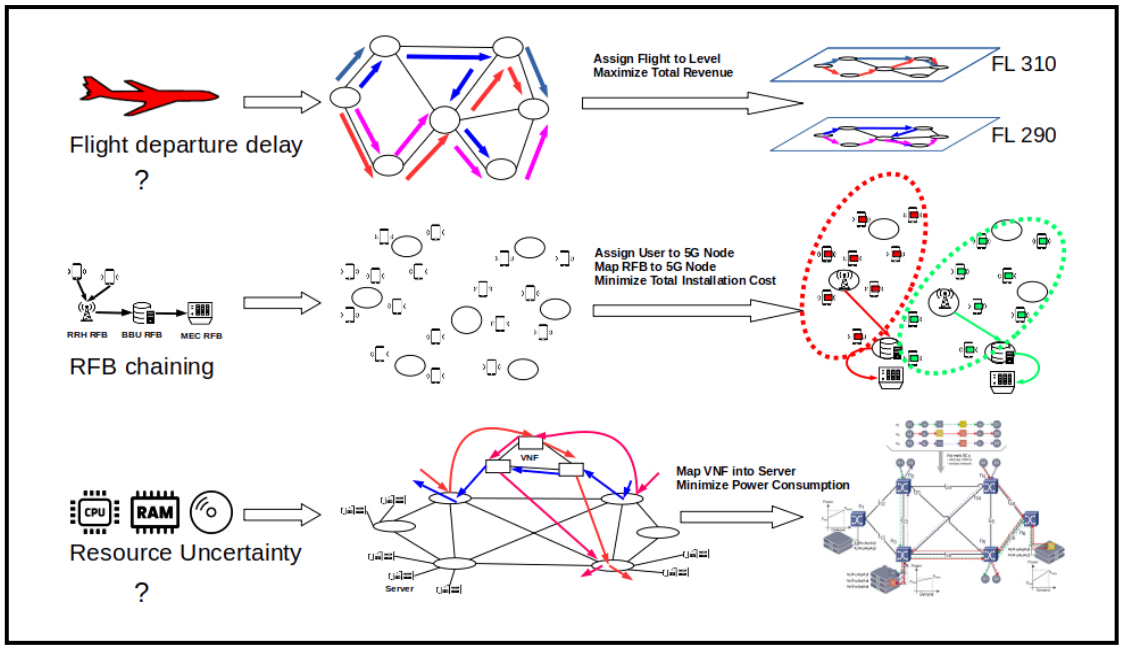
Contribution to robust network optimization

Chenghao Wang

Par **Chenghao WANG**

Contribution à l'optimisation robuste de réseaux

Thèse présentée pour l'obtention du grade de Docteur de l'UTC



Soutenue le 30 septembre 2021
Spécialité : Informatique : Unité de recherche Heudyasic (UMR-7253)

D2632

Université de Technologie de Compiègne
Heudiasyc Laboratory

Thèse

En vue de l'obtention du titre de Docteur
Spécialité « Informatique »

présentée par

Chenghao WANG

Contribution à l'optimisation robuste de réseaux

Thèse soutenue le 30 septembre 2021 devant le jury composé de :

M. Marc BUI	professeur	membre examinateur
M^{me} Ghada JABER	maître de conférences	membre examinateur
M. Niko KAÇIROTI	chercheur	membre rapporteur
M. Stefano CONIGLIO	professeur associé	membre rapporteur
M. Dritan NACE	professeur	directeur de thèse
M. Fabio d'Andreagiovanni	chargé de recherche	directeur de thèse

Contribution to robust network optimization

Chenghao WANG

September 30, 2021

«To be uncertain is to be uncomfortable, but to be certain is to be ridiculous» —Chinese proverb

Acknowledgment

The research work done in this thesis took place in Heudiasyc Laboratory, Université de Technologie de Compiègne from December 2017 to June 2021. It was funded by the French Ministry of High Education and Research.

At the end of this thesis, I would like to express my sincere gratitude to my supervisor Dr. Fabio D'Andreagiovanni for his valuable advice, constructive assistance, and precious help. I would like to thank him for his optimism, positivity, and enthusiasm. He always encouraged me whenever I felt nervous somehow. I am sincerely grateful for his company of the interview of P.h.D candidate, laboratory days of Heudiasyc, preparation and setdown in the visiting period in Germany, and especially the correction of the thesis manuscript.

I would like to express my sincere gratitude to my thesis supervisor, Prof. Dritan Nace, who supported me throughout the thesis period under the best possible conditions. Since the beginning of April 2017 for the internship of Engineer Diploma, we have already worked together for 4 years and a half. I am extremely grateful for his precious guidance when I were lost somehow in the research, his constructive advice for my research, his availability during all this time whenever I need of his help, and his great share of personal time for the countless revisions for all my paperwork. I would like to mention his meticulous attention to my research and life, especially during the COVID-19 period, when I was alone for the visit program in Germany, and when I was sick. So many cares from him, such that it is hard to enumerate all of them.

Frankly speaking, it was a honor to work together with my supervisors for these years. Besides them, I want to thanks all my co-authors Prof. Thomas Bauschert, Prof. Akli Fundo, Dr. Jean-Benoist Leger, Prof. Andreas Kassler, Dr.Liyang Xiao and the two master students Zhengpei Wang and Zheyi Tan, for their participation in these publications and their sincere help. Especially, I would like to thank Prof. Thomas Bauschert for his warm reception in Chemnitz, Germany, also for the

few but wonderful discussions of research topics. Finally, I am deeply grateful to my best friend Liyang in the academic domain, for his constructive advice in the collaborative research and precious help.

I want to thank all the members of the jury for agreeing to participate in my thesis committee.

A sincere thank goes to all the members and colleagues of Heudiasyc laboratory and specially the SCOP team for their friendly support during this period.

Finally, and most importantly, I want to thank my family and friends, especially Zhang Wenlin, for their love, care, patience, and unconditional and permanent support.

List of Publications

1. Akli Fundo, Dritan Nace, and Chenghao Wang. A heuristic approach for the robust flight level assignment problem. In *International Conference on Belief Functions*, pages 86–94. Springer, 2018

2. Thomas Bauschert, Fabio D'andreagiovanni, Andreas Kassler, and Chenghao Wang. A matheuristic for green and robust 5g virtual network function placement. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pages 430–438. Springer, 2019

3. Chenghao Wang, Fabio D'Andreagiovanni, and Dritan Nace. Solving a resource allocation problem in rfb-based 5g wireless networks. In *Third International Balkan Conference on Communications and Networking (BalkanCom 2019)*, 2019

4. Akli Fundo, Jean-Benoist Leger, Dritan Nace, and Chenghao Wang. Dealing with uncertainty in atm-the flight level assignment problem. In *21e congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2020)*, 2020

5. Liyang Xiao, Zhengpei Wang, Zheyi Tan, and Chenghao Wang. A solution method for the maritime pilot scheduling problem with working hour regulations. *Asia-Pacific Journal of Operational Research*, 38(03):2040015, 2021

Contents

List of Tables

List of Figures

Acronyms

$\mathcal{NP}$ Nondeterministic Polynomial time.

1G 1st generation of wireless networks.

2G 2nd generation of wireless networks.

3G 3rd generation of wireless networks.

3-GC 3-Graph (vertex) Coloring.

4D 4 Dimensions.

4G 4th generation of wireless networks.

5G 5th generation of wireless networks.

5G-RFB-RA RFB-based Resource Allocation problem under 5G Superfluid wireless Networks.

5G SF Network 5G Superfluid wireless Networks.

5G SF 5G Superfluid.

ATC Air Traffic Control.

ATFM Air Traffic Flow Management.

ATM Air Traffic Management.

BBU-RFB Base Band Unit RFB.

BLP Binary Linear Programming.

CCP Chance-Constrained Programming.

CO2 Carbon dioxide.

FL Flight Level.

FLA Flight Level Assignment.

GA Genetic Algorithm.

GMM Gaussian Mixture Model.

HD High Definition.

HeuMod method modified heuristic feasibility probability estimation method.

Heuristic method heuristic feasibility probability estimation method.

Hoeffding method Hoeffding's inequality feasibility probability estimation method.

IaaS Infrastructure as a Service.

ILP Integer Linear Programming.

InfraP Infrastructure Provider.

IP Internet Protocol.

ISPs Internet Service Providers.

LP Linear Programming.

LTE Long Term Evolution.

MC method Monte-Carlo simulation feasibility probability estimation method.

MEC-RFB Mobile/Multi-access Edge Computing RFB.

NFV Network Function Virtualization.

RASK Revenue per Available Seat Kilometer.

RDCL RFB Description and Composition Language.

REE RFB Execution Environments.

RF Radio Frequency.

RFB Reusable Functional Block.

RINS Relaxation Induced Neighborhood Search.

RO Robust Optimization.

RRH-RFB Resource Radio Head RFB.

RVSM Reduced Vertical Separation Minima.

SN Substrate Network.

Soyster method Soyster feasibility probability estimation method.

SP Stochastic Programming.

TFMRP Traffic Flow Management Re-routing Problem.

VM Virtual Machine.

VN Virtual Network.

VNF Virtual Network Function.

VNFC Virtual Network Function Component.

VNFP Virtual Network Function Placement.

VNO Virtual Network Operator.

VNP Virtual Network Provider.

Résumé

Titre Contribution à l'optimisation robuste de réseaux

Résumé Cette thèse a pour objectif la proposition de nouvelles approches algorithmiques et de modélisation pour la résolution de certains problèmes d'optimisation de réseau dans les domaines des transports et des télécommunications.

Plus précisément, les problèmes étudiés tombent dans le domaine du transport aérien, nommément le problème d'affectation des niveaux de vol dans l'espace aérienne, et dans le domaine des télécommunications où on traite des problèmes d'allocation de ressources dans les réseaux 5G. Un aspect important qui a été pris en compte dans cette étude est l'incertitude des données, c'est-à-dire le fait qu'une partie des données d'entrée ne sont pas connues de façon precise. Notre tâche a consisté à modéliser chacun des problèmes, proposer une formulation compacte des variantes déterministe et robuste, et proposer des approches appropriées pour les résoudre. Les problèmes étudiés tombent dans la catégorie des problèmes $\mathcal{NP}$-complets et ils sont difficile à résoudre même pour des instances de taille modeste. Ils deviennent encore plus difficiles dans leur version robuste.

Pour le problème d'affectation des niveaux de vols, nous avons considéré les incertitudes liées à l'heure de départ qui sont modélisées via un modèle de mélange gaussien. Le problème est modélisé comme un « chance-constrained problem » et résolu par un algorithme heuristique de génération de contraintes. Il s'agit d'une approche générale qui trouvera des applications plus large que ceux étudiés dans cette thèse.

Ensuite, nous avons étudié la conception optimale des réseaux sans fil de 5ème génération (5G) dans le contexte de l'architectures Superfluid. Plus précisément, l'architecture 5G Superfluid est basée sur des entités de réseau appelées « Bloc Fonctionnel Réutilisable » (RFB) qui sont à la base des réseaux 5G. Nous avons étudié le problème de conception d'un tel réseau Superfluid à coût minimum pour

lequel une formulation en programme linéaire mixte suivie d'une approche de résolution utilisant la décomposition de Benders a été implémentée ont été proposées. Enfin, le problème spécifique de conception de réseaux virtuels a été considéré sous l'angle de l'efficacité énergétique. Nous avons proposé une formulation de programmation linéaire en nombres entiers mixte du problème robuste, et présentons une nouvelle matheuristique basée sur la combinaison d'un algorithme génétique avec la recherche de voisinage.

Les résultats numérique ont porté sur des instances de taille réalistes et ont montré la validité des modèles et des approches proposées.

Mots-clés Optimization réseaux, Optimisation robuste, Chance-constrained Programming, Approches exactes, Metaheuristiques

Abstract

Title Contribution to robust network optimization

Abstract This Ph.D. Thesis is focused on proposing new optimization modeling and algorithmic approaches for dealing with real-world network optimization problems arising in the transportation and telecommunications fields. Since the focus has been on real-world applications, a relevant aspect that has been taken into account is data uncertainty, i.e. the fact that the value of a subset of input data of the problem is not exactly known when the problem is solved.

More precisely, in the context of transportation problems, it was considered the flight level assignment problem, which arises in air traffic management. It aims at establishing the flight levels of a set of aircraft in order to improve the total assignment revenue, to reduce the total number of flight conflicts and also the total en-route delay. In this context, we proposed a new chance-constrained optimization problem and iterative constraint-generation heuristic which is based on both analytical and sampling methods.

Besides transportation problems, this Thesis has also focused on the optimal design of 5^{th} generation of wireless networks (5G) considering Superfluid and virtual architectures. Specifically, the 5G Superfluid architecture is based on atomic virtual entities called Reusable Functional Block (RFB). We investigated the problem of minimizing the total installation costs of a 5G Superfluid network (composed of virtual entities and realized over a physical network) while guaranteeing constraint on user coverage, downlink traffic performance and technical constraints on RFBs of different nature. To solve this hard problem, we proposed a Benders decomposition approach.

Concerning instead the design of general virtual networks, we adopted a green paradigm that pursues energy-efficiency and tackled a state-of-the-art robust mixed integer linear programming formulation of the problem, by means of a new matheuris-

tic based on combining a genetic algorithm with exact large neighborhood searches. Results of computational tests executed considering realistic problem instances have shown the validity of all the new optimization modeling and algorithmic approaches proposed in this Thesis for the transportation and telecommunications problems sketched above.

Keywords Network optimization, Robust optimization, Chance-constrained Programming, Exact Approaches, Metaheuristics

1. Introduction

In our everyday life, infrastructures using networks have deeply pervaded our activities by putting at our disposal a wide range of services of very different nature that have become essential for us. If we want to give only a few major examples, we can think of telecommunications networks, both wireless and wired, by which we communicate via smartphones and internet; transportation networks, which allow us to move by air, ground and sea; and power networks, which transport and distribute the electrical energy powering a high number of personal and family devices that we use everyday. All these infrastructures are connected through a network lying at their core, which can be generically defined as a set of entities in which couples of them are connected according to some relation. When dealing with networks, it comes natural to model them by graphs made up of a set of nodes and a set of edges expressing the connection between pairs of nodes. Network optimization problems are thus naturally associated with problems related to graphs and typically model events in the network under the form of flows moving across the network. For an exhaustive introduction and discussion about network optimization problems, we refer to the famous book [Ahuja et al., 1993].

Given the interest of this Ph.D. Thesis in studying and proposing new results about optimization methods for network optimization problems arising in real-world applications, an important aspect that we had to take into account in our research has been the presence of data uncertainty, namely the fact that (part of) the input data of the problem at hand are typically not exactly known in value when the problem is solved.

To tackle a real-world problem, we commonly establish a deterministic mathematical model by assuming that the data inputs are precisely known and try to solve such model in an exact or approximate way in order to obtain a qualified so-called "optimal solution" in each context. However the impact of data uncer-

tainty in these constructed models on the a-posteriori optimality and feasibility of the obtained solutions are ignored. Thereby, the solution generated from reference data that differ from the actual values encountered in reality may lead to identify solutions that result far away from actual optimal solutions and may even violate feasibility constraints. [Ben-Tal and Nemirovski, 2000] have shown that even small (0.01%) perturbations in the value of input data can result in highly infeasible solutions for some benchmark optimization problems. Consequently, the need for methodologies capable of generating a robust solution, namely a solution immune to data uncertainty, naturally arises. The basic idea of robust optimization is to seek a solution which remains feasible and optimal even when deviations in the input data of an optimization problem occur.

To deal with relevant sources of data uncertainty arising in real-world network optimization problems related to transportation and communications networks, we have obtained a number of original modeling, algorithmic and computational results that are presented in the reminder of this Ph.D. Thesis, for which we provide an overview of the structure and of the main original contributions in the next sections.

1.1 Thesis organization

The thesis is organized in 6 chapters, including the introduction and the conclusion, as follows:

- **Chapter 2: Optimization Methods under Data Uncertainty.** This chapter is devoted to provide a concise introduction to optimization under uncertainty, highlighting the issues of dealing with uncertain data in optimization and offering an overview of major methodologies proposed over time.

- **Chapter 3 : Robust Flight Level Assignment problem.** This chapter is devoted to the flight level assignment problem. The problem arises in the Air Traffic Management (ATM) context where several flights compete to share the airspace resources, that is the flight levels. An appropriate assignment will lead to less conflicts and reduce the delays. We study in detail the Flight Level Assignment (FLA) problem and its robust variant. In practice, we model and

solve a Chance-Constrained Programming (CCP) linear problem where the coefficients are uncertain but follow a known Gaussian Mixture distribution. Our solution methodology stands in solving the CCP problem through an iterative heuristic approach. In each iteration a solution is provided and its feasibility probability evaluated. Then, an important element of the approach is the feasibility probability estimation where both analytical and sampling methods are proposed and experimented.

- **Chapter 4: Resource Allocation in 5G Superfluid Wireless Networks.** This chapter is devoted to address an optimization problem related to the design of 5^{th} generation of wireless networks (5G). Specifically, we have considered the 5G network architecture proposed and studied in the European Horizon 2020 project "Superfluidity" , detailed in Bianchi et al. [2016]. The 5G Superfluid architecture is based on atomic virtual entities called Reusable Functional Block (RFB)s, which are able to support the high level of flexibility, agility, portability and high performance required by 5G. We investigate the problem of minimizing the total installation costs of a Superfluid network composed of virtual and realized over a physical network, while guaranteeing constraint on user coverage, downlink traffic performance and technical constraints establishing relations between RFBs of different nature. We propose a new mathematical formulation which can enhance the possibility of optimally solving realistic networks instances and a Benders-like decomposition approach for accelerating the solution process.

- **Chapter 5: Green and Robust 5G Virtual Network Function Placement Problem.** We investigate the problem of optimally placing virtual network functions in 5G-based virtualized infrastructures according to a green paradigm that pursues energy-efficiency. This optimization problem can be modeled as an articulated Mixed Integer Linear Programming problem with a multicommodity flow model at its core. To model the data uncertainty that naturally affects the volume and features of the traffic associated with the requests of establishing virtual networks generated by users, we rely on adopting a robust optimization approach according to the Γ-robustness paradigm. Since the resulting robust counterpart may easily become challenging to solve even for instances of moderate size for state-of-the-art solvers, we propose a

new matheuristic for its solution. The matheuristic is based on combining a genetic algorithm with an exact large neighborhood search. Computational tests on realistic instances returned good results, showing that our algorithm can find better solutions in sensibly less time than a state-of-the-art solver.

- **Chapter 6: Conclusions.** This chapter concludes the Thesis, proposing a number of final considerations and possible directions for future work.

1.2 Contributions of this Ph.D. Thesis

The contribution of this thesis is threefold. We first investigate the Flight Level Assignment problem and its Chance-Constrained variant, proposing a compact deterministic mathematical formulation for the problem and a Chance-Constrained Programming counterpart. For its solution, we propose an iterative approach and another central original contribution is represented by the specific constraint generation approach solving the CCP problem associated with a given flight level. Using the results of Klopfenstein [2009], we reformulate the CCP as an ILP, in which the probability constraints are replaced taking into account the worst case, according to the procedure proposed in [Soyster, 1973]. These constraints are dynamically included when needed, through a constraint generation approach, until the desired feasibility probability is reached. All this gives a practical approach which may find application in a class of CCP problems. Another novelty of our approach is represented by the method used to check the feasibility probability of constraints. We study the case with uncertain parameters following a Gaussian Mixture Model (GMM) and propose an approximated method to estimate the feasibility probability of the solution. This may be of interest since the GMM is a powerful tool to capture characteristics distribution of a large number of real situations.

Concerning the design of 5G Networks based on the Superfluid architecture, the major contributions are constituted by proposing an alternative formulation for the optimization model by Bianchi et al. [2016] that we have taken as reference. Specifically, we were able to propose alternative feasibility constraints and characterize valid inequalities that express in a simpler way the technological constraints on the installation of the basic virtual entities, the so-called Reusable Functional

Blocks, in distinct 5G network nodes. This simpler constraints lead to a new mathematical model that can be better handled by solvers. Moreover, to accelerate and improve the capacity of solving realistic instances, we have also proposed a new solution approach based on Benders decomposition that breaks the complete model into a master and a slave problems decoupling the complicated relations linking the installation of reusable functional blocks of different nature. Results of computational tests show the advantages of this new modeling and solution approach that we proposed.

Finally, the last contribution is related to the design of virtual networks according to a green network paradigm that pursues energy minimization. In this context, we have taken as reference state-of-the-art works and proposed a new effective and efficient matheuristic for solving the robust counterpart of the problem, exploiting the integration of a genetic algorithm with exact neighborhood searches, which formulate the exploration of (very) large neighborhood as mathematical programming problems solved to optimality by means of state-of-the-art solvers. The rationale at the basis of such matheuristic is that, while a state-of-the-art solver may not be able to solve the complete problem, it can instead efficiently solve to optimality suitable subproblems. Computational tests over realistic virtual network instances confirm the advantages of adopting such new matheuristic integration.

The original results outlined above have been presented in the publications:

1. Akli Fundo, Dritan Nace, and Chenghao Wang. A heuristic approach for the robust flight level assignment problem. In *International Conference on Belief Functions*, pages 86–94. Springer, 2018

2. Thomas Bauschert, Fabio D'andreagiovanni, Andreas Kassler, and Chenghao Wang. A matheuristic for green and robust 5g virtual network function placement. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pages 430–438. Springer, 2019

3. Chenghao Wang, Fabio D'Andreagiovanni, and Dritan Nace. Solving a resource allocation problem in rfb-based 5g wireless networks. In *Third International Balkan Conference on Communications and Networking (BalkanCom 2019)*, 2019

4. Akli Fundo, Jean-Benoist Leger, Dritan Nace, and Chenghao Wang. Dealing with uncertainty in atm-the flight level assignment problem. In *21e congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2020)*, 2020

5. Liyang Xiao, Zhengpei Wang, Zheyi Tan, and Chenghao Wang. A solution method for the maritime pilot scheduling problem with working hour regulations. *Asia-Pacific Journal of Operational Research*, 38(03):2040015, 2021

and we are currently working to complete and finalize the journal versions of the conference papers.

2. Optimization Methods under Data Uncertainty

2.1 An introduction to data uncertainty

A fundamental assumption that is made in classical optimization is that all the data input of a problem are known *exactly* and *precisely* when the problem is solved. However, when dealing with optimization problems arising in the real world, one can often directly experience that such assumption does not hold and the input data are subject to uncertainty at some level, meaning that the value of (a subset of) coefficients appearing in the problem is not exactly known. The hypothesis of data certainty is typically adopted since this commonly leads to more tractable and less complex problems. However, obtained solutions under this artificial data certainty assumption may prove to be not useful in practice. As a consequence, adopting suitable modeling and solution techniques is imperative.

For an exhaustive introduction to the challenges and issues associated with optimization under data uncertainty, we refer the reader to the book [Ben-Tal et al., 2009] and to the survey [Bertsimas et al., 2011a], which, though not very recent, still constitute major references for fundamentals of optimization under uncertainty. For more up-to-date surveys, we also refer to the work, [Long et al., 2019], [Yanıkoğlu et al., 2019] and [Leyffer et al., 2020]. As discussed in [Ben-Tal et al., 2009], the presence of data uncertainty may be attributed to many causes, among which the most remarkable are:

- Errors due to prediction: in this case, the data are not known since they are related to events that will take place in the future and can be only guessed or forecast on the basis of historical data (when available). This is the case, for

example, of traffic conditions in telecommunications networks which depend on the future behavior of customers and users and requires to be forecast (we note that in this case, practitioners tend to provide conservative predictions). Moreover, it can also be cited the case of resolution of flight conflicts in Air Traffic Management, which depends directly or indirectly on the prediction of weather conditions and flight trajectories.

- Errors due to measurements: it may happen that the data required in the optimization problem cannot be precisely measured and are naturally subject to measurement errors that must be taken into account when solving the problem. Referring to the case of telecommunications networks, such as the 5G virtualized networks that we consider, this could be represented by real-time measurements of the delay within the network.

- Errors due finite precision numerical representation: another source of uncertainty may be simply represented by the fact that optimization problems are commonly modeled and solved with by means of ad-hoc computer software, which rely on a finite precision representation of numbers and on finite precision arithmetic. In most applications, such error may be neglected. However, in some other, like wireless network design including signal-to-interference constraints, this error cannot be neglected, since it would lead to non accurate solutions, and must be taken into account (see [D'Andreagiovanni et al., 2013]).

A very important observation that should be made at this point is that, over the years, many methodologies have been proposed in order to deal with data uncertainty in optimization problems and there is no just one "major and more effective" method that could adopted in all contexts. In what follows, we thus try to provide a concise overview of the main and most used methodologies, focusing more on those that are adopted as basis for our original developments.

2.2 Stochastic Programming

The first methodology that has considered the issue of data uncertainty in an optimization problem can be considered Stochastic Programming (SP), which can

be traced back to the seminal work of Dantzig presented in [Dantzig, 1955]. The main assumption at the basis of SP is that the stochastic distribution of data subject to uncertainty is known. The general form of a SP problem can be written as:

$$\begin{aligned} \min \quad & \mathbb{E}\left[c(x,R)\right] \\ \text{s.t.} \quad & f_i(x,R) \geq b \qquad \forall i \in I \\ & x \geq 0 \end{aligned}$$

In this problem, which aims at minimizing the expected value of the cost function, $x \in \mathbb{R}^n$ is the vector of decision variables, $c : \mathbb{R}^n \to \mathbb{R}$ is the cost objective function, $f_i : \mathbb{R}^n \to \mathbb{R}$ is the constraint function of the i^{th} constraint. Also, $R \in \mathbb{R}^{|I|}$ is a vector of random variables affecting the input coefficient and for which we know the corresponding distributions. For an exhaustive introduction to modeling and solution principles of Stochastic Programming, we refer the reader to [Shapiro et al., 2014].

For many years, SP has represented the main methodology for dealing with data uncertainty in optimization and has been heavily studied and improved. However, a major limitation of SP that has been identified over the years is that it requires to know the probability distribution followed by the uncertain data and, as it is know, in many real-world application such distributions are not known. As a consequence, as also discussed in [Ben-Tal et al., 2009], the application of SP is not so straightforward and accurate in a consistent number of relevant applications. Furthermore, it is also known that solving SP problems may be computationally challenging and expensive in practice, since typical solution approaches need to consider a high number of scenarios that are representative of the probability distributions of the uncertain data. This leads to very large problems that commonly require to be solved by suitable decomposition methods (see [Shapiro et al., 2014]).

2.3 Robust Optimization

With the aim of overcoming some of the drawbacks of Stochastic Programming, the paradigm of Robust Optimization has been introduced at the beginning of the

new millennium. Specifically, a cornerstone of RO is to not deal with probability distributions, but to assume that the possible realizations of the uncertain data are completely specified through a so-called uncertainty set U. Then RO has the objective of finding the optimal solution of a *robust counterpart* of the original problem, identifying the solution that is optimal under all possible realizations of the data specified by the uncertainty set U. More formally, let us consider a general optimization problem:

$$\begin{aligned} \min \quad & c(x) \\ \text{s.t.} \quad & f_i(x) \geq b \qquad \forall i \in I \\ & x \geq 0, \end{aligned}$$

in which we aim at minimizing the cost function $c : \mathbb{R}^n \to \mathbb{R}$ involving the vector of decision variables $x \in \mathbb{R}^n$, and in which $f_i : \mathbb{R}^n \to \mathbb{R}$ is the constraint function of the i^{th} constraint. For a given uncertainty set U, the general form of the robust counterpart of the previous problem can be written as:

$$\begin{aligned} \min \quad & \max_{u \in U} c(x) \\ \text{s.t.} \quad & f_i(x, u) \geq b \qquad \forall i \in I, u \in U \\ & x \geq 0 \end{aligned}$$

Feasible solutions of the previous problem must be feasible for all the realizations of the uncertain data specified by the uncertainty set U (indeed, we can notice that each feasibility constraint must be satisfied for each realization $u \in U$). Such feasible solutions are called *robust feasible solutions*. Moreover, we can notice that the objective function takes the form of a min-max problem and identifies as robust optimal that solution granting the best cost value under the worst realization of the uncertain data of U. We remark that, without loss of generality, we can focus on the case of problems in which the uncertainty is just present in the input data appearing in the constraints. Indeed, if uncertain data are present in the objective or in the right-hand-sides of constraints, such uncertainty can be easily reformulated as uncertainty affecting only the constraints, as detailed in [Ben-Tal et al., 2009, Bertsimas and Sim, 2004].

A critical task of RO is to "rationally " define the uncertainty set U. Indeed, since a solution is feasible for the robust counterpart only if it maintains its feasibility for all the realizations of the data specified by U, taking into account extreme and unlikely realizations of the data and including them in U may lead to very conservative solutions that are not useful in practice. As a consequence, a very substantial part of the literature of RO has been devoted to identifying those rational uncertainty sets that could provide satisfying level of robustness against data uncertainty without leading to over-conservative robust solutions. For an overview of the various kind of alternative uncertainty sets that have been proposed over time, we refer the reader to the book [Ben-Tal et al., 2009] and the more recent survey [Gabrel et al., 2014]. In particular, these works proposing uncertainty sets have tried to overcome the limits of what can be considered the first (very conservative) example of robust optimization, proposed in 1973 [Soyster, 1973]. Here, we focus on so-called *cardinality-constrained uncertainty sets* that have constituted the most used sets and to which belongs the famous Γ-Robustness model by Bertsimas and Sim [Bertsimas and Sim, 2004] that we have taken as reference in our studies. For defining a cardinality-constrained set in a more formal way, let us introduce two vectors $\bar{u}, \hat{u} \in \mathbb{R}^p$ and let us impose that all the realizations u of the uncertain data must belong to the set $[\bar{u} - \hat{u}, \bar{u} + \hat{u}]$. The central value $\bar{u}$ is commonly called *nominal value*, whereas $\hat{u}$ is commonly called highest or worst deviation. Furthermore, let us introduce an integer value β, called *robustness budget*, which indicate the highest number of input coefficients that may deviate from their nominal value simultaneously, then the cardinality-constrained uncertainty set may be written as:

$$U = \left\{ u \in \mathbb{R}^p : \boxed{\begin{array}{l} \bar{u}_k - \delta_k \hat{u}_k \leq u_k \leq \bar{u}_k + \delta_k \hat{u}_k, \\ \sum_{k=1}^{p} \delta_k \leq \beta, \\ \delta_k \in \{0, 1\} \end{array}} \right\}$$

According to this formalization, at most β coefficients may deviate up to their worst deviation. We note also that such set could be easily defined also for fractional value of the coefficients δ (see e.g., [Bertsimas and Sim, 2004]). After having provided such general definition of cardinality-constrained uncertainty set, we proceed to concentrate our attention on the famous Γ-robustness model.

2.3.1 Γ-Robust Optimization

As we discussed in the previous section, Γ-Robustness, proposed originally in the papers [Sim, 2003] and [Bertsimas and Sim, 2004] belongs to the family of cardinality-constrained uncertainty sets. Formally, it can defined as follows. First of all, the assumptions at the basis of this model of uncertainty are:

1. Each entry u_k of a vector u of the uncertainty set U represent a random variable following un unknown bounded and symmetrical probability distribution. These variables are assumed to be independent.

2. For each entry u_k, the distribution is symmetrical with respect to a nominal value $\bar{u}_k$ and, given a maximum deviation $\hat{u}_k$, is defined over the interval $[\bar{u} - \hat{u}, \bar{u} + \hat{u}]$.

3. Each realization of the data uncertainty u may have at most $\Gamma \geq 0$ elements that deviate from the nominal value $\bar{u}_k$.

Under these assumptions and exploiting the theoretical results presented in [Sim, 2003] and [Bertsimas and Sim, 2004] , let us derive the robust of the following generic Linear Programming:

$$\begin{aligned} \max \quad & c'x \\ \text{s.t.} \quad & a_i'x \leq b_i && \forall i \in I \\ & x \geq 0 \end{aligned}$$

Its generic robust counterpart can be written as:

$$\begin{aligned} \max \quad & c'x \\ \text{s.t.} \quad & a_i'x \leq b_i && \forall i \in I, \forall a_i' \in U_i \\ & x \geq 0, \end{aligned}$$

in which we consider all the possible realizations of the row vector of constraint i that respect the assumptions on the uncertainty set. As next step, a critical observation that can be made is that the previous program may have exponentially

many constraints. However, we can tackle this issue by considering the following equivalent problem:

$$\begin{aligned} \max \quad & c'x \\ \text{s.t.} \quad & \max_{a'_i \in U_i} a'_i x \leq b_i && \forall i \in I \\ & x \geq 0, \end{aligned}$$

which, however, is non-linear due to the presence of the max term in the constraints. A major result by Bertsimas and Sim has been to elegantly prove that such non-linearity may be tackled by noticing that, for a constraint i and for a fixed vector $\bar{x}$, computing the worst deviation expressed by the max term can be formulated as the following Binary Linear Programming problem:

$$\max_{a'_i \in U_i} a'_i x = \left\{ \begin{array}{ll} \max & \sum_{j=1}^{n} \bar{a}_{ij}\bar{x}_j + \max \sum_{j=1}^{n} \hat{a}_{ij}\bar{x}_j y_j \\ \text{s.t.} & \sum_{j=1}^{n} y_j \leq \Gamma \\ & y_j \in \{0,1\} \qquad \forall j = 1, \ldots, n \end{array} \right\}$$

As well-noted by Bertsimas and Sim, an integral optimal solution of the previous problem can be obtained by solving its linear relaxation, which is:

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} \bar{a}_{ij}\bar{x}_j + \max \sum_{j=1}^{n} \hat{a}_{ij}\bar{x}_j y_j \\ \text{s.t.} \quad & [z_i \geq 0] \qquad \sum_{j=1}^{n} y_j \leq \Gamma \\ & [v_{ij} \geq 0] \qquad 0 \leq y_j \leq 1 && \forall j = 1, \ldots, n \end{aligned}$$

Since the previous problem is linear, bounded and admits an optimal solution, we can define its dual problem which is also bounded and admits an optimal solution of identical value:

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} \bar{a}_{ij}\bar{x}_j + \Gamma z_i + \sum_{j=1}^{n} v_{ij} \\ \text{s.t.} \quad & z_i + v_{ij} \geq \hat{a}_{ij}\bar{x}_j \\ & z_i \geq 0 \end{aligned}$$

$$v_{ij} \geq 0 \qquad \forall j = 1, \ldots, n$$

Finally, we can then reinsert the minimization problem in our original non-linear problem to substitute the max term, obtaining the following compact and linear robust counterpart:

$$\begin{aligned}
\max \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} \bar{a}_{ij} x_j + \Gamma z_i + \sum_{j=1}^{n} v_{ij} \leq b_i && \forall i \in I \\
& z_i + v_{ij} \geq \hat{a}_{ij} \bar{x}_j && \forall i \in I, \forall j = 1, \ldots, n \\
& x_j \geq 0 && \forall j = 1, \ldots, n \\
& z_i \geq 0 && \forall i \in I \\
& v_{ij} \geq 0 && \forall i \in I, \forall j = 1, \ldots, n,
\end{aligned}$$

which has the advantage of being easy to derive and to provide to a state-of-the-art optimization solver like CPLEX or GUROBI for being solved. For an exhaustive formal description of all the passages, we refer the reader to the papers [Sim, 2003] and [Bertsimas and Sim, 2004] which have formally presented the Γ-robustness approach. To conclude this subsection, we highlight that the results illustrated above are at the basis of the original robust optimization results that we present in the chapters that follow.

2.4 Chance-constrained Optimization

We finally introduce some fundamentals of Chance-constrained Optimization, referring the reader to the next chapter for a deeper coverage, taking also into account the specific features of the real-world application that was considered. A natural way for including stochastic data in a mathematical optimization problem is represented by the definition of probabilistic constraints, imposing that, for a given value $\epsilon \in (0.1)$, a constraint including stochastic data should be satisfied with a

probability at least equal to $1 - \epsilon$. More in detail, if, for example, we refer to a generic linear constraint $a'x \leq b$, where a is a vector of stochastic data for which we know the corresponding probability distribution, then we would consider the following *chance-constrained* version of the linear constraint:

$$P\left[a'x \leq b\right] \geq 1 - \epsilon$$

expressing that we identify as feasible solutions those x that satisfy the linear constraint $a'x \leq b$ with probability at least $1 - \epsilon$.

The concept of chance-constrained mathematical programming problems is commonly traced back to the seminal work [Charnes and Cooper, 1959], which has dealt with the definition of an optimization approach to manage heating oil production while taking into account weather and demand uncertainty. While Chance-constrained optimization can be recognized as a natural way of including stochastic data in mathematical programming, at the same time, it is known to pose a number of computational challenges. Specifically, as discussed in [Ben-Tal et al., 2009], it typically leads to problems that are computationally intractable for two main reasons:

1. it may result hard to evaluate with accuracy the probability of the stochastic data appearing in the constraints, even when the probability distribution is "simple";

2. the feasible set associated with chance-constrained models is typically non-convex, thus leading to problems that are hard to solve.

To tackle such computational intractability, a way could be constituted by trying to define suitable convex approximations of the problem, as done for example in [Klopfenstein, 2009]. However, we remark that these challenges constitute a stimulus to develop new more effective and efficient solutions approaches, as we do in the next chapter.

3. Robust Flight Level Assignment Problem

3.1 Introduction

With the highly increasing demand for commercial flights each year, the air traffic has been heavily increased by around 14% in Europe in 2019 compared to 2014, with a total of over 11.1 million flights [EUROCONTROL, 2019b]. Leaving out the catastrophic 2020, the average annual growth was forecast at 2.0% per year for the next five years [EUROCONTROL, 2019c]. Although the current situation of traffic airspace is greatly underloaded due to the world pandemic situation, one may expect that there will still be a high level of congestion in airspace in a few years, leading to important delays. Among all-cause delays for airlines, the en-route Air Traffic Flow Management (ATFM) delay is a significant cause of delay to airlines. It is still far from the reference values (0.5 minute per flight), though it decreased to 1.6 minutes per flight in 2019, where the total ATFM delays (airport, en-route, and weather delay) reported by airlines decreased to 2.7 minutes. Moreover, the level of delay was the third-worst in the last 10 years, behind 2010 and 2018, with en-route ATFM delays during summer season remaining a problem for airlines [EUROCONTROL, 2019a]. Several solutions have already been proposed to deal with en-route congestion such as reducing the size of control sectors or the distance of separation, while the current Air Traffic Management (ATM) system seems to have reached the structural limits of the system. Apart from the aforementioned approaches, several degrees of freedom on the trajectories can be exploited to regulate the traffic in order to reduce the potential conflicts and hereafter to improve the Air Traffic Control (ATC) capacity, such as re-routing and

flight level allocation or assignment. Our focus is on flight level assignment, that is assigning each flight to an appropriate level (cruising altitude level), such that the total en-route ATFM delays are reduced. The uncertainty is an important factor to be taken into account when dealing with air traffic issues - there were over 12.5% of flights delayed by an ATFM regulation in 2019 where the average flight departure time delay is 13.1 minutes per flight and about 40% of these flights were delayed by more than 15 minutes [EUROCONTROL, 2019b]. All this pleads for careful modeling of the problem including uncertainty, and the need for robust optimization.

3.2 Related works

Optimization problems in ATM in relation to en-route congestion have been widely studied these last decades. Let us cite first the fundamental work of Bertsimas and Patterson, where the Traffic Flow Management Re-routing Problem (TFMRP) [Bertsimas and Patterson, 2000] is formulated and in-depth investigated. The authors show how to optimally control aircraft by re-routing, delaying, or adjusting the speed of the aircraft in the ATC system to avoid airspace regions that have reduced capacities, primarily due to dynamically changing weather conditions. This work has been extended in Bertsimas et al. [2008, 2011b] where different delay causes with respect to all flight phases of a flight have been included in a single optimization problem. Agustín et al. [2012a,b] studied the deterministic and stochastic TFMRP, where flight cancellation and re-routing, and arrival and departure capacity at the airport are taken into account in order to reduce the ground holding cost and air delays imposed on flights. Furthermore, the stochastic counterpart is reformulated by a deterministic equivalent model (a medium-scale mixed 0–1 model) considering the uncertainties of departure and arrival capacity of the airports, the capacity of sectors, and flight demands. More recently, Chen et al. [2017] proposed a polynomial approximation-based chance-constrained optimization method to address the uncertainty of capacity of sectors for the TFMRP. Sandamali et al. [2017] introduced a flight routing and scheduling model taking into account the uncertainty due to aircraft departures.

Flight Level Allocation or Flight Level Assignment (FLA) is another important approach to reduce the average fuel consumption per flight and to improve total

time savings per flight [Li and Trani, 2018]. Nace et al. [2003] proposed a linear programming approach for route and flight-level assignment in a trajectory-based ATM environment. Barnier and Brisset [2004] investigated the problem of flight level allocation considering direct routes only and vertically separate intersecting ones by allocating distinct flight levels. Abad and Barrington Clarke [2004] proposed en-route flight level allocation for aircraft to mitigate air traffic congestion and airline operating costs in airspace corridors. Constans et al. [2005] applied a genetic technique to the tactical Flight Level Assignment. Vela et al. [2009] proposed a compact formulation of a complete optimization model for speed control and flight level assignment to reduce fuel burn over time horizons between 15-45 minutes. The combining of flight level allocation and ground holding Barnier and Allignol [2011] was investigated to reduce the 4 Dimensions (4D) trajectory-based conflict. Allignol et al. [2012] proposed a flight level allocation schema to avoid the conflicts occurring during the cruise phase of intersecting flights. Soler-Arnedo et al. [2013] studied the contrail sensitive 4D trajectory planning problem performing the permitted step climbs to change flight level in order to minimize the fuel consumption and Carbon dioxide (CO2) emissions. Moreover, a two-step hybrid metaheuristic is proposed to solve the flight level allocation problem in order to avoid most losses of separation occurring between cruising flights before running the automated conflict resolution. More recently, Gimenez-Guzman et al. [2020] study the joint optimization of fuel consumption and Flight Level Assignment using graph coloring. However, most of them reformulated the flight level allocation/assignment problem considering only the crossing conflict (Figure 3.4.3a) at the same level (whereas trailing conflict (Figure 3.4.3b), converging conflict (Figure 3.4.3c) and hybrid conflict (Figure 3.4.3d) may encounter), without consideration of data uncertainties. To tackle the uncertainty of conflict around crossing waypoints, Constans et al. [2004] studied the optimal Flight Level Assignment taking account of uncertainty determination of flight crossing time at a given point. Klopfenstein and Nace [2008] introduced a mathematical model for the robust FLA via a chance-constrained optimization approach and proposed a fast approximation framework to solve the robust FLA efficiently, assuming the induced cost due to resolution of potential conflict is uncertain and bounded. Fundo et al. [2018] have investigated the robust FLA problem under the angle of uncertainty due to flight departure time delays. Specifically, the statistical delay model

of departures has been included in the solution method to improve the accuracy of the solution.

3.3 Contribution and organization

In this paper, we study the robust Flight Level Assignment problem dealing with uncertainty in flight departure time. The contribution of this work is twofold. We investigate first the FLA problem and its Chance-Constrained variant. A compact deterministic mathematical formulation and the Chance-Constrained Programming (CCP) counterpart problem are given, and a heuristic approach is provided. In essence, the approach stands in separating the problem by level altitudes and solve each of them consecutively in a certain order. A second contribution that can be drawn is the specific approach to solving the CCP FLA problem associated with each flight level. Using the results of Klopfenstein, we reformulate the CCP as an Integer Linear Programming (ILP) one where the probability constraints are replaced by the worst case like Soyster model ones [Soyster, 1973]. These constraints are added as they are needed through a constraint generation approach until the desired feasibility probability is reached. All this gives a practical approach that may find application in a class of CCP problems. Another novelty of the approach is the method used to check the feasibility probability of constraints. We consider the case with uncertain parameters following a truncated Gaussian Mixture Model (GMM) distribution and propose an approximated method to estimate the feasibility probability of the solution. This may be of high interest in practice since the GMM is a powerful tool to capture the characteristics distribution of a large number of real situations.

The remainder is organized as follows: Section 3.4 reports a brief description of airspace configuration. Section 3.5 presents the compact deterministic mathematical formulation of the FLA problem and its $\mathcal{NP}$-hardness is present. The robust counterpart of the FLA problem taking account of the uncertainty of the flight departure time delay is established in Section 3.6. We report in Section 3.7 the general procedure to tackle the robust FLA problem. This procedure includes a heuristic estimation method is proposed to estimate the feasibility probability of each obtained solution from the corresponding robust subproblem. Section 3.8 reports the computational results used to validate the proposed approaches. Some

discussion on the generality of the proposed method is reported together with concluding remarks in Section 3.9.

3.4 Airspace configuration

We present here a brief description of airspace configuration including flight mission, separation minima for security, potential conflict, and the computation of induced delay in line with a resolution of pairwise conflict. We assume known or given what follows.

- a set of flights with their traffic trajectory;
- a set of waypoints;
- a set of feasible flight levels for each concerned flight;
- an acceptable upper bound of en-route ATFM delay for each flight;
- the trajectory between two waypoints is straight;
- two aircraft are assumed having a potential conflict at a crossing waypoint if and only if there is a non null probability that the minimum separation distance between them is less than minimum separation (denoted with S);
- a potential conflict between two flights is occurred during the "Cruise Phase" for flights flying at the same level;
- pairwise potential conflicts are assumed independent events;
- flight departure delays are propagated constantly through the flight.

Flight phases A flight is subdivided into different phases as illustrated in Figure 3.4.1. The conflicts concerned in this work occur during the "Cruise Phase" for flights flying at same level. An induced en-route ATFM delay may occur due to the flight departure delay as illustrated in Figure 3.4.2. A set of feasible cruise altitudes exist depending on the aircraft type of which the operating costs are minimized at the optimum cruise altitude.

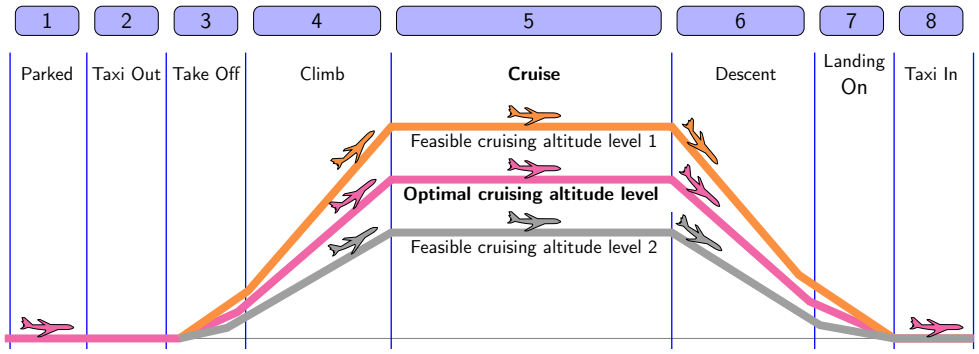


Figure 3.4.1 – The mission of a flight

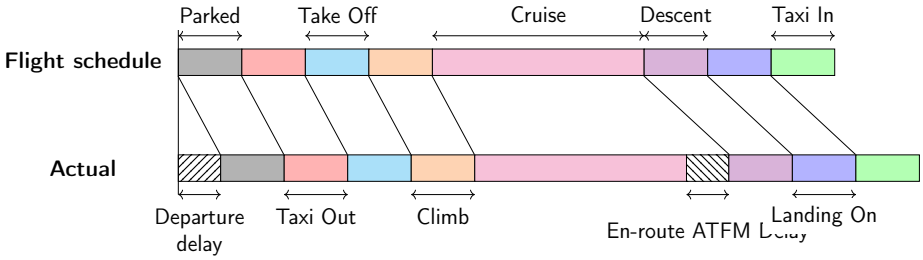


Figure 3.4.2 – Air transport delay

Separation Minima Separation minima defines the minimum separation distance between two aircraft for a safety consideration, including vertical separation, lateral separation and longitudinal separation. In this study, for the vertical separation we use the Reduced Vertical Separation Minima (RVSM) which is established between Flight Level (FL)290[1] and FL410 in order to increase the airspace capacity. In RVSM airspace, the minimum vertical separation is 10 FL in contrast to 20 FL in non-RVSM airspace. The lateral separation minima describe the minimum separation between aircraft in a horizontal plane such that the spacing between aircraft is never less than a specified amount where lateral separation is applied for aircraft following different tracks while the longitudinal separation is applied for aircraft following the same, converging or diverging tracks. When surveillance systems are used, the minimum separation prescribed in [ICAO, 2016] is 5 nm[2]. With respect to the flight level assignment, the so-called "Semicircular/hemispheric" rule is also applied. The eastbound (respectively, southbound) flights (Magnetic Track

[1]FL290 = 29 000 ft
[2]1 nm = 1.852 km

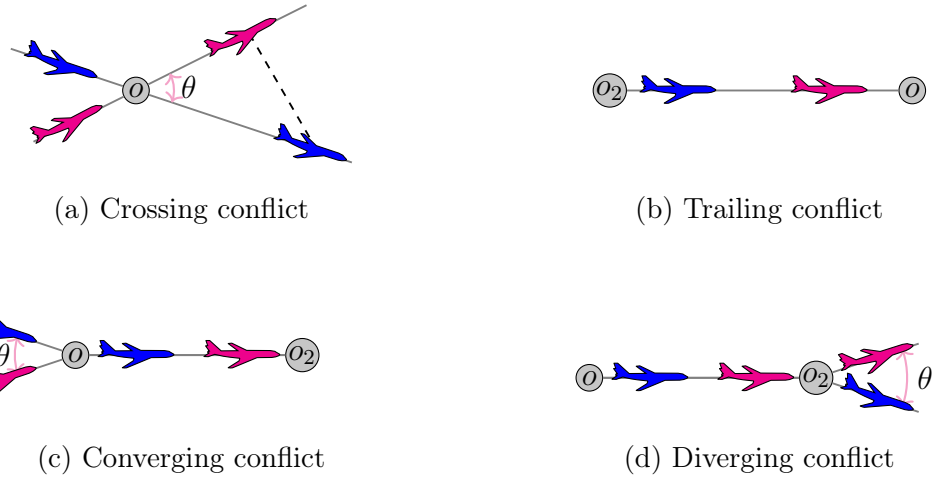
0 to π) use odd FL (e.g., FL290, FL310) and westbound (respectively, northbound) flights (Magnetic Track π to 2π) use even FL (e.g., FL300, FL320) for the airspace of Europe. However, 20 FL intervals are resumed to separate same-direction aircraft and only odd FLs are assigned at FL410 or above, depending on the direction of flight: Magnetic Track 0 to π uses FL410, FL450, etc; Magnetic Track π to 2π uses FL430, FL470, etc.

Potential conflict We identify four types of pairwise air conflicts due to a loss of separation at cruise phase (see Figure 3.4.3).

- Crossing conflict—may occur if the two aircraft cross at some point o and diverge afterwards.
- Trailing conflict—may occur if two aircraft follow the same route, as it is often the case on airways.
- Converging conflict—may occur if the two aircraft join at some point and remain the same afterwards, at least for a portion of the flight. Two aircraft are involved simultaneously in a crossing and trailing conflict around the waypoint o.
- Diverging conflict—may occur if the two aircraft share the same track and diverge afterwards. Two aircraft are involved simultaneously in a crossing and trailing conflict around the waypoint o_2.

In practice, all conflicts are solved through predetermined conflict solving procedures. We have analyzed these situations and computed the delay associated with such procedure as described in the following:

En-route ATFM delay ω_{ij} of potential conflict for two aircraft. Figure 3.4.3 shows us geometrically the conflict situation during cruise phase assuming the trajectory between two waypoints is straight. We assume that the potential conflict occurred at the first time for two aircraft is the only one taken into account if there are more than one potential conflict occur between the two aircraft. (indeed having more than one is practically improbable). Let assume that some potential conflict has encountered at waypoint o, and θ is the crossing angle. Furthermore,



(a) Crossing conflict

(b) Trailing conflict

(c) Converging conflict

(d) Diverging conflict

Figure 3.4.3 – En-route potential conflict between two aircraft cruising at same level

for sake of simplicity, the delay at other phases of a flight due to the departure delay is omitted, we consider only the en-route ATFM delay which is caused by the flight departure delay of flights. Note t_i^o, t_j^o is the time that aircraft i and j passes the conflict point o, respectively. Let t_{ij}^{msd} specify the minimum separation time instead of minimum separation distance for two aircraft to pass safely the conflict point, which is computed exhaustively in Appendix I. The induced en-route ATFM delay ω_{ij} of resolution for pairwise conflict is then formulated by:

$$\omega_{ij} = (t_{ij}^{\text{msd}} - t_i^o + t_j^o)\mathbb{1}(t_{ij}^{\text{msd}} - t_i^o + t_j^o)_{[0,\infty)}\mathbb{1}(t_i^o - t_j^o)_{[0,\infty)} \in [0, t_{ij}^{\text{msd}}], \tag{3.1}$$

where $\mathbb{1}(x)_A$ is an indicator function: $\mathbb{1}(x)_A = \{1, x \in A; 0, x \notin A\}$, the second term $\mathbb{1}(t_{ij}^{\text{msd}} - t_i^o + t_j^o)_{[0,\infty)}$ specifies that the induced en-route ATFM delay of associated flight due to resolution should be positive which in turn means that there exists a potential conflict, and $\mathbb{1}(t_i^o - t_j^o)_{[0,\infty)}$ denotes whether the aircraft i arrives latter than j at the potential conflict point o. The above formula can be expressed as a function of departure times and we assume that the flight departure time delay follows a GMM, then ω_{ij} follows a truncated GMM distribution.

3.5 The flight level assignment problem

In this section, we show first the $\mathcal{NP}$-hardness of the FLA problem via a 3-Graph (vertex) Coloring (3-GC) problem. A corresponding compact formulation is then described mathematically.

3.5.1 Complexity issues

The complexity of the FLA problem is $\mathcal{NP}$-hard in the strong sense. The proof is based on the 3-GC known to be $\mathcal{NP}$-Complete in the strong sense even for a planar graph [Wegener, 2005]. Before the details of how 3-GC can be polynomially reduced to the FLA decision problem, we first define formally these problems.

The 3-GC decision problem Given an undirected graph $G = (V, E)$ with V denoting the set of vertices and E specifying the set of edges, a set of 3 colors denoted C and the coloring function $c : V \to C$, is there any assignment of colors from C such that we have $c(u) \neq c(v)$ for any two adjacent vertices $u, v \in V$?

The FLA decision problem. Given a set of flights denoted $\mathcal{F}$, a set of eligible flight levels for each flight denoted $\mathcal{L}_i$ (usually, $|\mathcal{L}_i| = 3$, one is the most preferred level, and the other two feasible candidate levels) included in a set $\mathcal{L}$ (clearly, $|\mathcal{L}| \geq 3$) , and a function $\mathcal{P}^l : \mathcal{F} \times \mathcal{F} \to \{0, 1\}$ (i.e., for each pair (i, j) of flights flying at the same level $l \in \mathcal{L}$, $\mathcal{P}^l(i, j)$ takes value 0 if there is no en-route conflict between them and 1 if there is a potential one), is there any assignment of flights to their eligible levels such that we have $\mathcal{P}(f, f') = 0$ for any pair (f, f') of flights the same level? More in specifically, the airspace is defined by a set of waypoints (that is a reference point in the airspace used for purposes of navigation), and we assume that trajectory between two waypoints is straight. Moreover, two flights passing through the same waypoint within a short interval of time are assumed in potential conflict.

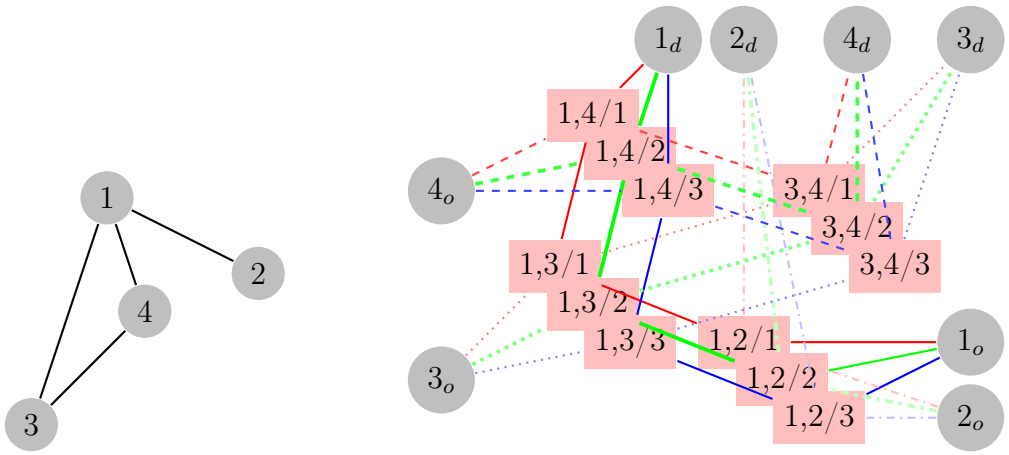
Proposition 3.5.1 *The FLA problem is $\mathcal{NP}$-complete in the strong sense.*

Proof We will show that for any instance of the 3-GC decision problem, we can construct in polynomial time an instance of the FLA decision problem accepting a solution if and only if the 3-GC accepts a solution. Let first show that the FLA

decision problem is in $\mathcal{NP}$, that is we can check in polynomial time with respect to the size of the problem instance that a given solution accepts a yes answer to the question [Wegener, 2005]. It is obvious that a verification of existence of potential conflict for a given assignment of flights to levels can be done in polynomial time. We just need to check if for each pair of flights assigned at the same level there is some potential conflict, which gives at most $n(n-1)/2$ verifications ($n = |\mathcal{F}|$).

Let us consider in the following the equivalence of both decision problems. For any instance of 3-GC we can construct an instance of FLA as follows: Let be given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with respect to an instance of the 3-GC problem. Let us now define a bijection $\mathcal{A}$ between vertices in $\mathcal{V}$ and flights in the instance of FLA that we are supposed to construct, that is $|\mathcal{F}| = |\mathcal{V}|$. Further, we have three levels in the FLA instance that we will build, which corresponds to the number of colors needed for the graph. Let us note these colors r, g, b and build a correspondence with levels in the FLA problem. Let us use now the vertices and edges in $\mathcal{G}$ to build an airspace network with respect to the FLA instance. The airspace network will be composed of $|\mathcal{V}|$ flights, $|\mathcal{E}|$ waypoints and only 3 flight levels ($|\mathcal{L}| = 3$, for sake of simplicity). For each flight f we have a specific origin and destination airport, denoted respectively f_o and f_d. The waypoints are identified by a pair of flights corresponding to the extremities of edges in $\mathcal{G}$, for instance some waypoint in level l corresponding to edge $e = (v, v') \in \mathcal{E}$ will be denoted with $(\mathcal{A}(v), \mathcal{A}(v'))_l = (f, f')_l$. We suppose that at each waypoint there is a potential en-route conflict between the corresponding flights when they are at the same level. Then, with each flight $f = \mathcal{A}(v)$ at level l we associate a route from f_o to f_d passing through waypoints corresponding the adjacent links to node v in the initial graph $\mathcal{G}$. The waypoints are traversed in increasing order of the corresponding adjacent nodes in $\mathcal{G}$ (e.g., $< 1, 2 >$). Given an instance of the 3-GC decision problem (see Fig.3.5.4), the route of all flights involved in corresponding instance of the FLA decision problem are summarized as follows:

- Flight 1: $1_o \rightarrow < 1, 2 > \rightarrow < 1, 3 > \rightarrow < 1, 4 > \rightarrow 1_d$
- Flight 2: $2_o \rightarrow < 1, 2 > \rightarrow 2_d$
- Flight 3: $3_o \rightarrow < 1, 3 > \rightarrow < 3, 4 > \rightarrow 3_d$
- Flight 4: $4_o \rightarrow < 1, 4 > \rightarrow < 3, 4 > \rightarrow 4_d$



(a) The 3-GC instance (b) The corresponding FLA instance

Figure 3.5.4 – FLA instance construction from 3-GC

It is straightforward that this construction can be done in polynomial time. Let us show now that solving the 3-GC problem provides a valid assignment, i.e., avoiding conflicts, for the corresponding instance of FLA. Indeed, let c be a valid color assignment to vertices in $\mathcal{G}$. Then, if we choose for each flight f in the airspace network the level l which corresponds to color l used to color vertex $\mathcal{A}^{-1}(f) \in \mathcal{V}$, we will never find two flights in the same level passing through the same waypoint, and hence there are no conflicts. Vice-versa, if there is a solution avoiding conflicts in our airspace network, let associate with flights on level l the color l. By using this coloring for the corresponding vertices in $\mathcal{G}$ we obtain a solution for this instance of 3-GC. As the 3-GC problem is $\mathcal{NP}$-complete in the strong sense, this proves that the FLA is as well, concluding thus the proof.

3.5.2 Compact mathematical formulation

Sets and indices:

$\mathcal{F}$: The set of flights, indexed by i, j.

$\mathcal{L}$: The set of flight levels, indexed by l.

$\mathcal{F}^l$: The subset of flights allowed flying at the flight level $l \in L$.

$\mathcal{L}_i$: The subset of eligible flight levels for each flight $i \in \mathcal{F}$, $|\mathcal{L}_i| = 3$. For instance for an aircraft of type Airbus A320, the three preferred level flights are FL390, FL410 and FL370.

$\mathcal{S}_i^l$: The subset of flights having a potential conflict with flight i at level l.

Parameters:

$\overline{W}_i$: The maximal acceptable en-route ATFM delay for a given flight i.

η_i: The penalizing cost for a flight *cancellation* (it happens when the total induced en-route ATFM delays exceeding this maximum value $\overline{W}_i$). This is called cancellation in the sense that this flight will not be counted as realized and its potential conflicts with other flights not accounted.

α_i: The average unit cost per minute and per flight of en-route ATFM delay.

b_i^l: The estimated profit corresponding to assignment of flight i at level l.

ω_{ij}: The induced en-route ATFM delay of flight i when resolving a potential conflict with j flying at the same level l.

M_i^l: A sufficient large number, e.g. $M_i^l = \Sigma_{j\in\mathcal{S}_i^l}\omega_{ij}$.

Variables:

x_i^l: A binary variable taking value 1 if the flight i flies on level l, 0 otherwise.

y_i: A positive continuous variable indicating the cumulative induced en-route ATFM delay for an assigned flight.

Using the above-mentioned notation, mathematical model associated with the deterministic compact FLA problem denoted by $\mathcal{CP}$ is described as below:

$$\max \sum_{i\in\mathcal{F}}\sum_{l\in\mathcal{L}_i}(b_i^l - \alpha_i y_i)x_i^l - \sum_{i\in\mathcal{F}}\eta_i(1 - \sum_{l} x_i^l) \tag{3.2}$$

$$\text{s.t.} \sum_{j\in\mathcal{S}_i^l}\omega_{ij}x_j^l + M_i^l(x_i^l - 1) \leq y_i \qquad \forall l \in \mathcal{L}, \forall i \in \mathcal{F}^l \tag{3.3}$$

$$\sum_{l\in\mathcal{L}_i} x_i^l \leq 1 \qquad \forall i \in \mathcal{F} \tag{3.4}$$

$$0 \leq y_i \leq \overline{W}_i \qquad \forall i \in \mathcal{F} \tag{3.5}$$

$$x_i^l \in \{0,1\} \qquad \forall l \in \mathcal{L}, \forall i \in \mathcal{F}^l \tag{3.6}$$

The objective function maximizes the assignment profit as well as minimizes the penalizing cost due to cancellation and en-route ATFM delays. Clearly the above objective function can be written as $\sum_{i\in\mathcal{F}}\sum_{l\in\mathcal{L}_i}(b_i^l+\eta_i-\alpha y_i)x_i^l$. Constraints (3.3) calculate the cumulative induced delay of each flight i assigned at level l and specify that the flight is canceled if the induced en-route ATFM delay exceeds the maximum acceptable upper bound $\overline{W}_i$. Constraints (3.4) specify that each flight

is assigned at most to one of its eligible levels. Constraints (3.5) and (3.6) define the feasible domain of decision variables.

The above model is nonlinear due to the bilinear terms $y_i x_i^l$, hence a linearization is applied thanks to McCormick Envelopes [McCormick, 1976]:

$$z_i^l \geq 0;\ z_i^l \leq \overline{W}_i x_i^l;\ z_i^l \leq y_i;\ z_i^l \geq \overline{W}_i(x_i^l - 1) + y_i \tag{3.7}$$

where z_i^l is the product of y_i and x_i^l.

3.6 The Robust counterpart of FLA problem

3.6.1 Modeling the Robust FLA problem

In a real ATM environment, flight departure delays are usually uncertain due to various sources of nature, such as airport weather conditions, airport capacity, airport disruption, and airport staffing. The Gaussian Mixture Model is a powerful tool to capture main characteristics of departure delay distribution, as investigated by [Tu et al., 2008]. We apply the coefficients of aforementioned GMM (present in Table 3.6.1) for the distribution of flight departure time delay in this study, where components C_1 and C_2 capture mainly the negative value of departure delays where C_2 has a higher peak to shape the skewness, component C_3 specifies some medium delays and C_4 accounts for the very large delays. All this makes

Table 3.6.1 – Coefficients of GMM for flight departure time delays

Component	C_1	C_2	C_3	C_4
c_i	0.37	0.40	0.15	0.08
μ_i	-17.15	-7.31	19.57	69.13
σ_i^2	88.20	89.33	1007.73	3926.00

ω_{ij} parameters to take random values in given interval. So, from now ω_{ij} express the random parameter varying in $[0, \overline{\omega}_{ij}]$, where $\overline{\omega}_{ij} = t_{ij}^{msd}$ gives the maximum induced en-route ATFM delay occurring in case of conflict between flights i and j. Furthermore, ω_{ij} follows a truncated GMM distribution since the flight departure time delay is GMM distributed (see Equation (3.1) and Appendix I).

Assuming separate probability conditions and taking account of above uncertainty of flight departure time delay, the compact mathematical formulation of the

robust FLA problem denoted by $\mathcal{RP}$ can be expressed via the Chance-Constrained Programming as follows:

$$\max \quad \sum_{i\in\mathcal{F}}\sum_{l\in\mathcal{L}_i}(b_i^l+\eta_i-\alpha_i y_i)x_i^l-\sum_{i\in\mathcal{F}}\eta_i \tag{3.8}$$

$$\text{s.t.} \quad \mathsf{P}\left(\sum_{j\in\mathcal{S}_i^l}\omega_{ij}x_j^l+M_i^l(x_i^l-1)\le y_i\right)\ge 1-\epsilon \qquad \forall l\in\mathcal{L},\forall i\in\mathcal{F}^l \tag{3.9}$$

$$\sum_{l\in\mathcal{L}_i}x_i^l\le 1 \qquad \forall i\in\mathcal{F} \tag{3.10}$$

$$0\le y_i\le \overline{W}_i \qquad \forall i\in\mathcal{F} \tag{3.11}$$

$$x_i^l\in\{0,1\} \qquad \forall l\in\mathcal{L},\forall i\in\mathcal{F}^l \tag{3.12}$$

where probabilistic constraints (3.9) ensure for each flight that the sum of induced en-route ATFM delays will not exceed the given upper bound of acceptable delay with a probability at least $1-\epsilon$ (ϵ gives the desired infeasibility (violation) tolerance of constraint (3.3)). For sake of simplicity, we will allow ourselves to use the same notation for ω_{ij}, M_i^l as in $\mathcal{CP}$, but here ω_{ij} is a random value (truncated GMM distributed) bounded in $[0,\overline{\omega}_{ij}]$ due to the uncertainty of flight departure delay and $M_i^l=\Sigma_{j\in\mathcal{S}_i^l}\overline{\omega}_{ij}$. Clearly the problem is difficult and becomes intractable even for moderate instances. Our solution approach stands in two paradigms: first, we proceed to a heuristic decomposition approach separating the problem per flight level, and then the specific (CCP) FLA problem for each single level is solved through Robust Optimization (RO) methods.

The key issue of above solution approach is how to formulate the associated robust subproblem, called $\mathcal{RP}^l$, into an ILP problem and solve it efficiently. This is in the focus of the next section.

3.6.2 Subproblem associated with a single flight level ($\mathcal{RP}^l$)

Before detailing the mathematical formulation of this subproblem, let give some precision on the notation. As there is no need to distinguish flight levels, the binary variable x_i^l is now replaced by x_i, and as before it takes value 1 when the flight i flies on level l and 0 otherwise. For sake of simplicity, we will allow ourselves to use the same notation for $\mathcal{F}^l$ as in $\mathcal{CP}$, but here it groups only flights 1) being eligible to fly at this level and not yet assigned to other flight level, or 2) whose

most preferred flight level is the processing one. Notice that y_i are now parameters of cumulative induced en-route ATFM delay for the corresponding flight i in $\mathcal{RP}^l$, (denoted as vector $y_{\mathcal{F}^l}$). The mathematical formulation associated with the CCP FLA restricted to level l denoted by $\mathcal{RP}^l(\mathcal{F}^l, y_{\mathcal{F}^l})$ then follows:

$$\max \quad \sum_{i\in\mathcal{F}^l}(b_i^l + \eta_i - \alpha_i y_i)x_i \tag{3.13}$$

$$\text{s.t.} \quad \mathsf{P}\left(\sum_{j\in\mathcal{S}_i^l}\omega_{ij}x_j + M_i^l(x_i - 1) \leq y_i\right) \geq 1-\epsilon \qquad \forall i \in \mathcal{F}^l \tag{3.14}$$

$$x_i \in \{0,1\} \qquad \forall i \in \mathcal{F}^l \tag{3.15}$$

The above $\mathcal{RP}^l(\mathcal{F}^l, y_{\mathcal{F}^l})$ is still a very difficult CCP one. Recall first that CCP is a specific model of stochastic optimization looking to optimize the objective, given an infeasibility probability tolerance. It is a very hard optimization area and one way to tackle it, largely studied these last two decades, is to go through Robust Optimization. In RO, the Soyster model gives a very conservative model preserving the feasibility at all scenarios. The Γ-robustness [Bertsimas and Sim, 2004] model looks for a better trade off between the feasibility and optimality. Concerning the probabilistic constraint (3.14), we have $\epsilon = \exp(-\Gamma_i^2/(2|\mathcal{S}_i^l|))$ from Γ-robustness model if the ω_{ij} is independently symmetrically distributed. However, this model is not directly usable because of asymmetric uncertain interval of ω_{ij}. In practice, given the desired value of ϵ for each concerned flight i, we can not easily compute the corresponding coefficient Γ_i for the involved constraint of each flight i in each associated subproblem in our study. On the contrary, the feasibility probability $1-\epsilon$ for the involved probabilistic constraint can be posteriorly estimated with a realized value of the corresponding Γ_i by sampling the possible value of ω_{ij}. A robust solution can be obtained through Γ-robustness, but it may be a time-costly solution due to the high combination of Γ_i values for the involved constraints of associated flights and the probability estimation sampling procedure. Hence, to solve the above problem we have opted to use the model introduced in Klopfenstein [2009], which proposes a simple heuristic algorithm to find good solutions to general chance-constrained integer linear problems. In this model, the author has introduced a parameter vector $\gamma \in [0,1]^{|\mathcal{F}^l|}$ which allows tuning the robustness of the solution in a convenient way. Applying this idea, we obtain the following

alternative model:

$$\max \quad \sum_{i\in\mathcal{F}^l}(b_i^l+\eta_i-\alpha_i y_i)x_i \tag{3.16}$$

$$\text{s.t.} \quad \min\left\{\sum_{j\in\mathcal{S}_i^l}\overline{\omega}_{ij}x_j, \gamma_i\sum_{j\in\mathcal{S}_i^l}\overline{\omega}_{ij}\right\}+M_i^l(x_i-1)\leq y_i \qquad \forall i\in\mathcal{F}^l \tag{3.17}$$

$$x_i\in\{0,1\} \qquad \forall i\in\mathcal{F}^l. \tag{3.18}$$

Let us focus on the robust constraint associated with flight i in (3.17). Either we consider the worst case (maximum induced en-route ATFM delay, $\overline{\omega}_{ij}$, for all flights $j\in\mathcal{S}_i^l$ in conflict with flight i), or we have a constraint: $\gamma_i\sum_{j\in\mathcal{S}_i^l}\overline{\omega}_{ij}+M_i^l(x_i-1)\leq y_i$. In this latter case, two sub-cases occur: when $\gamma_i\sum_{j\in\mathcal{S}_i^l}\overline{\omega}_{ij}>y_i$, then $x_i=0$; when $\gamma_i\sum_{j\in\mathcal{S}_i^l}\overline{\omega}_{ij}\leq y_i$, we have a dummy constraint which can be ignored. These three cases are in fact summarized in the two following ones:

- either flight i has cumulative induced en-route ATFM delays less than the given en-route ATFM delay y_i and no constraint is necessary to model this situation;
- or flight i is associated with maximum induced en-route ATFM delay with flights $j\in\mathcal{S}_i^l$, i.e., $\sum_{j\in\mathcal{S}_i^l}\overline{\omega}_{ij}x_j+M_i^l(x_i-1)\leq y_i$.

Hence, the analysis of the above robust model leads to a new one, which is very simple. Indeed, for a given value of γ_i we know in advance if the constraint corresponding to flight i is necessary to be put in the model or not. Let denote with $\mathcal{I}_c\subseteq\mathcal{F}^l$ a subset of concerned flights with respect to a given vector γ. In this way, instead of vector γ we use the subset $\mathcal{I}_c$ as a parameter enabling to tune robustness. The corresponding problem denoted by $\mathcal{RP}^l(\mathcal{I}_c,\mathcal{F}^l,y_{\mathcal{F}^l})$ is then formulated as follows:

$$\max \quad \sum_{i\in\mathcal{F}^l}(b_i^l+\eta_i-\alpha_i y_i)x_i \tag{3.19}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{S}_i^l}\overline{\omega}_{ij}x_j+M_i^l(x_i-1)\leq y_i \qquad \forall i\in\mathcal{I}_c \tag{3.20}$$

$$x_i\in\{0,1\} \qquad \forall i\in\mathcal{F}^l \tag{3.21}$$

With respect to set $\mathcal{I}_c$ considered, the size of the above ILP varies between few constraints (initially $\mathcal{I}_c$ is empty) and all constraints (i.e., $\mathcal{I}_c=\mathcal{F}^l$). The set $\mathcal{I}_c$

is said valid if the constraints (3.22) are satisfied for the obtained solution x^* of $\mathcal{RP}^l(\mathcal{I}_c, \mathcal{F}^l, y_{\mathcal{F}^l})$.

$$\mathsf{P}\left(\sum_{j\in\mathcal{S}_i^l}\omega_{ij}x_j^* + M_i^l(x_i^* - 1) \leq y_i\right) \geq 1-\epsilon \qquad \forall i \in \mathcal{F}^l \setminus \mathcal{I}_c \tag{3.22}$$

A natural approach to solve the subproblem $\mathcal{RP}^l$ is a constraint generation approach following the strategy proposed in [Klopfenstein, 2009]. The idea is to start with an empty set of constraints ($\mathcal{I}_c = \emptyset$), check the obtained and add the most violated one into the subset $\mathcal{I}_c$ until the feasibility probability of the obtained solution, the constraint (3.22) for each concerned flight, is satisfied.

Algorithm 1: Heuristic method for solving $\mathcal{RP}^l$

Input: $\mathcal{F}^l$: A set of flights eligible for processing level l, y_i: The given cumulative induced delay for associated flights $f \in \mathcal{F}^l$.
Output: x^*: An optimal solution of flight level assignment

```
Set I_c ← ∅;
Unsolved ← True;
do
    Solve RP^l(I_c, F^l, y_{F^l});
    Let x* be the optimal solution found;
    Estimate the concerned feasibility probability of obtained solution x*;
    if Constraints (3.22) for all concerned flights in F^l \ I_c are respected
     then
        Fix all assigned flights;
        Unsolved← False;
    else
        Mark index i as corresponding to the highest violation of
         constraints (3.22);
        I_c ← I_c ∪ {i};
    end
while Unsolved;
return Last obtained solution x*.
```

In other words, by refining the set $\mathcal{I}_c$ such that constraint (3.22) is satisfied for obtained solution x^* from $\mathcal{RP}^l(\mathcal{I}_c, \mathcal{F}^l, y_{\mathcal{F}^l})$, we then have a certified feasible solution for the associated robust subproblem. As highlighted in Algorithm 1, a

solution is obtained initially by an empty set $\mathcal{I}_c$. For each obtained solution, the concerned feasibility probability is estimated by the methods detailed in Section 3.6.3. Set $\mathcal{I}_c$ is updated iteratively by inserting the flight i that the associated constraint (3.22) is the most violated and the associated problem is resolved until the constraints (3.22) for all concerned flights are satisfied. Finally, a robust solution is obtained for the corresponding subproblem $\mathcal{RP}^l$, where the robustness (feasibility probability) of the solution is guaranteed by the minimum value of the posteriorly estimated feasibility probability over all associated constraints.

3.6.3 Estimation of feasibility probability of solution of $\mathcal{RP}^l$

Given a solution of an instance of $\mathcal{RP}^l(\mathcal{I}_c, \mathcal{F}^l, y_{\mathcal{F}^l})$, only constraints (3.22) may be violated due to uncertainty of ω_{ij}. Observe first that the constraint is dummy if x_i takes value of zero, which is always feasible regardless the value of ω_{ij}. Therefore, we restrict ourselves in ensuring that the feasibility probability of the concerned constraint is at least $1-\epsilon$ (i.e., $\mathsf{P}(\sum_{j\in\mathcal{S}_i^l}\omega_{ij}x_j \leq y_i) \geq 1-\epsilon$) for all $x_i = 1$ and $i \in \mathcal{F}^l \setminus \mathcal{I}_c$. The flowchart to estimate the feasibility probability of an obtained solution of $\mathcal{RP}^l$ by different methods is highlighted in Figure 3.6.5, Figure 3.6.6, and Figure 3.6.8. There are three methods used to estimate the feasibility probability which are: the first one is using Hoeffding's Inequality, the second is based on Monte-Carlo simulation and the last is a heuristic estimation method evaluating the sum of random variables following a truncated GMM distribution.

Conservative robust method For a comparison purpose, we consider first the Soyster model [Soyster, 1973], which solves a specific deterministic variant instead of seeking the best solution remaining feasible over all possible scenarios. We refer it as Soyster feasibility probability estimation method (Soyster method). The obtained optimal solution is surely feasible over all scenarios as all data takes their worst-case value (i.e., $\overline{\omega}_{ij}$). The biggest drawback is that it may lead to a costly solution that strays far away from the optimal one of a given scenario due to the over-conservatism.

Hoeffding's Inequality method Hoeffding's Inequality [Hoeffding, 1994] is a result in probability theory that gives an upper bound on the probability for

the sum of identical independent random variables to deviate from its expected value. We hereby refer it as Hoeffding's inequality feasibility probability estimation method (Hoeffding method). Let us recall the Hoeffding's Inequality: Let $\mathcal{X}_1, \mathcal{X}_2, \cdots \mathcal{X}_n$ be the identical independent random variables. Assume that $\mathcal{X}_i$ are almost surely bounded; that is , for $1 \leq i \leq n$ we have $\mathsf{P}(\mathcal{X}_i \in [a_i, b_i]) = 1$ for some finite a_i, b_i. Let be $S = \sum_{i=1}^{n} \mathcal{X}_i$ and $\mathbb{E}[S]$ its expected value. Then we have the inequality:

$$\mathsf{P}(S - \mathbb{E}[S] \geq nt) \leq exp\left(\frac{-2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right) \qquad \forall t > 0 \tag{3.23}$$

We apply the Hoeffding's Inequality. Noting that,

$$\mathsf{P}\left(\sum_{j\in\mathcal{S}_i^l} \omega_{ij}x_j \geq y_i\right) = \mathsf{P}\left(\sum_{j\in\mathcal{S}_i^l} \omega_{ij}x_j - \sum_{j\in\mathcal{S}_i^l} \mathbb{E}[\omega_{ij}]x_j \geq y_i - \sum_{j\in\mathcal{S}_i^l} \mathbb{E}[\omega_{ij}]x_j\right) \tag{3.24}$$

we obtain:

$$\mathsf{P}\left(\sum_{j\in\mathcal{S}_i^l} \omega_{ij}x_j \geq y_i\right) \leq exp\left(-\frac{2\left(y_i - \sum_{j\in\mathcal{S}_i^l} \mathbb{E}[\omega_{ij}]x_j\right)^2}{\sum_{j\in\mathcal{S}_i^l} \overline{\omega}_{ij}^2 x_j^2}\right) = \epsilon \tag{3.25}$$

When $t = y_i - \sum_{j\in\mathcal{S}_i^l} \mathbb{E}[\omega_{ij}]x_j \leq 0$, the probability $\mathsf{P}(\sum_{j\in\mathcal{S}_i^l} \omega_{ij}x_j \leq y_i)$ is set to zero. Whereas, in case that y_i is bigger than the sum of all upper bounds of random variables, then the probability is surely one. Thus, we obtain a piece-wise probability function as follows:

$$\mathsf{P}\left(\sum_{j\in\mathcal{S}_i^l} \omega_{ij}x_j \leq y_i\right) = \begin{cases} 0, \text{if } y_i \leq \sum_{j\in\mathcal{S}_i^l} \mathbb{E}[\omega_{ij}]x_j \\ 1, \text{if } \sum_{j\in\mathcal{S}_i^l} \overline{\omega}_{ij}x_j \leq y_i \\ 1 - \epsilon, \text{otherwise} \end{cases} \tag{3.26}$$

The flow chart of estimating the feasibility probability of obtained solution by means of Hoeffding's Inequality is presented in Figure 3.6.5. For an obtained solution x^* from solving $\mathcal{RP}^l(\mathcal{I}_c, \mathcal{F}^l, y_{\mathcal{F}^l})$ within an off-the-shelf solver (e.g., CPLEX), we check the feasibility of constraint (3.22) by Equation (3.26) for each $x_i^* = 1, i \in \mathcal{F}^l \setminus \mathcal{I}_c$. The minimum probability of all above is then the feasibility probability of obtained solution x^*.

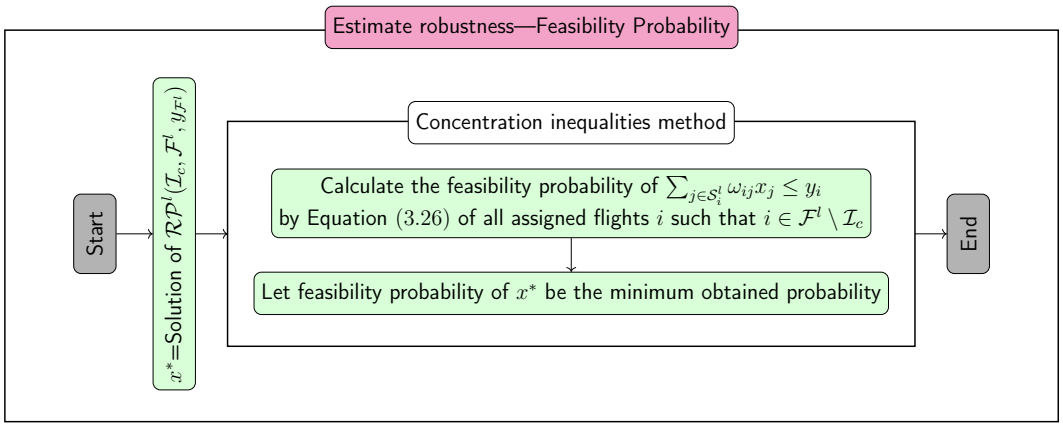


Figure 3.6.5 – Estimation of feasibility probability of obtained solution by the Hoeffding's Inequality

Monte-Carlo simulation method The Monte-Carlo Simulation method is frequently used in mathematical problems such as optimization, generating draws from a probability distribution. We refer it as Monte-Carlo simulation feasibility probability estimation method (MC method). The main idea behind this method is randomly generating a sufficiently large number of scenarios to obtain numerical results. In this study, we randomly generate the departure time delay following a Gaussian Mixture Model (see Table.3.6.1) for the concerned flights in each randomly generated scenario. We have assumed that for each flight, data uncertainty coming from departure time delay will be propagated constantly through the flight trajectory.

Note p the probability $\mathsf{P}\left(\sum_{j\in\mathcal{S}_i^l}\omega_{ij}x_j \leq y_i\right)$, N_{mc} is the number of scenarios for simulation, $1-\alpha$ is the confidence level of p, u_α is the corresponding quantile such that $\Phi(u_\alpha) = 1-\alpha$ and $\hat{p}$ is the frequency of event $\sum_{j\in\mathcal{S}_i^l}\omega_{ij}x_j \leq y_i$ of N_{mc} scenarios. Applying the central limit theorem [Rosenblatt, 1956] and Slutsky's theorem [Delbaen, 1998], we have:

$$\frac{\hat{p}-p}{\sqrt{\frac{\hat{p}(1-\hat{p})}{N_{mc}}}} \longrightarrow \mathcal{N}(0,1) \tag{3.27}$$

Then we deduce that:

$$\mathsf{P}\left(\frac{\hat{p}-p}{\sqrt{\frac{\hat{p}(1-\hat{p})}{N_{mc}}}} \leq u_\alpha\right) \geq 1-\alpha \Leftrightarrow \mathsf{P}\left(p \geq \hat{p} - u_\alpha\sqrt{\frac{\hat{p}(1-\hat{p})}{N_{mc}}}\right) \geq 1-\alpha \tag{3.28}$$

Note that $t = \frac{N_{mc}(1-\epsilon)+0.5u_\alpha^2}{N_{mc}+u_\alpha^2}$ and $t' = \frac{N_{mc}(1-\epsilon)^2}{N_{mc}(1-\epsilon)+0.5u_\alpha^2}$, we obtain:

$$\hat{p} - u_\alpha\sqrt{\frac{\hat{p}(1-\hat{p})}{N_{mc}}} \geq 1-\epsilon \Leftrightarrow \hat{p} \geq t + \sqrt{t^2 - t't} \tag{3.29}$$

Thus the concerned constraint is said feasible with probability $1-\epsilon$ with a confidence level at least $1-a$ when testing it for N_{mc} scenarios, and getting constraints (3.22) satisfied for at least $N_{mc}\hat{p}$ scenarios (where $\hat{p}$ is given in Table 3.6.2).

Table 3.6.2 – Corresponding observed $\hat{p}$ value when $\alpha = 0.05, N_{mc} = 10000$

ϵ	0.02	0.03	0.05	0.10	0.15	0.20	0.25
$\hat{p}(\%)$	98.22	97.27	95.35	90.48	85.58	80.65	75.71

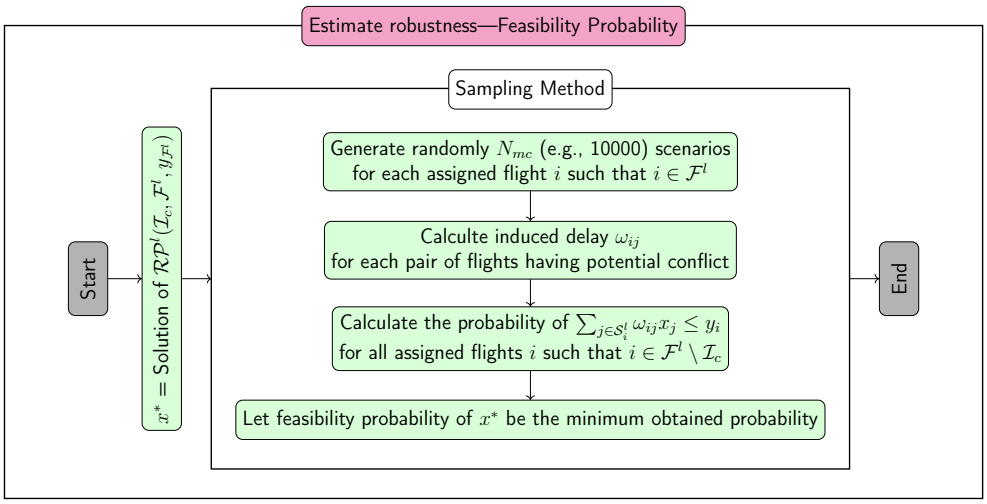


Figure 3.6.6 – Estimation of feasibility probability of obtained solution by Monte-Carlo simulation

The flow chart of estimating the feasibility probability of obtained solution by means of Monte-Carlo simulation is presented in Figure 3.6.6. The key brick of MC method is the estimation of the feasibility probability of the corresponding constraints (3.22) by sampling. We randomly generate a large bunch of possible

scenarios for the assigned flight $x_i^* = 1, i \in \mathcal{F}^l$. The induced ATFM en-route delay is then calculated in each generated scenario, and the number of scenario such that $\sum_{j \in \mathcal{S}_i^l} \omega_{ij} x_j \leq y_i, x_i^* = 1, i \in \mathcal{F}^l \setminus \mathcal{I}_c$ is countered. The feasibility probability of constraint (3.22) for corresponding x_i^* is hereby obtained.

Heuristic estimation method—Computing an estimation of sum of random values following truncated GMM distribution The Hoeffding's inequality gives us an upper bound of the probability of feasibility of obtained solution while it may be quite weak and lead to a costly solution. The Monte-Carlo simulation provides us a good robust solution than the former method, while the computation time may be extremely higher than the others. Therefore, it strives us to calculate an estimation of the sum of random variables ω_{ij}, based on a data driven approach taking advantage of their known distribution of each random variable ω_{ij} (i.e., following a truncated GMM distribution) while the corresponding computation remains tractable. The distribution for the sum of identical independent random variables is usually unknown or quite difficult to be characterized by a closed-form expression, even if the distribution of each random variable is known. However the sum of identical independent Gaussian Mixture Model distributed variables follows also a GMM, so as the Gaussian distribution, Poisson distribution and Gamma distribution do. Saying that, our intention is to approximate the distribution of uncertain data, i.e., the induced en-route ATFM delay ω_{ij} of flight i for a resolution of potential conflict with flight j at the same level, as a GMM variable which will give a GMM as well for their summation. We refer it as heuristic feasibility probability estimation method (Heuristic method). To do this we need to go through two steps (the algorithm's general scheme is described in Algorithm 2) :

- First, we approximate each ω_{ij} (which are truncated GMM and bounded in $[0, \overline{\omega}_{ij}]$), into a GMM one. We do this by applying an approximation operator implemented through a modified-Expectation-Maximization (modified-EM) algorithm (see Algorithm 3).

- For the second step, we need to approximate the sum of GMM, i.e., convolution of GMM. Remind here that the convolution of any two GMMs with K_1 and K_2 components will produce a GMM of $K_1 * K_2$ components, which leads

to an extremely large number of components for a sum of n GMM distributed variables. Then, to make the computation tractable, we propose to approximate the sum of two GMMs with K components (which gives a GMM with K^2 components due to convolution) to another GMM with only K components. This comes to an approximation of GMM with K^2 components into an another one with K components. We denote this operation with $\oplus$. This operation consists in merging components of the convolution according to the Hellinger distance (look at [Cutler and Cordero-Brana, 1996]). Applying this to all sum will give a GMM of K components (see Algorithm 4).

Algorithm 2: The general scheme of approximation of $\sum \omega_{ij}$ into a Gaussian Mixture Model of K components

Input: The distribution of n random variables ω_{ij} denoted ω^p . Each of them corresponds to a truncated GMM of K components.

Output: A Gaussian Mixture Model of K components

1 Compute $\tilde{\omega}^p$ for all $1 \leq p \leq n$ ($\tilde{\omega}^p$ is an approximation of ω^p following a GMM distribution of K components);

2 Compute $\omega^* = \tilde{\omega}^1 \oplus \tilde{\omega}^2 \oplus \cdots \oplus \tilde{\omega}^n$ ($\oplus$ is a merging operation using the Hellinger distance, the operation priority is illustrated in Figure 3.6.7);

3 **return** ω^*, a GMM of K components.

Approximate the distribution ω^p by the modified-EM algorithm. The modified-EM algorithm is detailed in Algorithm 3. To start with, the coefficients μ_k, σ_k, c_k for the GMM of K components of given distribution ω^p are initialized, and the involved distribution is sampled by a set of nodes X_i (i.e., $X_i \in [\mu - 6\sigma, \mu + 6\sigma]$, where μ, σ^2 are mean and variance of involved distribution, respectively). We then calculate and normalize the conditional probability $\mathsf{P}_k(X = X_i)$ of each sampling point under each component k of targeted GMM. Furthermore, the coefficients μ_k, σ_k and c_k of each component k of targeted GMM are updated iteratively according to the conditional probability and sampling points by the equations (3.31). The approximated GMM of K components is obtained by N_{em} iterations.

Approximate the convolution of two GMMs by the K-means algorithm. The convolution of any two GMMs of K_1 and K_2 components will produce a GMM of $K_1 * K_2$ components, which leads to an extremely large number

Algorithm 3: A modified-EM algorithm to approximate the distribution ω^p into a Gaussian Mixture Model of K components

Input: The distribution ω^p

Output: A Gaussian Mixture Model of K components

Initialize the coefficients μ_k, σ_k, c_k for the GMM of K components;

do

  Calculate the conditional probability of sampling points (X_i) under each component k:

$$p_{ik} = c_k \mathsf{P}_k(X = X_i); p'_{ik} = \frac{p_{ik}}{\sum_{i=1}^{N_{\text{samp}}} \sum_{k=1}^{K} p_{ik}} \tag{3.30}$$

  Update coefficient of GMM: μ_k, σ_k, and c_k:

$$\mu_k = \sum_{i=1}^{N_{\text{samp}}} p'_{ik} X_i; \sigma_k^2 = \sum_{i=1}^{N_{\text{samp}}} p'_{ik} X_i^2 - \mu_k^2; c_k = \sum_{i=1}^{N_{\text{samp}}} p'_{ik} \tag{3.31}$$

while *number of iterations* $< N_{em}$;

return an approximated distribution $\tilde{\omega}^p$ (a GMM of K components).

of components (some of them contribute few for the distribution) for a sum of n GMM distributed variables. Therefore, a merging operator is introduced to reduce the big number of components for a GMM due to convolution, hereby an approximation of GMM with K^2 components into K components such that the distribution of sum of finite GMM distributed variables follows a GMM of K components. The procedure of such approximation is given in Algorithm 4. Initially, we randomly choose K components (i.e., a Gaussian distribution $\mathcal{N}'$) of a GMM with K^2 components as cluster, then for each component of involved GMM (i.e., a Gaussian distribution $\mathcal{N}$), the Hellinger distance [Cutler and Cordero-Brana, 1996] of each pair of $(\mathcal{N}, \mathcal{N}')$ between each component of involved GMM and each cluster in targeted GMM is calculated. Find the nearest cluster $\mathcal{N}'$ for each $\mathcal{N}$. For each cluster $\mathcal{N}'$, merge all weighted Gaussian distributions $\mathcal{N}$ and update the cluster $\mathcal{N}'$ by equations (3.33). The approximated GMM of K components is then obtained by N_{km} iterations.

An example illustrated the computation for probability of sum of induced delay for an assigned flight is given in Figure 3.6.7. The computation of $\mathsf{P}(\omega_1 + \omega_2 + \omega_3 +$

Algorithm 4: K-means algorithm to approximate a Gaussian Mixture Model of K^2 components into the one of K components

Input: $\mathcal{G}$: A GMM of K^2 components
Output: $\mathcal{G}'$: A GMM of K components

Randomly choose K components from $\mathcal{G}$ as clusters for $\mathcal{G}'$;
do
  foreach *Gaussian distribution* $\mathcal{N}_i \in \mathcal{G}$ **do**
    Calculate the Hellinger distance between $\mathcal{N}_i$ and cluster $\mathcal{N}'_j$:

$$H^2(\mathcal{N}_i, \mathcal{N}'_j) = \frac{1}{2} \int \left(\sqrt{d\mathcal{N}_i} - \sqrt{d\mathcal{N}'_j} \right)^2 \tag{3.32}$$

    Find the nearest cluster $\mathcal{N}'_j$ for $\mathcal{N}_i$;
  end
  foreach *cluster* $\mathcal{N}'_j \in \mathcal{G}'$ **do**
    Let $\mathcal{S}$ be Gaussian distributions $\mathcal{N}_i$ whose nearest cluster is $\mathcal{N}'_j$;
    Update coefficients of the cluster $\mathcal{N}'_j$: μ_j, σ_j, and c_j:

$$\mu_j = \sum_{\mathcal{N}_i \in \mathcal{S}} c_i \mu_i; \sigma_j^2 = \sum_{\mathcal{N}_i \in \mathcal{S}} c_i \sigma_i^2 + \mu_i^2; c_j = \sum_{\mathcal{N}_i \in \mathcal{S}} c_i \tag{3.33}$$

  end
while *number of iterations* $< N_{km}$;
return the new GMM $\mathcal{G}'$.

$\omega_4)$ is transformed into calculation of $\mathsf{P}((\mathcal{A}(\omega_1) \oplus \mathcal{A}(\omega_2)) \oplus (\mathcal{A}(\omega_3) \oplus \mathcal{A}(\omega_4)))$, applying an approximation operator for each $\omega_i, i = 1, 2, 3, 4$ and a merging operator for each pair of composed summation.

Therefore the computation of $\mathsf{P}(\sum_{j \in \mathcal{S}_i^l} \omega_{ij} x_j \leq y_i)$ for corresponding flight i is then tractable and easily calculated by applying the approximation operator (Algorithm 3) and merging operator (Algorithm 4).

The flow chart of estimating the feasibility probability of obtained solution by means of heuristic estimation method is present in Figure 3.6.8. To estimate the feasibility probability of the involved constraints (3.22), we approximate the distribution of summation of independent GMM-approximated random variables by proposed K-Means algorithm. Especially, these GMM-approximated random variables are approximated by proposed modified-EM algorithm from the original

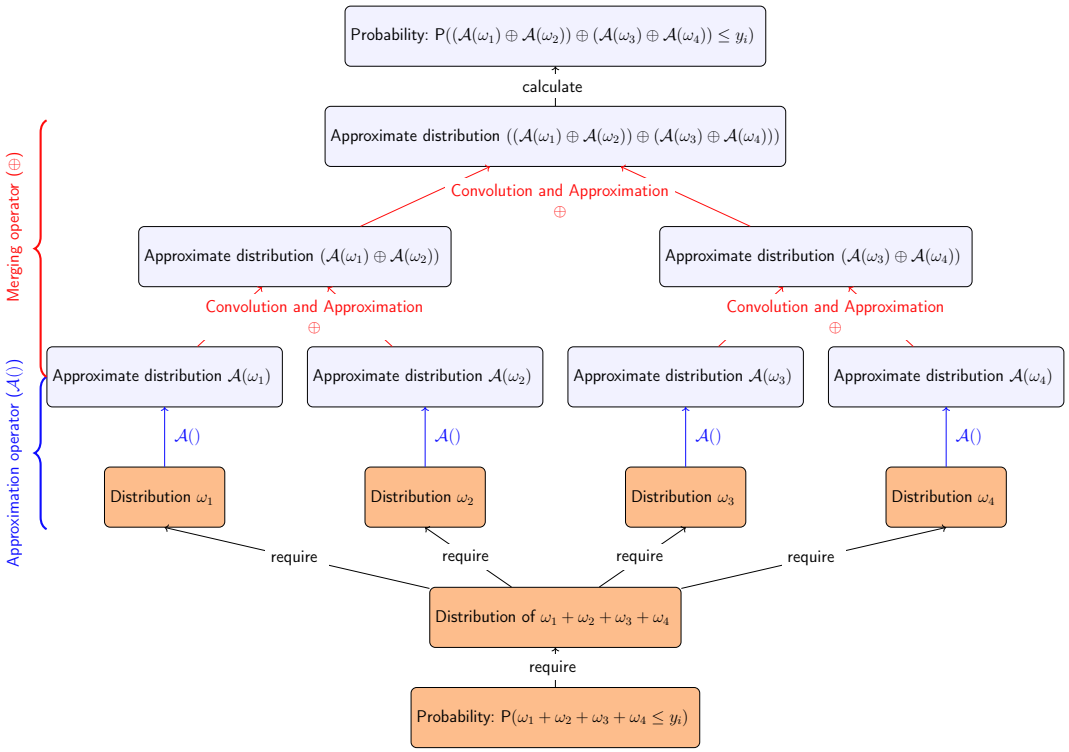


Figure 3.6.7 – An example illustrating the heuristic estimation of probability of the expression $\omega_1 + \omega_2 + \omega_3 + \omega_4$

random variables, a GMM approximation pool is applied to avoid a redundant approximation for the same variable which may be occurred in the constraint (3.22). The feasibility probability of constraint (3.22) for corresponding $x_i^* = 1$ is hereby tractable and obtained as the final evaluated term in constraint (3.22) approximately follows a GMM distribution.

3.7 Putting all pieces together

A general approximation framework to tackle the involved robust FLA problem is reported in Figure 3.7.9. Clearly the Algorithm 1 is the key brick of the solution approach.

The order examination of flight levels is done starting from the maximum loaded one (with the largest number of flights whose the most preferred level is the processing one) and so on. Then y_i is initialized as 0 for each concerned

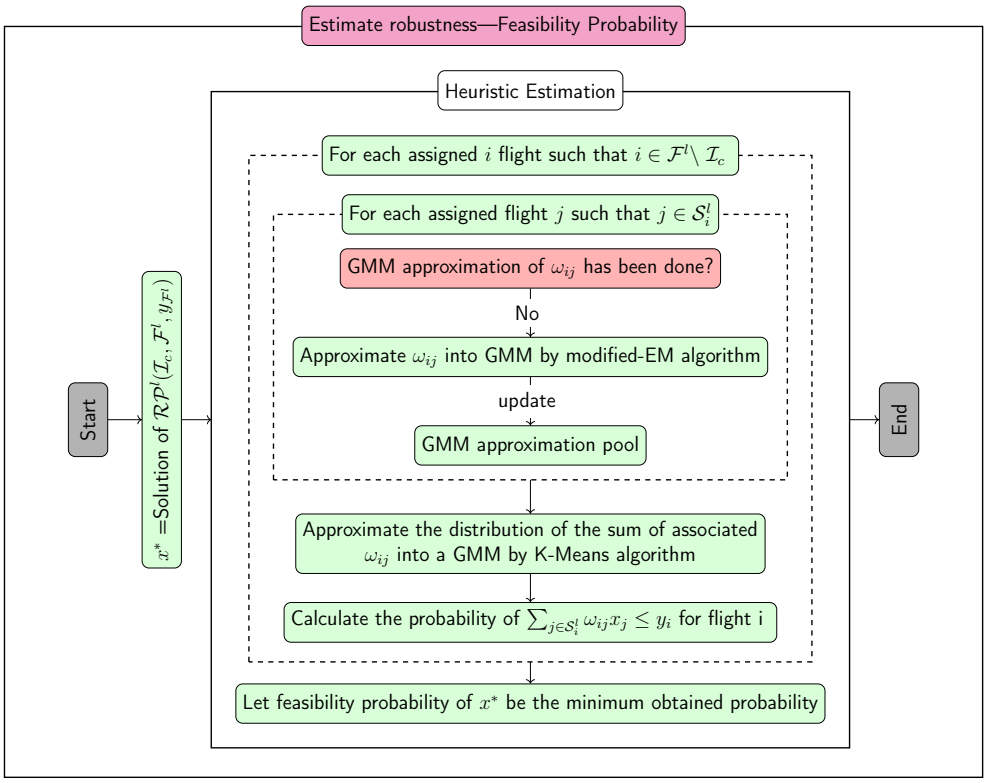


Figure 3.6.8 – Estimation of feasibility probability of obtained solution by heuristic estimation

flight i. At each iteration, for each flight level l, we accommodate a) the flights already assigned at current level, b) the fights not yet assigned but whose most preferred level is the processing one, and c) the non-assigned eligible ones at level l of which their most preferred level has been examined. Then the involved robust subproblem $\mathcal{RP}^l$ is solved by Algorithm 1. At the end of each iteration, if not all fights are either assigned or canceled, then we increase the acceptable induced delay y_i for concerned flights. Moreover, if y_i for some non assigned flight is already the maximum acceptable induced delay $\overline{W}_i$, then flight i is considered as "canceled" ; Otherwise, the process stops and we obtain a robust solution for the problem. Finally, for a post confirmation of the desired robustness, a posterior feasibility probability is computed by Monte-Carlo simulation.

More specifically, from point view of implementation, there are two possible cases of constraint $(\sum_{j \in \mathcal{S}_i^l} \overline{\omega}_{ij} x_j + M_i^l(x_i - 1) \leq y_i)$ for an unassigned flight i when we solve the model $\mathcal{RP}^l(\mathcal{I}_c, \mathcal{F}^l, y_{\mathcal{F}^l})$:

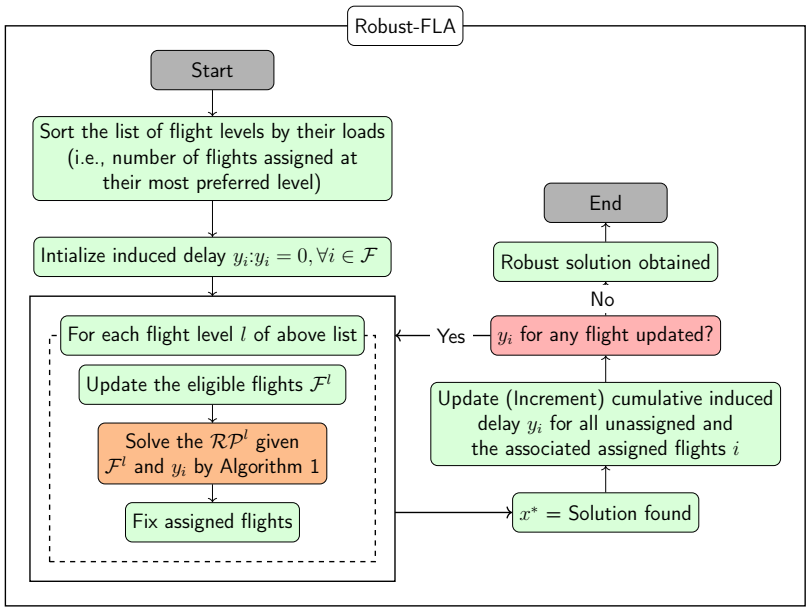


Figure 3.7.9 – General approximation framework tackling the Robust FLA problem

1. $\sum_{j \in \mathcal{S}_i^l} \overline{\omega}_{ij} x_j > y_i$;

2. $\sum_{j \in \mathcal{S}_i^l} \overline{\omega}_{ij} x_j \leq y_i$, but there may be some flights $k \in \mathcal{S}_i^l$ such that $x_k = 1$, $\sum_{j' \in \mathcal{S}_k^l \setminus \{i\}} \overline{\omega}_{kj'} x_{j'} \leq y_k$ and $\sum_{j' \in \mathcal{S}_k^l \setminus \{i\}} \overline{\omega}_{kj'} x_{j'} + \overline{\omega}_{ki} > y_k$.

For the first case, we increment y_i for the unassigned flight i. For the second case, we increment y_k for the associated assigned flight k rather than the unassigned flight i.

3.7.1 The modified heuristic feasibility probability estimation method

One may argue that the aforementioned robust FLA procedure involving the proposed heuristic feasibility probability estimation method can not guarantee that the posterior solution feasibility probability is larger than the desired one (1-ϵ). To assure that the obtained solution is robust with a probability at least 1-ϵ, we repeat the aforementioned robust FLA procedure with a higher probability target. More precisely, if at k^{th} iteration, the solution feasibility probability (denoted by P_k) has not the desired probability, we require a higher probability feasibility to

the Heuristic method for the concerned constraints in the next iteration of the robust FLA procedure. We refer to this method as the modified heuristic feasibility probability estimation method (HeuMod method).

3.8 Computational results

Numerical experiments are reported in this section. All algorithms were implemented in C++. Experiments were carried out on a server with Intel Xeon Gold 6138 2.0-GHz CPU and 125 GB of RAM under Linux.

3.8.1 Test instances

Table 3.8.3 – Characteristics of test instances

Instance	**Characteristics**			
	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{F}_c\rvert$	$\lvert\mathcal{S}_i\rvert_{\max}$	$\sum_{i\in\mathcal{F}}\lvert\mathcal{S}_i\rvert$
FR_1	1641	687	9	1612
FR_2	1887	810	9	1888
FR_3	2133	931	9	2090
FR_4	2461	1150	10	2782

Table 3.8.3 presents the characteristics of test instances corresponding to daily French air traffic with 134 airports and 715 waypoints. The instance FR_1 corresponds to French air traffic of August 12th, 1999. To accommodate current air traffic, we generate the instances FR_2, FR_3 and FR_4 by randomly generating 15%, 30%, and 50% supplementary traffics, respectively. Each instance is characterized by the number of flights ($|\mathcal{F}|$), the number of flights having at least one conflict with others at their most preferred level ($|\mathcal{F}_c|$), the maximum number of potential conflicts per flight at their most preferred level ($|\mathcal{S}_i|_{\max}$), total number of potential conflicts at their most preferred level ($\sum_{i\in\mathcal{F}}|\mathcal{S}_i|$).

In this study, the same aircraft of Airbus A320 is considered for all flights. The number of available flight seats is 180, and the Revenue per Available Seat

Kilometer (RASK) is 0.681 euros [AirFrance, 2019][3]. The en-route ATFM delay cost per minute per flight (α_i) is set to 70 euros if the en-route ATFM delay (y_i) is less than 30 minutes, and the cost is doubled for each supplementary 30 minutes (i.e., $\alpha_i = 140$ for $30 < y_i \leq 60$, and so on). The cancellation cost η_i for a flight i (when $y_i > \overline{W}_i$, the flight i is then considered as canceled) is set to 10 times of b_i^0 (the cancellation cost here is much larger than the assignment profit at their most preferred level as we want to minimize first the number of canceled flight, and the number of flight changing their level in order to maximize the total (robust) revenue), where b_i^0 indicates the estimated profit at their most preferred level. b_i^0 is calculated as: $b_i^0 = 180 * 0.681 * \text{MPM}_i$[4]. The level changing cost for a flight from the most preferred level to the other feasible alternative levels is considered as 10% of b_i^0. For example, a flight i flies from Bastia-Poretta to Lyon-Saint-Exupéry at level FL180 (the most preferred level, the two feasible alternative levels are FL160 and FL200), the corresponding flight miles is 588 Kilometers, it has a cumulative en-route ATFM delay of 140 (45 and 15, respectively) minutes with other flights assigned at the same level FL290 (FL270 and FL310, respectively). If we assign this flight i at level FL290, then we have the assignment profit: $b_i^0 = 180 * 0.681 * 588 = 72077.04$ euros, the en-route ATFM delay cost: $y_i * \alpha_i = 30*70+30*140+30*280+30*560+20*1120 = 53900$ euros, hereby the revenue of this assigned flight i at level FL180 is $72077.04 - 53900 = 18177.04$ euros. The revenue is reduced heavily due to the en-route ATFM delay. If this flight is canceled, then we considered a penalization as $72077.04 * 10 = 720770.4$ euros. Furthermore, if we assign this flight at level FL160 (FL200, respectively), we have a revenue $72077.04 * 0.9 - 30 * 70 - 15 * 140 = 67877.04$ ($72077.04 * 0.9 - 15 * 70 = 71027.04$, respectively) euros.

Table 3.8.4 – Posterior feasibility probability for all flights assigned at their most preferred flight level

Instance	FR_1	FR_2	FR_3	FR_4
Posterior feasibility probability	39.94%	38.98%	38.44%	38.95%

[3]In our calculations we have not involved different costs of running the flight as those connected to personnel, airports, airspace use, kerosene, etc.

[4]MPM: Maximum Permitted Miles

Table 3.8.4 indicates the posterior feasibility probability of all flights assigned at their most preferred flight level under the uncertainty of departure delays. Such assignment leads to a large number of flights (more than 60%) that become infeasible under real conditions involving uncertainty of flight departure time, hereby a costly maintainable solution may be induced to deal with data uncertainties.

3.8.2 Effectiveness of robust flight level assignment

Table 3.8.5 – Average en-route ATFM delay per flight and induced cost per flight when all ω_{ij} takes their worst-case value $\overline{\omega}_{ij}$ and all flights are assigned at their most preferred level

Instance	**FR_1**	**FR_2**	**FR_3**	**FR_4**	**Avg.**
$\tilde{y}_i$ [minutes]	2.42	2.45	2.58	2.95	2.63
Induced cost per flight [€]	15079.34	13934.76	23282.89	21221.09	18852.02

Table 3.8.5 specifies the average en-route ATFM delay per flight ($\tilde{y}_i$) and the corresponding induce cost per flight (the sum of the level changing cost, the en-route ATFM delay cost, and the cancellation cost per flight) when a Soyster model (all ω_{ij} takes their worst-case value $\overline{\omega}_{ij}$) is used to deal with the flight departure delay uncertainty for all flights assigned at their most preferred flight level, which means there is neither level changing their flight level nor flight cancellation. For these solutions of all different instances, there are in average 18852.02 euros of total induced cost per flight (equal to delay cost per flight as there is neither flight changing their flight level nor any flight canceled) due to en-route ATFM delay, where the average en-route ATFM delay in such solution is much far from the reference value (0.5 minute per flight).

Table 3.8.6 presents the average percentage of flights changing level (CHA%), the average percentage of canceled flights (CAN%), the average induced cumulative en-route delay per flight ($\tilde{y}_i$) and the average percentage of gain of total revenue (GanRev%=(total revenue A-total revenue B)/total revenue B*100%, where total revenue A is obtained by Robust FLA with a corresponding configuration and total revenue B is obtained by assigning all flights to their most preferred level with zero tolerance of infeasibility) when the robust flight level assignment is taken into account. The average delay cost per flight and the average total induced cost per

Table 3.8.6 – Computation results for robust FLA using the Soyster method

$\overline{W}_i$/**Instance**		**FR_1**	**FR_2**	**FR_3**	**FR_4**	**Avg.**
$\overline{W}_i$=5	GainRev%	20.85%	18.45%	46.33%	33.58%	29.80%
	CHA%	20.66%	21.63%	22.32%	23.81%	22.28%
	CAN%	0.24%	0.21%	0.28%	0.49%	0.32%
	$\tilde{y}_i$ [minutes]	0.25	0.23	0.19	0.23	0.23
	Delay cost per flight [€]	17.72	16.27	13.42	16.03	15.76
	Induced cost per flight [€]	4852.70	4730.81	5191.72	7102.88	5596.29
$\overline{W}_i$=10	GainRev%	24.29%	21.38%	51.43%	41.36%	34.62%
	CHA%	20.54%	21.52%	22.36%	23.97%	22.29%
	CAN%	0.06%	0.05%	0.05%	0.12%	0.07%
	$\tilde{y}_i$ [minutes]	0.29	0.27	0.23	0.29	0.27
	Delay cost per flight [€]	20.43	18.73	16.27	20.00	18.81
	Induced cost per flight [€]	3344.73	3412.44	3404.44	4211.46	3639.04
$\overline{W}_i$=15	GainRev%	23.22%	20.46%	50.39%	40.52%	33.65%
	CHA%	20.48%	21.47%	22.32%	23.93%	22.24%
	CAN%	0.12%	0.11%	0.09%	0.16%	0.12%
	$\tilde{y}_i$ [minutes]	0.28	0.26	0.22	0.28	0.26
	Delay cost per flight [€]	19.63	18.02	15.65	19.47	18.16
	Induced cost per flight [€]	3815.45	3823.97	3766.58	4525.33	4019.93
Avg.	GainRev%	22.79%	20.10%	49.38%	38.49%	32.69%
	CHA%	20.56%	21.54%	22.33%	23.91%	22.27%
	CAN%	0.14%	0.12%	0.14%	0.26%	0.17%
	$\tilde{y}_i$ [minutes]	0.28	0.25	0.22	0.26	0.25
	Delay cost per flight [€]	19.26	17.70	15.11	18.50	17.58
	Induced cost per flight [€]	4004.29	3989.07	4120.91	5279.89	4418.42

flight including en-route ATFM delay, level changing and cancellation are also reported. The tolerance of solution infeasibility ϵ are parameterized as 1%, 2%, 3%, 4%, 5%, 10%, 15%, 20% and 25%. For each subproblem involved at a processing flight level, the Soyster model is used to deal with the flight departure delay uncertainty where all random values take their worst-case value. For all test instances, we have that: a) the value $\tilde{y}_i$ are significantly reduced ($\tilde{y}_i$ is up to 0.29 minute per flight by applying Robust FLA within Soyster model) for different configuration of maximum acceptable en-route ATFM delay per flight ($\overline{W}_i$), compared to $\tilde{y}_i$ in the obtained solution in Table 3.8.5. Moreover, the associated average delay

cost per flight is drastically reduced and the average total induced cost per flight is reduced from 3 to 7 times; b) there may be until 23.97% (CHA%=number of flights changing level/total number of flights*100%) flights changing their flight level and may be until 0.49% flight canceled (CAN%=number of canceled flights/total number of flights*100%) for their induced en-route ATFM delay exceeding the maximum acceptable en-route ATFM delay (i.e., $y_i > \overline{W}_i$, $\overline{W}_i = 5$ minutes), 0.12% and 0.16% for $\overline{W}_i = 10$ and 15 minutes, respectively; d) there may be until 46.33%, 51.43% and 50.39% gain of total revenue for $\overline{W}_i = 5$, 10 and 15 minutes, respectively. Therefore, the robust flight level assignment can significantly reduce the en-route ATFM delay also its total induced cost per flight. The gain of total revenue by Robust FLA can vary between 20% and 50%.

3.8.3 Validation of the heuristic feasibility probability estimation method

Table 3.8.7 presents the comparison results in terms of the gain of total revenue (GainRev%) of Robust FLA, average percentage of flights changing level (CHA%), average percentage of canceled flights (CAN%), average en-route ATFM delay per flight ($\tilde{y}_i$), average delay cost per flight and average total induced cost, CPU time, minimum percentage of difference between posterior probability and desired feasibility probability (Min Diff.Proba.% = Minimum of $\frac{\text{Solution feasiblity probability}-(1-\epsilon)}{1-\epsilon} * 100\%$) and the maximum percentage for all three $\overline{W}_i$ (5, 10 and 15) and all infeasibility tolerance (1%, 2%, 3%, 4%, 5%, 10%, 15%, 20%, 25%) configurations, where Hoeffding method, MC method ($N_{mc} = 10000$), proposed Heuristic method ($N_{em} = 10$, $N_{km} = 10$), and the modified heuristic feasibility probability estimation method are applied. We find that the robust FLA with Soyster model performs not well in terms of gain of total revenue and gives the largest induced cost per flight due to the over-conservatism of uncertainty on ω_{ij}, which requires more flights canceled or their flight level changed to satisfy the constraints (3.22). The robust FLA using Hoeffding method to estimate feasibility probability is slightly better than the robust FLA with Soyster method, but spends roughly 3 times more computation time. The robust FLA with MC method performs significantly better than the two former, however enormous computation time is required to obtain such solution with certified feasibility probability. The robust FLA with

heuristic feasibility probability estimation method performs much better in terms of gain of total revenue, the en-route ATFM delay, its induced cost and computation time. Still, we notice that the solution feasibility probability has an error

Table 3.8.7 – Computational result for robust FLA by different feasibility estimation methods

Method	**Soyster**	**Hoeffding**	**MC**	**Heuristic**	**HeuMod**
GainRev%	32.69%	33.73%	38.22%	39.04%	38.62%
CHA%	22.27%	22.26%	11.08%	8.68%	9.42%
CAN%	0.17%	0.12%	0.06%	0.05%	0.06%
$\tilde{y}_i$ [minutes]	0.25	0.36	0.10	0.09	0.10
Delay cost per flight [€]	17.58	25.33	6.81	6.28	7.36
Induced cost per flight [€]	4418.42	4007.27	2067.06	1701.57	1871.62
CPU Time [seconds]	286.46	668.78	25481.05	472.50	1776.32
Min Diff.Proba.%	-	1.01%	0.00%	-1.83%	0.00%
Max Diff.Proba.%	-	33.00%	2.20%	2.39%	6.74%

of at most 1.83% for their lower bound value compared to the desired feasibility probability. For example, the posterior solution feasibility probability is at least 99%*(1-1.83%) = 97.19%, where the infeasibility tolerance (ϵ) is 1%. More specifically, this approximation error has its roots in the approximation operator and the merging operator. A trade-off between the robustness of solution and the total revenue (the operational cost could be level change, flight cancellation and cumulative en-route delay) is applied for the robust flight level assignment problem by the proposed modified heuristic estimation method in case when the posterior solution feasibility probability must be at least 1-ϵ for each concerned probabilistic constraint. This method can have a quasi-certified solution compared to the Monte-Carlo simulation method but requires much less computation time.

Comparison results of different configurations of $\overline{W}_i$

Table 3.8.8 presents the computational result for robust FLA with different configurations of $\overline{W}_i$ (i.e., $\overline{W}_i$=5, 10, and 15 minutes) and for different probability estimation method for probabilistic constraints (3.22). For all probability estimation methods, the solution obtained from Robust FLA have higher gain in terms of objective value and lower induced cost per flight (payed to flight level changing,

Table 3.8.8 – Computational result for Robust FLA by different feasibility estimation methods with different configurations of $\overline{W}_i$

$\overline{W}_i$	$\overline{W}_i$/**Method**	**Soyster**	**Hoeffding**	**MC**	**Heuristic**	**HeuMod**
$\overline{W}_i$=5	GainRev%	29.80%	31.44%	36.03%	38.06%	37.62%
	CHA%	22.28%	22.32%	11.02%	8.67%	9.28%
	CAN%	0.32%	0.24%	0.18%	0.10%	0.11%
	$\tilde{y}_i$ [minutes]	0.23	0.32	0.08	0.08	0.09
	Delay cost per flight [€]	15.76	22.61	5.72	5.52	6.39
	Induced cost per flight [€]	5596.29	4957.03	2962.87	2113.66	2293.22
	CPU Time [seconds]	169.09	464.87	23294.55	397.53	1265.76
	Min Diff.Proba.%	-	1.01%	0.03%	-1.72%	0.00%
	Max Diff.Proba.%	-	33.00%	2.20%	1.92%	4.91%
$\overline{W}_i$=10	GainRev%	34.62%	34.61%	39.28%	39.50%	39.03%
	CHA%	22.29%	22.23%	11.11%	8.69%	9.57%
	CAN%	0.07%	0.07%	0.01%	0.03%	0.04%
	$\tilde{y}_i$ [minutes]	0.27	0.38	0.10	0.09	0.11
	Delay cost per flight [€]	18.81	26.34	7.24	6.57	7.77
	Induced cost per flight [€]	3639.04	3641.34	1631.38	1505.93	1701.67
	CPU Time [seconds]	409.68	803.05	27198.54	487.01	2006.66
	Min Diff.Proba.%	-	1.01%	0.00%	-1.66%	0.00%
	Max Diff.Proba.%	-	32.72%	1.94%	2.32%	6.74%
$\overline{W}_i$=15	GainRev%	33.65%	35.15%	39.34%	39.55%	39.23%
	CHA%	22.24%	22.22%	11.09%	8.69%	9.40%
	CAN%	0.12%	0.05%	0.01%	0.03%	0.03%
	$\tilde{y}_i$ [minutes]	0.26	0.39	0.11	0.10	0.11
	Delay cost per flight [€]	18.16	27.06	7.47	6.76	7.91
	Induced cost per flight [€]	4019.93	3423.44	1606.94	1485.10	1619.97
	CPU Time [seconds]	343.87	738.43	25950.07	532.96	2056.54
	Min Diff.Proba.%	-	1.01%	0.00%	-1.83%	0.01%
	Max Diff.Proba.%	-	32.97%	1.76%	2.39%	5.17%

flight en-route ATFM delay, and the flight cancellation) when the upper bound of cumulative en-route ATFM delay (y_i) per flight ($\overline{W}_i$) is larger, since increasing $\overline{W}_i$ hwill result in possible longer ATFM delays. In the other hand, for the configuration of larger value of $\overline{W}_i$ (e.g., 10 minutes), they requires more computation time because the associated y_i is incremented step by step at end of each iteration when the corresponding constraint (3.22) is violated. For each configuration of $\overline{W}_i$,

the robust FLA with the proposed heuristic probability estimation method always performs the best (in terms of gain of total revenue, number of flight changing flight level, number of canceled flights, the en-route ATFM delay, its induced cost and computation time) within an error of solution feasibility probability less than 2%.

Comparison results of different configurations of ϵ when $\overline{W}_i = 5$ minutes

Table 3.8.9-3.8.10 show the computational result for Robust FLA of different configurations of solution infeasibility tolerance (ϵ) (i.e., ϵ=1%, 2%, 3%, 4%, 5%, 10%, 15%, 20% and 25%) when $\overline{W}_i = 5$ minutes for different probability estimation method for probabilistic constraints (3.22). The Robust FLA has higher gain in terms of total revenue and lower induced cost per flight when the desired solution feasibility probability is lower for all probability estimation methods. Focusing on $\epsilon = 25\%$, when MC method, Heuristic method or HeuMod method is applied, we can observe there is no flights cancelled, the robust FLA achieve more than 40% gain in terms of total assignment revenue by a few flight changing their level, compare to the solution where all flights are assigned to their most preferred level with zero tolerance of infeasibility (i.e., ϵ=0%). When we allow some flight to change their level, and apply a large tolerance of infeasibility such as 25%, the total assignment revenue can be significantly improved. When focusing on the proposed heuristic feasibility probability estimation method, the robust FLA can assign the flight into an appropriate level to make all flight conflict-free (i.e., $\tilde{y}_i$=0) by changing at most 4.15% flights without cancellation for $\epsilon \geq 10\%$. However, when looking at the Robust FLA with the MC method we notice that it requires more flights changing and cancellations to make all flights conflict-free for the solution infeasibility tolerance not less than 15%, while the computation time is much higher compared to the proposed Heuristic method. Concerning the error of posterior solution feasibility probability (i.e., less than 2%), it is found that the largest error comes from the configuration of ϵ at 4%. This error of posterior solution feasibility probability is actually less than 1% for all other configurations of ϵ excepting for ϵ =4%.

Table 3.8.9 – Computational result for Robust FLA by different estimation methods with different configurations of ϵ when $\overline{W}_i = 5$ minutes

	Method/ϵ	**1.00%**	**2.00%**	**3.00%**	**4.00%**	**5.00%**
Hoeffding	GainRev%	31.32%	31.32%	31.31%	31.32%	31.33%
	CHA%	22.26%	22.26%	22.28%	22.25%	22.23%
	CAN%	0.25%	0.25%	0.25%	0.25%	0.25%
	$\tilde{y}_i$ [minutes]	0.34	0.34	0.33	0.33	0.33
	Delay cost per flight [€]	23.74	23.65	23.42	22.94	22.86
	Induced cost per flight [€]	4992.03	4991.94	4993.15	4989.47	4985.86
	CPU Time [seconds]	702.69	685.54	293.78	671.20	285.47
	Min Diff.Proba.%	1.01%	2.04%	3.09%	4.17%	5.25%
	Max Diff.Proba.%	1.01%	2.04%	3.09%	4.17%	5.26%
MC	GainRev%	31.62%	32.99%	33.64%	34.15%	34.73%
	CHA%	21.86%	19.05%	16.75%	14.93%	12.88%
	CAN%	0.23%	0.21%	0.21%	0.21%	0.21%
	$\tilde{y}_i$ [minutes]	0.22	0.17	0.14	0.10	0.08
	Delay cost per flight [€]	15.21	12.00	9.58	7.29	5.42
	Induced cost per flight [€]	4867.50	4276.24	3991.47	3762.65	3506.85
	CPU Time [seconds]	59440.88	40787.45	32542.84	30854.94	19281.12
	Min Diff.Proba.%	0.03%	0.15%	0.03%	0.21%	0.18%
	Max Diff.Proba.%	0.20%	0.21%	0.14%	0.33%	0.43%
Heuristic	GainRev%	30.77%	34.14%	34.32%	36.40%	37.70%
	CHA%	20.43%	15.85%	13.23%	10.98%	8.99%
	CAN%	0.30%	0.18%	0.21%	0.14%	0.10%
	$\tilde{y}_i$ [minutes]	0.27	0.20	0.13	0.07	0.02
	Delay cost per flight [€]	19.17	13.95	9.26	5.16	1.39
	Induced cost per flight [€]	5188.70	3801.41	3674.57	2775.55	2219.17
	CPU Time [seconds]	818.12	806.13	366.96	844.04	265.08
	Min Diff.Proba.%	-0.28%	-1.72%	-0.39%	-0.26%	-0.34%
	Max Diff.Proba.%	-0.09%	-0.15%	-0.24%	-0.07%	-0.09%
HeuMod	GainRev%	30.85%	32.98%	34.43%	34.45%	37.52%
	CHA%	21.03%	17.90%	14.07%	11.44%	9.47%
	CAN%	0.28%	0.20%	0.18%	0.21%	0.10%
	$\tilde{y}_i$ [minutes]	0.30	0.24	0.16	0.08	0.02
	Delay cost per flight [€]	20.81	16.81	11.34	5.87	1.69
	Induced cost per flight [€]	5173.48	4201.16	3642.31	3574.66	2295.92
	CPU Time [seconds]	1813.64	1713.43	747.28	4877.05	1336.60
	Min Diff.Proba.%	0.01%	0.08%	0.06%	0.04%	0.00%
	Max Diff.Proba.%	0.22%	1.70%	0.19%	0.23%	0.25%

Table 3.8.10 – Continuous table of Table 3.8.9

	Method/ϵ	**5.00%**	**10.00%**	**15.00%**	**20.00%**	**25.00%**
Hoeffding	GainRev%	31.33%	31.32%	31.53%	31.92%	31.62%
	CHA%	22.23%	22.26%	22.28%	22.45%	22.62%
	CAN%	0.25%	0.25%	0.23%	0.21%	0.22%
	$\tilde{y}_i$ [minutes]	0.33	0.32	0.32	0.31	0.28
	Delay cost per flight [€]	22.86	22.44	22.27	21.49	19.91
	Induced cost per flight [€]	4985.86	4990.66	4895.68	4738.19	4874.92
	CPU Time [seconds]	285.47	525.61	279.67	478.21	261.64
	Min Diff.Proba.%	5.25%	11.10%	17.62%	24.69%	32.91%
	Max Diff.Proba.%	5.26%	11.11%	17.65%	24.79%	33.00%
MC	GainRev%	34.73%	36.69%	38.87%	39.01%	42.59%
	CHA%	12.88%	7.52%	3.08%	1.84%	1.31%
	CAN%	0.21%	0.20%	0.16%	0.17%	0.00%
	$\tilde{y}_i$ [minutes]	0.08	0.02	0.00	0.00	0.00
	Delay cost per flight [€]	5.42	1.06	0.16	0.09	0.21
	Induced cost per flight [€]	3506.85	2637.60	1678.99	1620.62	176.50
	CPU Time [seconds]	19281.12	13750.77	5265.16	5436.28	2291.50
	Min Diff.Proba.%	0.18%	0.08%	1.01%	0.55%	1.53%
	Max Diff.Proba.%	0.43%	0.81%	1.51%	0.90%	2.20%
Heuristic	GainRev%	37.70%	41.63%	42.33%	42.57%	42.70%
	CHA%	8.99%	4.15%	2.10%	1.36%	0.91%
	CAN%	0.10%	0.00%	0.00%	0.00%	0.00%
	$\tilde{y}_i$ [minutes]	0.02	0.00	0.00	0.00	0.00
	Delay cost per flight [€]	1.39	0.14	0.00	0.00	0.00
	Induced cost per flight [€]	2219.17	596.21	290.07	187.07	127.77
	CPU Time [seconds]	265.08	230.04	68.21	115.86	63.38
	Min Diff.Proba.%	-0.34%	-0.18%	-0.39%	-0.64%	1.28%
	Max Diff.Proba.%	-0.09%	0.08%	0.51%	0.49%	1.92%
HeuMod	GainRev%	37.52%	40.74%	42.32%	42.55%	42.70%
	CHA%	9.47%	5.23%	2.12%	1.39%	0.91%
	CAN%	0.10%	0.02%	0.00%	0.00%	0.00%
	$\tilde{y}_i$ [minutes]	0.02	0.01	0.00	0.00	0.00
	Delay cost per flight [€]	1.69	0.35	0.00	0.00	0.00
	Induced cost per flight [€]	2295.92	984.17	294.29	193.89	127.77
	CPU Time [seconds]	1336.60	439.21	110.58	266.65	87.37
	Min Diff.Proba.%	0.00%	0.01%	0.08%	0.05%	1.28%
	Max Diff.Proba.%	0.25%	4.91%	0.51%	0.69%	1.92%

Comparison results of lower bound of solution feasibility probability for different methods when $\overline{W}_i = 5$ minutes

Figure 3.8.10 presents the lower bound of posterior solution feasibility probability for different methods when $\overline{W}_i = 5$ minutes. The curve of posterior solution feasibility probability for the proposed heuristic feasibility probability estimation method is below the curve of desired solution feasibility probability (orange curve) excepting for 25%. Moreover, when $\epsilon = 4\%$, the error between the posterior probability for Heuristic method and the target one reaches the highest one (roughly 2%) as already discussed in the previous section. With respect to the Hoeffding's inequality feasibility probability estimation method, we can see that the curve of posterior feasibility probability (red curve) is always close to 99%-100% regardless of the desired feasibility probability. Concerning the Monte-Carlo simulation feasibility probability estimation method, the curve of posterior feasibility probability (blue curve) is always above the target curve which guarantees the robustness of solution. One issue may be argued, the proposed heuristic probability estimation can not guarantee the lower bound of the posterior solution feasibility probability as there is an error less than 2%. The obtained solution is not as robust as what we expect. Therefore, a modified variant which penalizes the estimated probability once the solution is not sufficiently robust and we repeat the Robust FLA until the solution becomes totally robust. The modified variant of the proposed Heuristic method now can guarantee the lower bound of the posterior solution feasibility probability (i.e., the green curve is always above the target curve) and has a better solution (in terms of total assignment revenue, induced cost per flight and computation time) than the MC method.

3.9 Conclusion and discussion

As conclusion, we may say that the robust FLA problem expressed as CCP, is a difficult one. All this justifies the use of heuristics as developed in the paper. The proposed approach, inspired from the work of Klopfenstein, yields a general CCP approach for a specific class of ILP problems. Suppose it is given an ILP (or LP) with uncertain coefficients. According to Soyster model, one can easily write the corresponding robust optimization problem by putting each constraint in its

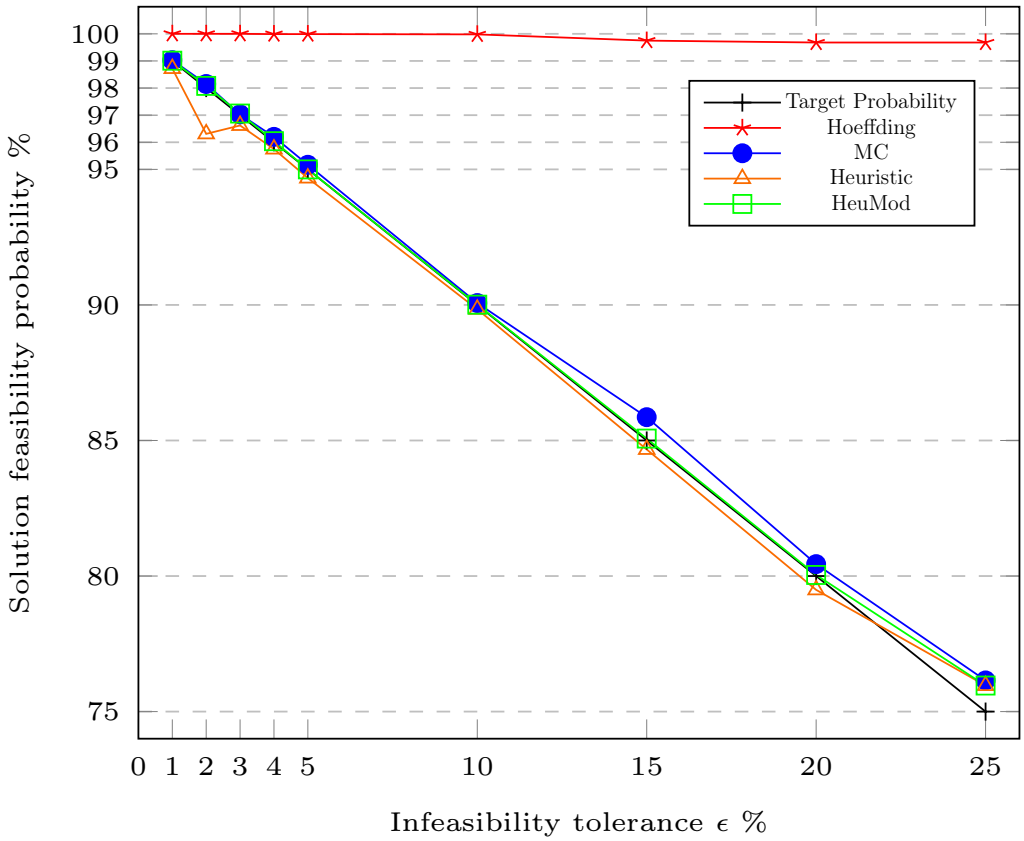


Figure 3.8.10 – Lower bound of posterior solution feasibility probability for the robust FLA by different estimation methods with different configurations of ϵ when $\overline{W}_i = 5$ minutes

"worst" scenario version. On way to remedy the very conservative nature of the Soyster model, is not to put all constraints in the beginning but to start with a few of them and add consecutively the others as they are needed, while keeping an eye at the feasibility. Obviously this leads to a constraint generation approach working as below:

1. Start with the master problem P. Solve P and denote with x^* the obtained solution.

2. Check the feasibility probability of x^* with respect to the desired feasibility probability p (p is the smallest one of the posterior probability for all associated probabilistic constraints).

3. If $p < 1 - \epsilon$, and c is the constraint achieving it, then add c to P. Repeat above procedure until $p \geq 1 - \epsilon$.

Clearly the above procedure is extremely simple. Still, it is naturally deduced as well from Klopfenstein as it is for our application. It can be especially usable for robust binary linear programming with if-else constraints. Such constraints raise generally in $\mathcal{NP}$ hard problems and are formally represented by a constraint involving binary variables and big M. All this yields difficult optimization problems which often become intractable when robustness has to be considered. In our focus are conditions of type: «if cond then thenexp else elseexp endif » where «cond» is a linear formulation involving uncertain coefficients while « thenexp » and « elseexp » are related to a decision binary variable, for instance thenexp $\sim$ x $\leq$ 1 and elseexp $\sim$ x $=$ 0. Then we will need to add M (x - 1) in the left hand of the condition. For instance if we have: $A_{11}x_1 + A_{12}x_2 \leq B \Longrightarrow x \leq 1$ else x $=$ 0. This may be expressed as: $A_{11}x_1 + A_{12}x_2 + M(x - 1) \leq B$, which are similar to (3.20) and where the same arguments may be used to strive a simplified version of Klopfenstein's model. We intend to investigate this issue in detail in the future.

Concerning the feasibility probability estimation method, the proposed heuristic approach pursues approximating the distribution of a summation for the independent asymmetrically distributed random variables into a GMM distribution by the proposed approximation operator and merging operator. The modified variant with penalization (for the estimated probability) of the proposed heuristic approach has a guarantee of the posterior solution feasibility probability, which means the obtained solution is robust as expected. This method is more efficient to get a good robust solution compared to the Hoeffding method while it remains much faster than the MC method.

4. Resource Allocation in 5G Superfluid Wireless Networks

4.1 Introduction

With the exponential rise of mobile users, overall mobile data traffic is expected to grow to 49 exabytes[1] per month by 2021, a seven-fold increase over 2016 [Cisco Mobile, 2017]. This is the natural result of the evolution and incredible expansion that mobile networks have known since the beginning of the new millennium: from the 1980s, new generations of mobile network technologies have been proposed to continuously offer higher speed, greater capacity and new innovative sets of services as presented in Figure 4.1.1. The 1st generation of wireless networks (1G) has granted the first (expensive) mobile services, offering voice-only calls by means of bulky cellular phones which, however, were considered a status symbol of successful business people. Then, the 2G appeared in 1991 and allowed a more democratic and cheaper access to cellular phones and services, better supporting texting. Since 1998, the 3G, besides voice calls (wireless connection can be achieved at high speed from 20 Kbps to 42.2Mbps), introduced mobile access to the internet and, during its technological life cycle, smartphones appeared. The 4G technologies starting in the 2008, offered enhanced performance especially for smartphones with a peak data rate of 100 Mbps (4G LTE), 150 Mbps (4G LTE Cat.4), and 1000 Mbps (4G LTE Advanced). During the 4G era, an impressive surge in the number of mobile users, asking for higher performance, has taken place. This has led to the concrete need for a 5G that could offer unprecedented performance and features.

"Ubiquitous connectivity", "Zero latency"and "High-speed Gigabyte connection",

[1] 1 exabyte = 10^6 Gigabytes

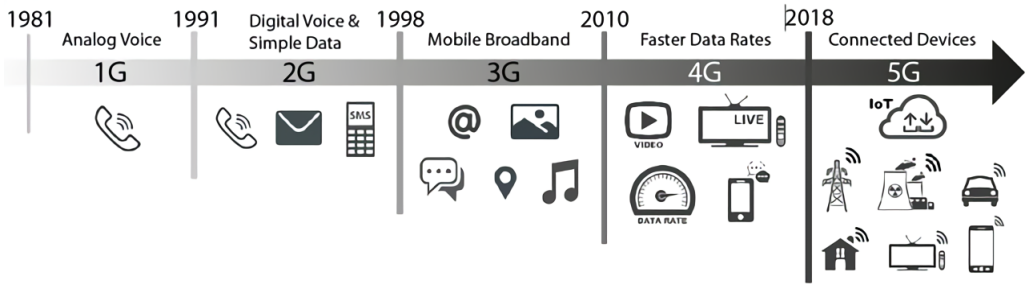


Figure 4.1.1 – Evolution of wireless network from 1G to 5G, source from [Gohar and Nencioni, 2021]

introduced in [Panwar et al., 2016], are considered to be essential features that a 5G technology should support. Ubiquitous connectivity can be essentially described as the capacity of granting connectivity to every device, everywhere, at every time; we remark that this question may result very challenging in 5G networks, which should effectively support the Internet-of-Things thus requiring to grant simultaneous wireless connectivity to an extremely high number of devices. Zero latency refers to the capacity of a 5G network of supporting null or negligible time lag during data transfer; this is considered particularly critical in applications such as telemedicine (e.g., telesurgery), which requires to remotely intervene on patients in real-time without suffering from negative effects of latency that could threat the life of a patient. High-speed Gigabyte refers to the capacity of a 5G network to establish wireless connection characterized by extremely high speed that is currently far beyond the possibility of 4G networks. To achieve this 5G network, lots of novel architectures (e.g., Superfluidity [Bianchi et al., 2016], SELFNET [Jiang et al., 2017], Flexible Functional Split [Harutyunyan and Riggio, 2017]) and new technologies (e.g., mMIMO (massive Multiple-Input Multiple-Output) [Larsson et al., 2014], mmWave (millimeter Wave) and beam-forming [Roh et al., 2014]) are proposed by active researchers.

Among the different investigated 5G architectures, it is worthwhile to mention 5G Superfluid [Bianchi et al., 2016] (SF) architecture. The goal of 5G Superfluid architecture is to design a new 5G network architecture, which ensures the required levels of flexibility, agility, portability and high performance. In a nutshell, 5G Superfluid architecture aims to achieve a superfluid state of the network, which is the ability to instantiate services on-the-fly, run them anywhere in the network

(core, aggregation, edge) and shift them transparently to different locations. The key brick of the 5G Superfluid architecture is the definition of the concept of Reusable Functional Block (RFB), which is a virtualized entity, used to decompose network functions and services, and it is deployed on top of a physical node. The deployment of the 5G network through the RFBs have notable features, including: i) the possibility to build chain of RFBs, in order to implement more complex functionalities and to provide different services to users; ii) the independence of the RFBs from a specific platform, i.e., RFBs can be realized via software functions, and they can be run on several hardware architectures; and, iii) the introduction of high levels of flexibility and performance, thanks to the fact that the RFBs can be deployed where and when they are really needed. Furthermore, the RFB concept is a generalization of the Virtual Network Function (VNF) concept proposed by [ETSI, 2014]. In particular, RFBs can be arbitrarily decomposed in other RFBs, while VNFs in the ETSI model cannot be composed or decomposed in other VNFs in a flexible way (see Figure 4.1.2 for their different class diagram). Moreover, the RFBs can be mapped to different software and hardware execution environments (see [Bianchi et al., 2016]), while the ETSI model focuses on mapping VNFs to Virtual Machines (or Containers) in traditional cloud infrastructures.

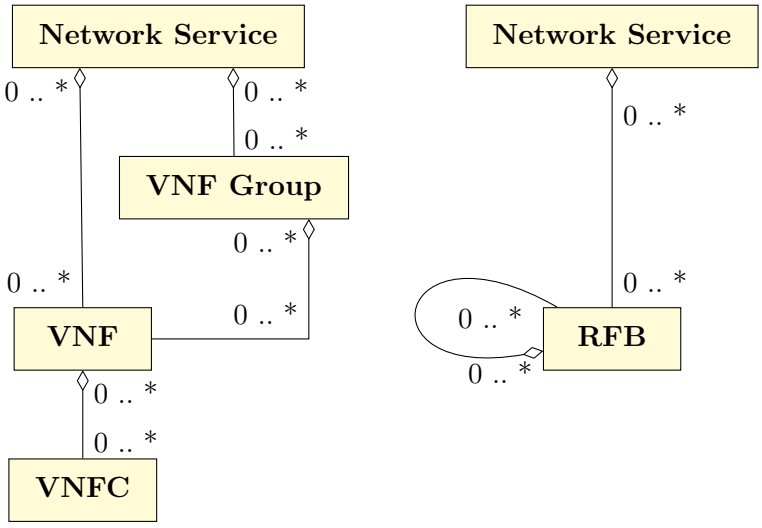


Figure 4.1.2 – Class diagram for the ETSI VNF (left side) and the 5G Superfluid RFB (right side), source from [Bianchi et al., 2016]

Concerning 5G SF architecture, one main issue is how to optimally minimize the

total installation costs of such a 5G SF Network network composed of RFBs and physical 5G nodes within guaranteeing a minimal required user coverage and minimum downlink traffic demand. Therefore, such a constructive 5G network could able to sustain the increase in the number of connected users, especially in very crowded environments, such as stadiums, airports, train stations, and shopping malls without a scarification of minimum downlink traffic per user. The RFB-based Resource Allocation problem under 5G Superfluid wireless Networks (5G-RFB-RA) was first proposed in [Chiaraviglio et al., 2018]. Since 5G-RFB-RA is $\mathcal{NP}$-hard, heuristic algorithms were proposed to solve it efficiently in [Chiaraviglio et al., 2019]. The problem was also tackled by a Particle Swarm Optimization algorithm in [Shojafar et al., 2017]. However, one drawback was represented by the fact that some users could obtain limited downlink bandwidth. In order to improve the capacity of solving instances of larger size associated with 5G Superfluid network design problems, we analyzed the polyhedral structure of the original model and propose new more efficient solution methods. Specifically, our original contributions can be summarized as follows:

- we propose an alternative formulation for the problem of minimizing the installation costs of a 5G SF network, taking as reference the model proposed in [Chiaraviglio et al., 2018];
- we strengthen the above proposed model, identifying new valid inequalities;
- we propose a hybrid Benders decomposition approach to tackle the simplified model.

Results of computational tests show that our new improved modeling and solution approach offers a higher performance both in computational time and used memory.

The remainder of this Chapter is organized as follows: in Section 4.2, we report a short discussion on 5G Superfluid architecture. The reference model and its improved version are proposed in Section 4.3. We then introduce a Benders-like decomposition approach to solve the problem more efficiently in Section 4.4. Finally, in Section 4.5, we report and discuss computational results.

4.2 Reusable Functional Blocks in Superfluid architecture

For an exhaustive introduction to Superfluid networks and Reusable Functional Blocks, we refer the reader to [Bianchi et al., 2016]. Here we recall some main facts about them. A Reusable Functional Block in 5G SF architecture is a logical entity that performs a set of functionalities and has a set of logical input/output ports. In general, a RFB can hold state information and can be combined with other RFBs to form other more complex and performing RFBs (see Figure 4.1.2). A RFB Description and Composition Language (RDCL) is introduced for characterizing and describing each RFB in a formal manner at the platform-agnostic node-level and network-level. Moreover, a RFB Execution Environments (REE) is specified to support the execution and deployment of the RDCL scripts and the relevant coordination of the signal/radio/packet/flow/network processing primitives as shown in Figure 4.2.3. The figure visualizes that the Reusable Functional Block (RFB)s are orchestrated recursively at each level (i.e., network level and node level). At each level, two main actors are identified: REE User and REE Manager. The REE User requests the deployment/execution of a service/service component described using a RDCL script to the REE manager, and the REE Manager is in charge of deploying/executing the RDCL script using the resources within its REE. Two main interfaces (denoted by UM API and MR API) are characterized to support the interaction between REE User and REE Manager, so that the REE User can deploy a service or a component into an REE, and between REE Manager and REE Resource Entity, so that the REE Manager can interact with the resources in its REE.

An RFB results analogous to a traditional VNF or VNFC, implemented as a fully-fledged VM running on a hypervisor or in a container. An RFB can correspond to a small footprint Unikernel VM running in a specialized hypervisor. RFBs can also be modules or components of special purpose execution environments, like extended finite state machines based on OpenFlow for packet processing, software routers, or radio signal processing chains. Hence the RFB concept can be applied to different heterogeneous environments, according to their specification as reported in [Bianchi et al., 2016]. In this study, we consider the RFBs that perform specific tasks in the network architecture, such as processing the video to users, or per-

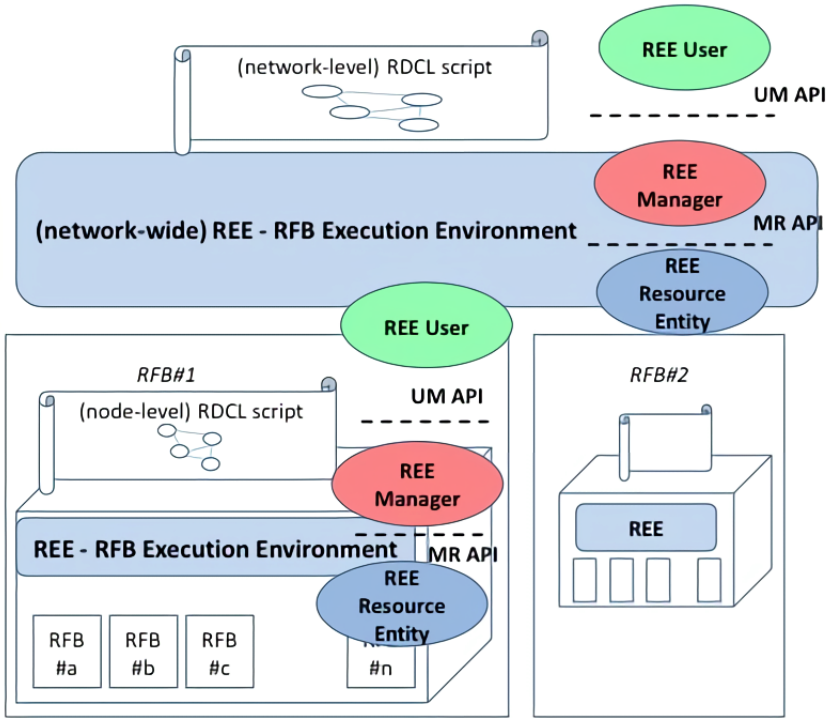


Figure 4.2.3 – 5G SF Architecture, source from [Bianchi et al., 2016]

forming networking and physical layer tasks [Chiaraviglio et al., 2017]. Focusing on the tasks realized by these RFBs, the following RFB are identified:

- Resource Radio Head RFB (RRH-RFB): it is in charge of providing the physical signal to the users. Specifically, it handles a set of Radio Frequency (RF) channels established with users and the corresponding baseband channels with the BBU-RFBs;

- Mobile/Multi-access Edge Computing RFB (MEC-RFB): it is in charge of managing an amount of traffic, such as the provisioning of a HD video service to users.

- Base Band Unit RFB (BBU-RFB): it acts as a middle interface between the RRH-RFBs and the MEC-RFBs. Specifically, the BBU-RFB exchanges an amount of IP traffic with the MEC-RFBs, and a baseband signal with the RRH-RFBs.

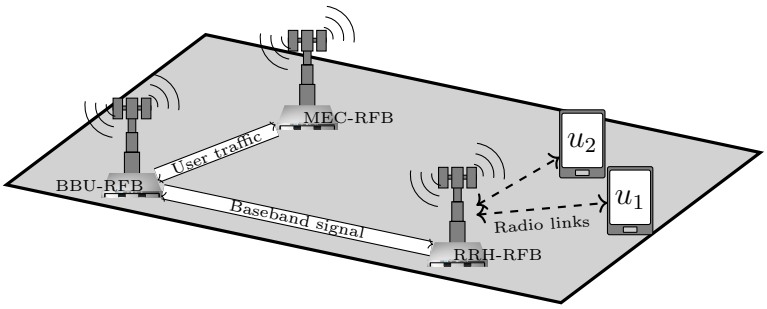


Figure 4.2.4 – A complete RFB chain and the exchanged information among them

From a logical point of view, each 5G node is able to host one RRH-RFB, one MEC-RFB and one MEC-RFB. Moreover, it can pool also BBU-RFBs and MEC-RFBs from other nodes. On the other hand, the RFBs are organized in chains where each RRH-RFB is connected to a BBU-RFB, which is in turn linked to a MEC-RFB as shown in Figure 4.2.4. We then assume that the 5G node can provide a service to user if and only if there exists a complete RFB chain composed of one RRH-RFB, one BBU-RFB and one MEC-RFB, linked to each other in this order. The RFB chain is hence not constrained to be located on the same 5G node, but it can be realized across several nodes (e.g., each 5G node holds one RFB module from the complete RFB chain as shown in Figure 4.2.4). RFBs are characterized by their resource requirements (i.e., storage, processing). The requirements in terms of consumed resources by the RFBs are then used in this work to properly dimension the 5G nodes. Finally, we consider on additional classification of each RFB into Micro and Macro type, depending on the area size and user number that they may serve (this corresponds with a classification of 5G base stations into micro and macro).

4.3 Mathematical Formulation

The optimization model that we present is based on the following sets and indices:

U: Set of users.

N: Set of candidate 5G nodes.

K: Set of RFB modules: RRH-RFB, BBU-RFB, and MEC-RFB.

Q: Set of type for RFB chains/RFB modules: Micro, Macro.

A_q^k: Available number of RFB module k in type q.

$U_q^{\max}$: Maximum number of users served by RRH-RFB in type q.

CAP_{unq}: Radio link capacity provided to user u at node n placed RRH-RFB in type q.

CAP_q^k: Maximum capacity managed by RFB module k in type q.

CONF_q: All the pairs of nodes that conflict for a RRH-RFB in type q.

t_{MIN}: Minimum data traffic required by user.

δ: Minimum fraction of users that has to be covered by the 5G service.

c_q^k: Construction cost of each RFB module k in type q.

4.3.1 Reference Mathematical Model

In this section, we show the model presented in [Chiaraviglio et al., 2018] that we have used as basis for our original developments.

Variables:

t_u: A continuous variable indicating the amount of downlink traffic served to user $u \in U$.

x_{un}: A binary variable taking value 1, if the user $u \in U$ is served by a RRH-RFB installed at node $n \in N$; 0, otherwise.

y_{nq}^{RRH}: A binary variable taking value 1, if the RRH-RFB of type $q \in Q$ is installed at node $n \in N$; 0, otherwise.

$v_{n_1n_2q}^{\text{BBU}}$: A binary variable taking value 1, if an BBU-RFB of type $q \in Q$ installed at node $n_1 \in N$ serves the RFB chain originating from the RRH-RFB installed at node $n_2 \in N$; 0, otherwise.

$v_{n_1n_2q}^{\text{MEC}}$: A binary variable taking value 1, if a MEC-RFB of type $q \in Q$ installed at node $n_1 \in N$ serves the RFB chain originating from the RRH-RFB installed at node $n_2 \in N$; 0, otherwise.

ϕ_{un}: A continuous variable indicating the amount of downlink traffic served to user u at node n, defined as $\phi_{un} \triangleq t_u x_{un}$.

θ_{unq}: An artificial binary variable indicating whether a user u at node n is served by the RRH-RFB of type q, defined as $\theta_{unq} \triangleq x_{un} y_{nq}^{\text{RRH}}$.

$\varphi_{un_1n_2q}$: An artificial continuous variable indicating the amount of downlink traffic served to user u charged by a RFB chain which originates from the RRH-RFB installed at node n_2 and using as sink to the MEC-RFB installed at node n_1, defined as $\varphi_{un_1n_2q} \triangleq \phi_{un_2} v_{n_1n_2q}^{\text{MEC}} = t_u x_{un_2} v_{n_1n_2q}^{\text{MEC}}$.

Using the above-mentioned notation, assuming that each RRH-RFB is compatible with the same type BBU-RFB and the same type of MEC-RFB, and a user is compatible with all types of RRH-RFB, the existing model presented in [Chiaraviglio et al., 2018] (denoted by Chiaraviglio's Model) can be rewritten as follows:

$$\min \quad \sum_{n\in N}\sum_{q\in Q} c_q^{\mathrm{RRH}} y_{nq}^{\mathrm{RRH}} + \sum_{n_1\in N}\sum_{n_2\in N}\sum_{q\in Q}\left(c_q^{\mathrm{BBU}} v_{n_1n_2q}^{\mathrm{BBU}} + c_q^{\mathrm{MEC}} v_{n_1n_2q}^{\mathrm{MEC}}\right) \tag{4.1}$$

$$\text{s.t.} \quad \sum_{n\in N} x_{un} \le 1 \qquad \forall u \in U \tag{4.2}$$

$$\sum_{u\in U}\sum_{n\in N} x_{un} \ge \lceil \delta |U| \rceil \tag{4.3}$$

$$\sum_{q\in Q} y_{nq}^{\mathrm{RRH}} \le 1 \qquad \forall n \in N \tag{4.4}$$

$$x_{un} \le \sum_{q\in Q} y_{nq}^{\mathrm{RRH}} \qquad \forall u \in U, \forall n \in N \tag{4.5}$$

$$\sum_{u\in U} x_{un} \le \sum_{q\in Q} U_q^{\max} y_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \tag{4.6}$$

$$\sum_{n\in N} y_{nq}^{\mathrm{RRH}} \le A_q^{\mathrm{RRH}} \qquad \forall q \in Q \tag{4.7}$$

$$\sum_{n_1\in N}\sum_{n_2\in N} v_{n_1n_2q}^{\mathrm{BBU}} \le A_q^{\mathrm{BBU}} \qquad \forall q \in Q \tag{4.8}$$

$$\sum_{n_1\in N}\sum_{n_2\in N} v_{n_1n_2q}^{\mathrm{MEC}} \le A_q^{\mathrm{MEC}} \qquad \forall q \in Q \tag{4.9}$$

$$y_{n_2q}^{\mathrm{RRH}} \le \sum_{n_1\in N} v_{n_1n_2q}^{\mathrm{BBU}} \qquad \forall n_2 \in N, \forall q \in Q \tag{4.10}$$

$$v_{n_1n_2q}^{\mathrm{BBU}} \le y_{n_2q}^{\mathrm{RRH}} \qquad \forall n_1, n_2 \in N, \forall q \in Q \tag{4.11}$$

$$y_{n_2q}^{\mathrm{RRH}} \le \sum_{n_1\in N} v_{n_1n_2q}^{\mathrm{MEC}} \qquad \forall n_2 \in N, \forall q \in Q \tag{4.12}$$

$$v_{n_1n_2q}^{\mathrm{MEC}} \le y_{n_2q}^{\mathrm{RRH}} \qquad \forall n_1, n_2 \in N, \forall q \in Q \tag{4.13}$$

$$\phi_{un} \leq \sum_{q \in Q} \mathrm{CAP}_{unq} y_{nq}^{\mathrm{RRH}} \qquad \forall u \in U, \forall n \in N \qquad (4.14)$$

$$\phi_{un} \leq \mathrm{CAP}_{u}^{\max} x_{un} \qquad \forall u \in U, \forall n \in N \qquad (4.15)$$

$$\phi_{un} \leq t_u \qquad \forall u \in U, \forall n \in N \qquad (4.16)$$

$$\phi_{un} \geq t_u - (1 - x_{un})\mathrm{CAP}_{u}^{\max} \qquad \forall u \in U, \forall n \in N \qquad (4.17)$$

$$\mathrm{CAP}_{u}^{\max} = \max_{q \in Q, n \in N} \{\mathrm{CAP}_{unq}\} \qquad \forall u \in U \qquad (4.18)$$

$$\sum_{u \in U} \mathrm{CAP}_{unq} \theta_{unq} \leq \mathrm{CAP}_{q}^{\mathrm{RRH}} \qquad \forall n \in N, \forall q \in Q \qquad (4.19)$$

$$\theta_{unq} \leq x_{un} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \qquad (4.20)$$

$$\theta_{unq} \leq y_{nq}^{\mathrm{RRH}} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \qquad (4.21)$$

$$\theta_{unq} \geq x_{un} + y_{nq}^{\mathrm{RRH}} - 1 \qquad \forall u \in U, \forall n \in N, \forall q \in Q \qquad (4.22)$$

$$\sum_{u \in U} \sum_{n_2 \in N} \varphi_{un_1n_2q} \leq \mathrm{CAP}_{q}^{\mathrm{MEC}} \sum_{n_2 \in N} v_{n_1n_2q}^{\mathrm{MEC}} \qquad \forall n_1 \in N, \forall q \in Q \qquad (4.23)$$

$$\varphi_{un_1n_2q} \leq \mathrm{CAP}_{u}^{\max} v_{n_1n_2q}^{\mathrm{MEC}} \qquad \forall u \in U, \forall n_1, n_2 \in N, \forall q \in Q \qquad (4.24)$$

$$\varphi_{un_1n_2q} \leq \phi_{un_2} \qquad \forall u \in U, \forall n_1, n_2 \in N, \forall q \in Q \qquad (4.25)$$

$$\varphi_{un_1n_2q} \geq \phi_{un_2} - (1 - v_{n_1n_2q}^{\mathrm{MEC}})\mathrm{CAP}_{u}^{max} \qquad \forall u \in U, \forall n_1, n_2 \in N \qquad (4.26)$$

$$y_{n_1q}^{\mathrm{RRH}} + y_{n_2q}^{\mathrm{RRH}} \leq 1 \qquad \forall q \in Q, (n_1, n_2) \in \mathrm{CONF}_q \qquad (4.27)$$

$$\sum_{q \in Q} \sum_{n_2 \in N} v_{n_1n_2q}^{\mathrm{BBU}} \leq \sum_{q \in Q} y_{n_1q}^{\mathrm{RRH}} \qquad \forall n_1 \in N \qquad (4.28)$$

$$\sum_{q\in Q}\sum_{n_2\in N} v^{\mathrm{MEC}}_{n_1 n_2 q} \leq \sum_{q\in Q} y^{\mathrm{RRH}}_{n_1 q} \qquad \forall n_1 \in N \qquad (4.29)$$

$$t_u \geq t^{\mathrm{MIN}} x_{un} \qquad \forall u \in U, \forall n \in N \qquad (4.30)$$

$$x_{un} \in \{0,1\} \qquad \forall u \in U, \forall n \in N \qquad (4.31)$$

$$y^{\mathrm{RRH}}_{nq} \in \{0,1\} \qquad \forall n \in N, \forall q \in Q \qquad (4.32)$$

$$v^{\mathrm{BBU}}_{n_1 n_2 q} \in \{0,1\} \qquad \forall n_1, n_2 \in N, \forall q \in Q \qquad (4.33)$$

$$v^{\mathrm{MEC}}_{n_1 n_2 q} \in \{0,1\} \qquad \forall n_1, n_2 \in N, \forall q \in Q \qquad (4.34)$$

$$t_u \geq 0 \qquad \forall u \in U \qquad (4.35)$$

$$\phi_{un} \geq 0 \qquad \forall u \in U, \forall n \in N \qquad (4.36)$$

$$\theta_{unq} \geq 0 \qquad \forall u \in U, \forall n \in N, \forall q \in Q \qquad (4.37)$$

$$\varphi_{un_1 n_2 q} \geq 0 \qquad \forall u \in U, \forall n_1, n_2 \in N, \forall q \in Q \qquad (4.38)$$

The objective function minimizes the total installation cost of a 5G SF Network. Constraint (4.2) specifies that each user is served by at most one node. Constraint (4.3) assures that he minimum number of users has to be served. Constraint (4.4) indicates that at most one type of RRH-RFB can be installed in each node. Constraint (4.5) denotes that if the node is serving a user, a RRH-RFB then has to be installed on the node. Constraint (4.6) guarantees that the number of users served by each RRH-RFB is then bounded by the maximum number of users that can be supported by that RRH-RFB. Constraint (4.7)-(4.9) specify that the number of installed RFB module of type $q \in Q$ is then bounded by its maximum available number. Constraints (4.10) and (4.11) indicate that a BBU-RFB of type $q \in Q$ installed in node $n_1 \in N$ can be part of the chain serving the RRH-RFB of the same type installed in node $n_2 \in N$. Similarly, constraints (4.12) and (4.13) indicate that a MEC-RFB of type $q \in Q$ installed in node $n_1 \in N$ can be part of the

chain serving the RRH-RFB of the same type installed in node $n_2 \in N$. Constraint (4.14)-(4.18) specify that for each user, the amount of downlink traffic is then limited by the maximum radio link capacity on node $n \in N$. Constraint (4.19)-(4.22) indicate that for each node, the total capacity provided to the connected users has to be lower than the maximum total capacity managed by a RRH-RFB of type $q \in Q$. (4.23)-(4.26) denote that the total traffic from users connected to the RRH-RFB placed at node $n_2 \in N$ has to be lower than the maximum total capacity managed by a MEC-RFB of type $q \in Q$ in the chain. Constraint (4.27) specifies that if a pair $(n_1, n_2) \in \text{CONF}_q$, then at most one RRH-RFB of type q can be installed either in n_1 or in n_2. Constraint (4.28)-(4.29) make sure that the BBU-RFBs and MEC-RFBs can be installed only in nodes already storing RRH-RFB. Constraint (4.30) assures that the traffic assigned to a user has to be higher than a minimum value. Constraints (4.31)-(4.38) define the feasible domain of decision variables.

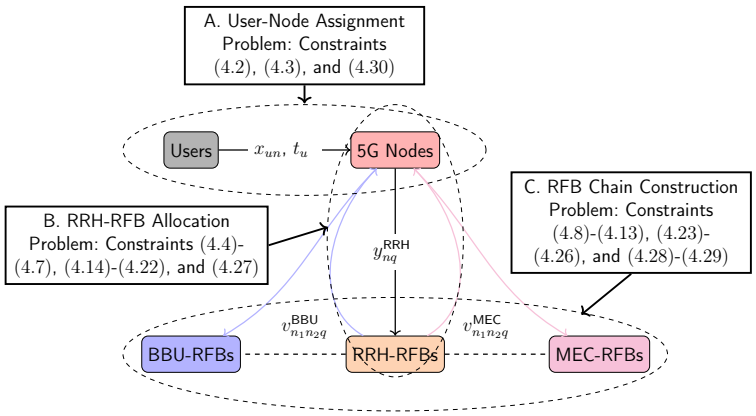


Figure 4.3.5 – Problem structure solved in [Chiaraviglio et al., 2018]

The problem solved in [Chiaraviglio et al., 2018] can be seen as composed of three parts (see Figure 4.3.5) : A) User-Node Assignment problem that optimally decides which user is served by which 5G node; B) RRH-RFB Allocation problem that optimally allocate the RRH-RFB at 5G node in order to serve the connected users; C) RFB Chain Construction problem that optimally places the BBU-RFBs and MEC-RFB to provide a complete RFB chain to satisfy the traffic demanded by users. However, there is an extremely high number of combinations of RFBs in the solution space, in particular concerning the RRH-RFB allocation problem and

RFB chain construction problem. In order to tackle this, we propose to represent in an alternative way the definition of RFB chains as presented in the next section.

4.3.2 A New Mathematical Model

New variables:

x_{nq}^{k}: A binary variable taking value 1, if an RFB module k is placed on node n serving a RFB chain in type q; 0, otherwise.

y_{unq}^{RRH}: A binary variable taking value 1, if an RRH-RFB is installed on node n serving an user u with a RFB chain in type q; 0, otherwise.

y_{unq}^{MEC}: A binary variable taking value 1, if a MEC-RFB is installed on node n serving an user u with a RFB chain in type q; 0, otherwise.

t_{unq}^{MEC}: A continuous variable indicating the amount of downlink traffic served to user u at node n with a MEC-RFB of type q, defined as $t_{unq}^{\mathrm{MEC}} \triangleq t_u y_{unq}^{\mathrm{MEC}}$.

Using the same notation, but the different decision variables, the compact model of RFB-based Resource Allocation problem under 5G Superfluid wireless Networks denoted by 5G-RFB-RA-$\mathcal{CP}1$ can be then described as below:

$$\min \quad \sum_{n\in N}\sum_{k\in K}\sum_{q\in Q} c_q^k x_{nq}^k \tag{4.39}$$

$$\text{s.t.} \quad \sum_{q\in Q} x_{nq}^k \leq 1 \qquad \forall n \in N, \forall k \in K \tag{4.40}$$

$$\sum_{n\in N} x_{nq}^k \leq A_{qk} \qquad \forall k \in K, \forall q \in Q \tag{4.41}$$

$$x_{n_1 q}^{\mathrm{RRH}} + x_{n_2 q}^{\mathrm{RRH}} \leq 1 \qquad \forall q \in Q, \forall (n_1, n_2) \in \mathrm{CONF}_q, n_1 \neq n_2 \tag{4.42}$$

$$y_{unq}^{\mathrm{RRH}} t_{\mathrm{MIN}} \leq x_{nq}^{\mathrm{RRH}} \mathrm{CAP}_{unq} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.43}$$

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \geq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \tag{4.44}$$

$$\sum_{n\in N}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \leq 1 \qquad \forall u \in U \tag{4.45}$$

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \leq \sum_{q\in Q} U_q^{\max} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \tag{4.46}$$

$$\sum_{u\in U}\sum_{n\in N}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \geq \lceil \delta |U| \rceil \tag{4.47}$$

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \mathrm{CAP}_{unq} \leq \sum_{q\in Q} \mathrm{CAP}_{q}^{\mathrm{RRH}} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \tag{4.48}$$

$$\sum_{n\in N} x_{nq}^{\mathrm{BBU}} \geq \sum_{n\in N} x_{nq}^{\mathrm{RRH}} \qquad \forall q \in Q \tag{4.49}$$

$$\sum_{q\in Q} x_{nq}^{\mathrm{BBU}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \tag{4.50}$$

$$\sum_{n\in N} x_{nq}^{\mathrm{MEC}} \geq \sum_{n\in N} x_{nq}^{\mathrm{RRH}} \qquad \forall q \in Q \tag{4.51}$$

$$\sum_{q\in Q} x_{nq}^{\mathrm{MEC}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \tag{4.52}$$

$$y_{unq}^{\mathrm{MEC}} \leq x_{nq}^{\mathrm{MEC}} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.53}$$

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{MEC}} \geq \sum_{q\in Q} x_{nq}^{\mathrm{MEC}} \qquad \forall n \in N \tag{4.54}$$

$$\sum_{n\in N}\sum_{q\in Q} y_{unq}^{\mathrm{MEC}} \leq 1 \qquad \forall u \in U \tag{4.55}$$

$$\sum_{n\in N}\sum_{q\in Q} t_{\mathrm{MIN}} y_{unq}^{\mathrm{RRH}} \leq t_u \leq \sum_{n\in N}\sum_{q\in Q} \mathrm{CAP}_{unq} y_{unq}^{\mathrm{RRH}} \qquad \forall u \in U \tag{4.56}$$

$$t_{unq}^{\mathrm{MEC}} \leq t_u \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.57}$$

$$t_{unq}^{\mathrm{MEC}} \leq y_{unq}^{\mathrm{MEC}} \mathrm{CAP}_{u}^{\max} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.58}$$

$$t_{unq}^{\mathrm{MEC}} \geq t_u + (y_{unq}^{\mathrm{MEC}} - 1)\mathrm{CAP}_{u}^{\max} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.59}$$

$$\sum_{u\in U}\sum_{q\in Q} t_{unq}^{\mathrm{MEC}} \leq \sum_{q\in Q} \mathrm{CAP}_{q}^{\mathrm{MEC}} x_{nq}^{\mathrm{MEC}} \qquad \forall n \in N \tag{4.60}$$

$$\mathrm{CAP}_{u}^{\max} = \max_{q\in Q, n\in N} \{\mathrm{CAP}_{unq}\} \qquad \forall u \in U \tag{4.61}$$

$$\sum_{u\in U}\sum_{n\in N}\sum_{q\in Q} y_{unq}^{\mathrm{MEC}} \geq \lceil \delta |U| \rceil \tag{4.62}$$

$$x_{nq}^{k} \in \{0,1\} \qquad \forall n \in N, \forall k \in K, \forall q \in Q \tag{4.63}$$

$$y_{unq}^{\mathrm{RRH}} \in \{0,1\} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.64}$$

$$y_{unq}^{\mathrm{MEC}} \in \{0,1\} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.65}$$

$$t_{unq}^{\mathrm{MEC}} \geq 0 \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.66}$$

$$t_{u} \geq 0 \qquad \forall u \in U \tag{4.67}$$

The objective function minimizes the total installation cost of a 5G SF Network. Constraints (4.40)-(4.41) specify each RFB module can be installed on a 5G node at most one of its type, and have an upper bound for available number of over the network in each type q. Constraint (4.42) denotes the conflict limitation of RRH-RFB installation in each type q due to radio interference. Constraints (4.43)-(4.45) assure that a user is served by at most one activated RRH-RFB placed on node n in type q, and only if the provided radio link capacity (CAP_{unq}) by this activated RRH-RFB to user is larger than the minimum required traffic demand (t_{MIN}). Constraint (4.46) specifies that activated RRH-RFB on node n has an upper bound for number of served users (U_q^{max}). Constraint (4.47) guarantees a minimum required coverage of users by activated RRH-RFBs in the network. Constraint (4.48) specifies that total radio link capacity of served users by an activated RRH-RFB on node n is then bounded by the maximal managed radio link capacity of this RRH-RFB. Constraints (4.49)-(4.50) denote the activation of BBU-RFBs. Similarly, constraints (4.51)-(4.52) denote the activation of MEC-RFBs. Constraints (4.53)-(4.55) guarantee that a user is served by at most one activated MEC-RFB placed on node n in type q. Constraint (4.56) denotes that the traffic of served user in real time is bounded by the minimum required traffic demand and maximum radio link capacity guaranteed by the activated RRH-RFB placed on node n. Constraints (4.57)-(4.60) specify that total traffic of served users by an activated MEC-RFB on node n is then bounded by the maximal managed traffic

capacity of this MEC-RFB. Constraint (4.62) guarantees a minimum required coverage of users by activated MEC-RFBs in the network. Constraints (4.63)-(4.67) define the feasible domain of decision variables. The problem structure of the new proposed model is presented in Figure 4.3.6.

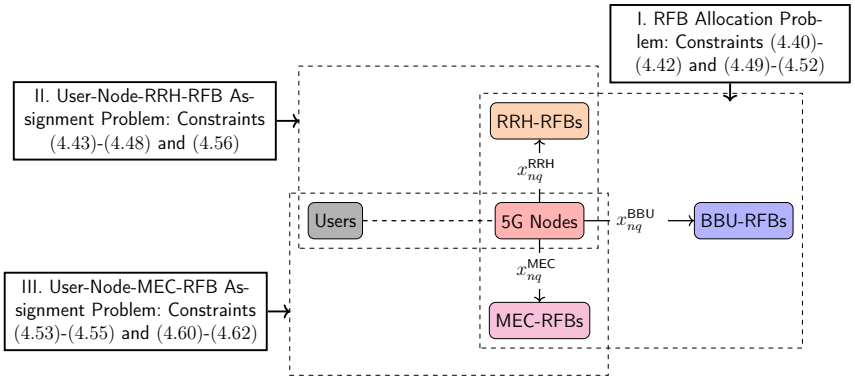


Figure 4.3.6 – Problem structure solved in the new proposed model

4.3.3 Simplification of the proposed Model

In an optimal solution of the model 5G-RFB-RA-$\mathcal{CP}$, the traffic of each user is set to t_{MIN}, as the objective function aims at minimizing minimize the total construction cost of a 5G SF network. Hence, constraints (4.56)-(4.60) can be rewritten as:

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\text{MEC}} t_{\text{MIN}} \leq \sum_{q\in Q} \text{CAP}_q^{\text{MEC}} x_{nq}^{\text{MEC}} \qquad \forall n \in N \qquad (4.68)$$

Furthermore, given decision value of x_{nq}^{MEC}, then $x_{nq}^{\text{MEC}} = 1, \forall n \in N$, we have:

$$\sum_{u\in U} y_{unq}^{\text{MEC}} \geq 1 \qquad \forall n \in N, \forall q \in Q \qquad (4.69)$$

$$\sum_{n\in N} y_{unq}^{\text{MEC}} \leq 1 \qquad \forall u \in U, \forall q \in Q \qquad (4.70)$$

$$\sum_{u\in U} y_{unq}^{\text{MEC}} t_{\text{MIN}} \leq \text{CAP}_q^{\text{MEC}} \qquad \forall n \in N, \forall q \in Q \qquad (4.71)$$

Then the maximum number of served users by the activated MEC-RFBs in

type q is an optimal solution of the following problem:

$$\max \quad \sum_{u\in U}\sum_{n\in N} y_{unq}^{\mathrm{MEC}} \tag{4.72}$$

$$\text{s.t.} \quad \sum_{n\in N} y_{unq}^{\mathrm{MEC}} \leq 1 \qquad \forall u \in U, \forall q \in Q \tag{4.73}$$

$$\sum_{u\in U} y_{unq}^{\mathrm{MEC}} t_{\mathrm{MIN}} \leq \mathrm{CAP}_q^{\mathrm{MEC}} \qquad \forall n \in N, \forall q \in Q \tag{4.74}$$

$$y_{unq}^{\mathrm{MEC}} \in \{0,1\} \qquad \forall u \in U, n \in N, q \in Q, \tag{4.75}$$

where constraint (4.73) is obviously implied by the objective function, is a so-called uniform 0-1 knapsack problem. Since all the weights of the items in the knapsack are identical, the problem is easy to solve and its optimal value is then defined by $\left\lfloor \frac{\mathrm{CAP}_q^{\mathrm{MEC}}}{t_{\mathrm{MIN}}} \right\rfloor$.

Therefore, the constraints (4.53)-(4.56), (4.62) and (4.65) can be replaced by the following constraints:

$$\sum_{n\in N}\sum_{q\in Q} x_{nq}^{\mathrm{MEC}} \left\lfloor \frac{\mathrm{CAP}_q^{\mathrm{MEC}}}{t_{\mathrm{MIN}}} \right\rfloor \geq \lceil \delta |U| \rceil \tag{4.76}$$

The overall mathematical model denoted by 5G-RFB-RA-$\mathcal{CP}2$ can be then rewritten as:

$$\min \quad \sum_{n\in N}\sum_{k\in K}\sum_{q\in Q} c_q^k x_{nq}^k \tag{4.77}$$

$$\text{s.t.} \quad \sum_{q\in Q} x_{nq}^k \leq 1 \qquad \forall n \in N, \forall k \in K \tag{4.78}$$

$$\sum_{n\in N} x_{nq}^k \leq A_{qk} \qquad \forall k \in K, \forall q \in Q \tag{4.79}$$

$$x_{n_1 q}^{\mathrm{RRH}} + x_{n_2 q}^{\mathrm{RRH}} \leq 1 \qquad \forall q \in Q, \forall (n_1, n_2) \in \mathrm{CONF}_q, n_1 \neq n_2 \tag{4.80}$$

$$y_{unq}^{\mathrm{RRH}} t_{\mathrm{MIN}} \leq x_{nq}^{\mathrm{RRH}} \mathrm{CAP}_{unq} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.81}$$

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \geq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \tag{4.82}$$

$$\sum_{n\in N}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \leq 1 \qquad \forall u \in U \qquad (4.83)$$

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \leq \sum_{q\in Q} U_q^{\max} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \qquad (4.84)$$

$$\sum_{u\in U}\sum_{n\in N}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \geq \lceil \delta |U| \rceil \qquad (4.85)$$

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \mathrm{CAP}_{unq} \leq \sum_{q\in Q} \mathrm{CAP}_q^{\mathrm{RRH}} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \qquad (4.86)$$

$$\sum_{n\in N} x_{nq}^{\mathrm{BBU}} \geq \sum_{n\in N} x_{nq}^{\mathrm{RRH}} \qquad \forall q \in Q \qquad (4.87)$$

$$\sum_{q\in Q} x_{nq}^{\mathrm{BBU}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \qquad (4.88)$$

$$\sum_{n\in N} x_{nq}^{\mathrm{MEC}} \geq \sum_{n\in N} x_{nq}^{\mathrm{RRH}} \qquad \forall q \in Q \qquad (4.89)$$

$$\sum_{q\in Q} x_{nq}^{\mathrm{MEC}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \qquad (4.90)$$

$$\sum_{n\in N}\sum_{q\in Q} x_{nq}^{\mathrm{MEC}} \left\lfloor \frac{\mathrm{CAP}_q^{\mathrm{MEC}}}{t_{\mathrm{MIN}}} \right\rfloor \geq \lceil \delta |U| \rceil \qquad (4.91)$$

$$x_{nq}^{k} \in \{0,1\} \qquad \forall n \in N, \forall k \in K, \forall q \in Q \qquad (4.92)$$

$$y_{unq}^{\mathrm{RRH}} \in \{0,1\} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \qquad (4.93)$$

4.3.4 Deriving new valid inequalities

Considering the RRH-RFBs allocation, the explicit upper bound of maximum number of served users of an activated RRH-RFB in type of q is equal to U_q^{max}. The minimum number of activated RRH-RFBs is then defined by:

$$\sum_{n\in N}\sum_{q\in Q} x_{nq}^{\mathrm{RRH}} U_q^{\max} \geq \lceil \delta |U| \rceil \qquad (4.94)$$

Naturally, the maximum number of served users of an activated RRH-RFB in

type q is actually limited by the minimal radio link capacity of eligible served users (where $\mathrm{CAP}_{unq} \geq t_{\mathrm{MIN}}$), hence $\mathrm{UB}_q = \left\lfloor \frac{\mathrm{CAP}_q^{\mathrm{RRH}}}{\min\{\mathrm{CAP}_{unq}:\ \mathrm{CAP}_{unq} \geq t_{\mathrm{MIN}}, \forall u \in U\}} \right\rfloor$. A better upper bound is then $U_q^{\max} = \min\left\{U_q^{\max}, \mathrm{UB}_q\right\}$, therefore, a stronger constraint and valid inequality are obtained compared to constraint 4.84 and (4.94) by writing:

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \min\left\{U_q^{\max}\mathrm{UB}_q\right\} \quad \forall n \in N \tag{4.95}$$

$$\sum_{n\in N}\sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \min\left\{U_q^{\max}, \mathrm{UB}_q\right\} \geq \lceil \delta |U| \rceil \tag{4.96}$$

Similarly to MEC-RFB allocation, for a given valorization of x_{nq}^{RRH}, then $x_{nq}^{\mathrm{RRH}} = 1, \forall n \in N$ and y_{unq}^{RRH} for $\mathrm{CAP}_{unq} \geq t_{\mathrm{MIN}}$ imply:

$$\sum_{u\in U} y_{unq}^{\mathrm{RRH}} \geq 1 \qquad \forall n \in N, \forall q \in Q \tag{4.97}$$

$$\sum_{n\in N} y_{unq}^{\mathrm{RRH}} \leq 1 \qquad \forall u \in U, \forall q \in Q \tag{4.98}$$

$$\sum_{u\in U} y_{unq}^{\mathrm{RRH}} \mathrm{CAP}_{unq} \leq \mathrm{CAP}_q^{\mathrm{RRH}} \qquad \forall n \in N, \forall q \in Q \tag{4.99}$$

Then, the maximum number of served users by the activated RRH-RFBs in type q (UB_{nq}) is an optimal solution of the problem below:

$$\max \quad \sum_{u\in U}\sum_{n\in N} y_{unq}^{\mathrm{RRH}} \tag{4.100}$$

$$\text{s.t.} \quad \sum_{n\in N} y_{unq}^{\mathrm{RRH}} \leq 1 \qquad \forall u \in U, \forall q \in Q \tag{4.101}$$

$$\sum_{u\in U} y_{unq}^{\mathrm{RRH}} \mathrm{CAP}_{unq} \leq \mathrm{CAP}_q^{\mathrm{RRH}} \qquad \forall n \in N, \forall q \in Q \tag{4.102}$$

$$y_{unq}^{\mathrm{RRH}} \in \{0,1\} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.103}$$

which is a uniform knapsack problem [Martello, 1990] that can be solved polynomially via sorting increasingly the selected users ($\mathrm{CAP}_{unq} > t_{\mathrm{MIN}}$) by their size. So a tighter upper bound and valid inequality can be defined as:

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \min\left\{U_q^{\max}, \mathrm{UB}_{nq}\right\} \quad \forall n \in N \tag{4.104}$$

$$\sum_{n\in N}\sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \min\left\{U_q^{\max}, \mathrm{UB}_{nq}\right\} \geq \lceil \delta |U| \rceil \tag{4.105}$$

Thus, the overall mathematical model denoted by 5G-RFB-RA-$\mathcal{CP}3$ can be rewritten as:

$$\min \quad \sum_{n\in N}\sum_{k\in K}\sum_{q\in Q} c_q^k x_{nq}^k \tag{4.106}$$

$$\text{s.t.} \quad \sum_{q\in Q} x_{nq}^k \leq 1 \qquad \forall n \in N, \forall k \in K \tag{4.107}$$

$$\sum_{n\in N} x_{nq}^k \leq A_{qk} \qquad \forall k \in K, \forall q \in Q \tag{4.108}$$

$$x_{n_1 q}^{\mathrm{RRH}} + x_{n_2 q}^{\mathrm{RRH}} \leq 1 \qquad \forall q \in Q, \forall (n_1, n_2) \in \mathrm{CONF}_q, n_1 \neq n_2 \tag{4.109}$$

$$y_{unq}^{\mathrm{RRH}} t_{\mathrm{MIN}} \leq x_{nq}^{\mathrm{RRH}} \mathrm{CAP}_{unq} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.110}$$

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \geq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \tag{4.111}$$

$$\sum_{n\in N}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \leq 1 \qquad \forall u \in U \tag{4.112}$$

$$\sum_{u\in U}\sum_{n\in N}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \geq \lceil \delta |U| \rceil \tag{4.113}$$

$$\sum_{n\in N} x_{nq}^{\mathrm{BBU}} \geq \sum_{n\in N} x_{nq}^{\mathrm{RRH}} \qquad \forall q \in Q \tag{4.114}$$

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \mathrm{CAP}_{unq} \leq \sum_{q\in Q} \mathrm{CAP}_q^{\mathrm{RRH}} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \tag{4.115}$$

$$\sum_{q\in Q} x_{nq}^{\mathrm{BBU}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \tag{4.116}$$

$$\sum_{n\in N} x_{nq}^{\mathrm{MEC}} \geq \sum_{n\in N} x_{nq}^{\mathrm{RRH}} \qquad \forall q \in Q \tag{4.117}$$

$$\sum_{q\in Q} x_{nq}^{\mathrm{MEC}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \tag{4.118}$$

$$\sum_{n\in N}\sum_{q\in Q} x_{nq}^{\mathrm{MEC}} \left\lfloor \frac{\mathrm{CAP}_q^{\mathrm{MEC}}}{t_{\mathrm{MIN}}} \right\rfloor \geq \lceil \delta |U| \rceil \tag{4.119}$$

$$\sum_{u\in U}\sum_{q\in Q} y_{unq}^{\mathrm{RRH}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} min\left\{U_q^{\max}, \mathrm{UB}_{nq}\right\} \qquad \forall n \in N \tag{4.120}$$

$$\sum_{n\in N}\sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \min\left\{U_q^{\max}, \mathrm{UB}_{nq}\right\} \geq \lceil \delta |U| \rceil \tag{4.121}$$

$$x_{nq}^{k} \in \{0,1\} \qquad \forall n \in N, \forall k \in K, \forall q \in Q \tag{4.122}$$

$$y_{unq}^{\mathrm{RRH}} \in \{0,1\} \qquad \forall u \in U, \forall n \in N, \forall q \in Q \tag{4.123}$$

4.4 A Benders decomposition Approach

Benders Decomposition is a major solution method used for optimization problems proposed by Benders, which has been intensively investigated over the five decades and used in many different application contexts, such as power system and network design ([Alguacil and Conejo, 2000], [Binato et al., 2001], [Costa, 2005], [Shahidehopour and Fu, 2005], [Fortz and Poss, 2009], [Rahmaniani et al., 2018]), planning and scheduling ([Hooker, 2007]), routing and scheduling ([Cordeau et al., 2001], [Mercier et al., 2005], [Cao et al., 2010]). For an exhaustive introduction to it, we refer the reader to the survey [Rahmaniani et al., 2017]. Here, we proceed to recall some fundamentals.

Let us consider the following general LP:

$$\min \quad f^T y + c^T x \tag{4.124}$$

$$\text{s.t.} \quad By + Dx = d \tag{4.125}$$

$$Ay = b \tag{4.126}$$

$$x \geq 0 \tag{4.127}$$

$$y \geq 0 \text{ and integer}, \tag{4.128}$$

where $x \in \mathbb{R}^+$ is a real valued variable and $y \in \mathbb{Z}^+$ is a non-negative integer complicating variable whose domain is defined by polyhedron $\mathbb{Y} : \{Ay = b\}$ with a known matrix A and a given vector b. A linking constraint $By + Dx = d$ between x and y must be satisfied with known matrix B and D for the associated x and y variables, and a given vector d. The objective function here is to minimize the total cost with the given cost vector f and c associated with x and y variables, respectively.

The Benders decomposition method partitions the problem in two: a master problem containing the y variables and a subproblem containing the x variables. With q(y) as the incumbent value for the x part, thus, we can define a LP using only variable y:

$$\min \quad f^T y + q(y) \tag{4.129}$$

$$\text{s.t.} \quad Ay = b \tag{4.130}$$

$$y \geq 0 \text{ and integer} \tag{4.131}$$

Then, we have the subproblem in terms of x. Note that if the subproblem is unbounded, the original problem is unbounded as well. If the problem is bounded, we calculate the value of $q(y)$ by solving the following LP:

$$\min \quad c^T x \tag{4.132}$$

$$\text{s.t.} \quad Dx = d - By \tag{4.133}$$

$$x \geq 0 \tag{4.134}$$

Considering the dual variable π, associated with the subproblem, the dual form of $q(y)$ can be rewritten as:

$$\max \quad \pi^T (d - By) \tag{4.135}$$

$$\text{s.t.} \quad D^T \pi \leq c \tag{4.136}$$

$$\pi \in \mathbb{R} \tag{4.137}$$

When the solution space is not empty, we can enumerate all extreme rays

(denoted by a vector ρ) and extreme points (denoted by a vector π) of the feasible region $D^T\pi \leq c$. Given a solution vector y^*, we can solve the dual problem by checking if we can find that:

1. a $\rho_i, i \in I$ such that $\rho_i(d - By^*) > 0$, in which case the dual problem of $q(y^*)$ is unbounded;

2. a $\pi_j, j \in J$ maximizing $\pi_j(d - By^*)$, in which case both the primal and dual of $q(y^*)$ reach their optimality at this extreme point.

In the former case, there is a direction of boundlessness ρ_i, this should be avoided due to the infeasibility of y in the master problem. A feasible cut $\rho_i(d - By^*) \leq 0$ is then added to the master problem to restrict the movement in this direction. In the latter case, the solution of $q(y^*)$ (denoted by q) is one of the extreme points $\pi_j, j \in J$. With the above notation of ρ an π, the master problem can be rewritten as:

$$\min \quad f^T y + q \tag{4.138}$$

$$\text{s.t.} \quad Ay = b \tag{4.139}$$

$$\rho_i(d - By) > 0 \qquad \forall i \in I \tag{4.140}$$

$$\pi_j(d - By) \leq q \qquad \forall j \in J \tag{4.141}$$

$$q \geq 0 \qquad \forall q \in \mathbb{R} \tag{4.142}$$

$$y \geq 0 \text{ and integer} \tag{4.143}$$

Since there is an exponential number of extreme points and extreme rays, and because their enumeration is $\mathcal{NP}$-hard, Benders decomposition starts without any cut and solves a relaxed master problem which gives an eligible candidate solution (y^*, q^*). Taking this solution, it solves the subproblem to obtain an optimal value $q(y^*)$. If q^* is equal to $q(y^*)$ then the candidate solution is optimal for the original problem. Otherwise, two cases may occur:

1. the dual is unbounded, then we select an extreme ray to generate a feasibility cut (i.e., (4.140));

2. the subproblem is solved optimally given a solution of y, then we select an extreme point to generate an optimality cut (i.e., (4.141)), which gives a lower bound of the incumbent q.

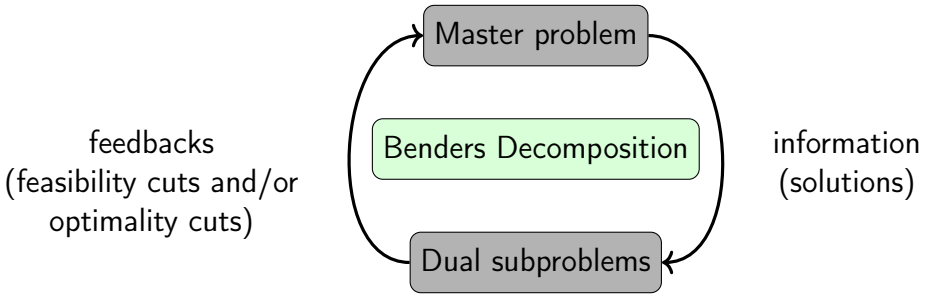


Figure 4.4.7 – General schema of Benders decomposition method

A general scheme of Benders decomposition is presented in Figure 4.4.7. The algorithm is based on an iterative constraint generation approach that runs until a convergence condition is met (e.g., $q^* = q(y^*)$), during which the objective value of the master problem provides an upper bound on the global solution while the value of the subproblem gives a lower bound at each iteration.

The key to generate a so-called Benders cut is to construct the inference dual (which is the problem of inferring a strongest possible bound from the constraint set, first proposed by Hooker and Yan) of the associated subproblems. In the case of BLPsubproblem, a logic-based Benders decomposition was proposed in [Hooker and Ottosson, 2003] to obtain an inference dual β (e.g., obtained by a branch-and-bound algorithm with a given solution $y*$) such that $f^T y * + c^T x \geq \beta$ provides a tighter lower bound of the master problem. However, when the integer or boolean decision variables do not appear in the objective function of the original problem, the logic-based Benders decomposition could be useless because only feasibility cut will be generated, the inference dual cannot be obtained via the subproblems as no need of optimality cuts. The Combinatorial Benders decomposition has been proposed in [Codato and Fischetti, 2006] to tackle a set of conditional linear constraints (i.e., if $x_i = 1$, then $A_i y = b_i$) between decision variables x and y occurred in the original problem. Thanks to the minimal (or irreducible) infeasible subsystem of the associated constraints of variables y, if the involved subproblem is infeasible, then we can observe that at least one binary variable x_i has to be

changed to break the infeasibility, leading to the definition of a Combinatorial Benders' cut: $\sum_{i:x_i^*=0} x_i + \sum_{i:x_i^*=1}(1-x_i) \geq 1$.

Trying to apply these considerations to our Superfluid resource allocation problem, we can first note that our problem can be seen as made up of two main problems. One is the RFB allocation problem (solution space X), the other is the User-RRH-RFB assignment problem (solution space Y) for a given solution of RRH-RFBs. Both the problems are BLP problems. Therefore, we propose a hybrid Benders decomposition to first project solution space $\Omega^r(X,Y)$ (where decision variable in Y is relaxed) into $\Pi(X)$. For an optimal solution s^* from $\Pi(X)$ is feasible for Y, then we check whether the obtained decision values from Y are integral or not; otherwise we add a feasible cut into $\Pi(X)$ found from above infeasibility. If the integrality requirement is met, then s^* is also an optimal solution for the whole problem; otherwise, we add a Combinatorial Benders cut to $\Pi(X)$, indicating at most one changeable decision variables that should be considered in the master problem.

More formally, we identify the RFB allocation problem as our master problem (denoted by P_M) in Benders decomposition and the User-RRH-RFB assignments problem as the slave problem (or subproblem) (denoted by $P_S(x_{nq}^{\mathrm{RRH}^*})$). The master problem can be written as follows:

$$\min \quad \sum_{n\in N}\sum_{k\in K}\sum_{q\in Q} c_q^k x_{nq}^k \tag{4.144}$$

$$\text{s.t.} \quad \sum_{q\in Q} x_{nq}^k \leq 1 \qquad \forall n \in N, \forall k \in K \tag{4.145}$$

$$\sum_{n\in N} x_{nq}^k \leq A_{qk} \qquad \forall k \in K, \forall q \in Q \tag{4.146}$$

$$x_{n_1q}^{\mathrm{RRH}} + x_{n_2q}^{\mathrm{RRH}} \leq 1 \qquad \forall q \in Q, \forall (n_1,n_2) \in \mathrm{CONF}_q, n_1 \neq n_2 \tag{4.147}$$

$$\sum_{n\in N}\sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \min\left\{U_q^{\max}, U_{nq}\right\} \geq \lceil \delta |U| \rceil \tag{4.148}$$

$$\sum_{n\in N} x_{nq}^{\mathrm{BBU}} \geq \sum_{n\in N} x_{nq}^{\mathrm{RRH}} \qquad \forall q \in Q \tag{4.149}$$

$$\sum_{q\in Q} x_{nq}^{\mathrm{BBU}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \qquad (4.150)$$

$$\sum_{n\in N} x_{nq}^{\mathrm{MEC}} \geq \sum_{n\in N} x_{nq}^{\mathrm{RRH}} \qquad \forall q \in Q \qquad (4.151)$$

$$\sum_{q\in Q} x_{nq}^{\mathrm{MEC}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}} \qquad \forall n \in N \qquad (4.152)$$

$$\sum_{n\in N}\sum_{q\in Q} x_{nq}^{\mathrm{MEC}} \left\lfloor \frac{\mathrm{CAP}_q^{\mathrm{MEC}}}{t_{\mathrm{MIN}}} \right\rfloor \geq \lceil \delta |U| \rceil \qquad (4.153)$$

$$x_{nq}^{k} \in \{0,1\} \qquad \forall n \in N, \forall k \in K, \forall q \in Q, \qquad (4.154)$$

where the objective function minimizes the total installation cost. Constraint (4.145) indicates that one 5G node can hold at most one RRH-RFB, one BBU-RFB, and one MEC-RFB. Constraint (4.146) specifies the available number of each RFB module. Constraint (4.147) denotes the interference constraint among RRH-RFB of each type placed on the node. Constraints (4.148) and (4.153) specify the minimum number of activated RRH-RFBs and BBU-RFBs. Constraints (4.149)-(4.152) indicate the activation of BBU-RFBs and MEC-RFBs. Finally, constraint (4.154) defines the domain of decision variables x.

The slave problem is instead:

$$\max \quad \sum_{u\in U}\sum_{n\in N} y_{un} \qquad (4.155)$$

$$\text{s.t.} \quad y_{un} t_{\mathrm{MIN}} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}^*} \mathrm{CAP}_{unq} \qquad \forall u \in U, n \in N \qquad (4.156)$$

$$\sum_{n\in N} y_{un} \leq 1 \qquad \forall u \in U \qquad (4.157)$$

$$\sum_{u\in U} y_{un} \leq \sum_{q\in Q} \min\left\{U_q^{\max}, U_{nq}\right\} x_{nq}^{\mathrm{RRH}^*} \qquad \forall n \in N \qquad (4.158)$$

$$\sum_{u\in U}\sum_{q\in Q} y_{un} x_{nq}^{\mathrm{RRH}^*} \mathrm{CAP}_{unq} \leq \sum_{q\in Q} \mathrm{CAP}_q^{\mathrm{RRH}} x_{nq}^{\mathrm{RRH}^*} \qquad \forall n \in N \qquad (4.159)$$

$$y_{un} \in \{0,1\} \qquad \forall u \in U, \forall n \in N, \qquad (4.160)$$

in which the objective function maximizes the number of users served by the activated RRH-RFBs. Constraint (4.156) indicates the minimum traffic demand required by users. Constraint (4.157) specifies that one user can be served at most by one 5G node. Constraint (4.158) gives an upper bound of served number of users by the activated RRH-RFBs. Constraint (4.159) denotes that the total served traffic at a 5G node by an RRH-RFB should not violate its capacity. Finally, constraint (4.160) defines the domain of decision variables y.

If the solution cost of $P_S(x_{nq}^{\mathrm{RR}H^*})$, denoted by $Z_{P_S}^*(x_{nq}^{\mathrm{RRH}^*})$, satisfies $Z_{P_S}^*(x_{nq}^{\mathrm{RRH}^*}) < \lceil \delta|U| \rceil$, then P_M is infeasible; otherwise P_M is optimal. For the former case,

$$\sum_{x_{nq}^{\mathrm{RRH}^*}=1:n\in N,\ q\in Q} (1 - x_{nq}^{\mathrm{RRH}}) + \sum_{x_{nq}^{\mathrm{RRH}^*}=0:n\in N,\ q\in Q} x_{nq}^{\mathrm{RRH}} \geq 1 \tag{4.161}$$

$$\sum_{x_{nq}^{\mathrm{RRH}^*}=1:n\in N,\ q\in Q} x_{nq}^{\mathrm{RRH}} U_{nq}^* + \sum_{x_{nq}^{\mathrm{RRH}^*}=0:n\in N,\ q\in Q} x_{nq}^{\mathrm{RRH}} \min\left\{U_q^{\max}, U_{nq}\right\} \geq \lceil \delta|U| \rceil \tag{4.162}$$

are the feasibility Benders cuts, where U_{nq}^* is obtained from P_M. Furthermore, consider the linear relaxation of $P_S(x_{nq}^{\mathrm{RRH}^*})$ (denoted by $P_{SC}(x_{nq}^{\mathrm{RRH}^*})$), if the maximum number of served users by the given activated RRH-RFBs is less than the desired covered number of users, then the master problem is infeasible as the solution cost of an LP problem is always less than or equal to the cost of its linear relaxation variant in sense of maximization optimization. In this case, the Benders feasibility cut (i.e., constraints (4.161) and (4.162)) can be added into the master problem such that at least one RRH-RFB assignment changed, and the user coverage is guaranteed by this new change of x_{nq}^{RRH}. Let us then consider the linear relaxation of $P_S(x_{nq}^{\mathrm{RRH}^*})$, denoted by $P_{SC}(x_{nq}^{\mathrm{RRH}^*})$, which can be written as below:

$$\max \quad \sum_{u\in U}\sum_{n\in N} y_{un} \tag{4.163}$$

$$\text{s.t.} \quad [\lambda_{un} \geq 0] \quad y_{un} \leq \sum_{q\in Q} x_{nq}^{\mathrm{RRH}^*} \left\lfloor \frac{\mathrm{CAP}_{unq}}{t_{\mathrm{MIN}}} \right\rfloor \qquad \forall u \in U, \forall n \in N \tag{4.164}$$

$$[\xi_u \geq 0] \quad \sum_{n\in N} y_{un} \leq 1 \qquad \forall u \in U \tag{4.165}$$

$$[\eta_n \geq 0] \quad \sum_{u\in U} y_{un} \leq \sum_{q\in Q} \min\left\{U_q^{\max}, U_{nq}\right\} x_{nq}^{\mathrm{RRH}^*} \qquad \forall n \in N \tag{4.166}$$

$$[\pi_n \geq 0] \sum_{u\in U}\sum_{q\in Q} y_{un} x_{nq}^{\mathrm{RRH}^*} \mathrm{CAP}_{unq} \leq \sum_{q\in Q} \mathrm{CAP}_q^{\mathrm{RRH}} x_{nq}^{\mathrm{RRH}^*} \quad \forall n \in N \tag{4.167}$$

$$y_{un} \geq 0 \qquad \forall u \in U, \forall n \in N \tag{4.168}$$

Let $\lambda_{un} \geq 0$, $\xi_u \geq 0$, $\eta_n \geq 0$, $\pi_n \geq 0, \forall u \in U, \forall n \in N$ be the dual variables associated with constraints (4.156), (4.157), (4.158), and (4.159), respectively. Then the dual form of $P_{SC}(x_{nq}^{\mathrm{RRH}^*})$, denoted by $P_{DSC}(x_{nq}^{\mathrm{RRH}^*})$, can be written as:

$$\min \sum_{u\in U}\sum_{n\in N}\sum_{q\in Q} \lambda_{un} x_{nq}^{\mathrm{RRH}^*} \left\lfloor \frac{\mathrm{CAP}_{unq}}{t_{\mathrm{MIN}}} \right\rfloor + \sum_{u\in U} \xi_u \tag{4.169}$$

$$+ \sum_{n\in N}\sum_{q\in Q} \mathrm{CAP}_q^{\mathrm{RRH}} x_{nq}^{\mathrm{RRH}^*} \pi_n + \sum_{n\in N}\sum_{q\in Q} \min\left\{U_q^{\max}, U_{nq}\right\} x_{nq}^{\mathrm{RRH}^*} \eta_n \tag{4.170}$$

$$\text{s.t.} \quad \pi_n \sum_{q\in Q} \mathrm{CAP}_{unq} x_{nq}^{\mathrm{RRH}^*} + \eta_n + \xi_u + \lambda_{un} \geq 1 \qquad \forall u \in U, \forall n \in N \tag{4.171}$$

$$\lambda_{un} \geq 0 \qquad \forall u \in U, \forall n \in N \tag{4.172}$$

$$\eta_n \geq 0 \qquad \forall n \in N \tag{4.173}$$

$$\pi_n \geq 0 \qquad \forall n \in N \tag{4.174}$$

$$\xi_u \geq 0 \qquad \forall u \in U \tag{4.175}$$

Focus on the solution cost, denoted by $Z^*_{P_{DSC}}(x_{nq}^{RRH^*})$, obtained from the dual form of relaxation variant of the subproblem, if we have $Z^*_{P_{DSC}}(x_{nq}^{RRH^*}) < \lceil \delta |U| \rceil$ holding, then P_M is clearly infeasible. Then a Benders feasibility cut is generated as below:

$$\sum_{u\in U}\sum_{n\in N}\sum_{q\in Q} \lambda^*_{un} x_{nq}^{\mathrm{RRH}} \left\lfloor \frac{\mathrm{CAP}_{unq}}{t_{\mathrm{MIN}}} \right\rfloor + \sum_{u\in U} \xi^*_u + \sum_{n\in N}\sum_{q\in Q} \mathrm{CAP}_q^{\mathrm{RRH}} x_{nq}^{\mathrm{RRH}} \pi^*_n$$
$$+ \sum_{n\in N}\sum_{q\in Q} \min\left\{U_q^{\max}, U_{nq}\right\} x_{nq}^{\mathrm{RRH}} \eta^*_n \geq \lceil \delta |U| \rceil \tag{4.176}$$

With the notation introduced above, the proposed Benders decomposition can be described as is Algorithm 22. The main idea is to solve first the RFB allocation problems with the known explicit upper bound of maximum served users by an activated RRH-RFB. Using the solution of the RFB allocation problem, especially,

we try to solve the linear relaxation variant of the involved problem. If the user coverage constraint is violated, then a Benders feasibility cut (4.176) is added into the master problem to improve the lower bound of total number of served users by a realized configuration of RRH-RFB allocation. Otherwise, we solve the associate subproblem with a given solution of RRH-RFB allocation. If the user coverage constraint is violated, then a Benders feasibility cut (4.162) is added to the master problem to make at least one change of RRH-RFB allocation to eliminate the incumbent infeasible solution. Otherwise, the optimal solution is obtained for the original problem.

Algorithm 5: Hybrid Benders Decomposition

```
Solve P_M;
if P_M infeasible or unbounded then
    Stop;
else
    isSolved ← False;
    repeat
        Obtain x_nq^RRH* from solved P_M;
        Solve P_SC(x_nq^RRH*);
        if Z*_{P_SC}(x_nq^{RRH*}) < ⌈δ|U|⌉ then
            Add a feasible benders cut (4.176) into P_M;
            Solve P_M;
        else
            Solve P_S(x_nq^RRH*);
            if Z*_{P_S}(x_nq^{RRH*}) < ⌈δ|U|⌉ then
                Add the feasible benders cuts (4.161) and (4.162) into P_M;
                Solve P_M;
            else
                isSolved ← True;
            end
        end
    until isSolved;
end
```

4.5 Computational Results

4.5.1 5G test scenarios

In order to evaluate the performance of the proposed decomposition approach, we considered a 5G scenario including a hexagonal cellular geometry as in [Marzetta, 2010]. In this scenario, 9x9 candidate 5G nodes were considered, distributed according to a regular grid, and the users are quasi-uniformly located around the center of this network as shown in Fig.4.5.8.

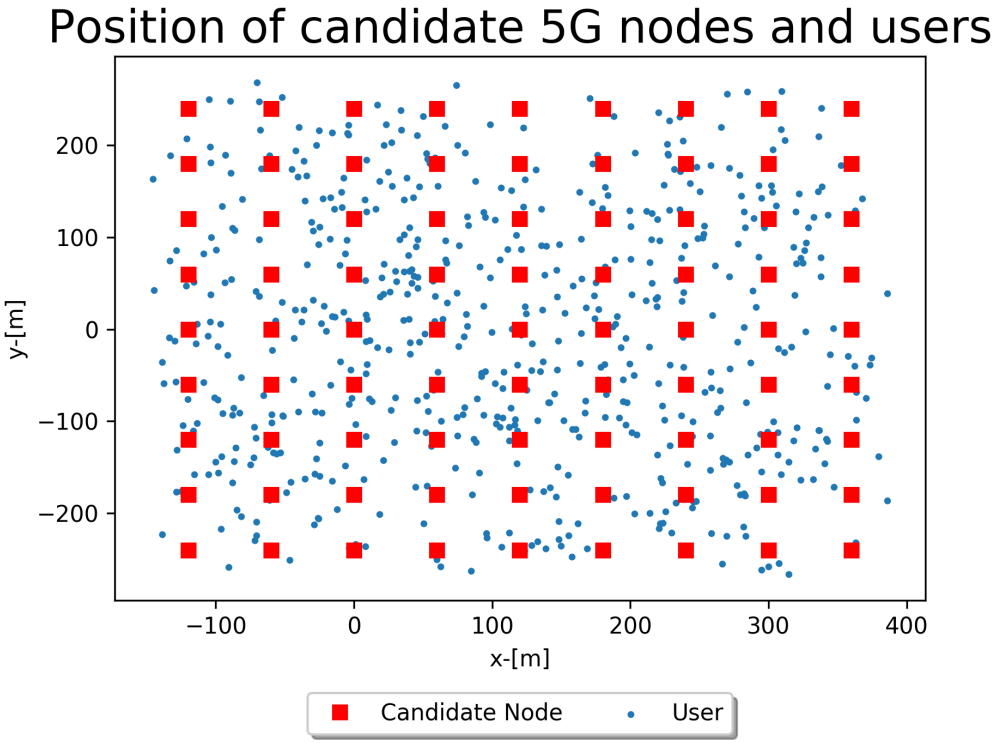


Figure 4.5.8 – A 5G network instance with 81 candidate nodes and 500 users

In the test instances that we considered, the number of users varies from 25 to 100 with a step of 25, from 100 to 300 with a step of 50, and from 500 to 1000 with a step of 250. In line with this, 11 5G instances have been derived, in which we set up the minimum user downlink traffic t_{MIN} varying from 10 to 50 Mbps with a step of 20, and the minimum required ratio of coverage $\alpha \in [0.1, 1.0]$ with a step of 0.1. Furthermore, concerning the computational settings of the machine

Table 4.5.1 – Input parameters in line with these presented in [Chiaraviglio et al., 2018]

Parameters	value	
	Micro	Macro
U_q^{MAX}	42	126
A_q^{RRH}	81	5
A_q^{BBU}	81	5
A_q^{MEC}	81	5
c_q^{RRH}	53951[€]	133951[€]
c_q^{BBU}	440[€]	1307[€]
c_q^{MEC}	440[€]	1307[€]
c_q^{RFBC}	54831[€]	136565[€]
δ_{unq}	calculated from model in [Marzetta, 2010]	
δ_q^{RRH}	10[Gbps]	30[Gbps]
δ_q^{MEC}	30[Gbps]	30[Gbps]

that we used, we imposed a limit of 8Gb of maximum virtual memory and a time limit limitation of 900 seconds for experimentation (based on discussions with experts of the Superfluid architecture taking into account realistic real-world requirements). The value of other coefficients and parameters appearing in the problem are detailed in Table 4.5.1, where M denotes the mathematical model involved in [Chiaraviglio et al., 2018], AM specifies the proposed model (5G-RFB-RA-$\mathcal{CP}1$), and AMS is the simplified version (i.e., 5G-RFB-RA-$\mathcal{CP}3$), respectively. N^{opt} calculates the number of optimal solutions, N^{infeas} counts the number of infeasible solutions, $N^{timeout}$ indicates the instances exceeding time limitation, and N^{memout} specifies the number of instances that cannot be solved within 8G virtual memory. Moreover, T_{avg}^{opt} and T_{avg}^{infeas} indicate the total averaged elapsed time on seconds for an optimal solution and the infeasible one.

The reference formulation proposed in [Chiaraviglio et al., 2018] may present a performance that can be very costly in terms of computational time and RAM memory. The new formulation that we have proposed has instead the advantage of being able to identify an optimal solution for a much larger number of instances as it can be seen from the table of results. This improved performance can be attributed to the new valid inequalities that we have characterized and that allow to more effectively catch and express the Combinatorial relations that link the

Table 4.5.2 – Comparison on solved instances for each method

	M	AM	AMS
N^{opt}	6	142	212
N^{infeas}	-	6	21
$N^{timeout}$	114	122	37
N^{memout}	150	-	-
$T^{opt}_{avg}(s)$	737.98	126.18	45.99
$T^{infeas}_{avg}(s)$	-	292.84	104.44

different types of Reusable Functional Blocks (RRH-RFB, BBU-RFB and MEC-RFB). However, it should also be noted that some hard instances cannot still be solved within the time limit, thus encouraging us to refine and deepen the analysis of the new proposed approach as future work. Furthermore, we intend also to extend the tests to instances based on different 5G scenarios, thus obtaining the possibility of obtaining further insights about the behavior of the algorithm.

5. Green and Robust 5G Virtual Network Function Placement Problem

5.1 Introduction

The Fifth Generation of wireless telecommunications systems, widely known as 5G, has attracted a lot of attention in recent times, since it is largely considered as a crucial element for a full realization of a digital society and a critical technology to support the deployment of smart cities (see, for example, the work of the European Commission about this topic, e.g. [European-5G-Observatory]). 5G is going to offer enhanced service performances unknown to previous wireless technologies, such as data rates of at least 40 Mbps for tens of thousands of users, data rates of 100 Mbps for metropolitan areas, enhanced spectral efficiency and a dramatic reduction of latency (see e.g, [Larsson, 2018, Dahlman et al., 2020]).

5G will be strongly based on Network Function Virtualization (NFV), according to which network functions run on a set of Virtual Machines that are hosted in cheap commodity hardware servers [Abdelwahab et al., 2016]. This will sensibly reduce the cost of network infrastructures, decreasing the need for expensive dedicated hardware. For a very effective and accessible introduction to the main concepts, principles and features of network virtualization, we refer the reader to [Chowdhury and Boutaba, 2009] and [Schaffrath et al., 2009].

A central entity of network virtualization is represented by the Virtual Network (VN), which can be defined as a combination of network elements (network nodes and network links) realized over a Substrate Network (SN). Virtual nodes are inter-

connected through virtual links, giving raise to a virtual topology. A determinant advantage associated with virtualization of elements like nodes and links of a SN is that a (high) number of distinct virtual network topologies characterized by very different performance characteristics may be defined using the same physical hardware as basis. Furthermore, another major advantage is represented by the fact that the characteristics and performance of these virtual topologies can be very easily changed by network operators, flexibly allocating or de-allocating resources from the physical hardware, allowing to vary them in very fast ways, giving a high degree of flexibility to adapt dynamically to changing requirements of users and traffic conditions.

A further critical advantage that is commonly attributed to Network Virtualization is represented by its potential of strongly supporting the concept of Infrastructure as a Service (IaaS) (see e.g., [Bhardwaj et al., 2010]), which is considered a very desirable property for next generation of internet architectures, in which the current figure of the Internet Service Providers (ISPs) should be split into two separate figures: 1) an Infrastructure Provider (InfraP) that has the task of constructing and maintaining the network equipment and 2) a Stochastic Programming (SP) that has the task of offering and managing end-to-end services. We note that this separation between the figure that maintains the infrastructure and the figure that offers services is spreading in many different engineering areas (e.g., public transportation and energy systems). As pointed out in works such as those cited above, the advent of network virtualization brings toward the identification of three distinct principal players that substitute the unique traditional service provider: a Virtual Network Provider (VNP) that has the task of arranging the virtual resources from a number of InfraPs, a Virtual Network Operator (VNO) that makes the virtual networks available according to the requirements of the SP, and an SP that can instead focus just on the offer of tailored virtual networks to the customers of the service.

The optimization of virtual network placement essentially requires to decide how to map a number of virtual network requests, corresponding to requested virtual topologies, to the available substrate network, while satisfying the demanded network resources with the available substrate network resources. In a more formal way, we can essentially and generally describe the placement problem as follows: we are given a substrate network SN(N, L), in which N is the set of substrate

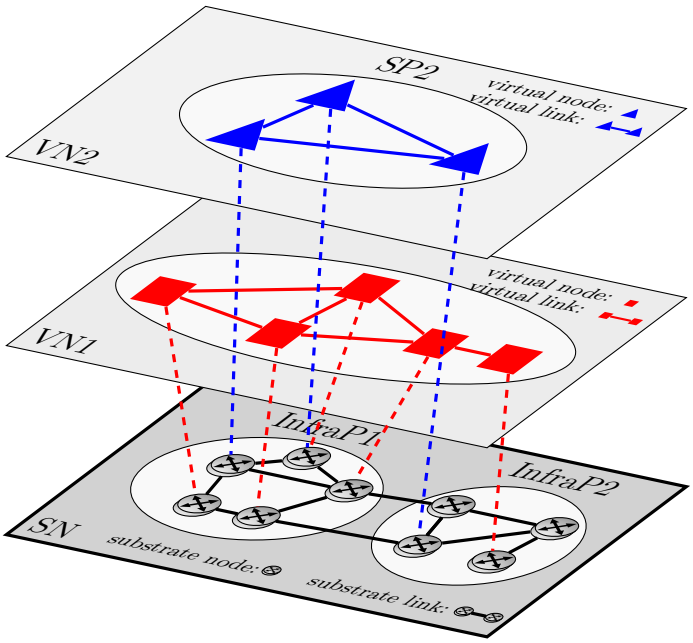


Figure 5.1.1 – Mapping of Virtual Networks to a Substrate Network

nodes and L is the set of substrate links, and a set of p virtual network requests $\mathrm{VNR}(N_k, L_k)$ with $k = 1, \ldots, p$, in which N_k is the set of substrate nodes and L_k is the set of substrate links of request k; additionally, let us define a vector space of resource vectors $\bar{R} = \prod_{\ell=1}^{m}$ defined for sets of distinct resources $R_1, R_2, \ldots, R_m$ and let CAP: $N \cup L \rightarrow \bar{R}$ be a function that associates an amount of each type of resource to each element of the substrate network. Also, let $\mathrm{DEM}_k : N_k \cup L_k \rightarrow \bar{R}$ be a function that identifies the amount of each type of resource demanded by each element of a virtual network request k. On this basis, the virtual network placement consists of defining two functions $F1_k : N_k \rightarrow N$ and $F2_k : L_k \rightarrow$ subSN $\subseteq$ SN, for every virtual network k, which must satisfy the resource demanded by the elements of each request k (i.e., it must satisfy DEM(n) $\leq$ CAP($F1_k(n)$) for every $n \in N_k$ and DEM(ℓ_k) $\leq$CAP(ℓ) for every $\ell_k \in L_k$ and $\ell \in F2_k(\ell_k)$). So this actually requires to define a mapping of virtual nodes to substrate nodes and of virtual links to paths in the substrate network, while taking into account the constraints of satisfying the demand of resources by the virtual elements without exceeding the capacity of the substrate elements. Such problem of virtual network placement is known to be $\mathcal{NP}$-Hard [Kolliopoulos and Stein, 1997]. Even remarkable subproblems of it, such as finding a virtual link mapping for a given node

mapping, are known to be $\mathcal{NP}$-Hard [Kolliopoulos and Stein, 1997].

Concerning the definition of a taxonomy of Network Virtualization problems in terms of their objectives, we can identify the following major classes of problems:

- maximization of the quality-of-service compliance of requests, in which the requests must be satisfied so that their features result as close as possible to the specifications fixed by the customers in terms of measures such as bandwidth, delay and CPU requirements (e.g., [Inführ and Raidl, 2011]);

- maximization of the profit of the infrastructure provider, in which the provider naturally attempts at obtaining the highest economical return and must carefully consider how to accept and manage user requests over a time horizon, in order to maintain sufficient spare resources for dealing with (more profitable) unexpected requests and future unknown requests (see e.g, [Chowdhury et al., 2012]);

- maximization of the survivability of the user requests, which requires to setup specific backup resources in case of possible failures of elements of the substrate network; similarly to the previous class of problems, also in this case the provider must carefully choose how to reserve backup resources: indeed, the reserved resources cannot be used to satisfy other requests and thus limit the possibility of accepting future requests and may lead to sensible reduction in profit if not carefully dimensioned (see e.g, [Shahriar et al., 2017, Li et al., 2020]);

- minimizing the total energy consumption, adopting a green network paradigm - this class of problems has emerged as one of the most important in computer networks in general, since the increase of such networks in size and complexity has lead to the consumption of huge amounts of energy. Such energy consumption not only represents a major cost for providers, but is also not sustainable from an environmental point of view (see e.g., [Bilal et al., 2014, Garroppo et al., 2020, Mohamed et al., 2021]);

As we started to discuss above, the problem of optimally designing virtual networks, allocating Virtual Network Function Component (VNFC)s to physical servers and managing the data flows between servers has received great attention

in recent times, in particular focusing on adopting a green networking perspective aiming at minimizing the overall power consumption (see e.g.,[Luizelli et al., 2015, Herrera and Botero, 2016, Mechtri et al., 2016, Marotta et al., 2017a, Baumgartner et al., 2018]). However, while purely heuristic solution approaches for virtual network design have been quite widely investigated, the development of hybrid exact-heuristic algorithms exploiting the potentialities of mathematical programming (so-called matheuristic) has received limited attention, as it may also found in a recent survey such as [Yang et al., 2021] and in other works that have focused on directly using state-of-the-art solvers to solve instances of small size or adopt simple ad-hoc heuristics (e.g., [Zhang et al., 2020]).

By our work presented here, we aim to start to fill this gap by proposing a new matheuristic for the green placement of Virtual Network Function in 5G, while taking into account the uncertainty of function requests which has been identified as a critical source of uncertainty (see e.g., [Marotta et al., 2017a, Baumgartner et al., 2018]).

The remainder of this chapter is organized as follows: in Section 5.2, we describe a Binary Linear Programming (BLP) model for modeling the green and robust placement of VNFCs; in Section 5.3, we present a new matheuristic to fast solve the placement problem: finally, in Section 5.4, we report preliminary computational results and derive some conclusions.

5.2 A Binary Linear Programming model for VNFC Placement

From a modeling point of view, we can essentially describe the topology of the 5G network that we consider through a graph $G(N, L)$, where N is the node set and L is the link set. Each link $\ell \in L$ corresponds to a pair (i, j), where $i, j \in N$ are the nodes it connects. Each link is associated with a bandwidth b_ℓ. The network interconnects a set of servers S and the node to which a server s is connected is denoted by $n(s) \in N$. Each server offers an amount of computational resources (e.g., CPU and RAM): denoting by R the set of resource types, the amount of resources available for each type $r \in R$ at a server $s \in S$ is denoted by a_{sr}. The set of VNFCs is denoted by V and the set of service chains offered in the network

is denoted by $\mathcal{C}$ (a service chain corresponds with a subset of VNFC that must be executed to provide a requested service to a user). When executed, a VNFC $v \in V$ requires an amount a_{vr} of each resource type $r \in R$. Each chain $C \in \mathcal{C}$ corresponds to a subset of pairs (v_1, v_2) belonging to $V \times V$. The exchange of data between v_1 and v_2 in a pair (v_1, v_2) requires an amount of bandwidth $b_{v_1}^{v_2}$ in each traversed network link. Concerning power consumption, every node $n \in N$ and link $\ell \in L$ consumes P_n and P_ℓ when used, respectively. Each server $s \in S$ has a consumption that is a linear function in the range $[P_s^{\min}, P_s^{\max}]$.

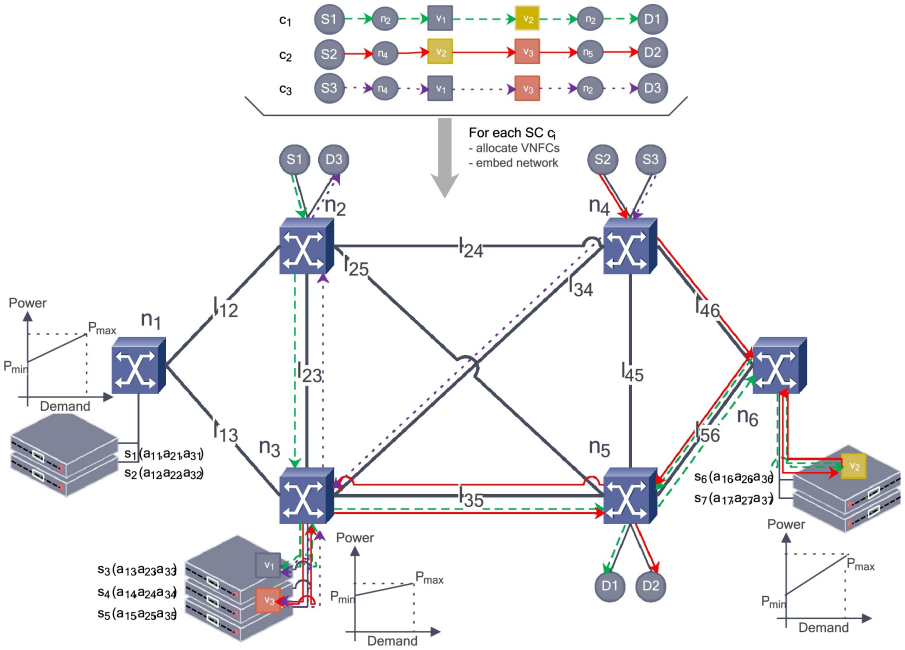


Figure 5.2.2 – an example of the VNF placement and mapping problem in [Marotta et al., 2017a]

Figure 5.2.2 illustrates the problem where we have in total seven servers ($s1$ until $s7$), each one with its own power profile (each server s has its own idle power $P_s^{\min}$ and maximum power consumption $P_s^{\max}$) and individual CPU, memory and disk capacities. In the example given in [Marotta et al., 2017a], server s_i has installed a_{1i} CPU, a_{2i} RAM and a_{3i} disk. Each server is connected to a unique router (for example, s_1 is connected to n_1). Each link has a dedicated capacity and latency (for example, the latency for the link between n_1 and n_2 is denoted as l_{12} - we omit bandwidth from Figure 5.2.2 to maintain readability). The servers, their capacities, together with the network nodes and links with their capacities form the

NFV infrastructure in terms of computing power, storage and network topology. In this example, we should map three service chains (denoted as $c1$, $c2$ and $c3$) into this NFV Infrastructure, each one with their own latency bounds. In total, we have three different VNFCs ($v1$, $v2$ and $v3$) and we assume that the traffic source for c_1 is the Sender S_1, which is connected to router n_2 and injects a certain volume of traffic into the service chain towards v_1. The VNFC v_1 processes the packets (for which it needs CPU, memory and disc) and forwards the processed traffic (which may have a different volume than the one injected) towards VNFC v_2, which again processes it and forwards a certain volume to the Destination D_1 that is connected to router n_2. Note that Figure 5.2.2 assumes additional source/sink nodes where traffic for a service chain is created/terminated, which are not explicitly mentioned in the model but they could be introduced by adding network nodes. The figure depicts an exemplary VNF placement and mapping into the physical Substrate Network. For example, the VNFC v_1 would be placed onto server s_3, v_3 onto server s_4 and so on. Servers hosting no VNFC would be powered down (e.g., s_1, s_2 or s_5) together with all the nodes not carrying any traffic (e.g., only n_1 in this case).

The optimization problem related to VNFC placement that we consider can be resumed as follows: given a 5G network interconnecting a set of servers, we want to decide how to establish a set of virtual chains in the network respecting the available resource budget of the servers and networks, while minimizing the overall power consumption.

In order to model the decision taken in the optimization problem, we introduce the following decision variables:

- variables $y_s \in \{0,1\}$, $\forall s \in S$ representing the activation of a server ($y_s = 1$ if s is turned on and 0 otherwise);
- variables $x_{vs} \in \{0,1\}$, $\forall v \in V, s \in S$ representing the allocation of a VNFC v to server s ($x_{vs} = 1$ if v is allocated to s and 0 otherwise);
- variables $z_n \in \{0,1\}$, $\forall n \in N$ representing the activation of a node n ($z_n = 1$ if n is turned on and 0 otherwise);
- variables $w_{ij} \in \{0,1\}$, $\forall (i,j) \in L$ representing the activation of a link $\ell = (i,j)$ ($g_{ij} = 1$ if $\ell = (i,j)$ is turned on and 0 otherwise);

- variables $f_{ij}^{(v_1,v_2)} \in \{0,1\}$, $\forall (i,j) \in L, (v_1,v_2) \in \bigcup_{C\in\mathcal{C}}$ representing that link (i,j) is used for data exchange between v_1 and v_2 belonging to some $C \in \mathcal{C}$.

These variables are employed in the following Binary Linear Programming, denoted by BLP-VP, modeling the VNFC optimal placement problem:

$$\min_{r=\text{CPU}} \sum_{s\in S}\left[P_s^{\min} y_s + \frac{P_s^{\max}-P_s^{\min}}{a_{rs}} \sum_{v\in V} a_{vr}x_{vs}\right] + \sum_{n\in N} P_n z_n + \sum_{(i,j)\in L} P_{ij} w_{ij} \quad (5.1)$$

$$\text{s.t.} \quad \sum_{s\in S} x_{vs} = 1 \qquad \forall v \in V \quad (5.2)$$

$$y_s \leq \sum_{v\in V} x_{vs} \qquad \forall s \in S \quad (5.3)$$

$$x_{vs} \leq y_s \qquad \forall s \in S, \forall v \in V \quad (5.4)$$

$$\sum_{v\in V} a_{vr}x_{vs} \leq a_{rs}y_s \qquad \forall s \in S, \forall r \in R \quad (5.5)$$

$$\sum_{(n,i)\in L} b_{v_1}^{v_2} f_{ni}^{v_1,v_2} - \sum_{(i,n)\in L} b_{v_1}^{v_2} f_{in}^{v_1,v_2} = \sum_{s\in S:n(s)=n} b_{v_1}^{v_2}(x_{v_1 s} - x_{v_2 s})$$

$$\forall n \in N, \forall (v_1,v_2) \in \bigcup_{C\in\{\mathcal{C}\}} C \quad (5.6)$$

$$\sum_{(v_1,v_2)\in\bigcup_{C\in\{\mathcal{C}\}} C} b_{v_1}^{v_2} f_{ij}^{v_1,v_2} \leq B_{ij} w_{ij} \qquad \forall (i,j) \in L \quad (5.7)$$

$$w_{ij} \leq z_i \qquad \forall (i,j) \in L \quad (5.8)$$

$$w_{ij} \leq z_j \qquad \forall (i,j) \in L \quad (5.9)$$

$$f_{ij}^{v_1,v_2} \leq z_i \qquad \forall (i,j) \in L \quad (5.10)$$

$$f_{ij}^{v_1,v_2} \leq z_j \qquad \forall (i,j) \in L \quad (5.11)$$

$$y_s \in \{0,1\} \qquad \forall s \in S \quad (5.12)$$

$$x_{vs} \in \{0,1\} \qquad \forall s \in S, \forall v \in V \quad (5.13)$$

$$z_n \in \{0,1\} \qquad \forall n \in N \qquad (5.14)$$

$$w_{ij} \in \{0,1\} \qquad \forall (i,j) \in L \qquad (5.15)$$

$$f_{ij}^{v_1,v_2} \in \{0,1\} \qquad \forall (i,j) \in L, \forall (v_1,v_2) \in \bigcup_{C \in \{\mathcal{C}\}} C \qquad (5.16)$$

The objective function (5.1) pursues the minimization of the total power consumption that is expressed as the sum of 1) a fixed power component, which represents the power consumed by activated servers, 2) a variable power component, which takes into account the amount of resources of servers that are used by the VNFCs assigned to the server, 3) the power consumed by the nodes of the substrate network, 4) the power consumed by the arcs of the substrate network.

Concerning the feasibility constraints, (5.2) impose that each VNFC must be allocated on exactly one server. Furthermore, constraints (5.3) are adopted to logically link the server activation and VNFC allocation decision variables: the activation of a server imposes that at least one VNFC is assigned to it. The constraints (5.4) are instead logical constraints imposing that, if the server is not activated, then no binary variable allocating a VNFC to the server can be activated. The constraints (5.5) model the resource capacity for each server and resource type, imposing that the summation of the amount of a specific resource consumed by VNFCs assigned to a server must not exceed the capacity of the server for that kind of resource. The constraints (5.6) are flow conservation constraints that regulate how links are used for transferring data between VNFCs, depending on which server the VNFCs are allocated to: the value of the right-hand-side depends upon the value of the difference of variables reflecting to which server the two involved VNFC of a chain are allocated. The set of constraints (5.7) is needed to represent the bandwidth capacity of each link, by imposing that it cannot be exceeded by the summation of bandwidth consumed by communications between VNFCs that are connected by means of link (i,j). The constraints (5.8) and (5.9) impose that using a link (i,j) requires to activate the two end nodes of the link. A similar role is played by the constraints (5.10) and (5.11), which impose to activate the end nodes of a link when traffic is sent over the link for supporting the communication between two VNFCs. Finally the constraints (5.12)-(5.16) define the feasible

domain of all the decision variables involved in the problem.

5.2.1 Guaranteeing protection against resource uncertainty

Traffic routed through telecommunication networks is normally subject to uncertainty since the behavior of the users of the network is typically not exactly known in advance. Concerning this point, we refer the reader to [Bauschert et al., 2014] for an accessible discussion about the motivations behind traffic uncertainty in telecommunications networks. Concerning the design of virtual network functions, the major uncertainty question is represented by the resources that are necessary to the functions generated by users and we reflect this in the robust optimization model that is derived here. Specifically, referring to the notation that we have introduced, we assume that the coefficients a_{vr} representing the amount of resource r requested by a VNFC v is subject to uncertainty and its exact value is not known when the design problem is solved. In order to tackle such data uncertainty, we rely on the well-known Γ-Robustness model that was initially proposed by [Bertsimas and Sim, 2004]. Since this is an interval uncertainty model, we must introduce a reference value $\bar{a}_{vr}$ (nominal value) for the unknown value and a value Δa_{vr} representing the highest deviation that may occur from the nominal value. As a consequence, we assume that the actual value of the uncertain coefficient a_{vr} lies in the symmetric interval:

$$a_{vr} \in [\bar{a}_{vr} - \Delta a_{vr}, \bar{a}_{vr} + \Delta a_{vr}] \tag{5.17}$$

Once this deviation interval is defined, we proceed to show how it may be used as basis to derive robust counterpart of the uncertain resource constraints. We now proceed to focus attention on the constraints that we need to robustify, namely:

$$\sum_{v \in V} \bar{a}_{vr} x_{vs} \leq a_{rs} y_s \tag{5.18}$$

that are the capacity constraints fixing the capacity of a server $s \in S$ for each type of resource $r \in R$ and containing the uncertain resource coefficients a_{vr}. We note that in these constraints we have included the nominal values $\bar{a}_{vr}$.

As first step to derive the robust counterpart of these constraints, we can rewrite the constraints adding one term $\text{DEV}_{rs}(\Gamma, x)$ that indicates the worst deviation in value that the coefficients may attain for a given solution x and for Γ coefficients

allowed to differ from their nominal values:

$$\sum_{v \in V} \bar{a}_{vr} x_{vs} + DEV_{rs}(\Gamma, x) \leq a_{rs} y_s \qquad \forall s \in S, \forall r \in R \tag{5.19}$$

The worst deviation value $\mathrm{DEV}_{rs}(\Gamma, x)$ can be computed as the optimal value of the following Combinatorial optimization problem in which we remark that the worst deviation of coefficient is represented by an increase in the resource that is requested by a VNFC v, since it tend to lead to a violation of the installed capacity:

$$\max \quad \sum_{v \in V} (\Delta a_{vr} x_{vs})\, \phi_{rsv} \tag{5.20}$$

$$\text{s.t.} \quad \sum_{v \in V} \phi_{rsv} \leq \Gamma \tag{5.21}$$

$$\phi_{rsv} \in \{0, 1\} \qquad v \in V \tag{5.22}$$

In the problem above, binary variables ϕ_{rsv} are introduced to represent whether a resource coefficient a_{vr} is subject to its worst deviation Δa_{vr} and the unique cardinality constraint imposes that at most Γ coefficients may deviate. This value Γ may range from 0 (no protection against deviation and lowest price of robustness) to $|V|$ (i.e., full protection against all VNFC deviating and highest price of robustness. By highlighting the fact that $DEV_{rs}(\Gamma, x)$ corresponds with an optimization problem, it becomes evident that (5.19) is actually non-linear. However, as elegantly proved by Bertsimas and Sim, it is possible to rely on duality theory to produce a linear robust constraint as follows. In primary, we may note that for given x, the value $DEV_{rs}(\Gamma, x)$ equals the optimal value of its linear relaxation, namely:

$$\max \quad \sum_{v \in V} (\Delta a_{vr} x_{vs})\, \phi_{rsv} \tag{5.23}$$

$$\text{s.t.} \quad [\delta_{rs} \geq 0] \qquad \sum_{v \in V} \phi_{rsv} \leq \Gamma \tag{5.24}$$

$$[\eta_{rsv} \geq 0] \qquad 0 \leq \phi_{rsv} \leq 1 \qquad v \in V \tag{5.25}$$

The dual variables δ_{rs}, $\eta_{rsv} \forall v \in V$ are introduced for the corresponding constraints (5.24) and (5.25), respectively. The dual of $DEV_{rs}(\Gamma, x)$ is then formulated

as follows:

$$\min \quad \Gamma\delta_{rs} + \sum_{v\in V} \eta_{rsv} \tag{5.26}$$

$$\text{s.t.} \quad \delta_{rs} + \eta_{rsv} \geq \Delta a_{vr} x_{vs} \qquad v \in V \tag{5.27}$$

$$\delta_{rs} \geq 0 \tag{5.28}$$

$$\eta_{rsv} \geq 0 \tag{5.29}$$

Noticing then that $\text{DEV}_{rs}(\Gamma, x)$-primal is a feasible and bounded problem, strong duality can be exploited and states that $\text{DEV}_{rs}(\Gamma, x)$-dual is also feasible and bounded and the optimal values of the two coincide problems. Following the Bertsimas and Sim procedure, the non-linear constraint (5.19) with the following robust version and with the related dual variables, getting the following compact model:

$$\sum_{v\in V} \bar{a}_{vr} x_{vs} + \left(\Gamma\delta_{rs} + \sum_{v\in V} \eta_{rsv} \right) \leq a_{rs} y_s \quad \forall s \in S, \forall r \in R \tag{5.30}$$

$$\delta_{rs} + \eta_{rsv} \geq \Delta a_{vr} x_{vs} \qquad \forall s \in S, \forall r \in R, \forall v \in V \tag{5.31}$$

$$\delta_{rs} \geq 0 \qquad \forall s \in S, \forall r \in R \tag{5.32}$$

$$\eta_{rsv} \geq 0 \qquad \forall s \in S, \forall r \in R, \forall v \in V \tag{5.33}$$

Similarly to uncertain a_{vr} in the objective function, we introduce an artificial decision variable ξ_s such that $[(P_s^{\max} - P_s^{\min})/a_{rs}] \sum_{v\in V} a_{vr} x_{vs} \leq \xi_s$, which indicates the power consumption for charging the different resource at each server s. The complete model that we consider is thus the following, which we denote by ROB-BLP-VP:

$$\min_{r=\text{CPU}} \quad \sum_{s\in S} \left[P_s^{\min} y_s + \xi_s \right] + \sum_{n\in N} P_n z_n + \sum_{(i,j)\in L} P_{ij} w_{ij} \tag{5.34}$$

$$\text{s.t.} \quad \frac{P_s^{\max} - P_s^{\min}}{a_{rs}} \left[\Gamma\delta_{rs} + \sum_{v\in V} (\bar{a}_{vr} x_{vs} \eta_{rsv}) \right] \leq \xi_s \qquad \forall s \in S \tag{5.35}$$

$$\sum_{s\in S} x_{vs} = 1 \qquad \forall v \in V \tag{5.36}$$

$$\Gamma\delta_{rs} + \sum_{v\in V}(\bar{a}_{vr}x_{vs} + \eta_{rsv}) \leq a_{rs}y_s \qquad \forall s \in S, \forall r \in R \qquad (5.37)$$

$$x_{vs} \leq y_s \qquad \forall s \in S, \forall v \in V \qquad (5.38)$$

$$\sum_{v\in V} a_{vr}x_{vs} \leq a_{rs}y_s \qquad \forall s \in S, \forall r \in R \qquad (5.39)$$

$$\sum_{(n,i)\in L} b_{v_1}^{v_2} f_{ni}^{v_1,v_2} - \sum_{(i,n)\in L} b_{v_1}^{v_2} f_{in}^{v_1,v_2} = \sum_{s\in S:n(s)=n} b_{v_1}^{v_2}(x_{v_1 s} - x_{v_2 s})$$

$$\forall n \in N, \forall (v_1, v_2) \in \bigcup_{C\in\{\mathcal{C}\}} C \qquad (5.40)$$

$$\sum_{(v_1,v_2)\in\bigcup_{C\in\{\mathcal{C}\}} C} b_{v_1}^{v_2} f_{ij}^{v_1,v_2} \leq B_{ij}w_{ij} \qquad \forall (i,j) \in L \qquad (5.41)$$

$$w_{ij} \leq z_i \qquad \forall (i,j) \in L \qquad (5.42)$$

$$w_{ij} \leq z_j \qquad \forall (i,j) \in L \qquad (5.43)$$

$$f_{ij}^{v_1,v_2} \leq z_i \qquad \forall (i,j) \in L \qquad (5.44)$$

$$f_{ij}^{v_1,v_2} \leq z_j \qquad \forall (i,j) \in L \qquad (5.45)$$

$$\delta_{rs} + \eta_{rsv} \geq \Delta a_{vr}x_{vs} \qquad \forall s \in S, \forall r \in R, \forall v \in V \qquad (5.46)$$

$$\delta_{rs} \geq 0 \qquad \forall s \in S, \forall r \in R \qquad (5.47)$$

$$\eta_{rsv} \geq 0 \qquad \forall s \in S, \forall r \in R, \forall v \in V \qquad (5.48)$$

$$y_s \in \{0,1\} \qquad \forall s \in S \qquad (5.49)$$

$$x_{vs} \in \{0,1\} \qquad \forall s \in S, \forall v \in V \qquad (5.50)$$

$$z_n \in \{0,1\} \qquad \forall n \in N \qquad (5.51)$$

$$w_{ij} \in \{0,1\} \qquad \forall (i,j) \in L \tag{5.52}$$

$$f_{ij}^{v_1,v_2} \in \{0,1\} \qquad \forall (i,j) \in L, \forall (v_1,v_2) \in \bigcup_{C \in \{\mathcal{C}\}} C \tag{5.53}$$

5.3 A New Matheuristic for Green Robust Virtual Network Function Placement

We present here a new matheuristic for optimal VNFC placement that is based on the integration of a Genetic Algorithm (GA) with an *exact* large neighborhood search, namely a search formulated as an optimization problem solved by a state-of-the-art solver such as CPLEX [IBM-ILOG-CPLEX]. The solver is also used for completing partial solutions of (ROB-BLP-VP) in an optimal way: for a fixed value configuration of a subset of decision variables, we employ the solver to find a feasible valorization of all the remaining variables while optimizing the objective function. At the basis of this matheuristic there is the consideration that, while a state-of-the-art solver may find difficulties in identifying good quality solutions for ROB-BLP-VP, it is instead able to efficiently identify good quality solutions for appropriate subproblems of ROB-BLP-VP, obtained by fixing the value of a consistent number of decision variables.

GAs are widely known meta-heuristics that draw inspiration from the evolution of a population (see [Goldberg and Holland, 1988] for an exhaustive introduction to the topic). The individuals of the population represent solutions of the optimization problem and the *chromosome* of an individual corresponds to a valorization of decision variables of a solution. The quality of an individual/solution is assessed through a *fitness function*. The GA begins with the definition of an initial population that then changes through evolutionary mechanisms like crossover and mutation of individuals, until some stopping criterion is met.

The general structure of the GA that we take as reference and adapt that is presented in Algorithm 6. We now proceed to detail how the elements and the phases that have been presented above for the generic hybrid genetic algorithm are adapted to be applied to the problem ROB-BLP-VP.

Algorithm 6: General structure of the hybrid genetic algorithm

```
Creation of the initial population;
while the arrest condition is not satisfied do
    Selection of individuals who generate the offspring;
    Generation of the offspring by crossover;
end
Exact improvement search;
```

5.3.1 Initialization of the population

Solution representation. Dealing with a genetic algorithm, the first step consists of establishing what the individuals constituting the population represent. We decided that the chromosome of an individual corresponds with a valorization of the decision variables (y, x) (of ROB-BLP-VP): these variables are those used to represent whether a server is activated and whether a VNFC is allocated to a specific server. These two decisions are particularly critical for the problem and once their values have been fixed, we obtain an easier subproblem of (ROB-BLP-VP) . Specifically, once the value of the variables (y, x) is fixed, (ROB-BLP-VP) reduces to a kind of robust network flow problem and is easier to be solved by a state-of-the-art optimization solver and it is much faster to identify an optimal solution for the restricted (ROB-BLP-VP) problem with (y, x) fixed.

Fitness function. In order to assess the quality of an individual, a natural choice consists of adopting the objective function (5.1) of (ROB-BLP-VP) as fitness function. In this way, we can establish a very simple correspondence between the genetic algorithmic interpretation and the optimization model and it is immediate to evaluate how good is an individual.

Initial population. The strategy that we explored to generate the initial group of individuals relies on the following principles: to generate an individual, we randomly activate a number $\sigma < |S|$ of servers and then we randomly assign each VNFC in V to one single activated server, checking that the resource constraints (5.5) are not violated. In this way, we obtain a valorization $(\bar{y}, \bar{x})$ of the server and allocation variables that we can then complete by solving the remaining subproblem of (ROB-BLP-VP) through a state-of-the-art solver. By this strategy, we can

obtain the optimal solution of (ROB-BLP-VP) for a fixed $(\bar{y}, \bar{x})$. We denote the set of individuals constituting the population at a generic iteration of the algorithm by POP.

5.3.2 Evolution of the population

Selection. The individuals chosen for being combined and generating the new individuals are chosen according to a *tournament selection* principle: we first create a number β of (small cardinality) groups of individuals by randomly selecting them from POP. Then the γ individuals in each group associated with the best fitness value are combined through crossover.

Crossover. We form the couples that generate the offspring according to the following procedure. From the tournament selection, we obtain $\beta\gamma$ individuals that are randomly paired in couples, each generating one offspring. Assuming that the two parents are associated with chromosomes/partial solutions (y^1, x^1) and (y^2, x^2), the chromosome of the offspring $(y^{\text{off}}, x^{\text{off}})$ is defined according to two rules:

1. if the parents have the same binary value in a position j, then the offspring inherits such value in its position j (i.e., if $(y^1, x^1)_j = (y^2, x^2)_j$ then $(y^{\text{off}}, x^{\text{off}})_j = (y^1, x^1)_j$);

2. if the parents have distinct binary values in a position j, then the offspring inherits a null value (i.e., if $(y^1, x^1)_j \neq (y^2, x^2)_j$ then $(y^{\text{off}}, x^{\text{off}})_j = 0$).

Possible violations in the constraints (5.2) and (5.5) associated with $(y^{\text{off}}, x^{\text{off}})$ are then repaired. The main rationale at the basis of this procedure is assuming that two solutions having the same valorization of a variable is a good indication that such valorization should be maintained also in the offspring.

5.3.3 Exact Improvement Search

We attempt at improving the best solution found by the GA through an *exact* large neighborhood search, namely a search that is formulated as a suitable Binary Linear Programming problem solved by a state-of-the-art optimization solver

[Blum et al., 2011]. The search is based on using the effective heuristic Relaxation Induced Neighborhood Search (RINS) (we refer the reader to [Danna et al., 2005] for an exhaustive description of it). Specifically, given a partial solution $(\bar{y}, \bar{x})$ of (ROB-BLP-VP) and $(y^{\text{TLR}}, x^{\text{TLR}})$ an optimal solution of a tight linear relaxation (specifically, the optimal solution obtained by removing the integrality requirements on the binary variables and considering the basic model strengthened by the cuts identified by the state-of-the-art solver), we solve a subproblem of (ROB-BLP-VP) where the value of the j-th component of the vectors (y, x) is fixed according to the following two rules:

1. If $(\bar{y}, \bar{x})_j = 0 \wedge (y^{\text{TLR}}, x^{\text{TLR}}) \leq \epsilon$, then $(y, x)_j = 0$;

2. If $(\bar{y}, \bar{x})_j = 1 \wedge (y^{\text{TLR}}, x^{\text{TLR}}) \geq 1 - \epsilon$, then $(y, x)_j = 1$.

The subproblem of (ROB-BLP-VP) subject to such variable fixing is then solved by the state-of-the-art solver, running with a time limit.

The overall pseudo-code of the hybrid genetic algorithm presented above is provided in Algorithm 7.

Algorithm 7: Adaption of the general GA algorithm

```
Generate a number n of individuals (each individual is defined by
  randomly activating σ < |S| servers and then randomly assign each
  VNFC in V to one single activated server, while checking that the
  capacity constraints are satisfied). These individuals constitute the
  starting population POP;
while a time limit is not reached do
    Define β groups of individuals by random selection from the
      population POP;
    For each group, extract the γ individuals with the best fitness;
    Randomly pair the βγ selected individuals and operate the crossover
      on them (if the parents present the same value in a position, then the
      offspring inherits such value, otherwise it inherits a null value);
    Include the new individuals in the population POP;
end
Run the exact search RINS;
```

5.4 Computational results

We assessed the performance of the proposed matheuristic by considering 20 instances that refer to a network made up of 10 nodes to which 50 servers are connected and that are defined for different VNFC features, defined referring to the works [Marotta et al., 2017a,b]. To execute the tests, we employed a Windows machine with 2.70 GHz professor and 8 GB of RAM. As optimization solver, we relied on IBM ILOG CPLEX 12.5, which is interfaced through Concert Technology with a C/C++ code. The global time limit imposed to CPLEX to solve (ROB-BLP-VP) is set to 3600 seconds. The same time limit is set for the matheuristic (denoted here by MatHeu), assigning 3000 seconds to the GA phase and 600 to the improvement phase based on RINS (in which we set $\epsilon = 0.1$). The initial population includes 100 individuals/solutions and, at each iteration, we consider $\beta = 10$ groups from each of which $\gamma = 2$ individuals are chosen.

The results of the computational tests are presented in Table 1, where: ID identifies the instance; T^* (CPLEX) and T^* (MatHeu) are the time (in seconds) that CPLEX and MatHeu needs to find the best solution within the time limit, respectively, whereas $\Delta T^*\%$ is the percentage reduction in time that MatHeu grants to find a solution that is at least as good as the best solution found by CPLEX. Finally, $\Delta P^*\%$ is the reduction in power consumption that the best solution found by MatHeu grants with respect to the best solution found by MatHeu within the time limit.

We have paid particular attention to the computational time aspect, since, according to discussions that we had with professionals of the sector, identifying high quality solutions within limited amount of time is considered a particularly important objective when establishing sets of virtual networks in a business context.

Concerning the difficulty of solving (ROB-BLP-VP), as highlighted in several works such as [Luizelli et al., 2015, Marotta et al., 2017a], even simplified deterministic versions of (ROB-BLP-VP) may prove difficult to solve for state-of-the-art optimization solvers also in the case of instances. We confirm such behavior in the case of our instances, which highlights the need for fast (heuristic) solution algorithms. On the basis of the results, we can say that $MatHeu$, for all the instances, is able to return a solution that is at least as good as the best solution found by CPLEX within the time limit in 24% less time, on average. Concerning

Table 5.4.1 – Computational results

ID	T^* (CPLEX)	T^* (MatHeu)	$\Delta T^*\%$	$\Delta P^*\%$
I1	3322	2580	22.3	5.4
I2	3194	2742	14.1	6.8
I3	3157	2335	26.0	6.2
I4	3552	2905	18.2	10.2
I5	3513	2536	27.8	6.9
I6	3402	2892	14.9	5.8
I7	3475	2642	23.9	8.6
I8	3362	3041	9.5	9.3
I9	3595	2587	28.0	7.6
I10	3488	2769	20.6	5.5
I11	3302	2281	30.9	19.4
I12	3260	2105	35.4	24.8
I13	3512	2447	30.3	13.5
I14	3396	2572	24.2	21.7
I15	3471	2076	40.1	19.4
I16	3395	2405	29.1	14.2
I17	3185	2953	7.2	22.1
I18	3338	2633	21.1	16.9
I19	3198	2194	31.3	21.6
I20	3056	2310	24.4	16.8

the reduction in consumed power, we can instead notice that *MatHeu* allows to find better quality solution than CPLEX within the time limit, with a reduction in power consumption that can reach 24% and on average is equal to about 13%. The better performance of the matheuristic results particularly evident for the second half of instances.

These results, which have been presented in our publication [Bauschert et al., 2019], have resulted remarkable and, as future work, have encouraged us to attempt at refining the solution construction mechanism, better exploiting the specific features of the mathematical model of (ROB-BLP-VP) to define the rules adopted to generate the initial population and to generate the offspring solutions by crossover.

As an alternative to the exact search based on RINS, we have also evaluated the possibility of adopting another exact neighborhood search based on using hamming distance constraints. Specifically, given a feasible solution associated with a fixing

$(\bar{y}, \bar{x})$ of the binary variables x and y, we considered the neighborhood of feasible solutions that can be obtained by modifying at most a number D of valorization of binary variables. Such neighborhood can be formally defined by adding the following constraint:

$$\sum_{j:(\bar{y},\bar{x})_j=0} (y,x)_j + \sum_{j:(\bar{y},\bar{x})_j=1} (1-(y,x)_j) \leq D,$$

which counts the number of binary variables that have switched their value from 0 to 1 and from 0 to 1, imposing that such number must not exceed the value D. Similarly to the RINS approach, the resulting neighborhood is explored by means of a state-of-the-art solver.

Table 5.4.2 – Comparison of alternative exact search - power reduction

ID	P^* (MatHeu-RINS)	P^* (MatHeu-HD)
I1	5.4	2.1
I2	6.8	2.6
I3	6.2	3.8
I4	10.2	6.3
I5	6.9	8.5
I6	5.8	3.9
I7	8.6	10.2
I8	9.3	3.3
I9	7.6	4.8
I10	5.5	1.9
I11	19.4	13
I12	24.8	11.5
I13	13.5	16.3
I14	21.7	8.4
I15	19.4	23.2
I16	14.2	5.9
I17	22.1	13.2
I18	16.9	10.7
I19	21.6	11.6
I20	16.8	9.9

In Table 5.4.2, we report comparisons of the performance granted by the two alternative exact searches, in terms of power reduction that the best solution found within the time limit by each search is able to grant with respect to the best

solution found by CPLEX within the time limit. Specifically, for each instance, we report the percentage power consumption reduction P^* (MatHeu-RINS) granted by the GA with RINS with respect to CPLEX, whereas P^* (MatHeu-HD) reports the percentage power consumption reduction P^* (MatHeu-HD) granted by the GA with the hamming distance constraint set to $D = 2$ with respect to CPLEX.

Evaluating the results, the adoption of RINS guarantees a higher reduction in power for all but four instances, indicating that neighborhood search defined with $D = 2$, while reducing the consumption with respect to the best solution found by CPLEX for instances, results not competitive with respect to RINS. However, the four cases in which the hamming distance-based constraint neighborhood performs better may suggest that tuning the value D and refining the definition of the neighborhood, involving a different set of variables, could possibly allow to define an alternative effective exact search. We consider this a possible subject of future research.

6. Conclusions and Future Work

Infrastructures based on networks nowadays constitute a fundamental component of our everyday life and continue to grow in size and complexity. Designing and managing the networks at their basis have become a very complex task and the adoption of mathematical optimization approaches has clearly shown to grant big advantages for identifying high quality design and management solutions. This Ph.D. Thesis has been focused on proposing new optimization modeling and algorithmic approaches for dealing with real-world network optimization problems arising in the transportation and telecommunications field. Since the focus has been on real-world applications, a relevant aspect that we have taken into account has been represented by data uncertainty, i.e. the fact that the value of a subset of input data of the problem is not exactly known when the problem is solved. In order to deal with such data uncertainty, we have also investigated the development of new modeling and algorithmic robust optimization approaches, which aim at identifying solutions that maintain their feasibility and optimality even when input data are subject to deviations in value.

More precisely, in the context of transportation problems, we have considered the flight level assignment problem, which arises in air traffic management and consists of establishing the flight levels of a set of aircraft in order to improve the total avenue from assignment, reduce the total number of potential en-route Air Traffic Flow Management (ATFM) conflicts and also the total ATFM delay. In this context, we proposed a new chance-constrained optimization problem and iterative solution heuristic which is based on both analytical and sampling methods. Besides transportation problems, this Thesis has also focused on the optimal design of 5^{th} generation of wireless networks considering Superfluid and virtual architectures. Specifically, the 5G Superfluid architecture is based on atomic virtual entities called Reusable Functional Block (RFB)s and we investigated the prob-

lem of minimizing the total installation costs of a Superfluid network composed of virtual entities and realized over a physical network, while guaranteeing constraint on user coverage, downlink traffic performance and technical constraints on RFBs of different nature. To solve this hard problem, we proposed a Benders decomposition approach. Concerning instead the design of general virtual networks, we adopted a green paradigm that pursues energy-efficiency and tackled a state-of-the-art robust mixed integer linear programming formulation of the problem, by means of a new matheuristic based on combining a genetic algorithm with exact large neighborhood searches.

Results of computational tests executed considering realistic problem instances have shown the validity of all the new optimization modeling and algorithmic approaches proposed in this Thesis for the transportation and telecommunications problems sketched above.

As ongoing work, we are studying the generation of the application used to solve the CCP FLA problem. Also, we are extending the computational tests to larger set of instances and attempting at strengthening the performance of the algorithm, investigating tuning strategies for setting the parameters at the basis of the various solution approaches. The aim is to include such new results in the journal versions of the conference papers that have been published during the Ph.D. Furthermore, as future work, we intend to investigate the identification of other class of valid inequalities in the context of 5G SF networks, better catching resource interactions between RFBs of different nature, with the aim of improving the convergence of our Benders decomposition solution approach. Moreover, concerning the matheuristic, we believe that a significant improvement in performance could be obtained by better integrating tight formulations in the solution process: stronger formulations could be used as basis for defining an initial population characterized by individuals with higher fitness, exploiting the valuable information coming from stronger linear relaxations, thus providing stronger individuals from the first iterations.

Bibliography

Antonio Abad and John-Paul Barrington Clarke. Using tactical flight level allocation to alleviate airspace corridor congestion. In *AIAA 4th aviation technology, integration and operations (ATIO) forum*, page 6456, 2004.

Sherif Abdelwahab, Bechir Hamdaoui, Mohsen Guizani, and Taieb Znati. Network function virtualization in 5g. *IEEE Communications Magazine*, 54(4):84–91, 2016.

A Agustín, Antonio Alonso-Ayuso, Laureano F Escudero, Celeste Pizarro, et al. On air traffic flow management with rerouting. part i: Deterministic case. *European Journal of Operational Research*, 219(1):156–166, 2012a.

A Agustín, Antonio Alonso-Ayuso, Laureano F Escudero, Celeste Pizarro, et al. On air traffic flow management with rerouting. part ii: Stochastic case. *European Journal of Operational Research*, 219(1):167–177, 2012b.

Ravindra Ahuja, Thomas Magnanti, and James Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, USA, 1993. ISBN 1611973422.

AirFrance. Airfrance operating result – full year 2019. https://www.airfranceklm.com/sites/default/files/q4_2019_press_release_en_8855736.pdf, 2019.

N Alguacil and AJ Conejo. Multiperiod optimal power flow using benders decomposition. *IEEE Transactions on power systems*, 15(1):196–201, 2000.

Cyril Allignol, Nicolas Barnier, and Alexandre Gondran. Optimized flight level allocation at the continental scale. 2012.

Nicolas Barnier and Cyril Allignol. Combining flight level allocation with ground holding to optimize 4d-deconfliction. 2011.

Nicolas Barnier and Pascal Brisset. Graph coloring for air traffic flow management. *Annals of operations research*, 130(1-4):163–178, 2004.

Andreas Baumgartner, Thomas Bauschert, Fabio D'Andreagiovanni, and Varun S Reddy. Towards robust network slice design under correlated demand uncertainties. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.

Thomas Bauschert, Christina Büsing, Fabio D'Andreagiovanni, Arie MCA Koster, Manuel Kutschka, and Uwe Steglich. Network planning under demand uncertainty with robust optimization. *IEEE Communications Magazine*, 52(2):178–185, 2014.

Thomas Bauschert, Fabio D'andreagiovanni, Andreas Kassler, and Chenghao Wang. A matheuristic for green and robust 5g virtual network function placement. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pages 430–438. Springer, 2019.

Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical programming*, 88(3): 411–424, 2000.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton university press, 2009.

Jacques F Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252, 1962.

Dimitris Bertsimas and Sarah Stock Patterson. The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach. *Transportation Science*, 34(3):239–255, 2000.

Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

Dimitris Bertsimas, Guglielmo Lulli, and Amedeo Odoni. The air traffic flow management problem: An integer optimization approach. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 34–46. Springer, 2008.

Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011a.

Dimitris Bertsimas, Guglielmo Lulli, and Amedeo Odoni. An integer optimization approach to large-scale air traffic flow management. *Operations research*, 59(1): 211–227, 2011b.

Sushil Bhardwaj, Leena Jain, and Sandeep Jain. Cloud computing: A study of infrastructure as a service (iaas). *International Journal of engineering and information Technology*, 2(1):60–63, 2010.

Giuseppe Bianchi, Erez Biton, Nicola Blefari-Melazzi, Isabel Borges, Luca Chiaraviglio, Pedro de la Cruz Ramos, Philip Eardley, Francisco Fontes, Michael J McGrath, Lionel Natarianni, et al. Superfluidity: a flexible functional architecture for 5g networks. *Transactions on emerging telecommunications technologies*, 27(9):1178–1186, 2016.

Kashif Bilal, Saif Ur Rehman Malik, Osman Khalid, Abdul Hameed, Enrique Alvarez, Vidura Wijaysekara, Rizwana Irfan, Sarjan Shrestha, Debjyoti Dwivedy, Mazhar Ali, Usman Shahid Khan, Assad Abbas, Nauman Jalil, and Samee U. Khan. A taxonomy and survey on green data center networks. *Future Generation Computer Systems*, 36:189–208, 2014.

Silvio Binato, Mário Veiga F Pereira, and Sérgio Granville. A new benders decomposition approach to solve power transmission network design problems. *IEEE Transactions on Power Systems*, 16(2):235–240, 2001.

Christian Blum, Jakob Puchinger, Günther R Raidl, and Andrea Roli. Hybrid metaheuristics in combinatorial optimization: A survey. *Applied soft computing*, 11(6):4135–4151, 2011.

Jin Xin Cao, Der-Horng Lee, Jiang Hang Chen, and Qixin Shi. The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods. *Transportation Research Part E: Logistics and Transportation Review*, 46(3):344–353, 2010.

Abraham Charnes and William W Cooper. Chance-constrained programming. *Management science*, 6(1):73–79, 1959.

Jing Chen, Long Chen, and D Sun. Air traffic flow management under uncertainty using chance-constrained optimization. *Transportation Research Part B: Methodological*, 102:124–141, 2017.

Luca Chiaraviglio, Lavinia Amorosi, Stefania Cartolano, Nicola Blefari-Melazzi, Paolo Dell'Olmo, Mohammad Shojafar, and Stefano Salsano. Optimal superfluid management of 5g networks. In *2017 IEEE Conference on Network Softwarization (NetSoft)*, pages 1–9. IEEE, 2017.

Luca Chiaraviglio, Fabio D'Andreagiovanni, Giulio Sidoretti, Nicola Blefari-Melazzi, and Stefano Salsano. Optimal design of 5g superfluid networks: Problem formulation and solutions. In *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 1–8. IEEE, 2018.

Luca Chiaraviglio, Fabio D'Andreagiovanni, Simone Rossetti, Giulio Sidoretti, Nicola Blefari-Melazzi, Stefano Salsano, Carla-Fabiana Chiasserini, and Francesco Malandrino. Algorithms for the design of 5g networks with vnf-based reusable functional blocks. *Annals of Telecommunications*, 74(9):559–574, 2019.

Mosharaf Chowdhury, Muntasir Raihan Rahman, and Raouf Boutaba. Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. *IEEE/ACM Transactions on Networking*, 20(1):206–219, 2012. doi: 10.1109/TNET.2011.2159308.

NM Mosharaf Kabir Chowdhury and Raouf Boutaba. Network virtualization: state of the art and research challenges. *IEEE Communications magazine*, 47 (7):20–26, 2009.

VNI Cisco Mobile. Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021 white paper, 2017.

Gianni Codato and Matteo Fischetti. Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766, 2006.

Sophie Constans, Nour-Eddin El Faouzi, Olivier Goldschmidt, and Rémy Fondacci. Optimal flight level assignment: introducing uncertainty. In *Proceeding of the 4th INO Workshop*, 2004.

Sophie Constans, Nicolas Gadenne, and Rémy Fondacci. Applying genetic techniques to the tactical flight level assignment. In *16th Mini-EURO Conference and 10th Meeting of EWGT Proceedings, Poznan, Poland*, pages 13–16, 2005.

Jean-François Cordeau, Goran Stojković, François Soumis, and Jacques Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation science*, 35(4):375–388, 2001.

Alysson M Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers & operations research*, 32(6):1429–1450, 2005.

Adele Cutler and Olga I Cordero-Brana. Minimum hellinger distance estimation for finite mixture models. *Journal of the American Statistical Association*, 91 (436):1716–1723, 1996.

Erik Dahlman, Stefan Parkvall, and Johan Skold. *5G NR: The next generation wireless access technology*. Academic Press, 2020.

Fabio D'Andreagiovanni, Carlo Mannino, and Antonio Sassano. GUB covers and power-indexed formulations for wireless network design. *Manag. Sci.*, 59(1):142–156, 2013. doi: 10.1287/mnsc.1120.1571. URL https://doi.org/10.1287/mnsc.1120.1571.

Emilie Danna, Edward Rothberg, and Claude Le Pape. Exploring relaxation induced neighborhoods to improve mip solutions. *Mathematical Programming*, 102 (1):71–90, 2005.

George B. Dantzig. Linear Programming under Uncertainty. *Management Science*, 1(3/4):197–206, 1955.

Freddy Delbaen. A remark on slutsky's theorem. In *Séminaire de Probabilités XXXII*, pages 313–315. Springer, 1998.

ETSI. Network functions virtualisation (nfv);architectural framework. https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf, 2014.

EUROCONTROL. Coda digest - all-causes delay and cancellations to air transport in europe-annual report 2019. http://www.atceuc.org/uploads/docs/eurocontrol-coda-digest-annual-report-2019.pdf, 2019a.

EUROCONTROL. Network operations report 2019-main report. https://www.eurocontrol.int/sites/default/files/2020-04/nm-annual-network-operations-report-2019-main-report.pdf, 2019b.

EUROCONTROL. European network operations plan 2019 – 2024. https://www.eurocontrol.int/sites/default/files/2019-07/european-network-operations-plan-2019-2024-2.1.pdf, 2019c.

European-5G-Observatory. European 5g observatory. http://5gobservatory.eu/observatory-overview/observatory-reports.

Bernard Fortz and Michael Poss. An improved benders decomposition applied to a multi-layer network design problem. *Operations research letters*, 37(5):359–364, 2009.

Akli Fundo, Dritan Nace, and Chenghao Wang. A heuristic approach for the robust flight level assignment problem. In *International Conference on Belief Functions*, pages 86–94. Springer, 2018.

Akli Fundo, Jean-Benoist Leger, Dritan Nace, and Chenghao Wang. Dealing with uncertainty in atm-the flight level assignment problem. In *21e congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2020)*, 2020.

Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3): 471–483, 2014. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2013.09.036. URL https://www.sciencedirect.com/science/article/pii/S0377221713007911.

Rosario Giuseppe Garroppo, Maria Grazia Scutellà, and Fabio D'Andreagiovanni. Robust green wireless local area networks: A matheuristic approach. *J. Netw. Comput. Appl.*, 163:102657, 2020. doi: 10.1016/j.jnca.2020.102657. URL https://doi.org/10.1016/j.jnca.2020.102657.

Jose Manuel Gimenez-Guzman, Alejandra Martínez-Moraian, Rene D Reyes-Bardales, David Orden, and Ivan Marsa-Maestre. Flight level assignment using graph coloring. *Applied Sciences*, 10(18):6157, 2020.

Ali Gohar and Gianfranco Nencioni. The role of 5g technologies in a smart city: The case for intelligent transportation system. *Sustainability*, 13(9):5188, 2021.

David E Goldberg and John Henry Holland. Genetic algorithms and machine learning. 1988.

Davit Harutyunyan and Roberto Riggio. Flexible functional split in 5g networks. In *2017 13th International Conference on Network and Service Management (CNSM)*, pages 1–9. IEEE, 2017.

Juliver Gil Herrera and Juan Felipe Botero. Resource allocation in nfv: A comprehensive survey. *IEEE Transactions on Network and Service Management*, 13 (3):518–532, 2016.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

John N Hooker. Planning and scheduling by logic-based benders decomposition. *Operations research*, 55(3):588–602, 2007.

John N Hooker and Greger Ottosson. Logic-based benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.

John N Hooker and Hong Yan. Logic circuit verification by benders decomposition. *Principles and practice of constraint programming: the newport papers*, pages 267–288, 1995.

IBM-ILOG-CPLEX. Ilog cplex. http://www-01.ibm.com/software.

ICAO. Icao doc 444 air traffic management. https://ops.group/blog/wp-content/uploads/2017/03/ICAO-Doc4444-Pans-Atm-16thEdition-2016-OPSGROUP.pdf, 2016.

Johannes Inführ and Günther R. Raidl. Introducing the virtual network mapping problem with delay, routing and location constraints. In Julia Pahl, Torsten

Reiners, and Stefan Voß, editors, *Network Optimization*, pages 105–117, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

Wei Jiang, Mathias Strufe, and Hans D Schotten. Intelligent network management for 5g systems: The selfnet approach. In *2017 European Conference on Networks and Communications (EuCNC)*, pages 1–5. IEEE, 2017.

Olivier Klopfenstein. Tractable algorithms for chance-constrained combinatorial problems. *RAIRO-Operations Research*, 43(2):157–187, 2009.

Olivier Klopfenstein and Dritan Nace. The robust flight level assignment problem, 2008.

Stavros G Kolliopoulos and Clifford Stein. Improved approximation algorithms for unsplittable flow problems. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 426–436. IEEE, 1997.

Christopher Larsson. *5G Networks - Planning, Design and Optimization*. Academic Press, Cambridge, 2018.

Erik G Larsson, Ove Edfors, Fredrik Tufvesson, and Thomas L Marzetta. Massive mimo for next generation wireless systems. *IEEE communications magazine*, 52 (2):186–195, 2014.

Sven Leyffer, Matt Menickelly, Todd Munson, Charlie Vanaret, and Stefan M. Wild. A survey of nonlinear robust optimization. *INFOR: Information Systems and Operational Research*, 58(2):342–373, 2020. doi: 10.1080/03155986.2020.1730676.

Shuopeng Li, Mohand Yazid Saidi, and Ken Chen. Survivable services oriented protection level-aware virtual network embedding. *Computer Communications*, 152: 34–45, 2020. ISSN 0140-3664. doi: https://doi.org/10.1016/j.comcom.2020.01.025. URL https://www.sciencedirect.com/science/article/pii/S0140366419306607.

Tao Li and Antonio A Trani. Evaluating of the benefits of allowing flight level and mach number adjustment for fuel efficiency for flight operations in oceanic airspace. In *2018 Integrated Communications, Navigation, Surveillance Conference (ICNS)*, pages 3D2–1. IEEE, 2018.

Daniel Zhuoyu Long, Melvyn Sim, and Minglong Zhou. The dao of robustness. *History*, 2019.

Marcelo Caggiani Luizelli, Leonardo Richter Bays, Luciana Salete Buriol, Marinho Pilla Barcellos, and Luciano Paschoal Gaspary. Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 98–106. IEEE, 2015.

Antonio Marotta, Fabio D'andreagiovanni, Andreas Kassler, and Enrica Zola. On the energy cost of robustness for green virtual network function placement in 5g virtualized infrastructures. *Computer Networks*, 125:64–75, 2017a.

Antonio Marotta, Enrica Zola, Fabio D'Andreagiovanni, and Andreas Kassler. A fast robust optimization-based heuristic for the deployment of green virtual network functions. *Journal of Network and Computer Applications*, 95:42–53, 2017b.

Silvano Martello. Knapsack problems: algorithms and computer implementations. *Wiley-Interscience series in discrete mathematics and optimiza tion*, 1990.

Thomas L Marzetta. Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE transactions on wireless communications*, 9(11): 3590–3600, 2010.

Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

Marouen Mechtri, Chaima Ghribi, and Djamal Zeghlache. A scalable algorithm for the placement of service function chains. *IEEE Transactions on Network and Service Management*, 13(3):533–546, 2016.

Anne Mercier, Jean-François Cordeau, and François Soumis. A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, 32(6):1451–1476, 2005.

Khalid S. Mohamed, Mohamad Y. Alias, Mardeni Roslee, and Yusuf M. Raji. Towards green communication in 5g systems: Survey on beamforming concept. *IET Communications*, 15(1):142–154, 2021. doi: https://doi.org/10.1049/cmu2.12066.

Dritan Nace, Jacques Carlier, Nhat Linh Doan, and Vu Duong. A linear programming approach for route and level flight assignment. In *5th EUROCONTROL & FAA ATM R&D Seminar*, 2003.

Nisha Panwar, Shantanu Sharma, and Awadhesh Kumar Singh. A survey on 5g: The next generation of mobile communication. *Physical Communication*, 18: 64–84, 2016.

Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. Accelerating the benders decomposition method: Application to stochastic network design problems. *SIAM Journal on Optimization*, 28(1):875–903, 2018.

Wonil Roh, Ji-Yun Seol, Jeongho Park, Byunghwan Lee, Jaekon Lee, Yungsoo Kim, Jaeweon Cho, Kyungwhoon Cheun, and Farshid Aryanfar. Millimeter-wave beamforming as an enabling technology for 5g cellular communications: Theoretical feasibility and prototype results. *IEEE communications magazine*, 52(2):106–113, 2014.

Murray Rosenblatt. A central limit theorem and a strong mixing condition. *Proceedings of the National Academy of Sciences of the United States of America*, 42(1):43, 1956.

Gammana Guruge Nadeesha Sandamali, Rong Su, Yicheng Zhang, and Qing Li. Flight routing and scheduling with departure uncertainties in air traffic flow management. In *2017 13th IEEE International Conference on Control & Automation (ICCA)*, pages 301–306. IEEE, 2017.

Gregor Schaffrath, Christoph Werle, Panagiotis Papadimitriou, Anja Feldmann, Roland Bless, Adam Greenhalgh, Andreas Wundsam, Mario Kind, Olaf Maennel,

and Laurent Mathy. Network virtualization architecture: Proposal and initial prototype. In *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, pages 63–72, 2009.

M Shahidehopour and Yong Fu. Benders decomposition: applying benders decomposition to power systems. *IEEE Power and Energy Magazine*, 3(2):20–21, 2005.

Nashid Shahriar, Shihabur Rahman Chowdhury, Reaz Ahmed, Aimal Khan, Raouf Boutaba, Jeebak Mitra, and Liu Liu. Joint backup capacity allocation and embedding for survivable virtual networks. In *2017 IFIP Networking Conference (IFIP Networking) and Workshops*, pages 1–9, 2017. doi: 10.23919/IFIPNetworking.2017.8264837.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory, Second Edition*. Society for Industrial and Applied Mathematics, USA, 2014. ISBN 1611973422.

Mohammad Shojafar, Luca Chiaraviglio, Nicola Blefari-Melazzi, and Stefano Salsano. P5g: A bio-inspired algorithm for the superfluid management of 5g networks. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–7. IEEE, 2017.

Dimitri Bertsimasand Melvyn Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1):49–71, 2003.

Manuel Soler-Arnedo, Mark Hansen, and Bo Zou. Contrail sensitive 4d trajectory planning with flight level allocation using multiphase mixed-integer optimal control. In *AIAA guidance, navigation, and control (GNC) conference*, page 5179, 2013.

Allen L Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, 21(5):1154–1157, 1973.

Yufeng Tu, Michael O Ball, and Wolfgang S Jank. Estimating flight departure delay distributions—a statistical approach with long-term trend and short-term pattern. *Journal of the American Statistical Association*, 103(481):112–125, 2008.

Adan Vela, Senay Solak, William Singhose, and John-Paul Clarke. A mixed integer program for flight-level assignment and speed control for conflict resolution. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 5219–5226. IEEE, 2009.

Chenghao Wang, Fabio D'Andreagiovanni, and Dritan Nace. Solving a resource allocation problem in rfb-based 5g wireless networks. In *Third International Balkan Conference on Communications and Networking (BalkanCom 2019)*, 2019.

Ingo Wegener. *Complexity theory: exploring the limits of efficient algorithms*. Springer Science & Business Media, 2005.

Liyang Xiao, Zhengpei Wang, Zheyi Tan, and Chenghao Wang. A solution method for the maritime pilot scheduling problem with working hour regulations. *Asia-Pacific Journal of Operational Research*, 38(03):2040015, 2021.

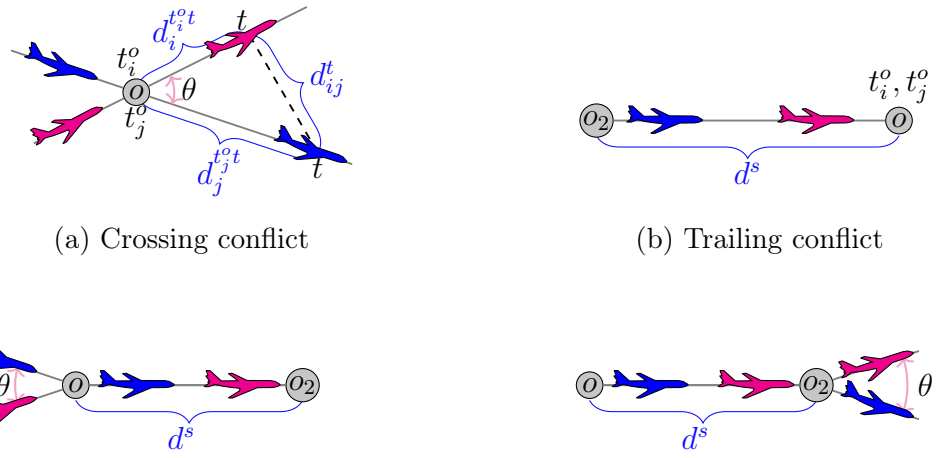
Song Yang, Fan Li, Stojan Trajanovski, Ramin Yahyapour, and Xiaoming Fu. Recent advances of resource allocation in network function virtualization. *IEEE Transactions on Parallel and Distributed Systems*, 32(2):295–314, 2021. doi: 10.1109/TPDS.2020.3017001.

İhsan Yanıkoğlu, Bram L. Gorissen, and Dick den Hertog. A survey of adjustable robust optimization. *European Journal of Operational Research*, 277(3):799–813, 2019. doi: 10.1016/j.ejor.2018.08.03.

Yuncan Zhang, Fujun He, and Eiji Oki. Network service mapping and scheduling under uncertain processing time. In *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–8, 2020. doi: 10.1109/NOMS47738.2020.9110298.

Appendix

I Computation of en-route ATFM delay ω_{ij}



(a) Crossing conflict

(b) Trailing conflict

(c) Converging conflict

(d) Diverging conflict

Figure 6.1.1 – En-route potential conflict between two aircraft cruising at same level

We assume that there may be at most a single potential conflict implying two aircraft. Let denote that some potential conflict is encountered at waypoint o with θ as the crossing angle. Note t_i^o, t_j^o the time that aircraft i and j passes through the conflict point o, respectively. Let $d_i^{t_i^o t}$, $d_j^{t_j^o t}$ be the distance that aircraft i (respectively, j) flies from conflict point o, respectively, until time t. Let d_{ij}^t indicate the separation distance (d_{ij}^{min} denote the minimum distance, respectively) between two aircraft i and j at instant t at the intersecting waypoint o for crossing conflict. The separation distance of the pair of aircraft is then formulated as:

$$d_{ij}^{t\ 2} = (d_i^{t_i^o t})^2 + (d_j^{t_j^o t})^2 - 2(d_i^{t_i^o t})(d_j^{t_j^o t}) \cos\theta \tag{6.1}$$

$$= v_i^2(t-t_i^o)^2 + v_j^2(t-t_j^o)^2 - 2v_iv_j(t-t_i^o)(t-t_j^o)\cos\theta \tag{6.2}$$

$$= v_i^2([1+\rho^2-2\rho\cos\theta]t^2 - 2[t_i^o+\rho^2t_j^o-\rho\cos\theta(t_i^o+t_j^o)]t + [t_i^{o2}+\rho^2t_j^{o2}-2\rho\cos\theta t_i^ot_j^o]) \tag{6.3}$$

$$= v_i^2[1+\rho^2-2\rho\cos\theta]\left(t - \frac{t_i^o+\rho^2t_j^o-\rho\cos\theta(t_i^o+t_j^o)}{\sqrt{1+\rho^2-2\rho\cos\theta}}\right)^2 + v_i^2[t_i^{o2}+\rho^2t_j^{o2}-2\rho\cos\theta t_i^ot_j^o] - v_i^2(t_i^o+\rho^2t_j^o-\rho\cos\theta(t_i^o+t_j^o))^2 \tag{6.4}$$

$$= v_i^2[1+\rho^2-2\rho\cos\theta]\left(t - \frac{t_i^o+\rho^2t_j^o-\rho\cos\theta(t_i^o+t_j^o)}{1+\rho^2-2\rho\cos\theta}\right)^2 + \frac{v_j^2(t_j^o-t_i^o)^2\sin^2\theta}{1+\rho^2-2\rho\cos\theta} \tag{6.5}$$

$$\geq \lambda^2v_j^2(t_j^o-t_i^o)^2 \text{ if } \sin^2\theta > 0, \tag{6.6}$$

where v_i, v_j are the velocity of aircraft i and j, respectively, $\rho = v_j/v_i$, $\lambda = \sin\theta/\sqrt{\rho^2-2\rho\cos\theta+1}$.

If $\sin^2\theta = 0$, then $\cos\theta = 1$ under the "Semicircular/hemispheric" rule, and the trailing conflict may occur for $t \in [\max(t_i^o, t_j^o), \min(t_i^o + d_s/v_i, t_j^o + d_s/v_j)]$ where d_s specifies the distance between two waypoints when a trailing conflict is encountered. We then have:

$$d_{ij}^t{}^2 = v_i^2(1-\rho)^2\left(t - \frac{t_i^o-\rho t_j^o}{1-\rho}\right)^2 \tag{6.7}$$

$$= \min\begin{cases} v_j^2(t_j^o-t_i^o)^2, t=t_i^o \\ v_i^2(t_j^o-t_i^o)^2, t=t_j^o \\ (v_j(t_j^o-t_i^o)+d_s(1-\rho))^2, t=t_i+d_s/v_i \\ (v_j(t_j^o-t_i^o)+d_s(1-\rho))^2/\rho^2, t=t_j+d_s/v_j \end{cases} \tag{6.8}$$

$$= (v_j(t_j^o-t_i^o)-d_s|1-\rho|)^2/\max(\rho^2,1) \tag{6.9}$$

Therefore, the minimum distance for two aircraft of crossing conflict and trailing conflict is formulated:

$$d_{ij}^{min} = \begin{cases} \lambda v_j|t_i^o-t_j^o|, \text{crossing conflict: } \sin\theta > 0 \\ (v_j|t_i^o-t_j^o| - d^s|1-\rho|)/\max(\rho,1), \text{trailing conflict: } \sin\theta = 0 \end{cases} \tag{6.10}$$

To cope with the converging conflict and diverging conflict, we put together the four types of conflict in a unified scenario in Figure 6.1.2. Then, the minimum

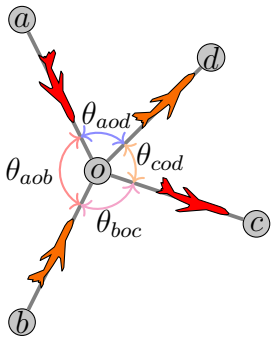


Figure 6.1.2 – A unified en-route potential conflict between two aircraft cruising at same level

separation time t_{ij}^{msd} for two aircraft to pass the conflict point for all possible scenarios is reformulated:

$$t_{ij}^{\text{msd}} = \max \begin{cases} t_{ij}^{min}(\theta_{aob}, d_{oa}, \rho, v_j, S) \\ t_{ij}^{min}(\theta_{aod}, d_{oa}, \rho, v_j, S) \\ t_{ij}^{min}(\theta_{boc}, d_{oc}, \rho, v_j, S) \\ t_{ij}^{min}(\theta_{cod}, d_{oc}, \rho, v_j, S) \end{cases} \tag{6.11}$$

$$t_{ij}^{min}(\theta, d^s, \rho, v_j, d_{ij}^{min}) = \begin{cases} \frac{d_{ij}^{min}}{\lambda v_j}, \text{if } \sin\theta > 0 \\ \frac{\max(1,\rho) d_{ij}^{min} + d^s |1-\rho|}{v_j}, \text{otherwise} \end{cases} \tag{6.12}$$

The two aircraft have a potential conflict at some intersecting waypoint o if and only if the minimum separation distance between them is less than minimum separation S. Then the probability of potential conflict is formulated as below:

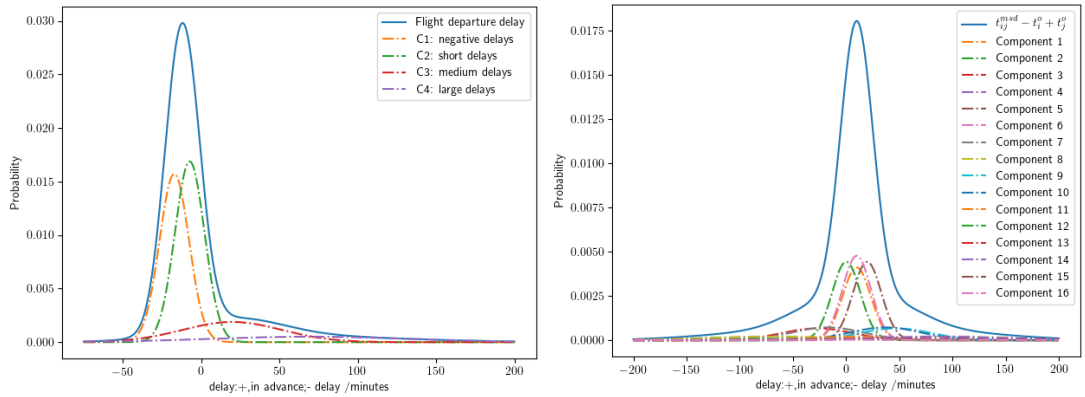
$$\mathsf{P}_{ij}^{conflict} = \mathsf{P}\left(|d_{ij}^{min}| \leq S\right) = \mathsf{P}\left(|t_i^o - t_j^o| \leq t_{ij}^{\text{msd}}\right) \tag{6.13}$$

The induced en-route ATFM delay ω_{ij} of resolution for pairwise conflict is then formulated by:

$$\omega_{ij} = (t_{ij}^{\text{msd}} - t_i^o + t_j^o)\mathbb{1}(t_{ij}^{\text{msd}} - t_i^o + t_j^o)_{[0,\infty)}\mathbb{1}(t_i^o - t_j^o)_{[0,\infty)} \in [0, t_{ij}^{\text{msd}}], \tag{6.14}$$

where $\mathbb{1}(x)_A$ is a indicator function: $\mathbb{1}(x)_A = \{1, x \in A; 0, x \notin A\}$, the second term denoted by $\mathbb{1}(t_{ij}^{\text{msd}} - t_i^o + t_j^o)_{[0,\infty)}$ specifies that the induced en-route ATFM delay of associated flight due to resolution should be positive meanwhile the potential conflict existed, $\mathbb{1}(t_i^o - t_j^o)_{[0,\infty)}$ denotes whether the aircraft i arrives latter than j at the potential conflict point o.

Assume that flight departure time delay follows a GMM (see Figure 6.1.3a),



(a) Distribution of flight departure delay (b) Distribution of $t_{ij}^{\text{msd}} - t_i^o + t_j^o$

Figure 6.1.3 – GMM distributions associated with flight departure delays

for which the component parameters are presented in Table 3.6.1. Then term $t_{ij}^{\text{msd}} - t_i^o + t_j^o$ follows also a GMM as illustrated in Figure 6.1.3b with a given value of t_{ij}^{msd}. Truncated by two indicator functions $\mathbb{1}(t_{ij}^{\text{msd}} - t_i^o + t_j^o)_{[0,\infty)}$ and $\mathbb{1}(t_i^o - t_j^o)_{[0,\infty)}$, the induced en-route ATFM delay ω_{ij} then follows a truncated GMM as shown in Figure 6.1.4.

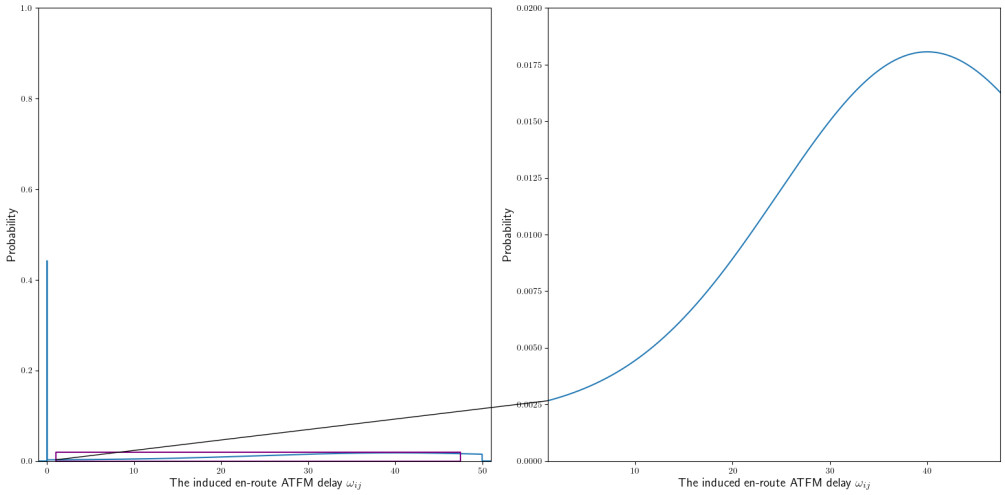


Figure 6.1.4 – truncated GMM distribution of ω_{ij}