Technische Universiteit Eindhoven
Master Degree in Industrial and Applied Mathematics

Politecnico di Torino
Master Degree in Mathematical Engineering

Short-Term Planning Model for ASML Field Factory DUV/YS Upgrades

Author:
Rosa Maria Siervo

Supervisor: Dr. Ir. C.A.J. (Cor) Hurkens
Department of Mathematics and Computer Science

Supervisor: Prof. Dr. Paolo Brandimarte
Department of Mathematical Sciences
"Giuseppe Luigi Lagrange"

Supervisor: B.S. Marion de Proost
Department of "Field Factory"
Supervisor: Drs. Sander Schepens MSc.
Department of "Operations Excellence"

October 30, 2017

"C' est une folie de haïr toutes les roses parce que une épine vous a piqué, d' abandonner tous les rêves parce que l' un d' entre eux ne s' est pas réalisé, de renoncer à toutes les tentatives parce qu' on a échoué... C' est une folie de condamner toutes les amitiés parce qu' une d' elles vous a trahi, de ne croire plus en l' amour juste parce qu' un d' entre eux a été infidèle, de jeter toutes les chances d' être heureux juste parce que quelque chose n' est pas allé dans la bonne direction. Il y aura toujours une autre occasion, un autre ami, un autre amour, une force nouvelle. Pour chaque fin il y a toujours un nouveau départ."

by Antoine de Saint-Éxupéry in Le Petit Prince

Abstract

ASML is one of the world's leading manufacturers of chip-making equipment.
FF DUV/YS Upgrades is a growing department within ASML that is responsible for performing complex field upgrades that involve ASML's DUV scanners and YS metrology systems. These upgrades require that teams of specialized people go to customers' locations which are spread over Europe, US and Asia.

In our research, we first develop a mathematical model for the optimization of short-term FF DUV/YS Upgrades manpower planning. This model turns out to be very complex in the number of variables and constraints and it cannot be efficiently solved for large scale instances by using traditional techniques.
Due to the complexity of the model, a decomposition approach is needed. We develop two different decomposition strategies, one based on Column Generation and the other one based on Benders' Decomposition combined with Column Generation.

We implement all our algorithms in AIMMS and perform herewith a set of computational experiments. It turns out that both these strategies seem to be promising, but none of the proposed heuristics and enhancement techniques succeed in solving the largest instance that we tested.

The results of our research open to possibilities for further development in this direction.

Keywords. Operations research, Optimization modeling, MIP, Manpower planning, Column Generation, Heuristics, Benders' Decomposition.

Executive Summary

This report is the result of a six-month internship at the department of FF DUV/YS Upgrades at ASML, in Veldhoven (The Netherlands). It was commissioned to examine the complexity of the short-term manpower planning problem inside the department and to investigate a solution methodology that could possibly solve the problem in the case of large input data.

ASML is one of the world's leading manufacturers of chip-making equipment. Its mission is to invent, develop, manufacture and service advanced technology for high-tech lithography, metrology and software solutions for the semiconductor industry. FF DUV/YS Upgrades is a growing department within ASML that is responsible for performing complex field upgrades that involve ASML's DUV scanners and YS metrology systems. These upgrades need teams of experts whose functions can be classified as upgrade engineers, coordinators, generalists or material handlers. The execution of a DUV or YS upgrade varies from a couple of days to 16 weeks and, during this phase, manpower should be at customer's location. Locations are spread over Europe, US and Asia.

FF DUV/YS Upgrades manpower planning is a very complex problem. To date, planners have to manually schedule about 220-300 people while guaranteeing the observance of some important constraints. For example, planning is currently done by program, which means that employees are specifically assigned to one upgrade type, even though their competences allow them to perform upgrades from other programs, if needed. The main reasons for this strategy are that the increasing number of programs made the department grow up to (more than) 300 FTE very quickly and that, moreover, dedicated teams let people work well together and so help to release a service of higher quality and within the cycle time. Nevertheless, this way of working may not be optimal in terms of utilization and travel costs; hence several other possibilities should be analyzed. The rate of growth of FF DUV/YS Upgrades and, more in general, of FF makes significant the efficiency that could be gained through a better planning strategy. Moreover, for planners it becomes more complex to even generate a suboptimal plan that satisfies all the constraints. That is why the investigation into the extent an efficiency improvement can be made by changing the way manpower is planned assumes now particular importance and it is not possible to postpone this research any further. For example, flexibility can be increased by using smaller teams for specific parts of the process and the larger capacity groups of engineers per competence could be used to increase utilization and reduce waiting time.

A first step in the development of a tool that allows for scenario analysis consists in the study of the mathematical complexity of FF DUV/YS Upgrades manpower planning problem and of a solution methodology that could solve this problem in the case of large input data. From a mathematical perspective, FF DUV/YS Upgrades manpower

planning problem is a Mixed Integer Program (MIP). MIPs are known to be NP-hard problems, which means that it is difficult (if not impossible) to construct an efficient algorithm that can solve it in a reasonable amount of time for large-scale instances.

In our research, we first identify the constraints and variables of the problem and analyze which constraints should be included in our model. Then, we write the problem in its mathematical formulation. This first phase already highlights the complexity in the optimization modeling of, among others, travel and rest period constraints. The model in its compact MIP formulation results indeed to be intractable for large input data and new solution strategies are therefore analyzed.
Due to the complexity of the model, a decomposition approach is needed. We develop two different decomposition strategies, one based on Column Generation and the other one based on Benders' Decomposition combined with Column Generation.

Column Generation allows to relax some integrality constraints from the original formulation and it manages to solve some instances for which the MIP problem could not find a feasible solution in a reasonable amount of time. However, it fails in solving the largest instance. Regarding Benders' Decomposition, it shows problems of weak convergence of the lower bound in order to prove optimality, even after the application of enhancement techniques. It can solve more instances than the original MIP formulation but it also fails for larger input data.

The results of our research suggest that a further investigation is still needed in order to find out the best approach to solve such a complex problem. The first insight is that a decomposition strategy is particularly indicated for FF DUV/YS Upgrades department. Therefore, additional enhancement techniques for Benders' Decomposition and Column Generation could be analyzed. Moreover, if these approaches turn out to be inefficient for this specific situation, many other techniques are available in literature. An example is a decomposition of manpower and upgrades according to their regions, so that in a first phase only local employees are assigned to upgrades. This approach is based on a decentralized vision of upgrades and manpower that leads to more local ownership. Other examples are application of pure heuristics or metaheuristics such as local search algorithms and evolutionary algorithms.

Acknowledgments

There are many people that I have had the pleasure to meet and work with throughout my studies, making these 5 years of university a valuable and rewarding experience, and I would like to take the opportunity here to give some special thanks.

Firstly, I would like to thank all my supervisors for their help and advice.
Prof. Cor Hurkens, because you introduced me to the field of operations research, you guided me throughout my thesis and you have always been available when I required any advice. Because even when your agenda was completely booked, you had time for me.
Sander and Marion, for your enthusiasm since my first day of internship, for your continuous effort in trying to help me and for your unconditional and constant support.
Prof. Brandimarte, for your precious suggestions during our e-mail correspondence.

I would like to heartily thank Dr. Maria Vlasiou and Dr. Rudi Pendavingh, for being the members of the assessment committee and for their patience in waiting for my thesis.

Thanks to the people I met in ASML, in particular Ramon, Neal and all FF DUV/YS Upgrades department. A special thanks to the business support team: Tony, Jordy, Ursus, Ingrid, Cindy, Fred and again Marion. For all our good time together, for our lunches, for our laughs and for your valuable advice for my future choices.

I am also grateful to Lianne, for her disinterested and genuine tips. I really appreciate the time she gave me.

Thanks to Alessandro, my partner in this experience abroad: together we tried out new culinary recipes, we went through many bureaucratic adventures and we also "studied" Dutch! I couldn't have asked for a better companion, thank you. And thanks to Arash, Aryya and all the "Boschdijk family", for always making me feel loved and for caring so much about me.

I cannot forget my 4 years in Torino and the first person I want to warmly acknowledge is prof. Cortese. Because your teachings have been and will always be of valuable inspiration to me. Knowledge is an infinite source of beauty and, as long as we let it guide our life, we can be sure that it will be well worth it in the end.

Thanks to all my dear friends in Torino, in particular Alex, Federico, Erika, Debhora, Lucia, Pamela, Fernanda, Luana, Gloria, Elisa and Eugenia. For your constant presence in my life and for all your words of support in these 5 years together.

I am grateful to Collegio Einaudi because living there was one of the best experiences of my life. In particular, I want to say thanks to all those crazy people on the third floor Crocetta, I missed them so much this year! A special thanks goes to Pietro. You are one of my best friends, you understand me like nobody else does and I cannot imagine my life without our video calls, funny conversations and (sometimes) deep discussions!

Thanks to the best friends of my whole life, Roby, Filli, Assu and Fiore, because the distance can never move us apart and, no matter where I am, I can always feel your warmth.

Thanks to my amazing family. In particular, mom and dad have always been there for me, with all their wise advice. Not only did they praise my successes and rewards, but they also shared my moments of sadness and disappointment, making them easier to cope with. Thank you!

Finally, there is a special person to whom I wish to dedicate this thesis. She was the strongest woman I have ever met and I have always admired her courage, determination and love for life. After one of our last meetings I wanted to buy her a book that could express all those feelings she could so easily transmit to me. Unfortunately, the day when I could buy her that book never came, but today I want at least to dedicate her my own thesis. This work is the result of my strongest passions and I put in it all my determination, efforts and love... Zia Silvia, this is for you.

Eindhoven, The Netherlands Rosa Maria
8 October 2017

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

AFSM(p)	Agenda Feasibility Subproblem Master for employee $p \in \mathcal{P}$
AS(p)	Agenda Subproblem for employee $p \in \mathcal{P}$
ASM(p)	Agenda Subproblem Master for employee $p \in \mathcal{P}$
ASML	Advanced Semiconductor Materials Lithography
ASP(p)	Agenda Subproblem Pricing for employee $p \in \mathcal{P}$
AT	Account Team
B&B	Branch-and-Bound
B&BC	Branch-and-Benders-Cut
BD	Benders' Decomposition
BMP	Benders' Master Problem
BS	Benders' Subproblem
CG	Column Generation
CP	Central Planning
CS	Customer Support
D&E	Development & Engineering
D&S	Demand & Supply
DUV	Deep Ultraviolet
EUV	Extreme Ultraviolet
FF	Field Factory
FTE	Full-Time Equivalent
GUI	Graphical User Interface
IC	Integrated Circuit

ILP	Integer Linear Program
LP	Linear Program
MIP	Mixed Integer Program
MP	Master Problem
PI	Performance Indicator
PP(p)	Pricing Problem for employee $p \in \mathcal{P}$
RMP	Restricted Master Problem
WLB	Work-Life Balance
YS	YieldStar

List of Symbols

$\mathcal{T}$	set of all teams
$\mathcal{P}$	set of all people/employees to be scheduled
$\mathcal{P}^{\mathcal{T}}_{tm}$	set of people in team $tm \in \mathcal{T}$
$\mathcal{L}$	set of locations
$\mathcal{PJ}$	set of all projects, i.e. an indivisible subset of days of an upgrade, usually of length one week
$\mathcal{PJ}^{\mathcal{P}}_{p}$	set of projects in which employee $p \in \mathcal{P}$ can be involved
$\mathcal{PJ}^{\mathcal{T}}_{tm}$	set of projects in which team $tm \in \mathcal{T}$ can be involved
$\mathcal{R}$	set of all roles needed to execute the upgrades
$\mathcal{R}^{\mathcal{PJ}}_{prj}$	set of roles required in project $prj \in \mathcal{PJ}$
$\mathcal{R}^{\mathcal{PPJ}}_{p,prj}$	set of roles required in project $prj \in \mathcal{PJ}$ and for which person $p \in \mathcal{P}$ becomes skilled at the latest in the starting date of $prj \in \mathcal{PJ}$. The set is empty if $prj \notin \mathcal{PJ}^{\mathcal{P}}_{p}$
$\mathcal{TD}$	set of possible number of consecutive travel days, that is $\mathcal{TD} := \{0, 1, 2, 3\}$
$\mathcal{D}$	set of all days in the planning horizon
$\mathcal{D}^{*}_{n}$	set of days in the planning horizon, with the last $n \in \mathbb{N}$ days excluded
$\mathcal{D}^{\mathcal{PJ}}_{prj}$	set of days in which project $prj \in \mathcal{PJ}$ is performed
$\mathcal{D}^{\mathcal{IN}}_{p}$	set of days in which person $p \in \mathcal{P}$ is involved in some activity (either direct, indirect or personal holidays) specified in input to the tool
$\mathcal{AD}_{p}$	set of all arc days of person $p \in \mathcal{P}$ in the planning horizon
$\mathcal{Q}_{p}$	set of feasible agendas of locations for employee p
$\mathcal{K}_{p}$	set that enumerates employee p's agendas in MP or ASM(p) or AFSM(p)

$\mathcal{K}'_p$	set that enumerates employee p's agendas in RMP or in restricted ASM(p) or in restricted AFSM(p)
$\mathcal{K}$	set of elements in at least one $\mathcal{K}_p$, that is $\mathcal{K} := \bigcup_p \mathcal{K}_p$
$\mathcal{D}^{\mathcal{WED}}$	set of all Wednesdays in the planning horizon
$\mathcal{D}_n^{*\mathcal{WED}}$	set of Wednesdays in the planning horizon, with the last $n \in \mathbb{N}$ Wednesdays excluded
$\gamma_{prj_1,prj_2} \in \{0,1\}$	takes value 0 if two projects $prj_1 \in \mathcal{PJ}$ and $prj_2 \in \mathcal{PJ}$ overlap in at least one day and occur in the same location
$\alpha_{prj,r} \in \mathbb{N}$	is the number of people needed for project $prj \in \mathcal{PJ}$ in role $r \in \mathcal{R}_{prj}^{\mathcal{PJ}}$
$\delta_{prj} \in \mathbb{N}$	is the duration of project $prj \in \mathcal{PJ}$, in number of days
$\tau_{l_1,l_2} \in \mathcal{TD}$	is the number of travel days from location $l_1 \in \mathcal{L}$ to location $l_2 \in \mathcal{L}$
$\beta_{p,d}^{IN}$	is the input location of person $p \in \mathcal{P}$, that is the location given as input when $d \in \mathcal{D}_p^{\mathcal{IN}}$
β_{prj}^{PJ}	is the location where project $prj \in \mathcal{PJ}$ takes place
β_p^H	is the home/base location of person $p \in \mathcal{P}$
$\eta_p \in \mathbb{N}$	is the threshold in days that determines whether a period outside home location is short ($\leq \eta_p$) or long ($> \eta_p$) for employee $p \in \mathcal{P}$
$\rho_p^{SH} \in \mathbb{N}$	is the length of rest period, in days, for person $p \in \mathcal{P}$ after a short period outside his home location
$\rho_p^{LH} \in \mathbb{N}$	is the length of rest period, in days, for person $p \in \mathcal{P}$ after a long period outside his home location
$\omega_p \in \mathbb{N}$	is the maximum number of days abroad that person $p \in \mathcal{P}$ can have before going in rest period
$M \in \mathbb{N}$	is the high penalization coefficient used for dummy variables.
$\xi^{(p),k} \in conv(\mathcal{Q}_p)$	are points in the convex hull of $\mathcal{Q}_p$ representing some feasible agendas $k \in \mathcal{K}_p$ for $p \in \mathcal{P}$
$\psi_{p,k} \in \mathbb{N}$	is the cost of agenda $\xi^{(p),k}$
$\pi_2^{(p)} \in \mathbb{R}^{\lvert\mathcal{PJ}_p^{\mathcal{P}}\rvert}$	is the vector of shadow prices of constraint (3.1) for CG approach and of constraint (3.8) for BD approach

$\pi_1^{(p)} \in \mathbb{R}$	is the shadow price of constraint (3.2) for CG approach and of constraint (3.9) for BD approach
$\pi_0^{(p)} \in \mathbb{R}$	is the shadow price of constraint (3.3) for CG approach and of constraint (3.10) for BD approach
$\tilde{\gamma}_{prj_1,prj_2} \in \{0,1\}$	takes value 0 when two projects prj_1 and prj_2 are "incompatible", that is when it is not possible for any employee to perform both of them, because of overlapping or no enough room for travel days between them
$X^{\mathcal{PR}}_{p,prj,r} \in \mathbb{R}_{[0,1]}$	represents whether person $p \in \mathcal{P}$ is assigned to project $prj \in \mathcal{PJ}^{\mathcal{P}}_p$ for role $r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}$
$X^{\mathcal{P}}_{p,prj} \in \{0,1\}$	represents whether person $p \in \mathcal{P}$ is assigned to project $prj \in \mathcal{PJ}^{\mathcal{P}}_p$
$X^{\mathcal{T}}_{tm,prj} \in \{0,1\}$	represents whether team $tm \in \mathcal{T}$ is assigned to project $prj \in \mathcal{PJ}^{\mathcal{T}}_{tm}$
$L_{p,l,d} \in \{0,1\}$	takes value 1 if person $p \in \mathcal{P}$ is at location $l \in \mathcal{L}$ on day $d \in \mathcal{D}$
$T_{p,d} \in \{0,1\}$	takes value 1 if person $p \in \mathcal{P}$ is traveling on day $d \in \mathcal{D}$
$A^{SAB}_{p,d} \in \mathbb{R}_{[0,1]}$	represents a short abroad arc of length η_p for person $p \in \mathcal{P}$ that starts on day $d \in \mathcal{D}$
$A^{LAB}_{p,d} \in \mathbb{R}_{[0,1]}$	represents a long abroad arc of length ω_p for person $p \in \mathcal{P}$ that starts on day $d \in \mathcal{D}$
$A^{USAB}_{p,d} \in \mathbb{R}_{\geq 0}$	represents a unit backward short abroad arc for person $p \in \mathcal{P}$ on day $d \in \mathcal{D}$
$A^{ULAB}_{p,d} \in \mathbb{R}_{\geq 0}$	represents a unit backward long abroad arc for person $p \in \mathcal{P}$ on day $d \in \mathcal{D}$
$A^{SH}_{p,d} \in \mathbb{R}_{[0,1]}$	represents a short home arc of length ρ^{SH}_p for person $p \in \mathcal{P}$ that starts on day $d \in \mathcal{D}$
$A^{LH}_{p,d} \in \mathbb{R}_{[0,1]}$	represents a long home arc of length ρ^{LH}_p for person $p \in \mathcal{P}$ that starts on day $d \in \mathcal{D}$
$A^{UH}_{p,d} \in \mathbb{R}_{[0,1]}$	represents a unit home arc for person $p \in \mathcal{P}$ on day $d \in \mathcal{D}$
$D^1_{prj,r} \in \mathbb{R}_{\geq 0}$	are dummy variables used to guarantee that the formulation of the original problem is always feasible
$L^{(p)} \in \mathcal{Q}_p$	is the array representing the agenda of employee $p \in \mathcal{P}$, that is $L^{(p)} := (L_{p,l,d})_{l \in \mathcal{L}, d \in \mathcal{D}}$
$\Psi_p \in \mathbb{N}$	is the cost of employee p's agenda $L^{(p)}$

$C_p \in \mathbb{R}_{\geq 0}$	are additional variables used to formulate the Benders' Master Problem
$D_p^2 \in \mathbb{R}_{\geq 0}$	are dummy variables used to formulate the AFSM(p)
$\Lambda_{p,k} \in \mathbb{R}_{[0,1]}$	represents the coefficients of the convex combination of agendas $\xi^{(p),k}$
$\xi^{(p),new} \in \mathcal{Q}_p$	represents a new agenda that is found in PP(p) or in ASP(p)
$\Psi_{p,new} \in \mathbb{N}$	is the cost of the new agenda $\xi^{(p),new}$ that is found in PP(p) or in ASP(p)
$L_{p,l,wed}^{WED} \in \{0,1\}$	takes value 1 if employee $p \in \mathcal{P}$ is at location $l \in \mathcal{L}$ on Wednesday $wed \in \mathcal{D}^{\mathcal{WED}}$
$N_{p,wed} \in \mathbb{N}$	represents the number of travel days that employee p does between $wed \in \mathcal{D}_1^{*\mathcal{WED}}$ and the next Wednesday

Chapter 1

Introduction

This introductory chapter aims to present the motivations and objectives of the thesis. Some background information about ASML and its Field Factory (FF) department is provided, followed by the project description.

1.1 ASML and Field Factory

About ASML

ASML is the world's leading provider of lithography systems for the semiconductor industry, manufacturing complex machines that are critical to the production of integrated circuits (also called ICs or chips).

Overall, ASML has more than 60 locations in 16 countries and its corporate headquarters is in Veldhoven, The Netherlands. The challenge in microlithography is to make chips as small as possible. This need for miniaturization follows a pattern, usually referred to as "Moore's Law", based on the observation of INTEL co-founder Moore who predicted as early as 1965 that the number of transistors on a chip would double every 18 months. ASML's guiding principle is continuing Moore's Law towards ever smaller, cheaper, more powerful and energy-efficient semiconductors that result in increasingly powerful and capable electronics and enable the world to progress within a multitude of fields, including health care, technology, communications, energy, mobility, and entertainment.

ASML's Systems

The company is organized in three main business lines: Deep Ultraviolet (DUV) scanners, Extreme Ultraviolet (EUV) scanners and Applications.

At the heart of ASML's product portfolio is the lithography system, also called a scanner. ASML is currently working on the development and production of two types of machines, the PAS 5500 series and the TWINSCAN series. Founded in 1984, the first ASML's scanner was a PAS system. In 2000 the TWINSCAN platform was introduced and it increased the throughput significantly thanks to its dual-stage approach, consisting in the exposure of one wafer while another wafer was being measured so that two wafers could be processed at the same time.

While both the TWINSCAN XT and the TWINSCAN NXT machines belonged to the DUV lithography, ASML achieved a major milestone with EUV lithography in 2010 when it introduced the TWINSCAN NXE systems, equipped with EUV light source technology. EUV is expected to become the predominant lithography technology for the coming years: NXE systems are targeted for production of ICs down to minimum features of 13 nm with single patterning, addressing current Memory and Logic roadmaps and processes down to the 5 nm node.

Applications are the third business line in ASML. Indeed, lithography systems are large (container sized), expensive (between 15 and 100 million dollar) and also complex. First, for the most advanced chips the scanner settings have to be optimized for each chip pattern that the customer wants to print. The pattern itself will also require some adjustments to enhance its printability. ASML offers a number of computational lithography products to make those optimizations. Secondly, once the optimum scanner settings have been determined, the lithography system must be constantly kept in this sweet spot. For this purpose, ASML has developed a metrology system and control software called YieldStar (YS). This metrology system measures wafers shortly after they have been exposed by the scanner. The collected data is used to calculate any necessary adjustments, which are then immediately fed back into the lithography system and converted into smart exposure corrections by the computational lithography products.

This combination of scanner, metrology and software products is called "Holistic Lithography" and is shown in Fig. 1.1.

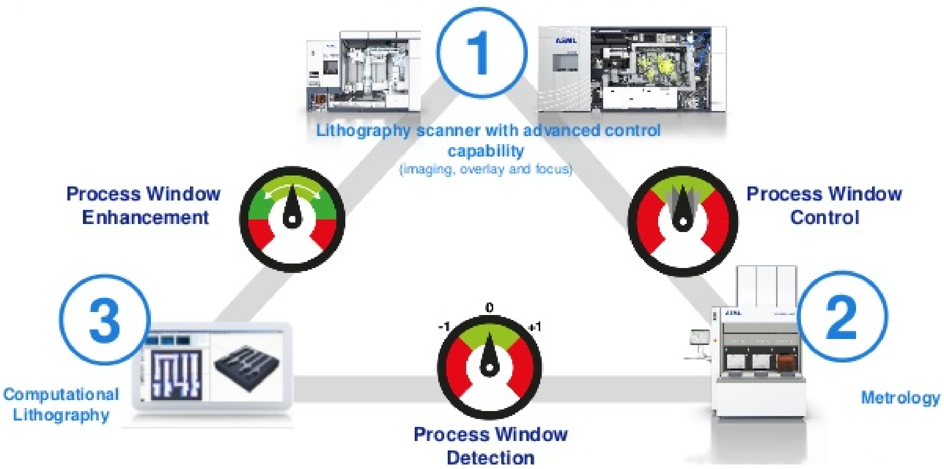


Figure 1.1: ASML Holistic Lithography seeks to maximize lithography process performance and control

ASML Field Factory

Introduction of new system types requires a large number of D&E developers for initial new product development. Systems are then manufactured and installed worldwide at customers. Finally, maintenance of ASML's installed base is performed by the Customer Support (CS) organization which has a worldwide presence in local offices located close to customer fabs.

Field Factory (FF) is a growing division in CS that is responsible for performing upgrades, installs and relocations in the field. FF is currently organized in 3 departments that focus on a subset of the 3 activities (upgrade, install, relocation of ASML's systems), separated on one of 2 product platforms (DUV/YS or EUV): FF DUV/YS Upgrades, FF DUV/YS Install & Relocation, FF EUV Install.

1.2 Project Description

FF DUV/YS Upgrades management is interested in analyzing new possibilities for short-term planning in FF DUV/YS Upgrades department. Indeed, the current way of working may not be optimal in terms of utilization and travel costs and, at the same time, the rate of growth of FF DUV/YS Upgrades and, more in general, of FF makes significant the efficiency that could be gained through a better planning strategy. For example, flexibility can be increased by using smaller teams for specific parts of the process and the larger capacity groups of engineers per competence could be used to increase utilization and reduce waiting time.

To achieve this, a tool for scenario analysis is needed. In particular it should allow for the comparison, within the FF DUV/YS Upgrades department, of several scenarios and their optimal plans, in a short-time horizon of 3 months. For the construction of a mathematical tool, a first step is the study of the mathematical complexity of FF DUV/YS Upgrades manpower planning problem. Therefore, the goal of this thesis consists in the formulation of a mathematical optimization model for FF DUV/YS Upgrades manpower planning problem and in the analysis, through the development of several algorithmic approaches, of solution methods in the case of large input data.

1.3 Thesis Outline

The here presented thesis proceeds in the following order.

After the introduction, the FF DUV/YS Upgrades manpower planning problem is described in Chapter 2 and a mathematical model is subsequently developed. A formal definition of the Performance Indicators (PIs), used to characterize an optimal plan, is also introduced in order to justify our choice of objective function. In Chapter 3, a detailed description of several solution methods is provided, together with their algorithms and some enhancement techniques that we implemented. The computational results are presented in the last section of the chapter, where we perform some comparisons between the different strategies. Consequently, the conclusion is drawn in Chapter 4, together with recommendations for further work.

An overview of the GUI implemented for the tool and a summary of the PIs are included in the appendix. Moreover, App. A is provided as a reference for the theoretical techniques applied in Chapter 3, while App. B contains the history of other approaches that we tried during our research.

Chapter 2

DUV/YS Upgrades: Model Formulation

In this chapter a description of the mathematical model for FF DUV/YS Upgrades department is presented.

2.1 About FF DUV/YS Upgrades Department

FF DUV/YS Upgrades department is responsible for the management of complex field upgrades that involve ASML's DUV scanners and YS metrology systems.

DUV upgrades are divided in several categories called programs: SNEP, OFP2, OFP, UVLS, PEP and TWINSCAN (DEP, FlexWave, etc. . .) upgrades. These upgrades need teams of specialized people whose functions can be classified as upgrade engineers, coordinators, generalists or material handlers. Upgrade engineers execute hardware upgrades and recovery in the field, coordinators prepare and coordinate the daily operations of an upgrade, generalists are highly experienced technicians and material handlers are responsible for the execution of logistical activities in the customer fab, including the transportation of materials between fab laydown areas and machines, and for their SAP administration. The execution of a DUV or YS upgrade varies from 1 to 16 weeks, depending on the program. The locations are spread over Europe, US and Asia.

Every machine upgrade is performed during the so-called execution phase. This phase usually involves all the functions and requires the manpower to be in the upgrade location. However, in order for the upgrade to be successful, two additional phases are also needed: preparation phase and wrap-up phase. The preparation phase involves especially coordinators, who have to work from their local office some weeks in advance to guarantee that the execution phase can occur in the established period. The wrap-up phase is planned for the two weeks immediately after the execution phase and, in this period, the last checks are made and all the service orders are closed.

When employees are not assigned to any machine upgrade, they can be involved in other activities such as traveling, training, etc. Also, an employee can be in rest period and do some work at his base location. Anyway, the main distinction, fundamental for the computation of PIs, is between direct, indirect or "excluded" activities. In particular, working on an upgrade phase is regarded as direct activity; training and traveling as indirect activities; holidays as excluded activities. Moreover, it can happen that an employee is not assigned to any of the previous activities even if he is available: in that case he will carry out some indirect work. Therefore availability is an indirect activity as well. During rest period, employees work from their home location but their work can be either direct or indirect depending on what they are planned to do.

Every month the Demand and Supply cycle, known as D&S, makes a review of all the ASML's DUV and YS systems worldwide in order to check which system could opt for an upgrade. Then, the Account Team contacts the customers and discusses opportunities of upgrades with them. If an agreement is reached, the Central Planning (CP) gets in touch with the Logistics department, which is responsible for checking the availability of all necessary materials at the proposed date, taking into account both suppliers and transportation. Once the Logistics department confirms the proposed plan, CP contacts FF DUV/YS Upgrades department which provides the manpower plan for the requested upgrade. If the necessary manpower is not available, a new starting date for that upgrade is proposed. Due to possible delays or other unexpected problems CP, FF and Logistics have a weekly meeting in which the needed changes in the overall plan for the DUV and YS upgrades are made.

Inside FF DUV/YS Upgrades department, manpower planning is currently done by program. More precisely, coordinators and upgrade engineers are specifically assigned to only one program (SNEP, OFP2,...). The main reasons for this strategy are that the increasing number of programs made the department grow up to more than 300 FTE very quickly and that, moreover, dedicated teams let people work well together and so help to release a service of higher quality and within the cycle time.

However, this way of working may not be optimal in terms of utilization and travel costs and hence several other possibilities should be investigated. For example, FF DUV/YS Upgrades management wonders if planning by competence could lead to a better utilization of resources than planning by program. And here is where a mathematical tool that allows for the analysis of several scenarios, together with their comparison in terms of PIs, plays an important role.

2.2 Constraints

FF DUV/YS Upgrades manpower planning is a very complex problem. To date, planners have to manually schedule about 220-300 people while guaranteeing the observance of important constraints, as listed below

- skill level and training
- teams
- location and travel
- visa
- holidays
- rest period
- minimum work-life balance
- preparation phase

- wrap-up phase.

All these constraints are now described in more detail but not all of them are included in our mathematical model. Given the intricacy of the problem, we decided to focus only on the main constraints and to leave out those related to teams, visa, minimum work-life balance, preparation phase and wrap-up phase. Once a good algorithm for this model is constructed, it is easy to extend it so as to cover also the remaining constraints.

We already said that every machine upgrade is performed during the so-called execution phase and that it can last several weeks, depending on the upgrade program. To reduce the size of the FF DUV/YS Upgrades manpower planning problem, we decided to avoid the assignment of employees to upgrades in terms of days but to split the execution phase of every upgrade in atomic parts of one week each. These parts will be denoted from now on under the name of projects and the term "atomic" means that an employee can be assigned to a project only for its whole duration and not for just a few days. This choice is also suggested by additional observations from real life: an employee is never assigned to an upgrade for less than one week, CP plans upgrades in weeks and, moreover, the information about roles and manpower in every program is provided as input for the tool in terms of weeks.

The sets that capture the dimensions of the problem, together with the input parameters and the decision variables that are used throughout this chapter, are listed in Tables 2.1, 2.2 and 2.3 respectively. To improve the readability of the mathematical formulations, we adopt, with some exceptions, the convention of representing the indices through lowercase letters, the input parameters by lowercase symbols from the Greek alphabet and the decision variables by means of capital letters.

Sets:	
$\mathcal{T}$	set of all teams
$\mathcal{P}$	set of all people/employees to be scheduled
$\mathcal{P}^{\mathcal{T}}_{tm}$	set of people in team $tm \in \mathcal{T}$
$\mathcal{L}$	set of locations
$\mathcal{PJ}$	set of all projects, i.e. an indivisible subset of days of an upgrade, usually of length one week
$\mathcal{PJ}^{\mathcal{P}}_{p}$	set of projects in which employee $p \in \mathcal{P}$ can be involved
$\mathcal{PJ}^{\mathcal{T}}_{tm}$	set of projects in which team $tm \in \mathcal{T}$ can be involved
$\mathcal{R}$	set of all roles needed to execute the upgrades
$\mathcal{R}^{\mathcal{PJ}}_{prj}$	set of roles required in project $prj \in \mathcal{PJ}$

$\mathcal{R}^{\mathcal{PPJ}}_{p,prj}$	set of roles required in project $prj \in \mathcal{PJ}$ and for which person $p \in \mathcal{P}$ becomes skilled at the latest in the starting date of $prj \in \mathcal{PJ}$. The set is empty if $prj \notin \mathcal{PJ}^{\mathcal{P}}_p$
$\mathcal{TD}$	set of possible number of consecutive travel days, that is $\mathcal{TD} := \{0, 1, 2, 3\}$
$\mathcal{D}$	set of all days in the planning horizon
$\mathcal{D}^*_n$	set of days in the planning horizon, with the last $n \in \mathbb{N}$ days excluded
$\mathcal{D}^{\mathcal{PJ}}_{prj}$	set of days in which project $prj \in \mathcal{PJ}$ is performed
$\mathcal{D}^{\mathcal{IN}}_p$	set of days in which person $p \in \mathcal{P}$ is involved in some activity (either direct, indirect or personal holidays) specified in input to the tool
$\mathcal{AD}_p$	set of all arc days of person $p \in \mathcal{P}$ in the planning horizon

Table 2.1: Sets used in the FF DUV/YS Upgrades manpower planning model

Input Parameters:

$\gamma_{prj_1,prj_2} \in \{0, 1\}$	takes value 0 if two projects $prj_1 \in \mathcal{PJ}$ and $prj_2 \in \mathcal{PJ}$ overlap in at least one day and occur in the same location
$\alpha_{prj,r} \in \mathbb{N}$	is the number of people needed for project $prj \in \mathcal{PJ}$ in role $r \in \mathcal{R}^{\mathcal{PJ}}_{prj}$
$\delta_{prj} \in \mathbb{N}$	is the duration of project $prj \in \mathcal{PJ}$, in number of days
$\tau_{l_1,l_2} \in \mathcal{TD}$	is the number of travel days from location $l_1 \in \mathcal{L}$ to location $l_2 \in \mathcal{L}$
$\beta^{IN}_{p,d}$	is the input location of person $p \in \mathcal{P}$, that is the location given as input when $d \in \mathcal{D}^{\mathcal{IN}}_p$
β^{PJ}_{prj}	is the location where project $prj \in \mathcal{PJ}$ takes place
β^{H}_p	is the home/base location of person $p \in \mathcal{P}$

$\eta_p \in \mathbb{N}$	is the threshold in days that determines whether a period outside home location is short ($\leq \eta_p$) or long ($> \eta_p$) for employee $p \in \mathcal{P}$
$\rho_p^{SH} \in \mathbb{N}$	is the length of rest period, in days, for person $p \in \mathcal{P}$ after a short period outside his home location
$\rho_p^{LH} \in \mathbb{N}$	is the length of rest period, in days, for person $p \in \mathcal{P}$ after a long period outside his home location
$\omega_p \in \mathbb{N}$	is the maximum number of days abroad that person $p \in \mathcal{P}$ can have before going in rest period
$M \in \mathbb{N}$	is the high penalization coefficient used for dummy variables.

Table 2.2: Input parameters used in the FF DUV/YS Upgrades manpower planning model

Decision Variables:

$X^{\mathcal{PR}}_{p,prj,r} \in \mathbb{R}_{[0,1]}$	represents whether person $p \in \mathcal{P}$ is assigned to project $prj \in \mathcal{PJ}^{\mathcal{P}}_p$ for role $r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}$
$X^{\mathcal{P}}_{p,prj} \in \{0,1\}$	represents whether person $p \in \mathcal{P}$ is assigned to project $prj \in \mathcal{PJ}^{\mathcal{P}}_p$
$X^{\mathcal{T}}_{tm,prj} \in \{0,1\}$	represents whether team $tm \in \mathcal{T}$ is assigned to project $prj \in \mathcal{PJ}^{\mathcal{T}}_{tm}$
$L_{p,l,d} \in \{0,1\}$	takes value 1 if person $p \in \mathcal{P}$ is at location $l \in \mathcal{L}$ on day $d \in \mathcal{D}$
$T_{p,d} \in \{0,1\}$	takes value 1 if person $p \in \mathcal{P}$ is traveling on day $d \in \mathcal{D}$
$A^{SAB}_{p,d} \in \mathbb{R}_{[0,1]}$	represents a short abroad arc of length η_p for person $p \in \mathcal{P}$ that starts on day $d \in \mathcal{D}$
$A^{LAB}_{p,d} \in \mathbb{R}_{[0,1]}$	represents a long abroad arc of length ω_p for person $p \in \mathcal{P}$ that starts on day $d \in \mathcal{D}$
$A^{USAB}_{p,d} \in \mathbb{R}_{\geq 0}$	represents a unit backward short abroad arc for person $p \in \mathcal{P}$ on day $d \in \mathcal{D}$

$A^{ULAB}_{p,d} \in \mathbb{R}_{\geq 0}$	represents a unit backward long abroad arc for person $p \in \mathcal{P}$ on day $d \in \mathcal{D}$
$A^{SH}_{p,d} \in \mathbb{R}_{[0,1]}$	represents a short home arc of length ρ^{SH}_p for person $p \in \mathcal{P}$ that starts on day $d \in \mathcal{D}$
$A^{LH}_{p,d} \in \mathbb{R}_{[0,1]}$	represents a long home arc of length ρ^{LH}_p for person $p \in \mathcal{P}$ that starts on day $d \in \mathcal{D}$
$A^{UH}_{p,d} \in \mathbb{R}_{[0,1]}$	represents a unit home arc for person $p \in \mathcal{P}$ on day $d \in \mathcal{D}$
$D^{1}_{prj,r} \in \mathbb{R}_{\geq 0}$	are dummy variables used to guarantee that the formulation of the original problem is always feasible

Table 2.3: Decision variables used in the FF DUV/YS Upgrades manpower planning model

Skill Level and Training

Every system upgrade needs a certain amount of skilled people. A first distinction of functions is between upgrade engineers, coordinators, generalists and material handlers but then there is a further subdivision according to the roles for which each employee is skilled. Moreover, employees can be involved in training programs after which they become skilled for new roles. The training periods are preassigned for each employee and cannot be changed in the optimization model, hence the training days are given in input together with other activities through the set $\mathcal{D}^{\mathcal{IN}}_p$.
Since an employee $p \in \mathcal{P}$ can be busy in some days with some activity given as input, then in these days he cannot be assigned to any project. Hence, let $\mathcal{PJ}^{\mathcal{P}}_p$ be the set of projects that $p \in \mathcal{P}$ can be assigned to and let $\mathcal{R}^{\mathcal{PJ}}_{prj}$ be the set of roles needed in project $prj \in \mathcal{PJ}$. If we denote by $X^{\mathcal{PR}}_{p,prj,r}$ the variable that defines whether employee p is assigned to project prj for role r, this variable can take value 1 only if p becomes skilled for role r at the latest in the starting day of the project. Hence, for every $p \in \mathcal{P}$ and $prj \in \mathcal{PJ}^{\mathcal{P}}_p$, we define the set $\mathcal{R}^{\mathcal{PPJ}}_{p,prj}$ of all roles $r \in \mathcal{R}^{\mathcal{PJ}}_{prj}$ for which person p is skilled on time.

If the variable $X^{\mathcal{P}}_{p,prj}$ denotes whether employee $p \in \mathcal{P}$ is assigned to project $prj \in \mathcal{PJ}^{\mathcal{P}}_p$, then it is related to $X^{\mathcal{PR}}_{p,prj,r}$ by the identity

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}^{\mathcal{P}}_p :$

$$X^{\mathcal{P}}_{p,prj} = \sum_{r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}} X^{\mathcal{PR}}_{p,prj,r}. \tag{2.1}$$

Let $\alpha_{prj,r}$ be the number of people needed in project $prj \in \mathcal{PJ}$ for role $r \in \mathcal{R}^{\mathcal{PJ}}_{prj}$. The allocation of skilled people to projects is then guaranteed by requesting that

$\forall prj \in \mathcal{PJ}, r \in \mathcal{R}^{\mathcal{PJ}}_{prj}$:

$$\sum_{\substack{p\in\mathcal{P}/\\ r\in\mathcal{R}^{\mathcal{PPJ}}_{p,prj}}} X^{\mathcal{PR}}_{p,prj,r} \geq \alpha_{prj,r}, \tag{2.2}$$

where inequality is used in place of equality because the presence of teams could lead to more manpower allocation than is strictly necessary.

If two projects overlap in some days or if there is no enough room for travel between their locations, then no employee can be assigned to both. The location and travel constraints already make it impossible for these situations to occur, except for the case of two overlapping projects having the same location. If we denote by γ_{prj_1,prj_2} the boolean parameter that takes value 0 for pairs of overlapping projects in a same location, we should ask for

$\forall p \in \mathcal{P}, prj_1, prj_2 \in \mathcal{PJ}^{\mathcal{P}}_p$ *s.t.* $\gamma_{prj_1,prj_2} = 0, prj_1 < prj_2$:

$$X^{\mathcal{P}}_{p,prj_1} + X^{\mathcal{P}}_{p,prj_2} \leq 1. \tag{2.3}$$

Note that by $prj_1 < prj_2$ we implicitly assume that an order relation has been defined over the elements of the set $\mathcal{PJ}$.

Teams

The concept of teams is very important because FF DUV/YS Upgrades department would like to analyze several scenarios and this is strictly related to the composition of teams. Let $\mathcal{P}^{\mathcal{T}}_{tm}$ and $\mathcal{PJ}^{\mathcal{T}}_{tm}$ be the set of people in team $tm \in \mathcal{T}$ and the set of projects in which tm can be involved, respectively. If $X^{\mathcal{T}}_{tm,prj}$ is the boolean variable for the assignment of teams to projects, we could request for example that

$\forall tm \in \mathcal{T}, prj \in \mathcal{PJ}^{\mathcal{T}}_{tm}$:

$$\sum_{p\in\mathcal{P}^{\mathcal{T}}_{tm}} X^{\mathcal{P}}_{p,prj} = |\mathcal{P}^{\mathcal{T}}_{tm}| \cdot X^{\mathcal{T}}_{tm,prj} \tag{Teams}$$

where $|\mathcal{P}^{\mathcal{T}}_{tm}|$ is here used to indicate the cardinality of the set $\mathcal{P}^{\mathcal{T}}_{tm}$. Through this constraint we assume that people in a same team should always work together on the same projects. If this constraint is too restrictive and people in a same team could also do sometimes different upgrades if necessary, then, depending on which scenarios are interesting for FF DUV/YS Upgrades department, a more appropriate constraint could be easily constructed. Anyway, since the formulation of team constraints is a matter of FF management choice, in our model we decided to leave it out because it is more suitable to incorporate it in a second moment, once a mathematical tool is ready to perform scenario analysis.

Location and Travel

The mathematical tool should accomplish two main tasks, that is assign people to projects and determine for every employee an agenda specifying his daily location. Indeed, even though the final goal is the manpower allocation to projects, the second task cannot be avoided because some constraints, such as travel or rest period, the PIs and consequently the objective function depend on the employees' daily locations.

We denote as $L_{p,l,d}$ the variable taking value 1 if $p \in \mathcal{P}$ is at location $l \in \mathcal{L}$ on day $d \in \mathcal{D}$ and whose array $L^{(p)} := (L_{p,l,d})_{l \in \mathcal{L}, d \in \mathcal{D}}$ characterizes employee p's agenda. For every project $prj \in \mathcal{PJ}$, let $\mathcal{D}^{\mathcal{PJ}}_{prj}$ be the set of days in which it is performed and let $\delta_{prj} := |\mathcal{D}^{\mathcal{PJ}}_{prj}|$ the number of days needed for it (in general, $\delta_{prj} = 7$). When a person is working on a project, he should be at project location because projects are parts of the execution phase, that is

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}^{\mathcal{P}}_p$:

$$\sum_{d \in \mathcal{D}^{\mathcal{PJ}}_{prj}} L_{p,\beta^{PJ}_{prj},d} \geq \delta_{prj} \cdot X^{\mathcal{P}}_{p,prj}. \tag{2.4}$$

If the variable $T_{p,d}$ indicates a travel day, then by imposing that

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$\sum_{l \in \mathcal{L}} L_{p,l,d} + T_{p,d} = 1, \tag{2.5}$$

it is guaranteed that every day employee $p \in \mathcal{P}$ is either in a location or traveling.

For every employee $p \in \mathcal{P}$ the tool receives in input a set $\mathcal{D}^{\mathcal{IN}}_p$ of days in which some activity (either direct, indirect or personal holidays) is already planned for p, together with the locations $\beta^{IN}_{p,d}$ of p in those days $d \in \mathcal{D}^{\mathcal{IN}}_p$. The data given in input cannot be changed by the tool and so those input locations are fixed in the agenda by requesting that

$\forall p \in \mathcal{P}, d \in \mathcal{D}^{\mathcal{IN}}_p$:

$$L_{p,\beta^{IN}_{p,d},d} = 1. \tag{2.6}$$

As far as travel is concerned, every time a person's location changes, in the previous days he must travel to that location. Written out in mathematical terms, this condition becomes

$\forall p \in \mathcal{P}, l_1 \in \mathcal{L}, td \in \mathcal{TD}, d \in \mathcal{D}^*_{td}$ $s.t.$ $td \geq 1$:

$$\sum_{\substack{l_2 \in \mathcal{L}/ \\ \tau_{l_1,l_2} < td}} L_{p,l_2,d+td} + T_{p,d+td} \geq L_{p,l_1,d} \tag{2.7}$$

where τ_{l_1,l_2} is the number of consecutive travel days that are needed to move from location l_1 to location l_2, $\mathcal{TD}$ is the set of possible numbers of travel days and $\mathcal{D}^*_{td}$ is the set of those days in the planning horizon that do not fall among the last td days. It is interesting to observe that the travel constraints are formulated by looking at employee p' s location at day $d \in \mathcal{D}$ and by wondering how far he is from that location after td days. Indeed, even though it could be more intuitive to explicitly ask for the right number of travel days every time there is a change from a location $l_1 \in \mathcal{L}$ to a new location $l_2 \in \mathcal{L}$, that modeling choice would lead to many more constraints, resulting in an unnecessary increase in complexity.

Visa

When people travel abroad, they should have a particular visa, depending on the aim of the travel. A list of factors that are usually involved in the visa process is the following:

- for every person and location of destination there could be several types of visa that person can apply for and they can differ not only in their validity period but also in the processing time that is needed to get them
- a visa can be in a precedence relation with others, that is it can only be requested before other types of visas, in each calendar year
- some visas can be renewed up to a maximum number of times or they cannot be renewed at all
- in some countries the application for a renewal can be done even before the current visa expires, while in other countries the employee should leave that country before being able to reapply for a visa or, however, he should wait for some weeks before going back to that country
- during the processing time for a visa, the employee cannot travel because he is without his passport for that period (so he can only work at his home location)
- during the processing time for a visa, the employee cannot apply for another visa in the same location of destination
- some roles can only be performed by local manpower because of visa problems (e.g. material handlers in US).

All these restrictions make visa constraint an important and challenging factor for manpower planning that cannot be dropped. However, in reality it is not possible to perfectly model all the previous points, because:

- most information is not available to FF, but should be provided by the immigration office IMO of ASML
- the input data required would take a lot of effort
- visa regulations change very quickly in time
- the size of the model would become very big.

Since a simplification is needed, a possible solution could be to integrate the visa constraint later in the model by adding some restrictions on the projects that employees can be assigned to. This can be easily performed by the tool in preprocessing, when the set $\mathcal{PJ}_p^{\mathcal{P}}$ is constructed. Not only is this implementation straightforward, but it also allows to easily rerun a scenario whenever planners should discover that an employee actually cannot be assigned to an upgrade because of new unexpected issues. However, a more detailed incorporation through additional constraints is also possible.

Holidays

Employees have two different types of holidays: public holidays, that vary depending on their home location, and personal holidays (sick, vacations, etc...). Despite personal holidays, which are included in the set $\mathcal{D}_p^{\mathcal{IN}}$, public holidays are not guaranteed to the employees: during public holidays an employee can also be assigned to a direct or indirect work or travel. That is why a public holiday should be considered as such only if no work activity is assigned. This means that public holidays do not represent a constraint to our model, they can simply be added in a post-processing phase whenever the employees are not allocated to anything else during the optimization phase.

Rest Period

Employees should always alternate a work period outside their home location with a rest period. Unlike what the name suggests, during rest period the employee keeps working but in his base location. This means that he can be assigned to any direct or indirect activity taking place at his home location or he can even take some holidays.

Generally, a rest period of 1 week follows a work period abroad of at most 2 weeks while for more work days abroad a rest period of 2 weeks is assigned, where from now on by "abroad" we mean every location different from the home location. However these values strictly depend on employees, thus the input parameters ρ_p^{LH} and ρ_p^{SH} specify for each person $p \in \mathcal{P}$ the number of days at rest he wants to have according on whether the work days he spends outside his base location are more than the threshold parameter η_p or not, respectively. It should be pointed out that after his rest period the employee can keep staying at his base location, he is not forced to go abroad. Moreover, after an upgrade abroad a person can also be assigned to another upgrade before starting his period of rest. However, the number of work days far away from the base location should not exceed a certain amount of days (usually from 4 to 6 weeks), expressed by the parameter ω_p.

The request for rest period after a period abroad is modeled using forward and backward arcs and imposing some flow-balance equations. This choice is suggested by the high number of variables and constraints that otherwise would be needed to keep track of all the possible cases of days abroad and days at home location.

Thanks to the concept of arcs, only 7 symbolic variables need to be defined instead, each one representing a different type of flow arc:

- the variable $A_{p,d}^{SAB}$ represents a short abroad arc and is a forward arc of length η_p days

- the variable $A^{LAB}_{p,d}$ represents a long abroad arc and is a forward arc of length ω_p days
- the variable $A^{USAB}_{p,d}$ represents a unit short abroad arc and is a backward arc of length 1 day
- the variable $A^{ULAB}_{p,d}$ represents a unit long abroad arc and is a backward arc of length 1 day
- the variable $A^{SH}_{p,d}$ represents a short home arc and is a forward arc of length ρ^{SH}_p days
- the variable $A^{LH}_{p,d}$ represents a long home arc and is a forward arc of length ρ^{LH}_p days
- the variable $A^{UH}_{p,d}$ represents a unit home arc and is a forward arc of length 1 day.

The idea is that every time on day $d \in \mathcal{D}$ an employee $p \in \mathcal{P}$ starts a long or short period abroad, the variable $A^{SAB}_{p,d}$ or $A^{LAB}_{p,d}$ takes value 1, respectively. This means that an arc of the corresponding length starts from day d and during its whole length person p is assumed to be abroad. However, since employee p can choose to stay abroad also for a shorter period, backwards arcs are needed in order to reduce the length of the abroad forward arc. That is, backward arcs $A^{USAB}_{p,d}$ and $A^{ULAB}_{p,d}$ are used in case an employee stays abroad for a shorter period than η_p or ω_p days. On the other hand, if the period abroad ends on day d, a short or a long period at home location should start, which means that the variable $A^{SH}_{p,d}$ or $A^{LH}_{p,d}$ should take value 1 respectively. Finally, after his rest period employee p can keep staying at his base location and this can be modeled by using a suitable number of consecutive unit home arcs.
Despite the previous interpretation, it should be pointed out that it is not necessary to define the arc variables in our model as integer variables; thanks to a flow formulation they already behave well in their relaxed form.

In mathematical terms, a person is at home location on day $d \in \mathcal{D}$ if and only if that day is covered by a home arc, which means that

$\forall p \in \mathcal{P}, d \in \mathcal{D}:$

$$L_{p,\beta^H_p,d} = A^{UH}_{p,d} + \sum_{\substack{\tilde{d}\in\mathcal{D}/\\ \tilde{d}\geq d-\rho^{SH}_p+1,\\ \tilde{d}\leq d}} A^{SH}_{p,\tilde{d}} + \sum_{\substack{\tilde{d}\in\mathcal{D}/\\ \tilde{d}\geq d-\rho^{LH}_p+1,\\ \tilde{d}\leq d}} A^{LH}_{p,\tilde{d}}. \tag{2.8}$$

Moreover, when an abroad arc starts, the employee is not at home location anymore, that is

$\forall p \in \mathcal{P}, d \in \mathcal{D}:$

$$1 - L_{p,\beta^H_p,d} \geq A^{SAB}_{p,d} + A^{LAB}_{p,d}. \tag{2.9}$$

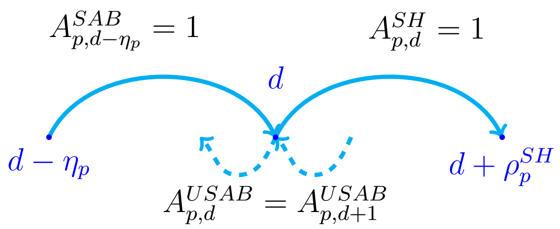
Finally, 3 balance equations are needed so that, every day, the incoming flow is equal to the outgoing flow. More precisely, at every point in time $d \in \mathcal{D}$:

- a short abroad arc must be followed by either a unit backward short abroad arc or a short home arc

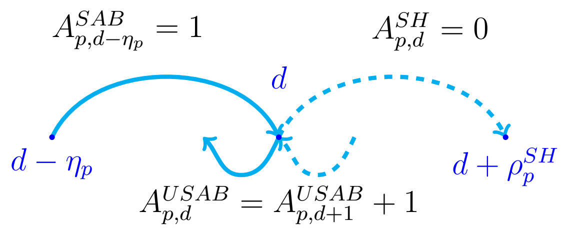
- a long abroad arc must be followed by either a unit backward long abroad arc or a long home arc

- a home arc must be followed by either a unit home arc or an abroad arc.

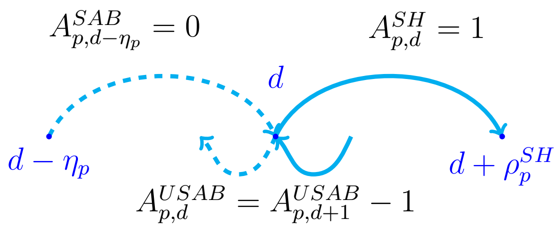
The balance equation for short period abroad is graphically shown in Fig. 2.1 (the balance equation for long period abroad is specular) and the one for home period is presented in Fig. 2.2.



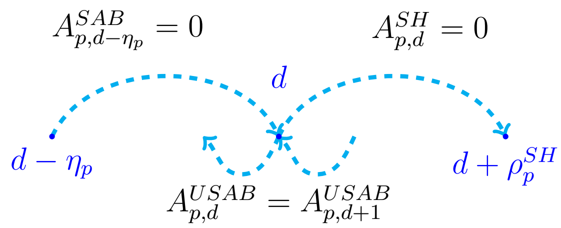
(a) case 1



(b) case 2

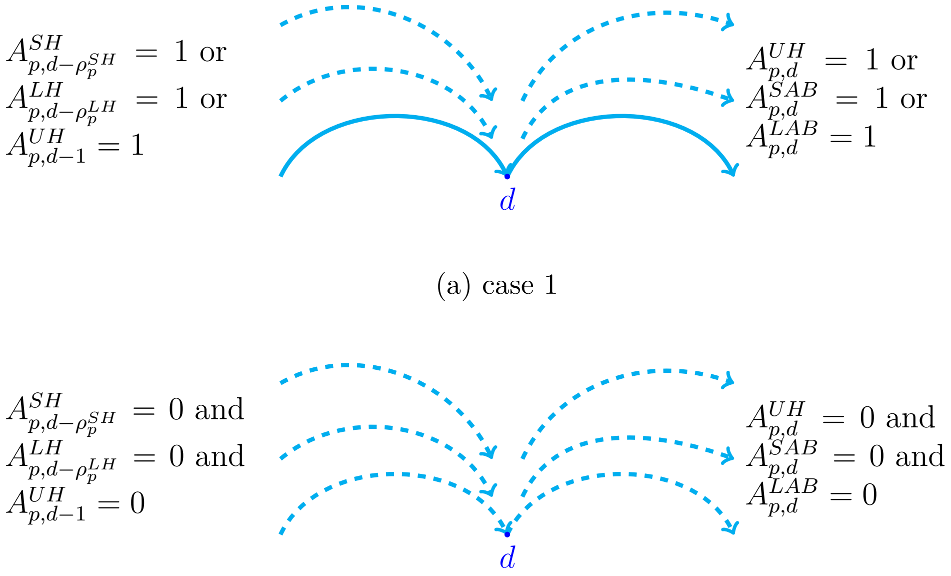


(c) case 3



(d) case 4

Figure 2.1: Graphical representation of the balance equation for short period abroad



(a) case 1

(b) case 2

Figure 2.2: Graphical representation of the balance equation for home period

Written out as mathematical constraints, the balance equations become respectively

$\forall p \in \mathcal{P}, d \in \mathcal{D}:$

$$A^{SAB}_{p,d-\eta_p} + A^{USAB}_{p,d+1} = A^{USAB}_{p,d} + A^{SH}_{p,d} \tag{2.10}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}:$

$$A^{LAB}_{p,d-\omega_p} + A^{ULAB}_{p,d+1} = A^{ULAB}_{p,d} + A^{LH}_{p,d} \tag{2.11}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}:$

$$A^{SH}_{p,d-\rho^{SH}_p} + A^{LH}_{p,d-\rho^{LH}_p} + A^{UH}_{p,d-1} = A^{UH}_{p,d} + A^{SAB}_{p,d} + A^{LAB}_{p,d}. \tag{2.12}$$

Regarding the previous equations, a clarification is needed about our abuse of notation. If flow arcs were defined over $p \in \mathcal{P}$ and $d \in \mathcal{D}$ then it could happen, in the given formulation, that $d - \rho^{SH}_p \notin \mathcal{D}$ and the same would hold for $d - \rho^{LH}_p$, $d-1$, $d-\eta_p$, $d-\omega_p$ and $d+1$. This suggests that, in reality, the second index in the flow variables does not take values in the set $\mathcal{D}$ of days, but it is defined upon a set $\mathcal{AD}_p$ of arc days that contains the days in $\mathcal{D}$, the first ω_p days immediately before the first day of $\mathcal{D}$ and, for reasons that will be more clear in the next constraints, the day immediately after the last day of $\mathcal{D}$. However the flow arcs having the second index with a value before the days in $\mathcal{D}$ are

not variables, but should be seen as parameters specified somehow in input and that are used as initialization in the case in which the schedule computed by the mathematical tool does not start from a blank sheet but has the days before $\mathcal{D}$ already planned.
In the simpler case of a schedule that should be planned without any consideration of the days before the tool horizon, we can imagine that all those flow arcs take value 0 except for the unit home arc of the day before the first day of the tool horizon, which should take value 1 in order to guarantee the soundness of the last balance equation.

Since the tool considers a finite planning horizon, additional care should be taken to ensure that all the abroad arcs falling outside this horizon will activate the right backward arcs, if any. This particular situation is tackled by adding a new day after the last day of the planning horizon (i.e. after the last day in $\mathcal{D}$), denoted as $\hat{d}_{Out}$, and forcing all arcs falling outside the horizon towards this new day. In this way, imposing the flow-balance equations for short and long period abroad also in $\hat{d}_{Out}$ ensures that the backward abroad arcs are linked to the correct forward arcs.
The mathematical constraints on day $\hat{d}_{Out}$ assume the form

$\forall p \in \mathcal{P}$:

$$\sum_{\substack{\tilde{d} \in \mathcal{D} / \\ \tilde{d} \geq \hat{d}_{Out} - \eta_p, \\ \tilde{d} \leq \hat{d}_{Out} - 1}} A^{SAB}_{p,\tilde{d}} = A^{USAB}_{p,\hat{d}_{Out}} + A^{SH}_{p,\hat{d}_{Out}} \tag{2.13}$$

and

$\forall p \in \mathcal{P}$:

$$\sum_{\substack{\tilde{d} \in \mathcal{D} / \\ \tilde{d} \geq \hat{d}_{Out} - \omega_p, \\ \tilde{d} \leq \hat{d}_{Out} - 1}} A^{LAB}_{p,\tilde{d}} = A^{ULAB}_{p,\hat{d}_{Out}} + A^{LH}_{p,\hat{d}_{Out}}. \tag{2.14}$$

Minimum Work-Life Balance

The manpower schedule should guarantee a minimum level of work-life balance (WLB) that is specified by employment contract. WLB is one of the main Performance Indicators (PIs) for FF and it is defined for every person as the percentage of days at his home location. In general, our tool optimizes the overall work-life balance and the constraints modeling the alternation between a work period outside home location and a period at home are also related to it but, without an explicit constraint, it is not guaranteed that the optimal schedule will satisfy the minimum WLB threshold for every employee. Nevertheless, we decided to avoid a further complication of the model by ignoring for now this detail. In the future it could be explicitly incorporate as constraint, so as to reject every optimal solution that does not guarantee the minimum work-life balance.

Preparation Phase

The preparation phase is an important step taking place before the execution of the upgrade. According to FF management, an ideal planning tool should wisely assign a preparation phase for every upgrade because a good preparation brings about a more effective execution. However, in general no exact rule exists in the assignment of preparation days. More precisely, once a coordinator is assigned to the preparation phase of an upgrade he can work from his home location and he has lots of flexibility in organizing his workload. For example, in the first couple of weeks he needs to spend only some hours (about 10%), while in the last weeks before the execution phase he must devote more time on that, about 70-80% depending on his experience. This means that during that period a coordinator can also be involved for a few days in other activities such as trainings or holidays. That's why in the current planning sheet the preparation phase is not always explicitly scheduled. Only for TWINSCAN upgrades the situation is slightly different because a rotational approach is adopted: every four weeks a new coordinator stays at home location and is designated for all the TWINSCAN preparation phases of that period. We do not handle this constraint in our model but, for a later incorporation, more insight is needed in order to define some rules that can be implemented in the tool. In this way, this phase can be modeled in terms of additional projects that have different requirements from the projects of the execution phase.

Wrap-Up Phase

Wrap-up phase occurs during the two weeks immediately after the execution phase. In this phase, all the service orders are closed and the final checks are made. The coordinators are in charge of it and it never takes all the two weeks; some days or 1 week are usually enough. The small amount of days and people involved suggested us to ignore the incorporation of this phase in our model, postponing its incorporation to a later stage.

2.3 Performance Indicators

A formal definition of Performance Indicators (PIs) for FF DUV/YS Upgrades manpower planning is fundamental. Indeed, PIs are needed not only for a tool in order to compare different scenarios, but also for FF managers and planners in order to get more insight into the current way of working and the quality of the current planning. Since no PIs have ever been defined, during our meetings with FF management and planners the following 5 new PIs have been identified as the main target for the planning activities:

- Utilization
- Number of Travels
- Local Use
- Travel Waste
- Work-Life Balance.

A short description follows and, for more details, a summary is provided by the tables in App. D.

Utilization

Utilization is defined for every person as the percentage of days of direct work over all his days of work (both direct and indirect). This PI is useful in order to measure the effective use of resources.

Number of Travels

Number of Travels is self-explanatory and it is useful as indicator for excess travel costs. Travels are differentiated in local travels, regional travels that are not local and interregional travels. Whenever travel cannot be avoided, local travel is preferable, followed by regional travel.

Local Use

Local Use is defined upon every upgrade and denotes the percentage of people assigned to that upgrade and working locally. The term “locally” can mean either at home location or within the home country. This PI measures how well the local staffing strategy is executed.

Travel Waste

Travel Waste in a given time horizon is defined as the minimum between the number of people going outside their home country and the number of people entering in that country in the same time period. It is meaningful especially when filtered by country and function because it investigates not only travel costs but also staffing, planning, training and cross utilization strategy.

Work-Life Balance

Work-Life Balance (WLB) represents the percentage of work days that an employee spends close to his home location. Here “close” can mean either just the home location or also the home country. Although the above definition does not express thoroughly the concept of WLB because a person working many hours per day at his home location should have a low WLB, the goal of this PI is to get a first important insight into the work-life balance of every employee.

2.4 Objective Function

The objective function should be chosen in a way that reflects what the PIs ask for.
As the PIs suggest, the main goal in planning FF DUV/YS manpower is to maximize the local use of resources and, whenever it is not possible for an employee to work at his home location, to minimize the number of travel days.
It is important to express both these aspects in the objective function. For instance,

- an objective function defined as the number of days spent by employees at their home location would make no difference in which upgrade they are assigned when they are not at their home location, possibly affecting Local Use and Number of Travels,

- an objective function representing only the total number of travel days would make no difference in the location of an employee as long as he does not travel, possibly affecting in this way WLB.

Therefore, we decided to summarize the PIs through the minimization of the linear objective function

$$Obj := \sum_{\substack{p \in \mathcal{P},\\ d \in \mathcal{D}}} \left(T_{p,d} + \left(1 - L_{p,\beta_p^H,d} \right) \right).$$

This cost function penalizes every day an employee is not at his home location and penalizes twice every travel day. Indeed, a travel day means that an employee is not at home location and at the same time it causes higher travel costs for the company.

2.5 Model Summary

Some dummy variables $D^1_{prj,r}$ are added in the final formulation of the original problem. By setting a high penalization (big-M) coefficient in the objective function, these dummies are used only if the original problem is infeasible and allow us to easily check in which projects and for which roles there is a deficit of manpower. Even though in this thesis capital letters are used for variables, we decided to keep the standard convention of using the M symbol for the penalization parameter.

In summary, here is the final formulation of the original problem:

Minimize:

$$M \cdot \sum_{\substack{prj \in \mathcal{PJ},\\ r \in \mathcal{R}^{\mathcal{PJ}}_{prj}}} D^1_{prj,r} + \sum_{\substack{p \in \mathcal{P},\\ d \in \mathcal{D}}} \left(T_{p,d} + \left(1 - L_{p,\beta_p^H,d} \right) \right)$$

Subject to:

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}^{\mathcal{P}}_p$:

$$X^{\mathcal{P}}_{p,prj} = \sum_{r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}} X^{\mathcal{PR}}_{p,prj,r} \tag{2.1}$$

$\forall prj \in \mathcal{PJ}, r \in \mathcal{R}^{\mathcal{PJ}}_{prj}$:

$$D^1_{prj,r} + \sum_{\substack{p \in \mathcal{P}/\\ r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}}} X^{\mathcal{PR}}_{p,prj,r} \geq \alpha_{prj,r} \tag{2.2}$$

$\forall p \in \mathcal{P}, prj_1, prj_2 \in \mathcal{PJ}^{\mathcal{P}}_p$ $s.t.$ $\gamma_{prj_1,prj_2} = 0, prj_1 < prj_2$:

$$X^{\mathcal{P}}_{p,prj_1} + X^{\mathcal{P}}_{p,prj_2} \leq 1 \tag{2.3}$$

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}_p^{\mathcal{P}}$:

$$\sum_{d \in \mathcal{D}_{prj}^{\mathcal{PJ}}} L_{p,\beta_{prj}^{PJ},d} \geq \delta_{prj} \cdot X_{p,prj}^{\mathcal{P}} \tag{2.4}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$\sum_{l \in \mathcal{L}} L_{p,l,d} + T_{p,d} = 1 \tag{2.5}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}_p^{\mathcal{IN}}$:

$$L_{p,\beta_{p,d}^{IN},d} = 1 \tag{2.6}$$

$\forall p \in \mathcal{P}, l_1 \in \mathcal{L}, td \in \mathcal{TD}, d \in \mathcal{D}_{td}^{*}$ $s.t.$ $td \geq 1$:

$$\sum_{\substack{l_2 \in \mathcal{L} / \\ \tau_{l_1,l_2} < td}} L_{p,l_2,d+td} + T_{p,d+td} \geq L_{p,l_1,d} \tag{2.7}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$L_{p,\beta_p^H,d} = A_{p,d}^{UH} + \sum_{\substack{\tilde{d} \in \mathcal{D} / \\ \tilde{d} \geq d - \rho_p^{SH} + 1, \\ \tilde{d} \leq d}} A_{p,\tilde{d}}^{SH} + \sum_{\substack{\tilde{d} \in \mathcal{D} / \\ \tilde{d} \geq d - \rho_p^{LH} + 1, \\ \tilde{d} \leq d}} A_{p,\tilde{d}}^{LH} \tag{2.8}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$1 - L_{p,\beta_p^H,d} \geq A_{p,d}^{SAB} + A_{p,d}^{LAB} \tag{2.9}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$A_{p,d-\eta_p}^{SAB} + A_{p,d+1}^{USAB} = A_{p,d}^{USAB} + A_{p,d}^{SH} \tag{2.10}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$A_{p,d-\omega_p}^{LAB} + A_{p,d+1}^{ULAB} = A_{p,d}^{ULAB} + A_{p,d}^{LH} \tag{2.11}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$A_{p,d-\rho_p^{SH}}^{SH} + A_{p,d-\rho_p^{LH}}^{LH} + A_{p,d-1}^{UH} = A_{p,d}^{UH} + A_{p,d}^{SAB} + A_{p,d}^{LAB} \tag{2.12}$$

$\forall p \in \mathcal{P}$:

$$\sum_{\substack{\tilde{d} \in \mathcal{D} / \\ \tilde{d} \geq \hat{d}_{Out} - \eta_p, \\ \tilde{d} \leq \hat{d}_{Out} - 1}} A_{p,\tilde{d}}^{SAB} = A_{p,\hat{d}_{Out}}^{USAB} + A_{p,\hat{d}_{Out}}^{SH} \tag{2.13}$$

$\forall p \in \mathcal{P}$:

$$\sum_{\substack{\tilde{d} \in \mathcal{D}/ \\ \tilde{d} \geq \hat{d}_{Out} - \omega_p, \\ \tilde{d} \leq \hat{d}_{Out} - 1}} A^{LAB}_{p,\tilde{d}} = A^{ULAB}_{p,\hat{d}_{Out}} + A^{LH}_{p,\hat{d}_{Out}} \tag{2.14}$$

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}^{\mathcal{P}}_{p}, r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}$:

$$X^{\mathcal{PR}}_{p,prj,r} \in \mathbb{R}_{[0,1]} \tag{2.15}$$

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}^{\mathcal{P}}_{p}$:

$$X^{\mathcal{P}}_{p,prj} \in \{0, 1\} \tag{2.16}$$

$\forall p \in \mathcal{P}, l \in \mathcal{L}, d \in \mathcal{D}$:

$$L_{p,l,d} \in \{0, 1\} \tag{2.17}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$T_{p,d} \in \{0, 1\} \tag{2.18}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$A^{SAB}_{p,d} \in \mathbb{R}_{[0,1]} \tag{2.19}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$A^{LAB}_{p,d} \in \mathbb{R}_{[0,1]} \tag{2.20}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$A^{USAB}_{p,d} \in \mathbb{R}_{\geq 0} \tag{2.21}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$A^{ULAB}_{p,d} \in \mathbb{R}_{\geq 0} \tag{2.22}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$A^{SH}_{p,d} \in \mathbb{R}_{[0,1]} \tag{2.23}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$A^{LH}_{p,d} \in \mathbb{R}_{[0,1]} \tag{2.24}$$

$\forall p \in \mathcal{P}, d \in \mathcal{D}$:

$$A^{UH}_{p,d} \in \mathbb{R}_{[0,1]} \tag{2.25}$$

$\forall prj \in \mathcal{PJ}, r \in \mathcal{R}^{\mathcal{PJ}}_{prj}$:

$$D^{1}_{prj,r} \in \mathbb{R}_{\geq 0}. \tag{2.26}$$

Chapter 3

Solution Methodology

The model developed in the previous chapter is a mixed integer linear program whose compact formulation, due to its complexity, cannot be efficiently solved for large scale instances. This is an important issue because FF DUV/YS Upgrades department has recently been increasing in both manpower and volume of upgrades to be performed. Actually, for a 3-month horizon the compact formulation already results to be intractable for instances with little manpower, like depicted by Table 3.5 in the last section of this chapter, dedicated to the experimental results. In particular, a deeper investigation brought to the conclusion that travel and rest period constraints are the main responsible for the big constraints size in the model.

This issue drove us to adopt a decomposition strategy, a powerful approach that has already been successfully applied in similar problems (see [6] and [9] for some examples). For this purpose, it is useful to remind that the optimization model described in the previous chapter accomplishes two main tasks. The first task is the manpower allocation to projects, while the second task consists in the delineation of an agenda for each employee with his daily location. As highlighted in Fig. 3.1, the model is characterized by a block structure: all specifications of variables domain aside, constraints (2.1)-(2.3) determine a block involving manpower allocation variables $X^{\mathcal{P}}_{p,prj}$, while constraints (2.5)-(2.14) determine $|\mathcal{P}|$ independent blocks involving agenda variables $L_{p,l,d}$, namely one block for every employee $p \in \mathcal{P}$. Indeed, all the constraints related to the location variables have p in their index domain (as shown in the figure by coloring p in blue), meaning that we can split them in blocks according to $p \in \mathcal{P}$. Finally, the inequalities in (2.4) play the role of linking constraints between these blocks because they involve both the variables $X^{\mathcal{P}}_{p,prj}$ and $L_{p,l,d}$.

Constraints (without variables domain)		Indices	
$X^{\mathcal{P}}_{p,prj} - \sum_{r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}} X^{\mathcal{PR}}_{p,prj,r}$	$= 0$	p, prj	(2.1)
$D^1_{prj,r} + \sum_{p \in \mathcal{P} / r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}} X^{\mathcal{PR}}_{p,prj,r}$	$\geq \alpha_{prj,r}$	prj, r	(2.2)
$X^{\mathcal{P}}_{p,prj_1} + X^{\mathcal{P}}_{p,prj_2}$	≤ 1	p, prj_1, prj_2	(2.3)
$\sum_{d \in \mathcal{D}^{\mathcal{PJ}}_{prj}} L_{p,\beta^{PJ}_{prj},d} - \delta_{prj} \cdot X^{\mathcal{P}}_{p,prj}$	≥ 0	p, prj	(2.4)
$\sum_{l \in \mathcal{L}} L_{p,l,d} + T_{p,d}$	$= 1$	p, d	(2.5)
$L_{p,\beta^{IN}_{p,d},d}$	$= 1$	p, d	(2.6)
$\sum_{l_2 \in \mathcal{L} / \tau_{l_1,l_2} < td} L_{p,l_2,d+td} + T_{p,d+td} - L_{p,l_1,d}$	≥ 0	p, l_1, td, d	(2.7)
$L_{p,\beta^H_p,d} - A^{UH}_{p,d} - \sum_{\tilde{d} \in \mathcal{D} / \tilde{d} \geq d - \rho^{SH}_p + 1, \tilde{d} \leq d} A^{SH}_{p,\tilde{d}} - \sum_{\tilde{d} \in \mathcal{D} / \tilde{d} \geq d - \rho^{LH}_p + 1, \tilde{d} \leq d} A^{LH}_{p,\tilde{d}}$	$= 0$	p, d	(2.8)
$1 - L_{p,\beta^H_p,d} - A^{SAB}_{p,d} - A^{LAB}_{p,d}$	≥ 0	p, d	(2.9)
$A^{SAB}_{p,d-\eta_p} + A^{USAB}_{p,d+1} - A^{USAB}_{p,d} - A^{SH}_{p,d}$	$= 0$	p, d	(2.10)
$A^{LAB}_{p,d-\omega_p} + A^{ULAB}_{p,d+1} - A^{ULAB}_{p,d} - A^{LH}_{p,d}$	$= 0$	p, d	(2.11)
$A^{SH}_{p,d-\rho^{SH}_p} + A^{LH}_{p,d-\rho^{LH}_p} + A^{UH}_{p,d-1} - A^{UH}_{p,d} - A^{SAB}_{p,d} - A^{LAB}_{p,d}$	$= 0$	p, d	(2.12)
$\sum_{\tilde{d} \in \mathcal{D} / \tilde{d} \geq \hat{d}_{Out} - \eta_p, \tilde{d} \leq \hat{d}_{Out} - 1} A^{SAB}_{p,\tilde{d}} - A^{USAB}_{p,\hat{d}_{Out}} - A^{SH}_{p,\hat{d}_{Out}}$	$= 0$	p	(2.13)
$\sum_{\tilde{d} \in \mathcal{D} / \tilde{d} \geq \hat{d}_{Out} - \omega_p, \tilde{d} \leq \hat{d}_{Out} - 1} A^{LAB}_{p,\tilde{d}} - A^{ULAB}_{p,\hat{d}_{Out}} - A^{LH}_{p,\hat{d}_{Out}}$	$= 0$	p	(2.14)

Figure 3.1: Block structure in FF DUV/YS Upgrades manpower planning model

The analysis of the compact formulation pointed out that the hard constraints were those related to travel and rest period, that is those in the agenda blocks. The main reason is that, in general, for every person $p \in \mathcal{P}$ there can be so many feasible agendas that looking for the optimal one could become computationally expensive. Therefore, if it were possible to restrict the attention of the solver on only those feasible agendas that improve the objective function, then the solution process could be faster. Actually this is possible because it is exactly what the Column Generation technique (CG) does. However, when applying CG a problem arises because it asks for the master problem to be LP instead of MIP, while our assignment variables $X^{\mathcal{P}}_{p,prj}$ are binary. To overcome this issue, two different solution methods are proposed, the first one relies on a heuristic framework, while the second one on another decomposition technique known as Benders' Decomposition (BD). A background introduction of CG and BD is presented in App. A and additional sets, parameters and decision variables introduced in this chapter are summarized in Tables 3.1, 3.2 and 3.3 respectively.

Sets:	
$\mathcal{Q}_p$	set of feasible agendas of locations for employee p
$\mathcal{K}_p$	set that enumerates employee p's agendas in MP or ASM(p) or AFSM(p)
$\mathcal{K}'_p$	set that enumerates employee p's agendas in RMP or in restricted ASM(p) or in restricted AFSM(p)
$\mathcal{K}$	set of elements in at least one $\mathcal{K}_p$, that is $\mathcal{K} := \bigcup_p \mathcal{K}_p$
$\mathcal{D}^{\mathcal{WED}}$	set of all Wednesdays in the planning horizon
$\mathcal{D}_n^{*\mathcal{WED}}$	set of Wednesdays in the planning horizon, with the last $n \in \mathbb{N}$ Wednesdays excluded

Table 3.1: Additional sets used in the decomposition approach

Input Parameters:	
$\xi^{(p),k} \in conv(\mathcal{Q}_p)$	are points in the convex hull of $\mathcal{Q}_p$ representing some feasible agendas $k \in \mathcal{K}_p$ for $p \in \mathcal{P}$
$\psi_{p,k} \in \mathbb{N}$	is the cost of agenda $\xi^{(p),k}$
$\pi_2^{(p)} \in \mathbb{R}^{\lvert \mathcal{PJ}_p^{\mathcal{P}} \rvert}$	is the vector of shadow prices of constraint (3.1) for CG approach and of constraint (3.8) for BD approach
$\pi_1^{(p)} \in \mathbb{R}$	is the shadow price of constraint (3.2) for CG approach and of constraint (3.9) for BD approach
$\pi_0^{(p)} \in \mathbb{R}$	is the shadow price of constraint (3.3) for CG approach and of constraint (3.10) for BD approach

$\tilde{\gamma}_{prj_1,prj_2} \in \{0,1\}$	takes value 0 when two projects prj_1 and prj_2 are "incompatible", that is when it is not possible for any employee to perform both of them, because of overlapping or no enough room for travel days between them

Table 3.2: Additional input parameters used in the decomposition approach

Decision Variables:

$L^{(p)} \in \mathcal{Q}_p$	is the array representing the agenda of employee $p \in \mathcal{P}$, that is $L^{(p)} := (L_{p,l,d})_{l\in\mathcal{L},d\in\mathcal{D}}$
$\Psi_p \in \mathbb{N}$	is the cost of employee p's agenda $L^{(p)}$
$C_p \in \mathbb{R}_{\geq 0}$	are additional variables used to formulate the Benders' Master Problem
$D_p^2 \in \mathbb{R}_{\geq 0}$	are dummy variables used to formulate the AFSM(p)
$\Lambda_{p,k} \in \mathbb{R}_{[0,1]}$	represents the coefficients of the convex combination of agendas $\xi^{(p),k}$
$\xi^{(p),new} \in \mathcal{Q}_p$	represents a new agenda that is found in PP(p) or in ASP(p)
$\Psi_{p,new} \in \mathbb{N}$	is the cost of the new agenda $\xi^{(p),new}$ that is found in PP(p) or in ASP(p)
$L_{p,l,wed}^{WED} \in \{0,1\}$	takes value 1 if employee $p \in \mathcal{P}$ is at location $l \in \mathcal{L}$ on Wednesday $wed \in \mathcal{D}^{\mathcal{WED}}$
$N_{p,wed} \in \mathbb{N}$	represents the number of travel days that employee p does between $wed \in \mathcal{D}_1^{*\mathcal{WED}}$ and the next Wednesday

Table 3.3: Additional decision variables used in the decomposition approach

3.1 Column Generation within a Heuristic Framework

3.1.1 Description

CG is an efficient iterative method that relies on the theory of LP and, in particular, on the observation that in a basic solution all non-basic variables have value 0. Since most of the variables will be non-basic, only some variables really need to be considered

while solving the problem. Hence, the process starts with just a subset of all possible variables in the model and then, at every iteration, the variables that have the potential to improve the objective function are generated on the fly and added to the model. For more information, see App. A.
To apply CG to the original problem described in Chapter 2, first a Dantzig-Wolfe reformulation is needed. Let relax for now the manpower allocation variables $X^{\mathcal{P}}_{p,prj}$, that is let assume that $X^{\mathcal{P}}_{p,prj} \in \mathbb{R}_{[0,1]}$. Let $L^{(p)} := (L_{p,l,d})_{l\in\mathcal{L},d\in\mathcal{D}}$ be the array representing the feasible agenda defined for employee p and let $\mathcal{Q}_p$ be the non-empty and bounded discrete set of p's feasible agendas, namely

$$\mathcal{Q}_p := \{L^{(p)} \in \{0,1\}^{|\mathcal{L}|\times|\mathcal{D}|} \quad : \quad \textit{constraints (2.5)-(2.14) and the corresponding variables domains are satisfied for employee } p\}.$$

If $\{\xi^{(p),k}\}_{k\in\mathcal{K}_p}$ is a finite set of points in the convex hull $conv(\mathcal{Q}_p)$ of $\mathcal{Q}_p$ containing all the extreme points, then every agenda in $\mathcal{Q}_p$ is a convex combination of these points, that is

$$L^{(p)} := \sum_{k\in\mathcal{K}_p} \Lambda_{p,k} \cdot \xi^{(p),k} \quad \forall p \in \mathcal{P}$$

where $\Lambda_{p,k} \in \mathbb{R}_{[0,1]}$ are such that $\sum_{k\in\mathcal{K}_p} \Lambda_{p,k} = 1$ and where $k \in \mathcal{K}$ is an index defined over the whole set $\mathcal{K} := \bigcup_p \mathcal{K}_p$. Moreover, every agenda $L^{(p)} \in \mathcal{Q}_p$ has a cost defined as

$$\Psi_p := \sum_{d\in\mathcal{D}} \Big(T_{p,d} + \Big(1 - L_{p,\beta^H_p,d}\Big)\Big) = \sum_{k\in\mathcal{K}_p} \psi_{p,k} \cdot \Lambda_{p,k}$$

with $\psi_{p,k}$ the cost of agenda $\xi^{(p),k}$. If the previous expressions are substituted in the original problem, we get its Dantzig-Wolfe relaxation.

At this point CG can be applied and its pseudocode is shown in Alg. 3.1. For every person $p \in \mathcal{P}$, let initialize the subset $\mathcal{K'}_p \subset \mathcal{K}_p$ with some feasible agendas. Then at every iteration the *restricted master problem* (RMP) is solved, that is

Minimize:

$$M \cdot \sum_{\substack{prj\in\mathcal{PJ},\\ r\in\mathcal{R}^{\mathcal{PJ}}_{prj}}} D^1_{prj,r} + \sum_{p\in\mathcal{P}} \Psi_p \tag{RMP}$$

Subject to:

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}^{\mathcal{P}}_p$:

$$X^{\mathcal{P}}_{p,prj} = \sum_{r\in\mathcal{R}^{\mathcal{PPJ}}_{p,prj}} X^{\mathcal{PR}}_{p,prj,r} \tag{2.1}$$

$\forall prj \in \mathcal{PJ}, r \in \mathcal{R}^{\mathcal{PJ}}_{prj}$:

$$D^1_{prj,r} + \sum_{\substack{p\in\mathcal{P}/\\ r\in\mathcal{R}^{\mathcal{PPJ}}_{p,prj}}} X^{\mathcal{PR}}_{p,prj,r} \geq \alpha_{prj,r} \tag{2.2}$$

$\forall p \in \mathcal{P}, prj_1, prj_2 \in \mathcal{PJ}_p^{\mathcal{P}}$ *s.t.* $\gamma_{prj_1,prj_2} = 0, prj_1 < prj_2$:

$$X^{\mathcal{P}}_{p,prj_1} + X^{\mathcal{P}}_{p,prj_2} \leq 1 \tag{2.3}$$

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}_p^{\mathcal{P}}$:

$$\sum_{k \in \mathcal{K}'_p} \sum_{d \in \mathcal{D}^{\mathcal{PJ}}_{prj}} \left(\xi^{(p),k}\right)_{\beta^{PJ}_{prj},d} \Lambda_{p,k} \geq \delta_{prj} \cdot X^{\mathcal{P}}_{p,prj} \tag{3.1}$$

$\forall p \in \mathcal{P}$:

$$\Psi_p = \sum_{k \in \mathcal{K}'_p} \psi_{p,k} \cdot \Lambda_{p,k} \tag{3.2}$$

$\forall p \in \mathcal{P}$:

$$\sum_{k \in \mathcal{K}'_p} \Lambda_{p,k} = 1 \tag{3.3}$$

$\forall p \in \mathcal{P}, k \in \mathcal{K}'_p$:

$$\Lambda_{p,k} \in \mathbb{R}_{\geq 0} \tag{3.4}$$

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}_p^{\mathcal{P}}, r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}$:

$$X^{\mathcal{PR}}_{p,prj,r} \in \mathbb{R}_{[0,1]} \tag{2.15}$$

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}_p^{\mathcal{P}}$:

$$X^{\mathcal{P}}_{p,prj} \in \mathbb{R}_{[0,1]} \tag{3.5}$$

$\forall prj \in \mathcal{PJ}, r \in \mathcal{R}^{\mathcal{PJ}}_{prj}$:

$$D^1_{prj,r} \in \mathbb{R}_{\geq 0} \tag{2.26}$$

where the dummy variables $D^1_{prj,r}$ guarantee it to be always feasible, no matter how the sets $\mathcal{K}'_p$ are initialized. It is crucial to observe that we do not ask for $\Lambda_{p,k} \in \mathbb{R}_{[0,1]}$ since the convexity constraint already guarantees the upper bound. This tiny difference changes the expression of the objective value for the pricing problems, that otherwise should also consider the contribution of this additional bound.

Once the RMP has been solved to optimality, the *pricing problems* PP(p) are constructed. If $\pi_2^{(p)}$, $\pi_1^{(p)}$ and $\pi_0^{(p)}$ denote the vectors of shadow prices corresponding to (3.1), (3.2) and (3.3) in RMP respectively, then $\forall p \in \mathcal{P}$ the pricing problem PP(p) assumes the form

Minimize:

$$-\sum_{prj \in \mathcal{PJ}_p^{\mathcal{P}}} \left(\pi_2^{(p)}\right)_{prj} \cdot \sum_{d \in \mathcal{D}^{\mathcal{PJ}}_{prj}} \left(\xi^{(p),new}\right)_{\beta^{PJ}_{prj},d} - \pi_1^{(p)} \cdot (-\Psi_{p,new}) - \pi_0^{(p)} \tag{PP(p)}$$

Subject to:

$$\xi^{(p),new} \in \mathcal{Q}_p \tag{3.6}$$

where $\Psi_{p,new}$ is the cost of the new agenda $\xi^{(p),new}$. This MIP checks for the existence of a new variable $\Lambda_{p,k^{new}}$ with $k^{new} \in \mathcal{K}_p \setminus \mathcal{K}'_p$ that, if added to the RMP, could possibly improve the solution value. Indeed, the objective function of the PP(p) is the reduced cost of the new agenda and, if the MIP results in a solution having negative reduced cost, this agenda will be added to the RMP.

Finally, CG stops when the optimal solutions of all the pricing problems have a non-negative objective value, meaning that no further improvement is possible.

Algorithm 3.1 Column Generation Algorithm

Initialize $\mathcal{K}'_p$ for all $p \in \mathcal{P}$.
repeat
  Solve the RMP to optimality.
  $Improvement \leftarrow 0$.
  for all employees $p \in \mathcal{P}$ **do**
    Solve the PP(p) to optimality.
    if its solution has a negative objective value **then** // negative reduced cost
      Add the variable $\Lambda_{p,k^{new}}$, corresponding to the new agenda $\xi^{(p),new}$, to the RMP.
      $Improvement \leftarrow Improvement + 1$.
    end if
  end for
until $Improvement > 0$
return.

The above algorithm, if used alone, is not suitable for our purposes, because it can only be applied under the assumption that $X^{\mathcal{P}}_{p,prj} \in \mathbb{R}_{[0,1]}$: CG relies on a duality argument that forces the master problem to be LP. An integer solution can be obtained by integrating the Column Generation algorithm within a heuristic framework. To get a solution hopefully not too far from the optimal solution of the original problem, we adopted a backtracking search strategy (Heuristic-1). By backtracking search we refer to a depth-first search within the MIP search tree of the original problem until a first feasible integer solution is found. And in every node of the search tree the LP problem is solved by CG. The benefit of this heuristic is that, in theory it always finds a feasible solution for the original problem, if it exists. However, in the worst case scenario, backtracking occurs many times and so the solution process becomes very slow. Since the previous heuristic can be computationally expensive, another possible heuristic (Heuristic-2) consists in applying CG to the Dantzig-Wolfe relaxation and in solving then the final RMP again but this time with its original integrality requirements reintroduced. In general, this second method is faster, but at the same time it can also terminate without any feasible solution for the original problem. Moreover, the drawback of both the heuristics is that they can result in solutions of bad quality and too far away from the optimum, so they do not guarantee optimality.

3.1.2 Implementation Algorithm

A pseudocode of the CG approach with iterative backtracking search (Heuristic-1) is presented in Alg. 3.2. Even though backtracking fits better in a recursive setting, our choice of an iterative algorithm was dictated by the fact that we implemented our code in AIMMS and it does not allow for recursive functions that use the AIMMS GMP library.

Algorithm 3.2 Column Generation with Iterative Backtracking Search

```
Step ← 1.
repeat
    if NumVisits(Step) = 0 then
        (p*, prj*) ← Select_Variable()
        if ∄(p*, prj*) then                                          // integer solution found
            return.
        end if
        Fix X^P_{p*,prj*} ← Select_Unassigned_Domain_Value(p*, prj*).  // Fix to 0 or 1
        Fixed(Step) ← (p*, prj*).
        NumVisits(Step) ← 1.
        Apply CG.
        if original problem is feasible then
            Step ← Step + 1.
        end if
    else if NumVisits(Step) = 1 then
        (p*, prj*) ← Fixed(Step).
        Fix X^P_{p*,prj*} ← Select_Unassigned_Domain_Value(p*, prj*).  // Fix to the last value left
        NumVisits(Step) ← 2.
        Apply CG.
        if original problem is feasible then
            Step ← Step + 1.
        end if
    else                                                             // backtracking occurs
        (p*, prj*) ← Fixed(Step).
        Unfix X^P_{p*,prj*}.
        NumVisits(Step) ← 0.
        Fixed(Step) ← ‘’.
        Step ← Step − 1.
    end if
until Step > 0
return.
```

First, every set $\mathcal{K}'_p$ is initialized with a single agenda which is constructed by filling it with as many home locations as possible, but that at the same time is consistent with the locations entered in input by the user for the fixed activities and with the number of travel days between different locations. This initialization is fine because the presence of dummy variables guarantees the initial RMP to be always feasible, no matter how the sets of agendas are initialized.

The backtracking search can then start. At every iteration first CG is applied and then a manpower allocation variable is picked and fixed to one of its integer values. The implementation of CG is similar to the Alg. 3.1 but with the only difference that, in order to speed up the solution process, the pricing problems are not solved to optimality but until an incumbent with negative objective value is found. They are thus solved to optimality only when no such incumbent exists. If at some point CG returns a solution containing some non-zero dummy variables or if the fixed variable returns an infeasible RMP, then we backtrack and try to fix another value for the last branching variable. If all the possible values have already been tried, then a further backward step is needed. The loop terminates when the first solution satisfying all the integrality requirements is found or when all the search tree has been explored without any solution found. In the latter case, it means that the original MIP problem is also infeasible.

It remains to specify the variable selection and the branching rule. Since the heuristic stops at the first solution found then we opted for a greedy selection, which consists in selecting the variable $X^{\mathcal{P}}_{p,prj}$ being closer to its upper bound 1. As a result, the branching rule consists in trying to fix it to 1 first. If the selected variable already takes value 1, then the step can be immediately increased without applying CG again.

As far as the second heuristic (Heuristic-2) is concerned, after all $\mathcal{K}'_p$ with $p \in \mathcal{P}$ are initialized, CG is applied to the RMP. At this point, the integrality requirements are introduced in the RMP and the resulting MIP is solved to optimality.

3.2 Benders' Decomposition combined with Column Generation

3.2.1 Description

As already pointed out in Fig. 3.1, the only link between manpower allocation to projects and employees' agendas is provided by the constraints (2.4) that we designated, in view of this, under the name of *linking constraints*. Furthermore, the agenda of every employee does not depend, at least in theory, on other employees' agendas but is characterized by an independent block of constraints. In practice there could be some dependencies between agendas of employees in a same team, but we decided to leave the team constraint out in our research. Finally, the integrality requirement on the assignment variables $X^{\mathcal{P}}_{p,prj}$ cannot be relaxed, therefore a decomposition algorithm that deals with integer variables is needed. All the previous considerations on the structure of the model motivated us to solve the problem by combining Benders' Decomposition (BD) together with Column Generation (CG) in order to get a powerful decomposition algorithm.

The original problem formulated in Chapter 2 is split into a *Benders' master problem* (BMP), containing at the beginning only the constraints involving the assignments of projects, and into *agenda subproblems* AS(p), one for every person $p \in \mathcal{P}$, containing the linking constraints and the constraints that define feasible agendas.

For every $p \in \mathcal{P}$, let C_p be the new variables added in the BMP according to the BD. The mathematical formulation of the BMP is then

Minimize:

$$M \cdot \sum_{\substack{prj \in \mathcal{PJ}, \\ r \in \mathcal{R}^{\mathcal{PJ}}_{prj}}} D^1_{prj,r} + \sum_{p \in \mathcal{P}} C_p \tag{BMP}$$

Subject to:

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}^{\mathcal{P}}_p$:

$$X^{\mathcal{P}}_{p,prj} = \sum_{r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}} X^{\mathcal{PR}}_{p,prj,r} \tag{2.1}$$

$\forall prj \in \mathcal{PJ}, r \in \mathcal{R}^{\mathcal{PJ}}_{prj}$:

$$D^1_{prj,r} + \sum_{\substack{p \in \mathcal{P} / \\ r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}}} X^{\mathcal{PR}}_{p,prj,r} \geq \alpha_{prj,r} \tag{2.2}$$

$\forall p \in \mathcal{P}, prj_1, prj_2 \in \mathcal{PJ}^{\mathcal{P}}_p \ s.t. \ \gamma_{prj_1,prj_2} = 0, prj_1 < prj_2$:

$$X^{\mathcal{P}}_{p,prj_1} + X^{\mathcal{P}}_{p,prj_2} \leq 1 \tag{2.3}$$

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}^{\mathcal{P}}_p, r \in \mathcal{R}^{\mathcal{PPJ}}_{p,prj}$:

$$X^{\mathcal{PR}}_{p,prj,r} \in \mathbb{R}_{[0,1]} \tag{2.15}$$

$\forall p \in \mathcal{P}, prj \in \mathcal{PJ}^{\mathcal{P}}_p$:

$$X^{\mathcal{P}}_{p,prj} \in \{0, 1\} \tag{2.16}$$

$\forall prj \in \mathcal{PJ}, r \in \mathcal{R}^{\mathcal{PJ}}_{prj}$:

$$D^1_{prj,r} \in \mathbb{R}_{\geq 0} \tag{2.26}$$

$\forall p \in \mathcal{P}$:

$$C_p \in \mathbb{R}_{\geq 0} \tag{3.7}$$

and, every time a solution of BMP is found, it is used to update the right-hand side of the linking constraints in all the AS(p). Every subproblem could produce then a feasibility or optimality cut that is added to the BMP.

A complication in this decomposition is provided by the observation that in our model the agenda subproblems are MIP, while BD requires the Benders' subproblems to be LP in order to apply the duality theory. Moreover, the subproblems are still quite big, meaning that the computational time spent there by the solver can be too high. Therefore, first a Dantzig-Wolfe relaxation is applied to the subproblems, and they are then solved by Column Generation (CG). Thanks to this approach, it is not needed in our case to reintroduce later the integrality condition for the agendas. Hence, if Q_p is the set of all the feasible agendas for employee p and $\{\xi^{(p),k}\}_{k\in\mathcal{K}_p}$ is a finite set of points in the convex hull $conv(\mathcal{Q}_p)$ of $\mathcal{Q}_p$ containing all the extreme points, then every agenda in $\mathcal{Q}_p$ is a convex combination of these points, that is

$$L^{(p)} := \sum_{k\in\mathcal{K}_p} \Lambda_{p,k} \cdot \xi^{(p),k} \quad \forall p \in \mathcal{P}$$

where $\Lambda_{p,k} \in \mathbb{R}_{[0,1]}$ are such that $\sum_{k\in\mathcal{K}_p} \Lambda_{p,k} = 1$ and are the coefficients of the convex combination. The cost Ψ_p of every agenda $L^{(p)} \in \mathcal{Q}_p$ is expressed by

$$\Psi_p := \sum_{d\in\mathcal{D}} \Big(T_{p,d} + \Big(1 - L_{p,\beta_p^H,d}\Big)\Big) = \sum_{k\in\mathcal{K}_p} \psi_{p,k} \cdot \Lambda_{p,k}$$

with $\psi_{p,k}$ the cost of agenda $\xi^{(p),k}$. If the previous identities are substituted in the $AS(p)$, we get its Dantzig-Wolfe relaxation (see App. A for more details)

Minimize:

$$\Psi_p \tag{ASM(p)}$$

Subject to:

$\forall prj \in \mathcal{PJ}_p^{\mathcal{P}}$:

$$\sum_{k\in\mathcal{K}_p} \sum_{d\in\mathcal{D}_{prj}^{\mathcal{PJ}}} \Big(\xi^{(p),k}\Big)_{\beta_{prj}^{PJ},d} \Lambda_{p,k} \geq \delta_{prj} \cdot X_{p,prj}^{\mathcal{P}} \tag{3.8}$$

$$\Psi_p = \sum_{k\in\mathcal{K}_p} \psi_{p,k} \cdot \Lambda_{p,k} \tag{3.9}$$

$$\sum_{k\in\mathcal{K}_p} \Lambda_{p,k} = 1 \tag{3.10}$$

$\forall k \in \mathcal{K}_p$:

$$\Lambda_{p,k} \in \mathbb{R}_{\geq 0} \tag{3.11}$$

$$\Psi_p \in \mathbb{R}_{\geq 0} \tag{3.12}$$

We refer to this problem as the *agenda subproblem master* ASM(p) for employee $p \in \mathcal{P}$. Since the ASM(p) could be infeasible, first of all a feasibility check is needed through the solution of the *agenda feasibility subproblem master* AFSM(p), formulated as

Minimize:

$$D_p^2 + 0 \cdot \Psi_p \tag{AFSM(p)}$$

Subject to:

$\forall prj \in \mathcal{PJ}_p^{\mathcal{P}}$:

$$D_p^2 + \sum_{k \in \mathcal{K}_p} \sum_{d \in \mathcal{D}_{prj}^{\mathcal{PJ}}} \left(\xi^{(p),k}\right)_{\beta_{prj}^{PJ},d} \Lambda_{p,k} \geq \delta_{prj} \cdot X_{p,prj}^{\mathcal{P}} \tag{3.13}$$

$$\Psi_p = \sum_{k \in \mathcal{K}_p} \psi_{p,k} \cdot \Lambda_{p,k} \tag{3.9}$$

$$\sum_{k \in \mathcal{K}_p} \Lambda_{p,k} = 1 \tag{3.10}$$

$\forall k \in \mathcal{K}_p$:

$$\Lambda_{p,k} \in \mathbb{R}_{\geq 0} \tag{3.11}$$

$$\Psi_p \in \mathbb{R}_{\geq 0} \tag{3.12}$$

$$D_p^2 \in \mathbb{R}_{\geq 0} \tag{3.14}$$

where D_p^2 is a dummy variable introduced to guarantee feasibility and $X_{p,prj}^{\mathcal{P}}$ is not a variable but a fixed value previously obtained by the BMP. If the optimal objective value of AFSM(p) is equal to 0, then the ASM(p) is feasible and, once solved to optimality, its dual vector leads to a new optimality cut that is added to the BMP. If instead the objective value is strictly positive, the ASM(p) is infeasible and so is the BMP solution for the original problem. In this case, the AFSM(p) provides a dual vector that leads to a new feasibility cut to be added to the BMP. Let $\pi_2^{(p)}$, $\pi_1^{(p)}$ and $\pi_0^{(p)}$ be the vectors of shadow prices corresponding to the first, second and third symbolic constraints respectively, for both ASM(p) and AFSM(p). Then, written out in mathematical terms, the optimality and feasibility cuts for employee $p \in \mathcal{P}$ become

$$\sum_{prj \in \mathcal{PJ}_p^{\mathcal{P}}} \left(\delta_{prj} \cdot X_{p,prj}^{\mathcal{P}} \cdot \left(\pi_2^{(p)} \right)_{prj} \right) \leq IsOptCut(p) \cdot C_p - \pi_0^{(p)}$$

where

$$IsOptCut(p) := \begin{cases} 1, & \text{for optimality cuts} \\ 0, & \text{for feasibility cuts} \end{cases}$$

As the formulations suggest, both the ASM(p) and AFSM(p) are expressed using the Dantzig-Wolfe relaxation and so they are then solved by CG. This means that at every iteration of CG only a restricted subset $\mathcal{K}'_p \subset \mathcal{K}_p$ of agendas is considered. When the restricted version of these subproblems has been solved to optimality then an *agenda subproblem pricing* ASP(p) is used to check for the existence of a variable $\Lambda_{p,k^{new}}$, with $k^{new} \in \mathcal{K}_p \setminus \mathcal{K}'_p$, corresponding to a new agenda $\xi^{(p),new}$ that, if added, could possibly improve the solution value. The objective function of the ASP(p) is the reduced cost of the new agenda and, if the MIP results in a solution having negative reduced cost, this agenda will be added to ASM(p) and AFSM(p).

For both ASM(p) and AFSM(p) the ASP(p) can be formulated as the MIP

Minimize:

$$- \sum_{prj \in \mathcal{PJ}_p^{\mathcal{P}}} \left(\pi_2^{(p)} \right)_{prj} \cdot \sum_{d \in \mathcal{D}_{prj}^{\mathcal{PJ}}} \left(\xi^{(p),new} \right)_{\beta_{prj}^{PJ},d} - \pi_1^{(p)} \cdot (-\Psi_{p,new}) - \pi_0^{(p)} \tag{ASP(p)}$$

Subject to:

$$\xi^{(p),new} \in \mathcal{Q}_p \tag{3.15}$$

where $\Psi_{p,new}$ is the cost of the new agenda $\xi^{(p),new}$. In general the pricing subproblems do not need to be solved to optimality, as long as a solution with negative reduced cost is identified. Moreover, it is important to observe that the pricing subproblems of ASM(p) and AFSM(p) have the same feasible space, they only change in the objective. Therefore every time a new column is added to ASM(p), it can also be added to AFSM(p) and vice-versa.

3.2.2 Implementation Algorithm

Unfortunately, the classic BD algorithm often suffers from slow convergence. The two main reasons leading to slow convergence are usually the quality of the produced Benders' cuts and the weakness of the lower bound obtained from the master problem during the first iterations.

Aside from the quality of the produced Benders' cuts, the weakness of the lower bound is caused by the fact that at the beginning the BMP contains no information at all on which assignments give agendas that are both feasible and with low cost. That is, during the first iterations of the classic algorithm the BMP is far away from the aim of the original problem.

A powerful strategy that tries to add some cuts to the BMP before solving the MIP problem by BD is the so-callled 2-phase Benders' algorithm. The pseudocode of this algorithm, implemented together with other enhancement techniques that will be described later, is shown in Alg. 3.3.

Algorithm 3.3 2-phase Benders' Algorithm

PHASE 1

In BMP, substitute (2.3) with (3.16) and add (3.22).
Relax the integrality requirements for BMP.
$UpperBound \leftarrow +\infty$.
$BendersAlgorithmFinished \leftarrow 0$.
while not $BendersAlgorithmFinished$ **do**
    Solve the BMP to optimality.
    $LowerBound \leftarrow$ solution of BMP.
    if $\sum_{\substack{prj \in \mathcal{PJ},\\ r \in \mathcal{R}^{\mathcal{PJ}}_{prj}}} D^1_{prj,r} > 0$ **then**
        return. // original problem is infeasible
    end if
    $NumberFeasibilityCuts \leftarrow 0$.
    $NewObjValue \leftarrow 0$.
    for all employees $p \in \mathcal{P}$ **do**
        Solve the AFSM(p) by CG.
        if solution has positive objective value **then**
            Add a feasibility cut to the BMP.
            $NumberFeasibilityCuts \leftarrow NumberFeasibilityCuts + 1$.
        end if
    end for
    if $NumberFeasibilityCuts = 0$ **then** // A new incumbent for the original problem is found
        for all employees $p \in \mathcal{P}$ **do**
            Solve the ASM(p) by CG.
            if solution has a bigger objective value than C_p **then**
                Add an optimality cut to the BMP.
            end if
        end for
        $NewObjValue \leftarrow M \cdot \sum_{\substack{prj \in \mathcal{PJ},\\ r \in \mathcal{R}^{\mathcal{PJ}}_{prj}}} D^1_{prj,r} + \sum_{p \in \mathcal{P}}$ obj value of ASM(p).
        if $UpperBound > NewObjValue$ **then**
            $UpperBound \leftarrow NewObjValue$.
        end if

if $NewObjValue - Lowerbound \leq \epsilon$ **then** // so also $UpperBound - Lowerbound \leq \epsilon$
    $BendersAlgorithmFinished \leftarrow 1$.
  end if
 end if
end while

PHASE 2
Reintroduce the integrality requirements for BMP.
Retain all cuts that have been added in **PHASE 1**.
Solve the BMP by branch-and-bound and:
if a new incumbent is found **then**
 $NumberFeasibilityCuts \leftarrow 0$.
 for all employees $p \in \mathcal{P}$ **do**
  Solve the AFSM(p) by CG.
  if solution has positive objective value **then**
   Add a feasibility cut to the BMP.
   $NumberFeasibilityCuts \leftarrow NumberFeasibilityCuts + 1$.
  end if
 end for
 if $NumberFeasibilityCuts = 0$ **then**
  for all employees $p \in \mathcal{P}$ **do**
   Solve the ASM(p) by CG.
   if solution has a bigger objective value than C_p **then**
    Add an optimality cut to the BMP.
   end if
  end for
 end if
end if
return.

In the first phase, the LP-relaxed BMP is solved applying BD. Then, the master problem created in the first phase is used for BD in the second phase but with the reintroduction of the integrality condition. The relaxed MIP problem can be solved more efficiently than the MIP problem itself and so, by adding the Benders' cuts found during the first phase, the Benders' Decomposition algorithm needs considerably less iterations in the second phase to solve our original MIP problem.
However, in our case, after the first phase the BMP becomes in general quite big and it is expensive to solve it to optimality at every Benders' iteration. Therefore, in the second phase a modern version of BD is implemented, the so-called Branch-and-Benders-Cut method (B&BC), in which a single search tree is explored for the BMP and, whenever a new incumbent is found, it is used to add new cuts to the BMP. If no cuts are added, then the incumbent is not rejected and goes to update the current best solution of the search tree.

Since in our problem BD is applied with multiple subproblems, we decided to add optimality cuts only in those iterations where all the agenda subproblems master were proved to be feasible. Another possibility consists in adding optimality cuts independently for every feasible agenda subproblem master, even if in the same Benders' iteration other agenda subproblems master result to be infeasible. Our choice was made after a series of tests between the two approaches.

To strengthen the lower bound even further, an important observation is in the different role played by feasibility and optimality cuts. The main role of feasibility cuts is to guarantee that the solution obtained from the BMP is valid for the initial problem while the main role of optimality cuts is to restrict the lower bound so that a low objective value obtained from the BMP is equivalent to a low objective value for the initial problem. Thus producing more optimality than feasibility cuts would lead to faster convergence. In our problem, infeasible iterations of the BMP may be reduced a priori by modifying the constraints (2.3). Indeed we can extend them so as to include all pairs of projects that overlap or do not have enough room for traveling between their locations. We say that such projects are "incompatible". If we denote by $\tilde{\gamma}_{prj_1,prj_2}$ the boolean parameter that takes value 0 for pairs of incompatible projects, then the modified constraints assume the form

$$\forall p \in \mathcal{P}, prj_1, prj_2 \in \mathcal{PJ}_p^{\mathcal{P}} \ s.t. \ \tilde{\gamma}_{prj_1,prj_2} = 0, prj_1 < prj_2 :$$

$$X^{\mathcal{P}}_{p,prj_1} + X^{\mathcal{P}}_{p,prj_2} \leq 1. \tag{3.16}$$

Thanks to the additional tight constraints, the algorithm does not spend too much time on infeasible iterations and the first lower bound derived by the master problem is improved.

In order to further narrow the solution space of the master problem and obtain improved lower bounds, we also tried to develop a series of valid inequalities, according to the features of our problem, that could be added to the BMP from the first iteration. The main idea is that the previous constraints introduce in the BMP more information about feasible assignments but no information is added in terms of which assignments are preferable. To do that, we tried to introduce 2 new variables in the BMP, $L^{WED}_{p,l,wed}$ and $N_{p,wed}$. The first variable keeps track of the location of employee $p \in \mathcal{P}$ every Wednesday $wed \in \mathcal{D}^{\mathcal{WED}}$, while the second one counts the number of travel days for employee $p \in \mathcal{P}$ in the week from $wed \in \mathcal{D}^{\mathcal{WED}}$ to the next Wednesday. First, additional constraints are added to the model in order to characterize the new variables. By requesting that

$$\forall p \in \mathcal{P}, l \in \mathcal{L}, wed \in \mathcal{D}^{\mathcal{WED}} :$$

$$L^{WED}_{p,l,wed} \geq \sum_{\substack{prj \in \mathcal{PJ}_p^{\mathcal{P}} / \\ \beta^{PJ}_{prj} = l, \\ wed \in \mathcal{D}^{PJ}_{prj}}} X^{\mathcal{P}}_{p,prj} \tag{3.17}$$

and

$\forall p \in \mathcal{P}, wed \in \mathcal{D}^{\mathcal{WED}} \cap \mathcal{D}_p^{\mathcal{IN}}$:

$$L^{WED}_{p,\beta^{IN}_{p,wed},wed} = 1 \tag{3.18}$$

we force employee $p \in \mathcal{P}$ to go every Wednesday to the location of its assigned project or to the location already given in input, if any. Moreover, every Wednesday should have exactly one location, that is

$\forall p \in \mathcal{P}, wed \in \mathcal{D}^{\mathcal{WED}}$:

$$\sum_{l \in \mathcal{L}} L^{WED}_{p,l,wed} = 1, \tag{3.19}$$

and the number of travel days among two consecutive Wednesdays is computed by asking for

$\forall p \in \mathcal{P}, l_1 \in \mathcal{L}, td \in \mathcal{TD}, wed \in \mathcal{D}_1^{*\mathcal{WED}}$ $s.t.$ $td \geq 1$:

$$N_{p,wed} \geq td \cdot \left(L^{WED}_{p,l_1,wed} + \sum_{\substack{l_2 \in \mathcal{L}/ \\ \tau_{l_1,l_2}=td}} L^{WED}_{p,l_2,wed+1} - 1 \right) \tag{3.20}$$

Finally, the valid inequalities that can be added to the BMP are

$\forall p \in \mathcal{P}$:

$$C_p \geq \sum_{wed \in \mathcal{D}_1^{*\mathcal{WED}}} 2 \cdot N_{p,wed} + \sum_{\substack{prj \in \mathcal{PJ}_p^{\mathcal{P}}/ \\ \beta^{PJ}_{prj} \neq \beta^H_p}} \delta_{prj} \cdot X^{\mathcal{P}}_{p,prj}. \tag{3.21}$$

Since the constraints containing information on the objective cost are included after decomposition in the Benders' subproblems rather than the BMP, these valid inequalities are to recover the function of these constraints in the BMP as much as possible. Therefore, we hoped they could restrict the solution space of the master problem, and thus result in better convergence behavior. Unfortunately, after a few analysis we realized that these valid inequalities only make the BMP bigger without giving any contribution in terms of speed. Actually, the 2-phase Benders works much better with cuts of the type

$\forall p \in \mathcal{P}$:

$$C_p \geq \sum_{\substack{prj \in \mathcal{PJ}_p^{\mathcal{P}}/ \\ \beta^{PJ}_{prj} \neq \beta^H_p}} \delta_{prj} \cdot X^{\mathcal{P}}_{p,prj} \tag{3.22}$$

even if they are less tight than those in (3.21). Hence, in the computational results we are going to present in the next section the valid inequalities (3.22) are added to the BMP, while the cuts (3.21) are not considered.

3.3 Computational Results

3.3.1 Input Data and Settings

All the tests were carried out on a PC having an Intel Core i7-3632QM processor with a speed of 2.20 GHz and a memory (RAM) of 4 GB. The model was implemented in AIMMS Platform and CPLEX 12.7.1 was used as solver.
In total we have created for our tests 9 cases, whose properties are summarized in Table 3.4. These cases have no relation with FF DUV/YS Upgrades department, they have been created with the only goal of testing how well our solution methods deal with larger amount of manpower and/or a longer time horizon. In all these cases, we set the following rule for rest period: 1 week of rest after at most 2 weeks abroad, otherwise 2 weeks of rest. Moreover, an employee cannot stay outside his home location for more than 4 weeks. In addition, no planned activity is specified in input but we assume to start with a blank schedule.

Case	#Employees	#Projects	Time Horizon	Average #Employees needed per Project
Case-1	50	8	1 month	7.5
Case-2	50	17	2 months	10.8
Case-3	50	25	3 months	10.8
Case-4	110	16	1 month	8.44
Case-5	110	31	2 months	10.3
Case-6	110	45	3 months	10.8
Case-7	224	24	1 month	11.9
Case-8	224	60	2 months	11.2
Case-9	224	97	3 months	10.7

Table 3.4: Cases used in the computational experiments

3.3.2 Analysis and Comparison

As already introduced at the beginning of the chapter, the MIP formulation of the problem without any decomposition strategy, due to its complexity, cannot be efficiently solved for large scale instances. Table 3.5 summarizes the experimental results obtained by solving the problem in its compact formulation. The time limit for all the cases was set to 18000 seconds (5 hours). The main issue was for a 3-month horizon, since the solver was not able to find a solution already for the instances with less manpower. Case-9 was not tested, given the infeasibility results already obtained for Case-3 and Case-6, which have its same time horizon of 3 months but contain few people to be scheduled, and for Case-8, which considers the same manpower but in a 2-month horizon.

Case	Iterations	Time	#Constr.	#Var.	Solution	Best LP Bound (GAP)
Case-1	3350	19.84 sec	100906	47034	105	105 (0.00%)
Case-2	68253	360.05 sec	181437	83016	843	843 (0.00%)
Case-3	2017939	18019 sec	270603	122751	no feas. sol.	1658.68 (-)
Case-4	117736	600.57 sec	224005	105046	313	313 (0.00%)
Case-5	1614880	18032.64 sec	402161	185118	1142	1112 (2.63%)
Case-6	1287050	18010.67 sec	598779	273449	no feas. sol.	2008.85 (-)
Case-7	905021	16042.99 sec	456934	215452	936	936 (0.00%)
Case-8	440559	18068.61 sec	825255	383257	no feas. sol.	1961.27 (-)
Case-9	-	-	1231814	567043	-	-

Table 3.5: Computational results for the MIP formulation

As far as the CG approach is concerned, we expected the time needed for CG with backtracking search (Heuristic-1) to be bigger than the time for CG with the second heuristic (Heuristic-2). This is confirmed by Table 3.6, where we can see that CG with backtracking search can hardly ever find a solution within the time limit of 18000 sec. In more detail, it did not succeed in solving Case-3 due to many backtracking steps. For Case-6, Case-8 and Case-9 backtracking never occurs, but the last two cases have many manpower allocation variables and therefore the method would require an excessive number of iteration steps in order to find a solution. The result does not change if the time limit is extended to 28800 sec (8 hours). The second heuristic shows promising results for Case-3, Case-6 and Case-8 because it returns feasible solutions with a GAP of 23.67%, 8.98% and 33.76% respectively from the best LP bound of Table 3.5. Unfortunately, it does not find any feasible solution for Case-9.

Case	CG with Heuristic-1		CG with Heuristic-2	
	Time	Solution	Time	Solution
Case-3	18006.36 sec	no sol. found	653.30 sec	2173
Case-6	14830.06 sec	2528	18012.04 sec	2207
Case-8	18020.18 sec	no sol. found	18027.72 sec	2961
Case-9	18018.73 sec	no sol. found	18008.32 sec	no feas. sol.

Table 3.6: Computational time for CG with heuristics

The previous results suggest that the choice of a good heuristic is fundamental to get an efficient algorithm based on CG. However the computational time also relies on CG, hence a further investigation in this direction may be interesting. As highlighted in Table 3.7, solving the Dantzig-Wolfe relaxation provides tighter lower bounds for the original MIP problem than the LP-relaxation and the elapsed time for its resolution is competitive for small instances. Furthermore, for larger cases it outperforms the LP-relaxation in terms of execution time. This means that a CG approach can be very efficient for our problem, but it should be integrated in a better framework in order to solve very large instances.

Case	LP-relaxation		Dantzig-Wolfe relaxation	
	Optimal Value	Elapsed Time	Optimal Value	Elapsed Time
Case-1	103.72	5.99 sec	105	4.53 sec
Case-2	713.59	27.14 sec	757.75	45.63 sec
Case-3	1198.10	233.38 sec	1407.87	235.08 sec
Case-4	116.5	20.88 sec	120.67	16.05 sec
Case-5	735.41	160.72 sec	756.87	79.83 sec
Case-6	1496.42	1162.30 sec	1552.03	334.89 sec
Case-7	823.16	46.81 sec	829.08	50.53 sec
Case-8	1938.28	4021.09 sec	1994.47	668.97 sec
Case-9	2800.55	6952.63 sec	2906.65	6265.02 sec

Table 3.7: Comparison between LP-relaxation
and Dantzig-Wolfe relaxation of the original problem

In regard to BD, we implemented a 2-phase algorithm because the classic version and the modern version alone were not efficient at all. Without setting any time limit on the first phase, the method would spend all the 18000 sec in that phase, therefore we decided to set there a time limit of 10800 sec (3 hours). The computational results of the BD approach are depicted in Table 3.8.

Case	Time Phase 1	Total Time	Solution	Best LP Bound (GAP)
Case-3	10805.25 sec	18012.02 sec	2031	1431 (29.54%)
Case-6	10807.48 sec	18005.31 sec	2908	1862 (35.97%)
Case-8	10802.80 sec	18010.04 sec	4303	2590 (39.81%)
Case-9	10804.72 sec	18014.18 sec	no sol. found	4404 (-)

Table 3.8: Computational results for 2-phase BD combined with CG

As for CG with Heuristic-2, the BD strategy combined with CG finds a feasible solution for Case-3, Case-6 and Case-8 but not for Case-9. Moreover, for Case-6 and Case-8 it returns a solution which is quite far away from the optimum. Despite all the enhancement techniques that we applied, the issue is still in the slow convergence during the second phase. The first phase plays a key role because without the cuts added in that stage the second phase would be slower, but even after 3 hours of first phase the second phase is not fast enough. If we extended the overall time limit from 5 to 8 hours, the GAP between the current best solution and the best LP bound would only make slight improvements. This analysis suggests that the BD strategy would be promising if the speed of the second phase could be somehow improved.

Chapter 4

Conclusion and Future Work

This thesis addresses the short-term manpower planning problem that arises inside the department of FF DUV/YS Upgrades at ASML, in Veldhoven (The Netherlands). The key features of the thesis are the mathematical modeling of the problem and a first investigation into a solution methodology that could possibly solve it for large-scale instances.

The problem presents a variety of modeling challenges and, among others, travel and rest period constraints play a major role in the model complexity. A traditional branch-and-bound approach could not solve this model for large input data and, therefore, a decomposition strategy is needed.

Two main decomposition approaches are discussed in Chapter 3. One of them is based on the integration of Column Generation within a heuristic framework and, in particular, two different heuristics are described. The first one consists into a backtracking search that at every step tries to fix one manpower allocation variable and apply Column Generation to the new RMP. Unfortunately, this approach is computationally too expensive to give results in an acceptable amount of time. A better heuristic consists in applying Column Generation only once and in solving then the MIP obtained by reintroducing the integrality requirements in the final RMP. Computational results show that this approach can tackle all the large instances used in our tests except one. However, it gives no guarantee of optimality. This consideration leads to the development of a Benders' Decomposition strategy combined with Column Generation. This approach suffers from slow convergence and therefore several enhancement techniques are proposed. Unfortunately, all these improvements are not enough to optimally solve the large-scale instances used in the experimental analysis.

The results of our research suggest that a further investigation is still needed in order to find out the best approach to solve such a complex problem. The first insight is that a decomposition strategy may be very powerful because of the big size and particular structure of FF DUV/YS Upgrades department. As far as Column Generation is concerned, it could be interesting to implement a branch-and-price algorithm, since it is an exact method and CG has shown to be quite fast for our problem. However, a branch-and-price approach should always be implemented together with techniques that could help in further reducing the runtime, such as heuristic exploration or efficient pruning techniques. Moreover, the pricing problems are independent and so they could also be solved in parallel on multiple threads. Also the Benders' Decomposition approach opens to further research, as it seems promising but suffers from slow convergence during the second phase. In particular, we tried to improve the weakness of its lower bound by introducing more information in the BMP before BD is applied. However, a second reason for the slow convergence is usually in the quality of the produced Benders' cuts. In regard to that, some techniques can be found in the literature and an example is given by the so-called Pareto-optimal cuts, as discussed in [11].

Aside these decomposition techniques, in the recent years lots of attention has been paid on pure heuristics or metaheuristics, in particular advanced local search algorithms and evolutionary algorithms. Sometimes their application to large-scale problems has even outperformed the other approaches.

Finally, another interesting approach could be to start looking at a more decentralized type of planning. That is, in a first stage manpower and upgrades could be clustered per regions and, in this way, only local people are assigned to upgrades. A second phase is then needed in order to allocate manpower to those roles in upgrades that are left uncovered after the first phase. This strategy of local ownership for manpower and upgrades can be useful in reducing the size of employees involved in the mathematical programs. A similar example is described in [8].

In our computational analysis, we focused on instances with several amounts of manpower. However, in the past months FF DUV/YS Upgrades department has grown a lot in manpower, therefore the current challenge is to find a solution methodology that could possibly deal with even larger input data.

Appendix A

Mathematical Background

The manpower planning problem addressed in this thesis is formulated as a MIP and is solved using some advanced decomposition techniques. In this appendix, the necessary theoretical background is provided (for more information see [3], [4], [10], [11] and [13]).

Linear and Mixed Integer Programs

A *linear program* (LP) is an optimization problem that, in its formulation as a minimization problem, can be written as

$$\begin{array}{lrcl} \min\limits_{x} & c^T x & & \\ \text{s.t.} & Ax & \geq & b \\ & x & \geq & 0 \end{array} \tag{A.1}$$

where $x \in \mathbb{R}^n$ is the vector of n decision variables, $c \in \mathbb{R}^n$ is the vector of objective coefficients, $b \in \mathbb{R}^m$ is the vector of lower bounds on constraints and $A \in \mathbb{R}^{m \times n}$ is the matrix of constraint coefficients. We use the convention of writing all vectors as column vectors.

If some variables must take integer values, then the previous problem becomes a (linear) *mixed integer program* (MIP) and the special case where all variables are integer is called *pure integer linear program* (ILP). It is well known that LPs can be solved in polynomial time and there are several efficient algorithms to solve this class of problems, such as the interior point method and the simplex method. Unlike LPs, MIPs and ILPs are NP-hard. However, thanks to the development of successful mathematical techniques and to high quality software, integer programming is nowadays a thriving area of optimization and is applied to a multitude of human endeavors. Indeed, a wide variety of practical problems can be modeled as integer programs: machine scheduling, production planning and crew rostering are a few examples. A number of solvers are commercially available to solve MIPs, such as the one used in this thesis, the CPLEX solver.

Research is currently very active in this field: beautiful and powerful mathematical results pervade the area of integer programming and two examples are Dantzig-Wolfe Decomposition with Column Generation and Benders' Decomposition.

The main idea behind these techniques is that solving a large MIP by the traditional branch-and-bound approach quickly becomes intractable as the number of variables and constraints increases. Unlike the traditional approach, multistage optimization algorithms divide the decision-making process into several stages and, in this manner, a series of small problems are solved instead of a single large problem.

Dantzig-Wolfe Reformulation of Integer Programs and Column Generation

Many problems can be formulated as integer programs. These formulations often result in very large and complicated models that, however, contain a lot of structure in the constraints. This structure allows for a Dantzig-Wolfe decomposition of the integer program, whose main idea is to decompose the constraint set $Ax \geq b$ of the relaxed original problem into two sets of constraints, $A_1 x \geq b^1$ and $A_2 x \geq b^2$, where the LP over $A_2 x \geq b^2$ can be easier to optimize than the original problem. Let consider the integer program:

$$\begin{array}{lrcl} \min\limits_{x} & c^T x & & \\ \text{s.t.} & A_1 x & \geq & b^1 \\ & A_2 x & \geq & b^2 \\ & x & \geq & 0 \\ & x & \in & \mathbb{Z}^n \end{array} \tag{A.2}$$

and let Q be the discrete set defined as

$$Q := \{x \in \mathbb{Z}^n \ \ : \ \ A_2 x \geq b^2, x \geq 0\},$$

so that the previous problem can be rewritten as

$$\begin{array}{lrcl} \min\limits_{x} & c^T x & & \\ \text{s.t.} & A_1 x & \geq & b^1 \\ & x & \in & Q \end{array} \tag{A.3}$$

The objective value of A.3 does not change if Q is replaced by its convex hull $conv(Q)$ and by the integrality condition. Therefore, it can also be formulated as

$$\begin{array}{lrcl} \min\limits_{x} & c^T x & & \\ \text{s.t.} & A_1 x & \geq & b^1 \\ & x & \in & conv(Q) \\ & x & \in & \mathbb{Z}^n \end{array} \tag{A.4}$$

For simplicity of notation, let assume that Q is non-empty and bounded. Then $conv(Q)$ is a convex polytope and, by Minkowski and Weyl theorems, every point in $conv(Q)$ is a convex combination of its extreme points. Hence, if $\{\xi^k\}_{k \in K}$ is a finite set of points in $conv(Q)$ containing all the extreme points, the relaxation of problem A.4 can be formulated as

$$\begin{array}{lrcl} \min\limits_{\lambda} & \sum\limits_{k \in K} (c^T \xi^k) \lambda_k & & \\ \text{s.t.} & \sum\limits_{k \in K} (A_1 \xi^k) \lambda_k & \geq & b^1 \\ & \sum\limits_{k \in K} \lambda_k & = & 1 \\ & \lambda & \geq & 0 \end{array} \tag{A.5}$$

and it is called the *Dantzig-Wolfe relaxation* of the integer program A.2. We denote problem A.5 under the name of *master problem* (MP). The *Dantzig-Wolfe reformulation* of the integer program is obtained from the Dantzig-Wolfe relaxation by enforcing the integrality conditions

$$\sum_{k \in K} \lambda_k \xi^k \in \mathbb{Z}^n \tag{A.6}$$

that, if we choose $\{\xi^k\}_{k \in K}$ to be the set of all points in Q, is equivalent to enforcing $\lambda_k \in \{0,1\}$ for all $k \in K$.

In general, Dantzig–Wolfe relaxation and reformulation decrease the number of constraints, but as counterpart they have a large number of variables, namely at least as many as the number of vertices of $conv(Q)$. However, this issue can be addressed thanks to Column Generation (CG). This technique strongly relies on the theory of LP and, in particular, on the observation that in a basic solution all non-basic variables have value 0. Since most of the variables will be non-basic, only some variables really need to be considered while solving the problem: CG is an efficient iterative method in which only a subset of all possible variables is considered in the MP at each iteration and the variables that have the potential to improve the objective function are generated on the fly and added for the next iteration. The master model restricted to this limited subset $K' \subset K$ of variables is called the *restricted master problem* (RMP). Initially, K' must be defined so that the feasibility of the RMP is guaranteed, otherwise dummy variables should be suitably introduced. When the RMP has been solved to optimality a MIP problem, the so-called *pricing problem* (PP), is used to check for the existence of a variable not yet in the RMP that, if added to the RMP, could possibly improve the solution value. Each of the constraints in the RMP provides a dual multiplier which can be used to determine the reduced costs of the potential new variables. The objective function of the PP is therefore the reduced cost and if this MIP results in a solution having negative reduced cost, the solution will be priced out and added to the RMP. If π_1 and π_0 are the vectors of shadow prices corresponding to the first and second symbolic constraints in the RMP respectively, then the PP can be formulated as the MIP

$$\begin{array}{lccl} \min\limits_{\xi} & (c^T\xi) - \pi_1^T(A_1\xi) - \pi_0 & & \\ \text{s.t.} & \xi & \in & Q \end{array} \tag{A.7}$$

Note that in general the pricing problem doesn't need to be solved to optimality, as long as a solution with negative reduced cost is identified.
CG stops when the optimal solution of the pricing problem PP has a non-negative objective value, meaning that no further improvement is possible. The solution to the RMP provided by CG does not satisfy the integrality constraints, hence at this point an integer solution can be obtained by integrating the Column Generation algorithm within a Branch-and-Bound framework (this technique is called Branch-and-Price) or by applying a heuristic algorithm.

A very interesting case is when the original integer problem presents a block structure, in which several blocks of constraints on disjoint sets of variables are linked by some linking constraints

$$\begin{array}{lrcll} \min\limits_{x} & \sum\limits_{j \in J} c^{(j)T} x^{(j)} & & & \\ \text{s.t.} & \sum\limits_{j \in J} D_j x^{(j)} & \geq & d & \\ & x^{(j)} & \in & Q^{(j)} & \forall j \in J \end{array} \tag{A.8}$$

where $Q^{(j)} := \{x^{(j)} \in \mathbb{Z}^{n_j} \;:\; A_j x^{(j)} \geq b^j, x^{(j)} \geq 0\}$.

If we remove the linking constraints, we are left with a decomposable problem that can be solved as a set of smaller independent subproblems. This is a very powerful observation, because the application of Column Generation to the Dantzig-Wolfe reformulation of this problem leads to one RMP

$$\begin{array}{lrcll} \min\limits_{\lambda} & \sum\limits_{j \in J} \sum\limits_{k \in K'(j)} (c^{(j)T} \xi^{(j),k}) \lambda_k^{(j)} & & & \\ \text{s.t.} & \sum\limits_{j \in J} \sum\limits_{k \in K'(j)} (D_j \xi^{(j),k}) \lambda_k^{(j)} & \geq & d & \\ & \sum\limits_{k \in K'(j)} \lambda_k^{(j)} & = & 1 & \forall j \in J \\ & \lambda_k^{(j)} & \geq & 0 & \forall j \in J, \forall k \in K'(j) \end{array} \tag{A.9}$$

and to $|J|$ independent pricing problems PP(j), expressed as

$$\begin{array}{lrcl} \min\limits_{\xi^{(j)}} & (c^{(j)T} \xi^{(j)}) - \pi_1^{(j)T} (D_j \xi^{(j)}) - \pi_0^{(j)} & & \\ \text{s.t.} & \xi^{(j)} & \in & Q^{(j)} \end{array} \tag{A.10}$$

whose size is much smaller than a single PP that incorporates all of them.

Benders' Decomposition

Benders' Decomposition (BD) is a mathematical technique consisting in splitting the original problem into a single MIP, called Benders' master problem (BMP), and a series of LP subproblems, called Benders' subproblems (BS). Given a solution of the BMP, every BS allows to generate either:

- an optimality cut, if that solution is feasible for the original problem but may be not optimal,
- a feasibility cut, if that solution is not feasible.

They are iteratively added to the BMP, until no more violated cuts are generated. Since cuts derive from duality arguments, the subproblems must be LP.
BD is similar to CG, with the difference that here at every iteration new rows are added instead of new columns.
The classic version of BD considers a generic MIP of the form

$$\begin{array}{lrcl} \min\limits_{x,y} & c^T x + d^T y & & \\ \text{s.t.} & Ax + By & \geq & f \\ & x & \in & \mathcal{X} \\ & y & \geq & 0 \end{array} \tag{A.11}$$

where we assume $\mathcal{X}$ to be discrete and B to be without empty rows. The matrix A may contain empty rows instead, so that constraints involving only continuous variables are included in the model. In general, B has a very special structure so that the problem, if in the y variable only, is a relatively easy problem. This MIP can be re-expressed as

$$\min_{x \in \mathcal{X}} c^T x + \hat{\eta}(x) \tag{A.12}$$

where $\hat{\eta}(x)$ is the LP

$$\begin{aligned} \hat{\eta}(x) \quad := \quad \min_{y} \quad & d^T y \\ \text{s.t.} \quad & By \geq f - Ax \\ & y \geq 0 \end{aligned} \tag{A.13}$$

and is called *Benders' subproblem* (BS). By LP strong duality, it can be equivalently formulated in terms of the dual vector π as

$$\begin{aligned} \hat{\eta}(x) \quad := \quad \max_{\pi} \quad & (f - Ax)^T \pi \\ \text{s.t.} \quad & \pi^T B \leq d^T \\ & \pi \geq 0 \end{aligned} \tag{A.14}$$

The feasible space of the dual subproblem has an interesting property, that is it does not depend on the variable x because x only appears in the slope of the objective function. Let $\mathcal{P}$ be the polyhedron representing this feasible space. If $\mathcal{P} = \emptyset$, the BS is infeasible for every x, meaning that the original problem is infeasible. Otherwise, in general it is not guaranteed that $\mathcal{P}$ is a polytope, hence there can be some directions of unboundedness. Let $\{\pi^k\}_{k \in K}$ and $\{r^j\}_{j \in J}$ be the extreme points and extreme rays of $conv(\mathcal{P})$, respectively. For a fixed value of $\hat{x}$, the slope of the objective function of the dual subproblem is $f - A\hat{x}$. Since it is a maximization problem, if there exists a direction r^j such that

$$(f - A\hat{x})^T r^j > 0$$

the dual subproblem is unbounded for $x = \hat{x}$, meaning that the primal subproblem BS is infeasible and so is the original problem. Hence $\hat{x}$ is not a feasible solution and must be discarded. On the other hand, if the dual problem is not unbounded for $x = \hat{x}$, the problem is feasible and the optimal dual solution is an extreme point, that is

$$\hat{\eta}(\hat{x}) \quad := \quad \max_{k \in K} \{(f - A\hat{x})^T \pi^k\} \quad = \quad \begin{aligned} \min_{\eta} \quad & \eta \\ \text{s.t.} \quad & \eta \geq (f - A\hat{x})^T \pi^k \quad \forall k \in K \end{aligned}$$

Therefore, if the original problem is feasible, the previous observations lead to the following reformulation

$$\begin{aligned} \min_{x, \eta} \quad & c^T x + \eta \\ \text{s.t.} \quad & \eta \geq (f - Ax)^T \pi^k \quad \forall k \in K \\ & 0 \geq (f - Ax)^T r^j \quad \forall j \in J \\ & x \in \mathcal{X} \end{aligned} \tag{A.15}$$

which is called *Benders' reformulation*.

The first two symbolic constraints determine the so-called *optimality cuts* and *feasibility cuts*, respectively. The complete enumeration of all these cuts is generally not practical.

Hence, Benders proposed an iterative algorithm that repeatedly solves the Benders' reformulation over the subsets $K' \subset K$ and $J' \subset J$, relaxing in this way the feasible region and leading to a new problem called *Benders' master problem* (BMP). At the beginning $K' = \emptyset$, $J' = \emptyset$ and the first BMP assumes the simple form

$$\begin{array}{rrcl} \min\limits_{x,\eta} & c^T x + \eta & & \\ \text{s.t.} & \eta & \geq & \bar{\eta} \\ & x & \in & \mathcal{X} \end{array} \tag{A.16}$$

where $\bar{\eta}$ is a lower bound on the variable η and the constraint $\eta \geq \bar{\eta}$ is needed as long as K' is empty in order to make the BMP bounded. For example, since we assumed that the variable y is non-negative, if also the vector d is non-negative then $\bar{\eta} = 0$ can be taken as a lower bound on $d^T y$. Once the optimal solution $\hat{x}$ of the BMP is found, the following *feasibility problem* is solved

$$\begin{array}{rrrcl} F := & \min\limits_{z,y} & z + 0^T y & & \\ & \text{s.t.} & Iz + By & \geq & f - A\hat{x} \\ & & y & \geq & 0 \\ & & z & \geq & 0 \end{array} \tag{A.17}$$

and, if $F = 0$, then the BS is feasible and its dual vector π leads to a new optimality cut $\eta \geq (f - Ax)^T \pi$ that is added to the BMP. If $F > 0$, the BS is infeasible and so is $\hat{x}$ for the original problem. In this case, the dual vector r of the feasibility problem satisfies $(1^T \quad 0^T) \geq r^T (I \quad B)$ and $0 < F = (f - A\hat{x})^T r$, meaning that r is a direction of unboundedness. Then, the new feasibility cut $0 \geq (f - A\hat{x})^T r$ is added to the BMP.
Every time the BMP is solved, its solution $(\hat{x}, \hat{\eta})$ leads to a new lower bound for the original problem because it is a relaxation of the full Benders' reformulation, that is

$$LB := c^T \hat{x} + \hat{\eta}.$$

Moreover, every time the BS is feasible, a new optimality cut is added and $(\hat{x}, \hat{\eta}(\hat{x}))$ is a feasible solution for the original problem, yielding a possible improvement in the upper bound

$$UB := \min\{UB, c^T \hat{x} + \hat{\eta}(\hat{x})\}.$$

Since it is difficult that the two bounds are identical due to numerical differences, it is customary to set a relative sufficiently small tolerance in the terminating condition: when $UB - LB \leq \epsilon$ is satisfied (or, if the relative error is used, $UB - LB \leq \epsilon \cdot |LB|$), then the iterative algorithm stops.

As for Column Generation, a very interesting case is when the matrix B presents a block structure, that is our original problem is in the form

$$\begin{array}{rrcll} \min\limits_{x,y} & c^T x + \sum\limits_i d^{(i)T} y^{(i)} & & & \\ \text{s.t.} & A_i x + B_i y^{(i)} & \geq & f^{(i)} & \forall i \in \mathcal{I} \\ & x & \in & \mathcal{X} & \\ & y^{(i)} & \geq & 0 & \forall i \in \mathcal{I} \end{array} \tag{A.18}$$

where $y^{(i)} \in \mathbb{R}^{n_i}$. Indeed, as the structure of the MIP suggests, Benders' Decomposition with multiple subproblems can be applied. The BMP then becomes

$$\begin{array}{rrcll} \min\limits_{x,\eta} & c^T x + \sum\limits_i \eta^{(i)} & & & \\ \text{s.t.} & \eta^{(i)} & \geq & (f^{(i)} - A_i x)^T \pi^{(i),k} & \forall i \in \mathcal{I}, \forall k \in K'^{(i)} \\ & 0 & \geq & (f^{(i)} - A_i x)^T r^{(i),j} & \forall i \in \mathcal{I}, \forall j \in J'^{(i)} \\ & x & \in & \mathcal{X} & \end{array} \tag{A.19}$$

and the $|\mathcal{I}|$ independent Benders' subproblems BS(i) can be formulated as

$$\begin{array}{rcrrcl} \hat{\eta}^{(i)}(x) & := & \min\limits_{y^{(i)}} & d^{(i)T} y^{(i)} & & \\ & & \text{s.t.} & B_i y^{(i)} & \geq & f^{(i)} - A_i x \\ & & & y^{(i)} & \geq & 0 \end{array} \tag{A.20}$$

whose size is much smaller than a single BS that incorporates all of them.

Appendix B

History of Other Implemented Approaches

During the 6-month internship in ASML Field Factory, many other approaches have been implemented but, once applied to the FF business case, they showed noncompetitive and unsatisfactory performances.
Hoping that this information could be useful for further research in the implementation of an efficient mathematical tool, in this appendix we present the main techniques that we applied, together with their performance difficulties.

Traditional Branch-and-Bound Approach

As already said in previous chapters, solving the FF manpower planning model, mathematically formulated in Chapter 2, using a traditional MIP Branch-and-Bound (B&B) approach was not possible, due to the huge size of the original formulation. In particular, the constraints related to travel and alternation between a work period abroad and a period at home turned out to be the most expensive for the solver.

Traditional Branch-and-Bound Approach but without explicit travel constraint in the model

A second approach to the problem consisted in trying to avoid the explicit assignment of travel days in the optimization phase. In particular, since the upgrades are assigned in terms of weeks (the so-called projects), it is reasonable to only keep track of a person location in a fixed day of the week (e.g. every Wednesday). The advantages of this approach are straightforward: travel days can still be implicitly estimated and used in the objective function, the arc approach representing the alternation between work period abroad and period at home location is now implemented by weeks instead of by days and the travel constraint is completely avoided, reducing the size of the problem. Since travel days are not assigned during the optimization phase, a boolean parameter is introduced into the model to check, for every pair of projects, whether it is possible to assign a person to both of them or whether there are not enough days to travel between their locations. The drawback of this approach is the assignment of travel days in post processing, because

these days take the place of days labeled as rest days in the optimization phase, reducing in this way the period at home location and not guaranteeing the minimal threshold anymore.
Even though the size of the original model was reduced a lot in this new approach, a traditional Branch-and-Bound technique turned out to scale very bad on our test instance representing the first quarter of 2017. Indeed, its implementation seemed to be promising for a 1-month horizon, taking only a couple of minutes to find the optimal solution, but turned out to be completely inefficient for a 3-month horizon: after 13 hours no solution was found and, moreover, the software stopped responding.

Column Generation with Heuristic

The inefficiency of a traditional approach suggested us the idea of applying a decomposition strategy. Decomposition techniques are powerful methods that rely on advanced mathematical concepts and that are applied a lot in scheduling problems. Two of the most famous decomposition techniques are Column Generation (CG) and Benders' Decomposition (BD). To apply CG to the FF DUV/YS Upgrades manpower planning problem, we observed that our original model assigns people to projects and defines for every employee an agenda of his daily locations such that the number of days at home location is maximized and the number of travel days is minimized. The only symbolic constraint that relates assignments of employees to projects with their location agendas is the constraint (2.4) and it is therefore called the linking constraint. Moreover, location agendas of different employees are independent of each other. Therefore, a Dantzig-Wolfe reformulation of the problem, together with a CG strategy, can be applied as follows:

- first, every location agenda is written as convex combination of some fixed agendas
- all the constraints that are only related to location agendas are deleted from the model, leading to the master problem of the CG
- every time the LP-relaxation of the Master Problem is solved, for every person a new agenda is generated on the fly thanks to a pricing problem $PP(p)$ that checks if there exists a new agenda that is not in the MP yet and that can improve the objective
- the Pricing Problems $PP(p)$ are MIP and, when no new agenda leads to improvement, the CG algorithm ends.

The result is a non-integer solution and some heuristic is needed in order to get an integer solution for the original problem.
A first heuristic we tried consisted in solving the master problem without integrality relaxation as soon as the CG algorithm stopped. This approach is described in Chapter 3.
A second approach we implemented consisted in inserting CG algorithm inside a loop and in fixing the agenda of one employee after every call to the CG procedure. Here we encountered an infeasibility issue due to the fact that our greedy heuristic, by fixing some agendas, can bring to an infeasible problem. To overcome this drawback, a backtracking procedure was implemented so that, in case of infeasible result, the algorithm could change the agenda previously fixed. An in-depth analysis of this technique showed

that lots of backtracking was necessary in our problem, leading to a completely inefficient procedure.
As last heuristic, we slightly modified the previous approach and fixed this time the assignment of employees to projects instead of the agendas. It was really good for medium-size instances but not for large data and it is better described in Chapter 3.

AIMMS built-in Benders' Decomposition procedures

Benders' Decomposition (BD) is another decomposition technique that splits the original problem in a Benders' master problem (BMP) and in one or more Benders' subproblems (BS). Before applying a combination of BD and CG as described in Chapter 3, we simply tried to apply BD to FF DUV/YS Upgrades manpower planning problem, meaning that the Benders' Subproblems were solved without CG. To implement BD, AIMMS provides 3 built-in procedures and allows also to use the Benders' algorithm by CPLEX. The implementation of BD using these 4 procedures is straightforward and the main differences between them are now briefly illustrated (for more information, see Chapter 21 in [12]):

- the **classic algorithm** solves at each iteration the BMP to optimality and then uses the BS to cut off the solution of the master problem by adding one or more constraints. This process of iteratively solving master problems and subproblems is repeated until no more cuts can be generated,

- the **modern algorithm** is known in the literature as *Branch-and-Benders-Cut* (B&BC) and centers on the observation that it is not necessary to solve the BMP to optimality to produce valid cuts. Hence, instead of building a new Branch-and-Bound tree at each iteration and spending time revisiting candidate solutions previously discarded, it only builds one search tree and BD is applied at every node of the tree where a feasible integer solution is found,

- the **two-phase algorithm** solves, during the first phase, the LP-relaxation of the BMP using the classic Benders' Decomposition algorithm. The generated cuts are added to the BMP and then it is solved as usual by applying the classic or modern algorithm to its MIP formulation,

- the **CPLEX Benders** is an implementation by CPLEX of the classic version of BD and it allows for multiple subproblems.

There are some limitations in applying these techniques to our problem, for example the first three procedures do not support multiple subproblems and, moreover, all these algorithms do not accept integer variables in the BS meaning that we had to relax our original subproblems. Due to the minimal effort required for their implementation, we decided to try all of them but they resulted in very bad performances. The main reason of this result is that they do not fit for our original problem because they do not make use of its particular structure.

Benders' Decomposition with Column Generation

In the end, we tried the same Benders' Decomposition techniques provided as built-in by AIMMS, but this time we solved the multiple subproblems by Column Generation. However, both classic Benders and modern Benders were not fast enough, while 2-phase Benders was promising and indeed is discussed in Chapter 3. Another interesting approach was trying to first solve the LP-relaxed original problem by CG. At this point the master problem obtained by CG contains some variables representing the convex coefficients for those agendas that were very useful to reduce the cost in the relaxed problem. Therefore, these agendas should be important also for the original MIP problem. That's why we decided at the end of CG to use all the master problem as BMP. This means that now BMP contains not only the assignment variables but also the variables of those agendas coefficients that were useful for the relaxed problem. In the Benders' subproblems, instead, we put all the other agendas that are not generated yet bu that can be generated during BD. Hence, we now apply BD to this BMP and we solve the Benders' subproblems by CG. The new agendas that will be generated in this second phase will not be inserted in the BMP as column but they will stay in the BS and will contribute to the BMP through the generation of Benders' cuts.
This new idea seemed very interesting but unfortunately gave bad results with all the BD implementations, even with the 2-phase Benders.

Appendix C

GUI Overview

In this appendix, an overview of the Graphical User Interface (GUI) for FF DUV/YS Upgrades manpower planning is presented. It allows to read input data from an Excel file, to update some information, to run the tool and, after the optimization phase, to visualize the PIs and export them into Excel. Fig. C.1 - C.7 show some of the pages that are accessible through the GUI.

Excel Input Data

The GUI takes as input an Excel file having 5 worksheets called Manpower, Skills, Locations, Upgrades and UpgradeTypes respectively. For a correct upload, the structure of these worksheets must be in compliance with specific requirements.

The Manpower worksheet contains columns with the following Excel named ranges:

i. *"PeopleRange"*, whose values should be unique because they identify the employees (a possibility can be full name together with its abbreviation),

ii. *"HomeLocationRange"*, containing the home location of every employee,

iii. *"HomeCountryRange"*, for the employees' home country,

iv. *"UpgradeFunctionsRange"*, specifying the employees' function (i.e. upgrade engineer, material handler, coordinator, generalist),

v. *"RestPeriodThresholdRange"*, containing the maximum number of days abroad that is considered as short period abroad. Usually its value is of 14 days,

vi. *"MinNumberOfRestDaysRange"*, representing the number of days at home location assigned to an employee after his short period abroad,

vii. *"MaxNumberOfRestDaysRange"*, representing the number of days at home location assigned to an employee after his long period abroad,

viii. *"MaxNumberOfDaysAbroadRange"*, for the maximum number of days that the employee can spend outside its home location,

ix. *"TeamsRange"*, specifying the name of the employee's team, if any.

The Skills worksheet contains the role matrix. For every pair (employee, role) the matrix entry can only assume values:

- 0, if the employee is not skilled for the role,
- 1, if the employee is skilled for the role,
- a date in the format yyww.d (e.g. 1702.3 refers to day 3 of week 02 of the year 2017), if the employee becomes skilled for the role from that date.

Moreover rows labels, columns labels and matrix entries are under the named ranges *"PeopleRange"*, *"RolesRange"* and *"SkillsRange"* respectively. Given the short-term horizon, it is reasonable to assume that an employee cannot switch from skilled to not skilled.

The Locations worksheet contains a matrix filled in with the number of travel days between every pair of locations. This number must be an integer value between 0 and 3, otherwise it is set by default to 0. Even though the matrix is expected to be symmetric, no error is thrown if the user prefers asymmetric values. Rows labels, columns labels and matrix entries are under the named ranges *"LocationsRange"*, *"LocationsBisRange"* and *"TravelingDaysRange"* respectively. Moreover, two columns with named ranges *"CountriesRange"* and *"RegionsRange"* are placed side by side with the row labels in order to specify the country and the region of every location.

The Upgrades worksheet contains columns with the following Excel named ranges:

i. *"MachinesRange"*, containing a unique identifier for the upgrades to be executed,

ii. *"UpgradeLocationsRange"*, specifying the location of the upgrades

iii. *"UpgradeStartDateRange"*, for the start date of the upgrade (in the format yyww.d such as 1702.3),

iv. *"UpgradeTypesRange"*, for the upgrade program (e.g. SNEP, PEP-K, etc.).

The UpgradeTypes worksheet contains a named range *UpgradeTypesRange* filled in with a list of all the upgrade programs, placed side by side with a named range *WeekDurationRange* that specifies the corresponding cycle time (expressed as number of weeks). Moreover, in this worksheet there is also a matrix having as row labels upgrade type and week number (within the cycle time), as columns the different roles and as entries the number of people with that role needed for every specific week of the given upgrade type. Rows labels, columns labels and matrix entries are under the named ranges *"UpgradeTypesAndWeeksRange"*, *"RolesRange"* and *"PeopleNeededRange"* respectively.

Excel Output Data

After the optimization takes place, the GUI gives also the possibility to export the output data in 2 Excel workbooks.

The values of all those variables and parameters defined in Chapter 2 that play a crucial role in the computation of the PIs are exported in the workbook PlanningOutputData.xlsx, each one in a different worksheet. The output is in sparse mode, that is only those rows having non-zero values will be shown in the workbook. In Excel, these values can be easily investigated through the creation of Gantt charts and pivot tables.

The results in terms of PIs are exported in a second Excel workbook called PlanningPerformanceIndicatorsData.xlsx, each one in a different worksheet. Because of the high amount of data, only the PIs indexed by just one index are exported with all their values. The remaining PIs are reported in a sparse mode and thus only non-zero values are shown. The creation of bar charts from the data is straightforward.

User Interface

The AIMMS software system has many built-in functions and display options to quickly produce a clear and user friendly interface. Our Graphical User Interface (GUI) for FF DUV/YS Upgrades manpower planning starts with a homepage from which it is possible to access all the other pages.

The page *Scheduling* contains a Gantt Chart, similar to the one currently used in Excel by planners, and some pivot tables with additional information about the employees assigned to each upgrade. From this page the user can set the time horizon for the analysis, upload an Excel file with the input data, run the tool and export the output in Excel. If the tool is run before an input file is uploaded, an error message is displayed to the user.

The page *Input Data for Employees* allows to add information about the days, within the time horizon, in which employees are not available because of holiday or any other activity already assigned. If a day is set to public holiday, no location needs to be specified because public holidays are only assigned if the employee is at home location. For the remaining cases, it is mandatory to specify also the location for that day, otherwise the location is assumed to be the home location. Another important assumption is that the locations in these days are consistent with the travel constraint of our model, that is the number of days among these activities should be enough to let the employee travel from one location to the other. If this is not satisfied, an error message is displayed to the user. Finally, it is important that the user checks whether the input data is consistent with the rest period constraint, because this is not directly checked by the tool during preprocessing.

The page *Info about Upgrades* shows the upgrades that occur during the specified time horizon and provides information about the roles required for them. This information is taken from the Excel input file but can be modified through this page.

The page *Info about Employees* displays information about the employees and their skill level as it is uploaded from the Excel input file. All this data can be here further modified.

The page *PI - Utilization* is a link to all the pages related to the computation of the Utilization. These pages contain charts, pivot tables and filters that allow to analyze the utilization of the employees according to the FF requests. In particular, it is computed on a weekly and monthly basis and in the whole time horizon.

The page *PI - Number of Travel* provides information on the Number Of Travels. The charts show the travels that occur on a monthly basis and in the whole time horizon, making a further distinction between local travels, regional travels that are not local and interregional travels.

The page *PI - Local Use* shows the values of local use for every upgrade in the specified time horizon, using either home location or home country to express what is considered as "local".

The page *PI - Travel Waste* provides charts, pivot tables and filters for a flexible investigation into the Travel Waste on a weekly and monthly basis.

The page *PI - Work Life Balance* is linked to the pages related to the analysis of work-life balance, on a weekly and monthly basis.

Finally, during the optimization phase, a Progress Window page is displayed so that the user is a little bit involved also in this technical phase and can see the GAP and the convergence trend of the current run towards the optimal solution.

An important note is that all the PIs are computed using the current ASML production calendar. This calendar never splits a week between two months or two years but applies the "Thursday rule" instead, that is every week is assigned to the same month (and, thus, year) as its Thursday.

Further Notes

Additional information about the tool, that can be useful for whoever wants to read its code and understand its implementation, is listed below:

- as already said in Chapter 2, the flow arcs are defined over a set $\mathcal{AD}_p$ of arc days and those having as second index a day before the days in $\mathcal{D}$ should be considered not as variables but as parameters defined somehow in input according to the plan already scheduled for the days before the tool horizon. In our current tool we assume that the planning horizon is not influenced by the past but starts from a blank sheet. Hence, those flow arcs that are not variables are set to 0 in our implementation in AIMMS by simply setting the first arc day equal to the first day and then we initialized the flow variables appropriately in the first day of the planning horizon, by setting the unit home arc to 1 and by reducing the index domain of the backward arcs. For further development of the tool so as to incorporate also partial plans, the first arc day should be changed accordingly and a procedure is needed in order to set the correct values of the flow parameters

- in order to let the user define a time horizon of his choice for the planning, a very big initial calendar has been created in AIMMS so that all the dates read in the input file do not give rise to any error in the tool. More precisely, this initial calendar contains all the dates from year 2016 to year 2030. If in the future the user wants to run the tool over a time horizon outside this range, then this predefined calendar needs to be modified accordingly.

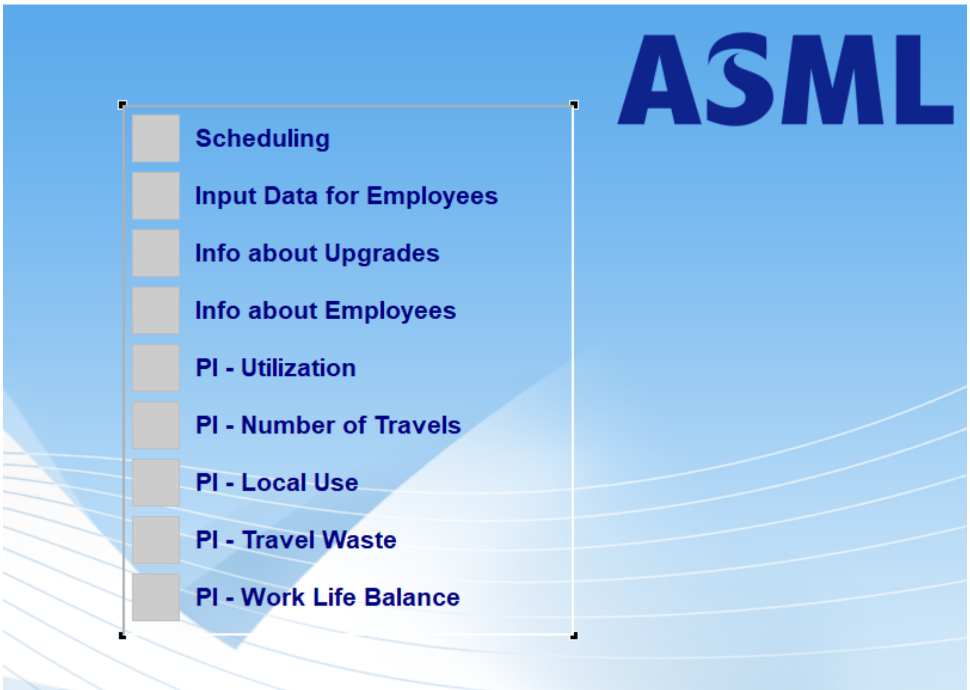


Figure C.1: GUI Overview - Homepage

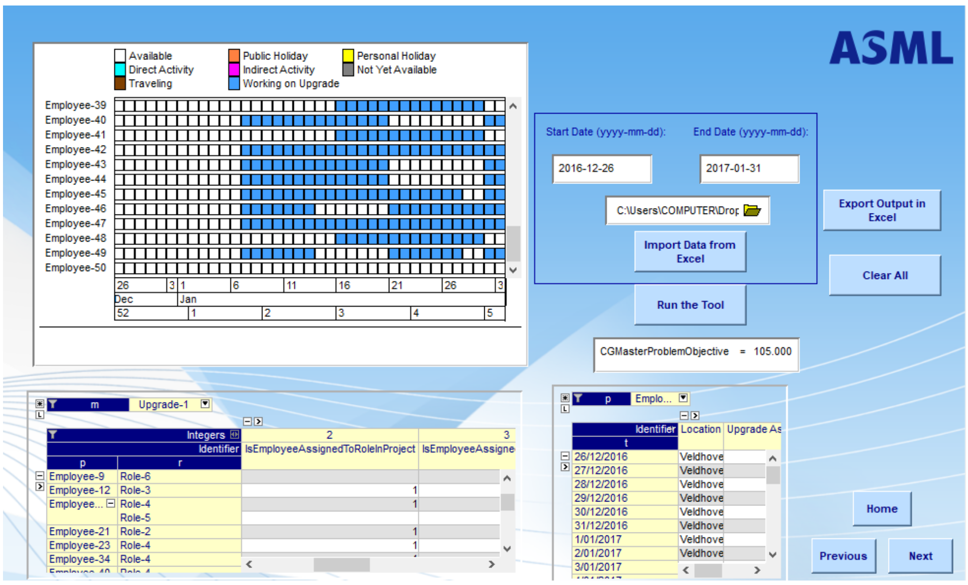


Figure C.2: GUI Overview - Scheduling page

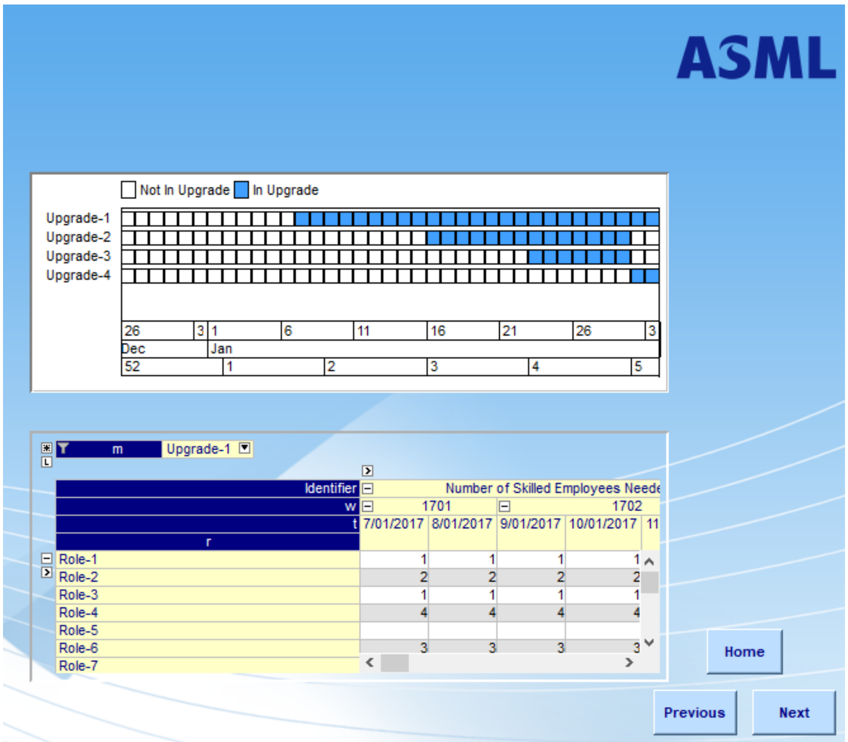


Figure C.3: GUI Overview - Info about Upgrades page

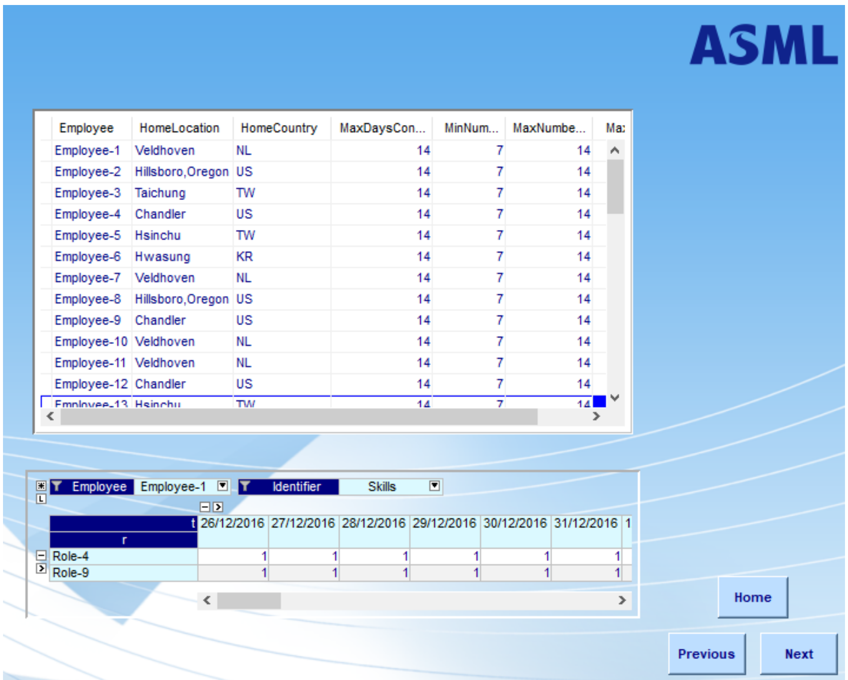


Figure C.4: GUI Overview - Info about Employees page

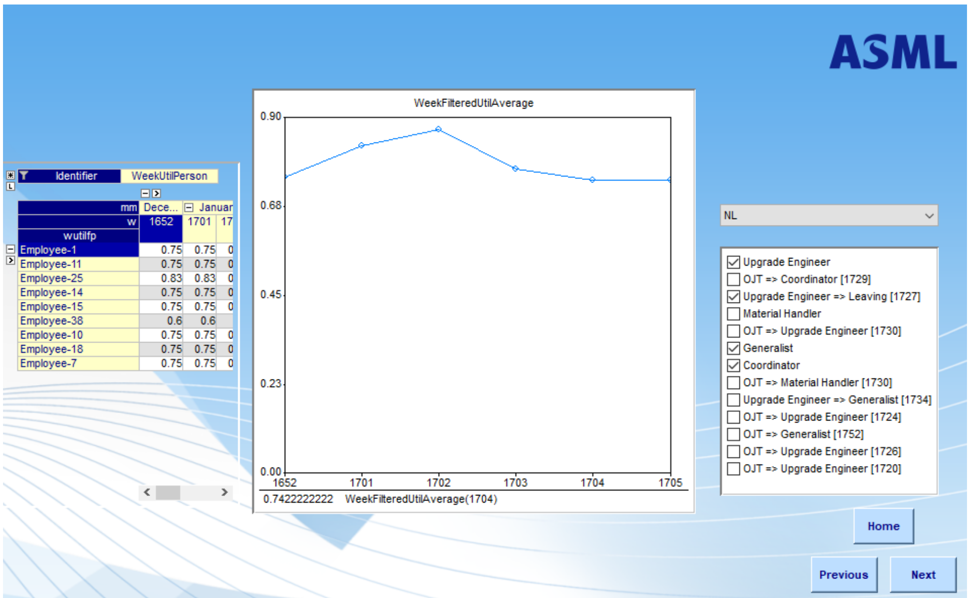


Figure C.5: GUI Overview - PI Weekly Utilization page

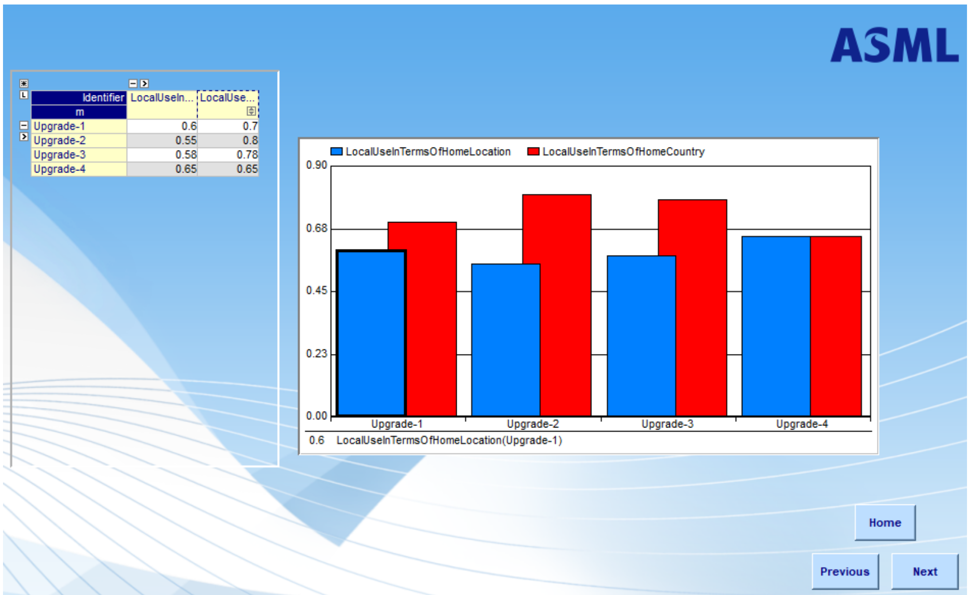


Figure C.6: GUI Overview - PI Local Use page

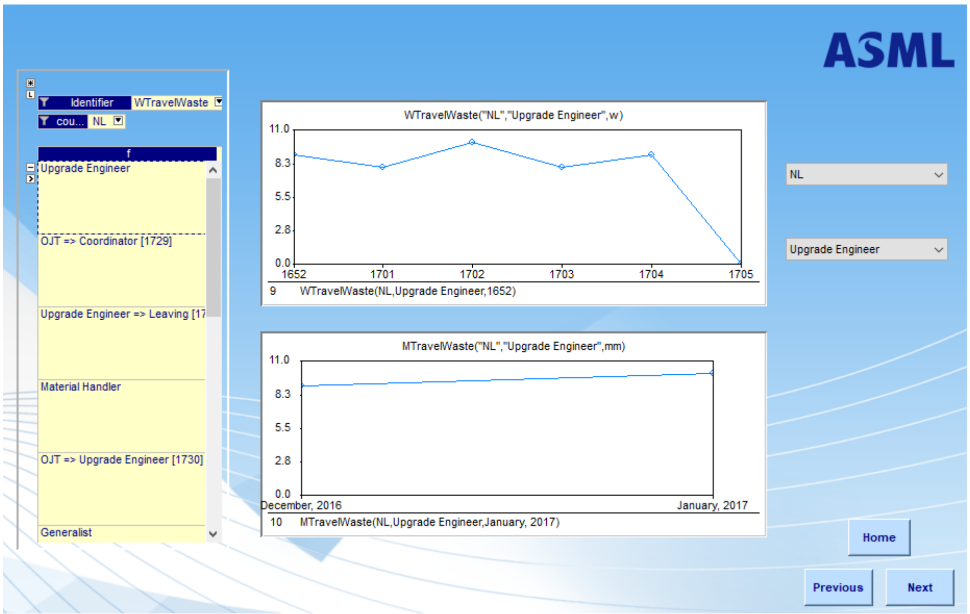


Figure C.7: GUI Overview - PI Travel Waste page

Appendix D

Performance Indicators: Summary

This appendix encompasses a summary of all the PIs defined within FF DUV/YS Upgrades department.

PLAN	**Name**	Utilization
	What is measured	Percentage of days of direct work over days of both direct and indirect work, for every direct-role employee
	Rationale	To measure the effective use of resources
	Target Type	Both Lower limit and Upper limit
DO	**Work Instruction**	$\frac{\#DaysDirectWork}{\#DaysDirectWork + \#DaysIndirectWork} \cdot 100$
	Comments	Public and personal holidays are excluded
	Filter	Total, by region (KR, US, ...), by function (Generalists, Coordinators, Upgrade Engineers, ...), by region & function, by employee
	Frequency	Weekly, monthly, in the whole tool horizon

Table D.1: Performance Indicators (PIs) - Utilization

PLAN	**Name**	Number of Travels
	What is measured	Number of travels for direct-role people
	Rationale	To get an indication for excess travel cost
	Target Type	Upper limit
DO	**Work Instruction**	Number of travels for direct-role people
	Comments	• Each individual flight booking is considered (e.g. traveling to and from location is counted as 2 travels) • It is different from the number of travel days
	Filter	Total and by distance (local, regional, interregional), by function, by home region, by function & home region. E.g. local is KR $\rightarrow$ KR, regional is TW $\rightarrow$ KR, interregional is US $\rightarrow$ KR
	Frequency	Monthly, in the whole tool horizon

Table D.2: Performance Indicators (PIs) - Number of Travels

PLAN	**Name**	Local Use
	What is measured	Percentage of people working locally (i.e. from their home location or home country), for every upgrade
	Rationale	To get more insight into how well the local staffing strategy is executed
	Target Type	Lower limit
DO	**Work Instruction**	$\frac{\#PeopleInUpgradeAndAtHomeLocation}{\#PeopleInvolvedInTheUpgrade} \cdot 100$ $\frac{\#PeopleInUpgradeAndInHomeCountry}{\#PeopleInvolvedInTheUpgrade} \cdot 100$
	Comments	The number of people involved in the upgrade includes the handover
	Filter	For every started or ongoing upgrade in the tool horizon (forecasting) For every completed upgrade in the tool horizon (reporting)
	Frequency	In the whole tool horizon

Table D.3: Performance Indicators (PIs) - Local Use

PLAN	**Name**	Travel Waste
	What is measured	Minimum between the number of people going outside their home country and the number of people entering in that country in the same time period
	Rationale	To investigate travel costs and staffing, planning, training and cross utilization strategy
	Target Type	Upper limit
DO	**Work Instruction**	MIN(IN,OUT) with IN=number of people entering in country C and having a different home country and OUT=number of people going outside their home country C
	Comments	• Going back to home location shouldn't be counted in IN • Only travel due to upgrades should be counted (no travel for indirect work)
	Filter	By home country, by function & home country
	Frequency	Weekly, monthly

Table D.4: Performance Indicators (PIs) - Travel Waste

PLAN	**Name**	Work-Life Balance
	What is measured	Percentage of work days "close" to home location, for every direct-role employee
	Rationale	To get a first insight into the work-life balance of employees
	Target Type	Lower limit (33%)
DO	**Work Instruction**	$\frac{\#DaysAtHomeLocation}{\#TotalDays} \cdot 100$ $\frac{\#DaysWithinHomeCountry}{\#TotalDays} \cdot 100$
	Comments	• Both direct and indirect work days and holidays are considered • "close" refers to either home location or home country
	Filter	Total, by home country, by function, by home country & function
	Frequency	Weekly, monthly

Table D.5: Performance Indicators (PIs) - WLB

Bibliography

[1] *AIMMS Tutorial for Professionals.* AIMMS B.V., 2016.

[2] *ASML Annual Report on Form* 20-*F.* ASML Holding N.V., pp. 16-24, 2016.

[3] J. Bisschop. *AIMMS Optimization Modeling.* AIMMS B.V., 2016.

[4] M. Conforti, G. Cornuéjols and G. Zambelli. *Integer Programming.* Springer, pp. 330-341, 2014.

[5] A.P. de Man. *Knowledge Management and Innovation in Networks.* Edward Elgar Publishing, pp. 54-64, 2008.

[6] G. Diepen. *Column Generation Algorithms for Machine Scheduling and Integrated Airport Planning.* Ph.D. Thesis, 2008.

[7] J. Hromkovic. *Algorithmics for Hard Problems: Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics.* Springer-Verlag, pp. 140-148,431-457, 2004.

[8] W. Laesanklang, D. Landa-Silva and J.A. Castillo-Salazar. *Mixed Integer Programming with Decomposition to Solve a Workforce Scheduling and Routing Problem.* International Conference on Operations Research and Enterprise Systems, pp. 283-293, 2015.

[9] S.J. Maher. *The Application of Recoverable Robustness to Airline Planning Problems.* Ph.D. Thesis, 2013.

[10] Y. Pochet and L.A. Wolsey. *Production Planning by Mixed Integer Programming.* Springer Series in Operations Research and Financial Engineering, Springer, pp. 185-204, 2006.

[11] R. Rahmaniani, T.G. Crainic, M. Gendreau and W. Rei. *The Benders decomposition algorithm: A literature review.* European Journal of Operational Research, 259(3), pp. 801-817, 2017.

[12] M. Roelofs and J. Bisschop. *AIMMS The Language Reference.* AIMMS B.V., 2017.

[13] L.A. Wolsey. *Integer Programming.* Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, pp. 185-194, 1998.