A Pareto set for two objectives consisting of Pareto points f^i, for $i = 1...v$. f^{min} and f^{max} denote the ideal and anti-ideal point respectively [7]. The exclusive hypervolume expands the dominated region of the Pareto front (white region enclosed between f^{max} and the Pareto front) which at the same time shrinks the non-dominated region.

In this paper, $\mathcal{H}_{\text{contr}}$ is used as the hypervolume contribution for I to compute the hypervolume-based probability of improvement (PoI) [7]. The hypervolume-based PoI can be written as the product of the improvement function $I(\hat{y}, \mathcal{P})$ and the multi-objective PoI $P[I]$,

$$P[I] = \int_{\mathbf{y} \in A} \prod_{j=1}^{m} \phi_j[y_j] dy_j, \tag{3.6}$$

$$P_{hv}[I] = I(\hat{\mathbf{y}}, \mathcal{P}) \cdot P[I], \tag{3.7}$$

where $\hat{\mathbf{y}} = (\hat{y}_1(\mathbf{x}), \ldots, \hat{y}_m(\mathbf{x}))$ is a vector containing the prediction models of each objective function for a point $\mathbf{x}$. The integration area A of $P[I]$ corresponds to the non-dominated region. The reader is referred to [7] for further details.

3.3.3 Integration of MOSBO in network architectures

Even though the MOSBO optimizer has been quite popular for optimizing electromagnetic and antenna designs, it has not yet been used in the scope of wireless networking. This section discusses how the MOSBO optimizer can be integrated in wireless network applications.

A general wireless network is shown in Figure 3.3, and consists of individual wireless nodes, a network manager and a control and data plane. The network manager utilizes the data plane to send or receive data to and from the wireless devices. In addition, there are a number of design parameters and performance

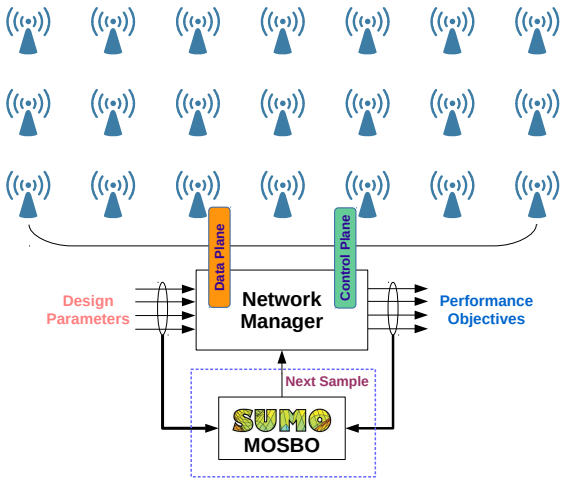


Figure 3.3: Generalized architecture of a wireless network showing the wireless nodes, the network manager and the associated data/control planes.

metrics that can be configured or retrieved on the devices of the wireless network using the control plane, typically using SNMP or similar network management protocols. In traditional networks, configuration parameters are often configured using safe (but non-optimal) settings, or configured based on predictions from non-realistic simulations. To find optimal settings, usually an exhaustive search of all design parameters is required. In contrast, the network manager can make use of optimization tools so that the time consuming part of the exhaustive search can be traded off with a minor quality degradation. To this end, MOSBO can be used in wireless networks to locate the optimum design settings within a relatively short amount of time. In order to do so, the MOSBO optimizer is connected to the network manager as shown in the Figure 3.3 (shown inside the broken rectangle). In the learning/exploration phase, the MOSBO optimizer builds a surrogate model out of the design parameters and their performance objectives. After that, MOSBO starts optimizing the system from the constructed model and provides new design parameters to be configured on the wireless network. This way, the MOSBO optimizer is integrated in wireless networks and near-optimum performance is realized within a relatively short duration of time.

3.4 Experimental Set-up

To evaluate the suitability of the MOSBO approach in wireless problems, a Wi-Fi conferencing experiment is set-up. This section outlines the scenario, the input parameters and the performance objectives of the Wi-Fi conferencing scenario.

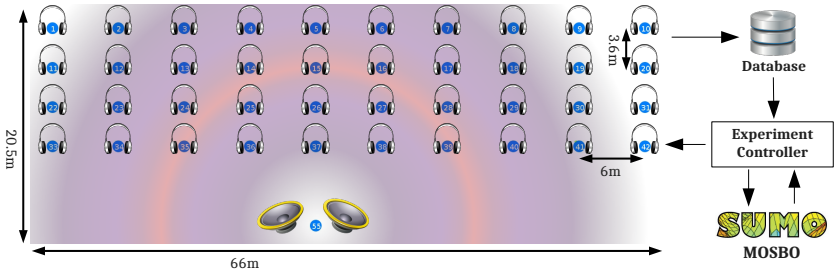


Figure 3.4: Top level view of the Wi-Fi conferencing set-up mapped onto the wireless testbed. Listener nodes are located on the first 4 rows (nodes 1-20, 22-31 and 33-42) and the speaker node is located on the bottom center (node 55).

3.4.1 Experiment Scenario

In a multilingual conferencing session, a speaker's voice is translated into different languages and streamed to listeners. Such an application is typically used in inter-country meetings where different people use different languages to communicate each other with the help of translators. Usually such a conferencing system relies on a wired network and scalability is often a challenge or building a new system is time consuming. The counter part, a Wi-Fi conferencing system, is used in this paper since it addresses the aforementioned challenges. Compared to the multilingual audio conferencing system, a Wi-Fi conferencing system broadcasts the translated audio stream via a Wi-Fi channel and the listeners pick and play it through a wireless headset.

Figure 3.4 shows the Wi-Fi conferencing set-up which is composed of a speaker node transmitting an audio signal on Wi-Fi channel 1 (2412 MHz center frequency), 40 listener nodes receiving the audio signal, a central database collecting the measurement data, a MOSBO optimizer optimizing design parameters and an Experiment Controller (EC) orchestrating the experiment. The hardware components and software tools used in the Wi-Fi conferencing set-up are shown in Table 3.1.

In order to realize the Wi-Fi conferencing set-up, the *IMEC w-iLab.t* wireless testbed [36] is used. The *IMEC w-iLab.t* wireless testbed is equipped with heterogeneous devices such as embedded PCs (having Wi-Fi, Zigbee and Bluetooth technologies), Long Term Evolution (LTE) femtocells/UE dongles, advanced spectrum sensing devices (i.e. Universal Software Radio Platform (USRP), IMEC Sensing Engines, and Wireless open Access Research Platform (WARP) boards) and roomba robots to facilitate mobility experiments.

Table 3.1: Experiment resource description: hardware components and software tools

#	**Resource**	**Description**
1	Wi-Fi nodes	ZOTAC NM10-A-E
2	Wi-Fi chipset	Atheros Sparklan WPEA-110N/E/11n mini PCI
3	Wi-Fi driver	ath9k
4	OS	Ubuntu 14.04 LTS
5	kernel	Linux 3.13.0-33-generic
6	Optimizer	MOSBO

Table 3.2: Input parameters of the Wi-Fi conferencing experiment. The design space consists of 6528 ($32 \times 3 \times 4 \times 17$) elements

#	**Input parameters**	**Range**
1	Opus codec Bit-Rate	[6400, 7200, 8000, ..., 31200] bps
2	Opus codec Frame-Length	[20, 40, 60] msec
3	Wi-Fi Tx-Rate	[6, 12, 18, 24] Mbps
4	Wi-Fi Tx-Power	[0, 1, 2, 3, ..., 16] dBm

3.4.2 Input parameters

While transmitting the audio signal, the speaker node can dynamically adapt four configurable parameters (i.e. Wi-Fi Tx-Power, Wi-Fi Tx-Rate, Codec Bit-Rate and Codec Frame-Length). Table 3.2 provides the description and ranges of each input parameter. The speaker's audio transmit path, indicating all input parameters, is shown in Figure 3.5. A raw audio file is given to an encoder unit which outputs compressed audio frames of a given bit rate and frame length using an opus compression format [37]. Opus is a highly versatile audio codec providing a wide range of bit rates and frame sizes which work from narrow band to full band audio frequencies. It also has a very low latency compared to other audio codecs which makes it attractive for real-time applications. Afterwards, the opus encoded frame is encapsulated, rate and power adjusted before sent over the air.

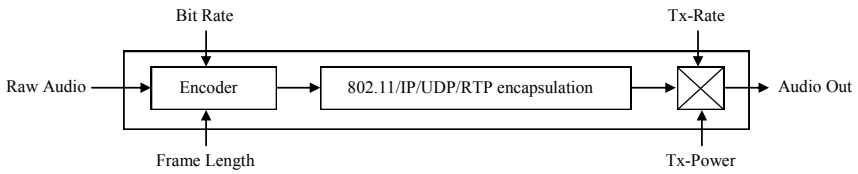


Figure 3.5: Audio transmit path of the Wi-Fi conferencing experiment. Four configurable parameters are available: codec bit rate, codec frame length, Wi-Fi Tx-rate and Wi-Fi Tx-power.

3.4.3 Performance objectives

The scenario is optimized towards two objectives: increasing the audio quality and reducing the transmission exposure. The main reason for selecting the two objectives is the conflicting influence each configurable setting has on their performance: all settings from Table 3.2 influence both audio quality and exposure objectives. Increasing the codec Bit-Rate of an audio signal, for example, will improve the audio quality but negatively impacts the exposure since more packets need to be transmitted. Similar conflicting influences are present for the Tx-Rate, codec Frame-Length and Tx-Power parameters. Although it is easy to predict that all settings from Table 3.2 will influence both objectives, without performing the experiments it is not possible to predict which of these settings will have the largest influence on each of the objectives and which combination of settings will result in optimal performances. This situation is very typical for many wireless systems: engineers typically have a-priori domain knowledge about expected impacts of parameter settings, but this knowledge does not suffice to identify the exact trade-offs and optimal settings. The next sections describe in more detail how each of the objectives are calculated.

3.4.3.1 Audio Quality

As a quantitative measure to evaluate audio quality, MOS scores are often used. A MOS score represent audio quality using a scale of 1 to 5 [38], with 5 being the best and 1 being the worst. In wireless networks, the audio quality typically degrades due to (i) the encoder settings and (ii) the transmission of the audio over the lossy medium. Most MOS score calculations do not differentiate between these two different influences. As such, a new formulation of the MOS score that takes into account both the degradation from the encoder settings and the network settings is shown in Figure 3.6. In this formulation, the audio quality (MOS) of the Wi-Fi conferencing scenario is calculated twice: first after the encoder unit and later after the wireless transmission.

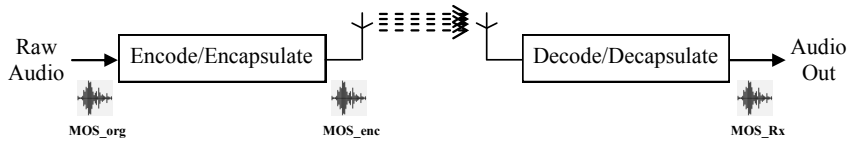


Figure 3.6: MOS calculation flowchart. The audio quality degradation is calculated in two phases: once after the encoder unit and again after the wireless transmission.

In an earlier work [39], the audio quality degradation over a wireless medium has been discussed. Starting from a reduced audio quality after the encoder unit (MOS_enc), the audio is impacted by transmission latency, jitter and packet loss

which results in the received audio quality (MOS_Rx).

$$MOS\_Rx = f(MOS\_enc,\ latency,\ jitter,\ packetLoss) \tag{3.8}$$

Whereas inside the encoder unit, a quality loss is introduced which is a function of the original audio quality, encoder bit rate, type of encoder and audio class used.

$$MOS\_enc = MOS\_norm(bitrate, class) * (MOS\_orig - 1) + 1 \tag{3.9}$$

Unlike the calculation of MOS degradation over a wireless medium, most of the work done to characterize the encoder losses is through subjective tests by using humans evaluating the quality of an encoder output at different bitrates and for different audio samples [37], [40], [41], [42]. Since subjective testing is not possible in an automated wireless system, in this work the Perceptual Objective Listening Quality Assessment (POLQA) method [43] is applied. POLQA is a digital speech analyser model which compares the original and the degraded audio samples and calculates the perception difference using the tradition MOS scale. POLQA is also fully automated and as such is an ideal candidate for the Wi-Fi conferencing scenario. The authors of [44] have evaluated the POLQA estimator by applying speech samples to the opus encoder at different sample rates. Because the audio sample rate inherently impacts the audio bandwidth, this also means that the POLQA estimator has already been applied for different classes of audio as shown in Table 3.3.

Table 3.3: Audio bandwidth and effective sample rate of different audio classes

Audio class	**Bandwidth**	**Sample rate**
NB (narrowband)	4 kHz	8 kHz
MB (medium-band)	6 kHz	12 kHz
WB (wideband)	8 kHz	16 kHz
SWB (super-wideband)	12 kHz	24 kHz
FB (fullband)	24 kHz	48 kHz

Since our goal is to calculate the MOS score after the encoder unit, we make the assumption that the reduced audio quality is independent of the original input quality but only on the encoder output bitrate. Therefore the different OPUS POLQA curves from [44] are normalized and the resulting output is shown in Figure 3.7. The performance curves in Figure 3.7 are not calculated from a single audio source but rather from 5 different audio speeches whose bandwidth utilization is according to their audio classes. If a wideband audio signal is to be encoded, for example, then the green curve is used to get a reasonable quality estimation after the encoder unit. Afterwards, the absolute MOS score is calculated first by scaling and later by translating the normalized MOS score as shown in Equation 3.9. The source audio file [45] used in this paper has a wideband audio class and a quality score of MOS_orig = 4.75.

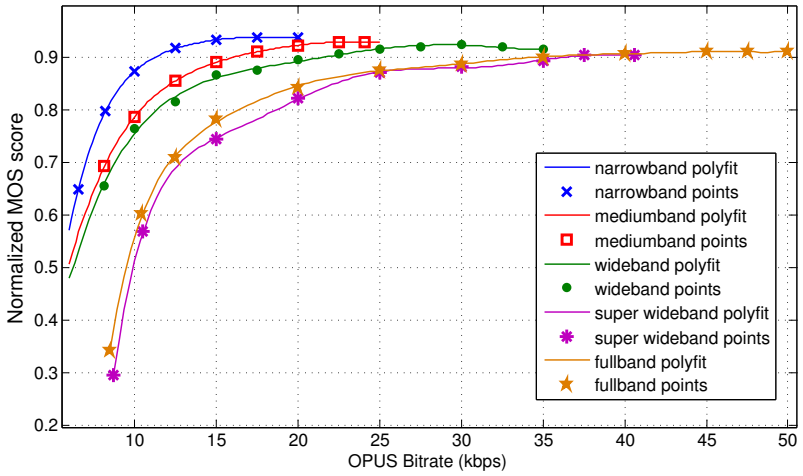


Figure 3.7: Normalized OPUS MOS scores as a function of bitrate for different audio classes

3.4.3.2 Transmission Exposure

For objectively assessing the transmission exposure, recently Varsier et al [8] defined a new metric, the Exposure Index (EI), which aggregates downlink and uplink exposure data and quantifies the total exposure of a population in an area. The EI in SAR units is given by Equation 3.10

$$EI^{SAR} = \frac{1}{T} \sum_{t,p,e,r,c,l,pos}^{N_T,N_p,N_E,N_R,N_C,N_L,N_{pos}} f_{t,p,e,r,l,c,pos} \left[\sum_{u}^{N_U} \left(d^{UL} \overline{P}_{TX} \right) + d^{DL} \overline{S}_{\text{inc}} \right] \left[\frac{W}{kg} \right] \tag{3.10}$$

where t is the period within the considered time frame T, p is the population category, e is the environment, r is Radio Access Technology (RAT), c is the cell type, l is the user load profile, pos is the posture, u is usage of the device, d^{UL} is the uplink dose in units of W/kg for 1W of transmitted power, $\overline{P}_{TX}$ is the average transmitted power by the mobile device, d^{DL} is the downlink dose in units of W/kg for $1W/m^2$ of received power density, $\overline{S}_{inc}$ is the average received incident power density and f is the fraction of the population p.

This exposure formulation is a generalization of the different possibilities that a person can be exposed from a wireless transmission. However, the SAR calculation in [8] is described mainly from a theoretical point of view, and was (i) never defined for real Wi-Fi traces and (ii) was never calculated in real-life using off-the-shelf radios. As such, for experimentally measuring the exposure of a Wi-Fi conferencing set-up, a new metric is needed that can be derived using off-the-shelf commercial Wi-Fi chips (in contrast to the use of dosimeters which are frequently used in exposure research). Applying the formula to the Wi-Fi conferencing set-

up with a single population category, a homogeneous environment, a Wi-Fi Radio Access Technology, an access point cell type, an audio broadcasting load profile, a standing posture and a single purpose device usage, the formula simplifies to

$$EI^{SAR} = \frac{1}{T}\sum_{t}^{N_T}\left[d^{UL}\overline{P}_{TX} + d^{DL}\overline{S}_{inc}\right]\left[\frac{W}{kg}\right] \tag{3.11}$$

Unlike Equation (3.10), the calculation of $\overline{P}_{TX}$ and $\overline{S}_{inc}$ in Equation (3.11) lead to an exact assessment of the EI (with known locations and wireless parameters) for the speaker and listener nodes respectively. For every transmitted and received packet, the speaker and the listener nodes calculate the time duration a packet has occupied the wireless medium, also known as Channel Occupation Time (COT). During the COT amount of time, the speaker antenna next to a speaker induces a SAR that is proportional to the transmitted power $\overline{P}_{TX}$ and the speaker antenna also induces a SAR that is proportional to the incident power density $\overline{S}_{\text{inc}}$ at the listeners. After that, the electromagnetic energy absorption per kilogram of body mass is calculated by applying the uplink and downlink absorption parameters $d^{UL} = 0.0070$ W/kg for 1W of transmitted power and $d^{DL} = 0.0028$ W/kg for $1W/m^2$ of received power density respectively [10]. Finally, the average exposure (power per kilogram of body mass) is calculated by summing all energy absorptions for every transmitted and received packets and dividing the result by the time duration T.

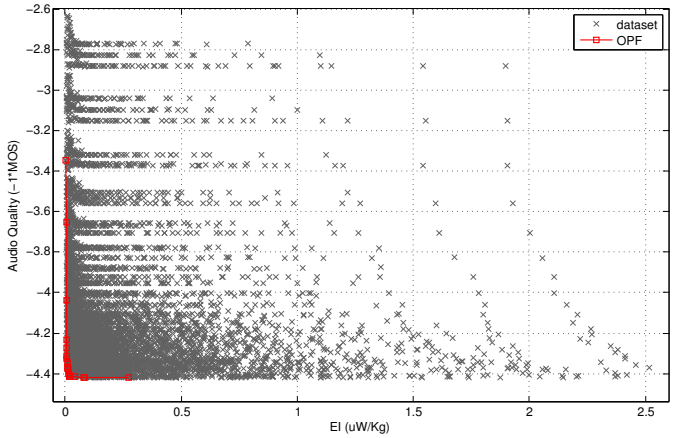
3.5 Result and discussion

In this section, the results from the Wi-Fi conferencing optimization experiment are discussed. First, the behavior of the system is analyzed by creating an exhaustive search model that includes all possible parameter combinations. The next sections analyze the performance of the MOSBO optimized system, by calculating the computational overhead, analyzing the impact of different stopping criteria, and investigating the impact of the initial sample size.

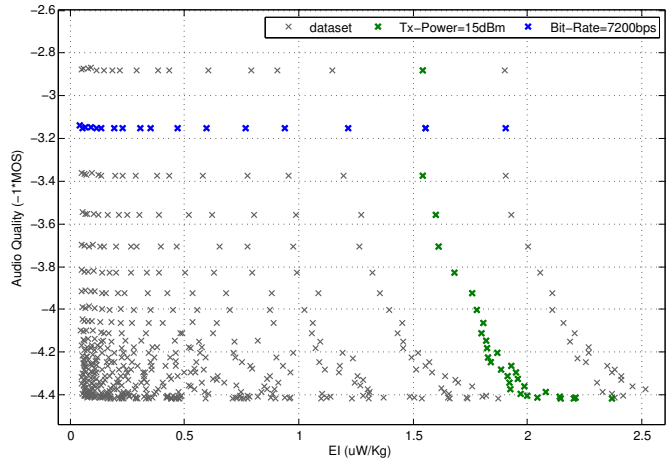
3.5.1 Exhaustive search model

The exhaustive search model is a plot of the objective performances using every input parameter combination. Performing an exhaustive search is not feasible in most realistic situations due to the large number of experiments that need to be performed (6528 experiments in our case, see Section 3.4.2). However the outcomes of an exhaustive search experiment are included in order to compare with the time-efficient MOSBO experiments.

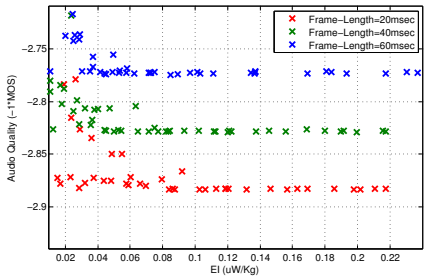
Figure 3.8(a) shows the results of the exhaustive search model. The OPF is calculated by selecting the non-dominated elements from the exhaustive search experiment and is indicated by a red line. By varying individual design parameters, interesting relations between the input parameters and the performance metrics can



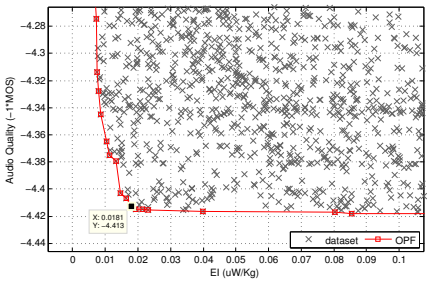
(a) Visualizing an exhaustive search by using all parameter combinations



(b) Visualizing Tx-Power and Bit-Rate variation while keeping Tx-Rate and Frame-Length fixed



(c) Zooming into the top three rows



(d) Zooming into the lower left knee point region

Figure 3.8: Exhaustive search model and OPF plot of the Wi-Fi conferencing experiment. The Y-axis represents the audio quality objective using an inverted MOS score [-1 to -5] and the X-axis represents the transmission exposure of combined uplink and downlink EI values in uW/kg

be discerned. Figure 3.8(b) shows the impact of varying Tx-Power and codec Bit-Rate parameters while keeping the Frame-Length and Tx-Rate fixed at their lowest values. By doing so, an exponentially rising and logarithmically spaced relation between the audio quality and exposure objectives can be observed. Varying the codec Bit-Rate parameter leads to the exponential rising relation (green points @ Tx-Power = 15dBm), whereas the logarithmic spacing is caused by varying the Tx-Power parameter (blue points @ Bit-Rate = 7200bps). Since exposure is proportional to the number of sent packets (which is directly related to the choice of codec Bit-Rate), also the audio-quality has an exponentially rising relation with the codec Bit-Rate [37].

To a lesser extent, the codec Frame-Length parameter also affects the audio quality objective. This is visualized in Figure 3.8(c) by zooming in the top three rows of the overall results. The codec Frame-Length parameter directly affects the latency of audio packets and thus the audio quality is reduced when the Frame-Length is increased which results in a linear relationship between the Frame-length parameter and audio quality objective (blue points @ Frame-Length = 60msec, green points @ Frame-Length = 40msec and red points @ Frame-Length = 20msec). In contrast, the audio quality objective shows no impact when Tx-Power is increased because the experiment was performed in a shielded environment where no interference was present.

The relations between input parameters and objective as illustrated above are difficult to predict exactly, even for domain experts. Although it is interesting to plot these interactions, in many situations the network operator is only interested in identifying the Pareto front with optimal operation points (i.e. the red line in Figure 3.8(a)). A zoomed in version is shown in Figure 3.8(d). Based on the exhaustive search, the best attainable values of the respective objectives is an exposure index of 0.004125 uW/kg and a MOS score of 4.41835. These are the best values of each objectives and can not be improved further by using any parameter combination. When optimizing a solution in which both metrics are considered, the knee point of the OPF is a usual point to consider. The knee point of the OPF is a Pareto point that is closest to a hypothetical intersection point formed by the asymptotic audio quality and exposure lines (i.e. MOS = 4.41835 and exposure = 0.004125 uW/kg). By doing so the knee point of the OPF, shown as a black dot in Figure 3.8(d), has design parameters [Bit-Rate = 28000bps, Frame-Length = 20msec, Tx-Rate = 24Mbps, Tx-Power = 0dBm] and performance objectives [MOS = 4.4128, exposure = 0.018097 uW/kg].

In summary, the exhaustive search model gives insights to relations existing between the objectives and the design parameters and resulted in an optimal Pareto front. The next sections will use the MOSBO optimizer introduced in Section 3.3 in order to approximiate the OPF using fewer experiments.

3.5.2 Stopping criteria

Since the number of experiments should be reduced, it is crucial to identify stopping criteria that determine when the approximate solution (in this case the OPF) is

accurate enough and the experimentation can be stopped, thereby avoiding unnecessary experiment iterations. Typically, stopping criteria are based on estimating the Approximate Pareto Front (APF) progress. There are a number of methods to estimate the APF progress, most of which have previously been applied to Multi-Objective Evolutionary Algorithms (MOEA), however for MOSBO these have not yet been evaluated. Stopping criteria are typically composed of (i) progress indicators, (ii) evidence gathering and (iii) a stopping decision. The next subsections will discuss each of them in more detail.

3.5.2.1 Progress Indicators (PI)

The progress indicator calculates from the collected dataset how much the solution has improved from the previous iteration.

1. Mutual Domination Rate (MDR)
   MDR is a progress indicator evaluated between consecutive Pareto sets [46]. The consecutive Pareto sets are compared and MDR calculates the domination rate of the recent set on the previous set. MDR values range from -1 to 1 where 1 indicates a highest domination, -1 indicates no domination at all and others represent scaled domination rates according to their amount. A high MDR value indicates that the newest Pareto set shows a significant improvement when compared to the Pareto set that was obtained during the previous experiment run.

   Mathematically the MDR is defined as,

$$I_{mdr}\left(P_t^*, P_{t-1}^*\right) = \frac{\left|\triangle\left(P_{t-1}^*, P_t^*\right)\right|}{\left|P_{t-1}^*\right|} - \frac{\left|\triangle\left(P_t^*, P_{t-1}^*\right)\right|}{\left|P_t^*\right|} \tag{3.12}$$

   where $|A|$ is the number of elements in A and $\triangle$(A, B) is the set of elements of A that are dominated by at least one element of B.

2. Epsilon Dominance (ED)
   Similar to the MDR indicator, the ED indicator is also calculated between consecutive Pareto sets [35]. The ED calculates a minimum factor ϵ by which the current Pareto set is better than the former Pareto set with respect to all objectives.

   Mathematically ED, for Pareto sets A and B each having n design objectives, is defined as,

$$I_{\epsilon+}\left(A, B\right) = \max_{Z^2 \varepsilon B} \min_{Z^1 \varepsilon A} \max_{1 \leq i \leq n} \left(Z_i^1 - Z_i^2\right) \tag{3.13}$$

$$I_{\epsilon*}\left(A, B\right) = \max_{Z^2 \varepsilon B} \min_{Z^1 \varepsilon A} \max_{1 \leq i \leq n} \left(Z_i^1 \;/\; Z_i^2\right) \tag{3.14}$$

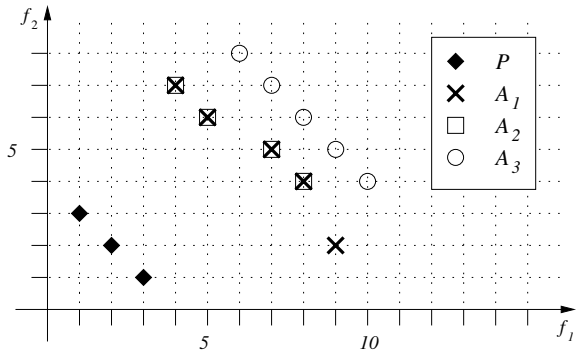


Figure 3.9: A hypothetical dual objective problem having OPF=P and three consecutive Pareto sets (i.e. A_1, A_2 and A_3). $I_\varepsilon(A_1, A_3)$ calculates the minimum factor ε that needs to be added or multiplied on all A_3 objectives to compare with the Pareto set of A_1. This gives $I_{\varepsilon+}(A_1, A_3)$ = -1 and $I_{\varepsilon*}(A_1, A_3)$ = 0.9. On the other hand, $I_{\varepsilon+}(A_1, A_2)$ and $I_{\varepsilon*}(A_1, A_2)$ evaluate to 0 and 1 respectively as all elements of A_2 exist in A_1.

for additive and multiplicative versions respectively. In the experiments performed in this paper, the additive ED indicator is only applied. Figure 3.9, provides a graphical explanation of the ED indicator.

3. Hyper Volume (HV)
   The HV indicator [47], previously discussed in section 3.3.2.1, calculates the volume of the dominated region of a given Pareto set bounded by a reference point r and dominated by all points in the Pareto set. As the optimization progresses, the Pareto set starts to converge to the OPF which also increases the HV indicator. Therefore, larger HV values indicate better Pareto sets.

3.5.2.2 Evidence Gathering Process (EGP)

Next, the evidence gathering process performs a statistical analysis to calculate the changes in the progress indicators over time.

1. Moving Average (MA)
   MA evidence gathering calculates the average of a given indicator values by moving along a fixed calculation window. In time, indicators are expected to reach a stable value, allowing it to compare with a threshold stopping decision at a later stage.

2. STandard DEViation (STDEV)
   STDEV evidence gathering calculates the standard deviation on a collection of indicator values to estimate a possible experiment stagnation. STDEV was used in [39] in conjunction with a combined objective indicator.

3. Linear Regression (LR)
   LR evidence gathering calculates the linear regression of a collection of indicator values and compares it against a stopping decision once the goodness of fit is satisfied. Types of Indicators often used with LR evidence gathering are MDR, HV and ED.

4. Kalman Filtering (KF)
   KF evidence gathering uses the Kalman Filter to estimate the state of a dynamic system from noisy measurements [46]. Kalman Filters assume a system to be linear and consecutive iterations to be only dependent on previous measurements. At each iteration, the state vector of the system and the covariance of the vector are updated based upon a new observation. The filtered estimate and its error value will then be used to make a stopping decision. MDR indicator is mostly applied with KF evidence gathering.

3.5.2.3 Stopping Decision (SD)

Finally, the result from the evidence gathering is compared against a predefined stopping decision.

1. ThresHoLD (THLD)
   Threshold stopping decision compares the evidence gathering estimate against a fixed value and stops iterating when the evidence gathering process falls below a predefined threshold. Threshold stopping decision is often used with STDEV and MA evidence gatherings.

2. ciNormal
   ciNormal is similar to the Threshold stopping decision but rather than making plain threshold comparison, it uses a normal-distribution to address the uncertainty of the EGP values. KF EGP is one typical example that uses the ciNormal stopping decision.

3. Conditional THreshoLD (C-THLD)
   Sometimes the values from EGP are not acceptable until a certain condition is fulfilled. LR EGP, for example, needs to have a very low Root Mean Squared Error (RMSE) between the PI values and their linear approximation before using the slope parameter as a stopping decision.

3.5.3 Performance evaluation

This section evaluates the performance of the MOSBO approach when optimizing the Wi-Fi conferencing scenario described in Section 3.4. To this end, the MOSBO experiment is conducted by selecting different stopping criteria (i.e. indicators, evidence gathering and stopping decision), and comparing the results with those ob-

tained from the exhaustive search model (Section 3.5.1). Two performance metrics are evaluated: (i) Speed up Factor and (ii) Population Domination Rate.

- The *Speed up Factor* (SuF) compares the number of experiment iterations that a MOSBO experiment requires compared to the number of iterations for an exhaustive search experiment. Higher values of the SuF correspond to faster optimizations.
- The *Population Domination Rate* (PDR) gives an indication of the closeness between the APF and the OPF. It expresses the percentage of elements dominated by the Pareto set from the MOSBO experiment when compared to the exhaustive search model. As such, higher PDR values indicate a better Pareto front estimation.

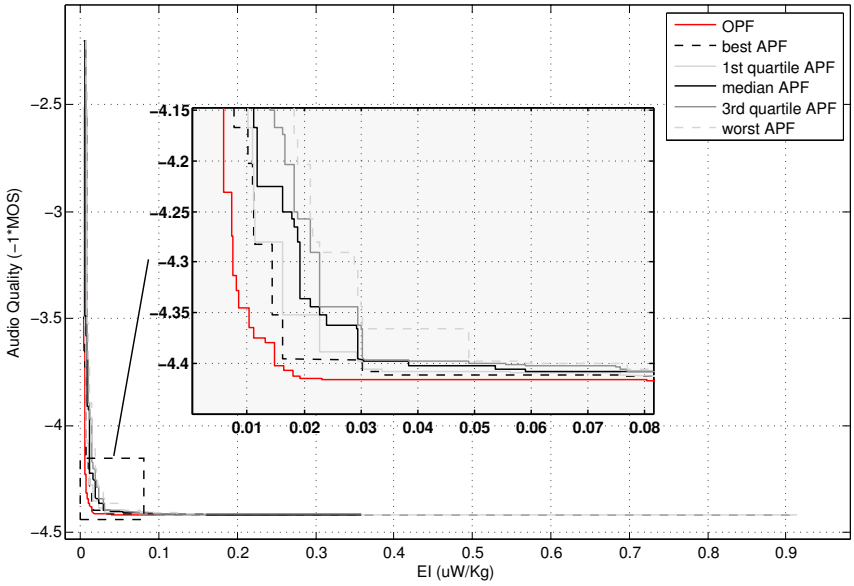
Table 3.4: Stopping criteria combinations used in MOSBO experiment

PI	**EGP**	**SD**	**Remark**
ED	STDEV	THLD	WIDTH=10 and STDEV-THLD=ε
MDR	KF	ciNormal	R=0.1, Pr=95% and KF-THLD=-0.9
HV	LR	C-THLD	WIDTH=10, RMSE=20ε, LR-THLD=10ε and Reference Point: (EI=2.877 uW/kg, MOS=2.4478)

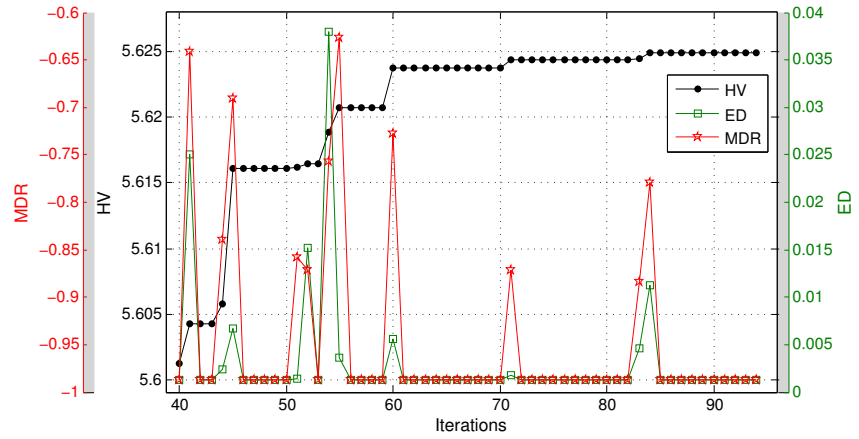
To ensure an adequate approximation of the OPF, multiple stopping criteria have been combined: the MOSBO optimizer stops only after satisfying all of them. Table 3.4 gives an overview of the different stopping criteria combinations used. Although an infinite number of stopping criteria combinations are possible, the selected values have shown to be useful in previous multi-objective optimization problems. In the EGP calculation, the WIDTH parameter is set to 10 for ED and HV progress indicators. The R and Pr parameters (values taken from [46]) are used as a noise value and as an uncertainty estimator of the KF respectively. Regarding the HV indicator, linear regression is applied once the RMSE is below a minimum value (20 times the value of epsilon). Finally SD thresholds of STDEV and LR take epsilon as the lower bound and KF take -0.9 (90% of minimum MDR) as a lower bound.

Figure 3.10(a) shows the snapshot after the end of the MOSBO optimization by comparing the OPF (red curve) with the summary attainment surface plots of 8 different experiments. Summary attainment surface plots [48] are means to visualize the distribution of different APFs according to statistical models. In Figure 3.10(a), 5 different estimators (best, 1^{st} quartile, median, 3^{rd} quartile and worst) are visualized. The best and worst plots represent the boundaries of all APFs but they are biased estimators of the population because they show wide variations for different experiments. On the other hand, the 1^{st} quartile, the median and the 3^{rd} quartile plots are stable estimators, median being the best of all showing the least variation across different experiments.

The average number of experiment iterations to obtain the APF is 94 and due to its limited count, the APF partially overlaps the OPF. Compared to the exhaustive search experiment, the PDR of the MOSBO solution corresponds to 96.58%,



(a) Pareto Front plot



(b) Progress Indicators plot

Figure 3.10: MOSBO Pareto optimization plot after the stopping criteria is met. (a) Pareto front overview, including the Pareto front of the exhaustive search model (red line), the Pareto front from the MOSBO experiment (blue line) and the intermediary experiments (dots). (b) Values of the progress indicators as calculated during the experiment iterations.

meaning that the APF dominates 96.58% of the complete design space. Although the MOSBO Pareto front locates most of the optimal solutions, some Pareto optimal solutions were not identified especially on the knee point region of the OPF.

To analyze the behavior of the MOSBO APF over time, a progress Indicator plot is shown in Figure 3.10(b). In the figure, three PIs (MDR, ED, HV) are visually presented, which together form the MOSBO stopping criteria. The MDR indicator is highly fluctuating while the others are relatively stable. As MDR indicator only accounts for the number of updated Pareto elements but not their magnitudes, this tells that the Pareto elements were showing slight variations which couldn't have noticeable effect on the ED and HV indicators. By applying a combined stopping criteria (Table 3.4) the MOSBO solution converged after 94 iterations, corresponding to a speed up factor of 6528/94 = 69.45 and Population Domination Rate of 96.58%.

3.5.4 Initial sample size sensitivity

In the previous section, it was mentioned that the MOSBO experiment converged after 94 experiments. Although the MOSBO optimizer includes selection criteria for identifying the most promising candidate settings (i.e. the expected improvement criteria, see Section 3.3.1), it relies on the size of the initial samples and the sampling method before generating the initial model. The choice of the sampling method was briefly discussed in the previous paper [39], where Latin Hypercube Sampling (LHS) was shown to have best results. This section expands on this discussion by making a sensitivity analysis over the iteration count.

Providing additional samples typically result in a better initial model, at the cost of additional experiments. As indicated in [39], the problem of exploration vs. exploitation trade-off can be addressed by a good selection of the initial sample size. The initial sample size should explore a problem adequately such that the APF can be retrieved in the shortest time possible. Although previous works in other applications analysed the optimal number of sample points (for example, [49] advises to use 10 times the number of design parameters), the optimal number depends on the smoothness (i.e. the predictability) of the objective functions over the design space.

To evaluate the sensitivity of different initial sample sizes, Figure 3.11 shows the iteration count of MOSBO experiments as a function of the initial sample size. The iteration count indicates the total number of experiments that are performed until the stopping criteria are satisfied. To indicate the efficiency of the MOSBO optimizer, the PDR metric is also plotted in Figure 3.11.

As indicated by Figure 3.11, the iteration count oscillates along a linear regression line (broken purple color on Figure 3.11) with a slope of $1.05 \approx 1.0$. On average, the iteration count increases with the same proportion as with the increase in the initial sample size. This indicates that on average, the MOSBO optimizer spends an equal number of iterations for any initial sample size selected. This is also related to the smoothness nature of the objective functions over the design space. Hence, the Wi-Fi conferencing scenario presented in this paper is not sen-

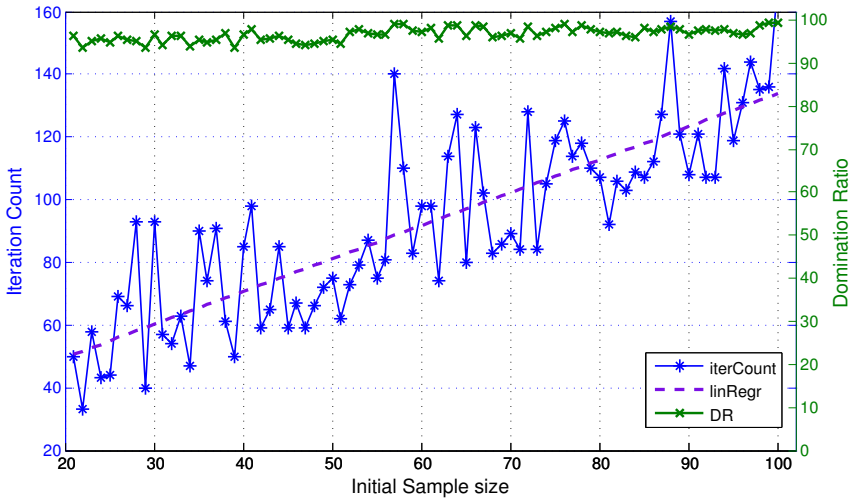


Figure 3.11: Sensitivity analysis of different initial sample sizes as a function of Iteration count

sitive to different initial sample sizes and 22 initial samples are sufficient to get the APF in the shortest time possible (33 iterations). However, as in many cases the smoothness of the evaluated solution is not known a priori, a more conservative amount of samples (i.e. 10*design parameters or 40 initial samples) have been used following the guidelines from [49].

3.5.5 MOSBO computational complexity

Besides the time required for experimentation, also the optimizer spends time calculating the next set of parameter settings that should be investigated. Constructing the Kriging models involves a Cholesky decomposition which has an order of $O(n^3)$ amongst other matrix operations [50], [32]. Since the Kriging models are constructed for each design objective, the overall computation is multiplied by the number of design objectives involved. On the second step, the constructed Kriging models are processed to generate the non-dominated region and the computational complexity is a function of the Pareto points (Figure 5a of [7]). At the last section of the actual optimization, cost prediction and variance prediction processes are involved which require $O(n)$ and $O(n^2)$ computational complexities respectively. The actual execution time of the MOSBO optimizer is dependent on the type of hardware used. Specific to our experimentation set-up (Table 3.1), a single MOSBO iteration needs an average of 1 second, which is significantly less than the time required for most real-life experiment iterations.

3.6 Conclusion

This paper provides a method to identify performance trade-offs in wireless systems through efficient Pareto optimizations. Wireless systems typically have a large design space with several interacting parameters. Analyzing these parameters using a wireless experiment typically consumes a significant amount of time due to the need to deploy, configure and monitor devices. To reduce the time complexity, the paper applied a Multi Objective Surrogate-Based Optimizer (MOSBO) which works by creating a surrogate (Kriging) model out of selected design points and their objective performances.

Afterwards, the advantages of the MOSBO optimizer are demonstrated using a Wi-Fi conferencing scenario where a speaker node adjusts four configurable parameters (i.e Tx-Power, Tx-Rate, Codec Bit-Rate, Codec Frame-Length) and measures two performance objectives (i.e. audio quality and transmission exposure). Both objectives are influenced by all selected parameters and can not be optimized individually since they negatively influence each other. An automated experimentation system was set up in which the Wi-Fi network was integrated with a MOSBO optimizer with the aim of identifying the Approximate Pareto Front (APF) using as few experiments as possible.

Based on an exhaustive search model, the OPF was determined and the interactions between input parameters and performance metrics were described. For the MOSBO approach, the influence of selecting different initial sample points was analyzed and multiple stopping criteria were discussed. A combination of three stopping criteria was proposed to ensure adequate covering of the APF. The Wi-Fi conferencing experiment compares the performance of the MOSBO approach against an exhaustive search experiment. The benefit of the MOSBO optimizer is demonstrated by finishing the experiment using 94 iterations out of the complete design space (6528 elements) and speeding up the experiment 6528/94 = 69.45 times while the APF dominates 96.58% of the complete design space. Moreover, the sensitivity analysis of different initial sample sizes on the performance of MOSBO is investigated and it is found that the Wi-Fi conferencing scenario is not sensitive to different initial sample sizes.

Acknowledgment

The research leading to these results has received funding from the European Unions Seventh Framework Programme FP7/2007-2013 (under Grant agreement No. 318273 LEXNET Project [www.lexnet-project.eu], IWT140048 SAMURAI Project [www.samurai-project.be]) and from the European Horizon 2020 Programme (under grant agreement No. 645274 WISHFUL Project [www.wishful-project.eu], 688116 eWINE Project [www://ewine-project.eu]). This work is also supported by the Interuniversity Attraction Poles (IAP) Programme BESTCOM initiated by the Belgian Science Policy Office.

References

[1] K. Deb. *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, chapter Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction, pages 3–34. Springer London, London, 2011. Available from: http://dx.doi.org/10.1007/978-0-85729-652-8_1, doi:10.1007/978-0-85729-652-8_1.

[2] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems: Second Edition*, chapter Basic Concepts, pages 1–60. Springer US, Boston, MA, 2007. Available from: http://dx.doi.org/10.1007/978-0-387-36797-2_1, doi:10.1007/978-0-387-36797-2_1.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. *A fast and elitist multiobjective genetic algorithm: NSGA-II*. Evolutionary Computation, IEEE Transactions on, 6(2):182–197, 2002. doi:10.1109/4235.996017.

[4] N. Beume, B. Naujoks, and M. Emmerich. *SMS-EMOA: Multiobjective selection based on dominated hypervolume*. European Journal of Operational Research, 181(3):1653–1669, 2007.

[5] E. Zitzler, M. Laumanns, and L. Thiele. *SPEA2: Improving the Performance of the Strength Pareto Evolutionary Algorithm SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. Technical report, Swiss Federal Institute of Technology, 2001.

[6] T. Rakotoarivelo, G. Jourjon, and M. Ott. *Designing and orchestrating reproducible experiments on federated networking testbeds*. Computer Networks, 63:173 – 187, 2014. Special issue on Future Internet Testbeds Part {II}. Available from: http://www.sciencedirect.com/science/article/pii/S1389128613004465, doi:http://dx.doi.org/10.1016/j.bjp.2013.12.033.

[7] I. Couckuyt, D. Deschrijver, and T. Dhaene. *Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization*. Journal of Global Optimization, 60(3):575–594, 2014.

[8] N. Varsier, D. Plets, Y. Corre, G. Vermeeren, W. Joseph, S. Aerts, L. Martens, and J. Wiart. *A novel method to assess human population exposure induced by a wireless cellular network*. Bioelectromagnetics, 36(6):451–463, 2015. Available from: http://dx.doi.org/10.1002/bem.21928, doi:10.1002/bem.21928.

[9] D. Plets, W. Joseph, K. Vanhecke, and L. Martens. *Exposure optimization in indoor wireless networks by heuristic network planning*. Progress In Electromagnetics Research, 139:445–478, may 2013. doi:10.2528/PIER13013003.

[10] D. Plets, W. Joseph, K. Vanhecke, G. Vermeeren, J. Wiart, S. Aerts, N. Varsier, and L. Martens. *Joint minimization of uplink and downlink whole-body exposure dose in indoor wireless networks*. BIOMED RESEARCH INTERNATIONAL, page 9, 2015. Available from: http://dx.doi.org/10.1155/2015/943415.

[11] L. Christianson and K. Brown. *Rate adaptation for improved audio quality in wireless networks*. In Mobile Multimedia Communications, 1999. (MoMuC '99) 1999 IEEE International Workshop on, pages 363–367, 1999. doi:10.1109/MOMUC.1999.819512.

[12] M. I. Tariq, M. A. Azad, R. Beuran, and Y. Shinoda. *Performance Analysis of VoIP Codecs over BE WiMAX Network*. International Journal of Computer and Electrical Engineering, 5(3):345–349, June 2013. doi:10.7763/IJCEE.2013.V5.729.

[13] I. S. H. C. Ilias and M. S. Ibrahim. *Performance Analysis of Audio Video Codecs over Wi-Fi/WiMAX Network*. In Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication, ICUIMC '14, pages 60:1–60:5, New York, NY, USA, 2014. ACM. Available from: http://doi.acm.org/10.1145/2557977.2558009, doi:10.1145/2557977.2558009.

[14] S. Thakolsri, W. Kellerer, and E. Steinbach. *Application-driven cross layer optimization for wireless networks using MOS-based utility functions*. In Communications and Networking in China, 2009. ChinaCOM 2009. Fourth International Conference on, pages 1–5, Aug 2009. doi:10.1109/CHINACOM.2009.5339767.

[15] S. Jadhav, H. Zhang, and Z. Huang. *MOS-based Handover Protocol for Next Generation Wireless Networks*. In 2012 IEEE 26th International Conference on Advanced Information Networking and Applications, pages 479–486, March 2012. doi:10.1109/AINA.2012.105.

[16] S. Koziel and A. Bekasiewicz. *Fast Multiobjective Optimization of Narrowband Antennas Using RSA Models and Design Space Reduction*. IEEE Antennas and Wireless Propagation Letters, 14:450–453, 2015. doi:10.1109/LAWP.2014.2367128.

[17] S. Koziel and A. Bekasiewicz. *Multiobjective Antenna Design By Means of Sequential Domain Patching*. IEEE Antennas and Wireless Propagation Letters, 15:1089–1092, 2016. doi:10.1109/LAWP.2015.2493240.

[18] S. Koziel and S. Ogurtsov. *Numerically Efficient Approach to Simulation-Driven Design of Planar Microstrip Antenna Arrays By Means of Surrogate-Based Optimization*. In Solving Computationally Expensive Engineering Problems, pages 149–170. Springer, 2014.

[19] I. Couckuyt, F. Declercq, T. Dhaene, and H. Rogier. *Surrogate-Based Infill Optimization Applied to Electromagnetic Problems*. Journal of RF and Microwave Computer-Aided Engineering: Advances in design optimization of microwave/rf circuits and systems, 20(5):492–501, 2010.

[20] K. Navaie and T. A. Le. *Fundamental performance trade-offs in co-existing wireless networks*. In 5G for Ubiquitous Connectivity (5GU), 2014 1st International Conference on, pages 246–251, Nov 2014. doi:10.4108/icst.5gu.2014.258069.

[21] M. Stehlk, A. Saleh, A. Stetsko, and V. Maty. *Multi-Objective Optimization of Intrusion Detection Systems for Wireless Sensor Networks*. In Advances in Artificial Life, ECAL 2013, pages 569–576, September 2013. Available from: http://mitpress.mit.edu/sites/default/files/titles/content/ecal13/978-0-262-31709-2-ch082.pdf, doi:http://dx.doi.org/10.7551/978-0-262-31709-2-ch082.

[22] R. Annauth1 and H. C.S.Rughooputh. *OFDM Systems Resource Allocation using Multi-Objective Particle Swarm Optimization*. International Journal of Computer Networks and Communications, 4:291 – 306, July 2012. Available from: http://airccse.org/journal/cnc/0712cnc19.pdf.

[23] R. Devarajan, S. Jha, U. Phuyal, and V. Bhargava. *Energy-Aware Resource Allocation for Cooperative Cellular Network Using Multi-Objective Optimization Approach*. Wireless Communications, IEEE Transactions on, 11(5):1797–1807, May 2012. doi:10.1109/TWC.2012.030512.110895.

[24] S. Fu, Y. Zhang, Y. Jiang, C. Hu, C. Y. Shih, and P. J. Marrn. *Experimental Study for Multi-layer Parameter Configuration of WSN Links*. In Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on, pages 369–378, June 2015. doi:10.1109/ICDCS.2015.45.

[25] R. Badhurshah and A. Samad. *Multiple surrogate based optimization of a bidirectional impulse turbine for wave energy conversion*. Renewable Energy, 74:749–760, 2015.

[26] X. M. Chen, L. Zhang, X. He, C. Xiong, and Z. Li. *Surrogate-Based Optimization of Expensive-to-Evaluate Objective for Optimal Highway Toll Charges in Transportation Network*. Computer-Aided Civil and Infrastructure Engineering, 29(5):359–381, 2014.

[27] M. Costas, J. Díaz, L. Romera, and S. Hernández. *A multi-objective surrogate-based optimization of the crashworthiness of a hybrid impact absorber*. International Journal of Mechanical Sciences, 88:46–54, 2014.

[28] D. R. Jones. *A Taxonomy of Global Optimization Methods Based on Response Surfaces*. Global Optimization, 21:345–383, 2001.

[29] M. Emmerich, A. Deutz, and J. Klinkenberg. *Hypervolume-based Expected Improvement: Monotonicity Properties and Exact Computation*. In IEEE Congress on Evolutionary Computation (CEC), 2011.

[30] A. I. Forrester and A. J. Keane. *Recent advances in surrogate-based optimization*. Progress in Aerospace Sciences, 45:50–79, 2009.

[31] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq. *A surrogate modeling and adaptive sampling toolbox for computer based design*. The Journal of Machine Learning Research, 11:2051–2055, 2010.

[32] I. Couckuyt, T. Dhaene, and P. Demeester. *ooDACE toolbox: a flexible object-oriented Kriging implementation*. The Journal of Machine Learning Research, 15(1):3183–3186, 2014.

[33] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard. *Aspects of the Matlab toolbox DACE*. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2002.

[34] M. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer-Verlag, 1999.

[35] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. G. da Fonseca. *Performance assesment of multiobjective optimizers: an analysis and review*. Evolutionary Computation, 7(2):117–132, 2003.

[36] S. Bouckaert, P. Becue, B. Vermeulen, B. Jooris, I. Moerman, and P. Demeester. *Federating Wired and Wireless Test Facilities through Emulab and OMF: The iLab.t Use Case*. In T. Korakis, M. Zink, and M. Ott, editors, Testbeds and Research Infrastructure. Development of Networks and Communities, volume 44 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 305–320. Springer Berlin Heidelberg, 2012. Available from: http://dx.doi.org/10.1007/978-3-642-35576-9_25, doi:10.1007/978-3-642-35576-9_25.

[37] A. Rämö and H. Toukomaa. *Voice Quality Characterization of IETF Opus Codec*. In INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, 08 2011. Available from: http://www.interspeech2011.org/conference/programme/program.html.

[38] I.-T. S. G. . (1993-1996). *Methods for subjective determination of transmission quality ITU-T Rec. P.800*. Technical report, INTERNATIONAL TELECOMMUNICATION UNION, August 1996.

[39] M. T. Mehari, E. D. Poorter, I. Couckuyt, D. Deschrijver, J. V.-V. Gerwen, D. Pareit, T. Dhaene, and I. Moerman. *Efficient global*

optimization of multi-parameter network problems on wireless testbeds. Ad Hoc Networks, 29(0):15 – 31, 2015. Available from: http://www.sciencedirect.com/science/article/pii/S1570870515000244, doi:http://dx.doi.org/10.1016/j.adhoc.2015.01.014.

[40] W. JIANG and H. SCHULZRINNE. *COMPARISONS OF FEC AND CODEC ROBUSTNESS ON VOIP QUALITY AND BANDWIDTH EFFICIENCY*. World Scientific, June 2002.

[41] A. Rm and H. Toukomaa. *Subjective quality evaluation of the 3GPP EVS codec*. In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, pages 5157–5161, April 2015. doi:10.1109/ICASSP.2015.7178954.

[42] A. Heuberger. *The Future of Communication: Full-HD Voice Powered by EVS and the AAC-ELD Family*. Technical report, Fraunhofer Institute for Integrated Circuits IIS, 2015. Last Accessed April 19, 2016. Available from: http://www.iis.fraunhofer.de/content/dam/iis/de/doc/ame/wp/FraunhoferIIS_Technical-Paper_Full-HDVoice_AAC-ELD_EVS.pdf.

[43] *P.863 : Perceptual objective listening quality assessment*. http://www.itu.int/rec/T-REC-P.863-201409-I/en, 2014. Last accessed April 19, 2016.

[44] R. Chen, T. Terriberry, J. Skoglund, G. Maxweel, and H. T. M. Nguyet. *Opus Testing*. Technical report, IETF, March 27 - April 01 2011. Available from: https://www.ietf.org/proceedings/80/slides/codec-4.pdf.

[45] I. Dyakonov and G. Pascutto. *64kbit/sec stereo multiformat listening test*. https://people.xiph.org/ greg/opus/ha2011/, 2011. Last accessed April 19, 2016.

[46] L. Marti, J. Garcia, A. Berlanga, and J. Molina. *An approach to stopping criteria for multi-objective optimization evolutionary algorithms: The MGBM criterion*. In Evolutionary Computation, 2009. CEC '09. IEEE Congress on, pages 1263–1270, May 2009. doi:10.1109/CEC.2009.4983090.

[47] J. L. Guerrero, J. Garcia, L. Marti, J. M. Molina, and A. Berlanga. *A Stopping Criterion Based on Kalman Estimation Techniques with Several Progress Indicators*. In Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, GECCO '09, pages 587–594, New York, NY, USA, 2009. ACM. Available from: http://doi.acm.org/10.1145/1569901.1569983, doi:10.1145/1569901.1569983.

[48] M. López-Ibáñez, L. Paquete, and T. Stützle. *Experimental Methods for the Analysis of Optimization Algorithms*, chapter Exploratory Analysis of Stochastic Local Search Algorithms in Biobjective Optimization, pages 209–222. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. Available

from: http://dx.doi.org/10.1007/978-3-642-02538-9_9, doi:10.1007/978-3-642-02538-9_9.

[49] J. S. Jason L. Loeppky and W. J. Welch. *Choosing the Sample Size of a Computer Experiment: A Practical Guide*. Technometrics, 51(4):366–376, 2009. doi:http://dx.doi.org/10.1198/TECH.2009.08040.

[50] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

4

Metamodel Based WSN MAC Optimization in Dynamic Environments using Cloud Repositories

Whereas chapters 2 and 3 utilized surrogate models for efficient network optimization, this chapter focuses on the use of metamodels for network characterization. Metamodels are used to characterize the behavior of a complex WSN network scenario and these models are stored in a cloud repository. Later deployments of WSN networks can inspect the cloud repository to find suitable characterizations of their current network conditions, and if so, use these models to quickly optimize the network configuration.

⋆ ⋆ ⋆

Michael Tetemke Mehari, Adnan Shahid, Tom Van Steenkiste, Jan Bauwens, Ivo Couckuyt, Violet R. Syrotiuk, Dirk Deschrijver, Tom Dhaene, Ingrid Moerman, Eli De Poorter

Submitted to IEEE/ACM Transactions on Networking 2018

Abstract Wireless Sensor Networks (WSNs) consist of constrained devices that are used to monitor large areas of buildings, farmlands, nature environments or industrial areas. To cope with diverse application requirements, a large number

of WSN medium access control (MAC) protocols exist in literature each having a wide set of configurable parameters. Furthermore, generic optimization solutions exist to optimize configuration parameters towards domain-specific applications and deployment requirements. However, these solutions assume static network conditions, and as such they can not cope with changes over time caused by deployment changes, environmental dynamics or application requirements. To remedy this, a novel optimization approach is proposed that supports network optimization in slow-changing environments. First, MAC protocols are optimized using state-of-the-art surrogate model-based optimization methods. The resulting models are stored in a cloud repository, where each model represents an optimized instance of the WSN for a specific static environment. Whenever environmental, application or deployment conditions change, the most representative models from the cloud repository are retrieved to quickly configure the MAC protocols, thereby avoiding the time-consuming model creation and optimization phase which otherwise is required. As a proof of concept, a single-hop WSN is set-up where conflicting objectives are Pareto-optimized in dynamic environments by configuring multiple physical layer and MAC layer parameters. Result shows that the proposed approach (model selection and model merging) only requires the execution of 10 experiments for optimizing a network deployed in new conditions, compared to 125 experiments for state-of-the-art optimization techniques or 4800 experiments for an exhaustive search based optimization.

4.1 Introduction

Wireless Sensor Networks (WSN) are a special class of wireless networks that are designed to monitor the physical environment. They are usually composed of a large number of cheap sensory devices. Collectively they sense the environment, send data to a monitoring station via a gateway node and finally apply a corrective action on the system. To collect data from WSNs, a wide range of network stacks exist, ranging from Zigbee to the recent popular IETF based stacks (i.e. IEEE802.15.4, 6lowpan, RPL and CoAP [1]). A commonality between all these protocols is that they consist of multiple layers with different configurable parameters.

Efficiently finding optimal parameter settings of these configurable protocol stacks is currently an active, ongoing research topic, resulting in academic solutions that aim to find Pareto-optimal settings in as little time as possible [2]. The holy grail of this type of research is to identify optimization solutions that are generic and protocol-independent, meaning they can be used for the optimization of parameters of a wide range of protocols stacks. Unfortunately, most of these generic solutions assume WSN networks to be mostly static. As a result, the optimality of the settings depends strongly on the current wireless conditions and application requirements.

Finding generic solutions that guarantee optimal operation in dynamic wireless environments [3] is often challenging, since changing application requirements

might require different network configuration settings. Similarly, a change in the wireless environment will bring sub-optimality in the WSN operation. For example, introduction of an external interference source will cause operation delays because of packet loss. Even worse, these influences can be uncontrollable and unpredictable, making it very difficult to consider at the network design level. This design-time unpredictability can have considerable real-life consequences, such as a forest protection WSN failing to warn the fire department in case of a lightning strike due to an adverse weather condition. In sum, it is quite important to monitor changes in the WSN environment and bring the network back to an optimal operating state, preferably without having to go through a time-consuming optimization process.

Dynamic wireless environments are broadly classified into two groups according to the pace they change overtime.

– **Slow changing environments** are characterized by slowly changing, gradual changes, for example due to environmental aspects such temperature or humidity, an extension of the network size resulting in increased node densities, a change in the application requirements (high vs low data traffic), the presence of a steady external interference (an access point broadcasting beacons), and others.

– **Fast changing environments** are characterized by rapidly changing network changes, for example due to human presence in the vicinity of the WSN, a sudden change in external interference (a Wi-Fi client downloading a file from an FTP server), sudden deep fades in the wireless medium due to mobility of nodes, and others.

Within this paper, we focus on the *slow changing environments*. The aim of this paper is to utilize a cloud repository to adapt the WSN to environmental or application requirement changes, without needing to go through a full optimization process. To this end, we assume the number of WSN deployments is growing rapidly, which is a reasonable assumption in light of the increasing popularity of new IoT standards [4]. As a result, an increasing number of WSNs will be deployed in strongly similar environments. Examples include nature area monitoring solutions deployed in similar outdoor conditions (plains, forests, ...), wireless building automation solutions deployed in suburbs consisting of only a few housing types, or wireless monitoring solutions on multiple office floors in office towers with similar construction materials and layouts. Although the exact behavior of these WSN deployments will vary due to subtle changes of the environment, similar behavioral trends can be found between deployments in similar conditions. This paper exploits this fact by introducing the concept of a cloud repository which stores an offline reference model of the WSN for each static environment where a network has been previously deployed (offline phase).

The conceptual architecture of the cloud repository is shown in Figure 4.1. When the WSN is first deployed, or whenever the environment or application requirements change (online phase), the WSN characterizes the current environment

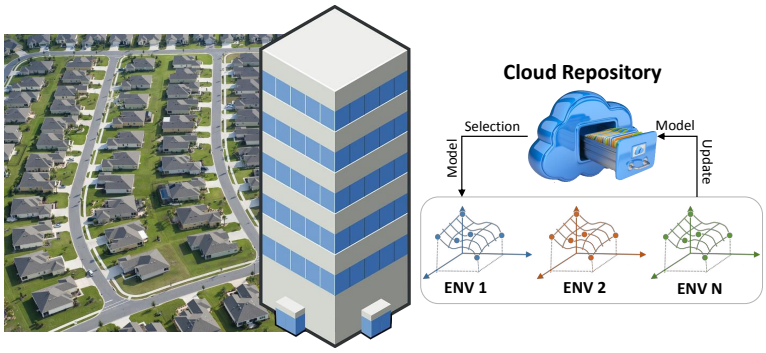


Figure 4.1: WSN optimization architecture using a cloud repository. Reference models are built upon first WSN deployments and stored in the cloud repository. In case of deployments in similar conditions (multiple office floors, multiple suburbs houses), these reference models (if available) can be reused for quickly optimizing new WSN deployments, or to quickly adapt existing WSNs to previously encountered dynamic conditions.

to *select* a representative model from the cloud repository. To select the most suitable performance model from the cloud repository, the WSN analyzes the performance of a set of unique design points (points drawn from the configuration space) which are specifically selected for this purpose. In case of a close match, the best matching model is considered to be representative of the current network conditions, and is used to find optimal configuration settings. Whereas the current performance does not match well to one of the existing reference models, different models are *merged* together to form a new model representing the new environment. Although the model selection and model merging steps require executing a number of trial experiments, the number of required experiments is significantly less compared to creating a new reference model. Moreover, if the model selection process is unsuccessful, the WSN can still be optimized using traditional optimization solutions [5], and the resulting model can be uploaded to the cloud repository for reuse in future deployments.

Finally, it is worth noting that our solution focuses on similarities between slowly changing environments with medium to long coherence time. In contrast to e.g. multi-path fading effects, we assume that the system response stays invariant [6] during a longer period. A typical system with long coherence time is a wireless building automation network where new devices are added less frequently. For fast-changing aspects, other protocol-specific optimization techniques exists (e.g. link estimators). In contrast, our solution focuses on protocol-independent WSN optimization for large configuration spaces, whereby a dynamic environment can be represented as a sequence of longer-duration static environments.

To this end, this paper provides the following contributions.

– The paper proposes an architecture in which the network performance of WSN deployments are stored in a cloud repository, and are used to speed up the opti-

mization of later WSN deployments under similar conditions.

– Our work describes mathematical solutions that can be used to select the most appropriate model amongst multiple models in a cloud repository.

– Finally, a methodology for merging multiple static models into a new model that better represents a unique environment is proposed and evaluated.

The remainder of the paper is organized as follows. Section 4.2 discusses related work regarding state-of-the-art WSN optimization in dynamic environments. The process of metamodeling (model creation, model evaluation and next sample selection) is discussed in Section 4.3. In Section 4.4, a single-hop WSN is set up to be optimized inside a dynamic wireless environment. The optimization process is discussed in Section 4.5 and the methodology is evaluated in Section 4.6. Finally Section 4.7 concludes the chapter.

4.2 Related Work

Although the concept of using a cloud repository to optimize WSNs in a dynamic environment is novel, several of our ideas are built further upon existing literature related to (i) optimization techniques for dynamic network changes and (ii) optimization of complex WSN protocol stacks.

4.2.1 WSN optimization for dynamic conditions

It is long-known that dynamic environments influence network performance. Since the mitigation strategies to cope with these dynamics are often protocol specific, many existing protocols use domain-specific knowledge to modify the behavior of a single protocol layer (e.g. MAC, routing, application, ...) or a single component of a protocol stack (link estimator, coding and modulation selection, ...) in the presence of changing conditions. Examples include the following:

– One approach to combat the influence of interference and variable link quality budgets is to dynamically adapt the modulation type of wireless transmissions. An example of this type of research is the use of an Adaptive Multiresolution Modulation (AMM) scheme [7]. By automating the AMM, the best modulation schemes are selected for different QoS and channel conditions which results in an efficient spectral utilization. A similar approach over fading channels is also proposed in [8], where the signal constellation is adapted according to the signal quality, using adaptive hierarchical modulation.

– For radio chips that do not support multiple modulation types, a similar result can be obtained by dynamically adjusting the transmission power. For example, in [9], the location and power settings of base stations is optimized towards specific application requirements. Prior to handling the dynamic environment (opening of doors and windows or movement of people), an optimum location

of base stations and a minimal power settings that guarantees signal coverage over a predefined area is calculated. Later on, the power settings of the base stations are dynamically updated in order to combat the dynamics of the indoor environment.

– Even at application level, network dynamics can be taken into account. For example, indoor localization using Received Signal Strength (RSS) fingerprinting in dynamic environments is proposed in [10], where a combination of offline and online phases are applied to localize a mobile device from fixed access points. During the offline phase, a radio map database is constructed by storing location and RSS information from every access point. Later on, the online phase calculates the current location of a mobile device by querying the database using access points' RSS information. Moreover, the radio map is updated using a Manifold alignment with Hidden Markov Model (Ma.HMM) in order to adapt to the dynamic environment.

Although these protocol-specific optimizations have proven very efficient, they suffer from several drawbacks.

– **Lack of portability**. The algorithms and optimizers from the same layer are often protocol specific, and can thus not be used interchangeably [11].

– **Ignoring the effects of cross-layer interactions**. Cross-layer interactions in wireless networks are naturally occurring [12], but the fact that these interactions are ignored in dynamic conditions, might not result in system-wide optimal behavior. Dedicated cross-layer solutions exist, such as [13] where a cross layer Time Division Multiple Access (TDMA) MAC protocol is scheduled based on routing information. Results show that the network latency has improved (23% to 29%) when compared to existing similar works. However, this cross-layer design limits the portability of the work even more strictly than the above solutions, since the same approach can not be applied to networks that utilize different MAC or routing protocols.

– **Unpredictable interactions between different optimization algorithms**. Since optimization algorithms from different protocol layers are designed independently, working as a unit might result in unpredictable interactions, ultimately undermining the stability of the overall system.

4.2.2 WSN system optimization techniques

To cope with the above drawbacks, full-system optimization techniques are often applied which aim to optimize the protocol stack by finding the optimal values of configurable WSN parameters. To be usable for a wide range of systems, these approaches need to be generic. To this end, they consider the system as a *black-box*, e.g. the optimization is performed without a prior knowledge of the system. In a previous work, we used Multi-Objective Surrogate Based Modeling

(MOSBO) tools to optimize a Wi-Fi conferencing system [2]. Configurable parameters were selected from multiple layers of the Wi-Fi conferencing system and a Pareto optimal solution was calculated in order to improve the audio quality and lower the electromagnetic exposure. These techniques are generic by design, and most of them can cope with unexpected cross-layer interactions. However, their main drawback is the assumption of static network conditions during the optimization phase. As such, when conditions change, the system has to go through the time-consuming optimization phase again to find the new optimal configuration settings.

4.2.3 WSN cloud repository

As discussed in the sections above, many dynamic network solutions exist, but they are mostly protocol specific and ignore parameter interactions between different protocol layers. In contrast, optimization techniques can optimize the global performance in a protocol-agnostic way, but are typically applied in a single static environment. This paper aims to solve these shortcomings by (i) creating efficient system models based on a limited number of experiment iterations, (ii) storing these system models in a cloud repository, (iii) allowing reuse of these models in case of new deployments or network dynamics that are similar to previously encountered conditions. As a result, our solutions allows efficient cross layer WSN optimization in a dynamic environment using cloud repositories. To the best of our knowledge, this paper is the first to propose this vision, as well as the first paper to provide the mathematical techniques for model selection and model merging that are necessary for this purpose.

4.3 metamodeling

As discussed in the introduction, our solution utilizes a cloud repository in which system models of the WSN performance are stored for different environmental conditions. In this paper, we opted to use metamodels to represent the WSN performance, mainly because these models show good trade-offs between model accuracy and computation complexity [14]. Metamodels, also known as surrogate models or response surface models, are compact analytical models that can mimic the behavioral response of experimental results [15]. It was shown in [16, 17] that they can play a valuable role in global modeling, optimization and sensitivity analysis of control parameters. Specifically in this paper, the metamodels consists of mathematical formulas predicting how input parameters (MAC and PHY configuration settings) impact different output parameters (network performance metrics such as latency, packet loss and energy consumption).

Creating the metamodels by exhaustively evaluating all possible input parameter combinations is not feasible within a limited time frame. Instead, to efficiently create these metamodels using a limited number of experiments or simulations, dedicated techniques are used to guide the data gathering process and to create

a model of the data which can then be used in subsequent analyses. The metamodeling process consists of a loop which evaluates and extends the model in each iteration. The process begins by defining various new parameter settings that lead to the collection of an initial sample set. These samples are spread throughout the sampling space, consisting of the control parameters to be optimised, and are used to construct a model. After training the metamodel, it is validated using cross-validation and appropriate accuracy metrics. When a predetermined accuracy threshold has been reached, the metamodeling process stops.

Within this paper, metamodels decribing the relation between input parameters (MAC and PHY settings) and the expected network performance are created for a wide range of conditions (different node densities, different traffic patterns, different interference levels), which are then stored in the cloud repository. The next subsections provide more details about the used metamodel creation, model validation and model sampling methods, but are not crucial for understanding the basic operations of the cloud repository.

4.3.1 Kriging

There exist various types of metamodels such as Kriging models [18], Gaussian process models [19] and support vector regression models [20], each with their own advantages and disadvantages. In this work, Kriging is used as it is a powerful modeling method for engineering use cases [18]. In the past, it has been used in e.g. antenna design [21] and multi-parameter network optimization problems [16]. It is a Gaussian Process based method for interpolating data. The model not only provides the metamodel output, i.e. the prediction mean, but also an estimate of the confidence bounds of the output for the specific input, i.e. the prediction variance.

Kriging is a kernel based modeling approach and the predictive function takes the form

$$\hat{f}(X) = \sum_{i=1}^{V} \alpha_i \, k(x, x_i) \tag{4.1}$$

where V is the amount of basis vectors, x represent the input data vector and $k(x, x_i)$ is the kernel function used.

In engineering, the Matérn class of kernels is often used, specifically the Matérn $\frac{3}{2}$ kernel [22]

$$k(x, x_i) = \sigma^2 \big(1 + \theta\sqrt{3}||x - x_i||\big) \exp\big(-\theta\sqrt{3}||x - x_i||\big) \tag{4.2}$$

where $\|.\|$ is the L_2 norm and σ^2 is the kernel variance.

In this work, the anisotropic form of the Matérn $\frac{3}{2}$ kernel is used, leading to a hyperparameter vector θ to be trained of size equal to the input data dimension. The hyperparameters of the Kriging model are optimised using maximum likelihood estimation.

4.3.2 Model evaluation

After the metamodel is trained, its performance is tested. This is often done using model accuracy measures such as the Root-Relative-Square-Error (RRSE) as used in this work

$$RRSE(y, \tilde{y}) = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4.3}$$

where n is the number of samples, y is the real value, $\tilde{y}$ is the predicted value and $\bar{y}$ is the mean of the real values.

For an accurate representation of the model accuracy, the RRSE metric is computed with new data on which the model has not been trained. The standard approach is to split the data into a training set and a test set. However, a more robust approach is to use k-fold cross-validation.

In cross-validation, the gathered data is split up into k folds. Then for each of these k folds, the model is trained on the other $k - 1$ folds and evaluated on the fold itself. After all splits have been trained and tested, the scores are aggregated.

4.3.3 FLOLA-Voronoi

If the previously computed model accuracy metric indicates that the model needs to be improved, more data is gathered. To extend the set of samples, several sampling strategies can be used. The most basic form is a one-shot design in which all sample locations and the sampling order are determined upfront [23]. Then, next samples are selected when needed.

A more powerful approach is to use sequential sampling strategies in which information on the already gathered samples is used to determine the new sampling location. In this work, a sequential sampling strategy called FLOLA-Voronoi [24] is used which is a computationally efficient approach of the LOLA-Voronoi [25] algorithm using fuzzy mathematics to increase the efficiency in higher dimensions.

The FLOLA-Voronoi algorithm balances two parts: FLOLA and Voronoi which are respectively the exploitation and exploration steps of the algorithm. In exploitation, the already gathered knowledge is used to pick samples with useful parameter settings. In exploration, samples are chosen in regions where few other samples have been selected. This balance between exploitation and exploration is essential in sequential sampling algorithms. The FLOLA-Voronoi algorithm selects random candidate points and scores them according to the FLOLA and Voronoi criteria. In the FLOLA part, new sample locations are ranked according to their non-linearity. As non-linear behaviour is more difficult to model, more samples are selected in these regions. In the Voronoi part, a Voronoi tesselation is used to detect large regions with no samples. If the distance of candidate points to already selected samples is large, these points get a high score to be selected as a next sample. When the FLOLA and the Voronoi scores have been computed for all candidate points, the two parts are aggregated to compute the final scores. The

samples are ranked and the sample with the highest score is returned as the next sample to be selected.

4.4 System Set-Up

This Section will illustrate the cloud repository based optimization process by introducing a WSN use case that is used throughout the paper. First the scenario is discussed, and afterwards the simulation environment is described.

4.4.1 Scenario

The scenario considered in this paper is a single-hop WSN consisting of 32 source nodes sending periodic messages to a central sink node. Although packets are sent periodically, the sending pattern is randomized at boot time. Later on, the messages are stored in a central location for further processing. The main goal of the optimization process is to ensure the sensor information is received reliably, with minimal energy consumption and minimum latency. Since these objectives can not be met simultaneously and the environment will impact the choice of the configuration settings, multiple Pareto-optimal configuration settings need to be found for multiple dynamic environments. To this end, a cloud repository of reference environment models is used to combat the problem. Figure 4.2 shows the WSN scenario used in this paper.

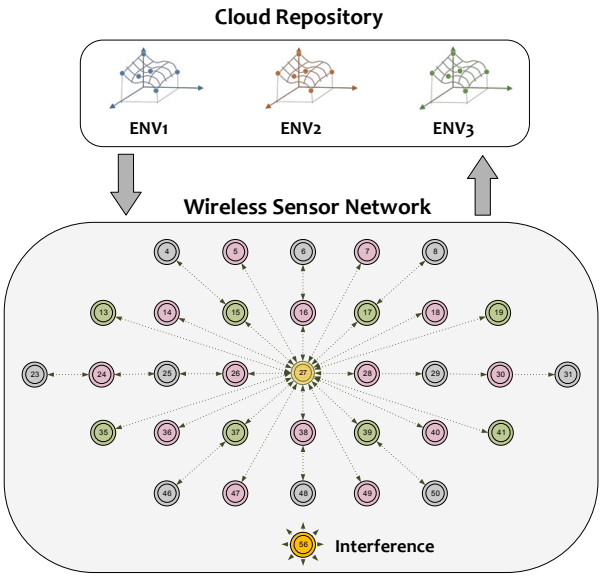


Figure 4.2: WSN scenario consisting of 32 source nodes, 1 sink node and 1 disturber node.

4.4.2 Simulation Environment

To simulate the single-hop WSN scenario, the Cooja simulator is used [26]. Cooja is a java based network simulator for the Contiki operating system and it is designed to simulate a wide range of wireless platforms such as wismote, RE-Mote, micaz, sky and others. Cooja is also a cross level simulator supporting all protocol layers. However, due to the specifics of our WSN scenario, only physical layer and MAC layer design parameters are simulated.

4.4.2.1 Physical layer

At the physical layer, the Unit Disk Graph radio Medium (UDGM) is used as a radio model [27]. UDGM is a simple radio abstraction model described using 2 pairs of parameters: probability of transmission/reception and range of transmission/interference. Even though more advanced radio models are supported inside Cooja (such as DGRM, MRM), UDGM is a sufficiently good choice to demonstrate the benefits of the cloud repository, as long as the range and probability parameters are carefully selected. More advanced models can be used in the future when a complex scenario is required.

4.4.2.2 MAC layer

At the MAC layer, ContikiMAC [28] is used as a radio duty cycling protocol. The operation of ContikiMAC is shown in Figure 4.3.

All ContikiMAC nodes use duty cycling, e.g. they turn off their radios during long sleep periods. The time between two awake periods is referred to as the *cycletime*. No synchronization protocol is used in ContikiMac, and as such the nodes are initially not aware of the sleeping patterns of other nodes.

– **Packet receptions.** A sink node wakes up at regular intervals to check for incoming messages. It first checks the energy level of the wireless medium using Clear Channel Assessment (CCA) mechanism. If the energy level is above a predetermined threshold *CCA-THLD*, the sink node receives the packet transmission. However if not, the sink node performs *CCA-RxMAX* energy detections before deciding no packets are currently being transmitted and turning of its radio.

– **Packet transmissions.** When a node wants to transmit a message, CCA is also used to determine if the wireless medium is clear or not. To this end, the sending node will perform *CCA-TxMax* energy detections. If all of these checks are below a predetermined threshold *CCA-THLD*, the medium is considered available for sending. Because of the duty cycling behavior of the receiver, a transmitting node will keep sending packets until the sink node acknowledges the reception or until the *cycletime* expires. If the *cycletime* expires, the source node restarts the whole process *Retx-max* times before reporting a packet failure to the upper layer. In order to make the process more efficient, all sending nodes keeps track

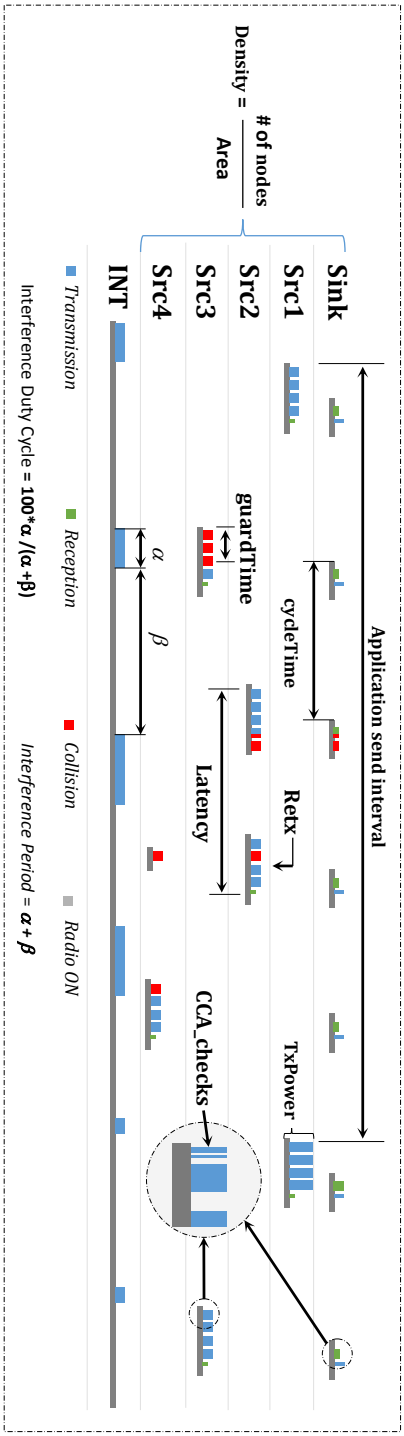


Figure 4.3: Operation of the ContikiMAC duty cycling protocol with indication of the configurable input parameters.

of the time an acknowledgement was received, allowing them to synchronize with devices to which they frequently send packets. This allows transmitting nodes to send packets only when the receiver node is about to wake up, with the addition of some *guardTime* to account for clock-drift.

The original ContikiMAC protocol was designed as an embedded C module and parameter reconfiguration is limited at runtime. To alleviate this problem, the Time Annotated Instruction Set Computer (TAISC) [29] is used to compile and execute an in-house built Contiki-MAC protocol. In terms of functionality, the TAISC Contiki-MAC protocol is similar to the original Contiki-MAC except with an added benefit such as run time parameter reconfiguration, on the fly MAC processor switching, among others [29].

4.4.3 Metamodel Input/Output

The performance of the WSN is influenced by the values of several configurable parameters. At all times, the WSN configures the design parameters as a unit making it a centrally controlled system. To this end, six design parameters, shown in Tables 4.1, are used to configure the WSN and at the same time serve as inputs to the metamodel. Based on the received packets, the sink node measures three performance metrics, shown in Table 4.2. These three performance metrics are either measured (during the time-consuming process of metamodel creation), or predicted (when using already existing metamodels to predict the best performing design parameters).

Table 4.1: WSN PHY and MAC configuration parameters. The configuration space consists of six configurable parameters, together resulting in 4800 ($5 \times 4 \times 3 \times 4 \times 4 \times 5$) potential configuration settings.

Parameters	**Description**	**Range**
Tx-Power	transmission power	[-4, -2, 0, 2, 4] dBm
cycletime	periodic wake up duration	[33, 66, 98, 131] msec
CCA_THLD	threshold for packet reception	[-95, -80, -65] dBm
CCA_RxMax	maximum # of receive CCA checks	[2, 3, 4, 5]
CCA_TxMax	maximum # of transmit CCA checks	[4, 5, 6, 7]
Retx-max	maximum # of cycletime retries	[0, 1, 2, 3, 4]

Table 4.2: WSN performance metrics

Performances	**Description**	**Unit**
LAT	Packet transmission latency of source nodes	msec
PER	Packet Error Rate between source and sink nodes	%
ENG	Total energy consumption of source and sink nodes	mJ

In a general optimization problem, optimum design parameters are expected to meet the performance objectives. However, in networked problems performance objectives tend to conflict with each other. A similar behavior is observed in the

proposed WSN scenario: increasing the *cycletime* parameter, for example, reduces *energy consumption* because the sink node wakes up less frequently, but at the expense of increased *latency*. Similarly, reducing *transmission power* favors *energy consumption* but increases *packet error rate*. On the other hand, increasing *cycletime* decreases *packet error rate* but it worsens the *latency* since it takes more time to successfully transmit a packet. These performance trade-offs between the different performance metrics are illustrated in Figure 4.4, which shows a 3D Pareto Front of the WSN performance when using different configuration settings. Of course, if environmental conditions change (different transmitter densities, different interference levels, etc.) the structure of this plot will change.

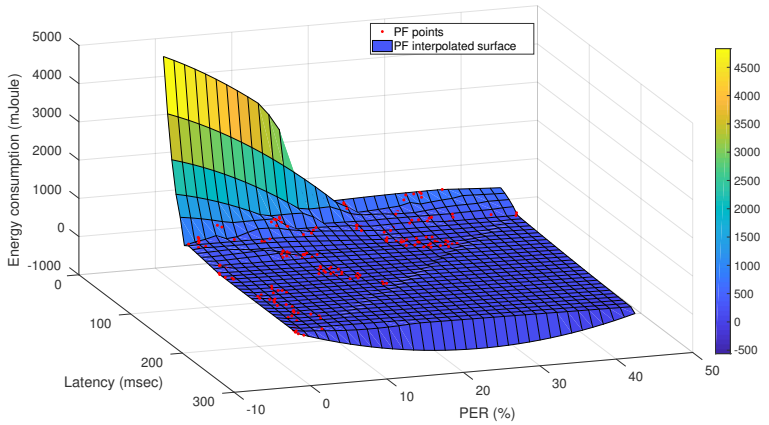


Figure 4.4: 3D Pareto Front (PF) of the system performance in a single environmental metamodel. Latency and packet error rate performance improvement is counteracted by an increase in the energy consumption and vice-versa.

4.4.4 Dynamic Environments

So far, the single-hop WSN was introduced without considering the influence of a dynamic environment. As mentioned before, our optimization approach assumes the presence of a slowly-changing dynamic environment as a sequence of static environments. Within this paper, we consider three potential network dynamics.

– External interference is the first parameter considered. Interference can be caused due to several reasons, but is mostly caused by competing wireless technologies. As a source, a periodic interference with a variable duty cycle is used (interference timeline in Figure 4.3). Inside the Cooja simulator, a controlled disturber mote is used (yellow colored node in Figure 4.2). This type of dynamism is representative of an *environmental change, outside the control of the WSN operator*.

– Secondly, the node density is considered representing the number of WSN nodes

in a given area. This represents scenarios in which the network is extended with newly added devices, or multiple similar deployments are considered with varying number of monitoring devices.

– Finally, an application send interval is considered describing the frequency of measurement messages sent by source nodes. It can, for example, be used to express the need for more frequent sensor monitoring updates.

Table 4.3 shows the details of these environment parameters. A metamodel describing the impact of WSN settings on the performance is typically only valid for one specific combination of environmental parameter conditions. Although within this paper metamodels for 45 different environments are created, the total number of potentially encountered environments is infinite.

Table 4.3: Dynamic environment parameters. The environment space consists of 45 ($3 \times 5 \times 3$) evaluated environments

Parameters	**Description**	**Range**
Density	WSN node density	[15, 26, 47] Nodes/decameter^2
INT_DY	External interference (duty cycle)	[0, 5, 10, 15, 20] %
send_intval	WSN message sending interval	[6, 12, 18] sec

4.5 Methodology

This section discusses the cloud repository methodology while optimizing the single-hop WSN set up. Figure 4.5 shows the different online and offline phases of the optimization process. During a first offline phase, the reference environment models are created and stored in the cloud repository. Whenever a new network is deployed, it starts immediately with the online phase, during which it characterizes the current environment by executing a small number of experiments to download the best reference model(s) from the repository. These models are used to predict the optimal system configuration for the current network conditions. The system then enters a continuous monitoring phase and checks for any change in the environment. If the change is significant, the optimization process is restarted thereby characterizing the environment and optimizing the WSN either by selecting or merging models. Each of these steps is discussed in more detail in the next sections.

A Reference Models

All reference metamodels are built during the offline phase by manually creating a simulation scenario using the dynamic environment parameters shown in Table 4.3. To create the metamodel, the surrogate model builder starts from 40 initial samples, using a Latin Hypercube Design (LHD) [30], and iterates sequentially, using a FLOLA-Voronoi sampling strategy, until an accurate model is created.

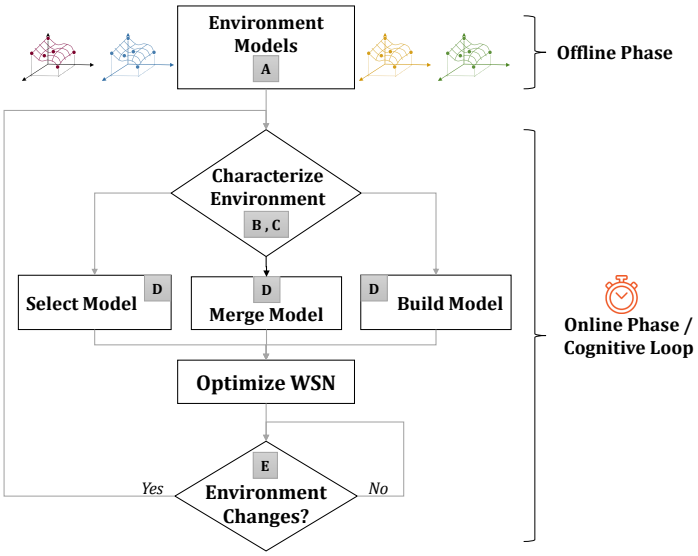


Figure 4.5: Flowchart showing the offline and online operations. The description of each block is referenced by a section letter.

During the sequential design phase, the accuracy of the model is continuously checked using a 10 fold cross-validation approach and the execution is stopped when it becomes sufficiently low or sufficiently stable (see Section 4.3). On average, 125 experiments are executed to create a stable model, compared to 4800 experiments if an exhaustive search approach was used.

To compensate for inherent network variability, multiple models for each environment are created by running 15 simulations with different random seeds. Outlier models are removed using the Median Absolute Deviation (MAD) method [31] and the remaining models are then averaged to create a representative, averaged reference model for a specific environment.

To understand the behavior of reference models, a variance based sensitivity analysis is applied. Table 4.4 shows the sensitivity analysis of one reference model from the cloud repository using Sobol indices [17]. Each Sobol index represents the variance contribution of a parameter on one of the performance metrics. All Sobol indices, from a given performance metric, add up to 1 and a higher Sobol index shows a higher parameter sensitivity.

The diagonal elements of Table 4.4, shaded in gray color, are the first-order Sobol indices from individual parameters. The upper triangular elements without the diagonals are the second order Sobol indices representing two parameter interactions. Usually first-order indices show the strongest influence (*cycleTime* in Table 4.4-a and 4.4-c, *Retx_max* in Table 4.4-b), but sometimes second order indices also have appreciable influences (*cycleTime*×*Retx_max* in Table 4.4-b, *cycle-*

Table 4.4: WSN sensitivity analysis. Density = 15 Nodes/decameter^2, INT_DutyCycle = 5% and send_interval = 6sec. Most sensitive parameters and parameter combinations are shown in bold.

(a) Latency metric

	Txpower	*Cycletime*	*CCA_THLD*	*CCA_RxMax*	*CCA_TxMax*	*Retx_max*
Txpower	**0.053841**	0.006253	0.000089	0.000061	0.000039	0.011249
Cycletime		**0.565720**	0.000128	0.000200	0.000094	**0.048139**
CCA_THLD			0.000083	0.000078	0.000023	0.000022
CCA_RxMax				0.000075	0.000008	0.000195
CCA_TxMax					0.000899	0.000388
Retx_max						**0.308922**

(b) Packet Error Rate metric

	Txpower	*Cycletime*	*CCA_THLD*	*CCA_RxMax*	*CCA_TxMax*	*Retx_max*
Txpower	**0.046884**	0.000322	0.000834	0.000125	0.000700	0.001688
Cycletime		0.006150	0.000244	0.000133	0.000388	**0.014565**
CCA_THLD			0.000332	0.000601	0.000346	0.002129
CCA_RxMax				0.000500	0.000819	0.001213
CCA_TxMax					0.001706	0.000939
Retx_max						**0.911772**

(c) Energy consumption metric

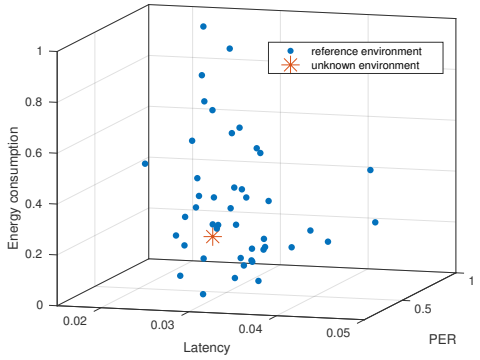
	Txpower	*Cycletime*	*CCA_THLD*	*CCA_RxMax*	*CCA_TxMax*	*Retx_max*
Txpower	0.017849	0.001331	0.000082	0.000233	0.000126	0.002276
Cycletime		**0.758106**	0.000355	**0.044985**	0.000163	0.000497
CCA_THLD			0.000055	0.000079	0.000009	0.000440
CCA_RxMax				**0.159748**	0.000038	0.000376
CCA_TxMax					0.000098	0.000071
Retx_max						0.011972

Time$\times$*CCA_RxMax* in Table 4.4-c). In section 4.4.3, we demonstrated the conflicting nature of *Energy Consumption* and *Latency* metrics using a 3D Pareto Front. This behavior can also be observed in Tables 4.4-a and 4.4-c where the first-order *cycletime* parameter shows a strong influence on both latency and energy consumption objectives, but only has a minor impact on packet error rate. These results demonstrate the complexity of optimizing wireless systems with multiple parameters for different conflicting performance objectives, especially since the value of the Sobol indices depend strongly on the current environmental conditions.

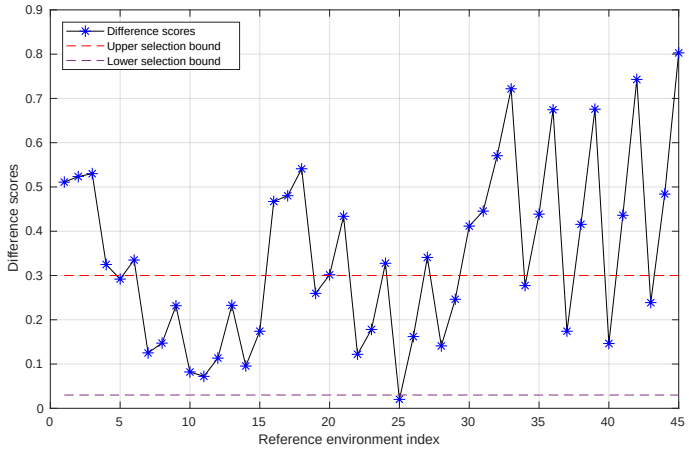
B Characterizing an Unknown Environment

After reference models are created, an unknown environment is characterized when the WSN starts operation or when the current environment changes (see Section 4.5-E). Ideally, the WSN can retrieve the reference model(s) from the cloud repository that correspond most closely to the current environmental conditions. However, many constrained embedded devices do not have the capability to identify all aspects of their environment (interference levels, node density, etc.). Instead, we estimate the best matching model by comparing the performance of the

unknown environment with the performance of the reference models in the cloud repository. To estimate the fitness of existing models from the cloud repository, the WSN performance of a *design point* from the unknown environment is compared with the predicted performance of the reference models. Figure 4.6 shows an environment estimation example using latency, packet error rate and energy consumption performance metrics.



(a) Performance space plot



(b) Environment space plot

Figure 4.6: Estimation of best matching reference environment. (a) Comparison of a design point performance under unknown environment (red asterisk) with the performance of the reference models from the cloud repository (blue dots). (b) Performance difference score of the design point between the unknown environment and the reference models from the cloud repository.

As shown in Figure 4.6a, due to variations in the environment, the design point performance will closely match the performance of several available reference

models, but will be very different from others. To calculate how closely each reference model matches the observed performance, all performance metrics are normalized to a [0 1] scale and the *Euclidean distance* is calculated, resulting in a *difference score* for each reference environment and the unknown environment, see Figure 4.6b. Afterwards, the *difference scores* are compared to a *lower selection bound* and an *upper selection bound* to determine whether the unknown environment can be represented by a reference model or by a unique merged model or by a newly built model.

– If any of the difference scores calculated is below the lower selection bound (unknown environment is very close to at least one reference environment), the procedure selects a reference model that corresponds to the smallest *difference score*.

– Like wise, if all difference scores are above the upper selection bound (unknown environment is far away from all reference environments), the procedure creates a new model from scratch.

– Otherwise, the procedure creates a unique model by merging the most closely matching reference models.

C Design point selection

So far, we have used a single design point to measure the WSN performance and estimate the environment it is operating in. But how do we select a design point from a huge configuration space? Even worse, there is a possibility for a single design point experiment to become an outlier. Thus, multiple design points are used (10 in our case), to make the environment characterization process more robust.

Moreover, to provide further information, design points should be selected that result in diverse possible output performances amongst all reference models. To select which design points (e.g. input settings) are best used for environment estimation, the concept of a *spreading factor (SF)* score is introduced. The SF score, for a specific input setting, calculates the spread of (normalized) performance values between all reference models. This is visually shown in Figure 4.6a, as to how much the reference points (blue dots) are physically spread out. The higher the SF score becomes, the better the environment characterization performs because an unknown environment can easily be matched to the closest reference model. To this end, the SF of an input setting i is defined as follows.

$$SF(i) = \frac{2}{M(M-1)} \sum_{j=1}^{M-1} \sum_{k=j+1}^{M} \left\| perf_{j,i}, perf_{k,i} \right\| \tag{4.4}$$

where M is the number of reference models, $perf_{j,i}$ is the normalized performance of reference model j using input configuration i and $\|a, b\|$ is the *Euclidean distance* between a and b.

The cloud repository calculates the SF for all possible combinations of the input settings. Off these, the top N settings showing the largest SF scores are selected

as design points. As such, to determine the set of design points, the following formula is used.

$$designPoints \quad = \quad top10(\underset{i=1\ldots X}{filter_D}(sort(SF(i)))) \tag{4.5}$$

where X represents the total number of input settings and $sort(.)$ is a function to sort the SF values from high to low and $top10(.)$ is a function to take the first 10 values from a list of numbers. It is likely that neighboring input settings with high SF's will also have high SF values. However, neighboring input settings are not interesting because they provide little information to the model selection process. For this reason, the $filter$ function is added, which selects all SF entries having D input parameter values different from earlier entries.

After calculating the difference scores of each design point, the environment is characterized by using an average plot computed from the 10 difference scores (Figure 4.8 shows an example).

D Model selection and merging

If the unknown environment is characterized for selection, the procedure is straightforward. The selected reference environment is downloaded from the cloud repository and the performance of the WSN is optimized accordingly. However, it is possible that none of the reference models is a direct match for the unknown environment, but instead multiple reference models with similar model *difference scores* are potential candidates. In this case, merging multiple reference models becomes a viable option. This merging approach is especially beneficial in case the operating conditions of the WSN with unknown performance are within the environmental conditions of two or more reference models.

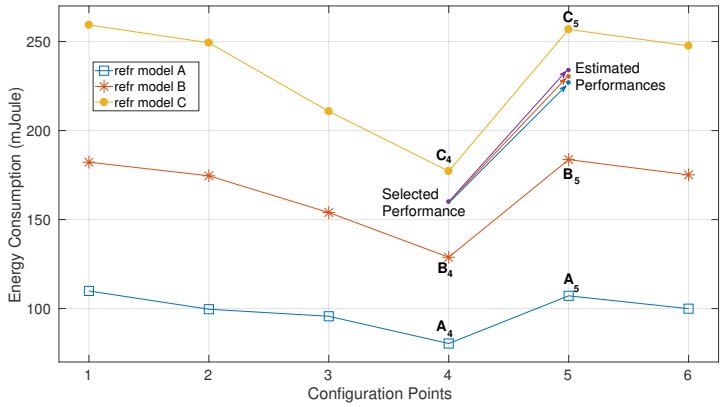


Figure 4.7: Simplified model merging procedure showing the predicted performance of 3 reference models for 6 possible input settings. The design point corresponding with input 4 is used to extrapolate the most likely performance at the remaining input settings (1,2,3,5 and 6).

Figure 4.7 illustrates a simplified model merging procedure. The overall configuration space consists of 6 input settings (x-axis). Multiple reference models are available in the cloud repository, each predicting the performance metric "energy consumption" (y-axis) for all 6 input settings. Out of all available models, the observed energy consumption (160 mJoule) from the unknown environment is most similar to reference models A, B and C. The 4^{th} configuration point corresponds to the design point that is used for environment characterization. To improve the prediction accuracy, these three models are merged at the remaining input settings (1,2,3,5 and 6). In total, 3 steps are involved during the merging process.

D.1 Linear model prediction using two reference models

The merging process takes two reference models at a time. From these two models, the predicted performance at the other input settings is linearly scaled. The predicted performance at design point i, based on models j and k, can be calculated using the following formula.

$$\forall i \in \{1, 2, ..., X\},$$
$$Est\_perf_{u,i}\,(j,k,sel) = \left(perf_{u,sel} - perf_{j,sel}\right) * \left(\frac{perf_{k,i} - perf_{j,i}}{perf_{k,sel} - perf_{j,sel}}\right) + perf_{j,i} \quad (4.6)$$

where i is the input setting, j and k are selected reference models, u is the unknown environment, sel is the selected design point and $perf_{a,b}$ is a performance metric at reference model a and input setting b.

For the simplified example of Figure 4.7, this will give $3 = \binom{3}{2}$ model estimates, by selecting 2 reference models from a group of 3 ($A - B$, $A - C$ and $B - C$).

D.2 Linear model prediction using M reference models

Next, a *representative model* of the estimated models is calculated by applying a weighted sum of the remaining points proportional to a *performance gap (PG)* score. A *PG* score defines the difference in performance between 2 reference models at the selected design point.

$$\forall i \in \{1, 2, ..., X\},$$
$$Repr\_perf_{u,i}(sel) = \frac{1}{PG\_inv\_total} * \left(\sum_{j=1}^{M-1} \sum_{k=j+1}^{M} PG\_inv_{j,k} * Est\_perf_{u,i}\,(j,k,sel)\right) \quad (4.7)$$

where

$$PG\_inv_{j,k} = \frac{1}{\|perf_{j,sel}, perf_{k,sel}\|} \quad ,$$
$$PG\_inv\_total = \sum_{j=1}^{M-1} \sum_{k=j+1}^{M} PG\_inv_{j,k} \quad ,$$

M is the number of selected reference models and $Est\_perf_{u,i}(j,k,sel)$ is a performance estimate of reference models j and k, unknown environment u, and input settings i and sel.

D.3 Model merging using multiple design points

Since we have a single *representative model* for a given design point, this will translate to 10 *representative models* to the number of design points selected for robust environment characterization (Section 4.5-C). The final model will be an average over all *representative models*.

$$\forall i \in \{1,2,...,X\},$$
$$Merged\_perf_{u,i} = \frac{1}{10}\sum_{sel=1}^{10} Repr\_perf_{u,i}(sel) \tag{4.8}$$

where $Repr\_perf_{u,i}(sel)$ is the representative performance value of unknown environment u at input settings i and sel.

Like wise, the merged model is used to optimize the WSN by changing the MAC & PHY input settings that result in the required WSN performance.

E Detecting Environment Changes

Once a representative model is available (either by selecting or merging reference models), the model is used to determine the optimal operation point. From this point on, the WSN operates in an optimum state until there is a change in the environment. However, due to inherent network variations, the performance and environment will continue to vary slightly over time. This raises the question: when do we trigger a new optimization phase? One way to handle this is by continuously looking at the performance metrics and trigger the characterization only when the performance falls below a predefined threshold. In this case, the performance metrics are continuously monitored and checked against a set of objective thresholds (maximum value for latency, packet error rate and energy consumption). A second approach is to use the confidence intervals of predicted values which are included in the created metamodels. For example, a 95% confidence interval can be used as a threshold limit: if one of the performance metrics is outside the confidence interval, we assume the environment has changed and the environment is characterized. Even more advanced approaches could be thought of, but is left as future work.

4.6 Evaluation

Until now, we have discussed model selection and model merging procedures. The next step is to validate these procedures. To this end, we will optimize the scenario described in Section 4.4 in varying conditions as shown in Table 4.5. To evaluate

the feasibility of model selection, the first two unknown environments use the same environment settings as existing reference models in the cloud repository. The remaining three environments are intentionally altered from the reference models so that model merging is a viable option.

*Table 4.5: List of environments to test selection or merging of reference models. Remarks with * use the same environment settings as an existing reference environment.*

	Density	INT duty cycle	send interval	Remark
1	15 Nodes/decameter^2	5 %	12 sec	*
2	26 Nodes/decameter^2	0 %	6 sec	*
3	15 Nodes/decameter^2	7 %	6 sec	
4	17 Nodes/decameter^2	17 %	9 sec	
5	37 Nodes/decameter^2	7 %	9 sec	

4.6.1 Design point selection

First, the design points are selected using the procedure outlined in Section 4.5-C. Table 4.6 shows the selected design points ($N = 10$) having at least 3 parameter value differences ($D = 3$).

Table 4.6: Top 10 design points for estimating an unknown environment ($N = 10$, $D = 3$). The Table shows the input settings for each design point, and the calculated spreading factor over all reference models.

Txpower	Cycletime	CCA_THLD	CCA_RxMax	CCA_TxMax	Retx_max	SF score
0	32768	-95	5	4	0	0.4059
0	32768	-80	5	7	1	0.4007
2	32768	-80	5	5	0	0.3971
4	32768	-65	5	7	0	0.3930
-2	32768	-95	5	6	1	0.3913
2	32768	-65	5	4	1	0.3793
4	32768	-80	4	4	0	0.3752
-4	32768	-95	5	7	2	0.3751
2	32768	-95	4	6	0	0.3749
0	32768	-95	4	5	1	0.3675

4.6.2 Model selection and merging

Next, these 10 design points are used to select one or more reference models that best match our observed performance. Using the design points, the average difference scores to each stored reference model are calculated and shown in Figure 4.8.

As stated in Section 4.5-B, the average difference scores are compared to a lower selection bound of 0.05 and an upper selection bound of 0.3 to decide for model selection, model merging and model building procedures. (i) If any of the average difference scores is smaller than the lower selection bound, decision is made to select a reference model. (ii) Whereas, if all the average difference scores

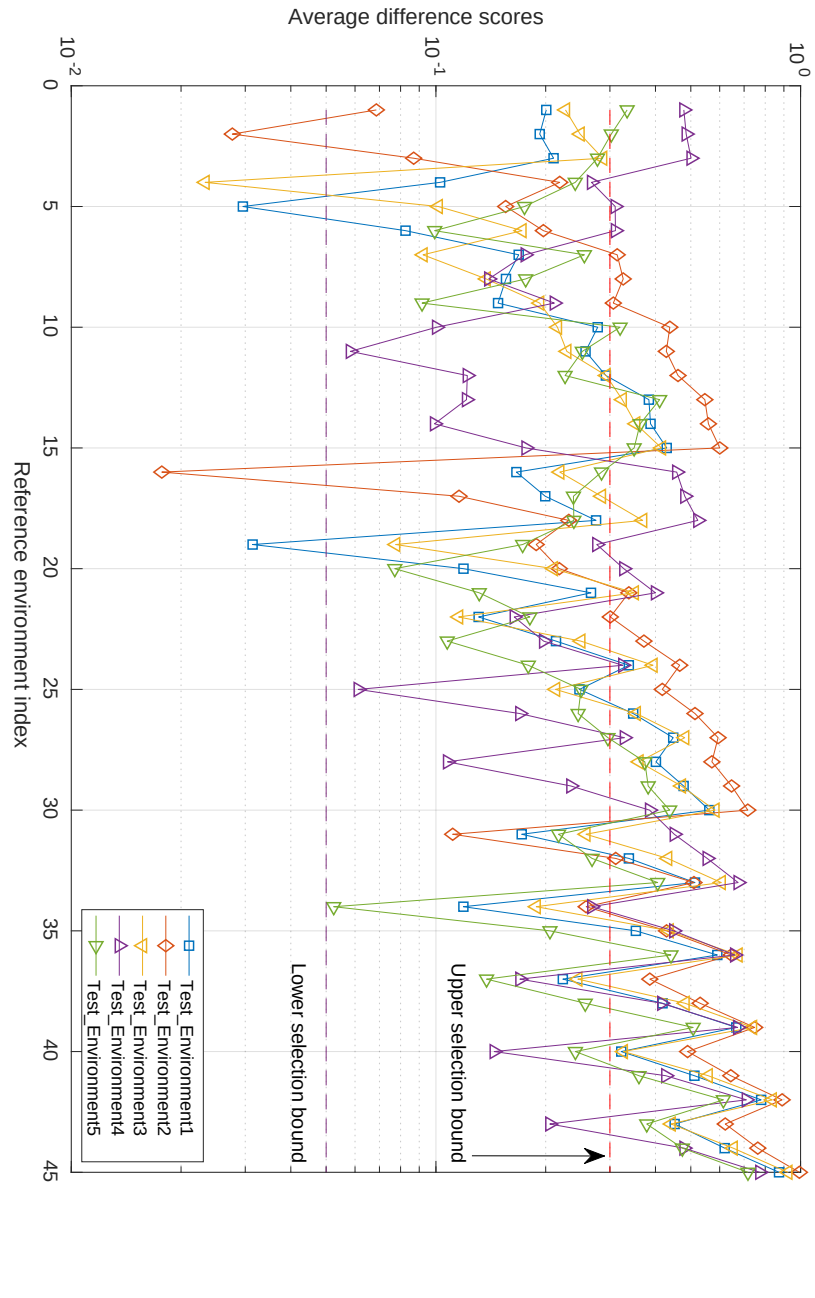


Figure 4.8: Model selection, model merging and model building decisions using average difference scores, lower selection bound and upper selection bound. A logarithmic scale is applied to the difference scores axis to better view the regions around the lower and upper selection bounds.

are above the upper selection bound, a model building procedure is initiated. (iii) Otherwise, model merging procedure is applied. Applying this procedure on Figure 4.8, we get the environment evaluation shown in Table 4.7.

Table 4.7: Environment evaluation using difference scores of Figure 4.8. Irrespective of the decision, the best selected environment is also displayed to be compared in the next section.

Test Environment		**Best Selected Environment**	**Procedure**
1	15 Nodes/decameter^2, 5%, 12 sec	15 Nodes/decameter^2, 5%, 12 sec	select
2	26 Nodes/decameter^2, 0%, 6 sec	26 Nodes/decameter^2, 0%, 6 sec	select
3	15 Nodes/decameter^2, 7%, 6 sec	15 Nodes/decameter^2, 5%, 6 sec	select
4	17 Nodes/decameter^2, 17%, 9 sec	15 Nodes/decameter^2, 15%, 12 sec	**merge**
5	37 Nodes/decameter^2, 7%, 9 sec	47 Nodes/decameter^2, 5%, 6 sec	**merge**

As expected, the first two test environments closely match existing reference models. As such, the corresponding reference models are retrieved from the cloud repository to optimize the WSN. The third test environment closely matches the performance of an existing reference model, and decides to use this model. Finally, the last two models are different from the reference models but not far enough to totally loose correlation, and thus the model merging procedure is applied to create a new model following the procedure described in Section 4.5-D. Specific to our scenario, 5 similarly performing reference models are used in the merging process, and this will give $10 = \binom{5}{2}$ model estimates. Afterwards, a representative model is linearly predicted from the 10 model estimates. Finally, the merged model is calculated by averaging 10 representative models (one for each design point) and the WSN is optimized using the merged model.

4.6.3 Validation

To verify if the system made the right decision, we create real models of the test environment that will be used for validation purposes only. The validation models are created exactly the same way a reference model is built during the offline phase. An RRSE score, defined in Section 4.3.2, is used to measure the similarity between validation vs selected models and real vs merged models, whereby a lower RRSE value indicates a good match. Table 4.8 shows the result of the model variability test, applied to all performance metrics.

The result from Table 4.8 validates the decisions made in Table 4.7 and indeed the best approaches (selection or merging) were picked all the time. Regarding the accuracy of the merged models, we see comparable results to the accuracy of selected models (average RRSE scores shown in bold color). Therefore, the merging procedure is also producing representative models. On the other hand, RRSE values from individual performance metrics have wide variations. While Latency and energy consumption metrics have predictable performances (low RRSE values), packet error rate metric is unpredictable (high RRSE values).

Table 4.8: RRSE model comparison between selected and merged models using validation models for all test scenarios. In all cases, the best approach was selected following the average column.

#	**best selected model vs validation model**				**merged model vs validation model**				**Remark**
	LAT	*PER*	*ENG*	*AVG*	*LAT*	*PER*	*ENG*	*AVG*	
1	0.17691	0.31013	0.09554	**0.19419**	0.37710	0.34230	0.26221	0.32720	Selected model performs better than merged model
2	0.13361	0.25994	0.10020	**0.16458**	0.16668	0.39691	0.11705	0.22688	Selected model performs better than merged model
3	0.19730	0.31333	0.15528	**0.22197**	0.24557	0.37674	0.08297	0.23509	Except for *ENG* metric, selected model performs better
4	0.18812	0.32559	0.16951	0.22774	0.19840	0.29205	0.12044	**0.20363**	Except for *LAT* metric, merged model performs better
5	0.35681	0.72924	0.34885	0.47830	0.15471	0.34895	0.15301	**0.21889**	Merged model performs better than selected model

4.6.4 Complexity Analysis

WSNs are designed to operate for long periods of time without human intervention. As such, most of the complexity and computationally intensive operations resides at server level rather than on the constrained devices. To this end, constrained devices (source nodes) follow a centralized approach where they send data and report messages to a gateway device (sink node). The gateway device does all the processing such as environment characterization, model selection, model merging and WSN optimization, both during the online and offline phase of Figure 4.5. The complexity of each phase is further analyzed below.

– The offline phase starts executing an initial sample of configurations and builds a surrogate model involving a Cholesky decomposition (complexity order $O(N^3)$) using matrix operations [32] and applies the FLOLA-Voronoi sampling strategy (complexity order $O(N)$) to calculate consecutive samples [24]. For the scenario considered in this paper, a typical laptop (Intel Core i5-4210U Processor) requires around 2-5 seconds to build a surrogate model of 3 objectives and calculate the next configuration points using FLOLA-Voronoi sampling strategy.

– During the online phase, the WSN continuously monitors the environment until the performance becomes sub-optimal, after which the environment is characterized using pre-determined design points. This operation has a constant complexity and does not change in time. Afterward, a selection or merging procedure is initiated depending on the similarity between the unknown and reference environments. The selection process picks a new model from the cloud repository with a constant complexity. Model merging involves performance estimation (shown in Equations 4.6, D.2 and 4.8) with a complexity order of $O(N^2) - O(N)$. Finally, optimum design parameters are configured on the WSN with constant complexity. For the scenario considered in this paper, the model selection and model merging procedures required around 18 and 24 seconds respectively using a typical laptop of an Intel Core i5-4210U Processor.

4.7 Conclusion

In this paper, we have introduced a novel method for optimizing a Wireless Sensor Network (WSN) in a dynamic environment using a cloud repository. The cloud repository is used to store multiple performance models of the WSN under the influence of a dynamic environment. Here a dynamic environment is treated as a group of static environments assuming it is slowly changing and has a long coherence time. Furthermore, the optimization procedure is divided into two phases, where i) an offline phase populates the WSN models into the cloud repository and ii) an online phase keeps the system in an optimum performance state for changes in the environment. In order to keep the performance intact, the online phase conducts a small set of designed experiments to characterize the environment and based on the result, it either selects a single model or merges multiple

models together. Selection is a simple procedure because the new environment has a representative model in the cloud repository and the WSN selects this model to optimize its performance. Model merging, on the other hand, is carried out because a representative model is not found in the cloud repository and multiple models are merged together to represent the unique environment. Furthermore, a single-hop WSN scenario is set-up to proof the concept of the designed methodology. The WSN configures 6 design parameters from the physical layer and medium access control (MAC) layer and measures 3 performance metrics. The performance metrics selected conflict towards each other and a Pareto front is used to explain the relationship. Moreover, several test environments are experimented in order to verify the operation of the online phase. Results show that models are selected when test environments are close to a reference model and models are merged together when otherwise. The merged models are further validated by comparing them to a separately built model and they are found to be close matches. Finally, the proposed approach requires the execution of only 10 experiments to optimize the network, compared to 125 experiments for state-of-the-art optimization techniques or 4800 experiments for an exhaustive search based optimization. In practical deployments, the network manager has the option to either pre-deploy the repository with a set of relevant reference models, or to automatically populate the repository over time whenever one of the managed networks encounters previously unseen network conditions. As such, our approach is very well suited for applications in which a large number of similar WSN deployments are managed by a single WSN operator.

Acknowledgment

The research leading to these results has received funding from the European Horizon 2020 Program (under Grant No. 688116 eWINE Project [https://ewine-project.eu] and 645274 WISHFUL Project [http://www.wishful-project.eu]), the FWO-SBO SAMURAI project [http://www.samurai-project.be] and in part by the U.S. National Science Foundation (NSF) under Grant No. 1421058.

References

[1] I. Ishaq, D. Carels, G. K. Teklemariam, J. Hoebeke, F. Van den Abeele, E. De Poorter, I. Moerman, and P. Demeester. *IETF standardization in the field of the Internet of Things (IoT): a survey*. JOURNAL OF SENSOR AND ACTUATOR NETWORKS, 2(2):235–287, 2013. Available from: http://dx.doi.org/10.3390/jsan2020235.

[2] M. T. Mehari, E. D. Poorter, I. Couckuyt, D. Deschrijver, G. Vermeeren, D. Plets, W. Joseph, L. Martens, T. Dhaene, and I. Moerman. *Efficient Identification of a Multi-Objective Pareto Front on a Wireless Experimentation Facility*. IEEE Transactions on Wireless Communications, 15(10):6662–6675, Oct 2016. doi:10.1109/TWC.2016.2587261.

[3] R. L. Cooper. *Sensing and Learning Channel State Information in a Dynamic Wireless Environment with Cognitive Radios and Networks*. PhD thesis, Carnegie Mellon University, 2012. Available from: http://repository.cmu.edu/dissertations/211.

[4] SGCC, NIST, IEC, S. Electric, Siemens, Haier, SAP, Hitachi, I. China, China-EPRI, and SIA. *Internet of Things: Wireless Sensor Networks*. Technical report, International Electrotechnical Commission, 2017. Available from: http://www.iec.ch/whitepaper/pdf/iecWP-internetofthings-LR-en.pdf.

[5] H. P. Shiang and M. van der Schaar. *Online learning in autonomic multihop wireless networks for transmitting mission-critical applications*. IEEE Journal on Selected Areas in Communications, 28(5):728–741, June 2010. doi:10.1109/JSAC.2010.100610.

[6] M. Pakparvar, D. Plets, E. Tanghe, D. Deschrijver, W. Liu, K. Chemmangat, I. Moerman, T. Dhaene, L. Martens, and W. Joseph. *A cognitive QoS management framework for WLANs*. EURASIP Journal on Wireless Communications and Networking, 2014(1):191, Nov 2014. Available from: https://doi.org/10.1186/1687-1499-2014-191, doi:10.1186/1687-1499-2014-191.

[7] O. Odejide, C. Bazil, and A. Annamalai. *Adaptive multiresolution modulation for multimedia traffic in dynamic wireless environment*. In 34th IEEE Sarnoff Symposium, pages 1–6, May 2011. doi:10.1109/SARNOF.2011.5876475.

[8] M. J. Hossain, P. K. Vitthaladevuni, M. S. Alouini, V. K. Bhargava, and A. J. Goldsmith. *Adaptive hierarchical modulation for simultaneous voice and multiclass data transmission over fading channels*. IEEE Transactions on Vehicular Technology, 55(4):1181–1194, July 2006. doi:10.1109/TVT.2005.863345.

[9] R. Minkara and P. Shepherd. *Optimising the location and power of wireless base stations within a dynamic indoor environment*. In 2014 Loughborough

Antennas and Propagation Conference (LAPC), pages 566–569, Nov 2014. doi:10.1109/LAPC.2014.6996453.

[10] X. Cai, L. Chen, and G. Chen. *Constructing Adaptive Indoor Radio Maps for Dynamic Wireless Environments*. In 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing, pages 41–47, Dec 2013. doi:10.1109/UIC-ATC.2013.20.

[11] A. Puschmann, M. A. Kalil, and A. Mitschele-Thiel. *A component-based approach for constructing flexible link-layer protocols*. In 8th International Conference on Cognitive Radio Oriented Wireless Networks, pages 244–249, July 2013. doi:10.1109/CROWNCom.2013.6636825.

[12] R. Edirisinghe and A. Zaslavsky. *Cross-Layer Contextual Interactions in Wireless Networks*. IEEE Communications Surveys Tutorials, 16(2):1114–1134, Second 2014. doi:10.1109/SURV.2013.101813.00023.

[13] L. Louail and V. Felea. *Latency optimization through routing-aware time scheduling protocols for wireless sensor networks*. Computers & Electrical Engineering, 56:418 – 440, 2016. Available from: http://www.sciencedirect.com/science/article/pii/S0045790616301586, doi:http://dx.doi.org/10.1016/j.compeleceng.2016.06.003.

[14] C. Wang, Q. Duan, W. Gong, A. Ye, Z. Di, and C. Miao. *An evaluation of adaptive surrogate modeling based optimization with two benchmark problems*. Environmental Modelling & Software, 60(Supplement C):167 – 179, 2014. Available from: http://www.sciencedirect.com/science/article/pii/S1364815214001698, doi:https://doi.org/10.1016/j.envsoft.2014.05.026.

[15] J. van der Herten, T. Van Steenkiste, I. Couckuyt, and T. Dhaene. *Surrogate Modelling with Sequential Design for Expensive Simulation Applications*. In Computer Simulation. InTech, 2017.

[16] M. T. Mehari, E. D. Poorter, I. Couckuyt, D. Deschrijver, J. V. Gerwen, D. Pareit, T. Dhaene, and I. Moerman. *Efficient global optimization of multi-parameter network problems on wireless testbeds*. AdHoc Networks, 29:15–31, 2015.

[17] T. Van Steenkiste, J. van der Herten, I. Couckuyt, and T. Dhaene. *Sensitivity analysis of expensive black-box systems using metamodeling*. In 2016 WINTER SIMULATION CONFERENCE (WSC), pages 578–589. Institute of Electrical and Electronics Engineers (IEEE), 2016. Available from: http://dx.doi.org/10.1109/wsc.2016.7822123.

[18] A. Forrester, A. Keane, et al. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[19] C. E. Rasmussen and C. K. Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

[20] C. Cortes and V. Vapnik. *Support-vector networks*. Machine learning, 20(3):273–297, 1995.

[21] S. Koziel, A. Bekasiewicz, I. Couckuyt, and T. Dhaene. *Efficient multi-objective simulation-driven antenna design using co-kriging*. IEEE Transactions on Antennas and Propagation, 62(11):5900–5905, 2014.

[22] M. L. Stein. *Interpolation of Spatial Data*. Springer Series in Statistics. Springer New York, 1999. doi:10.1007/978-1-4612-1494-6.

[23] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley and Sons, Inc., 7th edition, 2009.

[24] J. van der Herten, I. Couckuyt, D. Deschrijver, and T. Dhaene. *A fuzzy hybrid sequential design strategy for global surrogate modeling of high-dimensional computer experiments*. SIAM Journal on Scientific Computing, 37(2):A1020–A1039, 2015.

[25] K. Crombecq, D. Gorissen, D. Deschrijver, and T. Dhaene. *A novel sequential design strategy for global surrogate modeling*. SIAM Journal on Scientific Computing, 33(4):1948–1974, 2011.

[26] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. *Cross-Level Sensor Network Simulation with COOJA*. In Proceedings. 2006 31st IEEE Conference on Local Computer Networks, pages 641–648, Nov 2006. doi:10.1109/LCN.2006.322172.

[27] L. Sitanayah, C. J. Sreenan, and S. Fedor. *A Cooja-Based Tool for Coverage and Lifetime Evaluation in an In-Building Sensor Network*. Journal of Sensor and Actuator Networks, 5(1), 2016. Available from: http://www.mdpi.com/2224-2708/5/1/4, doi:10.3390/jsan5010004.

[28] A. Dunkels. *The ContikiMAC Radio Duty Cycling Protocol*. Technical Report ISSN 1100-3154, SICS Swedish ICT, 2011.

[29] B. Jooris, J. Bauwens, P. Ruckebusch, P. De Valck, C. Van Praet, I. Moerman, and E. De Poorter. *TAISC: a cross-platform MAC protocol compiler and execution engine*. COMPUTER NETWORKS, 107(2):315–326, 2016. Available from: http://dx.doi.org/10.1016/j.comnet.2016.03.027.

[30] F. A. C. Viana. *Things you wanted to know about the Latin hypercube design and were afraid to ask*. 10th World Congress on Structural and Multidisciplinary Optimization, May 19 -24 2013.

[31] B. Iglewicz and D. Hoaglin. *How to Detect and Handle Outliers*. American Society for Quality Control, Milwaukee WI, 1993.

[32] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

5

An Efficient Screening Method for Identifying Parameters and Interactions that Impact Wireless Network Performance

Until now, we have looked performance optimization (Chapters 2 and 3) and system characterization (Chapter 4) problems in complex wireless networks. This chapter deals with screening sensitive parameters of complex wireless networks. In recent years, wireless networks are getting more and more complex and they are starting to include a large number of configuration parameters in their design process. So far, the research community has used mostly domain knowledge to identify the most important parameters during optimization and characterization problems. Due to the increasing complexity of wireless networks, this is no longer possible and screening methods must be applied. To this end, this chapter applies a combinatorial design method called 'locating array' and a 'backtracking orthogonal matching pursuit' to analyze the screening experiment.

⋆ ⋆ ⋆

Randy Compton, Michael Tetemke Mehari, Charles J. Colbourn, Eli De Poorter, Ingrid Moerman, Violet R. Syrotiuk

Submitted to IEEE Transactions on Wireless Communications 2017

Abstract Wireless networks rely on a protocol stack to provide connectivity. Not only are the protocols at each layer reconfigurable, potential interactions arise among parameters of the protocol stack, operating system, hardware, and operating environment. Hence, there is a vital need to quickly determine the parameters and interactions that significantly impact the performance of a wireless system. In this paper, we use a new combinatorial design — a locating array (LA) — to efficiently identify the parameters impacting audio quality and RF exposure in the `w-iLab.t` wireless network testbed in Belgium. Different from many conventional techniques, the size of LAs grows logarithmically in the number of parameters. This makes LAs practical for such identification in complex engineered networks, such as `w-iLab.t`. Existing software tools, such as JMP, cannot be used to analyze the measured performance data directly as they assume a balanced structure in experimentation. Therefore, using a framework from compressive sensing, we propose a backtracking algorithm for the analysis, with pruning and backjumping to reduce the search space; this also provides robustness to noise in the wireless network. Using our analysis technique, we identify the significant parameters impacting audio quality and exposure and separately validate the results.

5.1 Introduction

Experimentation is a cornerstone of scientific advancement. Through experimentation we gather evidence to either support or refute a hypothesis. A crucial question is: *What parameters should be selected for experimentation?*

Domain experts often use their knowledge and experience to select these parameters. For example, we may think that transmission power levels impact the energy consumption of a wireless device (and we would be right!). But when the experiment and the system are complex, such as in [1, 2], an objective method that could answer this question would increase confidence in the whole experimental strategy.

While we certainly do not advocate that domain expertise be neglected, another consideration in networking is that interactions between protocols are known. Sometimes these interactions are between protocols in adjacent layers, but sometimes they are not. A famous two-way, MAC/transport layer, interaction is that TCP interpreted access delays in a wireless network due to poor signal quality as congestion, and hence responded incorrectly with congestion control [3]. A few other examples of cross-layer interactions in networking may be found in [4–9]; these are not always evident, even to domain experts. If not certain that a parameter or interaction affects performance, a knowledgeable domain expert may prefer to ignore it in order to reduce experimental cost. Yet this ensures that its impact on the performance is never observed.

In this paper, our interest is to use experimentation to identify the significant parameters and two-way interactions impacting wireless network performance. These are termed *screening experiments* [10]. To cope with the complexity in engineered network systems, screening *should be* an important first step before

conducting the intended experiment. The purpose of typical experiments include building an empirical *model* – a function of the parameters and interactions – to predict a system's performance, optimize system performance, or improve the robustness of the system to operating conditions, among many others.

In this paper, we study a Wi-Fi conferencing scenario, and our ultimate objective is to jointly optimize two responses: To maximize the quality of the audio for participants while minimizing their exposure to radio frequencies. The specific system we use for experimentation is the advanced IMEC `w-iLab.t` heterogeneous wireless testbed [11] located in Belgium.

In the field of *designed experiments*, it is considered impractical to experiment with more than about 10 parameters [10, 12]. Most protocols of a typical TCP/IP stack have at least 5-10 configurable parameters. Thus, there can be 25-50 parameters to vary in experimentation without considering wider aspects of the system. Indeed, parameters of the operating system (*e.g.*, kernel version, buffer sizes, queuing disciplines), the hardware (*e.g.*, chipset, drivers), and the operating environment (*e.g.*, indoors, outdoors) may also impact the network performance.

As we will see, for engineered network systems many traditional screening experiments are infeasible. This is because the *experimental design*, an array describing the tests in the experiment, is too large. While supersaturated designs (SSDs) can screen efficiently, their focus is only on identifying parameters [10, 13, 14]. In networking, ideally we are interested in a screening method that is also capable of identifying two-way interactions, because some parameters may be significant only as a result of their involvement in a significant interaction.

To address this need, we use a new combinatorial design called a *locating array* (LA) for screening. Locating arrays grow logarithmically in the number of parameters [15]. Therefore, they have the potential to screen efficiently a far larger number of parameters and two-way interactions than previous approaches. Not surprisingly, there is a trade-off: One reason locating arrays are small is because they are unbalanced. Balance relates to how equally a parameter or interaction is measured in the design. Most analysis tools, such as JMP [16], assume the underlying experimental design is balanced, or nearly balanced. Because locating arrays may be highly unbalanced, we are unable to apply the standard analysis techniques to recover the significant parameters and interactions from the measured performance.

Thus in addition to introducing locating arrays as a new screening methodology for engineered network systems, another contribution of this paper is to propose a new analysis technique to accomplish the identification. We use the framework of compressive sensing to recover a model whose terms correspond to the parameters and two-way interactions that significantly impact the performance. However, because measurements of real systems are noisy, *any* recovery approach could make an error in term identification, which could impact the identification of subsequent terms. Rather than recover a single model, we use a search tree to recover a number of alternative models, providing an analyst with flexibility in understanding the performance of a complex system. Because the tree is too large to search exhaustively, a scoring function, pruning, and backjumping are combined

to reduce the search space.

Recall that our original motivation is to screen, so that we can conduct a follow-on experiment having confidence that we will be varying the parameters and interactions that affect performance significantly. Because of the efficiency of locating arrays we do not need to reduce the number of parameters considered in screening a priori. Our methodology is an objective and efficient way to screen large design spaces, thereby reducing the likelihood that significant parameters or two-way interactions are missed in follow-up experimentation.

In summary, this paper makes the following major contributions:

1. We propose the use of a locating array as an efficient design for screening complex engineered networks. We conduct an experiment using a locating array to screen 24 parameters and their two-way interactions, where each parameter has from 2-5 values, in a conferencing scenario set up in the `w-iLab.t` wireless network testbed. The parameters are taken from the kernel's IP and UDP stacks, the Wi-Fi card driver, the audio codec, and a source of RF interference. The resulting LA has only 109 tests; this compares with $\approx 11.6 \times 10^{12}$, an infeasible number of tests in a full-factorial design.

2. We propose a tree-based search strategy to analyze the performance measures collected from a screening experiment whose design is a locating array. The branches of the tree are determined by a 'heavy-hitters' compressive-sensing approach [17], and techniques are applied to reduce the search space. Because measurements of real physical systems are noisy, our analysis methodology may identify alternate explanations for the performance measured.

3. We apply our analysis technique to the measurements collected from `w-iLab.t` to identify the significant parameters and two-way interactions impacting the mean opinion score (MOS), and the RF exposure. The results are validated using a fractional-factorial experimental design. To our knowledge, this is the first time an LA has been used for screening in a physical system.

Often screening will confirm initial observations by domain experts, and this can result in the mistaken belief that screening is not needed in these situations. Similarly, in some environments two-way interactions may only involve its significant main effects, and hence after the fact one might argue that a simpler screening design would have been sufficient. However, the purpose of screening is precisely to ensure that we do not overlook significant main effects or interactions, and we have no *a priori* guarantee that domain experts or simpler screening methods will not discount them.

To enable reproducibility of our results, we provide all the code, scripts, and tools necessary to run the screening experiment on the `w-iLab.t` wireless network testbed and analyze the measurements collected, or to analyze our data sets directly [18].

The rest of this paper is organized as follows. Section 5.2 overviews traditional designs used for screening and their analysis, and provides a definition of a locating array. Section 5.3 presents the proposed methodology for analyzing the results of experimentation using LAs. This is followed by the details of the experimental set-up in Section 5.4, and the results of applying the analysis methodology to the measured performance data, and their validation, in Section 5.5. Finally, we conclude in Section 5.6, suggesting several opportunities for future research.

5.2 Screening Designs

In this section, we begin by introducing some terminology and notation. In addition to traditional screening designs, two combinatorial designs — covering arrays and locating arrays — are formally defined.

5.2.1 Definition of a Test, an Experiment, and a Design

Suppose that the system under study has k parameters, $P_1, \ldots, P_k$, and that each parameter P_j has a set $V_j = \{v_{j,1}, \ldots, v_{j,\ell_j}\}$, of ℓ_j possible values. A *test* is an assignment of a value from V_j to P_j, for each parameter $j = 1, \ldots, k$. An *experimental design* (or, design for short) is a collection of tests.

When a design has N tests (or, *size* N), it is represented by an $N \times k$ array $A = (a_{ij})$ in which each row i corresponds to a test and each column j to a parameter; the entry a_{ij} specifies the value assigned to parameter j in the ith test. When run on the system, a test results in the measurement of one or more performance metrics. An *experiment* consists of running each test in the design.

The order in which the tests in an experiment are conducted is randomized so as not to introduce any dependencies on the run order. This step is important when the experiment is run on a physical system. Repetitions (replicates) of the experiment are run to determine the variance in the measured performance metrics, quantifying the noise in the system.

5.2.2 A Running Example

We introduce an example system used to explain properties of experimental designs in this section, and to illustrate the analysis methodology in Section 5.3. Consider a system with 4 parameters. Parameters A and B each have two values $V_A = V_B = \{0, 1\}$, while parameters C and D each have three values $V_C = V_D = \{0, 1, 2\}$. For short, we use the notation P_ℓ to denote parameter P set to the value equal to ℓ. For example, A_1 denotes A set to 1.

5.2.3 Traditional Screening Designs

A *full-factorial design* is the original screening design. It has tests that include all possible combinations of values of each parameter P_j across all k parameters [19].

Its size is equal to the product of $|V_j|$ for each parameter $j = 1, \ldots, k$. For the running example, a full-factorial design has $2^2 \times 3^2 = 36$ tests, and in general the number of tests grows exponentially in the number of parameters. From an *analysis of variance* (ANOVA) of the data collected from a full-factorial design all significant t-way interactions for $t = 1, \ldots, k$ can be identified. Traditionally, identifying significant main effects and two-way interactions, *i.e.*, $t = 1, 2$, have been of primary interest, as higher-order interactions tend to be rare and of lesser effect [10, 20].

More recently, *supersaturated designs* (SSDs) have been introduced to only identify significant parameters [13]. This focus comes from relying on an assumption of *strong heredity*, the condition that a significant two-way interaction has its component main effects also significant. However, strong heredity is not universally valid in real-world experiments [20].

There are many criteria used to optimize the size of supersaturated designs, with the aim of identifying fewer irrelevant parameters and/or obtaining more confidence in the parameters that are identified [21]. One popular one, *D-optimality*, minimizes the size of the joint confidence region for the model coefficients. Supersaturated designs require advanced analysis methods [14].

A problem with most traditional screening designs is that they do not ensure it is possible to estimate the effects of all interactions, or even that they all occur. If a significant assignment of values to parameters is missing from a design, it is impossible to determine this from the data collected in the experiment. Covering arrays address this issue.

5.2.4 Covering Arrays

Call an assignment of values to any subset $t \leq k$ of the parameters a *t-way interaction*. A *covering array* of strength t, is an $N \times k$ array in which for every $N \times t$ subarray, each t-way interaction is *covered* (*i.e.*, occurs) in at least one test [22]. A covering array of strength two on the four parameters of the running example is given in Table 5.1(a). Nine tests suffice to cover all 37 of the two-way interactions. For example, for the two-way interaction A_0C_2, we find at least one test that covers it, in this case, test 5.

Covering arrays have recently been introduced as experimental designs into the software tool JMP [16]. Analysis is simplified when a design is *balanced*, when each t'-way interaction, $t' \leq t$, is covered the same number of times. In general, covering arrays are close enough to balanced that the traditional approaches for analysis succeed.

While a covering array of strength t covers all t-way interactions, it does not ensure that it is possible to distinguish the influence of different t-way interactions. For example, if the performance metric measured for test 5 is different from the other rows, it is not possible to determine which of the three two-way interactions A_0B_1, A_0C_2, and C_2D_1 is responsible, because each one appears only in test 5. Locating arrays extend covering arrays to address this very issue.

Table 5.1: For the running example: (a) A covering array A_{CA} of strength 2; (b) a $(1,2)$-locating array A_{LA}.

(a)

Test	A	B	C	D
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	2
5	0	1	2	1
6	1	0	2	2
7	1	1	0	2
8	1	1	1	1
9	1	1	2	0

(b)

Test	A	B	C	D
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	0	2	2
6	0	1	0	2
7	0	1	1	2
8	0	1	2	0
9	0	1	2	1
10	1	0	0	2
11	1	0	1	2
12	1	0	2	0
13	1	0	2	1
14	1	1	0	0
15	1	1	0	1
16	1	1	1	0
17	1	1	1	1
18	1	1	2	2

5.2.5 Locating Arrays

A (d,t)-*locating array* [15] is a covering array of strength t with an additonal property: Any set of d interactions each involving t parameters can be distinguished from any other such set by appearing in a distinct set of tests. If a design satisfies this definition it has the (d,t)-*locating property*.

More precisely, for array $A = (a_{ij})$ and t-way interaction T, define $\rho(A,T)$ as the set of tests (or, rows) of A in which T is covered. For a set $\mathscr{T}$ of t-way interactions, $\rho(A,\mathscr{T}) = \cup_{T\in\mathscr{T}}\rho(A,T)$.

Let $\mathscr{I}_t$ be the set of all t-way interactions for an array, and let $\overline{\mathscr{I}_t}$ be the set of all t-way interactions of strength *at most* t. Consider a t-way interaction $T \in \overline{\mathscr{I}_t}$ of strength less than t. Any t-way interaction T' of strength t that contains T necessarily has $\rho(A,T') \subseteq \rho(A,T)$. Call a subset $\mathscr{T}'$ of t-way interactions in $\mathscr{I}_t$ *independent* if there do not exist $T, T' \in \mathscr{T}'$ with $T \subseteq T'$.

Definition 1 ([15]). *An array A is (d,t)-locating if whenever $\mathscr{T}_1, \mathscr{T}_2 \subseteq \overline{\mathscr{I}_t}$ and $\mathscr{T}_1 \cup \mathscr{T}_2$ is independent, $|\mathscr{T}_1| \leq d$, and $|\mathscr{T}_2| \leq d$, it holds that*

$$\rho(A,\mathscr{T}_1) = \rho(A,\mathscr{T}_2) \Leftrightarrow \mathscr{T}_1 = \mathscr{T}_2.$$

The covering array A_{CA} in Table 5.1(a) does not have the $(1,2)$-locating property because the set $\mathscr{T} = \{A_0B_1, A_0C_2, C_2D_1\}$ has $\rho(A_{CA},\mathscr{T}) = \{5\}$, *i.e.*, each of these two-way interactions in $\mathscr{T}$ appears only in test 5. However, we can construct a $(1,2)$-locating array A_{LA} for the running example with 18 tests (see Table 5.1(b)). Now we see that for each two-assignment in $\mathscr{T}$ there

is a row that distinguishes it from the others: $\rho(A_{LA}, A_0B_1) = \{6, 7, 8, 9\}$, $\rho(A_{LA}, A_0C_2) = \{5, 8, 9\}$, and $\rho(A_{LA}, C_2D_1) = \{9, 13\}$.

5.2.6 Unbalance

If unbalance arises from a few missing measurements in an experiment with an underlying balanced design (*e.g.*, occasionally some wireless nodes do not report), it may be possible to fill in the missing values so as to support an analysis method that requires balance [23].

While supersaturated designs (SSDs) are unbalanced, the optimality criterion used in their construction often ensures approximate balance. If a design is close enough to being balanced, techniques for balanced designs are often used directly or with slight modifications under the assumption that the imbalance does not significantly affect the outcome. For example, when analyzing an SSD, JMP estimates the p-values via a simulation that assumes balance and can give incorrect results otherwise [16]. Such techniques are unsuitable when designs are highly unbalanced, and are hence unsuitable for use with many locating arrays.

Techniques from the field of compressive sensing, such as LASSO, have also been used to analyze SSDs [24]. Next we show how we use these techniques in analyzing locating arrays.

5.3 Analysis of Locating Arrays

The highly unbalanced nature of locating arrays requires a non-traditional approach to the analysis of the measured performance metrics. For this we use the framework of compressive sensing, which uses *orthogonal matching pursuit* (OMP) to identify significant parameters and two-way interactions that impact performance. A trace of OMP is provided for the running example from Section 5.2 to explain the methodology. However, to cope with noise in measured responses that may lead OMP astray, we propose a search tree algorithm, with branches determined by OMP, and techniques to reduce the search space.

5.3.1 Orthogonal Matching Pursuit

Suppose we have an $N \times 1$ vector of measurements from running an experiment using an $N \times k$ $(1, 2)$-locating array A as the screening design. A compressive sensing matrix corresponding to A has a column for each possible term — parameter or two-way interaction — that could be significant. We utilize *orthogonal matching pursuit* (OMP) [17] to iteratively recover the terms of a model. In our setting, the terms of this model are those that have significant impact on the measured performance of our system. We choose OMP over other compressive sensing methods because it is incremental.

This resembles a 'heavy-hitters' approach used in compressive sensing [25]. An experimental design with the $(1, 2)$-locating property suffices because in each

iteration we recover one term; we are able to do so provided that there is a unique strongest parameter or two-way interaction.

algorithm 1: Orthogonal Matching Pursuit [17] : $\mathrm{OMP}(terms, M, data)$

Input: List of candidate terms, compressive sensing matrix, vector of responses
Output: Model (list of pairs of coefficients and terms)

1: $model \leftarrow$ [(mean of data, intercept term)]
2: $residuals \leftarrow data - model$
3: **while** the stopping criterion is not yet met **do**
4: $\quad i \leftarrow \arg\max_i |\widehat{M_i} \cdot \widehat{residuals}|$
5: $\quad model \leftarrow \mathrm{LLS}(\mathrm{terms}(model) \cup [terms_i], data)$
6: $\quad residuals \leftarrow data - model$
7: **end while**
8: **return** $model$

In OMP (see Algorithm 1), we maintain a current model and a current residual vector. The model is initialized as an intercept term with coefficient equal to the mean of the measured performance data, while the residuals are equal to the data minus the model's predictions of it. In the algorithm, we use square brackets to denote a list, and $\cup$ to denote list concatenation.

In each iteration, we select a term to add to the model based on the dot product of the residuals (scaled to a unit vector) with each candidate term's column in the compressive sensing matrix (also scaled to a unit vector). The term yielding the highest-magnitude dot product is added to the model, after which linear least squares (LLS) [23] is used to update the model coefficients. The residuals are then recalculated. These steps are repeated until a stopping criterion is met [17]. In our case, we make use of a limit on the number of terms and a desired R^2 (coefficient of determination) threshold.

A dot product is used because a dot product of zero indicates a complete lack of correlation between a term and the data, a dot product of 1 indicates perfect correlation, and a dot product of -1 indicates an anti-correlation, provided the vectors are normalized. We normalize the vectors before computing the dot product to ensure this interpretation. Also, the dot product is linear in each argument, which fits well with using linear least squares to fit because it leaves the dot product with each already-selected term's column equal to zero. Additionally, because each term selected gets a signed coefficient, term selection should only depend on the direction of the vector associated with the term. We can think of the terms as living in the projective space formed by identifying antipodal points on the unit sphere and inheriting its metric from the sphere. Then the goal is to find the closest term vector to the residual vector. Minimizing the distance is equivalent to maximizing the absolute value of the dot product, because the distance is equal to $\cos^{-1}|u \cdot v|$ for unit vectors u and v. This could be replaced with a different distance function depending on the distribution of the data to be modelled by changing line 4 in Algorithm 1.

5.3.2 Compressive Sensing Matrix

A *compressive sensing matrix* has rows corresponding to the tests in a design, and columns corresponding to terms that may be included in the model [26]. We construct a ± 1 compressive sensing matrix $M = (m_{ij})$ from the locating array as follows: $m_{ij} = +1$ if term j is present in test i of the locating array, and $m_{ij} = -1$ otherwise. The reason for using a ± 1 matrix instead of a binary matrix is to ensure that negating a column does not change the absolute value of its dot product with the residuals.

For example, for the locating array in Table 5.1(b) is repeated Table 5.2 for ease of understanding its corresponding compressive sensing matrix. It includes a column for each possible parameter and two-way interaction. For example, the interaction BC can take on six values because B has two levels and C three; the third column of BC corresponds to the two-way interaction B_0C_2. Because this two-way interaction is present only in tests 5, 12, and 13 in the locating array, the column value in the compressive sensing matrix is set to $+1$ only in these rows.

One obstacle to verifying the suitability of locating arrays for OMP in our setting is that existing compressive sensing recoverability tests (such as [27]) do not take the structure of the matrix obtained from a locating array into account, which may result in a worse guarantee than is actually achievable. There are equivalent ways of representing some models and no reason other than sparsity to prefer one representation over another because each yields the same result; a recoverability test treats this as an inability to select a unique best model.

To better understand this, consider a parameter X with three values $\{0, 1, 2\}$. Since it takes exactly one of its values in each test, we have that $X_0 + X_1 + X_2 = +1 + -1 + -1 = -1$. As a result, the model $a + bX_2$ is equivalent to $(a - b) - bX_0 - bX_1$. The first model has two terms while the second has three; the preference is for the sparser model, *i.e.*, the one having fewer terms. If we were to eliminate the term with X_2 from the compressive sensing matrix, only the second model would be possible. Therefore, we retain all three terms in the compressive sensing matrix in order to ensure the sparsest model is available to be chosen.

5.3.3 Trace of OMP on Running Example

To demonstrate recovery in the presence of noise using OMP, we generate a noisy data set for our running example. To generate the data set, we evaluate the model $-\frac{1}{2} + \frac{1}{2}C_2 - A_1B_1$, and add independent Gaussian noise with mean 0 and variance 0.1, for each row for the locating array in Table 5.2, to give the column 'Input' in Table 5.3. All values in the trace are given to 4 decimal places.

We show the iterations of Algorithm 1 (OMP) in Table 5.3, along with the R^2 of the model so far and the term selected in each iteration at the bottom. We begin by initializing the residual vector by subtracting the mean of -0.2172 from the each value in column 'Input' to obtain the column 'Centered.' We then compute the dot product with each column of the compressive sensing matrix, with both

Table 5.2: LA with corresponding compressive sensing matrix for the running example.

Compressive Sensing Matrix

A	B	C	D	AB	AC	AD	BC	BD	CD
0 1	0 1	0 1 2	0 1 2	0 0 1 1	0 0 1 1 1	0 0 1 1 1	0 0 1 1 1	0 0 1 1 1	0 0 1 1 1
				0 1 0 1	0 1 2 0 1	0 1 2 0 1	0 1 2 0 1	0 1 2 0 1	0 1 2 0 1
+	-	+ - -	- - -	- - - -	- - - - -	- - - - -	+ - - - -	+ - - - -	- - - - -
+	+	- + -	- + -	- - - -	- - - - -	+ - - - -	- + - - -	- - + - -	- - - - -
+	+	+ - -	- - +	- - - -	- - - - -	- + - - -	- - + - -	- - - + -	- - - - -
+	+	- + -	- + -	- + + -	- + + - -	- - - + -	+ - - - -	- + - - -	- - - - -
+	+	- - +	- - -	- - - +	- - - + -	- - - - +	- - - + -	- - - - +	- - - + -
+	+	- - +	+ - -	- - + -	- - - + -	+ - - - -	- - - + -	- - - - +	- - - + -
+	+	+ - -	- + -	- + - -	- + - - -	- - + - -	- + - - -	- + - - -	- - - - +
+	+	+ - -	- - +	- + - -	- + - - -	+ - - - -	- + - - -	- + - - -	- - - - +
-	-	- + -	- + -	+ - - -	+ - - - -	- + - - -	- - + - -	- - + - -	- - - + -
-	-	- + -	+ - -	+ - - -	+ - - - -	- - + - -	- - + - -	- - + - -	- - - + -
-	-	- - +	- - +	+ - - -	- - + - -	- - - + -	- - - + -	- - - + -	- - - + -
-	-	- - +	+ - -	+ - - -	- - + - -	- + - - -	- - - + -	- - - + -	- - - + -
+	+	+ - -	- + -	- - + -	- - - + -	- - - - +	+ - - - -	- - - - +	- - - - +
+	+	+ - -	- - +	- - + -	- - - + -	- - + - -	+ - - - -	- - - - +	- - - - +
+	+	- + -	- + -	- - + -	- - - + -	- + - - -	- - + - -	- - - + -	- - - - +
+	+	- + -	+ - -	- - + -	- - - + -	+ - - - -	- - + - -	- - - + -	- - - - +
+	+	- - +	- - +	- - + -	- - - + -	- - - - +	- - - + -	- - - - +	- - - - +
+	+	- - +	+ - -	- - + -	- - - + -	+ - - - -	- - - + -	- - - - +	- - - - +
-	-	+ - -	- + -	- - - +	- - - - +	- + - - -	- - - - +	- - - - +	- - - - +
-	-	+ - -	- - +	- - - +	- - - - +	- - + - -	- - - - +	- - - - +	- - - - +
-	-	- + -	- - +	- - - +	- - - - +	- - - + -	- - - - +	- - - - +	- - - - +
-	-	- + -	+ - -	- - - +	- - - - +	+ - - - -	- - - - +	- - - - +	- - - - +
-	-	- - +	- - +	- - - +	- - - - +	- - - + -	- - - - +	- - - - +	- - - - +
-	-	- - +	+ - -	- - - +	- - - - +	+ - - - -	- - - - +	- - - - +	- - - - +

Locating Array

Test	A	B	C	D
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	0	2	2
6	0	1	0	2
7	0	1	1	2
8	0	1	2	0
9	0	1	2	1
10	1	0	0	2
11	1	0	1	2
12	1	0	2	0
13	1	0	2	1
14	1	1	0	0
15	1	1	0	1
16	1	1	1	0
17	1	1	1	1
18	1	1	2	2

the residuals and the matrix columns scaled to unit vectors. For the A_1B_1 column, we get approximately 0.8088. The column with B_1D_2 (which we know is irrelevant) yields 0.0276, and the column with C_2 (which is chosen later) has the value 0.5448.

Among all columns, the dot product with A_1B_1 has the highest absolute value, so it is selected, and the model is updated to $-0.6973 - 1.0801A_1B_1$ with the residuals in the column 'Iteration 1.' At this point, the best choice of column is C_2 with dot product 0.9197; its dot product has changed because the residuals have changed. After the term C_2 is added to the model, the coefficients ($-0.4929 - 0.9893A_1B_1 + 0.4919C_2$) and residuals ('Iteration 2') are again updated.

Table 5.3: Example noisy data set based on $-\frac{1}{2} + \frac{1}{2}C_2 - A_1B_1$ processed by Algorithm 1.

Input	Centered	Iteration 1	Iteration 2
$+0.0132$	$+0.2304$	-0.3696	$+0.0087$
$+0.0044$	$+0.2216$	-0.3784	-0.0001
$+0.0906$	$+0.3078$	-0.2922	$+0.0861$
$+0.0012$	$+0.2184$	-0.3816	-0.0033
$+0.9056$	$+1.1228$	$+0.5228$	-0.0826
-0.0018	$+0.2154$	-0.3846	-0.0063
$+0.0246$	$+0.2418$	-0.3582	$+0.0201$
$+1.1360$	$+1.3532$	$+0.7532$	$+0.1478$
$+0.9032$	$+1.1204$	$+0.5204$	-0.0850
$+0.0519$	$+0.2691$	-0.3309	$+0.0474$
-0.0990	$+0.1182$	-0.4818	-0.1035
$+1.0229$	$+1.2401$	$+0.6401$	$+0.0347$
$+0.9241$	$+1.1413$	$+0.5413$	-0.0641
-1.9484	-1.7312	-0.1710	$+0.0257$
-2.0221	-1.8049	-0.2447	-0.0480
-2.0114	-1.7942	-0.2340	-0.0373
-1.9638	-1.7466	-0.1864	$+0.0103$
-0.9411	-0.7239	$+0.8363$	$+0.0493$
R^2	0.0000	0.8152	0.9966
Term	A_1B_1	C_2	(complete)

We stop because the model has $R^2 \approx 0.9966$, indicating a very good fit. While the final model has different coefficients than the original model due to the added noise, we have recovered all the parameter and two-way interaction terms that were present in the original model.

5.3.4 Backtracking OMP

Because OMP can only add terms to a model, if it finds other terms that explain one away, it cannot delete the term, only update its coefficient to be negligible. But subtracting the effect of an insignificant term might prevent finding a significant term by moving the residuals far from it in the projective space. Although the term whose vector is closest to the residuals is our best guess for which term has the greatest effect, the closer the residuals are to being the same distance from two

or more terms, the more likely it is that they could have been pushed across the selection decision boundary by terms of less effect and/or noise.

algorithm 2: Backtracking OMP : $\mathrm{BT\_OMP}(terms, M, data, model)$

Input: List of candidate terms, compressive sensing matrix, vector of responses, model
Output: Lazy list of models that meet the stopping criterion, labelled with safety values

if stopping criterion met **then**
  return $[(model, \text{safety} = 0)]$
end if
$residuals \leftarrow data - model$
calculate R^2 for $model$
$options \leftarrow terms$
sort $options$ by $i \mapsto \mathrm{dist}(M_i, residuals)$
remove terms from $options$ that are linear combinations of $\mathrm{terms}(model)$
$newModel \leftarrow \mathrm{LLS}(\mathrm{terms}(model) \cup [options_1], data)$
$submodels_1 \leftarrow \mathrm{BT\_OMP}(terms, M, data, newModel)$
for all $i \leftarrow 2, ..., |options|$ **do**
  $newModel \leftarrow \mathrm{LLS}(\mathrm{terms}(model) \cup [options_i], data)$
  $submodels_i \leftarrow \mathrm{BT\_OMP}(terms, M, data, newModel)$
  $\text{safety} = \frac{\mathrm{dist}(M_i, residuals) - \mathrm{dist}(M_{i-1}, residuals)}{1 - R^2}$
end for
$models \leftarrow$ merge $submodels_i$ for $i = 1, ..., |options|$
return $models$

To counter this problem, we propose a backtracking search algorithm BT_-OMP (Algorithm 2) to consider alternatives for the choices it makes using OMP. We invoke Algorithm 2 with four parameters: The candidate terms for the parameters and two-way interactions under consideration, the compressive sensing matrix (see Section 5.3.2), the measured performance data, and an initial model, which consists of just the intercept term as in Algorithm 1. BT_OMP returns a lazily-generated list of models, from which as many models as desired can be extracted. Taking the best model out of several can never do worse than OMP, because the first model returned is always the same as that returned by OMP. It can do better than OMP if BT_OMP selects a better option after reconsidering some choice.

In the implementation of BT_OMP, we first check whether the stopping criterion has been met, and if it has, we return the input model. Otherwise, we compute the residuals and R^2 for the input model. We then sort the candidate terms by increasing distance between the corresponding column of the compressive sensing matrix and the residuals of the current model; this orders the terms by 'strength' of effect. We then call BT_OMP recursively on each model formed by adding a term to the current model. We relabel the first model in each returned list of models (other than $submodels_1$) with its 'safety,' which we compute as the difference in distance (computed by the function dist) from the residuals between the term we chose for it and the previous term at this level, divided by $1 - R^2$. The closer the residuals are to being the same distance from two terms, the more uncertain we are that we chose the correct one. The reason for the division by $1 - R^2$ is because

changing a term can potentially affect not only its direct contribution to the model but also all subsequently-chosen terms, and we want the safety to decrease (but never to zero) if the choice can affect more of the total variance. Another heuristic with these properties could potentially be used in place of this definition of safety by changing line 14.

We then combine all the lists of models while preserving the ordering. The 'merge' operation in line 16 is as follows: Until all elements of all lists have been merged, we consider the first element of each list, select the one with minimal safety, remove it from that list, and add it to the merged output. Because we never relabel the first option in $submodels_1$, and the initial safety value for all models is 0 due to the base case (line 2), this first remains and still has label 0 after merging. To avoid generating all possible models, our implementation relies on *lazy evaluation*: Only the information necessary to produce the models the caller requests is computed.

The merge operation serves to reconsider the choice with lowest safety value instead of the most recent choice. This type of backtracking is similar to the method of backjumping used in SAT solvers [28]. The merge operation can also reconsider a choice from an earlier branch of the tree of candidate models, rather than just the choices made in reaching the most recently selected model, which ensures that no model is excluded from consideration. If we are uncertain a choice is correct, we are also uncertain that changing it is correct, so it is plausible that revisiting a choice on some earlier branch may lead to an overall improvement.

As an optimization, two models are *equivalent* if the linear space spanned by their terms is equal. Extending such models with the same terms results in further equivalent models, so the entire subtrees are identical. If at any point Algorithm 2 is called with a model equivalent to one on which it has previously been called, we can immediately return an empty list of models.

Returning to our example in Section 5.3.3, we now trace the operation of BT_-OMP. We show the R^2 value for the first few models explored, together with the terms chosen to obtain them, in Fig. 5.1. Solid edges have been used in building models, while dashed edges have not yet been explored. While R^2 values are shown for unexplored models, these would not actually be computed until the corresponding subtrees were expanded. Thin-edged nodes are models that are not returned, while thick-edged nodes are models that satisfy the stopping criterion and thus are returned by the algorithm. Finally, nodes with a dashed border are models that are equivalent to a previously-encountered model and so are excluded from consideration. The edges are labelled by the selected term and its distance from the residuals of the parent node's model. Also, edges corresponding to selections after the first have a safety value. The first selection at each node does not have a safety value because there is no previous choice with which to compare it.

Beginning at the root (node 0), we select A_1B_1 with distance 0.6287. The next-best option for that selection is C_2 with distance 0.9947. The difference in the distances is 0.3660, which we divide by $1 - R^2 = 1$ to obtain the safety, where $R^2 = 0$ since no terms have been selected previously. At the second step at node 1, we select C_2 with distance 0.4033, with B_1C_2 being the next best with distance

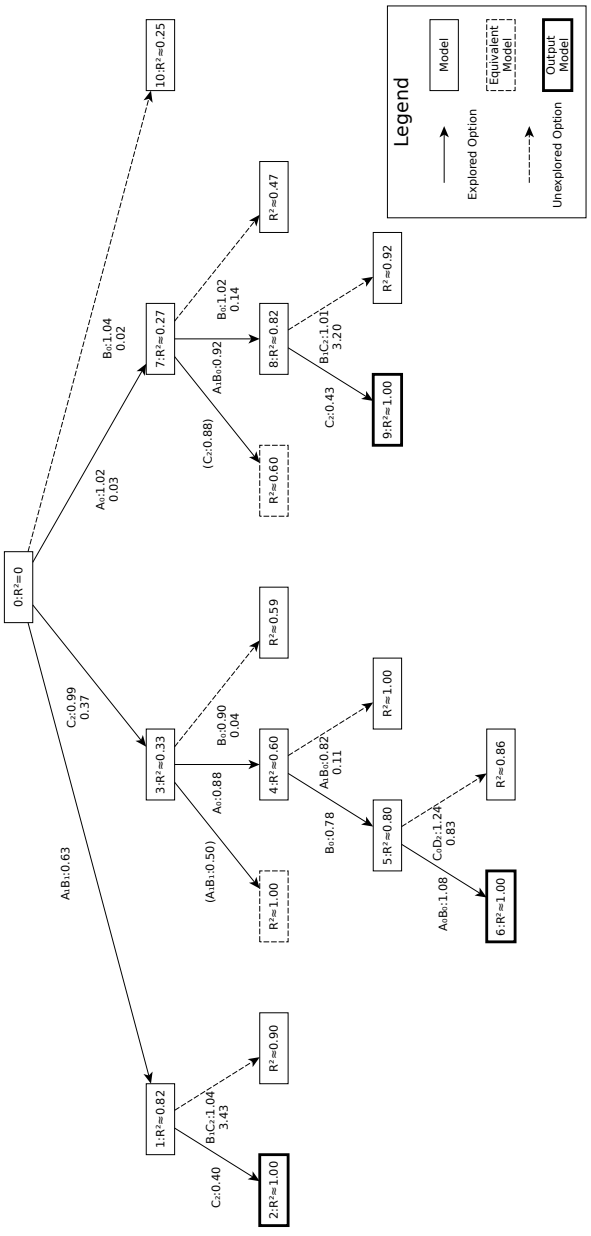


Figure 5.1: The first few models explored by Backtracking OMP applied to the running example. Nodes are labelled with R^2 for that model, while edges are labelled with the chosen term, its distance from the residuals, and its safety (when applicable). Nodes are numbered in the order in which they are encountered in the search.

1.0370. The difference in distance in this case is 0.6337, and $1 - R^2 \approx 0.1848$, yields a safety of 3.4285. This is greater than 0.3660, so we reconsider the first selection to get an alternative model. In this case, because the next option for the first selection happens to be the term selected in the second iteration, the model would in fact be the same model with the terms reordered, so we skip it and proceed to the next model. The next term selected (at node 3) is A_1 with distance 0.8773, obtaining $R^2 \approx 0.6060$ at node 4; this model eventually finishes at node 6 with $R^2 \approx 0.9968$. To find the next model, we again consider the next option for each selection made so far. At the selection made beneath node 0, we find that the third-best option, A_0, has a safety of 0.0272. At node 1, we still have the best (and only) option at that level, with safety 3.4285. At node 3, the best option is B_0 with a safety of 0.0398, so the closest decision is again at the root, and we select the term A_0 there. Continuing in this manner, we obtain another model at node 9, and at node 10 we would begin the path to a fourth model.

Even on this small example, we cannot do an exhaustive search: The root has 45 children, of which 39 have 44 children in turn, and the other 6 have 43 children. Thereafter the branching factor decreases, because some models meet the stopping criterion and others are equivalent to earlier models in our search order. The maximum depth of the tree is 16, because the rank of the compressive sensing matrix is 17.

5.4 Experimental Set-up

We now describe the details of the experiment we conducted, including the testbed used, the scenario, the parameters and performance metrics selected, and the locating array used as the experimental design.

5.4.1 The `w-iLab.t` Testbed

The IMEC `w-iLab.t`, located in Zwijnaarde, Belgium, is an advanced testbed that is used to perform heterogeneous wireless experimentation [11]. It is pseudo-shielded from external interference and is equipped with various wireless technologies, including IEEE 802.11, IEEE 802.15.4, Bluetooth dongles, Software Defined Radios (SDRs), LTE femto cells, and others. The `w-iLab.t` testbed is part of Emulab and uses the cOntrol Management Framework (OMF) for resource allocation, hardware and software configuration, and the orchestration of experiments. Finally, measurement data from each test is collected and stored in a central database over a wired control network for further processing.

5.4.2 Wi-Fi Conferencing Scenario

As a representative use case, a large-scale wireless conferencing scenario is considered. A high-level representation of the Wi-Fi conferencing scenario created in the `w-iLab.t` testbed is shown in Fig. 5.2. It is composed of a speaker node

broadcasting voice traffic over a Wi-Fi network and listener nodes receiving and playing the transmitted packets. The speaker can configure 24 different parameters (described in Section 5.4.3) that may influence the transmissions. The listeners continuously calculate audio quality and transmission exposure.

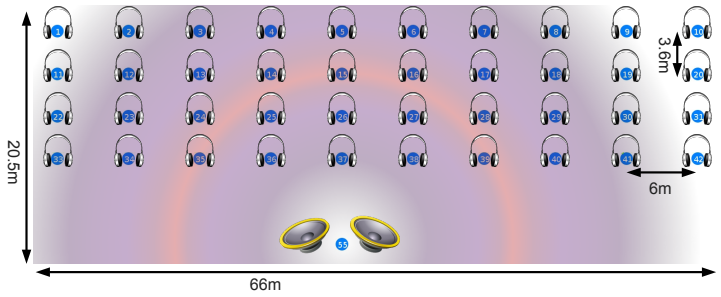


Figure 5.2: The experiment scenario as mapped to the wireless testbed. Listener nodes are in the first 4 rows and the speaker node is positioned at the bottom center.

To orchestrate the experiment, an OMF script processes the experimental design given by the locating array and iteratively executes each test. For the execution of each test, the system is first brought to a known state by resetting all wireless interfaces and caches of each node, followed by configuration of the parameters as specified by the test. After a warm-up period to avoid transient effects, measurements are collected. Table 5.4 shows the list of resources used for the Wi-Fi conferencing scenario.

Table 5.4: Experiment resource description.

Resource	Description
Wi-Fi chipset	Atheros Sparklan WPEA-110N/E/11n
Wi-Fi driver	ath9k
OS	Ubuntu 14.04 LTS
kernel	Linux 3.13.0-33-generic

5.4.3 Selected Parameters and Values

The testbed nodes we use can be configured by uploading an image containing the OS and application to run. We selected 24 parameters from the kernel's IP and UDP stacks, the Wi-Fi card driver, the audio codec used in our application, and a source of interference implemented via a dedicated SDR. Each parameter has from 2 to 5 values. Categorical parameters included all settings as levels, while numerical (*i.e.*, continuous) parameters had their levels spaced exponentially to avoid giving preference to a particular scale. For each parameter, we also ensure

that the default assigned by the Linux kernel and/or user-space tools is present. The full list of parameters and values is provided in Table 5.5.

Two parameters, robust header compression and sensing, were encoded into the locating array but were not implemented when the experiments were run. We also used `txpower` assignments 6 dBm lower for 2.4 GHz than for 5 GHz so that propagation effects would be approximately equal for the different bands. The reason for this is due to the *free-space path loss* (FSPL) difference between the 2.4 GHz and 5 GHz frequency bands [29].

5.4.4 Performance Metrics

The performance metrics we measured during experimentation were MOS and RF exposure. As we will see in Section 5.5, the locating array identifies parameters for each metric considered; its impact is also dependent on the metric (*e.g.*, transmission power improves audio quality but negatively impacts the exposure). The presence of multiple contradicting responses has been shown to be relevant in realistic scenarios, such as in [30].

The audio quality is quantified using an aggregate *mean opinion score* (MOS) [30] metric over the complete audio transmit path. The audio quality is first affected by the encoding process at the transmitter side and further reduced when transmitted over the air (see Fig. 5.3).

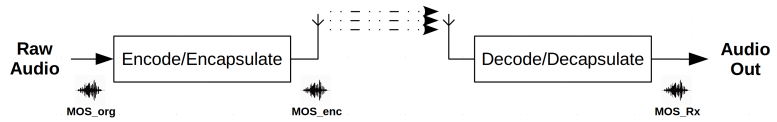


Figure 5.3: The audio quality degradation is calculated in two phases: once after the encoder unit and later after the wireless transmission.

Within the encoder unit, the first quality loss is introduced as a function of the encoder bitrate, type of encoder, and audio class used. Afterwards, the audio is transmitted over the air and a second quality loss is introduced due to packet loss, jitter, and latency impairments.

Radio frequency (RF) transmission exposure calculates the electromagnetic energy absorbed by a human body due to uplink and downlink wireless transmissions [31]. The RF *exposure index* (EI) is measured in *specific absorption ratio* (SAR) units of a given amount of power (Watts) over a given mass of human body (kg). The generalized formula of exposure covers a wide range of categories (*i.e.*, population, environment, radio access technologies, load profile, posture) but specific to our scenario, the following formula is used:

$$EI^{SAR} = \frac{1}{T}\left[\sum_{t}^{N_T}\left(d^{UL}\overline{P}_{TX}\right) + d^{DL}\overline{S}_{\text{inc}}\right]$$

Table 5.5: Parameters and values used in experimentation (default values from Ubuntu in bold).

Parameter	Identifier	Values
Band	`band`	2.4, 5 GHz
Channel	`channel`	1, 6, 11 (2.4 GHz); 36, 40, 44 (5 GHz)
Wi-Fi bitrate	`bitrate`	6, 9, 12, 24, 36 Mbps
Transmit power	`txpower`	1, 2, 4, 7, 10 dBm (2.4 GHz); 7, 8, 10, 13, **16** dBm (5 GHz)
MTU	`mtu`	256, 512, 1024, 1280, **1500** bytes
Transmit queue length	`txqueuelen`	10, 50, 100, 500, **1000** packets
Queuing discipline	`qdisc`	pfifo, bfifo, **pfifo_fast**
IP fragment low threshold	`ipfrag_low_thresh`	25%, 50%, **75%**, 100% of high threshold
IP fragment high threshold	`ipfrag_high_thresh`	16384, 65536, 262144, 1048576, **4194304** bytes
UDP receive buffer minimum	`udp_rmem_min`	**1.9231%**, 10%, 50% of maximum
UDP receive buffer default	`rmem_default`	0%, 25%, 50%, 75%, **100%** from minimum to maximum
UDP receive buffer maximum	`rmem_max`	2304, 10418, 47105, **212992** bytes
UDP transmit buffer minimum	`udp_wmem_min`	**1.9231%**, 10%, 50% of maximum
UDP transmit buffer default	`wmem_default`	0%, 25%, 50%, 75%, **100%** from minimum to maximum
UDP transmit buffer maximum	`wmem_max`	4608, 16537, 59349, **212992** bytes
UDP global buffer minimum	`udp_mem_min`	25%, **50%**, 75% of maximum
UDP global buffer pressure	`udp_mem_pressure`	0%, **33.338%**, 50%, 75%, 100% from minimum to maximum
UDP global buffer maximum	`udp_mem_max`	95, 949, 9490, **94896** pages
Robust header compression	`ROHC`	off, on (unimplemented)
Sensing	`sensing`	off, on (unimplemented)
Audio codec	`codec`	Opus, Speex
Audio codec bitrate	`codecBitrate`	7600, 16800, 24000, 34000 bit/s (or nearest allowed by codec)
Frame length aggregation	`frameLen`	20, 40, 60
Interference channel occupancy	`intCOR`	10%, 25%, 50%, 75%, 90%

During a given period of time, a transmitting antenna induces an exposure to a speaker proportional to the transmitted power $\overline{P}_{TX}$ and also induces an exposure to far away listeners that is proportional to the incident power density $\overline{S}_{\text{inc}}$. After that, the electromagnetic energy absorption per kilogram of body mass is calculated by applying the uplink and downlink absorption parameters $d^{UL} = 0.0070$ W/kg for 1W of transmitted power and $d^{DL} = 0.0028$ W/kg for $1W/m^2$ of received power density respectively [32].

5.4.5 The LA Screening Design

At present, general tools for the construction of locating arrays are not available. Therefore, we constructed a $(1, 2)$-locating array for the parameters and values in Table 5.5 as follows. First, three copies were made of a covering array of strength two on the selected parameters and values, with each copy having its columns and values within the columns permuted while respecting the number of values per column. The tests from these arrays were then combined, yielding an array that covers every two-way interaction at least 3 times. A few random tests were added until the array satisfied the $(1, 2)$-locating property.

Then we found the values that were unconstrained by the coverage and locating properties, removed tests consisting solely of unconstrained values, and replaced the remaining unconstrained values randomly. This was repeated a few times, except that on the final iteration, the values were chosen to make the value frequencies within each column as equal as possible. The final locating array screening design consists of 109 tests.

We performed a recoverability test on the compressive sensing matrix corresponding to the LA screening design (see Section 5.3.2). The ability to recover at least one term enables us to construct our model one a term at a time is exploited in the OMP and BT_OMP algorithms.

There are 4134 columns in the compressive sensing matrix arising from this locating array. This underscores the impossibility of exhaustively searching the space of all models, or even all models of a small size.

5.5 Results

We ran a total of eight screening experiments. Five replicates of the experiment using the locating array screening design were run, while three replicates of a fractional-factorial design were run for validation.

5.5.1 Results of LA Screening Experiments

Over a number of months, we ran five replicates of the experiment using the 109-test locating array as the design on the `w-iLab.t` testbed and collected measurements of downlink exposure and MOS values for each listener node, together

with the uplink exposure at the speaker node. We thus had exposure values for the speaker node and MOS values for each listener node.

In these replicates, the number of listener nodes available ranged from 33-35. Of these available listeners, only a few of them (0-5) collected responses for all tests. Most listeners failed to collect responses from 1-42 of the 109 tests. We attempted to determine the cause of these failures but were unable to do so. These may have been due to the true cause involving more than one interaction and/or being intermittent and thus violating the assumptions needed by the $(1, 2)$-locating property.

We present the results of aggregating experiments 2 through 5, as these are similar to the individual experiments' results. The first data set was incomplete, which prevents analyzing it correctly due to violating the locating property. For MOS, the distribution after aggregating across nodes and experiments is shown in Fig. 5.4 (left). The aggregation used involves first normalizing each node and then averaging across listener nodes. We excluded missing values and values of 1.0 (the minimum possible) from aggregation, except when every node returned 1.0 for a particular test. This was done because the piecewise linear shape of the distribution suggests that the outcomes being reported as 1.0 actually differ in audio quality, but are all so bad that MOS does not measure the difference. The distributions of each listener and of the aggregate all appear to be piecewise linear. Because there is a region of constant value, no transformation can bring it into a form optimized for linear modelling. We choose to analyze the data as-is, but it would be possible in principle to replace the distance function used in BT_OMP with a different function that treated values below 1.0 as being a match for 1.0 so as to fit a piecewise linear response directly. For each listener node, the MOS values used in subsequent analysis were normalized. For exposure, the distribution after taking logarithmic transform of the raw values, normalizing, and aggregating across experiments is shown in Fig. 5.4 (right). It appears that the logarithmic transform makes the data closer to normally-distributed, and the residuals are again close to normal (see Fig. 5.5).

Tables 5.6 and 5.7 show the results of the backtracking search algorithm BT_-OMP for the response of MOS and exposure, respectively. The model given is the top model by increasing R^2 out of the first 1024 returned, with an R^2 goal of 0.9 and a limit of 10 terms set. For exposure, the log-transformed data was analyzed, while for MOS, the different listener nodes were averaged after being individually normalized. For each term added to the model through the cutoff, we show its coefficient in the final model, the assignment of value(s) to the parameter(s) represented by that term, and the R^2 value obtained by the model including it and all previous terms. The worse R^2 for MOS is likely due to the piecewise-linear nature of its response distribution.

5.5.2 LA Screening Results

Table 5.8 gives the factors consistently identified as significant by the experiments using the LA as the screening design. For MOS, the interaction `band`=2.4 $\wedge$

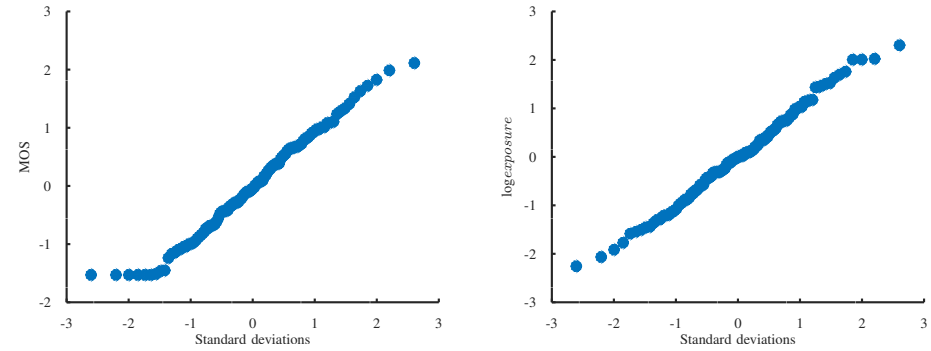


Figure 5.4: Normal probability plot for MOS (left), and log-normal plot for exposure (right).

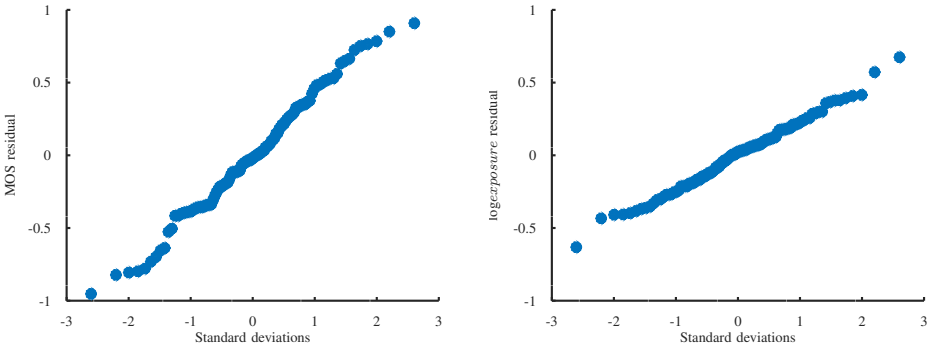


Figure 5.5: Normal probability plots for residuals for MOS (left) and log*(exposure) (right).*

Table 5.6: Top terms for MOS.

Index	Coefficient	Parameter(s) and Value(s)	R^2	ΔR^2
0	$+0.653$	(intercept)	0.000	0.000
1	-0.161	`band`=2.4	0.114	0.114
2	$+0.460$	`intCOR`=0.1	0.263	0.150
3	$+0.650$	`txpower`=10/16	0.392	0.128
4	$+0.412$	`txpower`=7/13	0.489	0.097
5	$+0.293$	`intCOR`=0.25	0.587	0.098
6	-0.201	`band`=2.4 $\wedge$ `sensing`=0	0.627	0.041
7	$+0.426$	`txpower`=4/10 $\wedge$ `mtu`=512	0.673	0.046
8	-0.195	`codecBitrate`=7600/7750	0.696	0.024
9	-0.481	`band`=2.4 $\wedge$ `intCOR`=0.9	0.760	0.064
10	-0.248	`ipfrag_low_thresh`=0.75 $\wedge$ `ROHC`=1	0.790	0.029

Table 5.7: Top terms in log*(exposure).*

Index	Coefficient	Parameter(s) and Value(s)	R^2	ΔR^2
0	$+0.147$	(intercept)	0.000	0.000
1	$+0.245$	txpower=10/16 $\wedge$ ROHC=0	0.119	0.119
2	-0.597	band=2.4	0.430	0.311
3	$+0.583$	bitrate=6	0.553	0.123
4	-0.627	txpower=1/7	0.655	0.103
5	$+0.551$	txpower=2/8	0.751	0.095
6	$+0.442$	bitrate=9	0.816	0.065
7	$+0.219$	frameLen=20	0.848	0.032
8	-0.332	txpower=4/10	0.897	0.050
9	$+0.302$	bitrate=12	0.943	0.046

Table 5.8: Significant parameters screened by the LA.

log(exposure)	MOS
txpower	band
band	intCOR
bitrate	txpower
frameLen	codecBitrate

intCOR=0.9 appears significant, while no interactions were consistently identified as significant for log(exposure). The first term for log(exposure) that involves an interaction with ROHC did not appear in any of the individual experiments' models, so we suspect it to be an artifact of imperfect agreement between the experiments. If our models explain the data well, we expect the residuals to be due to noise, resulting in them being normally distributed with zero mean. The normal probability plot of the residuals for log(exposure) (see Fig. 5.5) appears sufficiently close to a straight line that there is no reason to suspect non-normality. It appears our models for exposure are good explanations for the variation in the data and thus that the parameters we identified as significant should be reliable. For MOS, on the other hand, we see a deviation from normality at the low end, which is again suspected to be due to the piecewise linear nature of the MOS data.

5.5.3 Validation using Fractional-Factorial Experiments

We use a fractional-factorial experiment on the parameters screened by the locating array to validate the screening results. The analysis of the results of the fractional-factorial should confirm the identification of significant parameters if the results of the LA are correct.

For the fractional-factorial experiments, only the parameters band, channel, bitrate, txpower, codec, codecBitrate, and frameLen were varied. For bitrate, codecBitrate, and frameLen, only the lowest and highest values were included, while all values of the other parameters were included.

This yields a full-factorial with a total of 480 tests, but we chose to run a one-half fraction, which resulted in a fractional-factorial with 240 tests. Three replicates of the fractional-factorial experiment were run. We collected measurements

for only 199 tests in the first replicate (*i.e.*, the experiment was incomplete), but for all 240 tests in subsequent replicates.

As in the LA experiments, a logarithmic transform was applied to exposure. The standardized MOS values were averaged across all listener nodes. For the complete experiments, the parameters identified as significant for exposure were `txpower`, `bitrate`, `band`, and `frameLen`, which are the same as those identified by the locating array. The R^2 value was 0.98, indicating a very good fit. This is better than the LA experiments in part because they used a cutoff of 0.9 and in part because JMP can include 3-way and higher interactions when analyzing factorial designs. For MOS, `txpower`, `codecBitrate`, and `band` were identified as significant, with R^2 value of 0.50-0.72. These lower values of R^2 suggest that the 10 terms shown in Table 5.6 are not all significant.

The analysis of the experiments using the fractional-factorial designs support the screening results of the locating array. It agrees that the terms identified by the locating array are significant and that terms it did not select are not; this is apart from `intCOR`, which was not varied in the fractional-factorial designs and thus could not be selected by them.

The R^2 values for MOS and $\log(\text{exposure})$ are comparable for the two designs. Additionally, for both locating arrays and fractional factorial designs, the R^2 values were lower for MOS than for exposure. This is not surprising because exposure is calculated from values measured at the transmitter, while MOS is calculated from values measured at the listeners. Thus MOS incorporates more sources of noise.

The parameters we have identified as significant are plausible. Some would have been expected by an expert. For example, one would expect that `txpower` should have an effect on both exposure and MOS, and that interference should have an effect on MOS but not exposure.

5.6 Conclusions and Future Work

In this paper, a locating array is used for the first time to screen parameters and two-way interactions affecting MOS and RF exposure in `w-iLab.t`, a complex engineered network. It, together with our backtracking search analysis method, is able to screen out most of the parameters and two-way interactions as insignificant. The parameters identified as significant are similar between the different replicates despite the high level of noise in the data; this suggests some degree of robustness of the analysis.

It would be interesting to use locating arrays to screen a scenario in which cross-layer interactions are known to exist, and to conduct an experiment at an even larger scale.

Locating arrays with varying numbers of values per parameter can be generated by an ad hoc procedure, such as the one we used (described in Section 5.4.5). General algorithms for the construction of locating arrays are of interest. Because our analysis can fail to yield a good model if the residuals end up farther from

the term that should be selected than from some other term, it may be possible to produce a locating array that has better analysis properties by guiding the choices so as to maximize the minimum distance between any pair of terms' associated vectors. An investigation of alternate stopping criteria and definitions of safety are also of interest.

As a next step, we plan to conduct follow-on experimentation, using an optimization procedure (*e.g.*, SUMO [30, 31]) to determine which settings of the parameters we have identified as significant jointly minimize exposure and maximize MOS in our audio conferencing scenario.

An additional direction for future extension is to handle parameters that can be measured but not controlled, such as temperature or background interference, and potentially even to detect indirectly the presence of parameters that have a significant effect but cannot be measured.

Acknowledgment

This work is supported in part by the U.S. National Science Foundation under Grant No. 1421058, by the IWT project "SAMURAI: Software Architecture and Modules for Unified RAdIo control," and by the European Commission Horizon 2020 Programme under grant agreement n 688116 (eWINE).

References

[1] J. S. Panchal, R. D. Yates, and M. M. Buddhikot. *Mobile Network Resource Sharing Options: Performance Comparisons*. IEEE Transactions on Wireless Communications, 12(9):4470–4482, September 2013.

[2] V. Sevani, B. Raman, and P. Joshi. *Implementation-Based Evaluation of a Full-Fledged Multihop TDMA-MAC for WiFi Mesh Networks*. IEEE Transactions on Mobile Computing, 13(2):392–406, February 2014.

[3] R. Cáceres and L. Iftode. *Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments*. IEEE Journal on Selected Areas in Communications, 13(5):850–857, June 1995.

[4] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson. *Cross-Layer Design for Wireless Networks*. IEEE Communications Magazine, 41(10):74–80, 2003.

[5] G. Athanasiou, T. Korakis, O. Ercetin, and L. Tassiulas. *A Cross-Layer Framework for Association Control in Wireless Mesh Networks*. IEEE Transactions on Mobile Computing, 8(1):65–80, January 2009.

[6] W. Hu, H. Yousefi'zadeh, and X. Li. *Load Adaptive MAC: A Hybrid MAC Protocol for MIMO SDR MANETs*. IEEE Transactions on Wireless Communications, 10(11):3924–3933, November 2011.

[7] S. Efazati and P. Azmi. *Cross Layer Power Allocation For Selection Relaying and Incremental Relaying Protocols Over Single Relay Networks*. IEEE Transactions on Wireless Communications, 15(7):4598–4610, July 2016.

[8] A. Cammarano, F. L. Presti, G. Maselli, L. Pescosolido, and C. Petrioli. *Throughput-Optimal Cross-Layer Design for Cognitive Radio Ad Hoc Networks*. IEEE Transactions on Parallel and Distributed Systems, 26(9):2599–2609, September 2015.

[9] Y. Wang, M. C. Vuran, and S. Goddard. *Cross-Layer Analysis of the End-to-End Delay Distribution in Wireless Sensor Networks*. IEEE/ACM Transactions on Networking, 20(1):305–318, January 2012.

[10] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley and Sons, Inc., 9th edition, 2017.

[11] S. Bouckaert, W. Vandenberghe, B. Jooris, I. Moerman, and P. Demeester. *The w-iLab.t Testbed*. In T. Magedanz, A. Gavras, N. H. Thanh, and J. S. Chase, editors, Testbeds and Research Infrastructures. Development of Networks and Communities, pages 145–154. Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-17851-1_11.

[12] J. P. C. Kleijnen. *An Overview of the Design and Analysis of Simulation Experiments for Sensitivity Analysis*. European Journal of Operational Research, 164:287–300, 2005.

[13] S. G. Gilmour. *Factor Screening via Supersaturated Designs*. In A. M. Dean and S. M. Lewis, editors, Screening: Methods for Experimentation in Industry, Drug Discovery and Genetics, chapter 8, pages 169–190. Springer-Verlag, 2006.

[14] R. Li and D. K. J. Lin. *Analysis Methods for Supersaturated Designs: Some Comparisons*. Journal of Data Science, pages 249–260, 2003.

[15] C. J. Colbourn and D. W. McClary. *Locating and detecting arrays for interaction faults*. Journal of Combinatorial Optimization, 15(1):17–48, 2008.

[16] SAS Institute Inc. *JMP®, Version 13*. https://www.jmp.com/, 1989–2017.

[17] J. A. Tropp and A. C. Gilbert. *Signal recovery from random measurements via orthogonal matching pursuit*. IEEE Transactions on Information Theory, 53(12):4655–4666, December 2007.

[18] *Meta-Modelling for Complex Engineered Networks*. http://www.public.asu.edu/∼syrotiuk/meta-modelling.html.

[19] C. Croarkin, P. Tobias, J. J. Filliben, B. Hembree, W. Guthrie, L. Trutna, and J. Prins, editors. *NIST/SEMATECH e-Handbook of Statistical Methods*. NIST, April 2012.

[20] X. Li, N. Sudarsanam, and D. D. Frey. *Regularities in data from factorial experiments*. Complexity, 11(5):32–45, 2006. doi:10.1002/cplx.20123.

[21] B. Jones and D. Majumdar. *Optimal Supersaturated Designs*. Journal of the American Statistical Association, 109(508):1592–1600, 2014.

[22] A. Hartman. *Software and Hardware Testing Using Combinatorial Covering Suites*. In M. C. Golumbic and I. B.-A. Hartman, editors, Graph Theory, Combinatorics and Algorithms: Interdisciplinary Applications, chapter 10, pages 237–266. Springer US, Boston, MA, 2005. doi:10.1007/0-387-25036-0_10.

[23] S. R. Searle. *Linear models for unbalanced data*. John Wiley & Sons, 1987.

[24] R. Li and D. K. J. Lin. *Analysis methods for supersaturated design: Some comparisons*. Journal of Data Science, 1(3):249–260, 2003.

[25] C. J. Colbourn, D. Horsley, and V. R. Syrotiuk. *Frameproof Codes and Compressive Sensing*. In Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing, 2010.

[26] E. Candès and M. Wakin. *An introduction to compressive sampling*. IEEE Signal Processing Magazine, 25(2):21–30, 2008.

[27] G. Tang and A. Nehorai. *Computable Performance Bounds on Sparse Recovery*. IEEE Transactions on Signal Processing, 63(1):132–141, Jan 2015. doi:10.1109/TSP.2014.2365766.

[28] R. J. Bayardo Jr. and R. Schrag. *Using CSP Look-Back Techniques to Solve Real-World SAT Instances*. In Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI'97, IAAI'97), pages 203–208, 1997.

[29] *Free-space path loss*. https://en.wikipedia.org/wiki/Free-space_path_loss.

[30] M. T. Mehari, E. D. Poorter, I. Couckuyt, D. Deschrijver, G. Vermeeren, D. Plets, W. Joseph, L. Martens, T. Dhaene, and I. Moerman. *Efficient Identification of a Multi-Objective Pareto Front on a Wireless Experimentation Facility*. IEEE Transactions on Wireless Communications, 15(10):6662–6675, Oct 2016. doi:10.1109/TWC.2016.2587261.

[31] M. T. Mehari, E. De Poorter, I. Couckuyt, D. Deschrijver, J. V.-V. Gerwen, D. Pareit, T. Dhaene, and I. Moerman. *Efficient global optimization of multi-parameter network problems on wireless testbeds*. Ad Hoc Networks, 29:15–31, 2015. doi:http://dx.doi.org/10.1016/j.adhoc.2015.01.014.

[32] D. Plets, W. Joseph, K. Vanhecke, G. Vermeeren, J. Wiart, S. Aerts, N. Varsier, and L. Martens. *Joint minimization of uplink and downlink whole-body exposure dose in indoor wireless networks*. Biomed Research International, page 9, 2015.

6
Conclusion

"Everything I ever wanted to know I just ask a search engine and there's the answer. So the least I can do for my clients is share what I've learned."

– Derek Sivers (1969 -)

This famous quote from Derek Sivers is an indication of the technological advancement of search engines. As difficult as it used be when finding a piece of information, search engines have revolutionized this process and information is retrieved in a matter of milliseconds. Google search engine is a typical example but we researchers of this age, are on the list too. When we were kids, we used to remember everything that comes to our mind. But as we grew older, an excess of information overwhelms our capacity and we start to act like small search engines (i.e. indexing the information we learn). This was a huge part of my Ph.D. success.

Next-generation wireless networks have always posed challenges to the research community. When wireless networks are first standardized using new technologies and methodologies, there is a challenge of lacking prototyping software and hardware. In most of the cases, they end up working with simulation tools specifically designed for these wireless networks. Simulation tools also have their own limitations, because the physical layer models and traffic patterns used might not fully represent the underlying wireless system. On the contrary, prototyping software and hardware can become available, but due to the high cost of large-scale experimentation facilities, the design process becomes expensive. Large-scale experimentation is also challenging, due to orchestration overheads and domain knowledge during earlier periods of standard rectification.

The research work in this dissertation has focused on reducing the time complexity of optimization and characterization problems of complex wireless net-

works. The design methodologies considered are *performance optimization*, *system modeling* and *parameter screening*. Thus, we will first summarize the design methodologies followed by future directions in order i) to improve the methods and approaches used and ii) to show new directions in the research topic.

6.1 Summary

Complex wireless networks are often characterized by a large number of configuration parameters and a very large configuration space. A term to represent this phenomenon is coined as *curse of dimensionality*, where it becomes exponentially difficult to optimize, model and screen complex wireless networks for every addition of configuration parameters.

In the first two Chapters (2 and 3) of this dissertation, the experimentation effort of a *performance optimization* problem is tackled in complex wireless networks. While performance optimization covers the majority of wireless network problems, the goal is to search for optimum parameter settings that satisfy the performance objectives in a short period of time. A typical approach that is commonly applied to solve optimization problems is by using *heuristic algorithms*. Heuristic algorithms find approximate solutions of a problem through learning and discovering the environment. They have fast convergence speeds and their proposed solutions are acceptable in most of the cases. However, when the design process demands computationally intensive and complex operations, such as in complex wireless networks, heuristic algorithms fail to meet the requirements. The complexity can be in a number of ways. It can be due to i) a large configuration space, ii) the use of experimentation facilities instead of simulation tools, iii) the use of multiple objectives in the design process, and iv) overheads during experiment orchestration. Instead, *surrogate modeling* tools are applied to provide fast solutions at a reasonable performance accuracy. A surrogate model is an efficient mathematical representation of a *black-box* system, which in this case is the complex wireless network. From measured data points, a surrogate model predicts the performance of unknown data points in the form of Gaussian random variables, thus applicable in statistical analysis. Surrogate models have different variants (i.e. Kriging, Gaussian Process, support vector regression, neural networks, ...) that are used in different applications. In this research work, *Kriging surrogate models* are applied because they are found efficient in a number of engineering problems. Tightly coupled with the modeling process, sample point selection also plays an important role in the efficiency of performance optimization. To this end, the *Expected improvement* (EI) of every untested point, from the Kriging model, is compared and the one having the highest EI is selected for next round test. Finally, we have validated the approach by using a wireless audio conferencing system and optimized the configuration settings, in search of improved audio quality and reduced electromagnetic exposure. An exhaustive search approach would have required testing all configuration settings (6528 elements), but SUMO evaluated a near optimum solution by using only 94 experiments (Speed up Factor = 6528/94

= 69.45) and a solution accuracy of 96.58%. Furthermore, we have conducted the computational complexity of the optimization process and it is found negligible compared to the complexity of conducting the wireless experiment.

In Chapter 4, the problem of *system characterization* and *performance optimization* is jointly tackled by using surrogate models. The scenario considered in this Chapter is a single-hop wireless sensor network (WSN), under the influence of a dynamic environment. Because of the environment dynamics, performance becomes sub-optimal and configuration parameters need to be re-calibrated to bring the system back to an optimum operating state. Furthermore, a *cloud repository* is used to store multiple performance models of the WSN, each representing a static instance of the dynamic environment. This way, the WSN is characterized offline and its performance is adjusted online whenever the environment changes. Since system characterization is the underlying goal, a different sampling strategy is applied unlike the EI used in performance optimization. The goal of EI is to direct the optimization process to an optimum solution thus intermediate models are locally accurate around the optimum region. However, in system characterization, global accuracy is the primary goal and the *FLOLA-Voronoi* sampling strategy is used instead. The *FLOLA* part, a Fuzzy implementation of the LOLA sampling strategy, is responsible for exploiting non-linear regions of the system and it uses local linear approximations to predict measurement values. On the other hand, the *Voronoi* part explores the parameter space by searching sparsely sampled regions using a tessellation diagram. Finally, the scores from the FLOLA and Voronoi calculations are combined together to decide the next sample point for improving the accuracy of the model. The innovation of this Chapter goes beyond currently existing solutions by using a cloud repository approach. Specific to the single-hop WSN scenario, the proposed approach only requires to conduct 10 experiments to know the current environmental conditions and optimize its performance, as opposed to 125 experiments for state-of-the-art SUMO optimization approach or 4800 experiments for an exhaustive search based approach.

The majority of wireless networks focus on performance optimization and by adding system characterization, an almost complete solution was created (Chapters 2, 3 and 4). What is remaining is to handle the design complexity of wireless networks due to a large number of configuration parameters, and Chapter 5 is targeted to solve this specific problem. In order to reduce the design complexity, the most sensitive parameters are selected and passed on to the next steps for further design operations (modeling and optimization). In literature, *supersaturated designs* (SSDs) are used along with balanced analysis methods, such as *Orthogonal Matching Pursuit* (OMP). This approach, however, does not consider *parameter interactions*, which are common in wireless networks between different layers of the protocol stack. Therefore, the Chapter applies a new combinatorial design method called *locating array* (LA) to design a screening experiment for locating the most sensitive parameters and parameter interactions. Because of interaction designs, LA becomes unbalanced and traditional analysis methods cannot be applied. To this end, a *backtracking orthogonal matching pursuit* (BT-OMP) analysis method is used. Afterward, candidate solutions are evaluated with a given

coefficient of determination score (R^2), and they are combined to screen sensitive parameters and parameter interactions. The innovation of LA is not bound only to screening interactions but also scales logarithmically in size to the number of configuration parameters. This makes it very attractive to screen complex wireless networks that have a large number of configuration parameters. In this research work, a wireless audio conferencing system is set up and configured using 24 parameters, which are collected from different layers of the protocol stack. In total, the wireless system has $\approx 10^{13}$ configuration parameters but LA only applied 109 tests to screen the most sensitive parameters and parameter interactions.

6.2 Future Directions

The future of wireless networking is promising and a number of solutions are designed to service our demands. Similar to most other design works, this dissertation has tackled different components of the big picture, with the ultimate goal of having a unified solution. To this end, different methodologies (i.e. performance optimization, system characterization and parameter screening) are tackled and improved in the scope of solving wireless network problems. *The next step is to unify the different pieces together and aid in the development of next-generation wireless networks (i.e. IoT, 5G, industry 4.0)*. On the other hand, this dissertation has investigated surrogate model-based optimization and system characterization solutions, alternative to nature-inspired algorithms and machine learning tools. Thus, *another future direction is to compare the new solution with the alternatives*. Finally, there are a number of ideas that have not been tackled in this dissertation and are left as a future work. We believe they are important and will make the contribution much stronger. Here is a list of future directions that need further investigations.

– Three of the Chapters (2,3,5) in this dissertation have applied experimental validation, and outliers were detected using *PRE/POST experiment monitoring* solution (see Section 2.5.1). It works by monitoring the wireless channel during PRE and POST experiment execution and if the channel remains clean during these periods, then there is a high chance that the experiment itself is not interfered. Even though the solution is innovative, it is not full-proof because there is a small chance for an experiment to be interfered even when PRE and POST experiment monitorings are clean. A definitive answer is to use monitoring solution during experiment execution. One example could be multi-layer (i.e. feature and energy detection) and multi-technology (i.e. Wi-Fi, Zigbee, Bluetooth, ...) interference estimators. This way, the interference estimator distinguishes the external interference from the *system under test* (SUT), thereby quantifying the amount of influence it has on the experiment.

– Different from all Chapters, Chapter 4 is using simulation tools to obtain measurement data. One of the reasons behind this decision is the complexity of the topic (WSN optimization in dynamic environments). The future goal of this

topic is to switch to an *experiment-driven research* such that a WSN in a dynamic environment can be optimized real-time using cloud repositories. However, there are a couple of things that need to be done to achieve this goal. First, the Chapter used a very simple radio abstraction model (i.e. UDGM). This, however, is far from reality and more advanced radio models must be used such as *3D ray tracing* or advanced *heuristic tracing*. Thus as an intermediate solution, the same simulation environment should be re-evaluated using advanced radio abstraction models. Afterward, the experiment-driven research can be investigated and the solution will be compared to the simulation environment executed using the advanced radio models. As promising as it looks, the advanced radio models will not entirely represent the physical world and thus we can apply a feedback mechanism to fine tune the behavior using outputs from a real-world experiment. This is referred as *transfer learning*, where the behavior of a known system is transferred to another similar system for improving the performance. As a result, a real wireless system can be realized by augmenting information obtained from a simulation experiment.

– In the first three Chapters (2,3,4), *Kriging models* were used to optimize and model the performance of complex wireless networks. Kriging models have proven themselves in many engineering problems but they are limited to smooth response surfaces (i.e. transmission power on network lifetime, codec bitrate on audio quality, and MANET hop count on transmission latency). As a follow-up research work, we would like to investigate other types of surrogate models that will have a good response to categorical parameters (i.e. MAC protocols on throughput performance and codec types on audio quality). One potential candidate to be used in place of a Kriging model is *Neural Network*.

– The problem of *detecting environment change* in dynamic environments is another research work planned in the future. In Chapter 4, a very simple approach is used by continuously monitoring performance metrics, and an environmental change is detected when the metrics fall out of range (maximum value for latency, packet error rate and energy consumption). However, this approach is susceptible to noise and a more robust solution is a required. One approach is to remove outlier experiments before making the comparison but at the expense of increased experimentation time. Another approach is to use confidence intervals associated with Kriging models. From a given set of measurement points, SUMO builds a confidence interval over untested points that it expects the real measurement to be on and if performance metrics are outside a given confidence interval (i.e. 95%), an environmental change is detected.

– As a continuation of the previous point, a change in the environment triggers environment characterization and model selection or merging operation builds a representative model of the unknown environment. However, it is possible that selected or merged model is not representative enough and we will be left with building a new model all from scratch. This is one of the situations we want to avoid being caught up because the wireless system is operating online

and there is no clue on how to optimize the performance of the unknown environment. The only way to avoid such a problem is by carefully designing the offline phase, such that the cloud repository stores enough reference models that represent the dynamics of the environment. In Chapter 4, uniformly spaced dynamic environments (45 in total) are selected which by no means is an optimum selection. As a future work, an efficient selection method needs to be devised to pick as few dynamic environments as possible and this resembles the exploration phase of a general optimization process.

– The novelty of Chapter 5 goes beyond parameter screening by considering *interactions*. Parameter interactions are common in wireless networks because they are designed based on the principle of protocol layering. In literature, parameter screening usually does not consider interactions and screened parameters (main effects) are used to optimize the performance or model the system response. As a continuation of Chapter 5, we would like to screen main effects and parameter interactions of complex wireless networks followed by a performance optimization operation. Moreover, we also like to investigate *weak heredity* in complex wireless networks, a condition where significant interactions have at least one parameter significant. If this can be proved, then the solution of using *locating arrays* and *surrogate models* for screening and performance optimization respectively will become innovative.

– Recall in LA screening analysis, a number of model estimates have been created to combat measurement noise. At this point, however, there is no real measure to detect whether a model fits, under-fits, or over-fits the LA response. Having a higher R^2 value is not an indication of a good fit, and some statistical methods even put a range to this value and everything outside this range is treated either as an under-fit or an over-fit. The problem of under-fitting and over-fitting lies to the fact that a model will be a poor estimate for statistical inferences, such as screening analysis. Therefore, one must make sure a model estimate is a good fit. To this end, fitness can be detected by measuring the progress of R^2 values as a function of model terms.