ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre 3190

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche : Intelligent Processing Systems & Security (IPSS)
Discipline : Informatique
Spécialité : Intelligence Artificielle

Présentée et soutenue le : *22/04/2019* ***par :***

Sara RETAL

Artificial Intelligence Techniques in Cloud-Based Systems: 5G Network Architecture and Vehicular Ad-Hoc Network

JURY

Abderrafiaa KOUKAM	*PES,*	*Université de Technologie de Belfort Montbéliard, France*	*Président*
Abdellah IDRISSI	*PH,*	*Faculté des sciences, Université Mohammed V, Rabat*	*Directeur de Thèse*
Fatima Zahra BELOUADHA	*PES,*	*Ecole Mohammedia d'Ingénieurs, Université Mohammed V, Rabat*	*Rapporteur/ Examinateur*
Faouzia BENABBOU	*PES,*	*Faculté des sciences de Ben M'Sik, Université Hassan II, Casablanca*	*Rapporteur/ Examinateur*
Mohamed El Youssfi EL KETTANI	*PES,*	*Faculté des sciences, Université Ibn Tofail, Kénitra*	*Examinateur*

Année Universitaire : 2018/2019

"I would like to express my profound gratitude to my dear parents for their immense moral and material support, their presence and their encouragement throughout my studies. A lot of credit goes to them, and this achievement would not have been feasible without them. I also would thank Hamza, Ali, Iliasse, Rima, Yassamine and all my family members for their continuous encouragement. "

Sara Retal

Acknowledgements

This thesis was carried out in the department of computer science of the faculty of sciences in Rabat within Intelligent Processing Systems & Security (IPSS) team. I would like to thank Professor **Fouzia OMARY**, the director of IPSS for welcoming me to her laboratory and for her continuous support of my doctoral research and studies.

First, I would like to thank my thesis advisor Professor **Abdellah IDRISSI**, for having directed and supervised this work with great scientific rigor. The quality of his advice, the support and the trust he gave me, allowed me to evolve my ideas and give me another vision of research. These years spent at this team with him brought me an incredibly enriching experience, both in research and teaching.

I would like to thank Professeur **Abderrafiaa KOUKAM**, for agreeing to take part in my thesis committee as president of the jury and for his devoted precious time.

Furthermore, I would like to thank Professor **Fatima Zahra BELOUADHA**, to have accepted the task of the rapporteur and to have devoted precious time to the examination of this manuscript.

Besides, I would also like to thank Professor **Faouzia BENABBOU** for having accepted to evaluate this work and having consecrated precious time to the rapporteur task.

I would also thank Professor **Mohamed El Youssfi EL KETTANI**, for agreeing to examine my thesis and to have dedicated valuable time to the examination of this manuscript.

Abstract

The 5G architecture will be designed for a hyper-connected society that combines growing services. 5G architecture, Cloud Computing, and Vehicular Ad-Hoc Network are different emerging and intertwining domains. Therefore, mobile networks and vehicular networks markets take advantages of virtualization techniques to enhance services. Among the main challenges facing these new paradigms is the need for intelligent control of resources allocation. In this thesis, we propose different solutions that play the role of intelligent resource controllers in the cloud-based mobile network and the cloud-based vehicular ad-hoc network. The proposed approaches are implemented to improve the related works and give a trade-off solution between the conflicting objectives of the problem, an autonomous solution and an adaptable solution to external factors. Various Artificial Intelligence methods are used namely constraint satisfaction problem, multi-objective optimization, and fuzzy control system. The results revealed the efficiency of the proposed schemes as per the strategy of each solution.

Keywords: Artificial Intelligence, Constraint Satisfaction Problem, Resource Controller, 5G Architecture, Vehicular Ad-hoc Network.

Résumé

L'architecture 5G sera conçue pour une société fortement connectée offrant des services en croissance. L'architecture 5G, le Cloud Computing et le réseau ad hoc de véhicules sont de différents domaines émergents et imbriqués. Par conséquent, les réseaux mobiles et réseaux de véhicules exploitent les techniques de virtualisation pour améliorer les services. Parmi les principaux défis de ces nouveaux paradigmes, nous citons la nécessité d'une gestion intelligente des ressources. Dans cette thèse, nous proposons différentes solutions jouant le rôle de contrôleurs de ressources intelligents dans le réseau mobile et le réseau de véhicules basés sur le Cloud. Les systèmes proposés sont mis en œuvre pour améliorer les travaux de la littérature et offrent à la fois une solution autonome et adaptable aux facteurs externes, ainsi qu'une solution de compromis entre les objectifs contradictoires. Diverses méthodes de l'Intelligence Artificielle sont utilisées, à savoir la programmation par contraintes, l'optimisation multi objective et le système de contrôle flou. Les résultats ont révélé l'efficacité des schémas proposés conformément à la stratégie de chaque solution.

Mots-clés : Intelligence Artificielle, Problème de Satisfaction de Contraintes, Gestion de Ressources, Architecture 5G, Réseau Ad Hoc de Véhicules.

Résumé détaillé

L'union du Cloud Computing et de l'architecture 5G, comportant les réseaux mobiles et les réseaux ad-hoc vehiculaires, apportera une richesse précieuse en capacité, élasticité et fonctionnalité aux services de cette architecture. D'autre part, l'intelligence artificielle est une solution appropriée aux problèmes de gestion des ressources, car elle contient plusieurs techniques et leurs applications. Par conséquent, l'intelligence artificielle doit être utilisée pour une gestion de ressources meilleure et intelligente. L'union du Cloud Computing et de l'intelligence artificielle a acquis un développement significatif dans le monde des technologies de l'information et de la communication. Cette union a le potentiel d'améliorer la gestion des données, leur stockage et leur traitement dans divers centres de données répartis géographiquement. Cette combinaison offre également aux chercheurs une nouvelle possibilité d'étudier des probabilités illimitées pour l'avenir. À l'origine, l'objectif de cette thèse est de contribuer aux solutions de l'architecture 5G du futur, dans les deux secteurs suivants : les réseaux de communication mobile et les réseaux véhiculaires. À notre connaissance, les travaux cités dans l'état de l'art ne présentent pas des solutions en temps réel, autonomes et adaptables aux facteurs externes. Les solutions proposées dans cette thèse apporteront un plus dans la mise en œuvre de l'architecture 5G qui nécessite une gestion intelligente des ressources. Les deux problèmes principaux étudiés dans cette thèse sont le placement des fonctions réseau virtuelles dans les réseaux mobiles et la sélection des passerelles mobiles dans les réseaux véhiculaires. Le travail sur ces deux problèmes est essentiel pour une gestion adéquate des ressources. Cela nous conduit à utiliser des méthodes et des techniques d'intelligence artificielle pour proposer des contrôleurs de ressources respectant les objectifs et les contraintes de chaque problème.

À notre connaissance, les travaux cités dans l'état de l'art sur le placement de fonctions réseau virtuelles au regard des normes 3GPP ne présentent pas de solutions en temps réel, autonomes et adaptables aux facteurs externes. Par conséquent, nous proposons premièrement une solution pour définir les positions des passerelles S-GWs et P-GWs de manière efficace dans des centres de données, en utilisant la modélisation par satisfaction de contraintes (CSP) et une version améliorée de l'algorithme 'Forward-Checking', afin de garantir les besoins

des utilisateurs et des services. Cette solution apporte un placement en temps réel, contrairement à l'approche de la littérature basée sur la théorie des jeux, qui requiert les pires valeurs des objectifs avant de trouver un compromis. Nous avons appliqué un CSP avec différents objectifs pour le problème de placement de machines virtuelles relatives aux S-GWs et P-GWs, et cela en respectant les normes du 3GPP. Les méthodes ont donné des résultats satisfaisants au regard de leurs objectifs, en réduisant le nombre de relocalisations des S-GWs, le nombre de machines virtuelles correspondant au P-GWs, ainsi qu'en optimisant le coût du chemin afin d'améliorer la qualité de l'expérience. La contribution suivante est un système de placement de fonctions réseau virtuelles conçu pour offrir un maximum de flexibilité pour répondre aux préférences de l'opérateur et s'ajuster au comportement de l'utilisateur. Le système constitue une solution équitable compte tenu des contraintes conformes aux normes 3GPP, qui permettent de réduire les coûts de relocalisation des passerelles S-GWs et le coût du chemin reliant les passerelles P-GWs et les stations de base eNodeB. En outre, le système vise à réduire le coût des machines virtuelles allouées. L'approche proposée pour réaliser le solveur du contrôleur est la programmation par contraintes. Cette solution apporte un système de placement de fonctions réseau virtuelles en temps réel, comme dans la première contribution, pouvant jouer le rôle de contrôleur de ressources. Le composant principal du système fonctionne selon certaines règles pour obtenir une solution adéquate de façon à ne pas utiliser les poids comme dans la première contribution. Enfin, dans la troisième contribution, nous avons proposé un contrôleur de logique floue pour prendre en charge le placement de fonctions de réseaux virtuelles et fournir une solution adaptative pour gérer et organiser le réseau. Notre approche permet à la solution de s'adapter à la mobilité des équipements des utilisateurs et à leurs besoins en matière de qualité d'expérience. De plus, le système minimise les coûts de relocalisation des passerelles S-GWs et le chemin entre l'utilisateur et les passerelles P-GWs en tenant compte des capacités des ressources. Le problème est basé sur des objectifs contradictoires pour lesquels un compromis doit être trouvé, et dépend également de facteurs externes fondés sur le comportement et les besoins des utilisateurs. Dans l'approche proposée, nous remédions au problème de l'intervention du décideur. Le problème est formulé comme un problème d'optimisation multiobjectif et résolu en utilisant la méthode de la somme pondérée. Nous introduisons également en tant que première phase un contrôleur de logique floue qui fournit des poids flous à la méthode de la somme pondérée. De cette manière, la solution est adaptée au comportement des utilisateurs (à savoir, la mobilité des utilisateurs) et à leurs besoins (à savoir, la qualité de l'expérience en fonction des

applications utilisées). Par conséquent, l'objectif de la logique floue consiste à prendre en compte divers facteurs externes pour parvenir à une décision juste sans l'intervention du décideur ou l'administrateur.

Dans les travaux relatifs au réseau véhiculaire, nous montrons la nécessité de sélectionner une passerelle appropriée pour les véhicules sans accès à Internet en fonction de plusieurs critères et en se basant sur un système de découverte basé sur Cloud Computing. Cette solution est représentée en tant que contrôleur de ressources dans un serveur de découverte de passerelles mobiles. La solution proposée se base sur une méthode d'analyse décisionnelle multicritères encourageant la recherche de passerelles mobiles appropriées à notre problème en réduisant le nombre de passerelles surchargées. La sélection d'une passerelle appropriée pour tous les véhicules nécessitant un accès à Internet est analysée en adoptant l'approche normative de l'analyse multicritères PROMETHEE (méthode d'organisation du classement de préférence pour les évaluations d' enrichissement). Notre travail se concentre sur la portée des véhicules, leurs distances, leurs vitesses, leurs directions et le nombre de clients utilisant une passerelle afin d'éviter passerelles surchargées tout en connectant les véhicules clients. Dans le dernier travail sur la sélection des passerelles mobiles dans un réseau ad-hoc de véhicules, nous améliorons la qualité de la sélection d'une passerelle mobile en prenant en compte davantage de contraintes et d'objectifs. Contrairement aux solutions de la littérature, nous ajoutons des objectifs de haut niveau, tels que maximiser le nombre de véhicules connectés et minimiser le volume de trafic traité par les passerelles mobiles pour éviter les situations de surcharge. Le problème de la sélection est adapté pour être modélisé en utilisant plusieurs objectifs contradictoires ; par conséquent, nous utilisons une optimisation multiobjective pour trouver un compromis entre les objectifs. Dans cette solution, le décideur sera mieux placé pour faire un choix lorsque de telles solutions de compromis sont exposées. Nous proposons trois approches pour résoudre le problème multiobjectif.

Contents

List of Figures

List of Tables

List of Abbreviations

5G	**F**ifth **G**eneration mobile network
4G	**F**ourth **G**eneration mobile network
CN	**C**ore **N**etwork
QoE	**Q**uality **o**f **E**xperience
LTE	**L**ong **T**erm **E**volution
3GPP	3^{rd} **G**eneration **P**artnership **P**roject
UE	**U**ser **E**quipment
eNodeB	**eNode B**ase station
EPC	**E**volved **P**acket **C**ore
EUTRAN	**E**volved **U**iversal **T**errestrial **R**adio **A**access **N**etwork
MME	**M**obility **M**anagement **E**ntity
S-GW	**S**erving **G**ateway
P-GW	**P**acket Data Network **G**ateway
QoS	**Q**uality **o**f **S**ervice
RAN	**R**adio **A**access **N**etwork
APN	**A**ccess **P**oint **N**ame
PCRF	**P**olicy and **C**harging **R**ules **F**unctions
HSS	**H**ome **S**ubscriber **S**erver
VM	**V**irtuel **M**achine
NFV	**N**etwork **F**unction **V**irtualization
VNF	**V**irtual **N**etwork **F**unction
SDN	**S**oftware **D**efined **N**etworking
DC	**D**ata **C**enter
VANET	**V**ehicular **A**d-hoc **N**etwork
ITS	**I**ntelligent **T**ransport **S**ystems
CV	**C**lient **V**ehicle
MG	**M**obile **G**ateway
LP	**L**inear **P**rogramming
IP	**I**nteger **P**rogramming
CP	**C**onstraint **P**rogramming
CSP	**C**onstraint **S**atisfaction **P**roblem
COP	**C**onstraint **O**ptimization **P**roblem

AI	**A**rtificial **I**ntelligence
MOO	**M**ulti **O**bjective **O**ptimization
WS	**W**eighted **S**um
GT	**G**ame **T**heory
TFN	**T**riangular **F**uzzy **N**umber

Chapter 1

Introduction

1.1 Background of Research and Basic Concepts

1.1.1 Background of Research

The 5G, which is the fifth generation of standards for mobile telephony, is not yet defined and is not official, but the term is used to designate the next generation successor of 4G in some newspapers and documents. The project is submitted to the International Telecommunications Union (ITU) as a candidate technology for the year 2020 to allow commercial deployment (Union, 2018). However, many issues are identified, and many actors see this as an emerging market, potentially rich in new applications and opportunities; 5G could, for example, enable new digital uses in various fields such as health (automatic or remote diagnosis, remote-controlled surgery, and medication), work (telework), deployment of communicating objects (including cars and other vehicles without drivers), detectors and sensors of e-commerce, smart grids, artificial intelligence, security (remote monitoring, management of the flow of people, cars, goods, and services in real time ...), the education and access to information. 5G technology is a critical technology that could enable mobile data rates of several gigabits of data per second; These data rates are likely to meet the growing demand for data with the rise of smart-phones and communicating objects, connected in a network. This type of system should promote cloud computing, the integration, and interoperability of interacting objects and smart grids and other so-called smart networks, in an automation environment and a smart city. It could also develop 3D or holographic imaging, data mining, big data management and the Internet of Everything [1].

[1]The Internet of Everything (IoE) is a notion that enlarges the Internet of Things (IoT) concept on machine-to-machine (M2M) communications to define a more complex system that also includes people and processes. The idea of the Internet of Everything introduced at Cisco, which defines IoE as "the intelligent connection of people, process, data, and things," unlike the IoT, where all communications are between machines. Indeed, IoT and M2M are considered similar;

5G architecture, cloud computing, and the vehicular ad-hoc network are different emerging and intertwining domains. Thus, mobile network telecommunications and vehicular networks markets could take advantages of virtualization techniques provided by cloud computing and enhance its services. The union of 5G cellular and cloud technologies will afford valuable richness in capacity, elasticity, and functionality to the mobile network operator. 5G and cloud services will enable carriers to propose competitive services, such as 5G network access to IoT-based information technologies (IT) and clouds solutions. On another hand, Artificial Intelligence (AI) a is an appropriate solution to resource management problems since it holds several techniques and their applications. Therefore, AI must be used for better and intelligent resource management. Figure 1.1 shows the interaction between the different vital domains mentioned in this thesis. The union of cloud computing and AI has acquired a significant development in the world of information and communications technologies. It has the potential to improve the means data is managed, stored and, processed across various geographically distributed data-centers. This combination also offers a novel possibility for researchers to investigate across the unlimited probabilities of the future.

In every step of the way, artificial intelligence, the cloud technology, and 5G architecture are all required. Artificial intelligence is demanded to learn, to enhance management of resources autonomously and intelligently. Cloud computing technology is necessary, so we can reach more data that could be stored on a server with flexible manner. And 5G architecture is needed to enhance the vehicular and mobile networks of the future. This kind of cooperation and technological progress could be involved in every field which we can imagine of now, like education, health, agriculture, e-commerce and so on, as well as some areas that we may not even think of now.

1.1.2 Basic Concepts

Artificial Intelligence

Artificial intelligence is the set of theories and techniques used to create machines capable of simulating intelligence. The current achievements of artificial intelligence can intervene in the following functions: the help with diagnoses; decision support; solving complex problems, such as resource allocation issues; assistance by machines in hazardous tasks, or requiring great precision; task automation;

therefore, the IoE notion involves, besides M2M communications, Machine-to-people (M2P) and technology-assisted people-to-people (P2P) communications.

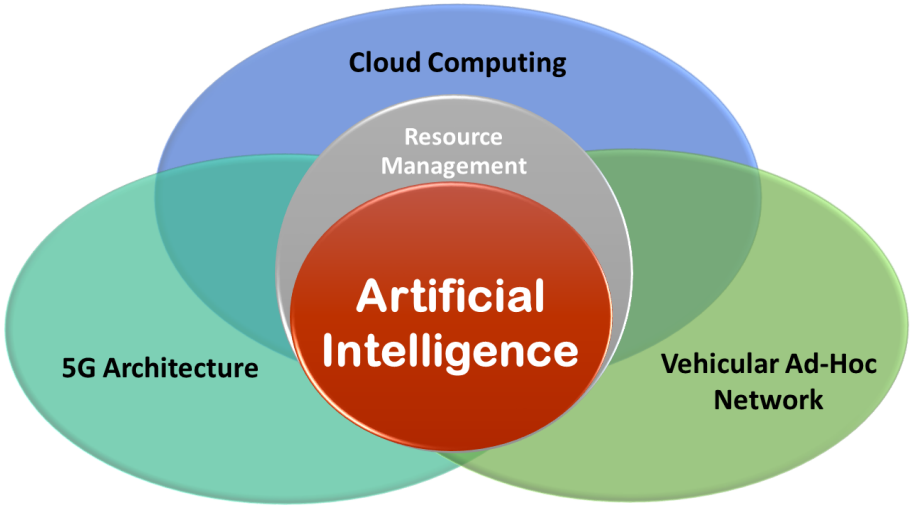


FIGURE 1.1: Background of research.

and so on. The concept was born in the 1950s thanks to the mathematician Turing. In Turing (2009), this author advances the question of bringing machines a form of intelligence. He presents a test known today as the "Turing Test" in which a subject communicates blindly with another human, then with a machine programmed to give meaningful responses. If the subject is not capable of making the difference between the human and the device, then the engine has passed the test and, according to the author, can genuinely be considered intelligent. Google, Microsoft, Apple, IBM, or Facebook, all major companies in the computer world are now working on the issues of artificial intelligence by trying to implement it to a few particular domains. Each of these companies set up networks of artificial neurons consisting of servers to deal with extensive calculations in massive databases.

Nowadays, artificial vision, for example, enables the engine to precisely recognize the content of an image and then classifies this content automatically according to the nature of the object, color or face that is defined. The algorithms can optimize their calculations as they perform procedures. Spam filters become more and more effective as the user identifies an undesired message thanks to these algorithms. Speech recognition is another example and is on the rise with virtual assistants able to reproduce the words formulated in natural language and then process the requests either by responding directly via speech synthesis or with an instant translation or by making a request relating to the command.

Artificial intelligence has acquired a large number of means to solve the most challenging problems in computer science. Some of the most generals of these techniques are explained below.

- Search and optimization: Many problems in artificial intelligence can be

interpreted in theory by effectively exploring through many possible solutions. Reasoning can be considered as performing a search. For example, logical proof can be seen as searching for a path that reaches from premises to conclusions, where each level is the application of an inference rule. Planning algorithms explore through trees of objectives and sub-objectives, trying to find a path to a purpose aim. Robotics algorithms for moving parts and taking objects use local searches in configuration space. Many learning algorithms employ search algorithms based on optimization.

Constraint Satisfaction Problems (CSP) are mathematical modelizations where one looks for states or objects satisfying a certain number of constraints or criteria. CSPs are the subject of intensive research in artificial intelligence. Some of the problems that can be modeled by a CSP include the eight-lady problem, the four-color theorem problem, the Sudoku game, and the Boolean satisfiability. Real world models that could be solved using CSPs include automated planning, lexical disambiguation, sentence understanding, and resource allocation. As another example, a constraint satisfaction model is used as a puzzle solution in a Sudoku solving algorithm. The existence of a resolution to a CSP can be seen as a decision problem. The decision can be determined by finding a solution or failing to find a solution after an exhaustive search. In some cases, the CSP might be perceived to have answers beforehand, through some other mathematical inference process.

- Logic is applied for knowledge description and problem-solving, but it can be used to other problems as well. Numerous different sorts of logic are used in artificial intelligence research. Propositional logic includes truth functions such as or and not. First-order logic joins quantifiers and predicates and can reveal facts about objects, their properties, and their relations with each other.

  Fuzzy logic is considered as an extension of classical logic to approximate reasoning. It consists of taking into account many different factors to choose that one wishes to accept. Formalized by Zadeh in 1965, it is an artificial intelligence tool which is used in various fields. Fuzzy logic is a rule-based scheme that can rely on the substantial experience of an operator, especially useful to obtain expert operator knowledge. Fuzzy logic is a kind of artificial intelligence tool; therefore, it is considered a branch of artificial intelligence. Since it is producing a form of decision making, it can be added as a member of the artificial intelligence software toolkit. Fuzzy logic can be

applied to in such varied fields applications. It has also been employed in medical diagnosis systems and handwriting recognizing applications. Indeed, a fuzzy logic system can be implemented to almost any sort of system that has inputs and outputs.

- Classifiers and statistical learning methods: The most straightforward artificial intelligence applications can be divided into two characters: classifiers and controllers.

  Controllers classify conditions before inferring actions, and therefore classification constitutes a fundamental part of many artificial intelligence systems. Classifiers are functions that employ pattern matching to define the closest match. They can be tuned according to models, making them very attractive for use in artificial intelligence. These cases are known as observations or patterns.

  A classifier can be perceived in various ways; there are multiple statistical and machine learning approaches. The decision tree is the most generally used machine learning algorithm. Other extensively used classifiers are the neural network and k-nearest neighbor algorithm. Classifier performance depends significantly on the characteristics of the data to be classified, such as the dataset size, the dimensionality, and the level of noise. Model-based classifiers work well if the assumed model is a perfect fit for the actual data.

- Artificial neural networks: A neural network is an interconnected combination of nodes, and the concept was introduced to imitate the architecture of neurons in the human brain. A simple "neuron" N accepts input from many other neurons, each of which, when activated, fixes a weight for or against whether neuron N should itself activate. Learning requires an algorithm to improve these weights based on the training data; a straightforward algorithm is to raise the weight between two connected neurons when the activation of one triggers the successful activation of another. Neurons have a continuous spectrum of activation; besides, neurons can process inputs in a nonlinear way rather than weighing straightforward votes.

  The principal classes of networks are acyclic or feedforward neural networks (i.e., where the signal moves in only one direction) and recurrent neural networks (i.e., which enable feedback and short-term memories of previous input events). Amongst the most common feedforward networks are perceptrons, multi-layer perceptrons, and radial basis networks. Neural networks can be implemented to the problem of intelligent control (e.g., for robotics) or learning.

Multi-objective Optimization

Optimization is a part of mathematics that tries to represent, examine and solve analytically or numerically the dilemmas of minimizing or maximizing a function on a set. Optimization is divided into sub-disciplines, according to the form of the objective function and that of the constraints. We cite the optimization in finite or infinite dimension, here we speak of the size of the vector space of the variables to be optimized. In the combinatorial optimization, the variables to be optimized are discrete. We also cite linear optimization which is distinguished with affine functions and quadratic which is identified with quadratic objective and affine constraints. In multi-objective optimization, a compromise between several conflicting goals is sought.

Multi-objective optimization is a branch of combinatorial optimization whose particularity is to seek to simultaneously optimize several objectives of the same problem (against a single goal for general combinatorial optimization). It differs from multidisciplinary optimization in that the objectives to be optimized here relate to an only problem.

Multi-objective issues are of growing interest in the industry, where managers are forced to try to optimize conflicting goals. Their resolution in a reasonable time becomes necessary and interests some of the researchers working in operational research. However, it is impossible to define the optimal value of a multi-objective optimization problem in general. Instead, there is a set of optimal values, forming a Pareto boundary.

Cloud Computing

Cloud computing consists of exploiting the computing or storage power of remote computer servers via a network, generally the Internet. The servers are rented on demand, most often by use, according to technical criteria (power, bandwidth, etc.), but also to the fixed price. The flexibility characterizes cloud computing, depending on the skill level of the client user, it is possible to manage your server or use remote applications in Software as a Service mode.

The primary services offered in cloud computing are SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service). Depending on the service, operating systems, infrastructure software and application software will be the responsibility of either the supplier or the customer. Large companies in the information and communication technologies sector are developing cloud computing by investing heavily in offering themselves and their customers' computing power and information storage; This is a significant

paradigm shift from computer systems, previously made up of scattered servers in businesses and communities. Cloud computing can save money, especially by pooling services across a large number of customers. Some analysts say that governments and private companies could achieve savings on their information and communication technologies budget if they migrated to cloud computing. As with virtualization, cloud computing can be as valuable to the customer as it is scalable. Indeed, the cost is a function of the duration of use of the service rendered and does not require any prior investment (i.e., man or machine). The elasticity of the cloud makes it possible to provide scalable services and can support load ramps. Furthermore, the supplier has control over the investments, is master of the rates and the catalog of the offers and can be paid more efficiently by the customers.

Other services are also available at cloud computing providers:

- Data as a service: Corresponds to the provision of relocated data somewhere on the network. It is a technique of charging a subscription for access to a data warehouse via an interface provided by the provider.
- Business process as a service: Known as BPaaS, which consists of outsourcing a sufficiently industrialized company procedure to directly address the managers of an organization, without requiring the help of IT professionals.
- Desktop as a service: Also called DaaS, virtual office or hosted virtual office is the outsourcing of a virtual desktop infrastructure to a service provider. Generally, the desktop as a service is offered with a paid subscription.
- Network as a service: NaaS is the provision of network services, following the concept of software-defined networking.
- Storage as a service: STaaS is the storage of files from external service providers who host them on behalf of their customers.
- Communication as a service: CaaS corresponds to the provision of communication solutions substituting resources and local servers for shared resources on the Internet.

Cloud computing features of interest to businesses are reducing the total cost of ownership of computer systems, the ease of increasing or decreasing resources. The use of cloud computing makes it possible to offload the information and communication technologies teams of companies, which have more availability for high value-added activities.

We distinguish three types of cloud computing deployment which are presented as follows:

- Private cloud is cloud infrastructure produced solely for a single organization, whether maintained internally or by a third party and hosted either inside the organization or externally. Undertaking a private cloud project needs significant commitment to virtualize the business environment, and requires the company to reevaluate decisions about existing resources. It can develop business, but every step in the project raises security problems that must be addressed to prevent serious vulnerabilities.

- A cloud is named a "public cloud" when the services are provided over a network that is accessible for public use. Public cloud services may be free. Technically there may be slight or no distinction between public and private cloud architecture, however, security concern may be substantially different for services (i.e., applications, storage, and other resources) that are made accessible by a service provider for the public and when communication is achieved over a non-trusted network. Regularly, public cloud service providers like Amazon Web Services (AWS), Oracle, Microsoft and Google own and manipulate the infrastructure at their data center, and access is generally via the Internet. AWS, Oracle, Microsoft, and Google also propose direct connect services named "AWS Direct Connect," "Oracle FastConnect," "Azure ExpressRoute," and "Cloud Interconnect" respectively, such connections need customers to buy or rent a private connection to a peering point proposed by the cloud provider.

- Hybrid cloud is a union of two or more clouds services (private or public) that remain separate entities but are bound together, giving the advantages of multiple deployment models. A hybrid cloud service can be defined as a cloud computing service that is composed of some combination of private, public cloud services, from different service providers. It permits one to enlarge either the capacity or the ability of a cloud service, by aggregation, integration or customization with an extra cloud service.

  Different use cases for hybrid cloud structure exist. As an instance, an organization may store sensitive client data on a private cloud application, but interconnect that application to a business intelligence other application provided on a public cloud as a software service. This sample of a hybrid cloud prolongs the capabilities of the enterprise to produce a specific business service through the addition of externally open public cloud services.

Hybrid cloud adoption depends on many factors such as data security, and compliance obligations, level of control demanded over data, and the applications an entity uses.

1.2 Motivation

The 5G architecture is a new promising framework that uses existing technologies including cloud computing to meet the requirements of many applications that increase daily data traffic. This thesis is part of the enhancement of 5G architecture solutions that are coming with a significant change for citizens in homes, workplaces, and roads. Experts believe that this technology brings hope and dramatically improves the mobile experiences of millions of smart-phone users, and will help enhance our homes through the Internet of Things. Indeed, the benefits of 5G will go far beyond improving the cities in which we live and work. Initially, the goal is to contribute to the solutions of the 5G architecture of the future, in the following two sectors: mobile communication networks and vehicular networks. To the best of our knowledge, the works cited in state of the art do not present solutions in real time, autonomous and adaptable to external factors. The proposed solutions will bring a plus in the implementation of the 5G architecture that requires intelligent resource management. The two main problems studied in this thesis are the virtual networks functions placement in the mobile networks and the mobile gateways selection in the vehicular networks. The work in both these two problems is essential for adequate resource management. It leads us to use artificial intelligence methods and techniques to propose resource controllers that respect the objectives and the constraints of each problem.

The main research questions were: Is there any modeling that improves the solutions already proposed for virtual network function placement and mobile gateway selection? Does the improved modeling give better results by finding a trade-off with the conflicting objectives? Can the implementation offer a real-time and autonomous solution? Can this solution help a decision maker to find an adequate solution depending on the external factors of the problem? Can we add the notion of adaptability to the solution taking into account external factors without the intervention of any decision maker? Can the solution be implemented efficiently in real systems?

1.3 Contribution

The novel and original contributions displayed in this thesis are listed in this section. It is worth stressing out that the main contribution is the amelioration of some resource controllers in cloud-based systems. These systems concerns two principal axes in 5G architecture. The first axis is mobile network communication where we improve the placement of the virtual network functions. And the second axis concerns the vehicular ad-hoc network where we enhance the selection of mobile gateways.

The contributions are detailed as follows:

- Axis 1: Mobile network communication
  - First, we have sought to improve the proposed solution in state of the art. The goal is to suggest a new multi-objective optimization modeling using constraint programming that provides a real-time placement of virtual functions in contrast to the solutions of the literature.
  - Then, the goal is to improve the solution proposed previously, by developing a new system that makes the placement of the virtual machines autonomous, so that it no longer depends on parameters set by the decision maker.
  - Finally, we propose an improved solution using a fuzzy controller that enables the solution to adapt to external factors, besides the optimization of the conflicting objectives.
- Axis 2: Vehicular ad-hoc network
  - First, we propose a solution that introduces a method of multiple-criteria decision analysis helping to find suitable gateways to the selection problem adding additional objectives in contrast to the solution of the literature.
  - Then, we improve the solution proposed previously, by developing a multi-objective optimization system that helps the decision maker to choose the appropriate selection depending on external factors.

1.4 Thesis Organization

This thesis consists of six chapters. The content of each chapter is detailed as follows:

- Chapter 1 presents a general introduction. In this chapter, the main thesis' topics are defined and discussed. The motivation of this research is given in this chapter, besides the main contribution.

- Chapter 2 gives a review of the two domains of application. 5G architecture is presented, and some of its challenges are displayed. A survey on virtual network functions placement in the mobile network communication, and another on the selection of mobile gateways in the vehicular ad-hoc network are given in this chapter. The review is not exhaustive instead presents some works of the most important ways to deal with these two problems. Finally, this chapter contains a definition of some multi-objective optimization methods, and a comparison of the most critical search and optimization solvers is studied and presented in this chapter.

- Chapter 3 presents a new approach for dealing with the problem of virtual network functions placement problem which is constraint programming. This approach is proposed to enhance the works presented in Chapter 2 and bring a real-time solution with multi-objective optimization. And to improve the first solution an autonomous system is also proposed.

- Chapter 4 presents an improvement of the work performed in the previous chapter. In this chapter, the multi-objective optimization solution is enhanced to become an adaptable solution to the external factors thanks to the use of a fuzzy controller.

- Chapter 5 presents a new approach for selecting an adequate mobile gateway in the vehicular ad-hoc network. This solution enhances the literature works by adding some objectives to the problem. And a multi-objective optimization system is also proposed to help the decision maker to choose the adequate solution depending on the high-level information that represents the external factors.

- Chapter 6 concludes the research work presented in this thesis and give a summary of contributions and results. This chapter also highlights some major suggestions for future research and some prospects. Finally, journal and conference publications are listed.

Chapter 2

State of the Art

2.1 5G Mobile Network Architecture

2.1.1 Introduction

New needs required by emerging communication obligations necessitate a fifth generation (5G) mobile network. Indeed, the evolution of communication thanks to all connected devices made the society highly networked. Consequently, information and data must be available everywhere and every-time for everyone. Unlike fourth-generation (4G) cellular networks, the fifth generation networks are designed to offer numerous new services that are launched efficiently and cost-effectively. Those services include all daily use cases for stakeholders in different domains such as automotive, transport, energy, food and agriculture, education, health-care, etc. 5G architecture is a new framework that uses existing technologies to meet the requirements of numerous applications which increase data traffic daily. The three main goals of this emerging network could be summarized, first, in producing ultrafast networks that deliver gigabytes of bandwidth. And secondly billions of connections between sensors and machines (Machine-type Communication). And finally, the high reliability that enables remote control of autonomous devices such as vehicles. These three aims bring a lot of challenges to the network conception; thus researchers determine and analyze the envisioned 5G use cases related to their research areas.

To accelerate the service delivery of 5G networks advanced integration of large computing and storage infrastructures is required. Cloud computing techniques are used to allocate appropriate computing and network resources to deploy specialized networking and computing functions that meet the needs of the service providers. Thereby, logical networks are created in the Cloud containing dedicated functions and are called network slices. The networks of the future enable the flexibility thanks to the elasticity allowed by the Cloud resources in

implementing network slices with an adaptability way and an efficient cost. Indeed, the network will shift from hardware entities to virtual function entities that create a new networking paradigm. This paradigm allows the telecommunication operators to deploy network slices on-demand depending on the available Cloud resources, the nature of the communication links and the desired topology. One of the bases of 5G architecture is the network softwarization that consists of running multiple logical network functions on physical infrastructure. In the mobile network of the future, hardware appliances and middle-boxes such as routers, firewalls, load balancers, etc. are replaced by VNFs that will perform a vital role particularly in the design of Core Network (CN) functions. Figure 2.1 highlights the most crucial challenges facing 5G architecture and the facilities to address these difficulties (Agyapong et al., 2014; Gupta and Jha, 2015). The 5G design must meet the high capacity and data rate besides the massive number of connections. In another hand, the latency and the cost have to be reduced while maintaining a consistent quality of experience (QoE). In this thesis, the work focuses on one of these 5G design principle depicted in this figure which is the use of an intelligent agent to manage QoE, routing, mobility and resources allocation.

2.1.2 Background

The evolution of wireless technologies has modified the style in which society lives. From the first generation (1G) to the 5G that replaces the fourth generation (4G) a remarkable change has been noted. Along with this evolution, Long-Term Evolution technology (LTE) and Fixed Worldwide Interoperability for Microwave Access (WiMAX) represented the future of mobile data services. LTE is a project led by the 3rd Generation Partnership Project (3GPP) standardization organization that aimed at writing technical standards. LTE allows data transfer at very high speed, with a more extensive range, a higher number of calls per cell (zone in which a mobile transmitter can connect with terminals) and a lower latency and it consists of three parts. The User Equipment (UE) represents the first part, the Evolved UMTS Terrestrial Radio Access Network (EUTRAN) is the radio part that incorporates eNodeB (eNB) base stations, and Evolved Packet Core (EPC) represents both the brain and the muscles part of the network. Evolved Packet System (EPS) includes both Radio Access Network (RAN) and EPC networks. The LTE network architecture is depicted in Figure 2.2 and consists of the following entities:

- The eNodeB has an interface S1 that links it with the core network. The interface S1 consists of S1-C (S1-Control) between the eNodeB and the MME

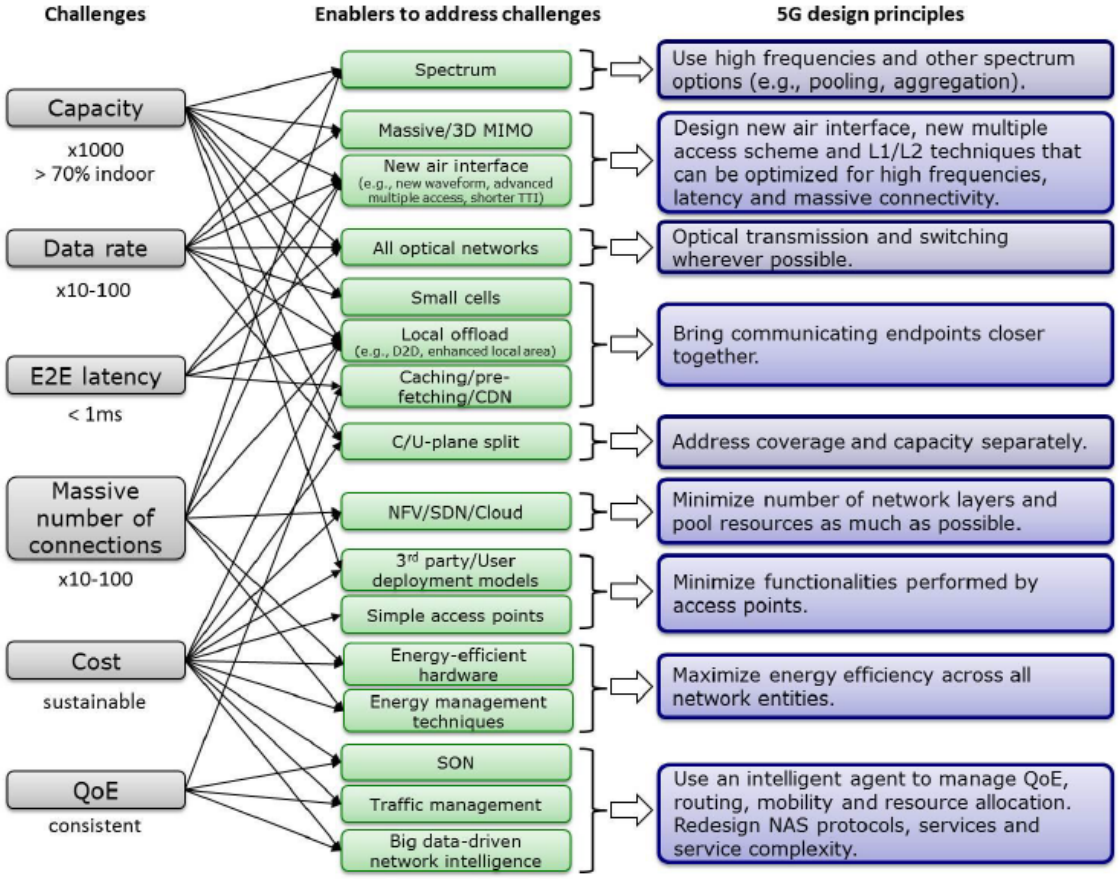


FIGURE 2.1: 5G challenges and solutions.

Source: Agyapong et al. (2014)

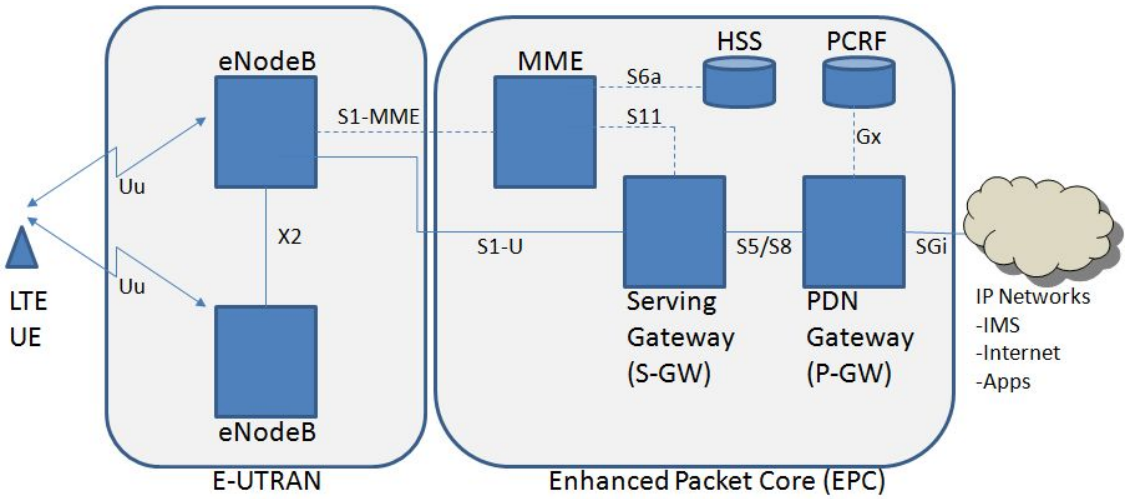


FIGURE 2.2: LTE Architecture.

and S1-U (S1-User) between the eNodeB and the Serving Gateway. A new interface named X2 has been defined between adjacent eNodeBs. Its role is to minimize the loss of packets during the mobility of the user in ACTIVE mode (handover). When the user moves in ACTIVE mode from one eNodeB to another eNodeB, new resources are allocated on the new eNodeB for the UE; however, the network continues to forward incoming packets to the old eNodeB until the new eNodeB informs the network that it is its role to relay the incoming packets for that UE. Meanwhile, the old eNodeB transmits the incoming packets on the interface X2 to the new eNodeB which delivers them to the UE.

- Mobility Management Entity (MME) is responsible for UE authentication based on information collected from HSS. The EU is informed of the location areas supported by the MME, called the Tracking Area. The EU updates its location when it finds itself in a Tracking Area that is not supported by its MME. It is up to the MME to select the Serving Gateway and the PDN Gateway that will be used to implement the Default Bearer when the EU is attached to the network. When the user is in the ACTIVE state and is moving from an area supported by an MME to another area that is under the control of another MME, then it is necessary that the handover involves the former and the new MME.

- Serving Gateway (S-GW) is on the signaling path for the establishment / release of the bearer and the media path (data packets exchanged by the UE). It is, therefore, a strategic point for the legal interception of media and control flows. The S-GW routes the outgoing packets to the appropriate PDN Gateway and relays the incoming packets to the eNodeB serving the UE. It acts as a router. The S-GW counts the number of bytes sent and received allowing the exchange inter-operator charging tickets for the payments. During an inter-eNode handover, the user traffic that was exchanged between the old eNodeB and the S-GW must be relayed from the new eNodeB to the S-GW.

- Packet Data Network Gateway (P-GW) is the entity that terminates the LTE mobile network and assures interfacing to IPv4 or IPv6 external networks. In other words, it communicates user data to and from external data networks (Internet) The P-GW assigns the UE its IP address and can allocate an IPv4 or IPv6 address. The essential functions of this LTE component are IP

allocation to the UE, maintaining the connection to the network while moving from one place to another, billing, charging support, Quality-of-service (QoS) functions, and packet filtering.

- Home Subscriber Server (HSS) contains subscription information for GSM, GPRS, 3G, LTE, and IMS networks. It holds track of the user's current MME address and holds important pre-shared material employed to generate session authentication data that assists authentication purposes. It is a database that is used simultaneously by 2G, 3G, and LTE networks, belonging to the same operator. To sum up, the authentication method in LTE is a challenge response protocol which is performed between the UE and the MME based on the data that the HSS has produced and provided.

- Policy and Charging Rules Function (PCRF) is a software node which is responsible for policy enforcement, as well as for commanding the flow-based charging functionalities which reside in the P-GW. The PCRF will give QoS information to the P-GW, determine the charging policy for data packets and dynamically manage data sessions.

Present wireless based technologies, like LTE technology, will be incorporating new technology components that will be helping to meet the challenges of 5G and to fulfill the needs of the future. Therefore, in the next section, we present a solution that marries the cloud computing with the telecommunication market.

2.1.3 Cloud Computing and Mobile Telecommunication

The evolution of information and communications technology has a significant impact on the telecommunication and the infrastructure services offered to customers. Mobile operators are experiencing an increase in the number of smartphones, tablets, and other connected devices and have to face it efficiently. The wide diversity of applications creates an augmentation of traffic data such as mobile multimedia streaming services that are primarily used by mobile users nowadays. Typical examples of user applications also, include social network applications and location-based check-in services. This changing in the behavior lead the mobile operator to adopt new solutions with a reduction in costs and energy consumption. Therefore, researchers take advantage of Cloud Computing techniques and propose a customized and dynamically scaling mobile network architectures instead of the current centralized architecture that does not evolve well with increasing needs. The principal enablers of the fusion of mobile networks and Cloud computing are the following technologies:

- Virtual machines (VMs) which are an illusion of a computer device and run programs in identical circumstances as those of the machine simulated. This illusion is performed by an emulation software that simulates the presence of hardware and software resources such as memory, processor, hard disk and even the operating system and drivers (Thielen, 1996). VMs allow abstracting from the properties of the physical machine and allow high portability of software. It is the lowest level service offered by Cloud Computing, and it consists of providing access to a virtualized computer park (Chee and Franklin Jr, 2010). The consumer can install an operating system and applications on VMs. Thus, he is exempt from the purchase of computer equipment. This service is similar to traditional data center hosting services. Amazon, Citrix, Google, HP, IBM, Intel, and Microsoft are among the leading companies in this industry. The elasticity of the cloud makes it possible to provide scalable services and can support charge inclinations. So the supplier has control over the investments.

- Network function virtualization (NFV) is a notion that aims to decouple the software component from the hardware component of a network node using virtual hardware abstraction techniques. Indeed, NFV uses the full-blown virtualization technology to change the current infrastructure of network operators. This concept implements network functions such as firewalls or P-GWs through software virtualization techniques and runs them on VMs. It allows to flexibly deploy a mobile network for mobile operators thanks to the physical node that dynamically runs different network functions as per the need. Several works have been conducted in the recent literature utilizing this idea to virtualize mobile networks. And other research works have been carried out to address the most technical challenges facing this concept such as manageability, reliability, and security (Han et al., 2015). Instantiation of Virtual Networks Functions (VNFs) must be in the right locations and at the right time. One of the use cases of NFV is the EPC' virtualization including the MME, HSS, SGW, PGW, and PCRF (Mijumbi et al., 2016).

- Software-defined networking (SDN) is a proper technology to interwork between the various virtualized network functions on the several VMs inside the same data center or over many data centers. This technology refers to a network architecture where the control plane (i.e., the entity that decides how to manage network traffic) is decoupled from the data plane (i.e., the entity that reroutes traffic according to the choices made by the control

plane). Additionally, the forwarding state in the data plane is managed by a remote control plane. In other words, this paradigm was proposed to guarantee the separation of the network's control logic from the underlying routers and switches (Kreutz et al., 2015). SDN aims to further logical centralization of network control and to program this network allowing new abstractions in networking.

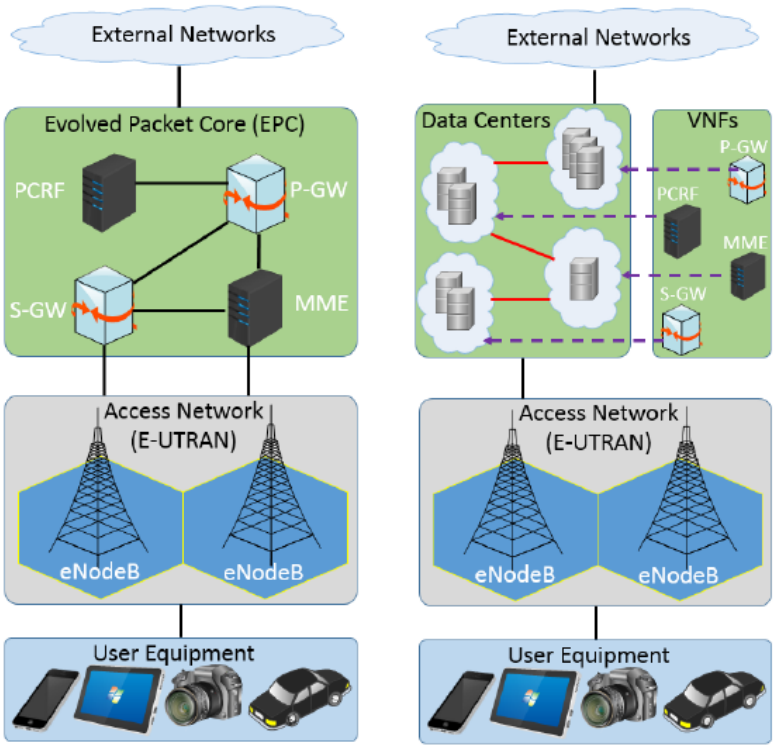


FIGURE 2.3: EPC virtualization.

Source: Mijumbi et al. (2016)

The abovementioned technologies are the fundamental enablers of the deployment of a mobile network on the Cloud that support on-demand modification and improvement. Figure 2.3 displays EPC architecture for virtualization, in the left; and in the right, the architecture after the virtualization. Since the existing LTE network architecture does not scale well to the growing demands of UEs, Taleb (2014) introduces a decentralized composition which provides an elastic mobile network as a Cloud service. This is conceivable by running VNFs over a distributed network of Cloud Computing data centers. A public network is connecting these data centers dispersed geographically. This virtual infrastructure is created using one or more Cloud providers (e.g., Amazon Web Services, Microsoft Azure, Google Cloud Platform, Rackspace, etc.). Both the RAN and the

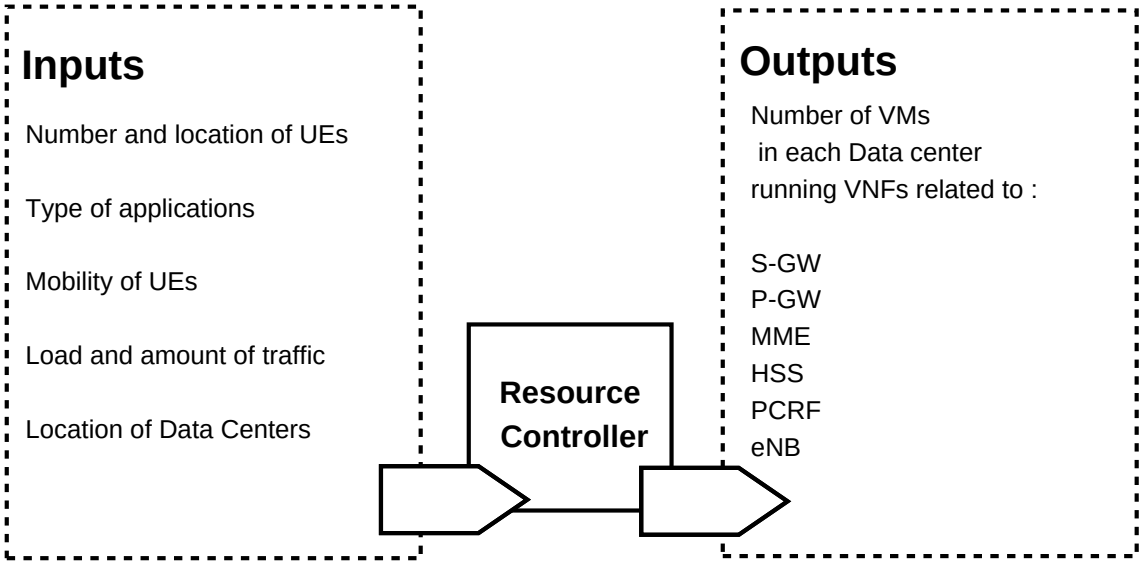


FIGURE 2.4: Resource Controller role.

mobile core network could be deployed in the Cloud. Additionally, a resource controller places and runs VNFs according to some metrics. LTE entities related images are set in the created VMs depending on the location of UEs and their mobility and application needs. Indeed, these needs are translated by the resource controller, and a network configuration is provided as output as depicted in Figure 2.4. With this proposed architecture, the mobile operator formulates his requirements to the provider of the mobile network as a service, as an example, N1 subscribers in the location 1, N2 subscribers in the location 2 and so on. Then, his needs are dynamically and automatically translated using the needs of users and even prediction of their behavior in the future. The purpose of the resource controller is crucial in providing the optimal configuration automatically.

In Taleb et al. (2015b), the authors introduce the fundamental factors in the architectural vision of EPC as a Service. The authors argue the feasibility of the mobile core network deployment into the cloud and present the different options to realize it, besides the most prominent requirements of this concept. Indeed, EPCaaS should be designed to support vertical scaling to allocate more resources to a VM and horizontal scaling to allocate more VMs to a service. Furthermore, load balancing among VMs and data centers to face a variable workload in near-real time should be assured; in addition to VMs migration and high availability of the components. It should produce the same resiliency and service quality provided by the traditional core network. The implementation of 3GPP EPC afforded "as a Service" in a cloud could be realized with complete virtualization where all control plane and user plane entities are executed in VMs or with incomplete virtualization where only control plane entities are implemented in VMs, while

user traffic is delivered and managed by hardware switches. Mobile users' data are forwarded from eNodeB via EPC through S-GW and P-GW towards a service platform or Internet, in another hand, the control plane nodes (i.e., MME, HSS, PCRF) complete the functional architecture. However, S-GW and P-GW are responsible for both control plane and data plane handling, that is why they play a fundamental role in the EPC. The authors in Basta et al. (2013) present a comparison study of EPC entities where migrating the functions to the cloud to find the optimal deployment solution. They have analyzed the EPC nodes and classified their functions according to their impact on data plane and control plane processing; these former are delivered in data centers to reduce operational cost and enhance deployment flexibility. It is worth stressing out that the manner in which the virtual entities of the EPC is configured and hosted in data centers influence on the performance of the network. For that purpose, the resource controller plays a crucial role in the virtualized LTE architecture in the cloud computing environment. Hereunder, we mention several recent works on VNFs placement.

2.1.4 Virtual Network Functions Placement

The future networks represent highly dynamic networks constructed of virtual nodes; these former are represented by instances that can be created and destroyed depending on traffic volumes, service requests, or high-level mobile operators goals. Behind the exploitation of cloud computing virtualization technology in the telecommunication domain, there is a hurdle to build software for managing and orchestrating virtualized network entities. Indeed, maintaining the VNFs life cycle under software control must be taken into consideration in the deployment of future networks and has been the subject of several works. This significant entity responsible for placing VNFs in data centers is called a resource controller as mentioned in the previous section (See Figure 2.4).

The problem of VNFs placement is perceived from various point of views as depicted in Figure 2.5. The first level consists of placing VMs on a single host taking into account the available resources such as CPU and memory usage. At this level, the problem is considered as a unidimensional problem. Several studies have been conducted to place VMs that are not necessarily VNFs in the same data center, having as objectives a more efficient resource utilization and even fewer overload situations. This level is considered a well-researched study where authors proposed numerous algorithms to find optimal solutions. For example, in Somani et al. (2012), the authors present an algorithm to place VMs into a Cloud infrastructure considering its continuous resource usage regarding CPU,

memory, storage, and networks bandwidth resources. The placement decision is made by calculating a per-VM 3-dimensional resource utilization vector. In other examples, the problem is often considered as an allocation or arrangement problem and solved using well-known Bin-packing, Simulated Annealing, and Ant Colony algorithms as well as others.

Regarding the second level, the problem becomes more complex by including more objectives. At this stage, the placement is across a large-scale distributed systems composed of multiple Data-Centers (DCs) and considers the elasticity over the Cloud, the interaction between VMS, and even the incurred costs. A genetic algorithm combined with fuzzy logic to find a trade-off between the objectives is proposed in Xu and Fortes (2010). The authors aim at minimizing total resource wastage, power consumption, and thermal dissipation costs. In Meng et al. (2010), an algorithm for improving the scalability of data center networks with a traffic-aware VMs placement is proposed. Another study proposes a dynamic load distribution policies that take into consideration all electricity-related costs in multiple geographically distributed DCs (Le et al., 2011). In the same logic, the authors in Biran et al. (2012) introduce a new optimization problem for a stable network-aware VM placement for Cloud systems. The authors introduce a framework for VMs placement and migration in Mann et al. (2011) to satisfy the constraint of network power reduction. This study takes into account both the network topology as well as network traffic demands. Concerning VNFs placement, an autonomous placement controller is presented for cost savings, better utilization of computing resources and less common overload situations is presented in Hyser et al. (2007). The main novelty in this article compared to others is the autonomic mapping of VMs and the DCs. All these previous works focalize on decreasing the cost of networking respecting the aggregated resources requirements. The problem of VMs placement aiming the elasticity in the cloud was also studied (Verma et al., 2008; Gong et al., 2010; Sharma et al., 2011; Shen et al., 2011). The elasticity over the cloud enables to install, remove or migrate VMs considering host resources capacities, and the migration allows going from one configuration to another with a transparency manner.

At this level, the elasticity and the costs such as the incurred cost of VMs or even the electricity cost as in Le et al. (2011) are highly regarded. As an example of the combination of these constraints together, we cite the work of Ghaznavi et al. (2015) where an elastic VNFs placement model is presented for minimizing operational costs in providing VNF services while considering the elasticity overhead and the host resources consumption.

In the third level of Figure 2.5, some constrained related to the nature of VNFs

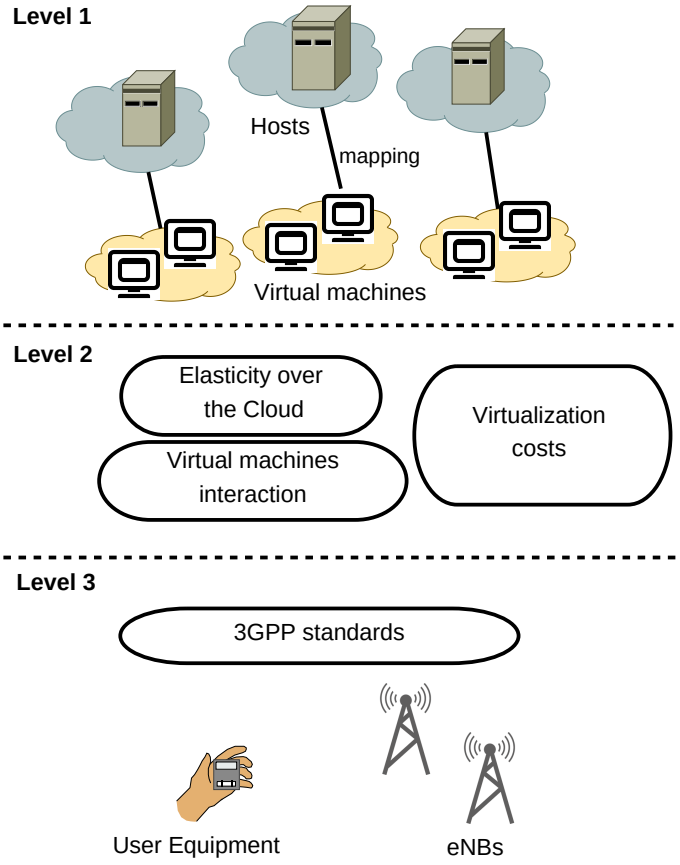


FIGURE 2.5: VNFs placement problem levels.

services are added, as an example, in Bari et al. (2015), dynamic programming was proposed for VNFs orchestration where the authors find the adequate number of VNFs that optimizes network operational costs and utilization without violating service level agreements. Another dynamic resource controller is proposed in Lyazidi et al. (2016); this former is modeled using mixed integer linear problem, in addition to a Knapsack formulation to provide a high satisfaction rate for mobile users and minimal power consumption. Automated, dynamic and topology-aware resource management approach for VNFs was introduced in Mijumbi et al. (2017); the authors used a graph neural network-based algorithm that predicts future resource requirements for each VNF component. In an SDN-based core network that allows decomposing the mobile network into the control plane and data plane functions, Hock et al. (2013) proposed a framework that provides to operators the adequate placement according to certain constraints related to the maximum latency, the failure tolerance, and the maximum number of nodes. Additionally, a trade-off between these metrics is sought in this paper. Clayman et al. (2014) implemented an orchestrator that ensures the automatic placement of VNFs and the allocation of network services on them. They proposed the least

use, least busy and N-at time algorithms for solving the problem of placement. In Mehraghdam et al. (2014), a Mixed Integer Quadratically Constrained Program for placing the VNFs in different DCs was proposed, this problem is based on tenants and operators requirements in term of remaining data rate, latency, and the number of used network nodes. Basta et al. (2014) aimed at minimizing the transport network load overhead against several parameters such as data-plane delay, number of potential DCs and SDN control overhead. In another work, a new integer linear programming formulation which minimizes the cost of occupied link and node resources was proposed (Baumgartner et al., 2015). Cohen et al. (2015) proposed a placement algorithm for VNFs EPC components with reducing the distance cost between the clients and the virtual functions by which they are served, as well as the setup costs of these functions. Their proposed algorithm outperforms Greedy Algorithm and introduces a constant factor related to some constraints that could be violated because the users have multiple demands depending on the service and the functions have a capacity and serve a certain number of clients.

Concerning 3GPP standards, some works have been conducted to place some EPC components namely S-GWs and P-GWs. All these works have dealt with the problem of the placement respecting some constraints related to 3GPP standards. In Taleb and Ksentini (2013), the authors proposed a greedy algorithm to minimize the frequency of S-GWs relocation. A gateway relocation is carried out when a UE moves from a serving area to another one, and this metric must be reduced due to the incurred cost and its impact on the overall QoE. The authors presented an optimal placement for VNFs as a solution to an NP-hard problem. In another hand, Bagaa et al. (2014) proposed a nonlinear optimization problem to manage the deployment of P-GWs. This placement reduces the paid cost by the operator with decreasing the number of the created instances. Additionally, their solution aims at improving the QoE thanks to the load balancing depending on the applications used by UEs. However, it is worth stressing out that a good solution for VNFs placement must consider all EPC entities together and their significant conflicting constraints. That is why Taleb et al. (2015a) introduced a solution that improves the QoE by placing P-GWs near its respected UEs and avoids S-GWs relocation to reduce the cost paid by the operator. As mentioned earlier, the constraints are conflicting, and a compromise must be sought because the second objective is realized when S-GWs are placed far away from UEs and service areas are more considerable. The authors opted for a game theory approach to solve the placement problem while achieving a trade-off. Game theory is also used in Ksentini et al. (2016), where the authors realized a compromise

between minimizing the number of S-GWs relocations and minimizing the load in S-GWs.

2.1.5 Conclusion

Mobile telecommunication domain is experiencing new trends with the combination of cloud computing techniques. Many hurdles in mobile network architecture, such as the lack of flexibility for deploying a new architecture that meets the UEs needs with a fair cost. Furthermore, NFV is a new paradigm that plays a crucial role in the migration of mobile services in the cloud. Indeed, VNF related to EPC components, for example, could be run on physical hosts all under software control. It remains to know how to efficiently place those VNF on distributed DCs provided by Cloud Computing services. Also, this placement involves different restrictions, for instance, some constraints related to the QoE and incurred costs. In Table 2.1, all essential metrics and constraints in the aforementioned related works are presented. Furthermore, the associated works considering 3GPP standards and mentioned earlier are shown in Table 2.2. We notify that most works take into consideration the constraints related to the physical resources limitations and the costs. Also, the elasticity of the cloud is highly regarded in the works. However, a few works have presented an autonomous placement that is considered self-sufficient or an adaptable placement to the change of the external environment. Despite the importance of these two constraints in obtaining a suitable and real solution, it is not considered in the works dealing with the placement of EPC entities according to the 3GPP restrictions. Resources allocation, the paid cost, and the elasticity are discussed in these works as mentioned in Table 2.2, but the real hurdle is how to deal with all the conflicting constraints and objectives.

TABLE 2.1: Considered metrics and constraints in related works.

	Resources-aware	Cost-aware	Elasticity-aware	Autonomous placement	Adaptable placement	Service-aware
Somani et al. (2012)	x					
Xu and Fortes (2010)	x	x				
Meng et al. (2010)	x		x		x	
Le et al. (2011)	x	x				
Biran et al. (2012)	x		x		x	
Mann et al. (2011)	x	x	x			
Hyser et al. (2007)	x	x		x		
Verma et al. (2008)	x	x	x			
Gong et al. (2010)	x	x	x	x		
Sharma et al. (2011)	x	x	x			
Shen et al. (2011)	x	x	x	x		
Ghaznavi et al. (2015)	x	x	x			
Bari et al. (2015)	x	x				x
Lyazidi et al. (2016)	x	x	x			x
Mijumbi et al. (2017)	x			x	x	x
Hock et al. (2013)	x	x				x
Clayman et al. (2014)	x	x		x		x
Mehraghdam et al. (2014)	x	x	x			x
Basta et al. (2014)	x	x				x
Baumgartner et al. (2015)	x	x				x
Cohen et al. (2015)	x	x			x	x

TABLE 2.2: 3GPP standards constraints and other metrics in related works.

	3GPP standards				
	P-GW constraints	S-GW constraints	Resources-aware	Cost-aware	Elasticity-aware
Taleb and Ksentini (2013)		x	x		x
Bagaa et al. (2014)	x		x	x	x
Taleb et al. (2015a)	x	x	x		x
Ksentini et al. (2016)		x	x	x	x

2.2 Vehicular Ad-hoc Network

2.2.1 Introduction

Vehicular Ad-Hoc NETwork (VANET) is a form of NETworks Mobile Ad-hoc (MANET) to provide communications within a group of vehicles within reach of each other and between vehicles and fixed equipment within range, usually called road equipment. VANET consists of a vehicle to vehicle and vehicle to infrastructure communications thanks to the wireless local area network technologies. Recently, the research fields recognize a significant investment and advancement in VANETs area. Different research domains are investigated such as security, transport performance, and information or entertainment applications. VANET can be used to support the development of Intelligent Transportation Systems (ITS). The idea of vehicles sharing useful information to ensure the safety of human life on the road is compelling because apart from the problem of security, all cities in the world are experiencing traffic jams and congestion. In recent years, researchers, government and the automotive industry have been interested in VANets, where several ITS applications have emerged not only for safety applications but also for more rich applications for drivers and the passengers. Therefore, many applications are proposed for VANets such as early warning and accident prevention, the best routes to the destination, congestion reduction, congestion prevention, Internet access, and peer-to-peer applications. The design and implementation of protocols, applications, and systems for VANETs need to consider its distinctive features, in particular, the high mobility of vehicles, the rapid change of topology and the intended path. Besides, it must also take into account several factors, such as different quality of service (QoS) requirements for a different type of application and reliable transmission link quality. Vehicular networks allow vehicles to communicate with each other via inter-vehicle communication (IVC) as well as with road equipment via equipment communication -to-Vehicle (Roadside-to-Vehicle Communication - RVC). The optimal goal is that vehicular networks will contribute to safer and more efficient roads in the future by giving timely information to drivers and interested authorities.

2.2.2 Background

Ad-hoc networks are wireless networks capable of organizing without previously defined infrastructure. For example from one equipment to another without an access point. Ad hoc networks, in their mobile configuration, are known as ad hoc mobile networks. Each entity (or node) interacts directly with its neighbor.

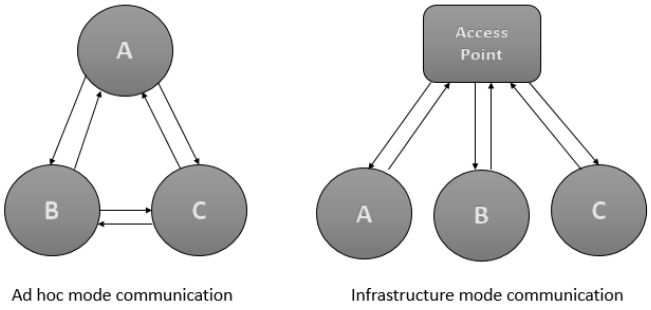


FIGURE 2.6: Ad hoc mode and infrastructure mode communication.

To communicate with different entities, it is necessary to pass on its data by others who will forward them. For this, it is first essential that the entities are located relative to each other, and can build routes between them: it is the role of the routing protocol. Thus, the operation of an ad-hoc network significantly differentiates it from a network such as the GSM network or Wi-Fi networks with access points: where one or more base stations are required for most communications between the different nodes of the network (Infrastructure mode), ad-hoc networks organize themselves and each entity can play different roles (See Figure 2.6). Among the applications of ad hoc networks, the typical application is the communication of emergency units over vast areas, for example during natural disasters. Another form is the military one since such a network can be used to connect the different groups of an army. Sensor networks, which are ad hoc networks where the nodes hold sensors, for example for temperature, are another very prominent application in several domains, including that of home networks.

MANET is the name of an IETF [1] working group, created in 1998, to standardize routing protocols based on IP technology for ad hoc wireless networks. MANET also refers to networks without the infrastructure in which all stations can be mobile. Since the birth of this working group, the proper name MANET is sometimes used as a familiar name to designate an ad hoc network. As a first step, the working group focused on performance issues in ad hoc networks and the development of a series of experimental routing protocols, both in the family of reactive (AODV, DSR) and proactive ones. (OLSR, TBRPF), or even hybrids (ZRP). Based on this set of protocols and some experience gained, MANET has introduced some of these protocols into Internet standards and worked on

[1]The Internet Engineering Task Force (IETF) develops and promotes Internet standards, in particular, the rules that make up the Internet Protocol Suite (TCP / IP). The IETF produces most of the new Internet standards. The purpose of the group is usually the writing of one or more Request for comments (RFC), the name given to the specification documents at the base of the Internet.

the different approaches proposed by the scientific community. Among existing protocols, we cite the proactive ones where routes are maintained periodically such as FSR, OLSR, DREAM, DSDV, Babel. In the reactive protocols, the route's construction is on demand such as AODV, DSR, RDMAR. The hybrid protocols are proactive and reactive such as ZRP and Tora. The hierarchical protocols are based on a specific structure around entities elected for particular roles such as HSR, VSR, CBRD. Geographical protocols use the information on the position of mobile nodes such as GPSR and GRID.

2.2.3 Cloud Computing and Vehicular Ad-hoc Network

VANET is a useful technique in the so-called Intelligent Transportation Systems (ITS) and plays a role in improving road security and guaranteeing passenger comfort. Among the current challenges in ITS, it is to provide real-time optimization and efficient systems. As a result, advances in cloud computing are used in this domain to enhance the services provided by ITS. The combination of these two fields becomes a massive research effort in the recent past. As argues by the authors of Guerrero-Ibanez et al. (2015), realizing a sustainable, intelligent transportation system expects the integration and combination with emerging technologies such as cloud computing. Thanks to this union, new concepts are appearing such as vehicular cloud which provides all the services required by the autonomous vehicles. The authors in Gerla et al. (2014) discuss the idea of the Vehicular cloud which will help transition to the Internet of Vehicles that provides all the services required by the autonomous vehicles. Nowadays, it is possible to optimize the traffic control in the road thanks to many proposed applications deployed by IT developers. However, cloud computing with its scalable access to computing resources represents a suitable solution for combining the Internet advantages and the technology improvements used on roads. Indeed, massive investment in hardware is not needed to implement the applications if cloud computing is merged to VANET as in Olariu et al. (2011); Hussain et al. (2012); Bitam et al. (2015). The authors in Olariu et al. (2011) recommend using vehicular clouds as a technically feasible idea to enhance the experience of conductors on roads. In Hussain et al. (2012), a robust architecture for vehicular clouds is presented with different clouds scenarios in VANET. The challenges are also discussed and security hurdles mainly. The impact of cloud computing on VANET is discussed in Bitam et al. (2015) and a VANET-Cloud framework is proposed. The employment of cloud computing in VANET also concerns the routing protocols. The authors in Lin et al. (2011b) suggest a model system for gateway discovery

based on cloud computing. The proposed gateway discovery scheme effectively increases the packet delivery rate, reduces the end-to-end delay, and decreases the signaling overhead for routing to the Internet in VANETs compared with a reactive protocol. In the next section, a discussion on mobile gateway discovery and selection is given.

2.2.4 Mobile Gateway Selection

VANET environment attracts the attention of a significant number of researchers and becomes a field of interest over the last several years. All the proposed applications providing useful or even crucial information to the Client Vehicles (CVs) (e.g., safety-related applications) need access to the Internet, and due to the mobility of vehicles and the dynamic nature of the network, this is considered as a big challenge. On another hand, the development of these networks engenders the deployment of new communication technologies for the transmission of data between vehicles from which we can take advantage of and connect a vehicle without access to the Internet. The vehicles interested in accessing Internet services from within the vehicular network can access the fixed gateways. These latter are part of road infrastructure such as Stationary gateways that are access points (AP) to WiFi, or WiMAX, or cellular networks base stations (BSS). However, this kind of connection can engender some access problems because of the velocity of vehicles, hence the need for using mobile gateways (MGs). The idea of exploiting MGs located in the network addresses several issues that are in the fields of research and include the interoperability of communication protocol, the mobility support, the communication efficiency, the discovery of Internet gateways, the handover of connections from one gateway to the next, etc. Several studies were conducted to demonstrate that the idea of MGs has an excellent chance to succeed in providing global connectivity to vehicles on the road and such approach is feasible using existing ITS radio technologies Bechler et al. (2003); Namboodiri et al. (2004); Setiwan et al. (2006). And it is just recently that the idea of exploiting gateways located in the network was born. A study on the feasibility of deploying a mobile gateway was conducted, and such architecture is feasible using existing intelligent transport systems radio technologies Lan et al. (2007). The authors in Iera et al. (2009) proposed a Framework that envisages cooperation between the protocols previously used in the gateways selection problem and ensures the selection of the right gateway depending on the availability of network resources and applications requirements.

There are mainly three approaches to the discovery process: *(i)* A proactive approach where the gateway periodically sends a message to other vehicles to signal its existence; *(ii)* A reactive approach where the vehicle may actively seek to connect to the Internet before receiving the gateway message; *(iii)* A proactive or reactive hybrid approach to minimize the disadvantages of proactive and reactive methods. In Badole and Raju (2014); Badole and Nikam (2014), the authors propose a hybrid approach where a discovery mechanism gives the best performance by decreasing the overhead. However, the selection of the appropriate gateway relies on several criteria involving the application requirements, the QoE, and the stability of the path (i.e., multi-hop) to the candidate gateway, network availability and so on. In Bechler et al. (2003), the authors propose a service discovery protocol for vehicular ad-hoc networks which choose the most suitable Internet gateway among others with the help of fuzzy methods. This selection takes into account the geographical position, the number of clients using the current gateway and the available bandwidth to meet the needs of the application's requirements. Moreover, this proposed protocol is based on a proactive approach. In all these works, a gateway selection consists of sending messages between vehicles, and the main drawback of these approaches is the overload situation when there are many nodes in the network.

Given that vehicles do not necessarily know the location and availability of all gateways in their neighborhood, the effectiveness of procedures for discovery and selection of the "best" gateway proves to be a factor of the utmost importance. A gateway discovery system assisted by the Cloud is proposed to obtain an Internet connection, advance the performance of the routing and fix the problem of overload (Lin et al., 2011a,b). In Lin et al. (2011a), both the proactive and the reactive approaches are proposed for the system discovery, and the selection relies on the predicted link lifetime between the MG and the CV. This measurement is calculated based on the speed, the direction, the geographical location, and radio propagation range of the two nodes Su et al. (2001); Namboodiri and Gao (2007). The study in Lin et al. (2011b) introduces a system discovery assisted by cloud computing. The authors suggest a scheme that uses two cloud servers namely Discovery as a Service (DaaS) Registrar which maintains information related to gateways and Discovery as a Service (DaaS) Dispatcher which is responsible for MGs discovery to meet the needs of vehicles requesting access to the Internet. The MG offering most extended link lifetime with the CV is selected as the next-hop gateway. Similarly, the authors in Pan et al. (2011) present a gateway controller that search the position of the CV and determines a set of MGs close by the destination to forward the packets. The transfer is made by choosing the longest link

lifetime path. In Sivaraj et al. (2011), VANET-LTE integrated network architecture is proposed where MGs are selected according to the transmission rate and the direction of vehicles. A new technique for gateway selection in VANET network merged to the LTE Advanced infrastructure is presented in Labiod et al. (2012); el Mouna Zhioua et al. (2014).

2.2.5 Conclusion

The discovery and selection systems assisted by the cloud proved to be an efficient solution to enhance the routing protocol and to choose the best mobile gateway for the vehicle in need to Internet access which is named CV. An example of one of these architectures is depicted in Figure 2.7. A service provider in the cloud gives the cloud server. The server needs to register its services to some well-known name servers in the cloud. Two particular servers are proposed in this system, namely, the DaaS Registrar and the DaaS Dispatcher. DaaS Registrar maintains related information of the gateways. DaaS Dispatcher is responsible for discovering and dispatching the gateways for the CVs. We recall that a Gateway is an entity which can connect to the Internet directly. According to the mobility of the gateways, two types of gateways are considered in the system. The Stationary Gateway (SG) is part of the roadside infrastructure such as access points (APs) of WiFi or WiMAX, or base stations (BSs) of cellular networks; While an MG is a vehicle on the road which can directly connect to the Internet. A CV is a vehicle which needs to connect to the Internet. A relay vehicle (RV) is used in the case that a CV locates outside the coverage of any gateway, one or more relay vehicles (RVs) forward packets from or to a CV to or from a gateway. DaaS Dispatcher is responsible for mobile gateway selection taking into account the speed, the direction, the geographical location, and the radio propagation range of the MG and the CV. On another hand, this solution helps to enhance the routing protocol by incrementing the packet delivery rate, diminishing the end-to-end delay and reducing the signaling overhead. Still, others selection solutions propose to take into account the number of clients using the current gateway and the available bandwidth to meet the needs of the application' s requirements to enhance the QoS; hence the need for the improvement of the gateway discovery system.

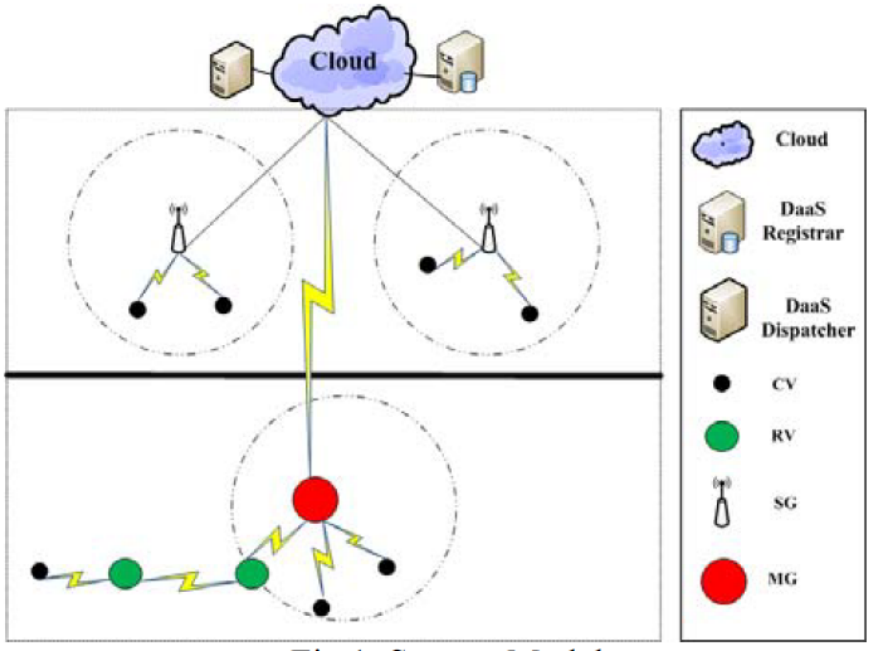


FIGURE 2.7: A gateway discovery system assisted by cloud computing.

Source: Lin et al. (2011b)

2.3 Search Methodologies and Optimization

Nowadays, an ample variety of domains, such as science, commerce, medicine, and so on, rely on computational search applications. The decision support systems found in these different areas are based on some search methodologies and optimization technologies. A computational problem is represented through mathematical modeling and deals with a collection of issues that computers might be able to solve. Optimization and search methodologies are one of the central objects of study in computer science literature. One of the significant needs today is to own systems that contain algorithms for solving computational problems efficiently. In this section, we present some relevant methods in optimization and decision support techniques, starting with some classical techniques like linear programming, going through the constraint programming and the multi-objective optimization and finally arriving at the fuzzy logic.

2.3.1 Classical Approaches

Linear Programming

Linear programming (LP) is an optimization problem with linear objective function and constraints. LP is a central domain of optimization which can model many real-world operational searches. The LP technique is commonly applied in the industrial areas. This is one of the applications, if not the main one that uses PL daily. It is the tool that allows the decision maker to find the optimal solution for production. To do so, the program must take into account many constraints. In general, a problem is represented through LP as follows:

$$\begin{cases} \textbf{min} \sum_{i=1}^{n} a_i x_i \\ \\ \textbf{Subject to} \\ \sum_{i=1}^{n} c_{1i} x_i \sim b_1 \\ \quad \ldots \\ \sum_{i=1}^{n} c_{mi} x_i \sim b_m \end{cases} \tag{2.1}$$

Where $\sim$ is $\leq$, $\geq$, or $=$. The linear objective function can be either minimized or maximized.

The simplex algorithm was introduced by Dantzig to solve linear programs. This algorithm has long been the most widely used method for solving linear optimization problems and is the first algorithm to minimize a function on a set

defined by inequalities. Variants of the simplex algorithms are available in different languages, as packages or libraries from various web-based sources. The worst case time in simplex methods is known to be essential, and that represents the main drawback even if these algorithms work very well.

Integer Programming

Integer Programming (IP) is a field of theoretical computing in which problems of optimization are modeled in a particular form. These problems are described by a cost function and linear constraints, and by integer variables. Thereby, IP is a form of LP with extra constraints. Indeed, LP turns into IP when some or all variable take an integer value, and the solution is limited to integers. Integer variables are useful in modeling a problem because it can represent quantities (e.g., 5 articles and not 4.6) and also a decision (e.g., 0 or 1). Many well-known issues could be expressed in the form of IP like the traveling salesman and the set packing problem. Furthermore, IP is applied in different areas such as planning, scheduling as well as in telecommunications networks. According to the theory of complexity, IP is considered difficult because it is an NP-hard problem (Karp, 1972). If some decision variables are not discrete, the problem is known as mixed-integer programming.

An IP problem can be put into two classical forms: the canonical form and the standard form. The canonical form for a maximization problem is:

$$\begin{cases} \textbf{max}\, a^T x \\ \\ \textbf{Subject to} \\ C^T x \leq b \\ x \geq 0 \\ x \in \mathbb{Z} \end{cases} \tag{2.2}$$

And the standard form is:

$$\begin{cases} \textbf{max}\, a^T x \\ \\ \textbf{Subject to} \\ C^T x + s = b \\ s \geq 0 \\ x \in \mathbb{Z} \end{cases} \tag{2.3}$$

Where a, b are vectors and C is a matrix with integer values.

Among the resolution methods, the naive one is the relaxation technique. An NP-hard optimization problem (IP) is transformed into a related problem that is

solvable in polynomial time (LP). For example, in a decision IP problem where constraints are in the form $x \in \{0, 1\}$, the relaxation version uses the linear constraint instead in the form $0 \leq x \leq 1$. But the solution may not be optimal or not feasible, and it may violate some constraint.

Exact algorithms are used to solve an IP, among them, we cite the cutting plane method. The principle of the method is to add constraints to the linear program to refine and bring it closer to integral solutions. Precisely, given a set of constraints, and an optimal solution x^* to the linear optimization problem, the method consists of creating new constraints, such that the whole optimal solution is preserved, but x^* violates one of the new constraints (Kelley, 1960).

Branch and bound method is also a very known technique for solving IP. It consists of separation and evaluation techniques through the help of a tree with branches. The branches represent the choices of the variable, and the nodes correspond to the partial solution (i.e., $x_1, x_2, ..., x_k$). The separation phase consists of dividing the problem into many subproblems which each have their set of feasible solutions. This separation principle can be applied recursively to each of the subsets of solutions obtained, as long as the sets are containing several solutions. The sets of solutions (and their associated subproblems) thus constructed have a natural tree hierarchy, often called a search tree or a decision tree. The purpose of evaluating a node in the search tree is to determine the optimum of the set of possible solutions associated with the node in question or, on the contrary, to mathematically prove that this set does not contain any interesting solution, for the resolution of the problem (typically, that there is no optimal solution). When such a node is identified in the search tree, it is unnecessary to perform the separation of its solution space. To determine that a set of feasible solutions does not contain an optimal solution, the most general method consists of identifying a lower bound of the cost of the solutions included in the set (if it is a problem of minimization). If we manage to find a lower bound that is higher than the cost of the best solution found so far, then we have the assurance that the subset does not contain the optimum. The most classical techniques for the calculation of lower bounds are based on the idea of relaxation of certain constraints. This technique is very commonly used in the field of operations research to solve NP-complete problems (Clausen, 1999). In particular, they are at the heart of linear integer optimization and also constraint programming solvers. The branch and the bound algorithm does not always solve an IP problem in a fair time. Indeed, it can produce an enormous number of subproblems.

Despite this complexity, IP is employed routinely to solve problems in the real world. And the key to an adequate solution is a suitable problem formulation.

Among the tips introduced in Burke et al. (2005), for solving IP, using the most current software embedding the latest methods is highly recommended. Moreover, it is worthing to use the software because it is easy, or available, or cheap. In another hand, solving some small instances and looking at the solutions to the linear relaxations is also useful. Often, it is more practical to add constraints to improve a formulation from a few small examples.

2.3.2 Constraint Programming

Constraint programming (CP) represents a programming paradigm that emerged in the 1970s and 1980s, to solve massive combinatorial problems such as scheduling problems (Haralick and Elliott, 1980; Mackworth, 1981). CP is used in different application fields such as in telecommunications, transportation, Internet commerce, electronics, bioinformatics, network management, supply chain management, and many other domains. A Constraint Satisfaction Problem (CSP) is constituted from a set of variables, and each variable has its domain. The constraints on the variables specify which combinations of value assignments are allowed and each assignment that satisfies the constraints is a solution to the problem. So a constraint is a relationship between one or more variables that limit the values of the domain that can simultaneously take each of the variables bound by the constraint.

A CSP is defined by a triplet (X, D, C) where:

- $X = \{x_1, ..., x_n\}$ is the set of the problem variables;
- $D = \{D_1, ..., D_n\}$ is the set of the domains variables, i.e., for all $k \in [1; n]$, we have $x_k \in D_k$;
- $C = \{C_1, ..., C_m\}$ is a set of constraints. A constraint $C_i = (X_i, R_i)$ is defined by the set $X_i = \{x_{i_1}, ..., x_{i_k}\}$ of variables in which it relates and a relation $R_i \subset D_{i_1} \times ... \times D_{i_k}$ which defines the set of values that can simultaneously take the variables of X_i.

A propositional satisfiability problem (SAT) is a particular case of a CSP where the variables are boolean, and the constraints are defined by propositional logic expressed in conjunctive normal form (Biere et al., 2009). CSPs can be expressed as constraint networks, where the variables are the nodes of the network, and the constraints are the arcs. Constraints including more than two variables can be represented with hypergraphs. However, the basic CSPs are represented with binary constraints including two variables. We mention that designating a domain of values for a variable is considered as giving a unary constraint on that

single variable. Consistency is a common term used in CP paradigm; We assume that a value for a variable is consistent with a value for another variable if both values satisfy the binary constraint between them. On another hand, we can regularly eliminate inconsistent values until any value for any variable is consistent with some value for all other variables, we say that we have reached arc consistency. Eliminating inconsistent values by this method is the most well-known form of the inference process which consists of reducing the space we must search through for finding a solution. Many algorithms have been generated to realize arc consistency for reducing the search space efficiently. (Nadel, 1988; Gent et al., 2008).

Regarding search algorithms, we distinguish between two main classes. The first one is systematic, complete, and determines a solution by extending a set of consistent values for a subset of the problem variables, repeatedly adding a consistent value for one more variable until a complete solution is given. The second algorithms class is stochastic, incomplete, and tries to find a solution by repairing an inconsistent set of values for all the variables, repeatedly changing an inconsistent value for one variable, until a complete solution is obtained. However, we can find hybrid algorithms where extension and repair methods are merged.

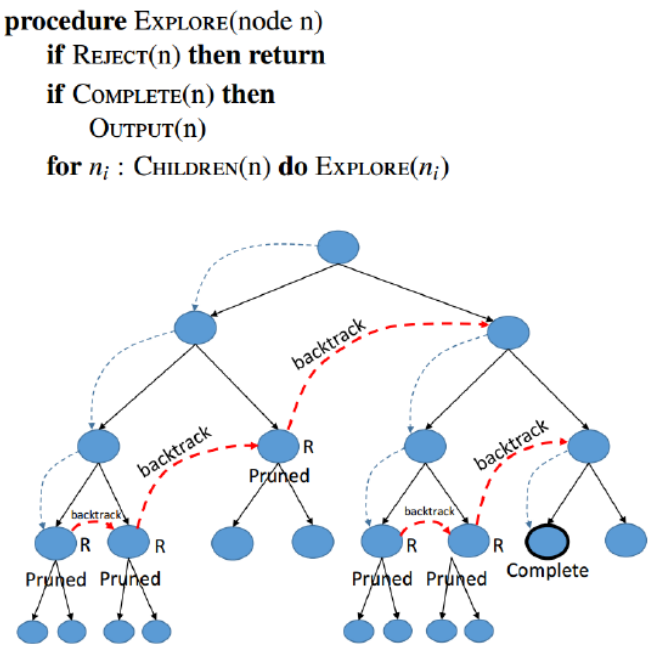


FIGURE 2.8: Backtrack algorithm.

Source: Andrews et al. (2016)

Backtrack algorithm is a well used classical extension method. Backtrack search can eliminate many combinations of values solely by identifying when

an assignment of values to a subset of the variables is already inconsistent and cannot be extended. Figure 2.8 shows a backtrack search tree depicting a track of a backtracking algorithm solving a problem. The algorithm assigns values to variables, one by one, traversing through the domains such that the constraints are satisfied. If the constraints related to the concerned variables are not met, a return backward is made, the problem is solved in a depth-first manner of the space. However, a wrong choice of values can require a tremendous number of backing and filling before the solution is reached. To address the search order problem, some heuristics are used along with the algorithm. For example, the minimal width ordering heuristic gives the variables a total ordering according to the minimal width of variables (i.e., some variables are constrained by more variables than others). Then, it labels the variables according to that ordering. To avoid backtracking, the variables which are constrained by fewer other variables are labeled last. There have been notable efforts to enhance the efficiency of the backtrack algorithm and to guide the search, including the notions of constraint propagation, intelligent backtracking, restarting policy, variable and value ordering heuristic, and so on (Dechter and Frost, 2002).

The forward checking algorithm is a method for solving constraint satisfaction problems is a common and robust alternative to backtracking. This algorithm is quite close to the Backtrack, and it assigns values to variables as and when. The difference with the procedure, seen previously, is in the management of dead ends when the algorithm can no longer find solutions. The forward checking, before choosing a value for a variable x_n, checks that this assignment corresponds to the constraints and checks that the other variables $x_{i(i>n)}$ can be affected. If the assignment of x_i cannot be made, the algorithm chooses another value for x_n. If there is no solution for x_n that allows the assignment of other variables, the procedure will roll back to x_{n-1} to change its value. This point remains identical to the Backtrack. If the CSP has no solution, we go back to the variable x_0.

In repair algorithms, the process begins with a complete assignment (i.e., all variables are affected by a value). Then, a move is made by changing the value of one variable to improve the solution. The most used repair methods are physical-inspired such as hill climbing and simulated annealing (Selman and Gomes, 2006; Dowsland and Thompson, 2012), and biological-inspired like neural networks and genetic algorithms (Haykin, 2004; Anderson-Cook, 2005). However, genetic algorithms build a new assignment by combining two previous solutions instead of replacing the old one with a neighbor solution.

If not all constraints in a problem could be satisfied, we seek a partial solution. Furthermore, if a problem has multiple solutions, there are some methods

to choose the best one. Constraints which must be satisfied are called hard constraints, and constraints which could be violated are soft constraints. These last indicate if a solution is better than another according to some preferences. For searching optimal solutions, Backtrack method could be generalized to Branch and Bound method. By doing this, a partial solution is rejected when it is evident that it could not be a better solution than one already found.

The CP paradigm supports simplicity, exactitude, and maintainability by separating the problem formulation as far as possible from the process of the algorithm applied to resolve it. The problem is formulated concerning its decision variables, the constraints on those variables, and an expression to be optimized, i.e., maximized or minimized in case of Constraint Optimization Problem (COP). That said, it is more convenient to use the tools that the constraints community uses to solve complex problems. Indeed, there are plenty of CP systems available, which support constraint propagation, search and a diversity of other methods.

2.3.3 Multi-Objective Optimization

Multi-objective optimization (MOO) is an essential and practical part of optimization. Indeed, all real-world problems contain conflicting objectives to be optimized. In this section, we present some principles of MOO, which is different from single objective optimization. Consequently, there are two or more distinct aims of optimization, instead of one, and it possesses two or more different search spaces. The objective functions constitute a multi-dimensional space. Moreover, the usual notion of only one optimal solution is not valid concerning MOO, that is why in many real-world problems analysis of the obtained trade-off between the objectives is done. Therefore, the question is which is the optimal solution belonging to this set of trade-off solution that the user can choose? We will seek to answer this question using a common problem where we are looking for a hotel that is cheap and close to the beach in any seaside town. In this case, we obtain a set of interesting solutions (i.e., hotels), however, the more the hotel is near the beach, the more is expensive. The concerned user can choose one solution depending on his budget, his ability to walk to the beach, his possession of a mean of transport or not, the state of the road to the beach, his preference concerning the two objectives (i.e., the paid price or the distance to the beach). As a result, the answer to our question involves many considerations and external factors that we will call high-level information. The user must compare between the set of optimal solution to make a choice, that is why the hard work to do is finding the set of Pareto optimal solutions by considering all objectives to be important.

A solution is called Pareto optimal when none of the objective functions can be improved in its value without affecting some of the other objective values.

The user or the decision maker must be aware of high-level information helping to solve a problem basing on the preferences related to the objectives. Hence, there are mainly three categories of MOO problem-solving approaches (Marler and Arora, 2004). In a posteriori articulation of preferences, an optimizer is used to solve the multi-objective problem, and a set of trade-off solutions is generated. The decision maker can use high-level information to choose one solution as depicted in Figure 2.9. In a priori articulation of preferences, the decision maker orders the objectives depending on its pertinence with the help of high-level information related to the problem. A relative importance vector is given to the optimizer, and one solution is generated as shown in Figure 2.9. The third category is when no articulation of preference is provided.

Regarding the solving methods of MOO problems, some known methods are given in this section. In the Weighted Sum (WS) method, all objectives are combined in a single objective function using weights for each objective. It is the most common method used to solve problems in the case of a priori articulation of preferences (Marler and Arora, 2010). On another hand, Genetic algorithms represent a crucial method in the posteriori articulation of preferences. Indeed, it generates a set of solutions, and it is helpful when it is difficult to determine an importance vector and to set weights for objectives Fonseca et al. (1993). When no articulation of preferences is given, methods that do not require any articulation of preferences are used. Nash bargaining approach which is a branch of Game Theory (GT) (Marler and Arora, 2004; Rao, 1987) is used to find a trade-off solution. This strategy is a non-cooperative game and is based on two elements. The first element is players and the second one is the utility function of the player such that each player is associated with one objective. Based on the Nash theory, each player seeks to improve the value of his objective and to avoid the worst value (Marler and Arora, 2004). The evaluation of the performance of solving methods is a vital issue in MOO. When optimizers give Pareto optimal solutions, it is hard to compare due to the number of objectives, unlike single objective optimization where we compare two solutions by maintaining the smaller value or, the greater value depending on the minimization or the maximization of objectives. Evaluating a solving method performance implies comparing the quality of solutions and the time of execution. In Zitzler et al. (2003) Pareto dominance approach is presented to compare the solutions, when a solution s_1 strictly dominates a solution s_2 (i.e., $s_1 \succ\succ s_2$) then s_1 is better than s_2 in all objectives. A solution s_1 dominates a solution s_2 (i.e., $s_1 \succ s_2$) when the solution s_1 is not worse

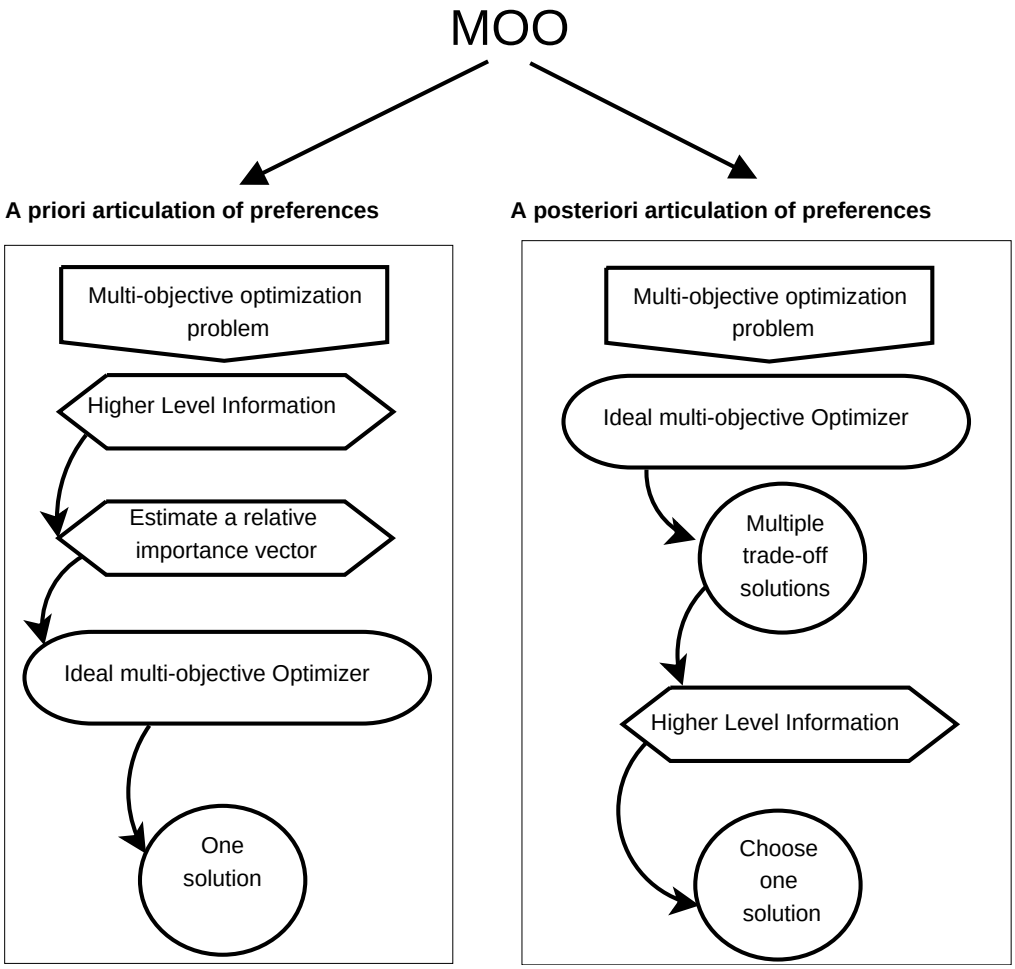


FIGURE 2.9: Multi-Objective Optimization categories

in all objectives and better in at least one objective. We say that s_1 weakly dominates s_2 (i.e., $s_1 \succeq s_2$) if s_1 is not worse than s_2 in all objectives. More often, we find some solutions that don't dominate other, and neither are dominated, in this case, we say that they are incomparable (i.e., $s_1||s_2$).

2.3.4 Fuzzy Logic

The conventional procedure executed by the computer is a block of instructions that takes data and provides an output in the form of true or false. That is why the author of Zadeh (1996) proved that ambiguous data could be represented. He invented a way to imitate human reasoning by taking into account other possibilities than yes or no and helped to deal with uncertainty in many problems. The inventor of fuzzy logic explained that the human decision includes a range of options between true and false that could be represented by linguistic labels like very true, quite true, not very true and not very false, and so on. This method of reasoning could be implemented in systems in various domains such

as control systems in hardware or software. The principal application fields of fuzzy logic are the automotive systems (e.g., Automatic Gearboxes, Four-Wheel Steering, etc.,) and the electronic and domestic systems (e.g., Television, Washing Machines, etc.,). Environment control also benefits from the fuzzy logic approach such as vehicle environment control or air conditioning control (Sousa et al., 1997; Naranjo et al., 2008). Some industrial applications in the domain of fuzzy logic control which have been designed during the past years are summarized in De Silva (2018). A fuzzy logic controller architecture is mainly composed of four modules. The first part of the controller is Fuzzification module which takes as input a crisp data then changes it in fuzzy sets. This step consists in defining linguistic terms which are input and output variables in the form of simple words or sentences (e.g., Large Positive, Medium Positive, Small, Medium Negative, Large Negative), then a degree of membership between 0 and 1 using the linguistic terms is expressed by membership functions. As an example of fuzzy numbers, we cite the triangular fuzzy number (TFN) that is characterized by the crisp numbers a, b, and c and the membership function μ defined as follows (Zadeh, 1996):

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & c \leq x \end{cases} \tag{2.4}$$

The second module of fuzzy logic controllers is the Knowledge Base Rules which stores If-Then instructions according to the problem constraints. The third module is the Inference Engine which receives fuzzy input sets, makes fuzzy inference using the knowledge base rules, and provides a fuzzy output sets. Finally, Defuzzification module transforms the fuzzy output set into a crisp value.

2.3.5 Search and Optimization Solvers

In this section, strong emphasis is placed on suited solvers for academic research use. CP gives strong support for decision-making in many real-world problems; this paradigm can search quickly through an extensive range of choices. The progress in CP paradigm has been executed in a variety of software toolkits, modeling languages with underlying solvers, and libraries. For example, rich modeling functionalities for formulating combinatorial optimization problems is proposed in both OPL and Minizinc. Common programming capabilities are provided by some systems such as Prolog III, Comet, and IBM ILOG CP4. On another hand, some systems also exist for combinatorial optimization using Integer

programming, for example, we can cite IBM ILOG CPLEX5 and SCIP. Solvers of combinatorial optimization problems already have an extensive commercial application, and the importance now is to examine and exploit the technology thoroughly. Indeed, a primary concern at present is to find the ease-of-use of the technology.

Concerning CP solvers, the best known and easy to use are Choco and Mistral which prove to be highly efficient by winning in several competitions (van Dongen et al., 2008). Choco is a java library for constraint for CP. It is created on an event-based propagation mechanism with backtrackable structures. Its main advantage is that it is an open-source solver just like Mistral. Mistral is an open source constraint library written in C++. Table 2.3 gives some benchmark results of the two solvers. The best values of the percentage of the solved instances and the CPU time are shown in bold. It is remarkable that Mistral solver gives the best values in most cases.

TABLE 2.3: Benchmark results of Mistral and Choco solvers.

Category (#instances)	Choco (Solved / CPU time)	Mistral (Solved / CPU time)
Binary Extensional (622)	89% / 95.78 s	89% / **70.21 s**
Binary Intentional (634)	82% / **55.89 s**	82% / 58.23 s
Global (501)	69% / 69.69 s	**80%** / **56.59 s**
N-ary Extensional (607)	73% / 189.10 s	**94%** / **83.67 s**
N-ary Intentional (660)	78% / 49.70 s	**80%** / **32.02 s**

Source:Hebrard (2008)

Regarding LP solvers, the most well-known and popular software toolkits are CPLEX and Gurobi, although these solvers are commercial. However, its use is free and without restrictions for research purpose. CPLEX and Gurobi prove its effectiveness and robustness compared with others free solvers. In Figure 2.10, commercial solvers are represented with a green color while free solvers are designated with blue. The running times' results show that Gurobi is the fastest. Furthermore, Table 2.4 shows the effectiveness of Gurobi in solving the highest number of instances compared to the others solvers. Commercial solvers are more effective than free ones; This may explain why the prices are high. Therefore, it is worthing to benefit from an academic license for CPLEX and Gurobi since XPRESS does not offer one. The studies which have been conducted shows that Gurobi and CPLEX exhibit approximately the same performance such in Lodi and Tramontani (2013). Nevertheless, Gurobi is easier to get for academic use

than CPLEX even if CPLEX has been older than Gurobi. The process of application to get the license for CPLEX is very long and takes some weeks. Furthermore, it is not very simple to download even after getting the license. Still, Gurobi is free to use in academia if you are connected from an educational domain name, and the download is considerably simple. Table 2.5 presents some solvers used for academic use and their brief description.

Finally, it is worth stressing out that some packages exist for benefiting from both CP and LP, and help to find the adequate model for a problem. Numberjack is a modeling package written in Python for embedding constraint programming and combinatorial optimization into great applications. It has been produced to seamlessly and efficiently hold many underlying combinatorial solvers (Hebrard et al., 2010). It provides a standard API for CP, MIP and SAT solvers. The supported solvers are the CP solvers Mistral and Gecode; a native Python CP solver; the MIP solver SCIP and Gurobi; and the MiniSat satisfiability solver. The main advantage is that the users of Numberjack can write their problems once and then specify which solver should be used.

TABLE 2.4: Benchmark results of commercial vs. free solvers.

	running time	instances solved	solved (%)
CBC	10.20	41	47.13
CPLEX	1.45	73	83.91
GLPK	22.11	3	3.45
GUROBI	1.00	77	88.51
LP SOLVE	19.40	5	5.75
SCIP-C	3.76	63	72.41
SCIP-L	6.40	52	59.77
SCIP-S	5.33	57	65.52
XPRESS	1.29	74	85.06

Source:Meindl and Templ (2012)

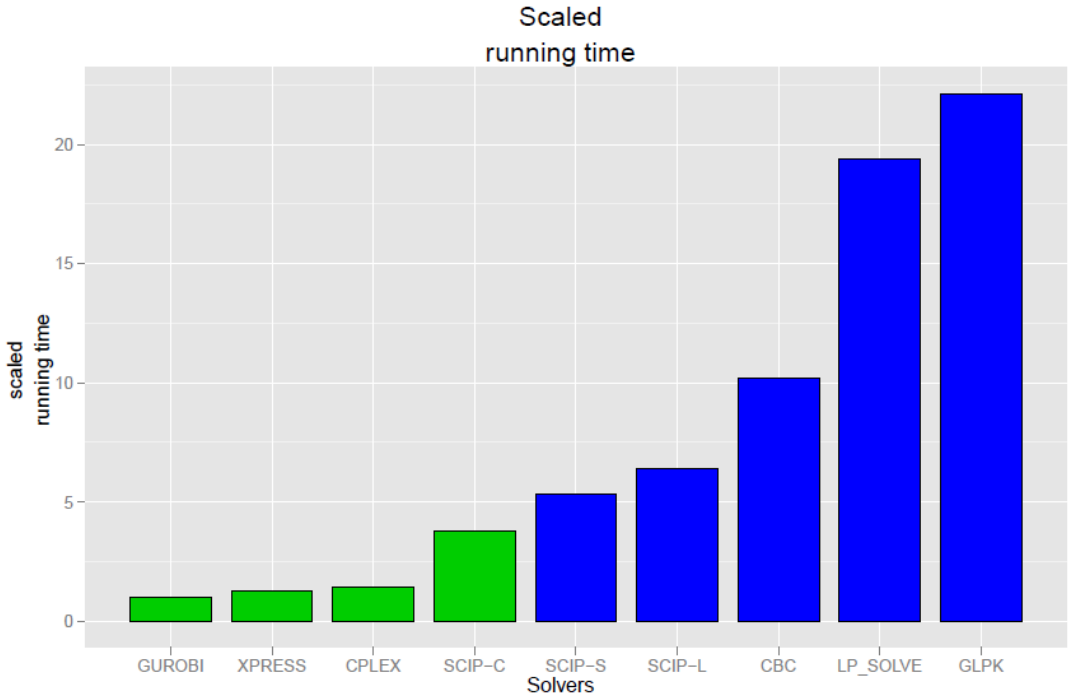


FIGURE 2.10: Running times of commercial vs. free solvers.

Source:Meindl and Templ (2012)

TABLE 2.5: Free search and optimization solvers for academic use.

Solvers	**Brief description**
Maple	a symbolic and numeric computing environment for linear, quadratic, nonlinear, continuous, and integer optimization. Also for Constrained and unconstrained optimization.
MATLAB	A general-purpose and matrix-oriented programming-language for numerical computing. Linear programming in MATLAB needs the Optimization Toolbox.
OPL	an algebraic modeling language for mathematical optimization models. It is part of the CPLEX software package and consequently tailored for the IBM ILOG CPLEX and IBM ILOG CPLEX CP Optimizers.
MiniZinc	a constraint modeling language for constraint satisfaction and optimization problems. It is a high-level independent solver taking advantage of a large library of pre-defined constraints.
Gurobi	a solver with parallel algorithms for large-scale linear programs, quadratic programs, and mixed-integer programs.
SCIP	a general-purpose constraint integer programming solver with an emphasis on Mixed Integer Programming.
CPLEX	popular solver with an API for several programming languages, and also has a modeling language and works with AIMMS, AMPL, GAMS, MPL, OpenOpt, OPL. The IBM ILOG CPLEX Optimizer solves integer programming problems, very large linear programming problems using either primal or dual variants of the simplex method or the barrier interior point method, convex and non-convex quadratic programming problems, and convex quadratically constrained problems.
Choco	a problem modeler and a constraint programming solver available as a Java library.
Mistral	a Constraint Satisfaction Library written in C++.
MiniSat	a minimalistic, open-source SAT solver, developed to help researchers and developers to solve SAT problems. It is released under the MIT license.

Chapter 3

A Virtual Network Functions Placement System using Constraint Programming

3.1 Contribution 1: Modeling and Optimization of the Network Functions Placement using Constraint Programming

3.1.1 Introduction

In 5G architecture, the on-demand edification of the mobile network on the cloud in an elastic way provides a flexible network thanks to the Network Functions Virtualization (NFV). It allows using cloud computing infrastructure, instead of having custom hardware appliances for each network function and setting up a mobile, flexible, dynamic, and rapidly deployable network. These features provide to the network the programmability and flexibility that makes it able to adapt to changing user's mobility and demand dynamically.

Among these NFVs are the Serving Gateway (S-GW) and Packet Data Network Gateway (PDN-GW or P-GW), which must be positioned effectively to respond to changes and behaviors of the user, on the one hand, and services, on the other side.

Our contribution is to propose a solution to define S-GWs and P-GWs positions dynamically in given DCs, based on Constraint Satisfaction Problem (CSP) and an improved version of Forward-Checking, to ensure the automatic adaptation to the requirements of users and services.

3.1.2 Problem formulation and constraint satisfaction problem

Problem formulation

We have a number of DCs, a number of users with their locations, a number of APN (each APN has access to one or more applications / services) and a number of users interested in the traffic of an application k. The problem is based on several constraints:

1. S-GW:
    - A user cannot have more than one S-GW simultaneously
    - It is necessary to minimize the number of relocations during the move of a user from one zone to another.
    - A load of traffic in a traffic area for a period of time must not exceed SGW_{max}.
    - We must minimize the number of virtual instances of SGWs.
2. P-GWS:
    - The amount of traffic for a type of application k must not exceed PGW_{maxk}.
    - All P-GWs for a type of traffic k must have a balance in their load.
    - The path between the P-GW and the user should be short.
    - We must minimize the number of virtual instances of PGWs.

Constraint satisfaction problem formulation

The solution and optimization network functions placement permit to predict the number and positions of S-GW and P-GW that would be needed given an initial placement of DCs, users' behavior and used applications. This can contribute to minimizing the load on DCs and the number of instances of S-GWs and P-GWs since the combinatorial optimization problems make it possible to select the best combination among all those possible. Hereafter is our CSP model. Further explanations will be given below.

1. Variables: $X = \{X_1, ..., X_n\}$
   $X_1 = DCs$
   $X_2 = UES$
   $X_3 = UEP$
   $X_4 = App$
   $X_5 = Use$

$X_6 = LoadS$
$X_7 = LoadP$
$X_7 = APNP$

2. Domains: $D_X = \{D_{X_1}, ..., D_{X_n}\}$
$D_{X_1} = D_{DCs} = (0,0), ..., (S,P)$
$D_{X_2} = D_{UES} = (id_{S_0}, X_{S_0}, Y_{S_0}), ..., (id_{Sm}, X_{S_m}, Y_{S_m})$
$D_{X_3} = D_{UEP} = (id_{P_0}, X_{P_0}, Y_{P_0}), ..., (id_{P_m}, X_{P_m}, Y_{P_m})$
$D_{X_4} = D_{App} = (Load_l, APN_1), ..., (Load_k, APN_k)$
$D_{X_5} = D_{Use} = (u_0, v_0), ..., (u_m, v_k)$
$D_{X_6} = D_{LoadS} = 0, ..., LoadS_{max}$
$D_{X_7} = D_{LoadP} = 0, ..., LoadP_{max}$
$D_{X_8} = D_{APNP} = 0, ..., APN_{Total}$

3. Constraints:
U :
$\forall\ UE_P\ ue_i \forall\ App\ app_j \forall\ DC\ dc_k, ue_i\ is_{near}\ dc_j\ \wedge\ \exists\ (ue_i, app_j) \in$
$D_{Use} \wedge Load_{app_j} + LoadP_k \leq LoadP_{max} \wedge Apn_{app_j} = Apn_{p_k}$
U' :
$\forall\ UE_P\ ue_i \forall\ App\ app_j \forall\ DC\ dc_k\ ue_i\ is_{near}\ dc_j \wedge\ \exists\ (ue_i, app_j) \in$
$D_{Use} \wedge Load_{app_j} + LoadP_k > LoadP_{max} \wedge Apn_{app_j} = Apn_{p_k}$
V :
$\forall\ UE_S\ ue_i \forall\ App\ app_j \forall\ DC\ dc_k\ ue_i\ is_{near-less-relocat}\ dc_j \wedge\ \exists\ use_m \in D_{Use} \wedge$
$Load_{Use_m} + LoadS_k \leq LoadS_{max}$
V' :
$\forall\ UE_S\ ue_i \forall\ App\ app_j \forall\ DC\ dc_k\ ue_i\ is_{near-less-relocat}\ dc_j \wedge\ \exists\ use_m \in D_{Use} \wedge$
$Load_{Use_m} + LoadS_k > LoadS_{max}$

- Constraint 1: $\neg U \vee LoadP_k = LoadP_k + Load_{app_j}$
- Constraint 2: $\neg U' \vee dc_k.P = dc_k.P + 1 \wedge LoadP_{k+1} = Load_{app_j} \wedge Apn_{p_k} = Apn_{app_j}$
- Constraint 3: $\neg V \vee LoadS_k = LoadS_k + Load_{Use_m}$
- Constraint 4: $\neg V' \vee dc_k.S = dc_k.S + 1 \wedge LoadS_{k+1} = Load_{Use_m}$

4. Objectives:

- min Relocations(UE_S)

- min Card ($dc_k.P$)
- min Cost

The variables and domains:

X_1 concerns the DCs, as inputs, we have the exact location of each DC. The values that could take each DC is the number of S-GWs and P-GWs, it varies from 0 to the maximum allowed number of instances of those gateways.

X_2 concerns the end users or equipment that need to pass the SGWs, and each user has a unique identifier and location, we can find several instances of that same user; that occurs when relocation may be needed.

X_3 concerns the end users or equipment that need to pass the PGWs; each user has a unique identifier and location.

X_4 involves the application (or services) that are available; each one has a traffic need values. We can know what application the user (or equipment) is using thanks to the variable X_5 that is a set of couples of users (or equipment) u and applications (or services) v.

X_6 is the load of each S-GW, X_7 is the load of each P-GW, and X_8 is the APN of each P-GW.

The constraints:

Constraint 1: When a user needs to use an application, with a specific traffic amount and a given APN and needs access to a P-GW, the best DC is located, if there is an available P-GW with a load and APN that could handle the request, then this P-GW is used. Note that if the objective is to reduce the cost, then the best DC selected is the one with the less cost whatever P-GWs it has. But, if the aim is to reduce the number of P-GWs, then the nearest DC with a P-GW that has an available load and APN is the best choice in this case. This same logic goes for constraint 2 as well.

Constraint 2: When a user needs to use an application, with a particular traffic amount and a given APN and requires access to a P-GW, the best DC is located, if there is no available P-GW with a load that could handle the request, then a new P-GW instance is created in that DC.

Constraint 3: When a user needs access to an S-GW, not only the nearest DC should be located, but based on its previous locations (variable X2), the best found DC is chosen, if there is an available S-GW with a load that could handle the request, then this S-GW is used. If the objective is to reduce the number of relocation, then the best DC is one of which the range covers the maximum number of mobile UE. If the objective is to reduce the cost, then the best DC is the one with less cost even if relocation would be needed. The same logic goes for constraint 4 as well.

Constraint 4: When a user needs access to an S-GW, not only the nearest DC should be located, but based on its previous locations (variable X2), the best located DC is chosen, if there is no available S-GW with a load that could handle the request, then a new S-GW instance is created in that DC.

The objectives:

To minimize only the costs, the solver used is the FCRC (Extended Forward checking to reduce costs).

To minimize only the number of P-GWs based on the knowledge of APN the solver used is the FCRNP (Extended Forward checking to reduce the number of P-GWs).

To minimize only the number of relocations the solver used is FCRR (Extended Forward checking to reduce relocations).

To optimize all the parameters, we use a smart solver we call it FCSMART, which aims to optimize the following function:

$$\textbf{min } F = \alpha\, Cost_{SGW} + (1-\alpha)\, relocations + \beta\, Cost_{PGW} + (1-\beta)\, number_{VNF_{PGW}} \tag{3.1}$$

with $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$.

Each solver will be detailed in the following.

The forward checking

To solve our problem using the CSP model, we used the forward checking algorithm, which consists of constructing a solution, by considering assignments to variables in a particular order; an order where the constraints are satisfied. The vector in our case is a set of assignments S-GWs and P-GWs to the available DCs. A solution vector is a set where the S-GWs and P-GWs choice satisfied the users' requests and reduce the maximum possible the relocations.

The forward checking is an interesting algorithm for solving constraint satisfaction problems; it's a successful alternative to backtracking. It is important to say that all the CSP algorithms that are related to NP-complete problems examine partial solutions, which are assignments to a subset of the variables, and try to extend these until all variables are assigned. As in our case, those kinds of problems can contain sub-classes of simpler problems. For example, in our case, the whole problem can consist of the fact of finding for one DC a number of S-GWs and P-GWs that satisfy, on one side, the mobility and requests of users and, on the other hand, the load and the maximum number of instances allowed.

The forward checking has shown itself to be a practical improvement over backtracking: it can solve problems that defeat backtracking. The forward checking examines partial solutions, which are assignments to a subset of the variables, and try to extend those partial solutions until all variables are assigned. In our case, the algorithm examines assignment of S-GWs and P-GWs, until it satisfies the constraints (which in the case it is considered as a solution) or not (the solver will then go to other S-GWs and P-GWs choices).

We improved the classic version of the forward checking by adding a best DC selection policy depending on the objectives given. This allows selecting the best DC neighboring a UE in need of S-GW or P-GW services. This selection can be based on the cost of the DC, the APN used by the UE, the DC that covers more mobiles users than the other DCs of the neighborhood (this reduces the number of relocations) or a compromise of all those parameters.

As stated before, the best DC selection ($Look - for - best(dc_k)$) depends on the main objectives:

- FCRC: selects the DC with the lowest cost.
  $Min\ Cost_{SGW} + Cost_{PGW}$
- FCRNP: selects the DC that has an available $P - GW_k$, this means that this P-GW can handle the needed traffic load and with the compliant APN_k.
  $Min\ number_{VNF_{PGW}}$
- FCRR: To reduce relocations, the best DC is the one that covers the maximum number of mobile UEs. (Mobile means UEs with frequent mobility in different locations).
  Min relocations
- FCSMART: Choose the DCs that offer the best compromise of costs, relocations, and number of PGWs.
  Min F

The pseudo code for the extended forward checking is given in Algorithm 1.

3.1.3 Implementation and Results

We evaluate the performance of our problem modeling and solving using FCRC, FCRNP, FCRR, and FCSMART. For this matter, we include three metrics:

- The number of the relocation of the S-GW (case of highly mobile users)
- The cost of path packet delivery time to the S-GW and P-GW

Algorithm 1 Extended Forward Checking (i)

procedure EXTENTED-FORWARD-CHECKING
  for $UEP_i \in UEP$ **do**
    $s_{Pi} \leftarrow v_{Pi}$
    $Look - for - best(DC_k)$
    if $Load(DC_k.P) + Load(App_{UE_{Pi}}) > Load_{MaxP} and APN(DC_k.P) = APN(App_{UE_{pi}})$ **then**
      create-new-P
    else
      $update - DC_k.P$
    if $i = P$ **then**
      $print\ sp_1, ..., sp_P$
    else if $Check - Forward(i)$ **then**
      $Forward - Checking(i+1)$
      $Restore(i)$
  for $UES_i \in UES$ **do**
    $s_{Si} \leftarrow v_{Si}$
    $Look - for - best(DC_k)$
    if $Load(DC_k.S) + Load(App_{UE_{Si}}) > Load_{MaxS}$ **then**
      create-new-S
    else
      $update - DC_k.S$
    if $i = S$ **then**
      $print\ sS_1, ..., ss_S$
    else if $Check - Forward(i)$ **then**
      $Forward - Checking(i+1)$
      $Restore(i)$

- The number of instance of P-GW and S-GW.

We consider a scenario where user mobility is high to test our methods. We set the number of DCs to 10, the number of UEs to 30 and the number of APN to 4. We set a maximum load of virtual machines hosting the VNFs S-GW and P-GW. The UEs and DCs are geographically distributed as shown in Figure 3.1. As shown therein, there are three areas according to the distribution of UEs, and in our scenario, there's high mobility of UEs between these three areas. It is assumed that a user can only be in two areas among the three at different times. The UEs are represented by white dots, and DCs containing at least one VNF are green, DCs without any network function deployed are red. We also find the number of S-GW and P-GW in each operational DC (green).

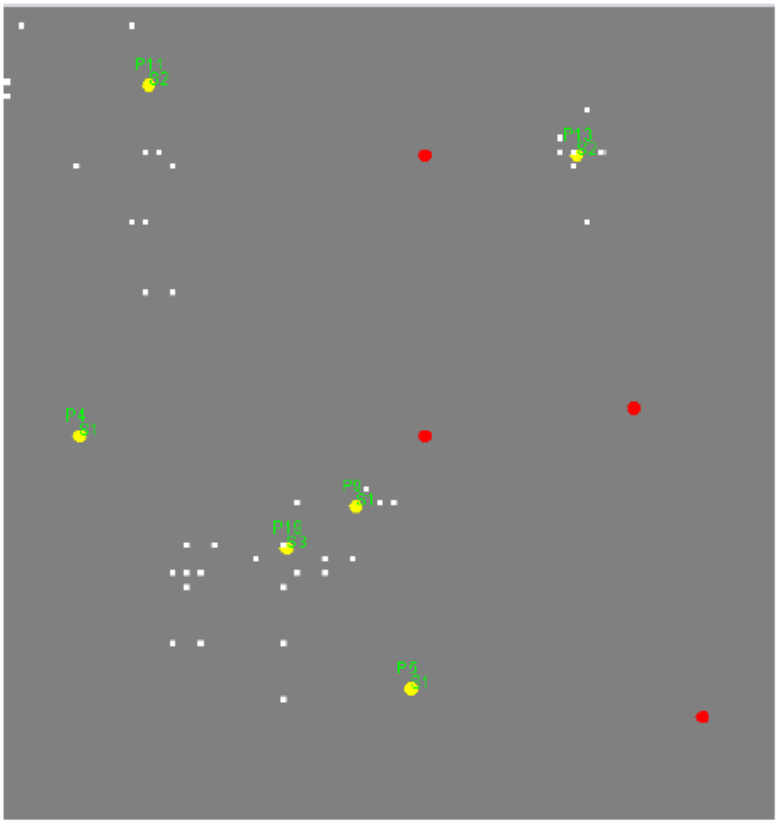


FIGURE 3.1: Graphical User Interface representing DCs, users and the deployed network functions.

S-GW relocation

After running the solvers on our scenario, we noticed that the number of S-GW relocation has experienced a remarkable decline with FCRR (See Figure 3.2), however, the values of CostSGW are the highest (See Figure 3.3), while the FCRC

offers the best CostSGW values (See Figure 3.3), but a very high number of relocations (See Figure 3.2). The best results are those of FCSMART offering low CostSGW values (See Figure 3.4) and a reasonable number of relocations (See Figure 3.2).

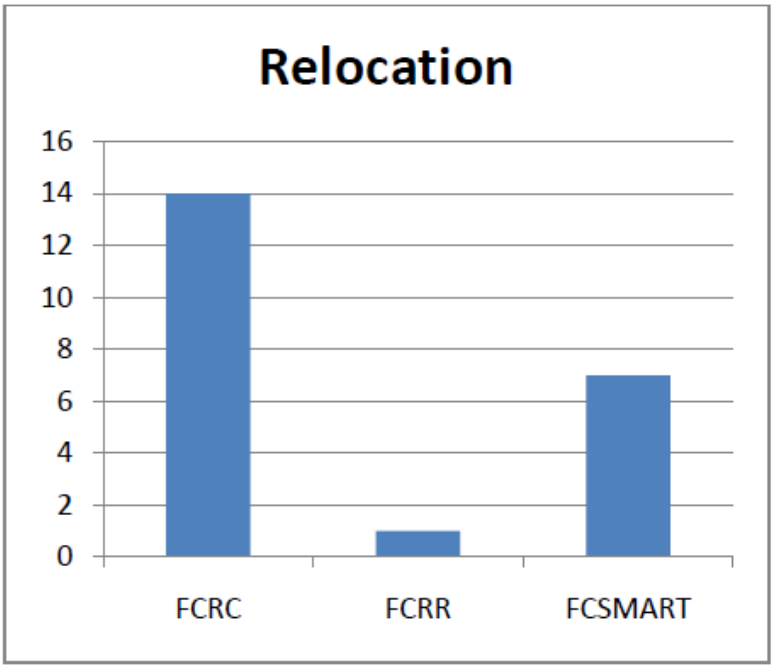


FIGURE 3.2: S-GW Relocation.

S-GW and P-GW costs

As stated before, since FCRR favors the DCs of which the range covers more mobile UEs than the other neighboring DCs, it has the highest cost values (See Figure 3.3). This also goes for FCRNP which gives higher packet delivery delays between UEs and P-GW, than FCRC (See Figure 3.3). However, the number of VM hosting P-GW is the lowest using FCRNP (See Figure 3.5). Here again, FCSMART offers the best balance of P-GW costs (See Figure 3.3) and number of P-GWs (See Figure 3.5). The gap of the number of P-GWs becomes more critical when the number of APN increases.

The number of virtual machines

The number of virtual machines running the S-GW and P-GW network functions, in each DC, is shown in Figures 3.5 to 3.7. The number of virtual machines running S-GW (See Figure 3.5), decreases in most DCs, without affecting the total

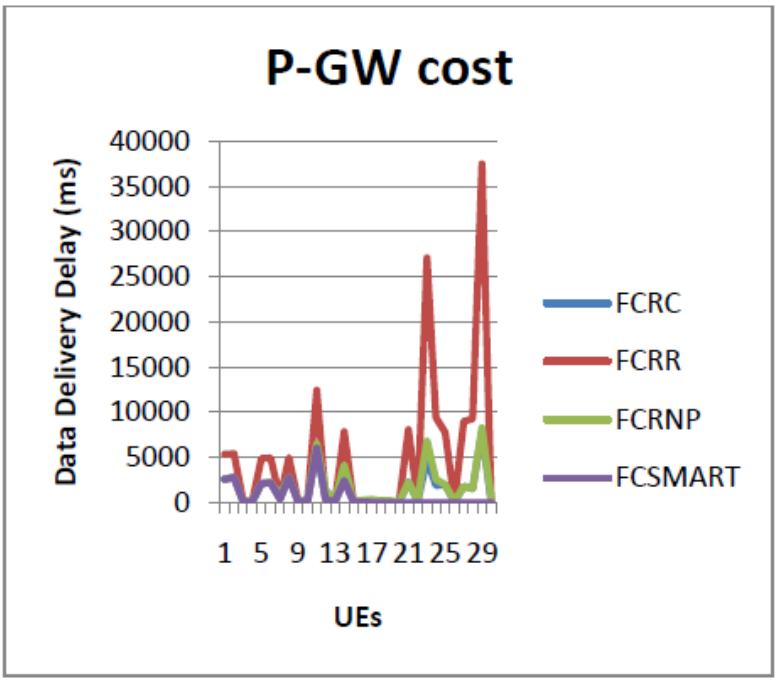


FIGURE 3.3: The cost of the path between users and P-GWs.

number of virtual machines as shown in Figure 3.6. This is explained by the fact that FCRR, in order to reduce relocations, choose more distant DCs and which covers a large number of UEs, resulting in the minimization of instances of SGWs. For the number of virtual machines running PGW, it decreases in some DCs for FCRNP, whence comes the importance of its use, which is to select already deployed PGWs with the needed APN instead of creating new ones in the nearest DCs (as FCRC does), we can notice that in Figure 3.6, where the number of P-GW virtual machines increases by almost 10 machines. In Figure 3.7 it is clear that FCSMART offers the best compromise between the cost and the number of instances.

3.1.4 Conclusion and limitations

In this work, we applied a CSP with different goals for the virtual machine placement problem running S-GW and P-GW, and this within the constraints required by the standards. The methods gave satisfactory results regarding their objectives, by reducing the number of the relocation of the S-GW, the number of virtual machines corresponding to P-GW, and also by optimizing the cost to access S-GW and P-GW, in order, to improve the quality of experience. The choice of method depends on the nature of user behavior; we must reduce the number of relocation facing many mobile users and improve the cost of the path otherwise.

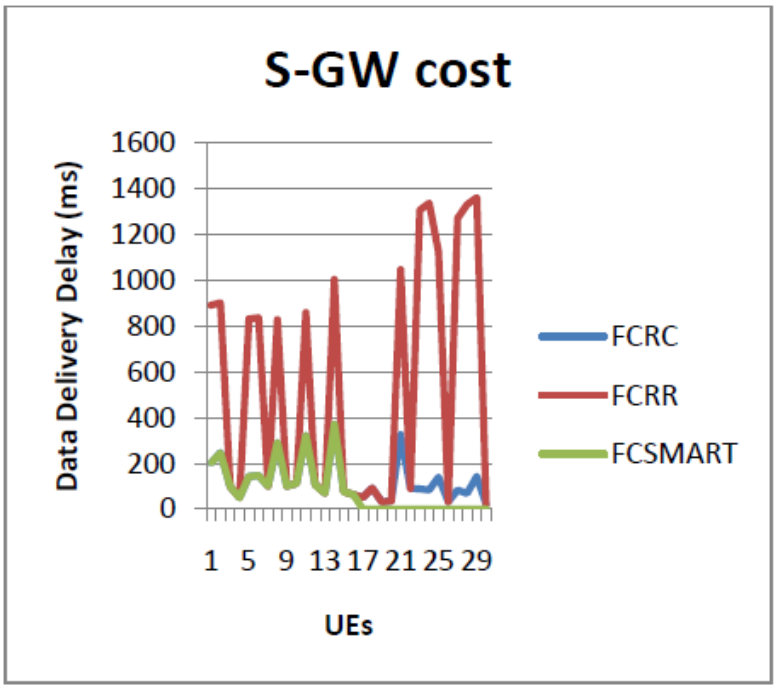


FIGURE 3.4: The cost path between users and S-GWs.

Indeed, user behavior plays a crucial role in optimizing the network configuration and the positioning of these network functions. Still that, a compromise of all those parameters is the best approach to find suitable DCs configurations. FCSMART offers the best compromise between the cost and the number of instances; however, it depends on two parameters namely α and β. On another side, the modelization of the problem proved to be complicated to implement. To avoid these drawbacks, in the next work, we propose a virtual network functions placement system.

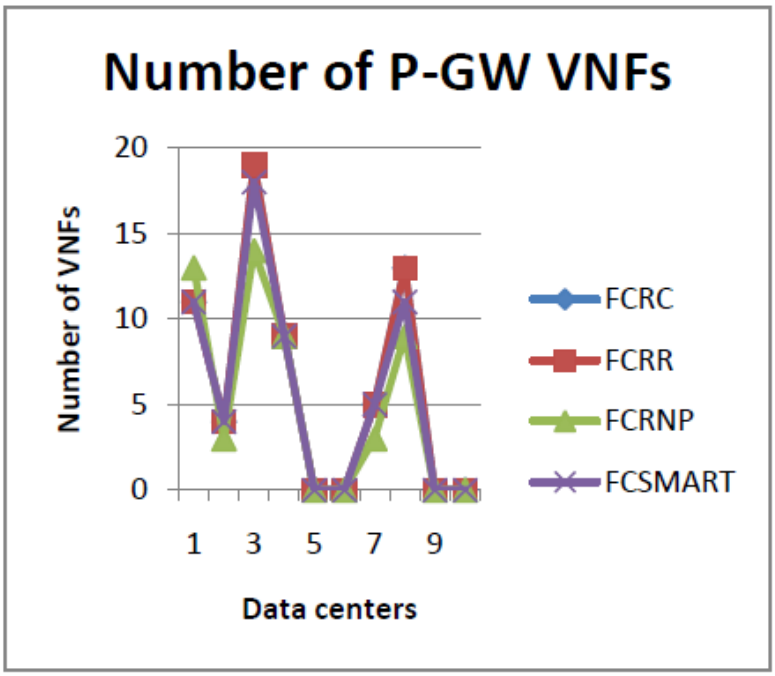


FIGURE 3.5: The number of VMs running the P-GWs.

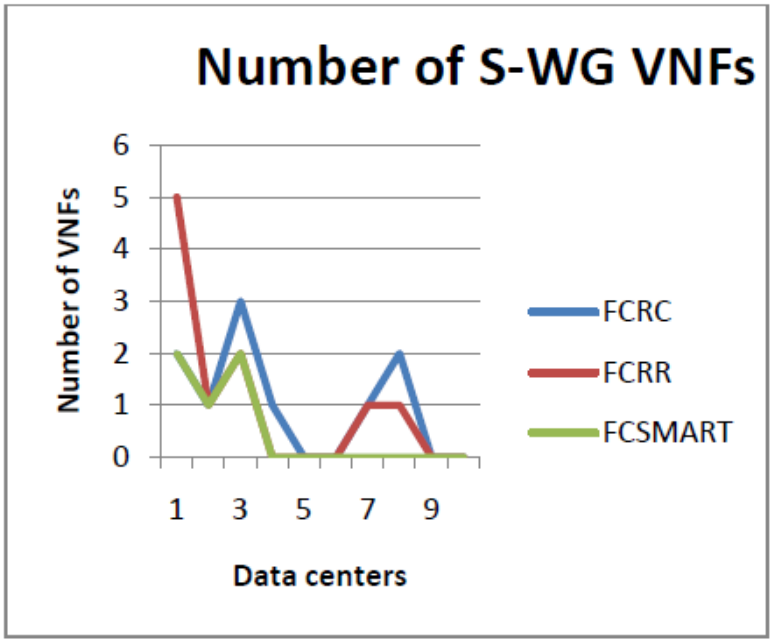


FIGURE 3.6: The number of VMs running the S-GWs.

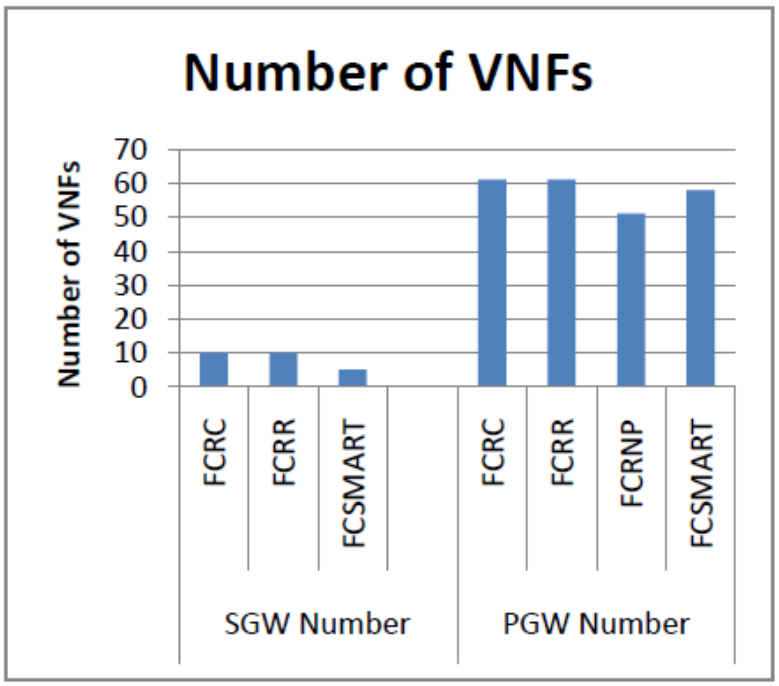


FIGURE 3.7: The number of VMs.

3.2 Contribution 2: Virtual Network Functions Placement System for 5G Mobile Network Architecture

3.2.1 Introduction

The mobile telecommunications market is encountering new trends taking advantage of network virtualization and Cloud Computing techniques. This work advances one of the most crucial challenges which is the placement of virtual network functions over the Cloud. In this vein, we propose a virtual network functions placement system which is designed to have the maximum level of flexibility for meeting the operators' preferences and adjusting to the users' behavior. The system determines a fair solution respecting the constraints conforming with the 3GPP standards which are minimizing Serving Gateways relocations cost and the cost of the path connecting Packet Data Network Gateways and eNodeB base stations. Moreover, the system aims at reducing the incurred cost of virtual machines. The proposed approach to achieve the system solver is Constraint Programming and is compared to Boolean Satisfiability, and Game Theory approaches.

In this work, we propose a real-time VNFs placement system that can play the role of the resource controller as described in Section 2.1 and that takes more than two objectives. In the remaining of this paper, only S-GWs and P-GWs network functions are considered because of their importance and their high requirements. However, the same logic applies to other entities. The main contribution of this paper is proposing a system that contains a solver which is mainly in charge of VNFs placement. The solver relies on Constraint Programming as suggested in the previous section. This real-time solution takes into account all 3GPP standards constraints and more than two objectives, unlike the game theory approach. Moreover, the solution is adjusting to the operators' preferences and the users' behavior.

3.2.2 Virtual Network Functions Placement System

3.2.3 System architecture

The architecture of our system is shown in Figure 3.8 The system is composed of a Virtual Machines Placement Agent (VMPA), a Placement Management Agent

(PMA) and an Environment Perception Agent (EPA). The VMPA is the main component of the system, which is responsible for P-GW and S-GW virtual machines placement. Hence, the mapping of eNBs to data centers hosting virtual network functions is carried out using some policies. This agent aims to find an efficient solution for the placement of S-GWs and P-GWs in an autonomously way or with administrator approval. Indeed, the virtual machines placement agent provides a programmatic interface (API) which allows the control of the mapping (i.e., each eNB to a dedicated DC). The autonomous placement is performed by taking into account some performance data which are the maximum permitted cost of the path between eNBs and data centers and the highest number of virtual machines. The administrator sets these data and sent by the PMA to the VMPA. Therefore, there is an API provided by the placement management agent. The system can perceive the environment and be aware of the network infrastructure using the environment perception agent. The latter sends to the VMPA all available and required information for the mapping regarding DCs, eNBs, and the average frequency of handovers between eNBs. The administrator can also set the maximum permitted amount of handovers through the programmatic interface of the placement management interface. All the agents mentioned above cooperate to find a mapping autonomously based on one of the three policies according to the state of the environment and the data defined by the administrator. More details are provided in the next section.

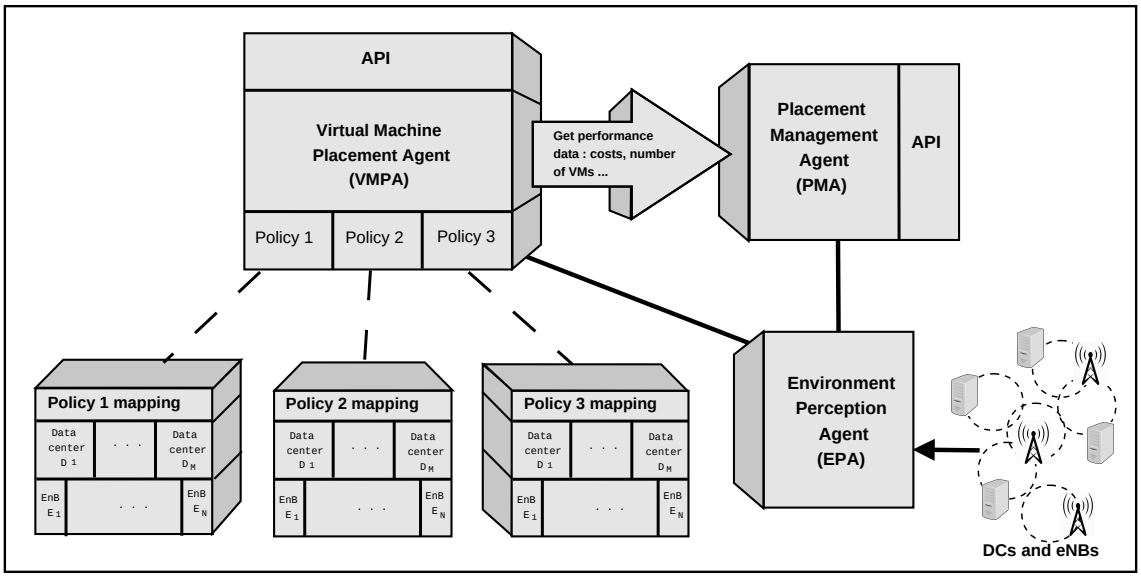


FIGURE 3.8: System architecture.

3.2.4 Virtual machines placement agent

The VMPA agent acts as the brain of the system; it works using three policies helping to find the best mapping of eNBs with data centers. This agent works intelligently by updating its environment state thanks to the environment perception agent. Therefore, the best mapping is chosen according to the mobility of UEs and the administrator's preferences (i.e., the operator's preferences). If the administrator does not intervene by selecting which parameter to reduce (path cost, relocation cost, VMs cost.) through the API, then the agent chooses a policy to adopt. Afterward, the solver finds a mapping taking into account the policy rules. Indeed, a policy is used by the solver so it can choose a mapping which aligns with the user's behavior and the administrator's preferences. Policies consist of rules providing information about how to compare all the objectives. Thus, the solver uses those procedures to find a mapping. We hereafter consider three goals: The first one is the minimization of the average frequency of handovers between eNBs so that we can reduce the relocation cost of S-GWs. The second one is the minimization of the path between eNBs and their relative DCs so we can obtain better QoE. And the last one is the minimization of the number of VMs hosting S-GW and P-GW network functions so we can reduce the incurred cost paid by the operator. Using the rules provided by the policy, the solver attempts to find a mapping of eNBs onto the physical hosts and the placement of VNFs in DCs. As shown in Figure 3.9, *Policy 1* takes as a priority the minimization of relocation cost, then the minimization of the path cost and the minimization of the number of VMs. *Policy 2* begins its priorities with the minimization of the path cost, as second objective the reduction of the number of VMs and then the relocation cost. And finally, the reduction of VMs objective is the priority of *policy 3*, then the relocation cost and the path cost objectives. This figure is a cycle demonstrating the priorities of each policy. The solver uses Constraint Programming which is a useful tool to implement S-GWs and P-GWs network functions placement constraints. More details on the VMPA algorithm are provided in the next section.

3.2.5 Problem Formulation and Solving Strategy

Problem formulation

To develop a problem model using Constraint Programming, the fundamental process consists of first defining the variables and their corresponding domains, i.e., what decisions need to be made and what are the possible outcomes that can be taken for each one. By doing this, separating the formulation and the search

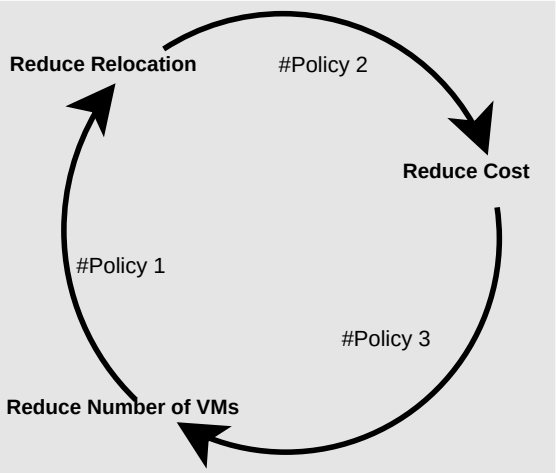


FIGURE 3.9: Policies priority.

strategy is guaranteed. Next, the constraints on the relationships between the variables must be defined. If some criterion is to be optimized, the objective functions need to be specified. Following these steps, we model the VNFs placement problem.

We have a number of DCs, a number of UEs connected to eNBs, a number of APN (each APN has access to one or more applications/services) and a number of users interested in the traffic of an application k. Our contribution is to propose a solution to define S-GWs and P-GWs positions in the given data centers autonomously. The problem is based on several constraints. Furthermore, it is necessary to minimize the number of relocations of S-GW during the move of a user from one zone to another, and the path between the P-GW and the user should be short. On another hand, the incurred cost paid by the operator for running the virtual machines must be minimized.

We represent the set of data centers by $\mathcal{DC}$ and $\mathcal{E}$ is the set of eNBs. The total amount of traffic generated by UEs in eNB j is represented by ω_j, while the amount of traffic of an application k generated by UEs is λ_j^k. We define the average frequency of handovers between eNB j_1 and eNB j_2 as $h(j_1, j_2)$ with $j_1, j_2 \in \mathcal{E}^2$. The cost between a DC $i \in \mathcal{DC}$ and an eNB $j \in \mathcal{E}$ is defined as $c(i, j)$. The relocation of S-GWs is with a cost which we denote by C_{Reloc}. PGW_{MAX}^k and SGW_{MAX} are respectively the maximum capacities of the VMs hosting an S-GW and a P-GW handling a traffic k.

We model the problem of VNFs placement as follows:

- A finite set of variables: $X = \{V, U\}$. Where: $V(\mathcal{E}, \mathcal{E})$ is a binary and symmetric matrix. When eNB j_1 and eNB j_2 are connected to the same data

center, then $V(j_1, j_2) = 1$ and $V(j_1, j_2) = 0$ otherwise. $U(\mathcal{DC}, \mathcal{E})$ is a binary matrix. when eNB j is connected to the data center i then $U(i, j) = 1$, otherwise $U(i, j) = 0$.

- A nonempty domain of possible values for each variable: $D_V = D_U = \{0, 1\}$
- A finite set of constraints:
  1. $\forall i \in \mathcal{DC}, \forall j_1, j_2 \in \mathcal{E}^2$: $V(j_1, j_2) = 0 \implies (U(i, j1) = 0) \vee ((U(i, j2) = 0)$;
  2. $\forall i \in \mathcal{DC}, \forall j_1, j_2 \in \mathcal{E}^2$: $V(j_1, j_2) = 1 \implies U(i, j1) = U(i, j2)$;
  3. $\forall j \in \mathcal{E}$: $\sum_{i \in \mathcal{DC}} U(i, j) = 1$;
- The objectives are:
  1. $\forall i \in \mathcal{DC}, \forall k \in \mathcal{APN}$: $min(\frac{\sum_{j \in \mathcal{E}} U(i,j) \times \omega_j}{S-GW_{MAX}} + \frac{\sum_{j \in \mathcal{E}} U(i,j) \times \lambda_j^k}{P-GW_{MAX}^k})$
  2. $min \sum_{j_1 \in \mathcal{E}} \sum_{j_2 \in \mathcal{E}} h(j_1, j_2) C_{Reloc}(1 - V(j_1, j_2))$
  3. $min \sum_{i \in \mathcal{DC}} \sum_{j \in \mathcal{E}} c(i, j) U(i, j)$

The constraints in the problem model are described as follows:

1. The first constraint ensures that if $V(j_1, j_2) = 0$, j_1 and j_2 must not connect to the same data center.
2. The second constraint ensures that if $V(j_1, j_2) = 1$, j_1 and j_2 must connect to the same data center.
3. The third constraint ensures that each eNB must be connected only to one data center.

The first objective aims at minimizing the number of VMs, the second one aims at reducing the cost of S-GWs relocations, and the last objective aims at reducing the cost of the path between the UE and the data center.

Algorithm description

The system provides a mapping of eNBs to DCs taking into account the state of the environment (e.g., DCs and eNBs locations, the frequency of handovers between eNBs, etc.) and the administrator'preferences. Therefore, an adequate VMs placement is obtained by the system. The administrator can set the maximum permitted cost of relocations, the maximum path cost, and the maximum

number of VMs that are respectively represented by $Reloc_{Max}$, $Cost_{Max}$ and $NVMs_{Max}$. Figure 3.10 represents the chart-flow of the VMPA algorithm. The process starts by receiving data from PMA and EPA using *Get Data function* and finding a primal mapping with a balance between the three objectives. Indeed, these are conflicting objectives and *Get Primal Mapping* function aims at finding a balance and a compromising solution. If the average frequency of handovers (i.e., the relocation cost) stills high than $Reloc_{Max}$ then *Policy 1* is applied and a solution where the total number of handovers do not exceed $Reloc_{Max}$ is obtained. In the case of low handovers, the cost is evaluated, and the minimum is obtained by applying *Policy 2* if it exceeds $Cost_{Max}$. If not, the number of VMs is calculated and *Policy 3* is carried out if the number is upper than $NVMs_{Max}$. The primal mapping is sent if the number of VMs, the path cost, the cost of relocations don't exceed the limit. The definition of the model is shown in Algorithm 2 where all variables, constraints, and objectives are specified. Algorithm 3 describes the *Primal Mapping* function. Variables, constraints, and objectives are added to the model, then the solver is called. *Policy 1* function aims at finding a total cost of relocations less than $Reloc_{Max}$ as described in Algorithm 4. *Policy 2* function finds a total path cost less than $Cost_{Max}$ and this method is presented in Algorithm 5. *Policy 3* function finds a mapping where the number of VMs is less than $NVMs_{Max}$ as described in Algorithm 6.

Solvers

In many real-world applications related to several areas such as scheduling, planning, vehicle routing, network design, and many others, multi-objective optimization has always been mentioned. Problems differ from one domain to another, but intelligent and automated approaches to these problems can provide high-quality solutions from many perspectives such as energy efficiency, cost, time and so on. Indeed, if the model is well defined, it can be passed off directly to a solver which will find a solution. For the VNFs placement problem we propose two solvers extracted each one from a correspondent formalism:

- Constraint Programming (CP): In the Constraint Programming paradigm variables take their values from finite sets of possibilities. A solution to a Constraint Satisfaction Problem (CSP) consists of a mapping from each variable to one of the values in its domain such that all constraints are satisfied. The solutions are found using a combination of systematic backtracking search and Polynomial-time inference algorithms that reduce the size of the search space (Rossi et al., 2006).

Algorithm 2 Initialization function

$U \leftarrow Matrix(\mathcal{DC}, \mathcal{E})$
$V \leftarrow Matrix(\mathcal{E}, \mathcal{E})$
for i $\in \mathcal{E}$ **do**
    $sum \leftarrow 0$
    for t $\in \mathcal{DC}$ **do**
        $sum \leftarrow sum + U[t][i]$
        for j $\in \mathcal{E}$ **do**
            $c1 \leftarrow (V[i][j] = 0) \Rightarrow (U[t][i] = 0 \vee U[t][j] = 0)$
            $c2 \leftarrow (V[i][j] = 1) \Rightarrow (U[t][i] = U[t][j])$
    $c3 \leftarrow sum = 1$
$sum \leftarrow 0$
for i $\in \mathcal{E}$ **do**
    for j $\in \mathcal{E}$ **do**
        $sum \leftarrow sum + h[i][j] \times C_{Reloc} \times (1 - V[i][j])$
$obj1 \leftarrow Minimise(sum)$
$c4 \leftarrow sum < Reloc_{Max}$
$sum \leftarrow 0$
$sgw \leftarrow 0$
$pgw \leftarrow 0$
for i $\in \mathcal{E}$ **do**
    for t $\in \mathcal{DC}$ **do**
        $sum \leftarrow sum + c[t][i] \times U[t][i]$
        $sgw \leftarrow sgw + \omega[j] \times U[t][i]$
        $pgw \leftarrow pgw + \lambda[j] \times U[t][i]$
$obj2 \leftarrow Minimise(sum)$
$c5 \leftarrow sum < Cost_{Max}$
$obj3 \leftarrow Minimise(\frac{sgw}{S-GW_{MAX}} + \frac{pgw}{P-GW_{MAX}})$
$c5 \leftarrow \frac{sgw}{S-GW_{MAX}} + \frac{pgw}{P-GW_{MAX}} < NVMs_{Max}$

Algorithm 3 Primal Mapping Function

procedure PRIMALMAPPING
    $Model.addVariables(U, V)$
    $Model.addConstraints(c1, c2, c3)$
    $Model.addObjectives(Obj1, Obj2, Obj3)$
    $U, V \leftarrow Solve(Model)$
    return U, V

Algorithm 4 Policy1 Function

procedure POLICY1
    $Model.addVariables(U, V)$
    $Model.addConstraints(c1, c2, c3, c4)$
    $Model.addObjectives(Obj2, Obj3)$
    $U, V \leftarrow Solve(Model)$
    return U, V

Algorithm 5 Policy2 Function

```
1: procedure POLICY2
2:     Model.addVariables(U, V)
3:     Model.addConstraints(c1, c2, c3, c5)
4:     Model.addObjectives(Obj1, Obj3)
5:     U, V ← Solve(Model)
6:     return U, V
```

Algorithm 6 Policy3 Function

```
1: procedure POLICY3
2:     Model.addVariables(U, V)
3:     Model.addConstraints(c1, c2, c3, c6)
4:     Model.addObjectives(Obj1, Obj2)
5:     U, V ← Solve(Model)
6:     return U, V
```

- Boolean Satisfiability (SAT): A satisfiability problem is defined regarding Boo-lean variables and a single form of constraint, namely a disjunction of Boolean variables or their negations. The task is to determine whether or not there exists a truth assignment to the variables such that the propositional formula assesses to true, and, if this is the case, to find this assignment. Instances are also solved using backtracking search, using unit-propagation for inference, as well as learning new clauses when failures are encountered (Biere et al., 2009).

In the next section, we discuss the performance of the two approaches regarding the VNFs placement problem and the proposed system. The two solvers are implemented to simulate our proposed solution.

3.2.6 Implementation and Results

In this section, we will evaluate the performance of the proposed model. The afore-described functions (i.e., policies) and the two solvers are analyzed regarding the following metrics:

- The S-GWs relocation cost.
- The cost of the path between eNBs and DCs.
- The number of VMs.
- The operation time.
- The number of nodes.

- The number of backtracks.

Furthermore, we compare in the simulation results the proposed approach namely Constraint Programming with the literature solution described in Ksentini et al. (2016) which is game theory approach. Since this former takes only two objectives, the evaluation is performed regarding the following metrics:

- The S-GWs relocation cost.
- The cost of the path between eNBs and DCs.
- The operation time.
- The memory usage.

To evaluate the solution proposed in this paper, we developed a simulator using Python programming language taking advantage of its capability of supporting multiple systems and platforms drawn up with other languages. The simulator is carried on using the Numberjack library, which allows the user to model and solve combinatorial optimization problems. It provides a standard interface to many underlying C/C++ solvers seamlessly and efficiently and offers a high-level modeling platform. Numberjack package proposes plenty of Constraint Programming (CP) and Satisfiability problem (SAT) solvers. In this simulator, we use Mistral which is an open-source constraint library to evaluate our solution using the CP approach. Minisat is a minimalistic, open-source SAT solver which we use to analyze our model using SAT approach. The game theory approach is implemented using Gurobi Optimizer. In the simulations, the network infrastructure (i.e., eNBs and DCs) is randomly deployed, and the solution is evaluated by varying the mobility of the users. In other words, we vary the average frequency of handovers between eNBs to evaluate the previously mentioned functions (i.e., *Primal Mapping*, *Policy 1*, *Policy 2* and *Policy 3* functions) and the two solvers (i.e., Mistral and Minisat solvers). In the simulation results, each plotted point represents the average of 100 times of executions. The plots are presented with 95 confidence interval. In each execution, C_{Reloc}, PGW^k_{MAX} and SGW_{MAX} remain unchanged, so we vary all the other parameters. The game theory approach is represented in what follows by GTA.

Functions evaluation

Figure 3.11 represents the evaluation of the proposed functions using Mistral solver (i.e., Constraint Programming) and demonstrates the efficiency of each proposed algorithm in achieving its key design goals. Figure 3.11(a) shows the

S-GWs relocation cost of *Primal Mapping*, *Policy 1*, *Policy 2* and *Policy 3* functions. The relocation cost increases while the average frequency of handovers increases concerning all functions except *Policy 1*. We notice that this former function shows the best performance concerning minimizing the relocation cost. However, the other algorithms have almost the same performance. The cost of the path between eNBs and DCs is shown in Figure 3.11(b), *Policy 2* shows best results concerning the cost minimization regardless of the mobility of users. However, *Policy 1* and *Policy 3* exhibit almost similar performance regarding the minimization of the path cost and have the worst performance regarding this objective. Figure 3.11(c) shows the algorithms'evaluation concerning the VMs number. *Policy 3* performs best, and the other functions maintain the same performance.

From the figures, it becomes apparent that regardless the average frequency of handovers the proposed functions successfully achieve their key design goals in placing VNFs at adequate DCs.

Mistral and Minisat solvers evaluation

The two solvers comparison is represented in Figure 3.12. We evaluate the two solvers representing CP and SAT approaches using *Primal mapping* function. From Figure 3.12(a), we notice that the cost of S-GWs relocations increases while varying the average frequency of handovers between eNBs, this is due to the mobility of users that become higher. Moreover, Minisat solver provides solutions with lower relocations cost in comparison with Mistral solver. In Figure 3.12(b), the cost of the path between eNBs and DCs is presented. We observe that Minisat solver can outperform Mistral solver and provides better solutions regarding the cost of S-GWs relocations and the cost of the path. Concerning the third minimization objective, the number of VMs is the same for both Minisat and Mistral solvers as depicted in Figure 3.12(c). Figure 3.12(d) shows the performance of the two solvers regarding the operation time, and we notice that Mistral solver exhibits the best performance. The number of nodes is also high regarding Minisat solver as shown in Figure 3.12(e), this figure demonstrates that a considerable difference is perceived between the two solvers. And this also applies to the number of backtracks. Indeed, Figure 3.12(f) shows that Mistral solver exhibits the best performance regarding the number of backtracks. Minisat solver can find best solutions concerning S-GWs relocations cost and path cost minimization. Consequently, it has a significant impact on the operation time, the number of nodes and the number of backtracks.

Mistral solver and GTA evaluation

Figure 3.13 shows the performance comparison between Mistral solver and GTA. Regarding the minimization of S-GWs relocation cost and the minimization of the path cost between eNBs and DCs, Mistral solver and GTA exhibit the same performance for the two objectives as depicted in Figure 3.13(a) and Figure 3.13(b). CP approach (i.e., Mistral solver) outperforms game theory approach only in the operation time as shown in Figure 3.13(c). Mistral solver realizes an operation time less than 0.09s while GTA operation time is high up to 0.11s. Moreover, Mistral exhibits a better performance concerning memory usage, as depicted in Figure 3.13(d). We notice that the gap is large when the average frequency of handovers is less than 160 p/s while it becomes minimal when the average frequency of handovers increases.

3.2.7 Conclusion and limitations

Building a mobile network on demand and in an elastic manner is possible thanks to cloud providers' offerings and virtualization techniques. However, efficient placement of the virtualized important network functions over the Cloud is of vital interest. In this paper, a real-time virtual network functions placement system is proposed for more relevant network functions, namely S-GWs and P-GWs. This system is designed to have the maximum level of flexibility for meeting the operators' preferences and adjusting to the users' behavior. VMPA which is the main component of the system works according to some policies to obtain an adequate solution. The policies are evaluated in the simulation and demonstrate its efficiency as per the strategy of the solution. CP and Boolean satisfiability approaches are implemented for the proposed solver system using Mistral and Minisat solvers. Boolean satisfiability solver can outperform CP solver in minimizing the objectives, although the CP approach is more efficient in term of operation time, the number of nodes and the number of backtracks. CP outperforms game theory approach in term of operation time and memory usage; although the two approaches provide the same results. The system is autonomous and runs in a cycle, however, to perform a policy, the placement depends on three values that are $Reloc_{Max}$, $CostMax$, and $NVMs_{Max}$. An administrator depending on some statistics sets these three inputs. In the next chapter, we worked on this limitation and proposed an adaptive solution for virtual network functions placement.

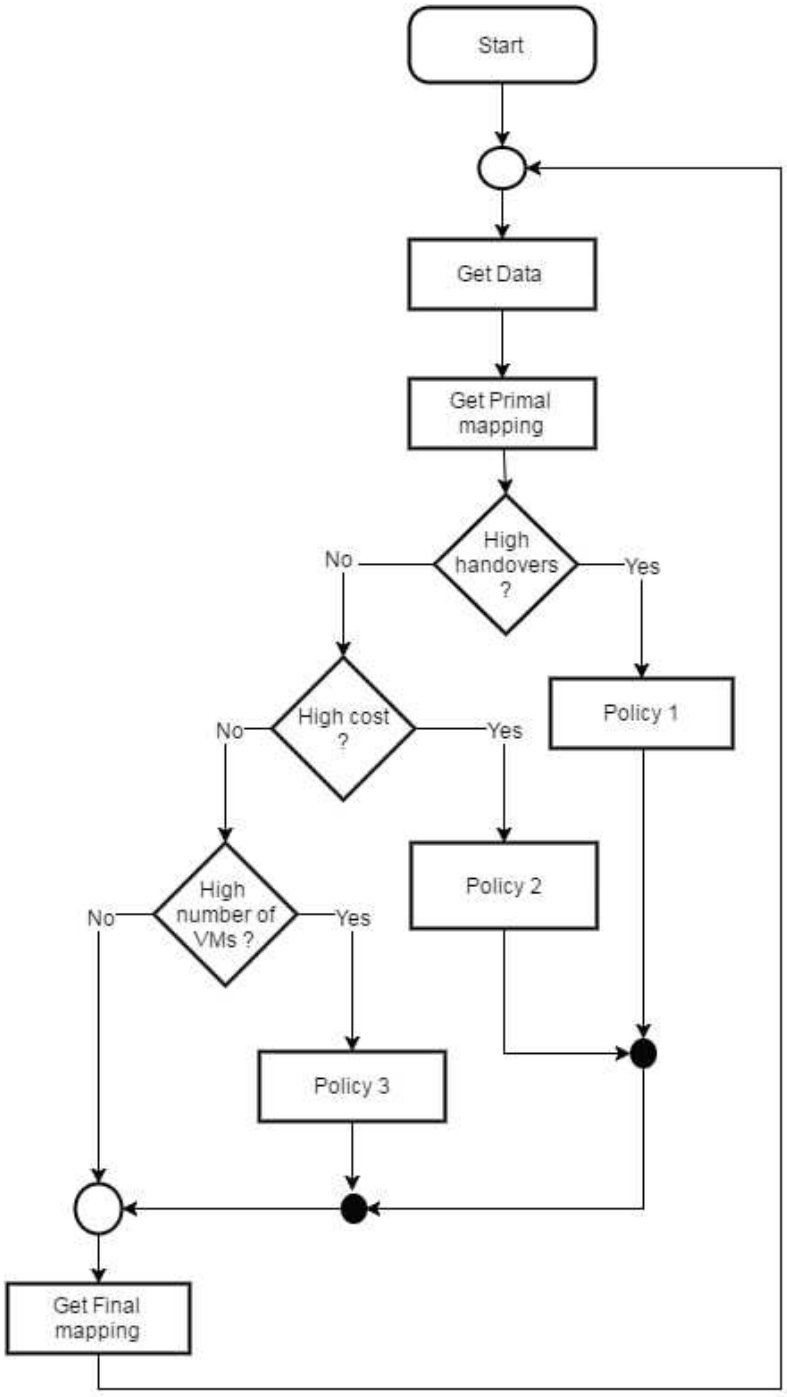
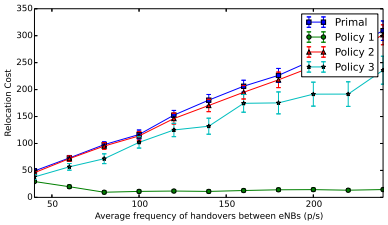
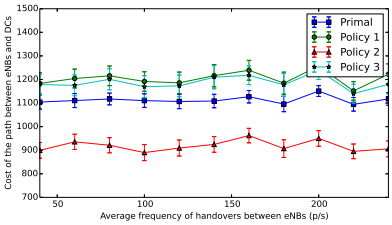


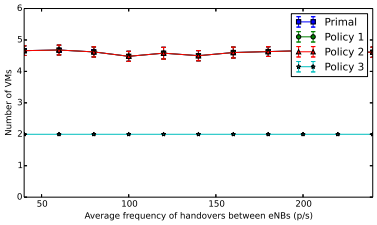
FIGURE 3.10: Chart-flow of VMPA algorithm.



(a) The relocation cost.

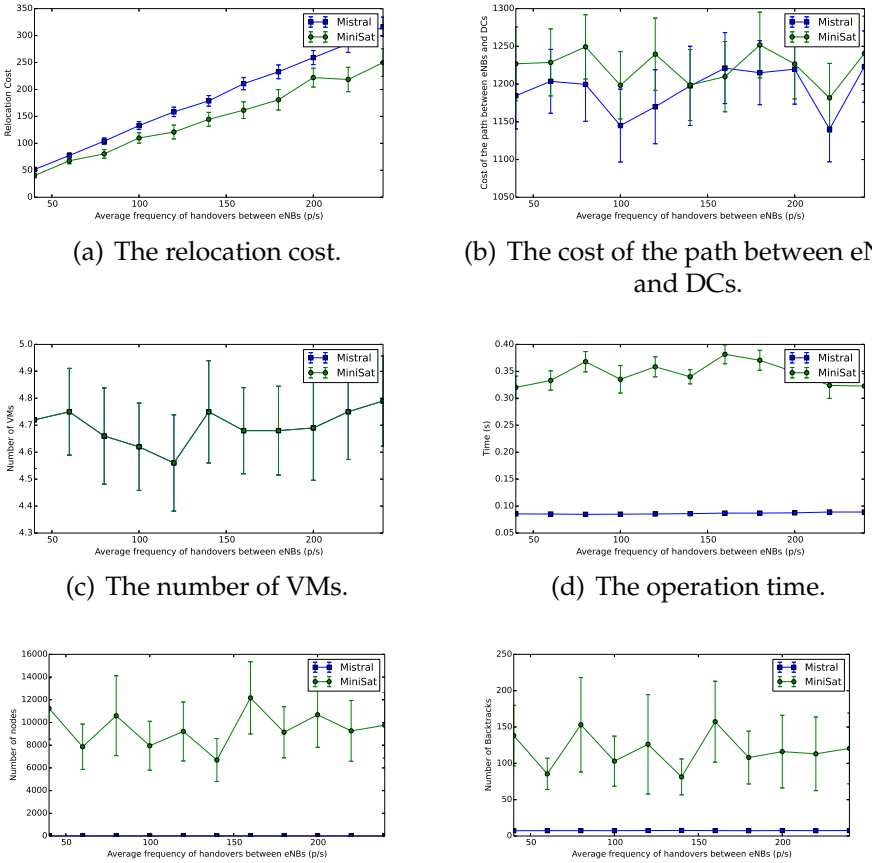


(b) The cost of the path between eNBs and DCs.



(c) The number of VMs.

FIGURE 3.11: Functions performance using Mistral solver.



(a) The relocation cost.

(b) The cost of the path between eNBs and DCs.

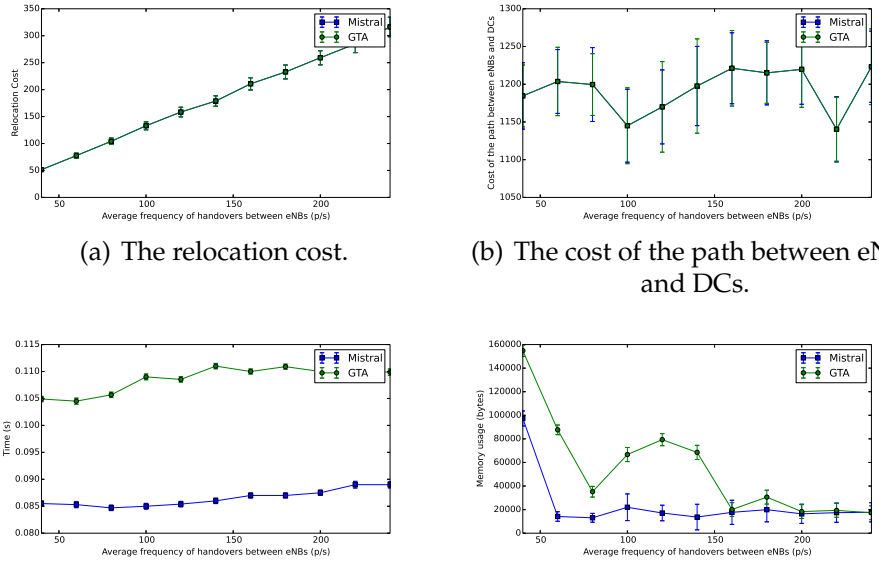
(c) The number of VMs.

(d) The operation time.

(e) The number of nodes.

(f) The number of backtracks.

FIGURE 3.12: Mistral and Minisat performance using *Primal mapping* function.



(a) The relocation cost.

(b) The cost of the path between eNBs and DCs.

(c) The operation time.

(d) The memory usage.

FIGURE 3.13: Mistral solver and GTA performance.

Chapter 4

An Adaptive Solution for Virtual Network Functions Placement in 5G Network Architecture

4.1 Introduction

In Cloud Computing field, computation and memory resources are becoming a relevant growing business. On the other hand, mobile network architecture faces many hurdles, including lack of flexibility for providing enhanced services and distributed architecture, and expensive cost to ensure a network topology that meets the users' equipment (UEs) needs. To cope with these problems, Cloud Computing is used in mobile telecommunications market thanks to network functions virtualization. In this chapter, we propose a fuzzy controller to support virtual network functions placement and provide an adaptive solution to manage and organize the network. Our approach enables the solution to adapt to UEs mobility and their needs in term of quality of experience. Furthermore, it minimizes Serving Gateways relocation cost and the path between UEs and Packet Data Network Gateways taking into account the resources capacities.

The proposed approach brings together the following essential points:

- The proposed approach is an adaptive solution, in other words, the solution depends on the behavior of users' equipment (UEs) regarding the nature of their mobility;
- The second point representing the adaptability of the solution is the setting of the target quality of experience (QoE) depending on UEs applications needs;
- The solution aims at reducing the relocation cost of S-GWs when a User Equipment (UE) moves from a service area to another;

- The cost of the path between eNBs and DCs hosting P-GWs is minimized for better QoE;
- Our approach allows us to find a compromise between the objectives mentioned above;
- Furthermore, the constraints on VMs resource capacities are considered.

The cited related studies mentioned in Section 2.1 have guided us to work on the VNFs placement problem to improve the multi-objective methods proposed in the previous chapter. Indeed, the problem is based on conflicting objectives where a compromise must be found, and also depends on external factors relied on UE behavior and needs. In our proposed approach, the problem is formulated as a multi-objective optimization problem and solved using the weighted-sum method. We also introduce as first phase a fuzzy controller that provides fuzzy weights to the weighted-sum method. By doing this, the solution is adapted to the UEs behavior (i.e., The mobility of UEs) and their needs (i.e., The QoE depending on the used applications). Therefore, the objective of using fuzzy logic consists of taking into account various external factors to achieve a fair decision. The main point of difference between the proposed approach and the solutions mentioned above is the real-time adaptation of the solution while minimizing the cost of the path between DCs and eNBs and the cost of S-GWs relocation.

4.2 Problem Statement and Trade-off Solutions in Literature

4.2.1 Problem Definition

We assume that the network is composed of a number $\mathcal{DC}$ of data-centers and a number $\mathcal{E}$ of eNBs. We aim to place some VMs running S-GWs and P-GWs VNFs. Thus, we consider that ω_j is the total amount of traffic generated by UE in an eNB $j \in \mathcal{E}$. Meanwhile, UE generates a total amount of traffic for an application $k \in \mathcal{K}$ in the eNB j that we denote as λ_j^k. When a UE moves from a service area to another the relocation of S-GWs occurs with a cost that we denote by C_{Reloc}. Moreover, a transfer is established between eNBs and the average frequency of handovers between two eNBs such that $j_1, j_2 \in \mathcal{E}^2$ is represented by $h(j_1, j_2)$. We assume that $c(i, j)$ is the cost of the path between a data-center $i \in \mathcal{DC}$ and an eNB $j \in \mathcal{E}$. We hereafter consider that the cost is in term of geographical distance, although other metrics could be taken into account. The maximum capacities of DCs hosting S-GWs and P-GWs VMs are, respectively, SGW_{max} and PGW_{max}^k given that VMs have a fixed quantity of hardware resources.

We define $\mathcal{U}(\mathcal{DC}, \mathcal{E})$ as a binary matrix. When an eNB j is connected to the data-center i then $\mathcal{U}(i, j) = 1$. Otherwise $\mathcal{U}(i, j) = 0$. Let $\mathcal{V}(\mathcal{E}, \mathcal{E})$ denotes the binary symmetric matrix, when two eNBs j_1 and j_2 are connected to the same data-center then $\mathcal{V}(j_1, j_2) = 1$ and $\mathcal{V}(j_1, j_2) = 0$ otherwise. The objective that minimizes the cost of the path between DCs and eNBs is represented as follows:

$$\textbf{min } Obj1 = \sum_{i \in \mathcal{DC}} \sum_{j \in \mathcal{E}} c(i, j)\, \mathcal{U}(i, j) \tag{4.1}$$

We aim at minimizing the cost of S-GWs relocation. Therefore, this objective is defined as follows:

$$\textbf{min } Obj2 = \sum_{j_1 \in \mathcal{E}} \sum_{j_2 \in \mathcal{E}} h(j_1, j_2) C_{Reloc} (1 - \mathcal{V}(j_1, j_2)) \tag{4.2}$$

We formulate our multi-objective VNFs placement problem as a weighted-sum integer programming as follows:

$$\begin{cases} \textbf{min} \;\; \alpha\, Obj1 + \beta\, Obj2 \\ \textbf{s. t.} \\ (4.3.1) \;\; \forall i \in \mathcal{DC}, \sum_{j \in \mathcal{E}} \omega_j \mathcal{U}(i,j) \leq SGW_{max} \\ (4.3.2) \;\; \forall i \in \mathcal{DC}, \forall k \in \mathcal{K}, \sum_{j \in \mathcal{E}} \lambda_j^k \mathcal{U}(i,j) \leq PGW_{max}^k \\ (4.3.3) \;\; \forall j \in \mathcal{E}, \sum_{i \in \mathcal{DC}} \mathcal{U}(i,j) = 1 \\ (4.3.4) \;\; \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \forall i \in \mathcal{DC}, \mathcal{U}(i,j_j) + \mathcal{U}(i,j_2) \leq 1 + \mathcal{V}(j_1,j_2) \\ (4.3.5) \;\; \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \forall i \in \mathcal{DC}, \mathcal{U}(i,j_j) - \mathcal{U}(i,j_2) \leq 1 - \mathcal{V}(j_1,j_2) \\ (43.6) \;\; \forall i \in \mathcal{DC}, \forall j \in \mathcal{E}, \mathcal{U}(i,j) \in \{0,1\} \\ (4.3.7) \;\; \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \mathcal{V}(j_1,j_2) \in \{0,1\} \\ (4.3.8) \;\; \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \mathcal{V}(j_1,j_2) = \mathcal{V}(j_2,j_1) \end{cases} \tag{4.3}$$

Constraints (4.3.1), (4.3.2), and (4.3.3) are problem-related constraints. The first and the second ones ensure that the total amount of traffic generated by UEs for S-GWs and P-GWs does not exceed the maximum capacities of DCs and the third one ensures that an eNB must be connected to only one DC. The remaining constraints are modelization topology-related constraints and are explained as follows:

- Constraint (4.3.4) guarantees that if $\mathcal{V}(j_1, j_2) = 1$ then eNB j_1 and eNB j_2 must be connected to the same DC.
- Constraint (4.3.5) guarantees that if $\mathcal{V}(j_1, j_2) = 0$ then eNB j_1 and eNB j_2 must not be connected to the same DC.
- Constraints (4.3.6) and (4.3.7) ensure that $\mathcal{U}$ and $\mathcal{V}$ are binary.
- Constraint (4.3.8) ensures that $\mathcal{V}$ is symmetric.

The solution that minimizes only the cost of the path between DCs and eNBs is named WSP (i.e., Weighted Sum method for Path reduction), where $\alpha = 1$ and $\beta = 0$. Unlike the first solution, WSR (i.e., Weighted Sum method for Relocation reduction) minimizes only the cost of S-GWs relocation, consequently, $\alpha = 0$ and $\beta = 1$. These two solutions help at finding the worst values of the two objectives which are used in the next section and that we denote as $Obj1_{worst}$ and $Obj2_{worst}$. Then, in the next section, we present the literature solutions used to find a compromise between the two objectives.

4.2.2 Trade-off Solutions in Literature

In this section, we present the literature solutions for VNFs placement namely Game Theory approach and Constraint Programming approach that are mentioned in Section 2.1 and Chapter 3.

Game Theory Approach

Nash bargaining theory is a branch of game theory which finds an optimal point taking into account reference values that are the worst obtained values of the objectives. Consequently, in our case, $Obj1_{worst}$ and $Obj2_{worst}$ are the reference values. The trade-off solution using Nash bargaining theory is called Game Theory Approach (GTA). GTA solution is formulated as follows:

$$\begin{cases} \textbf{max} \;\; (Obj1_{worst} - Obj1) \times (Obj2_{worst} - Obj2) \\ \textbf{s. t.} \\ (4.4.1) \;\; \forall i \in \mathcal{DC}, \sum_{j \in \mathcal{E}} \omega_j \mathcal{U}(i,j) \leq SGW_{max} \\ (4.4.2) \;\; \forall i \in \mathcal{DC}, \forall k \in \mathcal{K}, \sum_{j \in \mathcal{E}} \lambda_j^k \mathcal{U}(i,j) \leq PGW_{max}^k \\ (4.4.3) \;\; \forall j \in \mathcal{E}, \sum_{i \in \mathcal{DC}} \mathcal{U}(i,j) = 1 \\ (4.4.4) \;\; \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \forall i \in \mathcal{DC}, \mathcal{U}(i,j_j) + \mathcal{U}(i,j_2) \leq 1 + \mathcal{V}(j_1,j_2) \\ (4.4.5) \;\; \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \forall i \in \mathcal{DC}, \mathcal{U}(i,j_j) - \mathcal{U}(i,j_2) \leq 1 - \mathcal{V}(j_1,j_2) \\ (4.4.6) \;\; \forall i \in \mathcal{DC}, \forall j \in \mathcal{E}, \mathcal{U}(i,j) \in \{0,1\} \\ (4.4.7) \;\; \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \mathcal{V}(j_1,j_2) \in \{0,1\} \\ (4.4.8) \;\; \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \mathcal{V}(j_1,j_2) = \mathcal{V}(j_2,j_1) \end{cases} \tag{4.4}$$

The constraints in Equation (4.4) are the same as in Equation (4.3).

Constraint Satisfaction Problem

We also formulate our problem in the form of a constrained problem as in Chapter 3. Taking into account the two objectives, constraint satisfaction problem solution finds a compromise between minimizing the cost of the path between DCs and eNBs and reducing the cost of S-GWs relocation, where the variables are $\mathcal{U}$ and $\mathcal{V}$, and the domains are $D_{\mathcal{U}} = D_{\mathcal{V}} = \{0,1\}$. The constraints are (4.3.1), (4.3.2), (4.3.3), (4.3.4), (4.3.5) and (4.3.8). In the remainder of the paper, we name this solution Constraint Satisfaction Problem (CSP).

4.3 Contribution 3: A Fuzzy Controller for an Adaptive Virtual Network Functions Placement in 5G Network Architecture

4.3.1 The Proposed Approach: Fuzzy Controller and Weighted-Sum Adaptive Solution

In this section, we present the proposed solution for the multi-objective VNFs placement problem. This solution aims at finding a compromise between the objectives and adapting the solution to the nature of UE mobility and the target QoE. To do so, we define a fuzzy controller that provides a fuzzy weight to the VNFs placement system as depicted in Figure 4.1. The input of the fuzzy controller is the frequency of handovers between eNBs that represents the nature of UEs mobility. The second input is the target QoE which could be set according to the kind of applications used by UE. The membership function of the frequency of handovers between eNBs is shown in Figure 4.2(a), it is defined by the crisp numbers (Low, Medium, High). Figure 4.2(b) displays the membership function of the target QoE, its membership is determined by the crisp numbers (Poor, Average, Good). The output membership function is depicted in Figure 4.2(c) where the membership function of the weight is defined by (Low, Small, High). We define knowledge base rules and build a set of rules in the form of If-Then-Else structures (See Table 4.1). In Table 4.1, for example, if the frequency of handovers is high and the target QoE is good then the weight of $obj1$ is small, and so on. The crisp inputs are converted into fuzzy data using the membership functions and the data rules. Afterward, the fuzzy result is converted into a crisp data (i.e., the weight).

The fuzzy weights of the two objectives are named $\bar{\alpha}$ and $\bar{\beta}$, and the weighted-sum approach is used to find the mapping of eNBs to DCs depending on the

TABLE 4.1: The knowledge base rules

Handovers/QoE	High	Medium	Low
Good	Small	Low	Low
Average	High	Small	Low
Poor	High	High	Small

inputs. The integer programming in Equation (4.3) is modified as follows:

$$
\begin{cases}
\textbf{min} \;\; \bar{\alpha}\, Obj1 + \bar{\beta}\, Obj2 \\
\textbf{s. t.} \\
(6.1) \quad \forall i \in \mathcal{DC}, \sum_{j \in \mathcal{E}} \omega_j \mathcal{U}(i,j) \leq SGW_{max} \\
(6.2) \quad \forall i \in \mathcal{DC}, \forall k \in \mathcal{K}, \sum_{j \in \mathcal{E}} \lambda_j^k \mathcal{U}(i,j) \leq PGW_{max}^k \\
(6.3) \quad \forall j \in \mathcal{E}, \sum_{i \in \mathcal{DC}} \mathcal{U}(i,j) = 1 \\
(6.4) \quad \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \forall i \in \mathcal{DC}, \mathcal{U}(i,j_j) + \mathcal{U}(i,j_2) \leq 1 + \mathcal{V}(j_1,j_2) \\
(6.5) \quad \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \forall i \in \mathcal{DC}, \mathcal{U}(i,j_j) - \mathcal{U}(i,j_2) \leq 1 - \mathcal{V}(j_1,j_2) \\
(6.6) \quad \forall i \in \mathcal{DC}, \forall j \in \mathcal{E}, \mathcal{U}(i,j) \in \{0,1\} \\
(6.7) \quad \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \mathcal{V}(j_1,j_2) \in \{0,1\} \\
(6.8) \quad \forall j_1 \in \mathcal{E}, \forall j_2 \in \mathcal{E}, \mathcal{V}(j_1,j_2) = \mathcal{V}(j_2,j_1)
\end{cases}
\tag{4.5}
$$

In what follows, the solution obtained from the integer programming in Equation (4.5) is called Fuzzy Weighted-Sum (FWS) where the constraints are the same as in Equation (4.3). The proposed VNF placement system displayed in Figure gives an adaptive solution to the problem. However, it considers only two objectives. Adding more objectives will make the system more complex, and the set of rules hard to build.

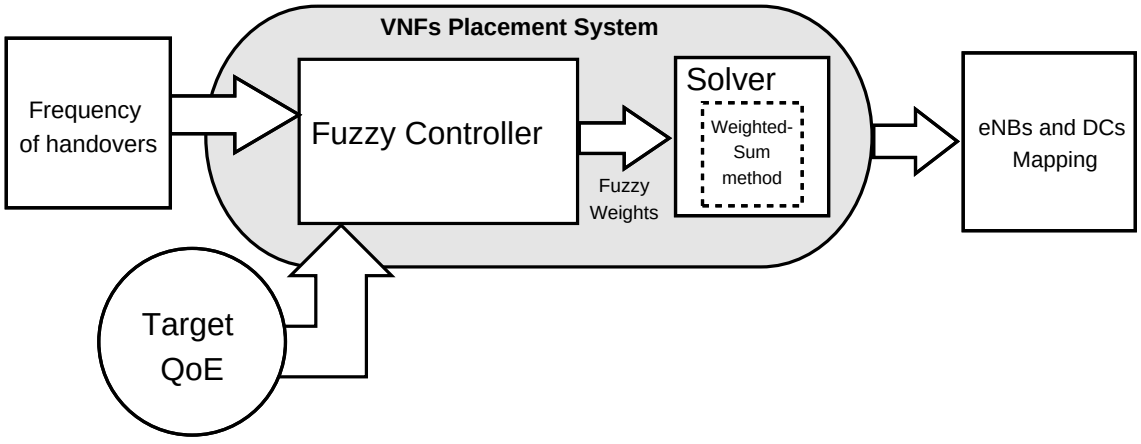
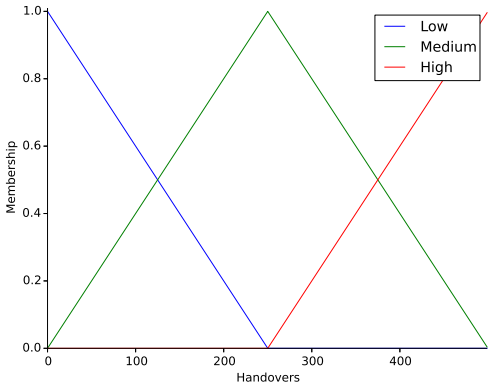
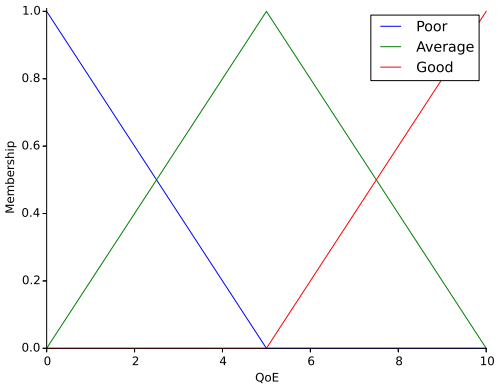


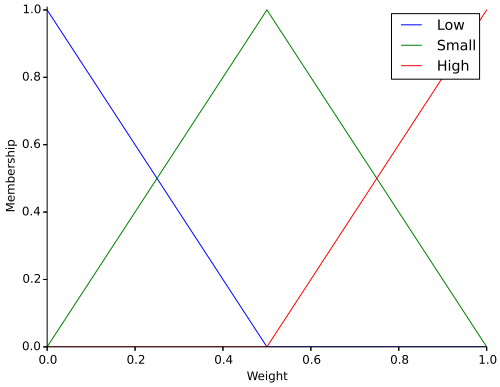
FIGURE 4.1: Fuzzy controller for VNFs placement



(a) The membership function of the frequency of handovers.



(b) The membership function of the QoE.



(c) The membership function of the weight.

FIGURE 4.2: The membership functions.

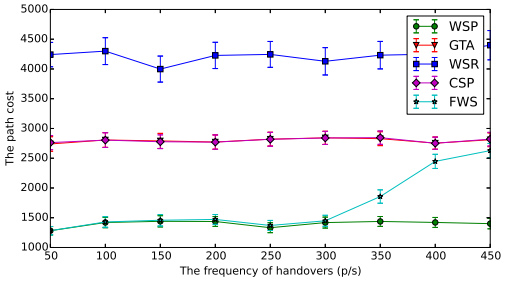
4.3.2 Implementation and Results

In this section, we present the deployed parameters used to compare the proposed approach with other literature methods (i.e., GTA and CSP). We implement the simulator using Python programming language. Scikit-Fuzzy is a fuzzy controller toolkit used in the simulator to generate the weights for the proposed FWS approach. Gurobi Optimizer solver is used to solve WSR, WSP, GTA, and FWS approaches. CSP approach is solved using Mistral library.

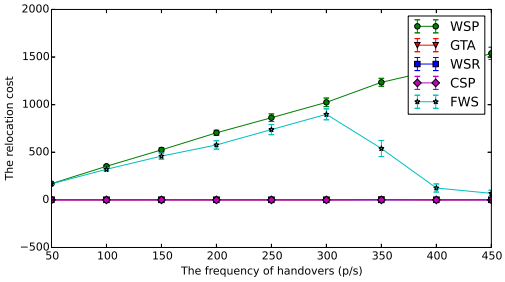
In this simulation, the network composed of DCs and eNBs is deployed randomly in an area of 2000m by 2000m. The number of DCs is fixed at 10, and the number of eNBs is fixed at 4. We hereafter consider three scenarios: Scenario 1 represents the case where the target QoE is good while scenario 2 shows the case where target QoE is average; Scenario 3 experiences a poor target QoE (See Table 4.2). In the results depicted in Figure 4.3, Figure 4.4, and Figure 4.5, each

TABLE 4.2: Simulation parameters

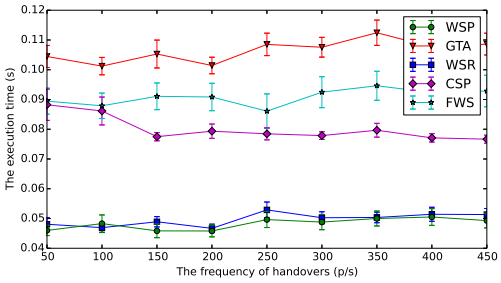
	Target QoE	X/Y coordinates	Number of DCs	Number of eNBs
Scenario 1	Good]5,10]	random[0,2000]	10	4
Scenario 2	Average [5]	random[0,2000]	10	4
Scenario 3	Poor [0,5[	random[0,2000]	10	4



(a) The cost of the path between eNBs and DCs.



(b) The cost of S-GWs relocation.



(c) The execution time.

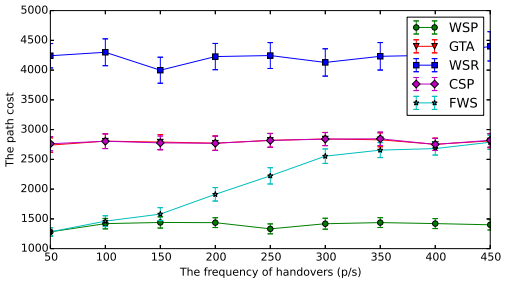
FIGURE 4.3: The performance evaluation in scenario 1.

plotted point represents the average of 100 times of executions. However, we vary all the parameters except C_{Reloc}, SGW_{max}, and PGW^{k}_{max} which remain unchanged. The plots are presented with 95% confidence interval. To assess WSR, WSP, GTA, CSP and the proposed approach which we named FWS, we vary the frequency of handovers between eNBs in the simulation from 50 to 450 p/s. In other terms, we vary the mobility of UE from low to high.

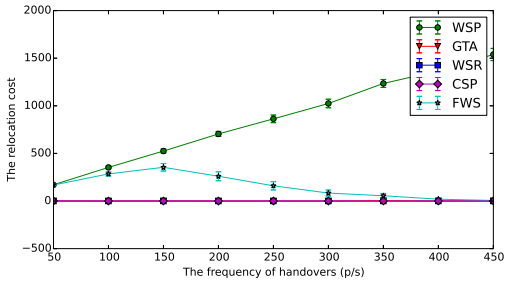
In the next section, the results are detailed, and the performance evaluation is carefully discussed.

Performance Evaluation

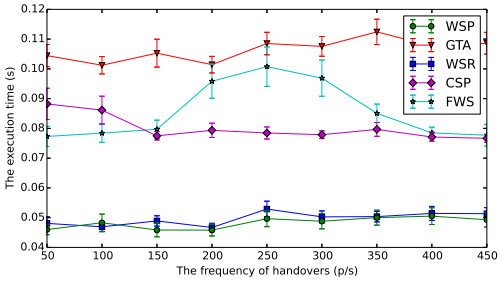
In this section, the performance evaluation of the approaches mentioned above in the three scenarios is detailed. Figure 4.3 shows the performance evaluation in scenario 1 and Figure 4.4 presents the performance evaluation in scenario 2.



(a) The cost of the path between eNBs and DCs.



(b) The cost of S-GWs relocation.

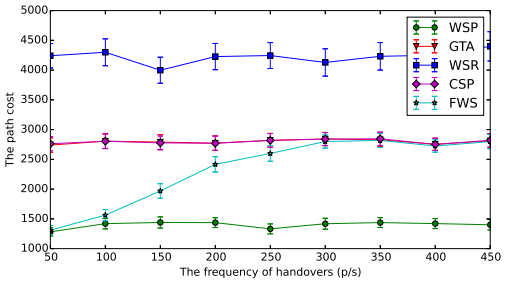


(c) The execution time.

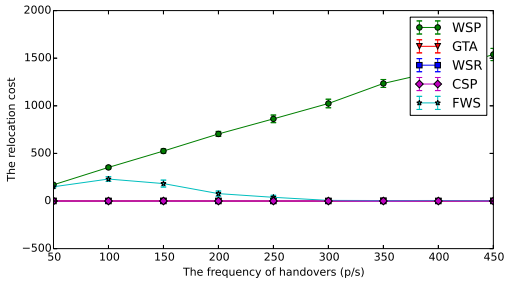
FIGURE 4.4: The performance evaluation in scenario 2.

Simulation results of scenario 3 are depicted in Figure 4.5. The approaches are evaluated regarding the cost of the path between eNBs and DCs, the cost of the relocation of S-GWs, and the execution time.

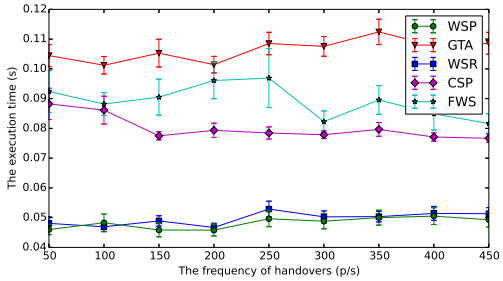
The cost of the path between eNBs and DCs. Figure 4.3(a), Figure 4.4(a), and Figure 4.5(a) represent the approaches simulation results, in, respectively, scenario 1, scenario 2, and scenario 3 regarding the cost of the path between eNBs and DCs. In the three scenarios, we notice that WSP shows the best performance in term of the cost of the path minimization while WSR exhibits the worst performance. However, GTA and CSP methods demonstrate almost the same performance and find a compromise between the two objectives. Concerning the four approaches, the cost remains stable while increasing the average frequency of handovers between eNBs. Whereas, the cost of the path increases regarding FWS approach. Accordingly, when the frequency of handovers is low FWS performs like WSP and when the frequency of handovers is high FWS works like GTA, and CSP approaches. Table 4.3 displays the percentage of reduction concerning the cost of the path between eNBs and DCs in the three scenarios. This percentage is calculated based on the worst values of the cost performed by WSR. GTA and CSP solutions exhibit the same percentage of reduction regardless of the target



(a) The cost of the path between eNBs and DCs.



(b) The cost of S-GWs relocation.



(c) The execution time.

FIGURE 4.5: The performance evaluation in scenario 3.

QoE and the frequency of handovers. In scenario 1, when the frequency of handovers is high, the reduction percentage is equal to 50.74%, and when it is low, the percentage of reduction is equal to 66.36%. In scenario 3, where the target QoE is poor, the reduction is only 34.50% when the frequency of handovers is high. And it is 56.71% when the frequency of handovers is low. Therefore, we notice that the FWS approach is more adaptive compared to the other approaches regarding the target QoE and the mobility of UEs.

The cost of the S-GWs relocation. Figure 4.3(b), Figure 4.4(b), and Figure 4.5(b) show the cost of S-GWs relocation in the three scenarios. WSP approach exhibits the worst performance and demonstrates that the cost of relocation increases while the frequency of handovers between eNBs grows. WSR, GTA, and CSP approaches show the same performance in term of the minimization of the S-GWs relocation cost. In the three scenarios, the FWS approach finds a compromise between the two objectives. However, we observe that the cost of S-GWs relocation decreases while the mobility of UEs becomes high concerning this last approach. For that reason, we notice a second time that FWS approach is an adaptive solution. The percentage of reduction of the cost of S-GWs relocation for the three scenarios is presented in Table 4.4. GTA and CSP approaches show a reduction percentage up to 100%. However, the reduction remains constant whether the

TABLE 4.3: The cost of the path between eNBs and DCs improvement while varying the frequency of handovers.

	Scenario 1		
	High	Medium	Low
GTA	33.95 %↓	33.56 %↓	33.74 %↓
CSP	33.81 %↓	33.61 %↓	33.71 %↓
FWS	50.74 %↓	67.72 %↓	66.36 %↓
	Scenario 2		
	High	Medium	Low
GTA	33.95 %↓	33.56 %↓	33.74 %↓
CSP	33.81 %↓	33.61 %↓	33.71 %↓
FWS	37.23 %↓	47.62 %↓	62.80 %↓
	Scenario 3		
	High	Medium	Low
GTA	33.95 %↓	33.56 %↓	33.74 %↓
CSP	33.81 %↓	33.61 %↓	33.71 %↓
FWS	34.50 %↓	38.76 %↓	56.71 %↓

target QoE is good or poor. Whereas, the FWS approach exhibits a reduction percentage equal to 68.41%, in the high frequency of handovers, in scenario 1. And it exhibits a reduction percentage equal to 99.60% when the frequency of handovers is high, in scenario 3. Likewise, when the mobility of UEs is low the reduction percentage is 12.95%, in scenario 1, and it increases to 63.40%, in scenario 3. From high to low mobility of UEs, the fuzzy approach allows decreasing the percentage of reduction. Moreover, from good target QoE to poor one, the percentage of reduction of the cost of S-GWs relocation increases. Therefore, FWS is more adaptive concerning the second objective.

The execution time. Figure 4.3(c), Figure 4.4(c), and Figure 4.5(c) display the execution time of all approaches, in scenario 1, scenario 2, and scenario 3, respectively. WSR and WSP solutions exhibit almost the same performance and the execution time tends to be constant for both methods while it does not exceed 0.06s. GTA shows the worst values of execution time which reach up to 0.12s. FWS approach outperforms GTA concerning the metric of execution time which not exceed 0.11s for the proposed method. Whereas, the CSP approach provides better execution time compared to GTA and FWS. We notice that the execution time regarding CSP approach does not exceed 0.09s despite having the same performance as GTA regarding minimizing the cost of the path between DCs and eNBs, and the cost of S-GWs relocation. Therefore, the CSP approach outperforms GTA.

TABLE 4.4: The relocation cost of S-GWs improvement while varying the frequency of handovers.

	Scenario 1		
	High	Medium	Low
GTA	99.90 %↓	100.0 %↓	99.92 %↓
CSP	100.0 %↓	100.0 %↓	100.0 %↓
FWS	68.41 %↓	14.43 %↓	12.95 %↓
	Scenario 2		
	High	Medium	Low
GTA	99.90 %↓	100.0 %↓	99.92 %↓
CSP	100.0 %↓	100.0 %↓	100.0 %↓
FWS	96.82 %↓	81.47 %↓	39.11 %↓
	Scenario 3		
	High	Medium	Low
GTA	99.90 %↓	100.0 %↓	99.92 %↓
CSP	100.0 %↓	100.0 %↓	100.0 %↓
FWS	99.60 %↓	95.49 %↓	63.40 %↓

Discussion

In this simulation, we notice the effectiveness of FWS in adapting to external factors. In Chapter 3 game theory and constraint satisfaction problem approaches give a trade-off solution to the conflicting solutions, though, the external environment is not considered. However, our proposal gives different solutions according to the nature of the mobility of users' equipment and their target QoE. Indeed, when we vary the frequency of handovers between eNBs and the target QoE, the S-GWs relocation cost and the cost of the path between eNBs and DCs change without crossing the worst values. In other words, the adaptability of the solution to the environmental conditions is reached while established a compromise between the objectives like in literature solutions, hence the interest of fuzzification and defuzzification. The fuzzy controller enhanced the performance of the VNFs placement by providing fuzzy weight to the weighted-sum method, in an acceptable execution time which not exceeds the execution time of game theory approach. However, adding a third objective as in literature is a new challenge to cope with because additional objectives make the set of rules hardly to define.

4.3.3 Conclusion and limitations

The mobile telecommunications market is moving towards a new era. The marriage of Cloud Computing and the mobile network is a promising solution to

improve the network of the future, and this drives to many challenges in different research fields. In this vein, we addressed in this work the main challenging problem in this new mobile network paradigm which is the placement of VNFs.

A fuzzy controller was proposed besides multi-objective optimization for an adaptive VNFs placement over the Cloud. The results proved the efficacity of our proposed approach in finding a trade-off solution while adapting to users' equipment mobility and QoE needs. Indeed, the evaluation of performance shows that our adaptive solution outperforms literature methods while producing a fair time of execution compared to these same methods.

Further objectives to the problem of VNFs placement could be included, among them, we cite reducing the cost paid by the operator by minimizing the allocated VMs (i.e., the number of the VNFs of S-GWs and P-GWs). Nevertheless, it brings a new challenge which is how to define complicated base rules with more than two inputs. Additionally, more EPC components such as MME could be added to the system model.

Chapter 5

Multi-Objective Optimization for Mobile Gateways Selection in Vehicular Ad-Hoc Networks

5.1 Contribution 4: Gateway selection in Vehicular Ad-hoc Network

5.1.1 Introduction

Mobile gateways are deployed to improve Internet services in Vehicular Ad-hoc Network (VANET). In this vein, this work argued the need for selecting an appropriate gateway for vehicles without access to the Internet according to specific criteria and based on a gateway discovery system (See Section 2.2). The proposed solution introduces a method of multiple-criteria decision analysis helping to find suitable gateways to our problem with two main design goals: *i)* minimizing the number of overloaded gateways *ii)* maximizing the number of connected vehicles. The two design goals effectively represent two conflicting objectives that we deal with finding a tradeoff. The selection of a suitable gateway for all vehicles in need of internet access is studied by adopting the prescriptive approach of multi-criteria analysis PROMETHEE (Preference Ranking Organization Method for Enrichment Evaluations) (Brans and Vincke, 1985). This is a method of multiple-criteria decision analysis helping to find suitable gateways to our problem according to several criteria, by avoiding overloaded gateways and realizing a fair distribution (Load Balancing). Indeed, the aim is to minimize the overloaded gateways number while maximizing the number of connected vehicles. PROMETHEE method is used for more refined modeling and to arrange all gateways from best to worst and get a relative valuation of each of them by giving a weight to each criterion according to its importance. Our work focuses

on the distance, speed, direction and the number of clients using a gateway to minimize the number of overloaded gateways while connecting the maximum of clients. But other criteria can be taken into account, and that is the advantage of the PROMETHEE method.

5.1.2 Problem formulation

It is assumed that a VANET network consists of a number $\mathcal{CV}$ of CVs and a number $\mathcal{MG}$ of MGs. The Euclidean distance between a CV $i \in \mathcal{CV}$ and a MG $j \in \mathcal{MG}$ is:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{5.1}$$

V_i and V_j are respectively the velocities of a CV $i \in \mathcal{CV}$ and a MG $j \in \mathcal{MG}$. D_i and D_j are respectively the direction of a CV $i \in \mathcal{CV}$ and a MG $j \in \mathcal{MG}$ such that $0 \leq D_i, D_j \leq 2\pi$, r is vehicles radio propagation range and $Overload(j)$ is a value representing the number of CVs connected to the MG j, j is the best MG for the vehicle i when the following conditions are satisfied:

$$\forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, dij \leq r \tag{5.2}$$

$$\forall i \in \mathcal{CV}, dij = min \sum_{j \in \mathcal{MG}} dij \tag{5.3}$$

$$\forall i \in \mathcal{CV}, |V_i - V_j| = min \sum_{j \in \mathcal{MG}} |V_i - V_j| \tag{5.4}$$

$$\forall i \in \mathcal{CV}, |D_i - D_j| = min \sum_{j \in \mathcal{MG}} |D_i - D_j| \tag{5.5}$$

- The constraint in Equation (5.2) ensures that the MG is not outside the area of the client radio propagation range.
- The constraint in Equation (5.3) ensures that the adequate MG is the closest.
- The constraint in Equation (5.4) ensures that the adequate MG has the closest velocity.
- The constraint in Equation (5.5) ensures that the adequate MG has the same direction as the CV.

To remedy to the overloaded gateways problem, another constraint is added which ensures that a given MG must have a minimum number of connected CVs:

$$\forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, min\ Overload(j) \tag{5.6}$$

In other words, this problem relies on the selection of adequate MG for the client requesting access to the Internet according to the five criteria mentioned above. The last criterion allows minimizing the number of customers connected to a gateway, but the maximum of vehicles requesting Internet access must be connected. The PROMETHEE method described in the next section allows reaching a tradeoff.

5.1.3 The proposed solution

PROM4: A solution with basic constraints

The multi-criteria decision aid method PROMETHEE is adapted to our gateway selection problem. PROMETHEE is a prescriptive approach of the multi-criteria analysis problem that presents a number of actions (or decisions) evaluated according to several criteria; it is based on the basic mechanism of pairwise actions comparison for each criterion. For our problem, decision makers are the CVs and actions are the gateways to choose. First, the matrix of four criteria (See above equations Equation (5.2), Equation (5.3), Equation (5.4) and Equation (5.5)) according to G various alternatives must be determined by assigning a weight to each criterion according to its importance. The constraint in Equation (5.2) has the priority because if the gateway is outside the radio propagation range, the communication delay between the MG and the vehicle will increase. Note that if $D_i = D_j$ and $V_i = V_j$ the link lifetime tends to ∞, but if D_i is different from D_j the link lifetime will be short. Indeed, the constraint in Equation (5.5) is more important than the constraint in Equation (5.4). The constraint in Equation (5.3) comes last. The weights (the criterion importance as a percentage close to 1 if very important close to 0 if very insignificant) are chosen such as $\sum_{i=1}^{4} \omega_i = 1$. The choice of weights is made randomly following the priorities (1 to 4) (See Table 5.1).

TABLE 5.1: PROM4 priorities

	Range	Direction	Velocity	Distance
Priority	1	2	3	4

Data are presented in the form of a table containing $Gx4$ assessments; each line corresponds to an action and each column corresponds to a criterion (f_{Range},

$f_{Direction}$, $f_{Velocity}$, $f_{Distance}$). First, a pairwise comparison is made between all the actions for each criterion. $d_k(a_i, a_j)$ denotes the difference between assessments of two actions for the criterion f_k and calculated as follows: $d_k(a_i, a_j) = f_k(a_i) - f_k(a_j)$. Then the notion of preference function is introduced to translate the difference into a unicriterion preference degree as follows: $\Pi_k = P_k[d_k(a_i, a_j)]$. Where P_k: $\mathbb{R} \Rightarrow [0, 1]$ is a positive non-decreasing preference function such that $P_k(0) = 0$. The usual function Equation (5.7) is used for the criteria in Equation (5.2) and Equation (5.5), and the U-Shape function Equation (5.8) is used for the criteria Equation (5.3) and Equation (5.4), where q_j denotes the indifference threshold.

$$P_j(d_j) = \begin{cases} 0 & if \quad d_j \leq 0 \\ 1 & if \quad d_j > 0 \end{cases} \tag{5.7}$$

$$P_j(d_j) = \begin{cases} 0 & if \quad |d_j| \leq q_j \\ 1 & if \quad |d_j| > q_j \end{cases} \tag{5.8}$$

When a preference function has been associated with each criterion by the decision maker, all comparisons between all pairs of actions can be made for all the criteria. A multicriteria preference degree is then computed to globally compare every couple of actions: $\Pi = \sum_{k=1}^{q} P_k(a, b)\omega_k$. Complete ranking is obtained by ordering the actions according to the PROMETHEE II.

PROM5: A solution with additional constraints

To reduce the number of connected CVs to each MG and avoid overloaded gateways, the constraint in Equation (5.6) is introduced as a criterion in the $PROM5$ method, which is an improved $PROM4$ method. This method keeps the same steps of $PROM4$ by adding the $f_{Overload}$ criterion using the U-Shape function in Equation (5.8). This time, the weights are set by giving priority to $f_{Overload}$ criterion to achieve a fair distribution of CVs (See Table 5.2). $PROM4$and $PROM5$ have been simulated, and the results are presented in the following section.

TABLE 5.2: PROM5 priorities

	Range	Direction	Velocity	Distance	Overload
Priority	2	3	4	5	1

TABLE 5.3: Simulation scenarios

	Number of MGs
Scenario 1	10
Scenario 2	20
Scenario 3	50
Scenario 4	80

TABLE 5.4: Parameters

Paramater	Setting
CVs number	100
Direction	0° - 360°
Velocity	0-100 km/h
X	0-2000m
Y	0-2000m
Z	0
Transmission range	500m
$q_{j(Distance)}$	100
$q_{j(Speed)}$	10

5.1.4 Implementation and Results

The number of cvs connected to a gateway

Figure 5.1 shows that the $PROM5_{Q5}$ method has more effect on scenarios 1 and 2 because there are more overloaded MGs than scenarios 3 and 4. The gateway with the maximum number of connected CVs is located in scenario 1. The most influential $q_{j(Overload)}$ value is 5 while the least influential is 10. In scenario 4, there is no change between the results of $PROM4$ and $PROM5$ because the overload is minimal or nonexistent, but from scenario 3 the change is remarkable between the two methods. Hence varying the $q_{j(Overload)}$ value helps to find a tradeoff between minimizing overloaded gateways and maximizing the connected CVs number.

Gateway selection according to each criterion

Figure 5.2(a) is related to the first scenario; more than 80 CVs have selected an appropriate MG located within the transmission range. For the rest of CVs, MGs are beyond their transmission range. In scenarios 2 and 3, over 90 CVs found adequate MG and 100 CVs in scenario 4, because of the large number of available MG compared to the number of CVs in this scenario. Note that the scenario with the highest change between the $PROM5$ and $PROM4$ methods is the first

TABLE 5.5: The indifference threshold variation

	$PROM5_{Q10}$	$PROM5_{Q8}$	$PROM5_{Q5}$
$q_{j(Overload)}$	10	8	5

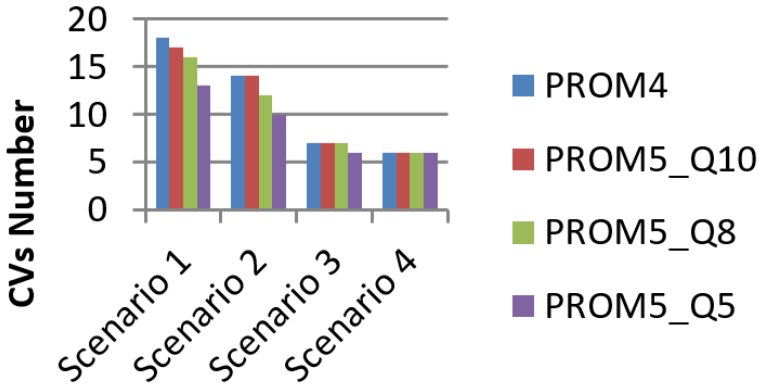


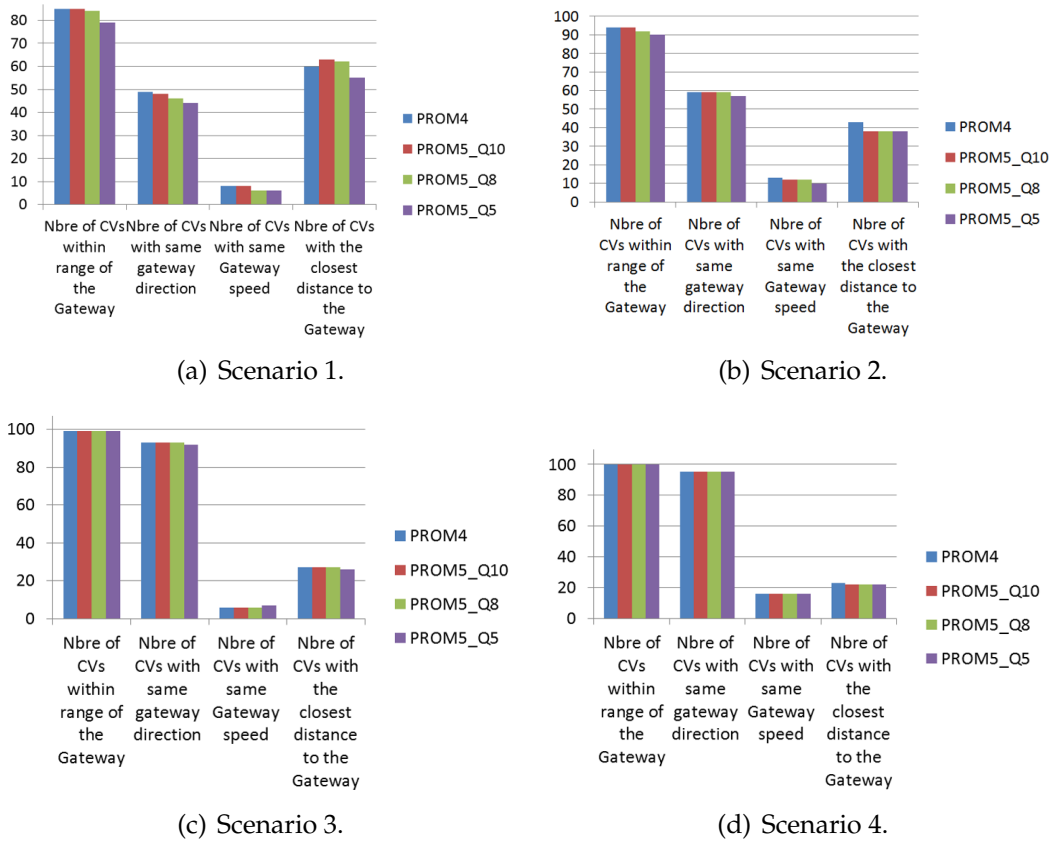
FIGURE 5.1: The maximum number of CVs connected to a Gateway

given that the number of MGs is minimal and, therefore, there are more overloaded gateways while no change for the scenario 4 is noted. The number of CVs that have selected an MG within their transmission range does not change between $PROM4$ and $PROM5_{Q10}$ methods in the four scenarios, but the change is remarkable with the $PROM5_{Q5}$ method. The number of CVs that have chosen an MG with their directions decreases from the $PROM4$ method to the $PROM5_{Q5}$ method, and it is the same thing for scenario 2. In scenarios 3 and 4, there is no noticeable change because both scenarios do not contain a significant overloaded gateways number (See Figure 5.1).

5.1.5 Conclusion and limitations

Mobile gateways are used to better Internet services in VANET. In this work, we show the need for selecting an appropriate gateway for vehicles without access to the Internet according to specific criteria. The selection of a mobile gateway is performed on a gateway discovery system based on cloud computing. The main contribution is introducing two additional criteria that are the number of connected vehicles and the traffic charge of each gateway using a multiple-criteria decision method.

Ineed, a new method of selecting an appropriate mobile gateway, for vehicles in the request for access to the Internet, is presented, based on an existing gateway discovery system. This method is based on the PROMETHEE prescriptive



(a) Scenario 1. (b) Scenario 2.

(c) Scenario 3. (d) Scenario 4.

FIGURE 5.2: Simulation results in the four scenarios.

approach of multi-criteria analysis and allowed to consider several criteria with different weights and to find a trade-off between the number of connected vehicles and the number of overloaded gateways. Still, among the limitations of this work we can cite that the solutio is a multi-criteria selection and there is no optimization. Furthermore, there is an obligation of using thresholds to find a trade-off.

5.2 Contribution 5: A Multi-Objective Optimization System for Mobile Gateways Selection in Vehicular Ad-Hoc Networks

5.2.1 Introduction

Vehicular Ad-Hoc Network provides essential Internet services to users. Hence, mobile gateways (MGs) are deployed to guarantee access to the Internet for the entire network. A significant issue in MGs discovery is the selection of the best gateway taking into account some constraints and trying to reach some conflicting high-level objectives. The number of connected Client Vehicles must be maximized while a fair load distribution must be performed. Therefore, we propose a multi-objective optimization system for MGs selection based on two models using different solving strategies allowing the decision maker to choose the adequate solution.

In this work, we improve the quality of selecting an MG by adding more constraints and objectives. Indeed, unlike literature solutions, we consider some high-level objectives such as maximizing the number of connected vehicles and minimizing the traffic amount handled by MGs to avoid overload situations. Since our problem is suited to be modeled using multiple conflicting objectives, we use multi-objective optimization which proves to be an excellent way to find solutions that constitute a trade-off between the objectives. As a result, the decision maker will be in a better position to make a choice when such trade-off solutions are exposed. In this paper, we propose three approaches to solve the multi-objective problem. The weighted-sum approach is used in the case of a priori articulation of preferences. Game theory and constraint programming approaches are used in the case of no articulation of preferences.

5.2.2 Problem formulation and solving strategy

System Overview and Methodology

The purpose of this study is selecting MGs in a gateway discovery system while maximizing the number of connected vehicles and minimizing the amount of traffic handled by each MG. A CV is a vehicle which wants to connect to the Internet, and an MG is a vehicle which can directly connect to the Internet. The discovery system is composed of two servers that provide services in the cloud. The first one maintains all information concerning MGs and the second one is responsible for affecting a CV to an MG. We propose a system that can be integrated

into the second server that selects the appropriate gateway. Integer Optimization Problem (IOP) and Constraint Optimization Problem (COP) are the adopted models in the proposed system. By doing this, separating the formulation and the search strategy is guaranteed. The decision maker can add or remove constraints to the problem depending on the state of the network and can set his preferences regarding the objectives. To solve the IOP, we use the WS approach in the case of a priori articulation of preferences. The GT approach is proposed in case there is no articulation of preferences to solve the IOP. The COP is solved using the Backtracking algorithm. The decision maker can choose the solution depending on high-level information with the help of methods comparison and diagrams.

Integer Optimization Problem

Problem statement. VANET network consists of a set of MGs which is represented by $\mathcal{MG}$ and a set of CVs in need to access the Internet which is represented by the $\mathcal{CV}$. The Euclidean distance between a CV $i \in \mathcal{CV}$ and a MG $j \in \mathcal{MG}$ is represented by d_{ij} such in Equation (5.1). Let ω_i denotes the traffic amount requested by the CV i. D_i and D_j are, respectively, the directions of a CV i and a MG j such that $0 \leq D_i, D_j \leq 2\pi$ and V_i and V_j are respectively the velocities of the CV and the MG. Let r denotes the MGs radio propagation range and V_{Max} denotes the maximum permitted velocity difference between a CV connected to an MG. Our contribution is to propose a solution to select MGs for the given CVs.

The relationship of CVs to MGs is represented through the binary matrix $\mathcal{X}(\mathcal{CV}$,
$\mathcal{MG})$. If and only if the CV i is connected to the MG j, then $\mathcal{X}(i,j) = 1$, otherwise $\mathcal{X}(i,j) = 0$. We define the binary symmetric matrix $\mathcal{Y}(\mathcal{CV}, \mathcal{CV})$, if and only if $i_1 \in \mathcal{CV}$ and $i_2 \in \mathcal{CV}$ are connected to the same MG, then $\mathcal{Y}(i_1, i_2) = 1$, otherwise $\mathcal{Y}(i_1, i_2) = 0$. The problem of MG selection is represented through the following integer program:

$$
\begin{cases}
(5.9.1) \quad \textbf{max} \ \sum_{i\in\mathcal{CV}} \sum_{j\in\mathcal{MG}} \mathcal{X}(i,j) \\
(5.9.2) \quad \textbf{min} \ \sum_{i_1\in\mathcal{CV}} \sum_{i_2\in\mathcal{CV}} \omega_{i_1} \mathcal{Y}(i_1,i_2) \\
\textbf{s. t.} \\
(5.9.3) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i_1,j) + \mathcal{X}(i_2,j) \leq 1 + \mathcal{Y}(i_1,i_2) \\
(5.9.4) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i_1,j) - \mathcal{X}(i_2,j) \leq 1 - \mathcal{Y}(i_1,i_2) \\
(5.9.5) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, d(i,j)\mathcal{X}(i,j) \leq r \\
(5.9.6) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, |V_i - V_j|\mathcal{X}(i,j) \leq V_{Max} \\
(5.9.7) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, |D_i - D_j|\mathcal{X}(i,j) = 0 \\
(5.9.8) \quad \forall i \in \mathcal{CV}, \sum_{j\in\mathcal{MG}} \mathcal{X}(i,j) = 1 \\
(5.9.9) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i,j) \in \{0,1\} \\
(5.9.10) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \mathcal{Y}(i_1,i_2) \in \{0,1\} \\
(5.9.11) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \mathcal{Y}(i_1,i_2) = \mathcal{Y}(i_2,i_1)
\end{cases}
\tag{5.9}
$$

The first objective aims to maximize as much as possible the number of connected CVs in need to access the Internet. Meanwhile, the second one points at reducing the number of CVs connected to the same MG by minimizing the traffic amount handled by the MGs of all the connected CVs. The constraints in the problem model are described as follows:

- Constraint (5.9.3) ensures that if $\mathcal{Y}(i_1, i_2) = 0$, i_1 and i_2 must not connect to the same MG.
- Constraint (5.9.4) ensures that if $\mathcal{Y}(i_1, i_2) = 1$, i_1 and i_2 must connect to the same MG.
- Constraint (5.9.5) ensures that if a CV i is connected to a MG j then i must be within the range of j.
- Constraint (5.9.6) ensures that if a CV i is connected to a MG j, then the difference between the two velocities must not exceed V_{Max}.
- Constraint (5.9.7) ensures that if a CV i is connected to a MG j, then they must have the same direction.
- Constraint (5.9.8) ensures that each CV must be connected only to one MG.
- Constraints (5.9.9) and (5.9.10) ensure that the matrices $\mathcal{X}$ and $\mathcal{Y}$ are binary.
- Constraint (5.9.11) ensures that the matrix $\mathcal{Y}$ is symmetric.

As a rule, it is more convenient to study and solve an optimization problem that aims to minimize or to maximize all the objectives. That is why the integer program in Equation (5.9) is reformulated and simplified as follows:

$$
\begin{cases}
(5.10.1) \quad \textbf{min} \quad \sum_{i \in \mathcal{CV}} \sum_{j \in \mathcal{MG}} (1 - \mathcal{X}(i,j)) \\
(5.10.2) \quad \textbf{min} \quad \sum_{i_1 \in \mathcal{CV}} \sum_{i_2 \in \mathcal{CV}} \omega_{i_1} \mathcal{Y}(i_1, i_2) \\
\textbf{s. t.} \\
(5.10.3) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i_1, j) + \mathcal{X}(i_2, j) \leq 1 + \mathcal{Y}(i_1, i_2) \\
(5.10.4) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i_1, j) - \mathcal{X}(i_2, j) \leq 1 - \mathcal{Y}(i_1, i_2) \\
(5.10.5) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, d(i,j)\mathcal{X}(i,j) \leq r \\
(5.10.6) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, |V_i - V_j|\mathcal{X}(i,j) \leq V_{Max} \\
(5.10.7) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, |D_i - D_j|\mathcal{X}(i,j) = 0 \\
(5.10.8) \quad \forall i \in \mathcal{CV}, \sum_{j \in \mathcal{MG}} \mathcal{X}(i,j) = 1 \\
(5.10.9) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i,j) \in \{0,1\} \\
(5.10.10) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \mathcal{Y}(i_1, i_2) \in \{0,1\} \\
(5.10.11) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \mathcal{Y}(i_1, i_2) = \mathcal{Y}(i_2, i_1)
\end{cases}
\tag{5.10}
$$

In the next section, a solution based on a priori articulation of preferences is proposed to solve our problem.

Weighted-Sum Approach

The integer program is solved using the Weighted-Sum Approach (WSA) as presented as follows:

$$\begin{cases} (5.11.1) \quad \textbf{min} \quad \alpha \sum_{i \in \mathcal{CV}} \sum_{j \in \mathcal{MG}} (1 - \mathcal{X}(i,j)) + \beta \sum_{i_1 \in \mathcal{CV}} \sum_{i_2 \in \mathcal{CV}} \omega_{i_1} \mathcal{Y}(i_1, i_2) \\ \textbf{s. t.} \\ (5.11.2) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i_1,j) + \mathcal{X}(i_2,j) \leq 1 + \mathcal{Y}(i_1,i_2) \\ (5.11.3) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i_1,j) - \mathcal{X}(i_2,j) \leq 1 - \mathcal{Y}(i_1,i_2) \\ (5.11.4) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, d(i,j)\mathcal{X}(i,j) \leq r \\ (5.11.5) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, |V_i - V_j| \mathcal{X}(i,j) \leq V_{Max} \\ (5.11.6) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, |D_i - D_j| \mathcal{X}(i,j) = 0 \\ (5.11.7) \quad \forall i \in \mathcal{CV}, \sum_{j \in \mathcal{MG}} \mathcal{X}(i,j) = 1 \\ (5.11.8) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i,j) \in \{0,1\} \\ (5.11.9) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \mathcal{Y}(i_1,i_2) \in \{0,1\} \\ (5.11.10) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \mathcal{Y}(i_1,i_2) = \mathcal{Y}(i_2,i_1) \end{cases} \tag{5.11}$$

To solve the integer program in Equation (5.11), the two objective functions must be arranged in order of importance. We define four weighted-sum methods with different preferences as detailed in Table 5.6. α and β take different values according to the importance of each objective function such that $\alpha + \beta = 1$. WSA1 solution minimizes only the first objective. Meanwhile, WSA2 solution aims at reducing only the second objective. With these two solutions, the worst values of the first objective and the second objective functions are obtained and are, respectively, $Obj1_{WORST}$ and $Obj2_{WORST}$. The WSA3 solution takes the first objective as a priority. Meanwhile, the WSA4 solution takes objective 2 as the first choice. These two solutions aim to find a compromise between the different objectives. However, in the next section, we propose another trade-off solution without articulation of preferences.

TABLE 5.6: Weighted-sum methods

	α	β
WSA1	1	0
WSA2	0	1
WSA3	0.7	0.3
WSA4	0.3	0.7

Game Theory Approach. Game theory approach (GTA) is a trade-off solution which is applied to find a compromise between the first objective and the second one. To do so, we use the Nash bargaining approach and, the two objectives are considered as two players. The game is non-cooperative since each player aims at minimizing its objective without knowing the state of the other player. The first player seeks an objective value not better than $f^*(\mathcal{X}, \mathcal{Y})$. On another hand, the second player should not wait for value for his objective better than $g^*(\mathcal{X}, \mathcal{Y})$ (See Equation (5.12) and Equation (5.13)). However, it must be guaranteed that the two objective values should not be worse than $Obj1_{WORST}$ and $Obj2_{WORST}$, respectively for the player 1 and player 2.

$$\sum_{i \in \mathcal{CV}} \sum_{j \in \mathcal{MG}} (1 - \mathcal{X}(i,j)) \leq f^*(\mathcal{X}, \mathcal{Y}) \tag{5.12}$$

$$\sum_{i_1 \in \mathcal{CV}} \sum_{i_2 \in \mathcal{CV}} \omega_{i_1} \mathcal{Y}(i_1, i_2) \leq g^*(\mathcal{X}, \mathcal{Y}) \tag{5.13}$$

In Nash bargaining theory, we aim to find an optimal point taking into account reference values which are the worst obtained values of the objectives such that:

$$f^*(\mathcal{X}, \mathcal{Y}) \leq Obj1_{WORST} \tag{5.14}$$

$$g^*(\mathcal{X}, \mathcal{Y}) \leq Obj2_{WORST} \tag{5.15}$$

The trade-off between the two objectives is obtained through Equation (5.12), Equation (5.13), Equation (5.14), Equation (5.15), and the following integer program:

$$\begin{cases} (5.16.1) \quad \textbf{max} \ (Obj1_{WORST} - f^*(\mathcal{X}, \mathcal{Y})) \times (Obj2_{WORST} - g^*(\mathcal{X}, \mathcal{Y})) \\ \qquad\qquad\qquad \textbf{s. t.} \\ (5.16.2) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i_1,j) + \mathcal{X}(i_2,j) \leq 1 + \mathcal{Y}(i_1,i_2) \\ (5.16.3) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i_1,j) - \mathcal{X}(i_2,j) \leq 1 - \mathcal{Y}(i_1,i_2) \\ (5.16.4) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, d(i,j)\mathcal{X}(i,j) \leq r \\ (5.16.5) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, |V_i - V_j|\mathcal{X}(i,j) \leq V_{Max} \\ (5.16.6) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, |D_i - D_j|\mathcal{X}(i,j) = 0 \\ (5.16.7) \quad \forall i \in \mathcal{CV}, \sum_{j \in \mathcal{MG}} \mathcal{X}(i,j) = 1 \\ (5.16.8) \quad \forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, \mathcal{X}(i,j) \in \{0,1\} \\ (5.16.9) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \mathcal{Y}(i_1,i_2) \in \{0,1\} \\ (5.16.10) \quad \forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \mathcal{Y}(i_1,i_2) = \mathcal{Y}(i_2,i_1) \end{cases} \tag{5.16}$$

After we formulated our problem as an IOP where the objective functions and the constraints are linear, we propose a COP model in the next section.

Constraint Optimization Problem

Problem statement. In this section, we develop the gateway selection problem model using Constraint Programming. The primary process consists of first defining the variables and their corresponding domains and then, determining the constraints of the problem. If some criterion is to be optimized, the objective functions need to be specified. We model our problem as a COP as follows:

- A finite set of variables: $X = \{\mathcal{X}, \mathcal{Y}\}$.
  Where:
  $\mathcal{X}(\mathcal{CV}, \mathcal{MG})$ is a binary matrix. When the CV i is connected to the MG j, then $\mathcal{X}(i,j) = 1$, otherwise $\mathcal{X}(i,j) = 0$. $\mathcal{Y}(\mathcal{CV}, \mathcal{CV})$ is a binary and symmetric matrix. When $i_1 \in \mathcal{CV}$ and $i_2 \in \mathcal{CV}$ are connected to the same MG, then $\mathcal{Y}(i_1, i_2) = 1$, otherwise $\mathcal{Y}(i_1, i_2) = 0$.

- A nonempty domain of possible values for each variable: $DOM(X) = D_{\mathcal{X}} = D_{\mathcal{Y}} = \{0, 1\}$

- A finite set of constraints:

  (C1). $\forall i_1, i_2 \in \mathcal{CV}^2, \forall j \in \mathcal{MG}$: $\mathcal{Y}(i_1, i_2) = 0 \implies (\mathcal{X}(i_1, j) = 0) \vee ((\mathcal{X}(i_2, j) = 0)$;

  (C2). $\forall i_1, i_2 \in \mathcal{CV}^2, \forall j \in \mathcal{MG} : \mathcal{Y}(i_1, i_2) = 1 \implies \mathcal{X}(i_1, j) = \mathcal{X}(i_2, j)$;

  (C3). $\forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, d(i,j)\mathcal{X}(i,j) \leq r$;

  (C4). $\forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, |V_i - V_j|\mathcal{X}(i,j) \leq V_{Max}$;

  (C5). $\forall i \in \mathcal{CV}, \forall j \in \mathcal{MG}, |D_i - D_j|\mathcal{X}(i,j) = 0$;

  (C6). $\forall i \in \mathcal{CV}$: $\sum_{j \in \mathcal{MG}} \mathcal{X}(i,j) = 1$;

  (C7). $\forall i_1 \in \mathcal{CV}, \forall i_2 \in \mathcal{CV}, \mathcal{Y}(i_1, i_2) = \mathcal{Y}(i_2, i_1)$;

- The objectives are:

(Obj1). **min** $\sum_{i \in \mathcal{CV}} \sum_{j \in \mathcal{MG}} (1 - \mathcal{X}(i,j))$

(Obj2). **min** $\sum_{i_1 \in \mathcal{CV}} \sum_{i_2 \in \mathcal{CV}} \omega_{i_1} \mathcal{Y}(i_1, i_2)$

The constraints and the objectives in the COP model are described as follows:

- (C1) ensures that if $\mathcal{Y}(i_1, i_2) = 0$, i_1 and i_2 must not connect to the same MG.

- (C2) ensures that if $\mathcal{Y}(i_1, i_2) = 1$, i_1 and i_2 must connect to the same MG.

- (C3),(C4),(C5),(C6) and (C7) are the same as in the integer program in Equation (5.10).

- (Obj1) and (Obj2) are the same as in the integer program in Equation (5.10)

To deal with this model, we propose hereafter a Constraint Optimization Problem solution. Backtracking: Constraint Optimization Problem Solution. There are mainly three solution strategies for solving a COP namely backtracking, dynamic programming, and local search. These algorithms are classified into two categories: complete and incomplete. Local search represents an incomplete algorithm which tries to find a solution if it exists and an approximation to the optimal solution. On another hand, backtracking and dynamic programming are complete algorithms which can be used to prove that the problem has no solution or to find an optimal solution. Dynamic programming requires an exponential execution time to find all solutions while backtracking works on only one solution at a time and require a polynomial time of execution Rossi et al. (2006). In what follows, we use Backtracking to solve the COP ensuring a minimal execution time. Backtracking algorithm consists of assigning values to variables, one by one, traversing through the domains such that the constraints are satisfied. The solution is represented by a vector containing all the assigned values of the variables. At each step where the constraints related to the concerned variables are not met, a return backward is carried out hence the name of backtracking. The problem is solved in a depth-first manner of the space to find a solution. Assuming that *Vector* is a solution to our COP by applying Backtracking, the steps of the generalized algorithm may be resumed in Algorithm 7. A starting point (i.e., a variable) is chosen to implement backtracking to our problem, *Choose Variable* function helps to start with one possible variable of many available variables in X. *Backtrack* function traverses all the variables recursively, from the root down, in depth-first order as mentioned before. At each variable v, the function checks whether the value $s \in DOM(X)$ can satisfy all the constraints and construct a valid solution, if yes the algorithm proceeds otherwise a reversal is performed, until *Vector* contains all instantiated values of all variables. *Accept* function returns true if s is a solution and false otherwise. *Root* function returns the root of a variable v while *Next* function returns the next candidate.

However, *Choose Variable* and *Next* functions play a crucial role in the progress of the algorithm. Indeed, the start of the procedure has an impact on the solution, and several methods have been proposed for ordering variables as an improvement to Backtracking strategy. In Boussemart et al. (2004), a fruitful dynamic and adaptive variable ordering heuristic is proposed which is a combination of look-back and look-ahead schemes. This heuristic derives benefit from information about previous states of the search process. To do so, a weight is

Algorithm 7 Generalized Backtracking Algorithm

```
1: v ← CHOOSE VARIABLE(X = {𝒳, 𝒴})
2: Vector ← ∅
3: function BACKTRACK(Vector,v)
4:     while length(Vector) ≠ length(X) do
5:         if ACCEPT(Vector ∪ s) then
6:             Vector ← Vector ∪ s
7:             v ← NEXT(v)
8:             BACKTRACK(Vector,v)
9:         else
10:            v ← ROOT(v)
11:            Backtrack(Vector,v)
12:    return Vector
```

associated with each constraint, and it increases whenever the associated constraint is violated during the search. On another hand, the notion of the impact of a variable is introduced in Refalo (2004). The impact represents the importance of a variable for the reduction of the search space and using this concept the performance is improved. Indeed, measuring the impact with the observation of domain reduction during search proves to be a useful criterion for choosing variables. Based on the concept of the weight, each constraint $C(\mathcal{V})$ linked to a set of variables $\mathcal{V}$ has a weight $\psi(C)$, and the variable v is chosen using the domain size as extracted from Hebrard (2008) as follows:

$$min \frac{|D(v)|}{\sum_{v \in \mathcal{V}} \psi(C(\mathcal{V}))} \tag{5.17}$$

Based on the second strategy, we assume that $\mathcal{I}(v = i)$ is the impact of the decision $V = i$ as mentioned in Hebrard (2008). The variable v is chosen as extracted from Hebrard (2008) as follows:

$$min \frac{\sum_{i \in D(v)} 1 - \mathcal{I}(v = i)}{\sum_{v \in \mathcal{V}} \psi(C(\mathcal{V}))} \tag{5.18}$$

To find an optimal solution the approach is to solve a sequence of satisfaction problems using the described backtracking algorithm. For the sake of effectiveness, AC3 algorithm is used to reduce the domains of the variables Mackworth (1977). In the next section, the proposed MOO system for gateways selection is detailed.

Multi-Objective Optimization System for Gateways Selection

An interactive MOO system for gateways selection is proposed to support decision making. This system helps a decision maker to select the best solution among a set of alternatives and is based on the models mentioned above. To select MGs, a decision maker can express his preferences regarding maximizing the number of connected CVs and minimizing the amount of traffic handled by MGs. These two objectives are conflicting, and the preferences can be set according to some high-level information that includes all information concerning the vehicular ad-hoc network. The proposed system is depicted in Figure 5.3, it helps guiding the decision maker to explore the solutions with best match with his preferences through these three phases: *(i)* Construction phase; *(ii)* Resolution phase; *(iii)* Visualization phase. In the first phase, the decision maker can interact with the system by constructing the problem, he adds or removes the constraints and sets his preferences. It can be a priori articulation of preferences, in this case, WS approach is used in the second phase or a posteriori articulation of preferences where all approaches are used. On another hand, if there is no articulation of preferences, GT and COP approaches are executed. In the resolution phase, the approaches are implemented and executed using a multi-objective optimizer. The high-level information represents the state of the environment containing CVs and MGs. In the last phase, the decision maker can study the problem using the projection of the possible solution according to his previous choice. He can choose one solution using high-level information with the help of 2D and 3D diagrams. This step may be considered as a posteriori articulation of preferences where the Pareto solutions are visualized according to the first objective, the second objective and the execution time. According to these three criteria, all approaches are compared using the comparison methods mentioned in Section 2.3.

5.2.3 Implementation and Results

In this section, the proposed approaches are evaluated through simulation. The MOO system for gateways selection is implemented using Python programming language. Gurobi Optimizer Optimization (2014) is executed to solve GT and WS approaches whereas Mistral library Hebrard (2008) is used for solving COP. Both of these tools prove to be highly efficient. In the simulation, the network containing the CVs and the MGs is randomly deployed, and the simulation parameters are shown in Table 5.7. We generate a random infrastructure in both x-axis and y-axis, for reason of simplicity, we consider that z=0. We evaluate the performance

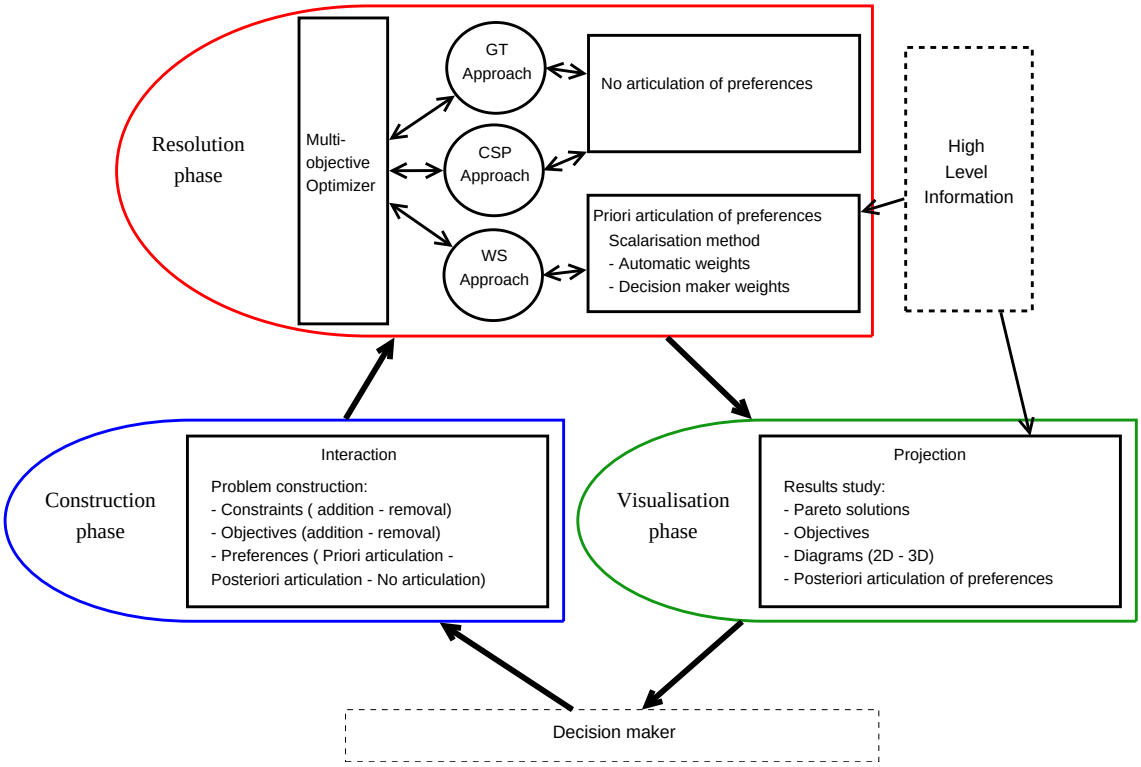


FIGURE 5.3: Multi-Objective Optimization System for Gateways Selection

of the proposed models (i.e., WSA1, WSA2, WSA3, WSA4, GTA, and COP) compared to the literature solution (i.e., The predicted link lifetime Namboodiri and Gao (2007)) which we name PLET. The evaluation is performed according to the following metrics:

- Objective 1 which must be maximized and concerns the number of connected CVs;
- Objective 2 which must be minimized and concerns the average of traffic amount handled by MGs;
- The execution time.

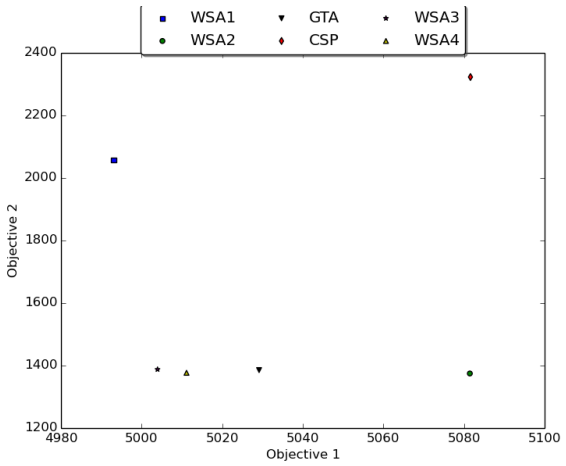
At first, we set the number of CVs to 50, and the number of MGs to 30 while V_{Max} and r remain fixed. Afterward, in the second simulation, the approaches are evaluated and compared with PLET solution by varying the number of MGs while the number of CVs is fixed to 50 and by changing the number of CVs while the number of MGs is fixed to 50.

TABLE 5.7: Simulation parameters

Parameter	Setting
Direction	$0 \leq \theta \leq 2\pi$
Velocity	0-20 m/s
X-coordinate	0-2000 m
Y-coordinate	0-2000 m
Z-coordinate	0 m
Transmission range	500 m

A case study

In this simulation, all approaches are executed for the same case study. The number of CVs is 50, and the number of MGs is 30. Figure 5.4 and Figure 5.5 represent the projection phase of our case study using all approaches. Figure 5.4(a) shows a 2D diagram where the x-axis represents objective 1, and the y-axis represents objective 2. The methods comparison of this diagram is depicted in Figure 5.4(b), WSA1, and WSA2 are incomparable, the first approach shows the best value concerning the primary objective while the second approach shows the best value concerning objective 2, WSA1 is also incomparable to WSA3, WSA4, and GTA. WSA2 is incomparable to WSA3, WSA4, and GTA. WSA3 is incomparable to WSA4 and GTA. WSA4 is incomparable to GTA, and finally, all approaches strictly dominate COP. In Figure 5.5(a), the 3D diagram is depicted, the x-axis represents the objective 1, the y-axis represents the objective 2, and the z-axis represents the time execution. Figure 5.5(b) shows the output of this diagram methods comparison, as in Figure 5.4(b), WSA1, and WSA2 are incomparable regarding the three metrics. In this case, all methods are incomparable to each other, and WSA1 strictly dominates COP. Otherwise, COP exhibits an excellent performance regarding the execution time.

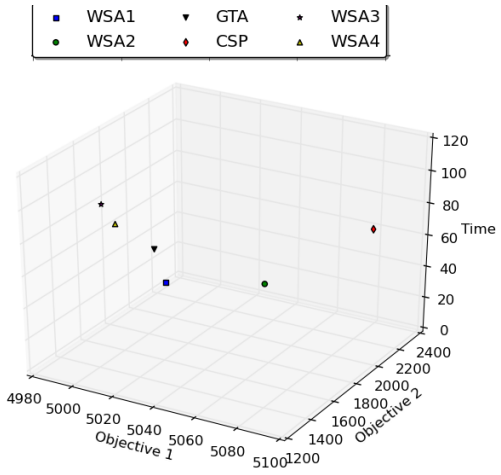


(a) 2D diagram

```
Methods Comparison 2D diagram :
WSA1 || WSA2      GTA >> COP
WSA1 || GTA       GTA >COP
WSA1 >> COP       GTA ⩾COP
WSA1>COP          GTA|| WSA3
WSA1⩾COP          WSA4 || GTA
WSA1 || WSA3      WSA3 >> COP
WSA1 || WSA4      WSA3>COP
WSA2 || GTA       WSA3⩾COP
WSA2 >> COP       WSA4 >> COP
WSA2>COP          WSA4>COP
WSA2⩾COP          WSA4⩾COP
WSA2 || WSA3      WSA3 || WSA4
WSA2 || WSA4
```

(b) 2D comparison output.

FIGURE 5.4: The system solutions projection using all approaches for objective 1 and objective 2.



(a) 3D diagram

```
Methods Comparison 3D diagram :
WSA1 || WSA2
WSA1 || GTA
WSA1 >> COP
WSA1>COP
WSA1⩾COP
WSA1 || WSA3
WSA1 || WSA4
WSA2 || GTA
WSA2 || COP
WSA2 || WSA3
WSA2 || WSA4
GTA|| COP
GTA|| WSA3
GTA|| WSA4
COP || WSA3
COP || WSA4
WSA3 || WSA4
```

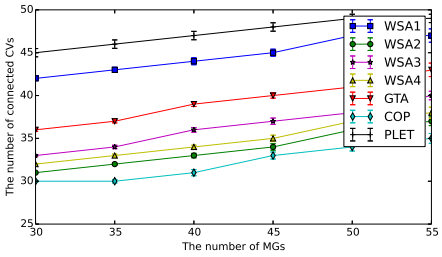
(b) 3D comparison output.

FIGURE 5.5: The system solutions projection using all approaches for objective 1, objective 2, and the execution time.

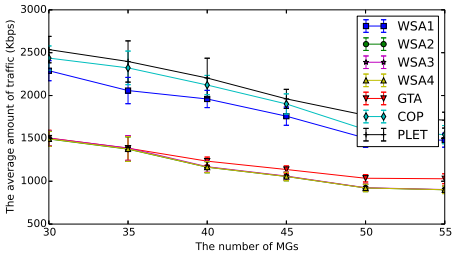
5.2.4 The proposed approaches evaluation

In this simulation, we evaluate the proposed approaches implemented in the MOO system for gateways selection, and we evaluate the literature solution PLET. To do this, as mentioned earlier, we vary the number of MGs and fix the number of CVs; likewise, we change the number of CVs and fix the number of MGs. In

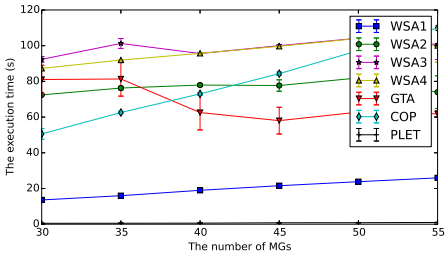
this simulation results, each plotted point represents the average of 20 times of executions. The plots are presented with 95 confidence interval. In each execution, V_{Max} and r remain unchanged, so we vary all the other parameters. Figure 5.6 shows the approaches evaluation using Mistral and Gurobi while changing the number of MGs from 30 to 55. The number of CVs is fixed to 50. Figure 5.6(a) represents the objective 1 which increases while the number of MGs becomes high. This is due to the number of choices that increases. We notice that WSA1 shows the best performance concerning the number of connected vehicles after the solution PLET which connects all CVs except those out of range. Figure 5.6(b) represents the average of the traffic amount of MGs, by varying the number of MGs the objective 2 decreases. WSA2 exhibits the best performance concerning this parameter, and PLET shows the worst performance. Figure 5.6(c) shows the execution time of the proposed approaches and PLET solution. We notice that in general, it increases while the number of MGs grows. As depicted in this figure, PLET exhibits the best performance since there is no optimization. WSA1 shows the best value of execution time compared to the other approaches. Figure 5.7 represents the evaluation of the proposed approaches and PLET solution while varying the number of CVs from 30 to 55 and fixing the number of MGs to 50. The graph of objective 1 is shown in Figure 5.7(a), as in Figure 5.6(a), WSA1 shows the best performance after the solution PLET. Moreover, the objective 1 increases while the number of CVs becomes high. In Figure 5.7(b), objective 2 is represented, and it increases by varying the number of CVs. Again, WSA2 exhibits the best performance and PLET has the worst values. Figure 5.7(c) represents the execution time, and again, PLET exhibits the best performance concerning time. This simulation demonstrates the efficiency of each proposed model in achieving its key design goals compared to PLET solution; it is noticeable that WSA1 shows the best results regarding maximizing the number of connected CVs while WSA2 shows the best values regarding minimizing the amount of traffic handled by each gateway. WSA3, WSA4, and GTA reach them goals by finding a compromise between the two objectives, unlike COP that shows only better execution time in this case study. Table 5.8 displays the improvement percentage of objective 2 while maximizing the number of connected vehicles for the first simulation where we vary the number of MGs. The best reduction percentage regarding objective 1 is provided by WSA1 where the number of connected vehicles is reduced only to 5,63%, and the worst one is provided by COP by reducing the number to 32,04%. For objective 2, the best reduction percentage is performed by WSA2 which is 49,07% while COP has the worst one which is 6,42%. Table 5.9 represents



(a) The number of connected CVs.



(b) The average amount of traffic handled by MGs.



(c) The execution time.

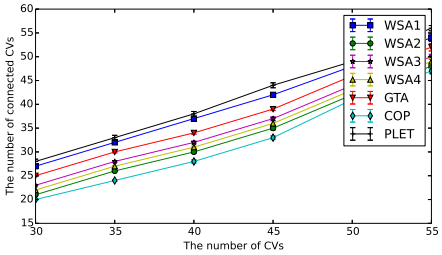
FIGURE 5.6: Approaches performance using Mistral and Gurobi while varying the number of MGs.

the improvement percentage of objective 2 while maximizing the number of connected vehicles for the second simulation where we vary the number of CVs. The percentage is 3,22% for objective 1 and is provided by WSA1 while COP has the worst percentage which is 22,17%. The best reduction value concerning objective 2 is performed by WSA2 and is equal to 49,07% while COP exhibits the worst percentage value which is 6,42%. In both simulations, all approaches improve the objective 2 by up to 49,07% while maximizing the first one by up to 3,04% of the difference to PLET solution.

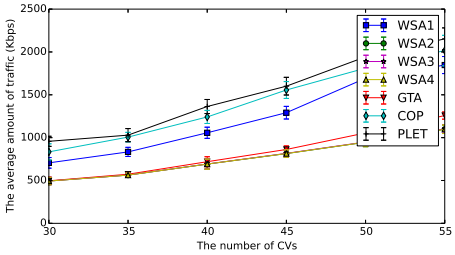
Finally, it is worth stressing out that the proposed approaches demonstrate its efficiency in helping a decision maker to realize the adequate mapping of CVs and MGs. Indeed with an articulation of preferences or without, the results prove that the solutions find the fair trade-off between maximizing the number of connected vehicles and minimizing the average amount of traffic handled by MGs.

The impact of the transmission range et the permitted velocity difference

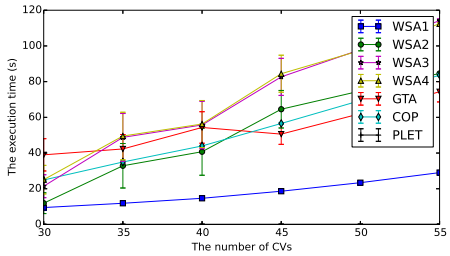
In this simulation, we consider two scenarios. In the first one, the number of CVs is 50, and the number of MGs is 30. In the second scenario, the number of CVs is the same, and the number of MGs is 50. We vary the transmission range r



(a) The number of connected CVs.

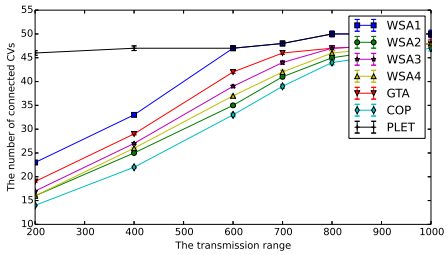


(b) The average amount of traffic handled by MGs.

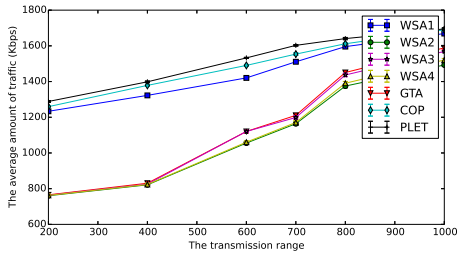


(c) The execution time.

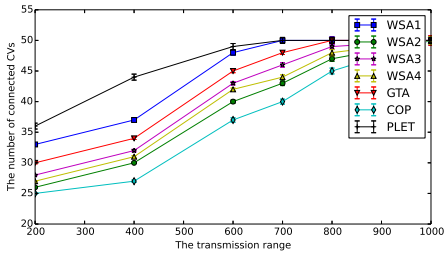
FIGURE 5.7: Approaches performance using Mistral and Gurobi while varying the number of CVs.



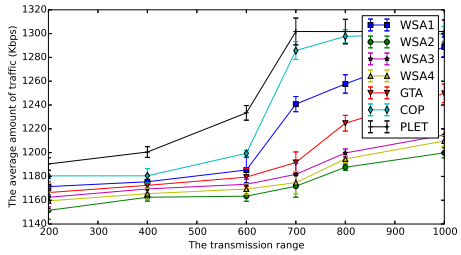
(a) Scenario 1: The number of connected CVs.



(b) Scenario 1: The average amount of traffic handled by MGs.



(c) Scenario 2: The number of connected CVs.



(d) Scenario 2: The average amount of traffic handled by MGs.

FIGURE 5.8: The impact of the transmission range.

TABLE 5.8: The average amount of traffic handled by MGs improvement while maximizing the number of connected CVs and varying the number of MGs.

	COP	WSA1	WSA2	WSA3	WSA4	GTA
Objective 1	32,04 % ↓	5,63% ↓	28,52% ↓	23,23% ↓	26,40% ↓	16,60% ↓
Objective 2	5,18% ↓	12,18% ↓	45,08% ↓	44,78% ↓	45,03% ↓	41,78% ↓

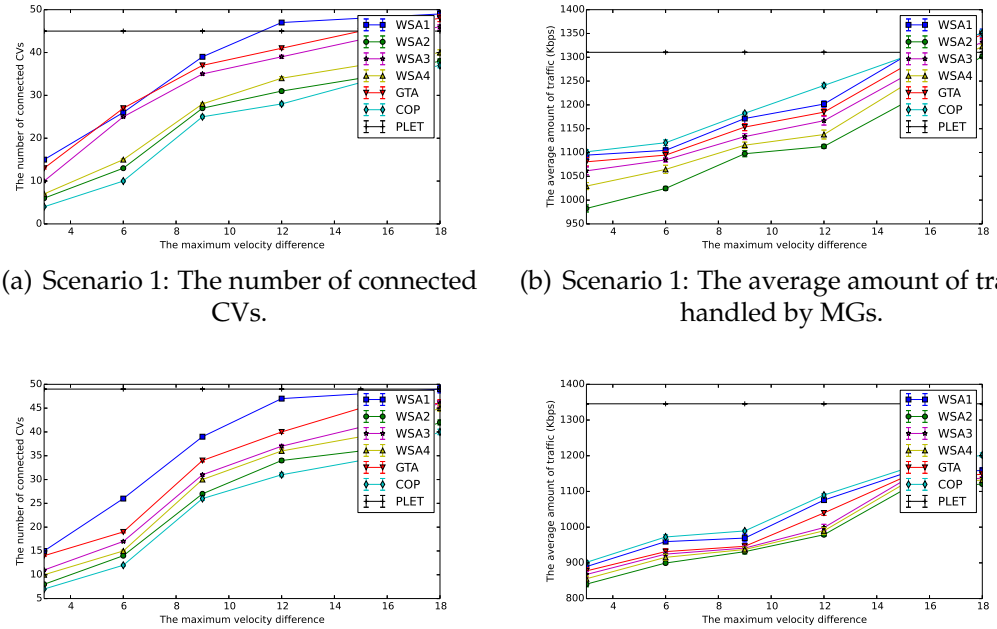
TABLE 5.9: The average amount of traffic handled by MGs improvement while maximizing the number of connected CVs and varying the total number of CVs.

	COP	WSA1	WSA2	WSA3	WSA4	GTA
Objective 1	22,17% ↓	3,22% ↓	18,54% ↓	13,70% ↓	16,12% ↓	8,87% ↓
Objective 2	6,42% ↓	17,89% ↓	49,07% ↓	48,94% ↓	49,06% ↓	45,12% ↓

and the permitted velocity difference V_{Max} to study its impact on the number of connected CVs and the average amount of traffic handled by MGs. Each plotted point represents the average of 20 times of executions. The plots are presented with 95 confidence interval. Figure 5.8 displays the impact of the transmission range. We notice that the number of connected CVs become higher when we increase the transmission range in both scenarios. Likewise, the average amount of traffic increases when the transmission range rises. However, when the number of MGs is large, the average amount of traffic is not as high as in the first scenario (See Figure 5.8(b) and Figure 5.8(d)). Figure 5.9 presents the impact of the permitted velocity difference. Since PLET solution does not depend on this measurement, the result stills the same. Regarding other solutions, the number of connected vehicles and the average amount of traffic handled by MGs increases while we vary the permitted velocity difference between CVs and MGs. Furthermore, the two metrics are more increasing in the first scenario. We notice that the transmission range and the permitted velocity difference have a significant impact on the number of connected vehicles and the average amount of traffic handled by MGs for all the proposed approaches.

5.2.5 Conclusion and limitations

In this work, a multi-objective optimization system for mobile gateways selection is proposed to improve the gateways discovery system. This latter is assisted by cloud computing and provides to vehicles in need to access to Internet an appropriate gateway. Indeed, research proves the efficiency of mobile gateways in



(a) Scenario 1: The number of connected CVs.

(b) Scenario 1: The average amount of traffic handled by MGs.

(c) Scenario 2: The number of connected CVs.

(d) Scenario 2: The average amount of traffic handled by MGs.

FIGURE 5.9: The impact of the permitted velocity difference.

comparison to the fixed ones which are road infrastructure. Consequently, a suitable selection of gateways for vehicles in need of Internet access must be carried on. Our proposed system consists of different models namely Integer Optimization and Constraint Optimization providing to the decision maker a palette of choices according to his preferences. This system also helps the decision maker by projecting results in 2D and 3D diagrams and using some comparative methods. The simulations show the efficiency of the system in comparing the different solving strategies, the effectiveness of the weighted sum method in the case of a priori articulation of preferences, the game theory approach in finding a trade-off between the conflicting objectives and the efficiency of backtracking algorithm in term of execution time. These approaches are compared with the literature solution and prove to be effective in minimizing the traffic amount handled by each gateway while maximizing as much as possible the number of connected client vehicles. We intend on developing the system by integrating more constraints and objectives. We also aim at strengthening the system by making it autonomous. Indeed, without the intervention of the decision maker, the system may be more adaptive to the vehicular ad-hoc network environment by proposing the best strategy for gateways selection and an automatic priori-articulation of preferences.

Chapter 6

Conclusion and Prospects

6.1 Conclusion

The 5G promises to push the limits concerning capacity and speed. This architecture promises ultra-fast connectivity, fluid fixed-mobile convergence and broad coverage ensuring continuity and better quality of user experience across all environments. It will enable the launch of vital real-time services of high reliability such as industrial pilot robots and remote health assistance. This future network will be designed for a hyper-connected world that integrates expanding services such as the Internet of Things, the connected vehicles, virtual and augmented reality and will revolutionize the use of businesses, communities, and individuals. The 5G architecture is built on the concept of virtualization of network functions thanks to Software Defined Networking, Network Functions Virtualization, and Cloud Computing techniques. Furthermore, it will offer a lot of agility in the configuration of innovative solutions, no longer carried out on each hardware element of the network, but by centralized software functions. It will considerably reduce the time required to create a new service that operators can implement, in a very flexible and responsive way, across a set of vertical markets (e.g., transport, industry, healthcare, smart cities, etc.) through network slicing. Virtualization solutions are already available at the core network (EPC)level. At the radio access network (RAN) level, the C-RAN (Cloud RAN) virtualizes and centralizes the signal processing functions at one point. It supports resource sharing by allocating them dynamically on each radio site distributed over a given area and reduces operating costs. This architecture can be deployed in areas well served by optical fiber, its equipment can communicate via fiber given the high expected speeds. These various developments will contribute to improving the energy efficiency of the network, thus reducing operator operating costs (OPEX). Therefore, mobile network telecommunications and vehicular networks markets take advantages of virtualization techniques and enhance the quality of service.

In this thesis, two major axes are addressed and are essential for the future challenges of 5G architecture. The virtual network functions placement in the mobile network communication of the future, and the mobile gateways selection in the vehicular ad-hoc network are two problems of resource management. 5G architecture requires intelligent agents to manage routing, quality of experience, and resource allocation. In one hand, in mobile network communication, virtual network functions need an effective placement through geographically distributed data centers given some constraints and some conflicting objectives. On another side, in the vehicular ad-hoc network, adequate mobile gateways must be carried for vehicles in need of internet access given some selection criteria and reaching some conflicting high-level objectives. Our work is a contribution to these two axes, and the deployed solutions are considered as resource controllers in cloud-based systems.

6.1.1 Virtual Network Functions Placement

To the best of our knowledge, the works cited in the state of the art of virtual network functions placement considering 3GPP standards do not present a real-time, autonomous, and adaptable to external factors solutions. First, we propose a solution to define S-GWs and P-GWs positions dynamically in given data centers, based on Constraint Satisfaction Problem (CSP) and an improved version of Forward-Checking, to ensure the requirements of users and services. This solution brings a real-time placement, unlike the game theory literature approach which needs the worst values firstly before finding a trade-off. We applied a CSP with different goals for the virtual machine placement problem running S-GW and P-GW, and this within the constraints required by the standards. The methods gave satisfactory results regarding their objectives, by reducing the number of the relocation of the S-GW, the number of virtual machines corresponding to P-GW, and also by optimizing the cost of UEs to access S-GW and P-GW, in order, to improve the quality of experience. The determination of the method depends on the nature of user behavior; we must reduce the number of relocation facing many mobile users and improve the cost of road otherwise. Indeed, user behavior plays a crucial role in optimizing the network configuration and the positioning of these network functions. Still that, a compromise of all those parameters is the best approach to find suitable DCs configurations. FCSMART offers the best trade-off between the cost and the number of instances. However, this solution needs two parameters to be defined by the decision maker.

The next contribution is virtual network functions placement system which is designed to have the maximum level of flexibility for meeting the operators' preferences and adjusting to the users' behavior. The system attains a fair solution considering the constraints conforming with the 3GPP standards which are decreasing Serving Gateways relocations cost and the cost of the path linking Packet Data Network Gateways and eNodeB base stations. Besides, the system aims at reducing the incurred cost of virtual machines. The proposed approach to realize the system solver is Constraint Programming. This solution brings a real-time virtual network functions placement system that can play the role of the resource controller. The main component of the system works according to some policies to obtain an adequate solution. Constraint Programming outperforms game theory approach in term of operation time and memory usage; although the two methods provide the same results in finding a trade-off solution to the conflicting objectives. The system is autonomous and runs in a cycle, however, to perform a policy, the placement depends on the three maximum values of the goals. An administrator depending on some statistics sets these three inputs.

Finally, in this work, we proposed a fuzzy controller to support virtual network functions placement and provide an adaptive solution to manage and organize the network. Our approach allows the solution to adapt to user equipment mobility and their needs in term of quality of experience. Furthermore, it minimizes Serving gateways relocation cost and the path between the user equipment and Packet data network gateways taking into account the resources capacities. The problem is based on conflicting objectives where a compromise must be found, and also depends on external factors relied on user equipment behavior and needs. In our proposed approach, the problem is formulated as a multi-objective optimization problem and solved using the weighted-sum method. We also introduce as first phase a fuzzy controller that provides fuzzy weights to the weighted-sum method. By doing this, the solution is adapted to the user equipment behavior (i.e., The mobility of user equipment) and their needs (i.e., The quality of experience depending on the used applications). Therefore, the objective of using fuzzy logic consists of taking into account various external factors to achieve a fair decision.

The contributions of the work on virtual network functions placement problem in 5G mobile network communication are summarized in Figure 6.1.

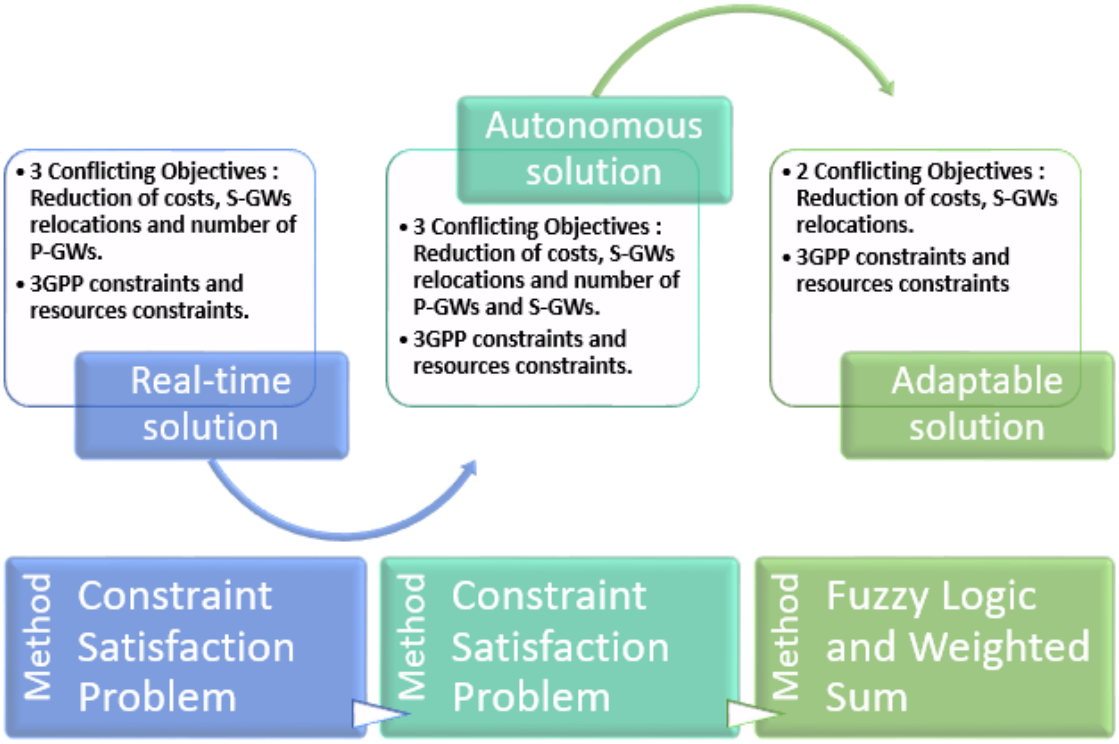


FIGURE 6.1: Virtual network functions placement contributions

6.1.2 Mobile Gateway Selection

In the initial work, we show the necessity for selecting a suitable gateway for vehicles without access to the Internet depending on several criteria and based on a gateway cloud-based discovery system. This solution is represented as a resource controller in a discovery server. The proposed solution advances a method of multiple-criteria decision analysis encouraging to find suitable gateways to our problem with decreasing the number of overloaded gateways. The selection of an appropriate gateway for all vehicles in need of internet access is analyzed by adopting the prescriptive approach of multi-criteria analysis PROMETHEE (Preference Ranking Organization Method for Enrichment Evaluations). It is a method of multiple-criteria decision analysis aiming at finding suitable gateways to our problem according to several criteria, by avoiding overloaded gateways and realizing a fair distribution (Load Balancing). PROMETHEE method is used for more refined modeling and to arrange all gateways from best to worst and get a relative valuation of each of them by giving a weight to each criterion depending on its importance. Our work concentrates on the range, distance, speed, direction and the number of clients using a gateway to minimize the number of overloaded gateways while connecting the maximum of clients. However, in this work, the selection is just a multi-criteria choice, and there is no optimization. On another hand, there is an obligation of using weights to find a trade-off. In the

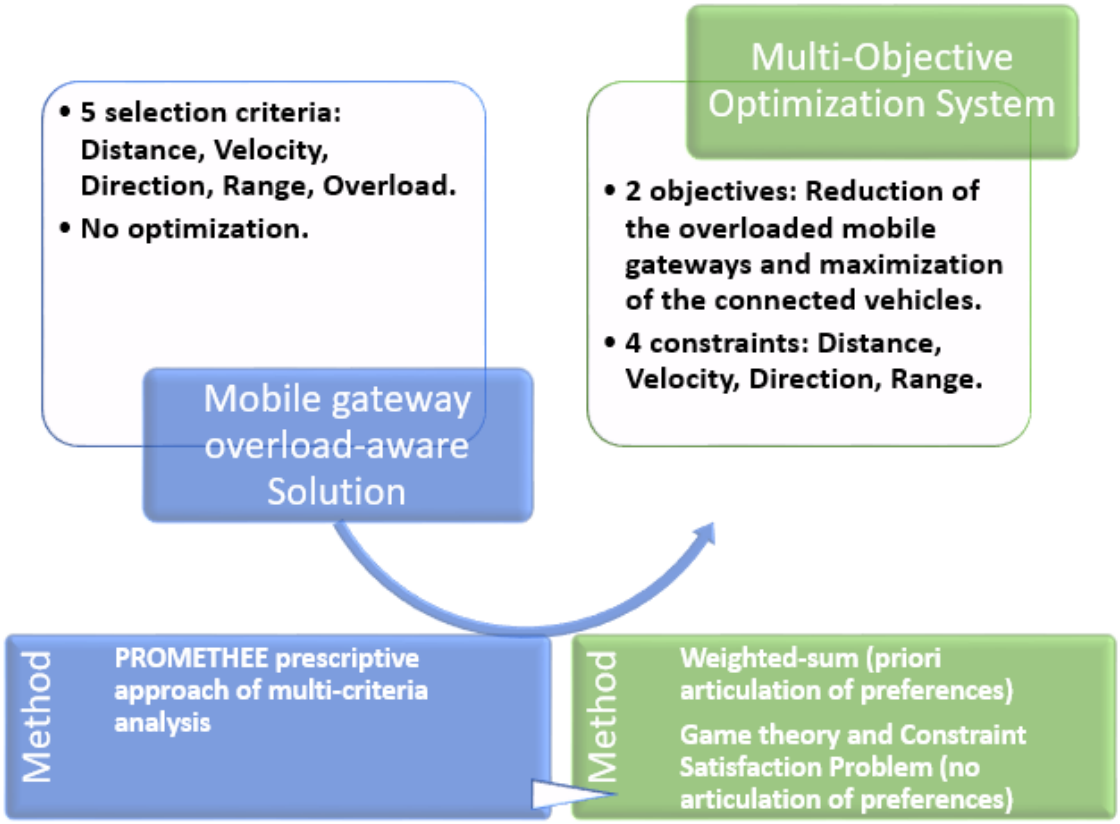


FIGURE 6.2: Mobile gateways selection in vehicular ad-hoc network contributions

last work concerning mobile gateways selection in a vehicular ad-hoc network, we advance the quality of selecting a mobile gateway by considering more constraints and objectives. Unlike the state of art solutions, we add some high-level objectives such as maximizing the number of connected vehicles and minimizing the traffic amount handled by mobiles gateways to avoid overload situations. The problem of selection is suited to be modeled using multiple conflicting objectives; hence, we employ multi-objective optimization to find a trade-off between the goals. In this solution, the decision maker will be in a better position to make a choice when such trade-off solutions are exposed. We propose three approaches to solve the multi-objective problem. The weighted-sum approach is used in the case of a priori articulation of preferences. Game theory and constraint programming approaches are used in the case of no articulation of preferences. These approaches are compared with the literature solution and prove to be efficient in reducing the traffic amount handled by each mobile gateway while increasing as much as possible the number of connected client vehicles.

Figure 6.2 summarizes the principal contributions in mobile gateways selection in the vehicular ad-hoc network.

6.2 Prospects

This section shows some research directions for future works. This thesis' attempts give satisfactory and encouraging results. However, we aim to advance better the proposed resource controller systems; This helps to consider some interesting prospects that are listed as follows:

- In the adaptive virtual network functions placement, further objectives to the problem could be included, among them, we cite reducing the cost paid by the operator by reducing the allocated virtual machines. We have to face the difficulty of defining complicated base rules with more than two inputs.
- In virtual network functions placement, more Evolved Packet Core components such as the Mobility Management Entity could be added to the system model besides the Serving Gateways and the Packet Data Network Gateways.
- We expect on improving the multi-objective system for mobile gateways selection in the vehicular ad-hoc network by integrating more constraints and objectives. We also intend at developing the system by making it autonomous. In other words, the system may be more adaptive to the vehicular ad-hoc network environment by proposing the best strategy for gateways selection and an automatic priori-articulation of preferences without the intervention of the decision maker.
- We also intend to develop a fuzzy controller for the system proposed for mobile gateways selection in the vehicular ad-hoc network. Depending on the nature of the network (i.e., the number of client vehicles and the traffic amount handled by each mobile gateway), the system must be adaptable to these external factors.
- And last, but not least, the critical prospect is to develop a joint autonomous, and adaptable system for both virtual network functions placement and mobile gateways selection problems. This system must be adjustable to both issues and every similar other problem by only changing the constraints and objectives. The efficiency of this system that plays the role of a resource controller could be studied in different application areas and compared.

6.3 Publications

In this section, the published articles are presented. The following publications involve international journals and international conferences. The articles published in international journals have been the outcome of the research presented in this thesis. Some of the international conferences issued articles concern this thesis directly, and some of these articles have been published during the preparation of this thesis on other projects.

6.3.1 International Journals

The published articles in the international journals are:

- Retal, S. and Idrissi, A. A Multi-Objective Optimization System for Mobile Gateways Selection in Vehicular Ad-Hoc Networks
  Computers & Electrical Engineering.
  https://doi.org/10.1016/j.compeleceng.2018.12.004

- Retal, S. and Idrissi, A. A Fuzzy Controller for an Adaptive VNFs Placement in 5G Network Architecture.
  International Journal of Computational Science and Engineering.
  https://doi.org/10.1504/IJCSE.2019.10018399

- Retal, S. and Idrissi, A. Virtual network functions placement system for 5g mobile network architecture.
  International Journal of Internet Technology and Secured Transactions.
  https://doi.org/10.1504/IJITST.2020.10018314

6.3.2 International Conferences

The published articles in the international conferences are:

- Retal, S. and Idrissi, A. Virtual Network Functions Placement in 5G Architecture: A Multi-Agent System Approach (To appear).
  The International Conference on Modern Intelligent Systems Concepts (MISC).

- Retal, S., Bagaa, M., Taleb, T., Flinck, H., Content delivery network slicing: QoE and cost awareness.
  IEEE International Conference on Communications (ICC)
  https://doi.org/10.1109/ICC.2017.7996499

- Laghrissi, A., Retal, S., Idrissi, A., S-GW and P-GW placement optimization using constraint programming.
  The International conference on Big Data and Advanced Wireless technologies (BDAW)
  `https://doi.org/10.1145/3010089.3010137`

- Idrissi, A., Retal, S., Rehioui, H., Laghrissi, A., Gateway selection in Vehicular Ad-hoc Network.
  The International Conference on Information Technology and Accessibility (ICTA)
  `https://doi.org/10.1109/ICTA.2015.7426937`

- Idrissi, A., Laghrissi, A., Retal, S., Rehioui, H., VANET congestion control approach using empathy.
  The International Conference on Information Technology and Accessibility (ICTA)
  `https://doi.org/10.1109/ICTA.2015.7426938`

- Idrissi, A., Rehioui, H., Laghrissi, A., Retal, S., An improvement of DENCLUE algorithm for the data clustering
  The International Conference on Information Technology and Accessibility (ICTA)
  `https://doi.org/10.1109/ICTA.2015.7426936`

Appendix A

Some used state of the art notions

A.1 Quality of Service vs Quality of Experience

In this thesis, both the Quality of service (QoS) and the Quality of Experience (QoE) terms are used in the literature. However, these two terminologies are not the same concept since QoE is used in mobile communications and more precisely in the virtual network functions problem state of the art, while the QoS is used in vehicular ad-hoc networks and the problem of mobile gateways selection related works. In what follows, the emphasis is put on these two terms and the difference between them.

Quality of service is the characterization or measure of the overall performance of a service, such as a mobile or a workstation network or a cloud computing service. To quantitatively measure the quality of service, diverse related perspectives of the network service are often studied, such as throughput, packet loss, bit rate, transmission delay, availability, jitter, and so on. In the field of mobile communications and telephony, this measurement was defined by the International Telecommunication Union in 1994. Quality of service involves requirements and specifications on all the features of a connection, such as service response time, loss, signal-to-noise ratio, interrupts, crosstalk, echo, loudness levels, frequency response, and so on.

Quality of Experience is a measurement of the delight or displeasure of a customer's experiences with a service (e.g., web browsing, streaming, phone call, TV broadcast). QoE concentrates on the entire service experience; it is a global notion, similar to the concept of User Experience, but with its version in telecommunication. This measurement was affirmed in 2016 by the International Telecommunication Union.

QoE has historically risen from Quality of Service, which tries to objectively include service parameters similar to packet loss rates or average throughput. QoS measurement is most often not associated with a customer, but to the media or network itself. QoE, nevertheless, is an entirely subjective measurement

from the user's point of view of the overall quality of the service furnished, by capturing people's artistic and hedonic needs.

A.2 PROMETHEE: The Preference Ranking Organization Method for Enrichment Evaluations

In this thesis, PROMETHEE (preference ranking organization method for enrichment evaluations) is used and is a family of multi-criterion decision support methods. PROMETHEE is a prescriptive approach to multi-criteria problem analysis with many actions (or decisions) evaluated according to several criteria. The prescriptive approach, called PROMETHEE, provides the decision maker with both a complete and partial ranking of the actions (or alternatives) to choose. PROMETHEE has been successfully used in many decision-making contexts around the world. A non-exhaustive list of scientific publications on extensions, applications, and discussions related to PROMETHEE methods has been published recently and presented in Behzadian et al. (2010).

PROMETHEE differs from ELECTRE[1] in that it constructs a valued upgrading relation reflecting a preference intensity. We can consider that this method is halfway between the over-classing approach and the MAUT methods whose methods of construction of the partial utility functions they use.
The PROMETHEE II is the version used in this thesis, and it has been chosen as the favored method for some selection process systems since its results are consistent, easy to understand, and require less information from decision makers compared to others methods such as AHP[2] as shown in Balali et al. (2014).

[1]ELECTRE is a family of non-compensatory methods of multi-criteria analysis. This method ELECTRE considers that the projects are not stable and not always comparable. Indeed, it is not still possible to define a strategy better than all the others in the absolute. In the framework of multicriteria methods analysis, the value given to a strategy is relative. It is a model of aggregation of preferences. In contrast to Anglo-Saxon multi-criteria analysis methods, which consist in aggregating then comparing the different criteria, the ELECTRE method and its derivatives compare and then aggregate them.

[2]The analytic hierarchy process (AHP) technique is a structured method for organizing and analyzing complex decisions, based on mathematics and psychology. It has distinct employment in group decision making and is used around the world in an extended variety of decision circumstances, in disciplines such as politics, business, commerce, healthcare, and education.

Bibliography

Agyapong, P. K., Iwamura, M., Staehle, D., Kiess, W., and Benjebbour, A. (2014). Design considerations for a 5g network architecture. *IEEE Communications Magazine*, 52(11):65–75.

Anderson-Cook, C. M. (2005). Practical genetic algorithms.

Andrews, R., Suriadi, S., Wynn, M. T., ter Hofstede, A. H., Pika, A., Nguyen, H. H., and La Rosa, M. (2016). Comparing static and dynamic aspects of patient flows via process model visualisations. *Information and Software Technology*.

Badole, M. H. and Raju, T. (2014). Protocol design for an efficient gateway discovery & dispatching for vehicular ad hoc network. *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, 3(2):117–120.

Badole, M. M. H. and Nikam, M. P. (2014). Performance evaluation of an efficient cloud-assisted gateway discovery for vehicular ad hoc network. *Performance Evaluation*, 1 (7)(2014).

Bagaa, M., Taleb, T., and Ksentini, A. (2014). Service-aware network function placement for efficient traffic handling in carrier cloud. In *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, pages 2402–2407. IEEE.

Balali, V., Zahraie, B., and Roozbahani, A. (2014). A comparison of ahp and promethee family decision making methods for selection of building structural system. *American Journal of Civil Engineering and Architecture*, 2(5):149–159.

Bari, M. F., Chowdhury, S. R., Ahmed, R., and Boutaba, R. (2015). On orchestrating virtual network functions. In *Network and Service Management (CNSM), 2015 11th International Conference on*, pages 50–56. IEEE.

Basta, A., Kellerer, W., Hoffmann, M., Hoffmann, K., and Schmidt, E.-D. (2013). A virtual sdn-enabled lte epc architecture: A case study for s-/p-gateways functions. In *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*, pages 1–7. IEEE.

Basta, A., Kellerer, W., Hoffmann, M., Morper, H. J., and Hoffmann, K. (2014). Applying nfv and sdn to lte mobile core gateways, the functions placement problem. In *Proceedings of the 4th workshop on All things cellular: operations, applications, & challenges*, pages 33–38. ACM.

Baumgartner, A., Reddy, V. S., and Bauschert, T. (2015). Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization. In *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, pages 1–9. IEEE.

Bechler, M., Wolf, L., Storz, O., and Franz, W. J. (2003). Efficient discovery of internet gateways in future vehicular communication systems. In *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, volume 2, pages 965–969. IEEE.

Behzadian, M., Kazemzadeh, R. B., Albadvi, A., and Aghdasi, M. (2010). Promethee: A comprehensive literature review on methodologies and applications. *European journal of Operational research*, 200(1):198–215.

Biere, A., Heule, M., and van Maaren, H. (2009). *Handbook of satisfiability*, volume 185. IOS press.

Biran, O., Corradi, A., Fanelli, M., Foschini, L., Nus, A., Raz, D., and Silvera, E. (2012). A stable network-aware vm placement for cloud systems. In *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on*, pages 498–506. IEEE.

Bitam, S., Mellouk, A., and Zeadally, S. (2015). Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks. *IEEE Wireless Communications*, 22(1):96–102.

Boussemart, F., Hemery, F., Lecoutre, C., and Sais, L. (2004). Boosting systematic search by weighting constraints. In *ECAI*, volume 16, page 146.

Brans, J.-P. and Vincke, P. (1985). Note—a preference ranking organisation method: (the promethee method for multiple criteria decision-making). *Management science*, 31(6):647–656.

Burke, E. K., Kendall, G., et al. (2005). *Search methodologies*. Springer.

Chee, B. J. and Franklin Jr, C. (2010). *Cloud computing: technologies and strategies of the ubiquitous data center*. CRC Press.

Clausen, J. (1999). Branch and bound algorithms-principles and examples. *Department of Computer Science, University of Copenhagen*, pages 1–30.

Clayman, S., Maini, E., Galis, A., Manzalini, A., and Mazzocca, N. (2014). The dynamic placement of virtual network functions. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–9. IEEE.

Cohen, R., Lewin-Eytan, L., Naor, J. S., and Raz, D. (2015). Near optimal placement of virtual network functions. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 1346–1354. IEEE.

Dantzig, G. B. (1990). *Origins of the simplex method*. ACM.

De Silva, C. W. (2018). *Intelligent control: fuzzy logic applications*. CRC press.

Dechter, R. and Frost, D. (2002). Backjump-based backtracking for constraint satisfaction problems. *Artificial Intelligence*, 136(2):147–188.

Dowsland, K. A. and Thompson, J. M. (2012). Simulated annealing. In *Handbook of natural computing*, pages 1623–1655. Springer.

el Mouna Zhioua, G., Labiod, H., Tabbane, N., and Tabbane, S. (2014). Fqgws: A gateway selection algorithm in a hybrid clustered vanet lte-advanced network: Complexity and performances. In *Computing, Networking and Communications (ICNC), 2014 International Conference on*, pages 413–417. IEEE.

Fonseca, C. M., Fleming, P. J., et al. (1993). Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In *Icga*, volume 93, pages 416–423.

Gent, I. P., Miguel, I., and Nightingale, P. (2008). Generalised arc consistency for the alldifferent constraint: An empirical survey. *Artificial Intelligence*, 172(18):1973–2000.

Gerla, M., Lee, E.-K., Pau, G., and Lee, U. (2014). Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 241–246. IEEE.

Ghaznavi, M., Khan, A., Shahriar, N., Alsubhi, K., Ahmed, R., and Boutaba, R. (2015). Elastic virtual network function placement. In *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*, pages 255–260. IEEE.

Gong, Z., Gu, X., and Wilkes, J. (2010). Press: Predictive elastic resource scaling for cloud systems. In *Network and Service Management (CNSM), 2010 International Conference on*, pages 9–16. Ieee.

Guerrero-Ibanez, J. A., Zeadally, S., and Contreras-Castillo, J. (2015). Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies. *IEEE Wireless Communications*, 22(6):122–128.

Gupta, A. and Jha, R. K. (2015). A survey of 5g network: Architecture and emerging technologies. *IEEE access*, 3:1206–1232.

Han, B., Gopalakrishnan, V., Ji, L., and Lee, S. (2015). Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 53(2):90–97.

Haralick, R. M. and Elliott, G. L. (1980). Increasing tree search efficiency for constraint satisfaction problems. *Artificial intelligence*, 14(3):263–313.

Haykin, S. (2004). A comprehensive foundation. *Neural networks*, 2(2004):41.

Hebrard, E. (2008). Mistral, a constraint satisfaction library. *Proceedings of the Third International CSP Solver Competition*, 3:3.

Hebrard, E., O'Mahony, E., and O'Sullivan, B. (2010). Constraint programming and combinatorial optimisation in numberjack. In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, pages 181–185. Springer.

Hock, D., Hartmann, M., Gebert, S., Jarschel, M., Zinner, T., and Tran-Gia, P. (2013). Pareto-optimal resilient controller placement in sdn-based core networks. In *Teletraffic Congress (ITC), 2013 25th International*, pages 1–9. IEEE.

Hussain, R., Son, J., Eun, H., Kim, S., and Oh, H. (2012). Rethinking vehicular communications: Merging vanet with cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pages 606–609. IEEE.

Hyser, C., McKee, B., Gardner, R., and Watson, B. J. (2007). Autonomic virtual machine placement in the data center. *Hewlett Packard Laboratories, Tech. Rep. HPL-2007-189*, 189.

Iera, A., Molinaro, A., Polito, S., and Ruggeri, G. (2009). A multi-layer cooperation framework for qos-aware internet access in vanets. *Ubiquitous computing and communication journal*, 4(3):1–10.

Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer.

Kelley, Jr, J. E. (1960). The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712.

Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.

Ksentini, A., Bagaa, M., and Taleb, T. (2016). On using sdn in 5g: the controller placement problem. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–6. IEEE.

Labiod, H., Tabbane, N., Tabbane, S., et al. (2012). An efficient qos based gateway selection algorithm for vanet to lte advanced hybrid cellular network. In *Proceedings of the 15th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 353–356. ACM.

Lan, K.-c., Kanhere, S., Setiwan, G., Iskandar, S., and Wu, Z. M. (2007). Feasibility study of using mobile gateways in public transportation vehicles for its applications. *V2VCOM*, 9.

Le, K., Bianchini, R., Zhang, J., Jaluria, Y., Meng, J., and Nguyen, T. D. (2011). Reducing electricity cost through virtual machine placement in high performance computing clouds. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, page 22. ACM.

Lin, Y.-W., Shen, J.-M., and Weng, H.-C. (2011a). Gateway discovery in vanet cloud. In *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*, pages 951–954. IEEE.

Lin, Y.-W., Shen, J.-M., and Weng, H.-J. (2011b). Cloud-assisted gateway discovery for vehicular ad hoc networks. In *Information Science and Service Science (NISS), 2011 5th International Conference on New Trends in*, volume 2, pages 237–240. IEEE.

Lodi, A. and Tramontani, A. (2013). Performance variability in mixed-integer programming. In *Theory Driven by Influential Applications*, pages 1–12. INFORMS.

Lyazidi, M. Y., Aitsaadi, N., and Langar, R. (2016). Dynamic resource allocation for cloud-ran in lte with real-time bbu/rrh assignment. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–6. IEEE.

Mackworth, A. K. (1977). Consistency in networks of relations. *Artificial intelligence*, 8(1):99–118.

Mackworth, A. K. (1981). Consistency in networks of relations. In *Readings in Artificial Intelligence*, pages 69–78. Elsevier.

Mann, V., Kumar, A., Dutta, P., and Kalyanaraman, S. (2011). Vmflow: Leveraging vm mobility to reduce network power costs in data centers. In *International Conference on Research in Networking*, pages 198–211. Springer.

Marler, R. T. and Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395.

Marler, R. T. and Arora, J. S. (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, 41(6):853–862.

Mehraghdam, S., Keller, M., and Karl, H. (2014). Specifying and placing chains of virtual network functions. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, pages 7–13. IEEE.

Meindl, B. and Templ, M. (2012). Analysis of commercial and free and open source solvers for linear optimization problems. *Eurostat and Statistics Netherlands within the project ESSnet on common tools and harmonised methodology for SDC in the ESS*, 20.

Meng, X., Pappas, V., and Zhang, L. (2010). Improving the scalability of data center networks with traffic-aware virtual machine placement. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE.

Mijumbi, R., Hasija, S., Davy, S., Davy, A., Jennings, B., and Boutaba, R. (2017). Topology-aware prediction of virtual network function resource requirements. *IEEE Transactions on Network and Service Management*, 14(1):106–120.

Mijumbi, R., Serrat, J., Gorricho, J.-L., Bouten, N., De Turck, F., and Boutaba, R. (2016). Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys & Tutorials*, 18(1):236–262.

Nadel, B. A. (1988). Tree search and arc consistency in constraint satisfaction algorithms. In *Search in artificial intelligence*, pages 287–342. Springer.

Namboodiri, V., Agarwal, M., and Gao, L. (2004). A study on the feasibility of mobile gateways for vehicular ad-hoc networks. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 66–75. ACM.

Namboodiri, V. and Gao, L. (2007). Prediction-based routing for vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 56(4):2332–2345.

Naranjo, J. E., Gonzalez, C., Garcia, R., and De Pedro, T. (2008). Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver. *IEEE Transactions on Intelligent Transportation Systems*, 9(3):438.

Olariu, S., Khalil, I., and Abuelela, M. (2011). Taking vanet to the clouds. *International Journal of Pervasive Computing and Communications*, 7(1):7–21.

Optimization, G. (2014). Inc.,"gurobi optimizer reference manual," 2015. *URL: http://www. gurobi. com*.

Pan, H.-Y., Jan, R.-H., Jeng, A. A.-K., Chen, C., and Tseng, H.-R. (2011). Mobile-gateway routing for vehicular networks. In *Proceedings of the 8th IEEE Asia Pacific wireless communication symposium (APWCS 2011)*.

Rao, S. (1987). Game theory approach for multiobjective structural optimization. *Computers & Structures*, 25(1):119–127.

Refalo, P. (2004). Impact-based search strategies for constraint programming. In *International Conference on Principles and Practice of Constraint Programming*, pages 557–571. Springer.

Rossi, F., Van Beek, P., and Walsh, T. (2006). *Handbook of constraint programming*. Elsevier.

Selman, B. and Gomes, C. P. (2006). Hill-climbing search. *Encyclopedia of Cognitive Science*, 81:82.

Setiwan, G., Iskander, S., Kanhere, S. S., Chen, Q. J., and Lan, K.-C. (2006). Feasibility study of using mobile gateways for providing internet connectivity in public transportation vehicles. In *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, pages 1097–1102. ACM.

Sharma, U., Shenoy, P., Sahu, S., and Shaikh, A. (2011). A cost-aware elasticity provisioning system for the cloud. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 559–570. IEEE.

Shen, Z., Subbiah, S., Gu, X., and Wilkes, J. (2011). Cloudscale: elastic resource scaling for multi-tenant cloud systems. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, page 5. ACM.

Sivaraj, R., Gopalakrishna, A. K., Chandra, M. G., and Balamuralidhar, P. (2011). Qos-enabled group communication in integrated vanet-lte heterogeneous wireless networks. In *Wireless and Mobile Computing, Networking and*

Communications (WiMob), 2011 IEEE 7th International Conference on, pages 17–24. IEEE.

Somani, G., Khandelwal, P., and Phatnani, K. (2012). Vupic: Virtual machine usage based placement in iaas cloud. *arXiv preprint arXiv:1212.0085*.

Sousa, J., Babuška, R., and Verbruggen, H. (1997). Fuzzy predictive control applied to an air-conditioning system. *Control engineering practice*, 5(10):1395–1406.

Su, W., Lee, S.-J., and Gerla, M. (2001). Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management*, 11(1):3–30.

Taleb, T. (2014). Toward carrier cloud: Potential, challenges, and solutions. *IEEE Wireless Communications*, 21(3):80–91.

Taleb, T., Bagaa, M., and Ksentini, A. (2015a). User mobility-aware virtual network function placement for virtual 5g network infrastructure. In *Communications (ICC), 2015 IEEE International Conference on*, pages 3879–3884. IEEE.

Taleb, T., Corici, M., Parada, C., Jamakovic, A., Ruffino, S., Karagiannis, G., and Magedanz, T. (2015b). Ease: Epc as a service to ease mobile core network deployment over cloud. *IEEE Network*, 29(2):78–88.

Taleb, T. and Ksentini, A. (2013). Gateway relocation avoidance-aware network function placement in carrier cloud. In *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems*, pages 341–346. ACM.

Thielen, D. (1996). *Writing Windows VxDs and Device Drivers: Programming Secrets for Virtual Device Drivers*. Elsevier.

Turing, A. M. (2009). Computing machinery and intelligence. In *Parsing the Turing Test*, pages 23–65. Springer.

Union, I. T. (2018). International telecommunications union. `http://https://www.itu.int/en`. Accessed: 2018-10-30.

van Dongen, M., Lecoutre, C., and Roussel, O. (2008). Third international csp solvers competition. *Instances and results available at http://www. cril. univ-artois. fr/CPAI08*.

Verma, A., Ahuja, P., and Neogi, A. (2008). pmapper: power and migration cost aware application placement in virtualized systems. In *Proceedings of the*

9th ACM/IFIP/USENIX International Conference on Middleware, pages 243–264. Springer-Verlag New York, Inc.

Xu, J. and Fortes, J. A. (2010). Multi-objective virtual machine placement in virtualized data center environments. In *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pages 179–188. IEEE.

Zadeh, L. A. (1996). Fuzzy logic= computing with words. *IEEE transactions on fuzzy systems*, 4(2):103–111.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132.

Résumé

L'architecture 5G sera conçue pour une société fortement connectée offrant des services en croissance. L'architecture 5G, le Cloud Computing et le réseau ad hoc de véhicules sont de différents domaines émergents et imbriqués. Par conséquent, les réseaux mobiles et réseaux de véhicules exploitent les techniques de virtualisation pour améliorer les services. Parmi les principaux défis de ces nouveaux paradigmes, nous citons la nécessité d'une gestion intelligente des ressources. Dans cette thèse, nous proposons différentes solutions jouant le rôle de contrôleurs de ressources intelligents dans le réseau mobile et le réseau de véhicules basés sur le Cloud. Les systèmes proposés sont mis en œuvre pour améliorer les travaux de la littérature et offrent à la fois une solution autonome et adaptable aux facteurs externes, ainsi qu'une solution de compromis entre les objectifs contradictoires. Diverses méthodes de l'Intelligence Artificielle sont utilisées, à savoir la programmation par contraintes, l'optimisation multi objective et le système de contrôle flou. Les résultats ont révélé l'efficacité des schémas proposés conformément à la stratégie de chaque solution.
Mots-clés : Intelligence Artificielle, Problème de Satisfaction de Contraintes, Gestion de Ressources, Architecture 5G, Réseau Ad Hoc de Véhicules.

Abstract

The 5G architecture will be designed for a hyper-connected society that combines growing services. 5G architecture, Cloud Computing, and Vehicular Ad-Hoc Network are different emerging and intertwining domains. Therefore, mobile networks and vehicular networks markets take advantages of virtualization techniques to enhance services. Among the main challenges facing these new paradigms is the need for intelligent control of resources allocation. In this thesis, we propose different solutions that play the role of intelligent resource controllers in the cloud-based mobile network and the cloud-based vehicular ad-hoc network. The proposed approaches are implemented to improve the related works and give a trade-off solution between the conflicting objectives of the problem, an autonomous solution and an adaptable solution to external factors. Various Artificial Intelligence methods are used namely constraint satisfaction problem, multi-objective optimization, and fuzzy control system. The results revealed the efficiency of the proposed schemes as per the strategy of each solution.
Keywords: Artificial Intelligence, Constraint Satisfaction Problem, Resource Controller, 5G Architecture, Vehicular Ad-hoc Network.

Année Universitaire : 2018/2019