Analysis for Network Scenario Optimization Under Data Uncertainty: Algorithms, Computer Codes and Case Studies.

Antonio Manca

Ph. D. thesis in

Electronical and Computer Science

University of Cagliari

Abstract

Analysis for Network Scenario Optimization Under Data Uncertainty: Algorithms, Computer Codes and Case Studies.

Ph. D in Electronical and Computer Science

Supervisor: Prof. Alessandro Giua, Prof. Paola Zuddas

February 2006, 111 pages

Many dynamic planning and management problems are typically characterised by a level of uncertainty regarding the value of data input such as supply, capacity and demand patterns. Assigning inaccurate values to them could invalidate the results of the study. Consequently, deterministic models are inadequate for the representation of these problems where the most crucial parameters are either unknown or are based on an uncertain future. In these cases, the scenario analysis technique could be an alternative approach. Scenario analysis can model many real problems in which decisions are based on an uncertain future, where uncertainty is described by means of a set of possible future outcomes, called "scenarios". An example being a scenario analysis approach applied to water system management and air-traffic delay. In scenario analysis applied to water system management, data uncertainty can be modelled by a robust chance optimisation model obtaining a so-called barycentric value with respect to selected decision variables. The successive reoptimisation model based on this barycentric solution allows planning a part of the risk of making a wrong decision, reducing the negative consequences deriving from it. In scenario analysis applied to air traffic management, I have introduced a tactical optimization model for the ground delay problem and presented a computational experience by using a lagrangian decomposition approach and a scenario tree solver. Instances are based on a dynamic space-time network flow model with uncertainty on the capacity of the airport, due to environmental and human factors. In these contexts, scenario approach may be trustworthy because it permits decision makers to select the scenarios most appropriate to the situation and useful because it quickly provides robust solutions. Combination of open source codes, modular programming and standardization rules are very important as these permit us implementing and estending computer codes in order to give some advice to decision makers about the most advantageous methodological approach.

Key words: Optimization under Uncertainty, Network Flow, Air Traffic Management, Air Traffic Delay, Air Traffic Capacity, Water System Management, Min Cost Flow, Dynamic Networks, Robust Optimization, Decomposition Methods, Convex Analysis, Scenario Analysis, Risk Analysis, Operational Research, Management Science.

a mamma

Acknowledgements

I recognize that this research would not have been possible without the financial assistance of the University of Cagliari, the Department of Electrical and Electronic Engineering (DIEE) the Department of Land Engineering. I express my gratitude to those agencies. I have been fortunate to interact with many other people who have influenced me greatly, one of the pleasures of finally finishing is this opportunity to thank them. I would like to thank my tutors professor Paola Zuddas and professor Alessandro Giua whose expertise, understanding, and patience, added considerably to my PhD experience. I would like to thank professor Antonio Frangioni for his support in computer programming. I wish to thank the following professor for their help and collaboration on these three years: professor Alessandra Fanni, professor Carla Seatzu, professor Alexei Gaivoronski, professor Theodor Crainic, professor Giovanni Maria Sechi, professor Alessandro Olivo. I wish to remember in these rows professor Stefano Pallottino, my professor of "Theory and Optimization Methods" at University of Pisa, he left us in 2004. Finally, I am indebted to my many colleagues for providing a stimulating and fun environment in which to learn and grow. I am especially grateful to the following engineers: Francesca Salis, Maria Antonietta Loi, Massimo Di Francesco, Saverio Liberatore, Andrea Sulis for their supports and friendship.

Table of Contents

List of Tables

Table

List of Figures

Figure

Even a correct decision is wrong when it was taken too late.

Lee Iacocca 1924, CEO of Chrysler.

Chapter 1

Optimization Under Uncertainty: A General View

1.1 Introduction

This thesis focuses on issues related to integration of uncertainty into the mathematical model and the advantages offered by scenario analysis to give decision support. This is done in particular by considering network dynamic models that specifically incorporate uncertainty at some level involving data (for example: input, capacity, supply and demand). In many planning and management problems the decision maker will need to make choices that can not be put off or left to a later decision but "here and now". If exact information on the states of nature is not fully revealed, it would be impossible to predict the conditions a decision maker will face starting from a certain point in time [98]. Seeing that it would be difficult to deal with unpredicability, could be assumed that the predictions made by a Decision Support System are always accurate. In this case, no explicit account is taken for uncertainty in the predictions and that these predictions are updated over time. Decision managers must therefore allow for uncertainty by making intuitive adjustments based on their experience and expertise. Because this is not a satisfactory way to deal with the ubiquitous problem of uncertainty, it is important to determine how to satisfactorily incorporate this complexity into a decision model. An early approach in the investigation of these effects was the use of sensitivity analysis, unfortunately, as shown in [50], this approach shows a number of limitations and may provide misleading conclusions in respect of the nature of the solutions. Hence, in general sensitivity analysis is not a suitable approach for understanding the effects of random behaviour about the model parameters. Uncertainty can appear in the model as parameter having a nature of type either subjective or stochastic. Different methodological approaches have been proposed to handle uncertainty, in general they are classified as: distribution problems with the subclass "wait and see problems" and expected value problems, recourse problems with the two subclass distribution problems and scenario analysis based problems and change constrained problems [32].

1.2 Scenario Analysis

Scenario based methods are being affirmed as an useful methodological approach to handle uncertainty in order to offer a "robust" decision policy, in the sense the risk of wrong decisions is minimized and the solution can be accepted by decision maker [35]. There are many modelling and algorithmic approaches that could belong to scenario analysis based methods, like: stochastic programming with recourse ([96], [94]), chance constrained programming, stochastic control and dynamic programming, robust optimization [8]. Scenario Analysis is a scenario based method to handle with uncertainty, it gives the possibility to consider a fan, called **scenario tree**, of different possible configurations of the system, each one called **scenario**, which represents the possible sequence of realisations over the time horizon. Scenario analysis permits us to solve a problem associated to a system without having to consult each scenario separately. In general, the solution generated by scenario generation methods (conditional sampling, moment matching, sampling from specified marginals and correlations, path-based methods, optimal discretization) must not depend by particular scenario. In general, in order to test this property particular tests are often used that try to give a measure of the quality of a generated scenario tree, such as in_sample stability and out_of_sample stability ([26], [56]). For example, this is important when the system contains uncertain parameters with a continuous probabilistic distribution. In these cases the problems are difficult to solve, using scenario analysis, we build a scenario tree which is a discrete representation of the continuous problem, the loss of information of the scenario tree (discrete model) in respect to the continuous model, depends on the number of scenarios that have been introduced. This is important because the problem the "user time" often depends on the size of the scenario tree, but it is also true that a width tree produces a more robust information in output. If any probabilistic distribution is not known, a scenario tree can be built assigning to each scenario a parameter. It represents a some source of information related to the "importance" given to a single scenario in the scenario tree context. Because such models are usually very large and involve both integer and continue variables, problems with uncertainty are unmanageable directly by classical algorithms. One possible approach can be to implement new algorithms based on interior point methods [19]. Decomposition algorithms such as nested Benders decomposition [92], stochastic decomposition methods, importance Sampling, quasi gradient methods [34] are used as basis solver in order to propose new robust versions by exploiting the algebraic model.

1.3 Software Packages

The use of implement policies scenario analysis holds its strength when applied to practical problems as reported in bibliography. In this thesis I present two practical application inhering the usage of scenario approach to real world contexts, that show the effectiveness of such methods when uncertainty is included in such models. The analysis of the obtained results is often difficult to carry out without the use of database systems and customised friendly viewers. Up to now the largest effort has been dedicated to the development of solution techniques and related software systems suitable for the solution of large scale uncertainty programming model. Table 1.1 contains a list of software packages which deal with uncertainty problems ignoring the different stages of completion and use. Moreover, various research programs are in progress, such as in the Open Source Interface project (http://www.coin-or.org), developing a new tool called Stochastic Modelling Interface (SMI), for optimization under uncertainty SMI stands for Stochastic Modelling Interface. It is an interface for problems in which uncertainty and optimization appear together. SMI routines deal with commonly encountered programming issues, such as handling probability distributions, providing problem generation in SMPS (Stochastic Mathematical Programming System) stochastic standard format, interacting with solvers (such as the only open source linear programming solver CLP) to obtain solution information, etc.

This thesis focuses on the issues related to the potentiality of some open sources computer codes in order to build new tools that can deal with real problems including data uncertainty. Starting with a common platform and standard rules, "open source" codes and following standardisation rules are then implemented.

name	affiliation	system name
JJ Bisshop, et al.	Paragon Decision Technology	AIMMS
A Meeraus	GAMS	GAMS
B Kristjansson	Maximal Software	MPL
MAH Dempster	Cambridge University	STOCHGEN
E Fragniere	University of Geneva	SETSTOCH
A King	IBM	OSL/SE
HI Gassmann	Dalhousie University	MSLiP
G Infanger et. al.	Stanford University	DECIS
P Kall	University of Zurich	SLP-IOR
G Mitra	Brunel University	SPInE
A Gaivoronski	NTNU	SQG

Table 1.1: List of available commercial and non commercial packages

1.4 Decisions and deterministic models

It is important to dedicate some word about the crucial relationships between the times at which decisions are made, together with the moments at which we have certain data. If the uncertainty is solved before any decision is made, a deterministic model will be appropriate, otherwise uncertainty models occur to deal with unpredicability [49]. Often, linear deterministic models generally do not reflect the times at which decisions are made, nor do they distinguish between what will be known and what will remain uncertain when the decisions are made. Therefore it is essential to understand what and when a given parameter might change as it is critical to the acceptance of the methodology.

Chapter 2

Uncertainty in linear programming

This chapter introduces the classic definition of a deterministic model that can be associated at the "wait and see" approach in models containing uncertainty that is in strong relation with the "here and now" concept [98]. As is well known, a Linear Programming (LP) problem can be expressed in a compact standard form:

Model (P)

$$
\begin{aligned}
min\ cx \\
Dx = b \\
l < x < u
\end{aligned}
\tag{2.1}
$$

where data are classified as:

- $x \in \Re^m$ is the vector of decision variables;
- "cost" vector $c \in \Re^m$, in the objective function, whose components c_j can represent a cost, a benefit, a penalisation or a specific weight assigned by the manager to a unit of the variable x_j;
- a RHS (Right Hand Side) vector $b \in \Re^n$, in the constraints system,

whose component b_i can represent a supply or a demand associated to a node **i** , i.e., to the activity represented by node i;

- lower and upper bound vectors u and $l \in \Re^m$, whose components l_j and u_j represent lower and upper limits (possible zero and infinity, respectively) on the variable x_j by physical, technological, environmental and/or political, requirements;

matrix $D \in \Re^{n \times m}$ represents the coefficient matrix of the constraints system associated to vector of decision variables $\{x \in \Re^m\}$. In the deterministic approach certainty derived from available historical data submitted to statistical validation on the basis of a forecast and adopted as reference scenario. In deterministic optimisation model we assume that the decision maker has a perfect knowledge of the future evolution of each scenario without distinction among what is known and what remain uncertain. As a consequence, the solution obtained is strictly connected to the adopted scenario. Given a finite set $G = g_1, ..., g_s$ of predefined scenarios, for a specific scenario g the corresponding LP model, can be expressed as:

Model (Pg)

$$min\ (c_g x_g) \tag{2.2}$$

$$D_g x_g = b_g$$

$$l_g < x_g < u_g$$

The index g identifies vectors c, b, l and u of data related to the given scenario g. Moreover, x_g represents the vector comprehensive of all variables in scenario **g** and all constraints are represented as lower and upper bounds or in equation form. Using scenario analysis a set of different scenarios are considered together in order to obtain a set of decision variable on the whole set of scenarios. More

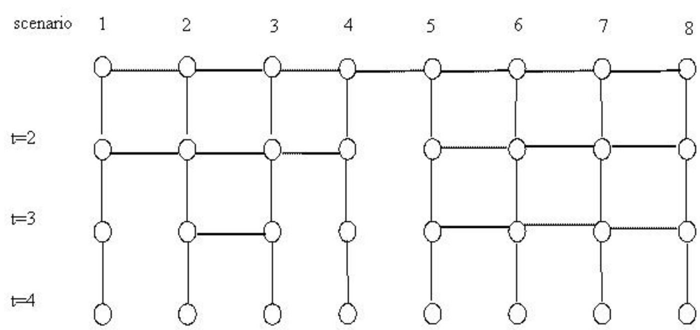

Figure 2.1: A set of parallel scenarios with different branching times

precisely, we consider a set of parallel scenarios which share at first a common part but, starting from a certain event one o more scenario could evolve in a their own manner. In the resulting mathematical model this relationships among scenarios is described by adding complicated constraints, it requires the subsets of decision variables, corresponding to the indistinguishable part of the different scenarios must be equal among them. These constraints are called in various manner: coupling constraints, non-anticipativity constraints, aggregation constraints. In scenario analysis the partial aggregation can be exploit in order to build a new structure to better manage scenarios and their aggregation: the scenario tree ([74],[97],[76]). In the scenario tree each node resume the partial description of a system and can contain one or more overlapped scenarios; a branch corresponds to add a new level to scenario depth and describes a possible evolution of either a single or a set of scenarios. Last horizontal arc joining two or more scenarios correspond to the last instant in which such scenarios are identical. In figure 2.1 are reported 8 parallel scenarios where horizontal lines among two or more nodes the non-anticipativity constraints. The solution obtained by scenario tree must be in any given time independent by information not yet available, this is done by introducing the non-anticipativity or congruity constraints. Hence, scenario tree follow two purposes: to specify the scenario generation and to define the algebric model structure including recursive non anticipativity restrictions. The mathematical model of the scenario aggregation can be resumed in the following structure:

Model (Pa)

$$min \sum_g (c_g x_g) \tag{2.3}$$

$$x_g \in R_g$$

$$x^s \in S$$

where x^s is the vector of variables submitted to congruity constraints, called also "wait and see" variables, $x_g \in R_g$ the set of standard constraints for each scenario $\mathbf{g} \in G$; $x^s \in S$, the set of congruity constraints. Resuming, the main

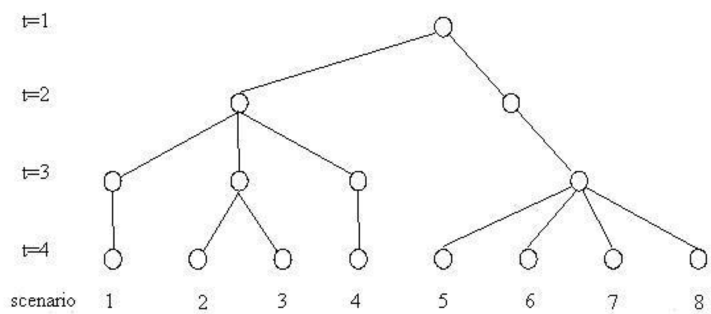

Figure 2.2: Scenario tree representation for the situation illustred in figure 2.1

rules adopted to organise a set of scenarios are:

- Branching: to identify branching-times τ as time-periods at which to bundle parallel scenarios, while identifying the stages at which to divide the scenario horizon.
- Bundling: to identify the number, β_τ, of bundles at each branching-time.
- Grouping: to identify groups, Γ_τ, of scenarios to include in each bundle.

The root of the scenario-tree corresponds to the time at which decisions have been taken (common to all scenarios) and the leaves of the scenario-tree represent the system performance at the end of the time-horizon. Each back path from a leaf to the root identifies a possible scenario.

Now, let t_0, t_1, t_2, t_3 ...., a set of increasing times in the horizon T and I a set of system components, a system is dynamic if each component $i, j \in I$ is associated to a discrete time $t \in T$, and if $x_{i,j}$ is a decision variable, $t_i \geq t_j$. In figure 2.1, fixed a scenario, each node represents a time t_i. Given ξ_g a scenario input uncertain parameter (i.e. capacity, deficit, supply), a set $\mathbf{G}$ of different scenarios and a time t, we define for each couples of scenarios the so called branching time $\tau(g_i, g_j) = max(t : \xi(g_i^t) = \xi(g_j^t),\ \xi(g_i^{t+1}) \neq \xi(g_j^{t+1}); g_i, g_j \in G,\ i \neq j)$, as the maximum time the input parameters ξ_{g_i} and ξ_{g_j} are identical. In this way, we obtain a set of branching time τ, the part of the dynamic system among two branching-times is called stage, the stages are ordered in increasing mode. Given a set of scenarios G, the time $min(\tau(g_i, g_j) : \ (i, j) \ \in \ G)$ divides the deterministic part by uncertain part of the dynamic problem. The deterministic part is called first stage. For illustrative purposes in the following model is given an example with 2 stages and 3 scenarios. In the model, the RHS vector b will be composed by subvectors b_1, b_2, b_3. The vector x will include the subvectors x_1, x_2 and x_3, the scenario variables. Each subvector x_g, g=1,2,3 is grouped in tree subvectors $x_g^1, x_g^{1,2}, x_g^2$, where x_g^1 describes the first stage variable, $x_g^{1,2}$ the so-called interlink variables which is included in the first stage and x_g^2 describes the variables include in the second stage. In general given a set of scenarios if it has been decided to move the branching time to an other instant of time then the uncertain parameters change. For a fixed scenario g we assume the constraint matrix associated to model Pa decomposed in A_g that represents the matrix of coefficient constraints describing the deterministic part, B_g the matrix of coefficient constraints for the second stage, $A_g^{1,2}$ and $B_g^{1,2}$, $g \ \in \ G$ the coefficient constraint matrix for the linking variables. the constraints matrix can be written as in table and assumes the shape reported in figure 2.1.

Model (Pe)

$$
\begin{aligned}
&\min \textstyle\sum_{i=1}^{3} c_i^1 x_i^1 + c_i^2 x_i^{1,2} + c_i^3 x_i^2 \\
&A_1 x_1^1 + (A_1^{1,2} + B_1^{1,2}) x_1^{1,2} + B_1 x_1^2 = b_1 \\
&A_2 x_2^1 + (A_2^{1,2} + B_2^{1,2}) x_2^{1,2} + B_2 x_2^2 = b_2 \\
&A_3 x_3^1 + (A_3^{1,2} + B_3^{1,2}) x_3^{1,2} + B_3 x_3^2 = b_3 \\
&I_1 x_1^1 - I_2 x_2^1 = 0 \\
&I_2 x_2^1 - I_3 x_3^1 = 0 \\
&I_{1,2} x_1^{1,2} - I_{1,2} x_2^{1,2} = 0 \\
&I_{1,2} x_2^{1,2} - I_{1,2} x_3^{1,2} = 0 \\
&l_1^1 <= x_1^1 <= u_1^1 \\
&l_1^2 <= x_1^2 <= u_1^2 \\
&l_2^1 <= x_2^1 <= u_2^1 \\
&l_2^2 <= x_2^2 <= u_2^2 \\
&l_3^1 <= x_3^1 <= u_3^1 \\
&l_3^2 <= x_3^2 <= u_3^2 \\
&l_1^{1,2} <= x_1^{1,2} <= u_1^{1,2} \\
&l_2^{1,2} <= x_2^{1,2} <= u_2^{1,2} \\
&l_3^{1,2} <= x_3^{1,2} <= u_3^{1,2}
\end{aligned}
\tag{2.4}
$$

Where c_g^1, x_g^1 represent respectively, the cost vectors and the vectors of the decisional variable associated to scenario g and first stage. $c_g^{1,2}$, $x_g^{1,2}$ represent respectively, the cost vectors and the vectors of the linking decisional variable associated to scenario g among stages 1 and 2. I_k and $I_k^{1,2}$ are two identity matrices defining the so called coupling constraints. Hence, the scenario analysis approach attempts to face the uncertainty factor by taking into account a set G, of different supposed scenarios corresponding to the different possible time evolution of some crucial data. Unlike simulation, the different scenarios can be

x_1^1	$x_1^{1,2}$	x_1^2	x_2^1	$x_2^{1,2}$	x_2^2	x_3^1	$x_3^{1,2}$	x_3^2
A_1	$A_1^{1,2}$							
	$B_1^{1,2}$	B_1						
			A_2	$A_2^{1,2}$				
				$B_2^{1,2}$	B_2			
						A_3	$A_3^{1,2}$	
							$B_3^{1,2}$	B_3
I_1			$-I_2$					
	$I_1^{1,2}$			$-I_2^{1,2}$				
			I_2			$-I_3$		
				$I_2^{1,2}$			$-I_3^{1,2}$	

Table 2.1: Matrix of constraints in a 3 - scenario tree and 2 stages

considered together to obtain a global set of decision variables on the whole set of scenarios. More precisely, two scenarios sharing a common initial portion of data must be considered together and partially aggregated with the same decision variables for the aggregated part, in order to take into account the possible evolutions in the subsequent non-common part. The aggregation rules guarantee that the solution in any given period is independent of the information not yet available. In other words, model evolution is only based on the information available at the moment and, if necessary, scenario modification is allowed. The problem supported by the scenario tree, is described by a mathematical model that includes all single-scenario problems (Pg), $g \in G$, plus some inter-linking scenario constraints representing the requirement that if two scenarios g_1 and g_2 are identical up to time τ on the basis of information available at that time, then the corresponding set of decision variables, x_1 and x_2, must be identical up to time τ. This means that the subsets of decision variables corresponding to the indistinguishable part of different scenarios must be equal among themselves. In table 2.2, the constraint matrix structure is shown after the first stage decision variables are grouped, it is visible the structure called L-shaped. For each scenario a "weight" can be assigned representing the "importance" given by the manager to the running configuration. In the model the weights can be viewed as the probability of occurrence of the examined scenario. More often they are determined on the basis of background knowledge about the system. The resulting mathematical model is named "chance-model" to indicate that it is not stochastically based but, due to the impossibility of adopting probabilistic rules and/or to the necessity of inserting information that cannot be deduced by historical data, it attempts to represent the set of possible performances of the system, as uncertain parameters change. The chance model can be expressed as the collection of one deterministic model for each scenario $g \in G$ plus a set of

x^1	$x^{1,2}$	x^2_1	x^2_2	x^2_3
A_1	$A^{1,2}_1$			
	$B^{1,2}_1$	B_1		
A_1	$A^{1,2}_1$			
	$B^{1,2}_2$		B_2	
A_1	$A^{1,2}_1$			
	$B^{1,2}_3$			B_3

Table 2.2: Matrix of constraints in a 3 - scenario tree and 2 stages. First stage variables are grouped

congruity constraints representing the requirement that the subsets of decision variables, corresponding to the indistinguishable part of different scenarios, must be equal among themselves. In this case, the chance mathematical model has the following structure:

Model (Pc)

$$min\ (\sum_{g} w_g c_g x_g) \tag{2.5}$$
$$D_g x_g = b_g,\ \ \forall\ g\ \in\ G$$
$$l_g \leq x_g \leq u_g,\ \ \forall\ g\ \in\ G$$
$$x^s \in S$$

Where: w_g represents the vector of weights assigned to a scenario $g\ \in\ G$; x^s represents the vector of variables submitted to congruity constraints; $x^s \epsilon S$, the set of congruity constraints. Regarding weight definitions, if the manager were able to evaluate the weight w_g as the probability that scenario g will occur, he could estimate it by some stochastic technique or statistical test. More often the manager has no, or a few, possibilities to do this due to the difficulty in deriving a probabilistic rule from conceptual considerations. Instead, in scenario analysis, a weight w_g assigned to a scenario g can be interpreted as the "relative importance" of that scenario in the uncertain environment. In other words, in scenario analysis, weights are interpreted as subjective parameters assigned on the basis of the experience of management. Different weights can also be assigned to different stages. Then, the definitive weight in the objective function will be calculated considering the contribution of scenarios and stages. A good compromise in weight settlement might be to assign scenario importance on the basis of subjective considerations, and assign weights to stages on the basis of statistical tests. Deterministic equivalent model can be solved by using decomposition iterative algorithms such as Benders decomposition, Lagrangian relaxation techniques, bundle methods, cutting plane methods and their variants. Such exploit the special structure of constraints and are also called rows generation methods because at each iteration insert a new row of constraints that restrict the research area, such methods approach the problem under a dual point of view. Dantzig-Wolfe approach is a column generation methods, at each iteration. The mentioned methods are equivalent among them, it is possible starting from the structure model (for example Bender's scheme), by making some simple algebra, recognizing the structure of an alternative scheme. When the problem is huge, resort to parallel computing can be adopted, such as in ([79],[15], [36]) whilst in [57] has been proposed a decomposition algorithm running on a computational grid. Because constraints make the model redundant we have seen that some scenarios are overlapped in some time-periods and some variables are redundant as a consequence, the model components (variables and constraints), that are associated to overlapped scenarios, can be reported only once, this get us to introduce the so-called deterministic equivalent problem, table 2.2 shows the associated matrix constraints representation. The optimal solution x^* in (Pc) hedge against all possible events or scenario g that can occur. Model (Pc) is know as a multistage uncertainty problem with recourse. The model is supported by a scenario tree of multiperiod stages, subjected to non-anticipativity constraints, a decision made in time t should take into account all future achievements of uncertainty parameters. In other words a final decision at time t should be taken only after decisions at time $t+1, t+2, ...$ are decided.

2.1 The deterministic equivalent model

The vector b_g in (Pc), can be decomposed in b_1, the deterministic part and the vector d_g that contain uncertainty. Since, in (Pc) redundant first stage variables,considering matrix constraints in 2.2, we can rewrite the first stage of (Pc) as:

Model (Pc_r)

$$min\ c_1^1 x_1^1 + c_1^{1,2} x_1^{1,2} + \sum_g w_g c_g^2 x_g^2 \tag{2.6}$$

$$A_1 x_1^1 + A_1^{1,2} x^{1,2} = b_1 \tag{2.7}$$

$$B_g x_g^2 + B_1^{1,2} x_1^{1,2} = d_g, \ \ \forall\, g \ \in \ G \tag{2.8}$$

$$0 \leq x_g, \ \ \forall\, g \ \in \ G$$

Where, I supposed $\sum_g w_g = 1\,, g \in G$ and scenarios have the same first stage costs. this permits to use c_1^1 and $c_1^{1,2}$ in model Pc_r. In general, if a distribution of probability does not exist, given a "chance" or "weight" w_g, the cost vector associated to x_1^1 is computed as $\sum_g w_g c_g^1$. The linking vector of variables $x^{1,2}$ of proper dimension appear in both the set of constraints in 3.29. Moreover I define the convex sets (closed and bounded):

$$X = \{(x_1^1, x_1^{1,2}) \mid A_1 x_1^1 + A_1^{1,2} x_1^{1,2} = b_1\} \tag{2.9}$$

and

$$X_g = \{(x_1^{1,2}, x_g^2) \mid B_g x_g^2 + B_1^{1,2} x_1^{1,2} = d_g, \ \forall\, g \ \in \ G\} \tag{2.10}$$

and $Conv(X) = \{(x_1^{1,2}(t), x_1^1(t)) \mid t = 1, ..v\}$ and $Conv(X_g) = \{(x_1^{1,2}(t), x_g^2(t) \mid t = 1, .., v_g\}$ respectively the extreme points of the convex sets X (convex hull) and $\{X_g \mid g \ \in \ G\}$. The dual model for Pc_r can be written in the following manner:

Model (DPc_r)

$$max\ yb_1 + \sum_g \pi_g d_g \tag{2.11}$$

$$A_1^T y \leq c_1$$

$$(A_1^{1,2})^T y + \sum_g (B_1^{1,2})^T \pi_g \leq c_{1,2}$$

$$(B_g^2)^T \pi_g \leq w_g c_g^2, \ \forall \ g \ \in \ G$$

$$y, \ \pi_g : unconstrained, \ \forall \ g \ \in \ G$$

Where, I designated as y, π_g, with $g \ \in \ G$ respectively, the dual variable vectors associated to constraints 2.7 and 2.8 in Pc_r. Deterministic equivalent model can be rewritten as a sum of two linear programming problems which have in common the interlink variables, as follow:

Model (Pc_{r1})

$$min \ c_1^1 x_1^1 + c_1^{1,2} x_1^{1,2} + \sum_g w_g Q(x, g) \tag{2.12}$$

$$A_1 x_1^1 + A_1^{1,2} x_1^{1,2} = b_1$$

$$x_1^1 \geq 0$$

$$x_1^{1,2} \geq 0$$

where: $PQ(x, g)$

$$Q(x, g) = \kappa'(d_g - B_g^{1,2} x_1^{1,2}) = min \ c_g^2 x_g^2 \tag{2.13}$$

$$B_g x_g^2 = d_g - B_g^{1,2} x_1^{1,2}, \ \forall \ g \ \in \ G$$

$$x_g^2 \geq 0, \ \ \forall g \ \in \ G$$

For a given scenario $g \ \in \ G$, the dual problem of PQ(x,g), can be state as:

PDQ(π,g)

$$DQ(\pi\text{,g}) = \kappa(d_g - B_g^{1,2} x_1^{1,2}) = max \ \pi_g (d_g - B_g^{1,2} x_1^{1,2}) \tag{2.14}$$

$$\pi_g B_g \leq c_g^2, \ \ \forall \ g \ \in \ G$$

$$\pi_g, \ \ unconstrained$$

The function Q(x,g) also referred to as the "recourse function", is in turn defined by the linear program. Recourse function include the "technology matrix" B_g, also known as the "recourse matrix", the right-hand side d_g, the objective function coefficient c_g^2 contains uncertain parameters. The presence of uncertainty has effect in the dimension of the problem, as the number of scenario grows, the full problem could become hard to solve with standard algorithms such as simplex methods. Some constraint can be relaxed and decomposition methods can be useful to implement quantitative methods. Such methods return a lower bound for the full problem. In optimization under uncertainty, convex properties can be exploited in order to implement decomposition based methods can be adopted. In fact, let $\gamma \in (0,1)$, it is possible to show that $\kappa(z)$, the function defined in 2.15 is a piecewise linear concave function in z, that is:

$$\begin{aligned} &\gamma(\kappa(z_1)) + (1-\gamma)(\kappa(z_2)) \\ &\leq \kappa(\gamma(z_1) + (1-\gamma)z_2) \end{aligned} \tag{2.15}$$

Hence, if $\kappa(z_1), ..., \kappa(z_t)$ is an arbitrary set of concave functions also

$$max_{i:i=1,..,t}\kappa(z_i)$$

is concave.

Moreover, let z_1 and z_2 two solutions for the function Q(z,g) is a piecewise linear convex function in z, that is:

$$\begin{aligned} &\gamma Q(z_1, g) + (1-\gamma)Q(z_2, g) \\ &\geq Q(\gamma z_1 + (1-\gamma)z_2, g) \end{aligned} \tag{2.16}$$

Hence, the sets X_g are convex closed polyhedron [96], for a complete understand [82]. $Q(x,g)$ is also piecewise because the set of items of the admissible region is finite. Objective function convexity (or concavity) is important as (Pc_{r1}) can in principle be solved by a method for piecewise linear problems or by a general algorithm for constrained non-smooth optimization. Although, the pieces of $\kappa'(x)$ and the facets of X_g are not given explicitly, it is possible to extract from $Q(x,g)$ at successive points $x_1^{1,2}(1), x_1^{1,2}(2), x_1^{1,2}(3)$... information about the piece of $\kappa'(x_1^{1,2})$ [82]. Hence, let $\Pi_g = \{\pi_g^l : l = 1,..,v\}$ the set of extreme points for DQ(x,g), each of these points is potentially an optimal solution for (Pc_{r1}). In fact, we know that no optimal solution better occur in an extreme point. For a given achievement of g, the corresponding recourse action x_g^2 is obtained by solving the problem PQ(x,g). Assuming $x_1^{1,2}(1),\ x_1^1(1)$ is the optimal result inhering the first stage problem, we can formulate the so called "Master problem":

(Pc_{r2})

$$\begin{aligned} min \sum_g c_1^1 x_1^1 + c_1^{1,2} x_1^{1,2} \\ A_1 x_1^1 + A_1^{1,2} x_1^{1,2} = b_1 \\ x_1^1 \geq 0 \\ x_1^{1,2} \geq 0 \end{aligned} \tag{2.17}$$

Once solved the master problem, it is possible to obtain a current approximation of Q(x,g) by fixing $x_1^{1,2}(1)$, the interlink primal vriables result at first iteration. Now, if we set Q(x)= $\sum_g w_g Q(x,g))$, we can observe that $u = \sum_g -B_g^{1,2} w_g \pi_g^1$ is a sub-gradient for Q(x) in the point $x_1^{1,2}(1)$, where π_g^1, for each scenario, is the optimal result obtained by solving the dual problem $PDQ(\pi, g)$, obtained at first iteration. For sub-gradient inequality we have:

Q(x,g) $\geq Q(x_1^{1,2}(1), g) - \pi_g^l B_g^{1,2}(x - x_1^{1,2}(1))$ in other words $\sum_g Q(x_1^{1,2}(1), g) - \pi_g^l B_g^{1,2}(x - x_1^{1,2}(1))$ is a supporting hyperplane of Q(x)(2.3) at point $x = (x_1^{1,2}(1))$, where $Q(x_1^{1,2}(1), g) = c_g^2 x_g^2(1) = DQ(x_1^{1,2}(1), g) = (d_g - B_g^{1,2} x_1^{1,2}(1))^T \pi_g^1$. This

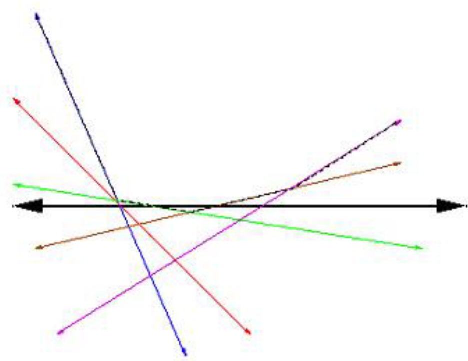

Figure 2.3: A set of supporting hyperplanes

suggest to build up, an increasingly better approximation of Q(x), by adding to each step a new supporting hyperplane, i.e. an other row of constraints is added to the master problem 2.20. Hence, given a set of $v-1$ sub-gradients of Q(x),$u_1 \in \delta Q(x_1^{1,2}(1))$, $u_2 \in \delta Q(x_1^{1,2}(2))$,$u_3 \in \delta Q(x_1^{1,2}(3))...u_{v-1}\delta Q(x_1^{1,2}(v-1))$, we can obtain a current approximation of starting problem obtaining new values $x_1(l)$, $x_2(l)$, $x_1^{1,2}(l)$, $l = 1,..,v-1$, at each iteration. Then, the current approximation to the original problem (Pc_r) is:

(Pc_{c2})

$$min \sum_g c_1^1 x_1^1 + c_1^{1,2} x_1^{1,2} + \theta \tag{2.18}$$

$$A_1 x_1^1 + A_1^{1,2} x_1^{1,2} = b_1$$

$$\theta \geq Q(x_1^{1,2}(l)) + u_{(l,g)}(x_1^{1,2} - x_1^{1,2}(l)), \quad l = 1, .., v-1$$

$$x_1^1 \geq 0$$

$$x_1^{1,2} \geq 0$$

As Q($x_1^{1,2}(l)$) $=\sum_g w_g (d_g - B_g^{1,2} x_1^{1,2}(l))^T \pi_g^l - \pi_g^l B_g^{1,2}(x - x_1^{1,2}(l))$,the sub-gradient inequality $Q(x_1^{1,2}, g) \geq Q(x_1^{1,2}(l), g) - \pi_g^l B_g^{1,2}(x_1^{1,2} - x_1^{1,2}(l))$ can be reduced to $Q(x_1^{1,2}, g) \geq Q(x_1^{1,2}(l), g) - \pi_g^l B_g^{1,2}(x_1^{1,2} - x_1^{1,2}(l))$ and (Pc_{c2}) can be restated as:

$$min \sum_g c_1^1 x_1^1 + c_1^{1,2} x_1^{1,2} + \theta \tag{2.19}$$

$$A_1 x_1^1 + A_1^{1,2} x_1^{1,2} = b_1$$

$$\theta \geq \sum_g w_g \pi_g^l d_g - B_g^{1,2} x_1^{1,2}, \quad l = 1, .., v-1,$$

$$x_1^1 \ \geq \ 0$$

$$x_1^{1,2} \ \geq \ 0$$

The structure of (Pc_r) model can be exploited in order to build an algorithm producing a new cut for the “Master” problem to each iteration and returning both a lower bound and an upper bound for the (Pa) problem. This is known as “Benders decomposition” or “L-shaped” algorithm. The general idea which is applied in such algorithms is that the ”Master” problem starts with computing a solution where the complicating constraints involving x_g^2, are relaxed in a manner that the starting problem is composed only of simple constraints involving x_1^1 and $x_1^{1,2}$, the first stage variables. The dual information obtained in the second stage problems, by solving the inner problems, is used in the first stage in order to produce a new constraint involving both first stage variables, interlink variables and θ. The solution of the outer problem produces a lower bound for the original problem because first stage variables deal with less constraints but, if the new solutions satisfy the constraints in (Pc_{r1}), we have done. Once fixed first stage variables, the inner problem produces an upper bound for (Pc_{r1}) and a new dual vector which will produce a new cut in the "Master" problem. If at the current step l the master problem proposes the point $x_1^{1,2}(l)$ and it produces infeasibility in the inner problem for each scenario, this new point $x_1^{1,2}(l)$ introduces a feasible cut which cut-off the current point. If the inner problem is feasible, all the artificial variables $\mathbf{t}$ are out the final basis. Otherwise, the problem Pc_{rx} has solution:

Model $Q(x, g, t)$

$$min\ et \tag{2.20}$$

$$B_g x_g^2 + It = d_g - B_g^{1,2} x_1^{1,2},\ \forall g\ \in\ G$$

$$x_g^2\ \geq\ 0,\ t\ \geq\ 0$$

where e is a row vector (m-dimensional) with all coefficients 1. We can observe that the supplemental problem is feasible and $et \geq 0$. If the solution is zero, we can take out all the auxiliary variables t and we have a solution to the first system. If the minimum is greater then zero, applying the duality theorem to PQ(x,g), we can assert that a vector u exists such that $u(d_g - B_g^{1,2} x_1^{1,2}) > 0,\ uB_g \leq 0,\ uI \leq e$. In order to prevent infeasibility $x_1^{1,2}(l)$ again, we set $u = \pi_g^l$ and introduce the feasibility cut $\pi_g^l(d_g - B_g^{1,2} x_1^{1,2}) \leq 0$ in the master problem. Thus for a scenario g, a matrix basis R exists such that $R[x_g^2\ \ t_R]^T + Nt_N = d_g - B_g^{1,2} x_1^{1,2}$, where t_N are the artificial variables settled to zero. If the optimal value of the problem is positive, then inequality $\pi_{v,g}(d_g - B_g^{1,2} x_1^{1,2})$ is a feasible cut for problem PQ(x,g). Let $t_i, t_i \geq 0, i = 1, .., m$ be the largest value artificial variable. Multiplying by the inverse of the basis matrix and by the i-th unit vector e_i we obtain $t_i = e_i^T R^{-1}(d_g - B_g^{1,2} x_1^{1,2})$. Finally, a variable change $\delta x = x_1^{1,2}(l) - x$ of the trial point $x_1^{1,2}(l)$ is needed to reduce t_i is reduced to zero, by requiring that $e_i^T R^{-1}(d_g - B_g^{1,2} x_1^{1,2}) \leq 0$. In the following general algorithmic scheme we shown the main step of Benders decomposition algorithm:

```
\* Initialization *\
```

l = 1: (iteration counter);

UB = MAXINT (Best Upper Bound);

LB = -MAXINT (Best Lower Bound);

```
\* Solve the initial master problem ( first stage ) *\
```

$$min\ c^1 x_1 + c^{1,2} x_1^{1,2}$$

$$A_1 x_1^1 + A_1^{1,2} x_1^{1,2} = d$$

$$x_1 \geq 0$$

$$x_1^{1,2} \geq 0$$

```
\* Check if the initial master problem is feasible *\
```

if $(x_1(1) \cup x_1^{1,2}(1))$

```
\* read the optimal value of the master problem *\
```

$X_1 = x_1(1) \cup x_1^{1,2}(1)$;

else

Stop; (Problem Unfeasible);

```
\* Main Loop *\
```

While $((UB - LB)/(1 + LB) \geq TOL)$

(

```
\* Solve the second stage problem *\
```

for ($g \in G$)

do

(

$$min\ c_g^2 x_g^2 \tag{2.21}$$
$$B_g x_g^2 = d_g - B_g^{1,2} x_1^{1,2}$$
$$x_g^2 \geq 0$$

$\pi_g^{l\,2} = \pi_g^2$ $(return\ optimal\ dual\ value\ vector)$;

$if(second\ stage\ problem\ is\ feasible)$

$\mathrm{x}_{(l,g)}^2 = x_g^2$ $(return\ optimal\ primal\ value\ vector)$;

$else$

$\pi_g^{l\,2}(d_g - B_g^{1,2} x_1^{1,2}) \leq 0$ $(return\ a\ feasibility\ cut)$;

) (end for;)

if $(All\ the\ second\ stage\ problems\ return\ feasible\ solution)$

$UB = min(UB, \sum_g [c^1, c^{1,2}] X_l + c_g^2 x_g^2(2))$; $(update\ upper\ bound)$;

```
\*  Solve the master problem *\
```

$$min\ \sum_g c_1^1 x_1^1 + c_1^{1,2} x_1^{1,2} + \theta \tag{2.22}$$
$$A_1 x_1^1 + A_1^{1,2} x_1^{1,2} = b_1$$
$$\theta \geq \sum_g w_g(\pi_g^l [d_g - B_g^{1,2} x_1^{1,2}])\ \ l = 1,..,\nu\ \ \forall\ g\ \epsilon\ G\ \ (optimality\ cut)$$
$$\pi_g^l (d_g - B_g^{1,2} x_1^{1,2}) \leq 0\ \ l = 1,..,v - \nu\ \ (feasibility\ cut)$$

$$x_1^1 \geq 0$$

$x_1^{1,2} \geq 0$

$X_l = x_1(l) \cup x_1^{1,2}(l)$ $\quad$ (*assign primal optimal value at iteration* l; x_l);

LB $= [c_g^1, c_g^{1,2}]X_l + \theta$ (*compute lower bound*);

l = l + 1;

)

This process is finite because there is a finite number of possible cuts, and we shall find an optimal solution or detected the infeasibility. In fact, in case some second stage problem is infeasible at iteration l for some scenario g, a cut $\pi_g^l(d_g - B_g^{1,2}x_1^{1,2}) \leq 0$, is generated and $x_1^{1,2}(l)$ is cut off since $\pi_g^l(d_g - B_g^{1,2}x_1^{1,2}) > 0$ by construction. If termination has been obtained then the upper bound UB has been reached. Otherwise, if the second stage problems are feasible, an optimality cut is generated and the constraint $\theta \geq \sum_g w_g \pi_g^l(d_g - B_g^{1,2}x_1^{1,2})$ is added to the model. From $\theta \leq \sum_g w_g \pi_g^l(d_g - B_g^{1,2}x_1^{1,2})$, it follows that the pair (X_l, θ) is cut off as possible future solutions. To prove optimality, we note that $[c^1, c^{1,2}]X_l + \sum_g w_g c_g^2 x_g^2(l))$ is always an upper bound on the value of Pc_{r1}. This sort of decomposition is used very often. Some advantages are: the subproblems can be solved independently, even on different computers! At no time is any large l.p. solved, only ones as big as the subproblems plus the master problem, the central controller does not need to get into details on how the proposals are generated. It is enough that it can be for any cost function. If the subproblems have special structure (e.g. the subproblems are transportation problem or min cost flow problems) then those specialized optimization techniques can be used. Practically, the main drawback of this approach is in possible convergence problems. In fact sometimes Benders decomposition suffers of ”stability” problems, especially in the first iterations when a good enough approximation of $Q(x, g)$ is not known or if started at a good guess at the solution. Hence, the trust region concept is borrow from non linear programming and applied in the model at the generic iteration k as adding constraint: $\|x - x^k\| < \Delta_k$, where x^k is supposed a primal admissible solution at iteration k. If Δ_k is taken "large", then the algorithm is the same as Benders decomposition. If Δ_k is taken "small" the solution x^{k+1} at step $k+1$ stays very close to x^k. This is called "regulating" the method [82]. An other way to reduce instability consists in dually penalizing the step that will be taken $min\ \sum_g c_1^1 x_1^1 + c_1^{1,2} x_1^{1,2} + \theta + 1/2\varrho\|x - x^k\|^2$, this is known as regulated decomposition method [80]. If ϱ is taken "large", the the behaviour is like Benders decomposition, if ϱ is taken "small" the solution stay very close

Generally, if the number of scenario is not very large, Benders decomposition produces fast solutions.

A possibility to face huge deterministic equivalent problems can be given by so called stochastic quasi gradient methods [34]. In the simplest case they start from some initial point x_0 and update the current approximation of x step by step. Because $Q(x, g)$ convex function of x with sub-gradient $\partial Q(x, g) = -(B_g^{1,2})^T \pi_g^l$, where π_g^l is the optimal dual solution for the dual problem, such information can be exploited, in order to compute a new current point. In quasi-gradient method, the iteration scheme considers an initial point X_0 to each iteration is updated by making a step $\varphi(l)$ and a direction ϕ taken as $\phi(l) = c_g^{1,2} - (B_g^{1,2})^T \pi_g^l$. According to quasi-gradient method, the new first stage solution X_{l+1} is computed in the following manner $X_{l+1} = \Pi_X(X_l - \phi(l)\varphi(l))$, where the direction $\varphi(l)$is taken opposite to the current estimate ϕ. Π_X is the projection operator on X which transforms an arbitrary $z \in \Re^n$ into the point $\Pi_X(z) \in X$ such that $\|z - \Pi_X(z)\| = min_{x \in X}\|z - x\|$ (2.4). This last formulation is a quadratic programming problem for for which fast efficient solver algorithms exist. Now, by introducing a new dual variable z, we can rewrite the problem

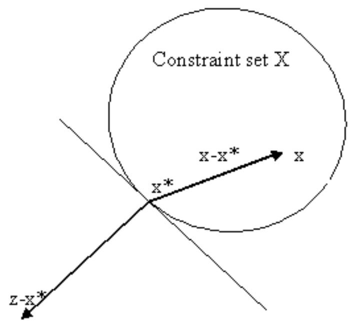

Figure 2.4: Projection on X of an arbitrary point z

Pc_{r1} in the equivalent model:

Model (Pc_{r3})

$$max\ z \tag{2.23}$$

$$z \leq c_1^1 x_1^1 + c_1^{1,2} x_1^{1,2} + \sum_g w_g(\pi_g(d_g - B_g^{1,2} x_1^{1,2})) \tag{2.24}$$

$$A_1 x_1^1 + A_1^{1,2} x_1^{1,2} = d$$

$$B_g^T \pi_g \leq w_g c_g^2,\ \forall\ g\ \in\ G \tag{2.25}$$

$$x_1 \geq 0$$

$$x_1^{1,2} \geq 0$$

Assuming now $(x_1^1(1), x_1^{1,2}(1)), .., (x_1^1(t), x_1^{1,2}(t))$ as the extreme points of Conv(X), potentially each pair $(x_1^1(i), x_1^{1,2}(i)),\ i = 1, .., t$ could be an optimal solution of the Master Problem, so if we rewrite the model as combination of such points, we obtain:

Model (Pc_{r4})

$$max_{\pi} \sum_{j=1,..,t} c_1^1(j)x_1^1(j) + (c_1^{1,2}x_1^{1,2}(j) + \sum_g w_g(\pi_g d_g - B_g^{1,2}x_1^{1,2}(j)) \tag{2.26}$$

$$B_g^T \pi_g \leq w_g c_g^2,\ \forall\ g\ \in\ G \tag{2.27}$$

In fact, starting from the dual problem (D_1Pc_r), (Pc_{r4}) can be obtained by modifying the structure of the problem. Hence, we obtain:

Model (D_1Pc_r)

$$max\ yb_1 + \sum_g \pi_g d_g \tag{2.28}$$

$$A_1^T y \leq c_1$$

$$\sum_g (B_1^{1,2})^T(\pi_g) + A_{1,2}^T y \leq c_{1,2}$$

$$B_g^T \pi_g \leq w_g c_g^2,\ \forall\ g\ \in\ G$$

$$y,\ \pi_g,\ g = 1,2,3\ unconstrained \tag{2.29}$$

hence, if the second line of constraint is relaxed with respect to the linking variable $x_1^{1,2}$ and moved in the objective function, there obtain:

Model (Pc_{r4})

$$min_{x_1^{1,2}} max_{y,\pi_g}\ y(-A_{1,2}x_1^{1,2} + b_1) + \sum_g (\pi_g d_g + (c_{1,2} - \sum_g (B_1^{1,2})^T(\pi_g))x_1^{1,2}) \tag{2.30}$$

$$A_1^T y \leq c_1$$

$$B_g^T \pi_g \leq w_g c_g^2,\ \forall\ g\ \in\ G$$

$$w,\ v_g,\ g = 1,2,3\ unconstrained$$

$$x_1^{1,2} \geq 0$$

and after if it is reintroduced the variable x_1, (Pc_{r4}) becomes:

Model (Pc_{r5})

$$min_{x_1^{1,2},x_1} c_1 x_1 + max_{\pi_g} \sum_g (\pi_g d_g + (c_{1,2} - \sum_g (B_1^{1,2})^T(\pi_g))x_1^{1,2}) \tag{2.31}$$

$$A_1x_1 + A_{1,2}x_1^{1,2} = b_1$$

$$B_g^T\pi_g \leq w_g c_g^2,\ \forall\ g\ \in\ G$$

$$\pi_g,\ g = 1,2,3\ unconstrained$$

$$x_1 \geq 0,\ x_1^{1,2} \geq 0$$

At end, by introducing a new dual variable z, (Pc_{r5}) can be rewritten in the following manner:

Model (Pc_{r6})

$$min_{x_1^{1,2},x_1} c_1x_1 + max_{(\pi_g:\,g\in G),z} \sum_g \pi_g d_g + z \tag{2.32}$$

$$z \leq \sum_g((c_{1,2} - (B_{1,2})^T\pi_g)x_1^{1,2}) \tag{2.33}$$

$$A_1x_1 + A_{1,2}x_1^{1,2} = b_1$$

$$B_g^T\pi_g \leq w_g c_g^2,\ \forall g\ \in G$$

$$z \geq 0$$

Now,by enumerating all the t vertexes of Conv(X), the above problem Pc_{r6} can be rewritten as:

Model (Pc_{r7})

$$min_{j=1,..,t} c_1x_1(j) max_{(\pi_g\ g\in G),z} \sum_g \pi_g d_g + z \tag{2.34}$$

$$z \leq \sum_g((c_{1,2} - B_{1,2}^T\pi_g)x_1^{1,2}(j))$$

$$B_g^T\pi_g \leq w_g c_g^2,\ \forall\ g\ \in\ G$$

$$z \geq 0$$

deleting z in Pc_{r7}, Pc_{r4} is obtained. Because in most cases the above problem is hard to solve directly because it typically involve too many constraints, as bender's decomposition this problem can be face with a row generation (constraints generation) technique called cutting plane method (outer linearization), as same as in bender's algorithm is adopted a relaxation strategy in which only a few set of constraints are explicitly maintained. Let be X_β, a subsets of X, a constraints generation algorithm at each iteration try to enlarge the set X_β by adding new solutions to the inner problem.

Model ($Pc_{r7,\beta}$)

$$\begin{aligned} min_{j\,\in\beta} c_1 x_1(j) max_{(\pi_g,\, g\,\in\, G),z} \sum_g \pi_g d_g + z \\ z \leq \sum_g ((c_{1,2} - B_{1,2}\pi_g) x_1^{1,2}(j)) \\ B_g^T \pi_g \leq w_g c_g^2,\ \forall\ g\ \in\ G \\ z \geq 0 \end{aligned} \tag{2.35}$$

Assuming z(.) the optimal solution, we have $z(Pc_{r7,\beta}) \geq z(Pc_{r7})$. A typical cutting plane algorithm, starting by a initial set X_β search among the first stage primal (inner problem) solution, that are used in order to add new cuts in the master problem, obtaining at every step a better approximation at the model. Following it is given the algorithmic scheme of a typical cutting plane algorithm:

Initialize β;

l = 1;

`/* Solve the master problem */`

while $(z(.) > \varphi(\pi_g^l) - c_1 x_1^1(l))$

(

$$\begin{aligned} ((\pi_g^l : g = 1,2,..), z) = argmax_{(\pi_g : g=1,2,..),z}\ \{\textstyle\sum_g \pi_g d_g + z\ | \\ z \leq \textstyle\sum_g (c_{1,2} - B_{1,2}^T \pi_g) x_1^{1,2}(l),\quad l\ \in\ \beta \\ B_g^T \pi_g \leq w_g c_g^2,\ \forall\ g\ \in\ G\ \} \end{aligned}$$

`/* Solve the inner problem */`

$$(x_1^1(l), x_1^{1,2}(l)) = argmin\{c_1 x_1^1 + \sum_g((c_{1,2} - B_{1,2}\pi_g^l)x_1^{1,2}) \mid$$
$$A_1 x_1 + A_{1,2}x_1^{1,2} = b_1\}$$

`/* Update partial solutions */`

$$\varphi(\pi_g^l) = \{\sum_g \pi_g^l(d_g - B_{1,2}x_1^{1,2}(l)) + c_1 x_1^1(l) + c_{1,2}x_1^{1,2}(l)\}$$
$$X_\beta = X_\beta \bigcup (x_1^{1,2}(l), x^1(l))$$
$$\mathrm{l} = \mathrm{l} + 1;$$

) (end while)

We can observe that at each iteration $(d_g - B_{1,2}x_1^{1,2}(l))$ is a subgradient for $\varphi(.)$ at point $\pi_g^(l)$, it is the only change present in the inner problem, the convex set X does not change, so the solution returned is in one of the extreme points. For each iteration it is also valid $(z \geq \varphi(_(g)^(l)) - c_1 x_1^1(l))$, otherwise we are done. An other general price directive optimization strategy can equivalently state by considering the problem $((Pc_{r4}))$ as a problem depending on the dual variables $\pi_1, .., \pi_g$, with g is the number of scenarios. Hence, we define a function $\rho(\pi_1, .., \pi_g)$ as :

$(Pc_{r8}) = max\{\rho(\pi_1, .., \pi_g) : B_g^T \pi_g \leq w_g c_g^2, \ \forall \ g \ \in \ G\}$

where:

$$\rho(\pi_1, .., \pi_g) = \sum_g \pi_g d_g + min\{c_1 x_1 + (c_{1,2} - \sum_g \pi_g B_{1,2})x_1^{1,2} : A_1 x_1 + A_{1,2}x_1^{1,2} = b_1\}$$

(Pc_{r8}) is know as a lagrangian dual to original scenario problem and $\rho(\pi_1, .., \pi_g)$ is the associated lagrangian subproblem. Because X is close and bounded, as same as in the previous approach, we can rewrite the inner problem $\rho(\pi_1, .., \pi_g)=$ $\sum_g \pi_g d_g + min\{c_1 x_1(j) + (c_{1,2} - \sum_g \pi_g B_{1,2})x_1^{1,2}(j) : j = 1, .., t\}$ that is a piecewise concave function of $(\pi_1, .., \pi_g)$. In this row generation approach we can maintain a set of indexes β, where the constraints $z \leq \sum_g \pi_g d_g + min\{c_1 x_1(j) + (c_{1,2} - \sum_g \pi_g B_{1,2})x_1^{1,2}(j) : j \in \beta\}$ are the tangential supports that actually describe the function $\rho(.)$ and z the value along the $\rho(.)$ axis. Assuming $\bar{w} =$

$(\pi_1(*),..,\pi_g(*))$ a set of admissible dual solutions $\bar{z}$ is an upper bound to dual lagrangian whilst if $\bar{z} = \rho(\bar{w})$ we are done. Otherwise, if at iteration k, $\bar{z} > \rho(\bar{w}) = \sum_g \pi_g(*)d_g + min\{c_1x_1 + (c_{1,2} - \sum_g \pi_g(*)B_{1,2})x_1^{1,2} : A_1x_1 + A_{1,2}x_1^{1,2} = b_1\} = \sum_g \pi_g(*)d_g + min\{c_1x_1(k) + (c_{1,2} - \sum_g \pi_g(*)B_{1,2})x_1^{1,2}(k)$ a new tangential support $z \leq \sum_g \pi_g(*)d_g + min\{c_1x_1(k) + (c_{1,2} - \sum_g \pi_g(*)B_{1,2})x_1^{1,2}(k)$ is generated and included in the outer linearization of $\rho(.)$, the set X_β is enlarged with the couple $x_1(k), x_1^{1,2}(k)$. This process is repeated until termination is obtained. It is possible to demonstrate that to solve $\rho(\pi_1,..,\pi_g)$ by using primal decisional variables of type either integers or real is equivalent [29]. If we suppose that a set of cardinality β of tangential supports have been generated, the following row generation problem:

Model (DDW)

$$max_{z,\pi_{g,\, g\,\in\,G}} \quad z$$

$$z \leq \sum_g \pi_g d_g + min_{j,\, j\,\in\,\beta}\{c_1x_1(j) + (c_{1,2} - \sum_g \pi_g B_{1,2})x_1^{1,2}(j),\ j \in \beta\}$$

$$B_g^T \pi_g \leq w_g c_g^2,\ \forall\, g \in G$$

in the variables $(z, \pi_{g,\, g\,\in\,G})$ can be exploited to generate an other classical scheme called Dantzig-Wolfe, called also column generation scheme. Let X as bounded, in order to obtain the Dantzig-Wolfe scheme, for each row, we introduce a new variable $\theta_j,\ j = 1,..,t,\ t = |\beta|$:

Model (DW)

$$min_{\theta_j, x_g^2} c_1 \sum_{j\,\in\,\beta}(x_1(j)\theta_j) + c_{1,2}\sum_{j\,\in\,\beta}(x_1^{1,2}(j)\theta_j) + \sum_g w_g c_g^2 x_g^2$$

$$B_{1,2}\sum_{j\,\in\,\beta} x_1^{1,2}(j)\theta_j + B_g x_g^2 = d_g,\ \forall\, g \in G$$

$$\sum_{j=1,..,t} \theta_j = 1$$

$$x_g \geq 0,\ \forall\, g \in G$$

where θ is the unitary simplex of proper dimension, with $\Theta = \sum_{j=1,..,t}\theta_j =$

1 is called the convexity constraint. In (DW) given X compact, the optimal solution of $x_1^{1,2}$ and x_1, given a set β of vertices are computed respectively as $\sum_{j\,\in\,\beta} x_1^{1,2}(j))\theta_j$ and $\sum_{j\,\in\,\beta} x_1(j))\theta_j$, otherwise if X is not compact then $x_1^{1,2} = \sum_{j\,\in\,\beta_1} x_1(j))\theta_j + \sum_{j\,\in\,\beta_2} x_1(j))\mu(j) \mid \beta_1 \cup \beta_2 = \beta,\ \mu(j) \geq 0$ are the extreme the rays of X. Assuming the master problem (DW) has been solved using a method that for each scenario yields a dual vector calling $\pi_g^j,\ g\ \in\ G$ corresponding to the constraint: $(B_{1,2} \sum_{j\,\in\,\beta}(x_1^{1,2}(j)\theta_j) + B_g x_g = d_g,\ \forall\ g\ \in\ G$
can be computing for example by using a method that solves simultaneously both the primal and the dual of a linear program, such as the simplex method. Once computed, π_g^j can be used in in a column generation algorithm in order to compute a new primal solution, having a promising reduced cost, by solving the inner problem (pricing problem) and to extend the set β:

$$(\mathrm{x}_1^1(j), x_1^{1,2}(j)) = argmin\{c_1 x_1^1 + \textstyle\sum_g((c_{1,2} - B_{1,2}\pi_g^j)x_1^{1,2}) : A_1 x_1 + A_{1,2}x_1^{1,2} = b_1\}$$

The main steps of a delayed column generation technique can be resume d in the following way:

Procedure Dantzig-Wolfe

Initialize $l = 1, \theta_1 = (1, 0, ..., 0),\ \pi_g^1$;

while (true)

do

$(x_1^1(l), x_1^{1,2}(l)) = argmin\{c_1 x_1^1 + \sum_g((c_{1,2} - B_{1,2}\pi_g^l x_1^{1,2})|\ A_1 x_1 + A_{1,2}x_1^{1,2} = b_1\}$

if (θ_j can be added to DW)

{

add the new variable θ_j to DW with:

coefficient cost $(c_1 x_1(l) + c_{1,2}x_1^{1,2}(l))$;

coefficient column $([B_{1,2}x_1^{1,2}(l)\ 1])$;

}

solve the master problem DW and return $\pi_g^l,\ g\ \in\ G$ and $\theta_j,\ j=1,..,l$;

$l = l + 1$;

if (termination conditions are verified) break;

end (while)

Chapter 3

Bundle methods

The dual structure of the scenario models we deal with, using lagrangean relaxation, are particular concave functions, called polyhedral functions, that can be expressed as: $\varphi(y) = min_{i=1,..,l}\{b_i + yg_i\}$, where g_i represents the gradient of one of the defining hyperplanes in a point $y \in Y$, where Y is a convex subset of $\Re^n$. The goal is to find the maximum value of the function $\varphi(y)$, in general difficult to solve, because $\varphi(.)$ is unknown. The useful information that can be retrieved from $\varphi(.)$ by giving a vector $\bar{y}$ as input parameter, can be the value of $\varphi(\bar{y})$ and a sub-gradient $g = g(\bar{y})$ of $\varphi(.)$ in $\bar{y}$, that satisfies the following inequality:

$$\varphi(\bar{y}) \leq \varphi(y) + g(y - \bar{y}),\ \forall\ \bar{y}\ \in\ R^n$$

Assuming $\varphi(.)$ unconstrained, an other way to represent these kind of problems, can be done by introducing the definition of epigraph of $\varphi(.)$, $Epi(\varphi(.))$, as:

$$\text{Epi}(\varphi()) = \{(y, v) \in R^{n+1} \mid v \leq \varphi(y)\}$$

3.1 The dual viewpoint

Geometrically, a sub-gradient is the gradient of a supporting hyperplane of $Epi(\varphi())$. The maximization problem we have to solve is:
Model (D)

$$\max_v\{v \ : \ v \ \leq \ min_{i=1,..,l}\{b_i + yg_i\}\}$$

As example, in figure 3.1, we report a set of supporting hyperplanes for problem D. An upper approximation of (D) can be done by considering a set $\beta = (1, .., q)$ of points y_i such that:
Model (D_β)

$$\max_v\{v \ : \ v \ \leq \ (\varphi_\beta(y) =) min_{i \in \beta}\{\varphi(y_i) +_{y_i} (y - y_i)\}$$

We cannot guarantee the sub-gradient is an ascent direction, however it can be used to implement algorithms (see figure 3.2) which maximize such polyhedral functions exploiting, at each iteration, the information regarding both $\varphi(\bar{y})$ and $g(\bar{y})$ in order to compute a tentative ascent direction $d = y - \bar{y}$. The displacement with respect to the current point $\bar{y}$ and a new current point $y = \bar{y} + d$, adding a new index on set β. Hence, D_β can be rewritten in function of the displacement $d = y - \bar{y}$ with respect to the current point $\bar{y}$. As a conseguence $\bar{y}$, the new displacement d can be computed as:
Model $(D_{\beta,\bar{y}})$

$$\max_{v,d}\{v \ : \ v \ \leq \ (\varphi_\beta(d) =) min_{i \in \beta}\{b_i + g_{y_i}(d)\}$$

Bundle methods have been used frequently to solve non-smooth optimization problems. In these methods sub-gradient directions coming from past iterations, are accumulated in a bundle and a trial direction is obtained by performing quadratic programming based on the information contained in the bundle. The

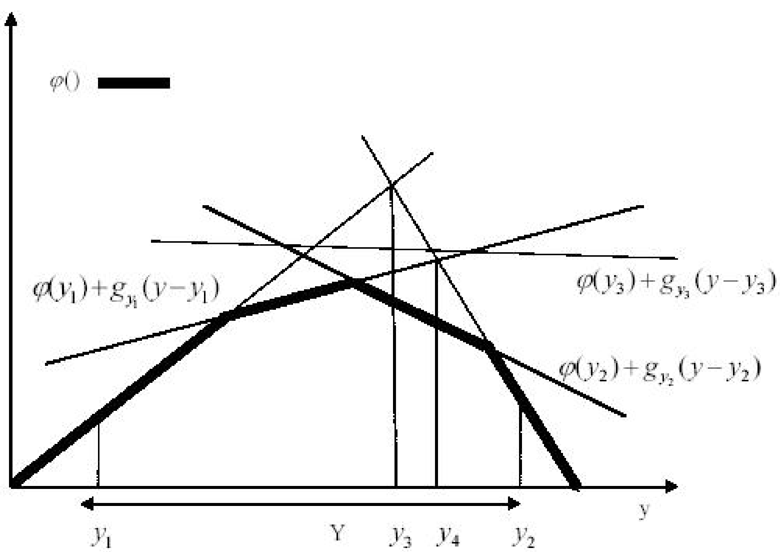

Figure 3.1: A set of supporting hyperplanes generated using a cutting plane method

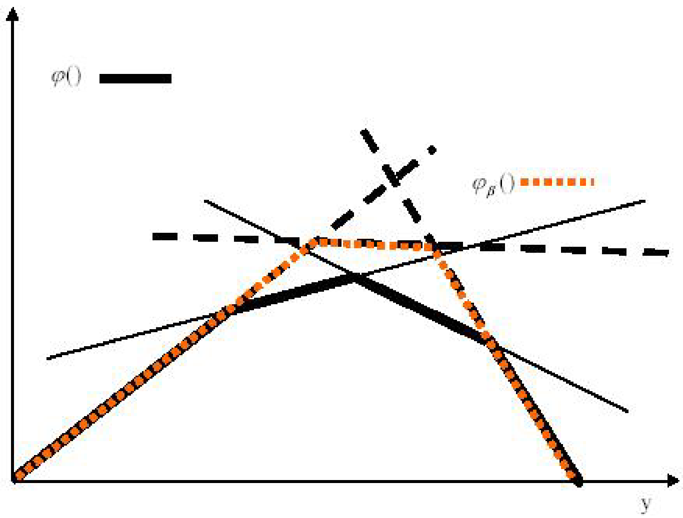

Figure 3.2: $\varphi_\beta(.)$ approximates $\varphi(.)$

algorithms based on bundle methods employ a concept called $\epsilon-subdifferential$, defined as:

$$\delta_\epsilon\varphi(y) = \{g \in R^n : \varphi(\bar{y}) \leq \varphi(y) + g(y - \bar{y}) + \epsilon, \ \forall \ \bar{y} \ \in \ R^n\}$$

Elements in $\delta_\epsilon\varphi(y)$ are called $\epsilon - subgradients$. In figure 3.4, we show two $\epsilon - subgradients$ at point $\bar{y}$. Correspondingly, the $\epsilon - directional$ derivative along the direction d at y is defined as:

$$\varphi_\epsilon^{'}(y, d) = sup_{t>0}[\varphi(y + td) - \varphi(y) - \epsilon]/t$$

Where:

$$\varphi_\epsilon^{'}(y, d) = inf_{\delta_\epsilon\varphi(y)} g^{'} d$$

For a convex function $\varphi_0(y, d)$ $\forall$ y and $\forall$ direction exists.

If a direction d can be found such that $\varphi_\epsilon^{'}(y, d) > 0$, then the dual cost can be increased by at least ϵ. Therefore, it is desidered to select a search direction d^* such that the directional derivative is maximized, i.e.:

$$\begin{aligned} \mathrm{d}^* &= argmax_{||d||=1}\{\varphi_\epsilon^{'}(y, d)\} \\ &= \mathrm{argmax}_{||d||=1}\{\varphi_\epsilon^{'}(y, d)\} \\ &= \mathrm{argmin}_{g\in\delta_\epsilon\varphi(y)} max_{||d||=1} g'd \\ &= \mathrm{argmin}_{g\in\delta_\epsilon\varphi(y)} || \ g \ || \end{aligned}$$

Generally, since the $\epsilon - subdifferential$ is very difficult to obtain, the idea of bundle methods is to accumulate sub-gradients of the past iterates in a bundle $\beta = \{g_1, .., g_b\}$ and to approximate $\delta_\epsilon\varphi(y)$ by the convex hull of the bundle elements:

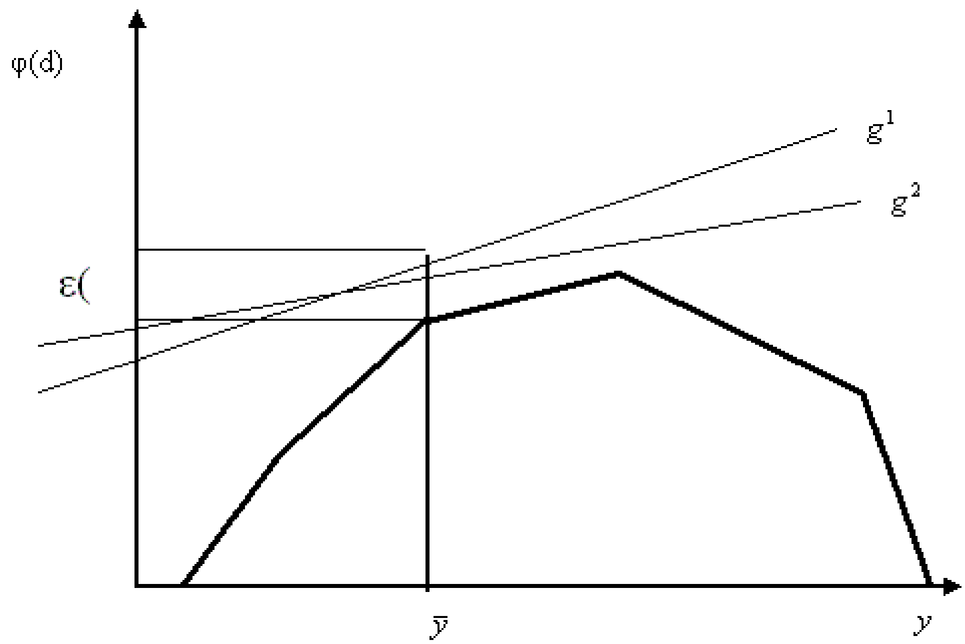

Figure 3.3: two ϵ-subgradients

$$P_b = \{g \mid g = \sum_{i=1}^{b} \vartheta_i g_i, g_i \in, \vartheta_i \geq 0, \sum_{i=1}^{b} \vartheta_i = 1, \sum_{i=1}^{b} \alpha_i \vartheta_i \leq \epsilon\}$$

where α_i, the linearization error for element **i**, is:

$$\alpha_i = \{\varphi(y_i) + g_i(y - y_i) - \varphi(y)\}$$

the $\epsilon - subgradient$ enjoys the so-called "information transport property":

$$\text{g} \in \delta_\epsilon \varphi(\bar{y}) \Rightarrow g \in \delta_\nu \varphi(\bar{y}) \quad \forall \nu \geq \alpha = \varphi(\bar{y}) + g(y - \bar{y}) - \varphi(y) + \epsilon$$

That is, a vector g belonging to the $\epsilon - subdifferential$ of $\varphi(.)$ in some point $\bar{y}$ also belongs to the $\alpha - subdifferential$ of $\varphi(.)$ in any other point y. Hence, the first order information about $\varphi()$ in a point $\bar{y}$, can be translated into an information about a completely unrelated point y, at the cost of a known linearization error α. Historically, the first algorithm presented as bundle method was based on the following (QP) problem:

Model (D_β^ϵ)

$$\min_\theta\{\| \sum_{j\in\beta} g_j \theta_j \|: \sum_{j\in\beta} g_j \alpha_j \leq \epsilon, \sum_{j\in\beta} \vartheta_j = 1\}$$

where, α_j is the "cost", of proper dimension, associated to each θ_j. Such approach depends on critical parameter ϵ, that must be dynamically update in order to achieve good practical performance. In order to avoid working with ϵ parameter, it has been proposed a new approach working with a fixed parameter t:

Model$(D_{\beta t})$

$$\min_\theta\{1/2t \| \sum_{j\in\beta} g_j \theta_j \|^2 + \sum_{j\in\beta} g_j \alpha_j : \sum_{j\in\beta} \vartheta_j = 1\}$$

that can be considered as the lagrangean relaxation of (D_β^ϵ) different from $(D_{\beta t})$. It can be viewed as the relaxation of (D_β^ϵ), where a lagrangian multiplier (2/t) is used to dualize the constraint $\sum_{j\in\beta} g_j \alpha_j \leq \epsilon$, having the following problem:

$$\min_\theta\{1/2t \parallel \sum_{(j\in\beta)} g_j\theta_j \parallel^2 + \sum_{(j\in\beta)} g_j\alpha_j : \sum_{j\in\beta}\vartheta_j = 1\} - \epsilon t/2$$

that differs from $(D_{\beta t})$ for the parameter t in the objective function. The Karush-Khun-Tucker conditions for the two problems are essentially the same, i.e. return the same optimal primal solution θ and the respective (v, d) are scaled of a factor of t. This problem can be viewed as minimizing the distance between

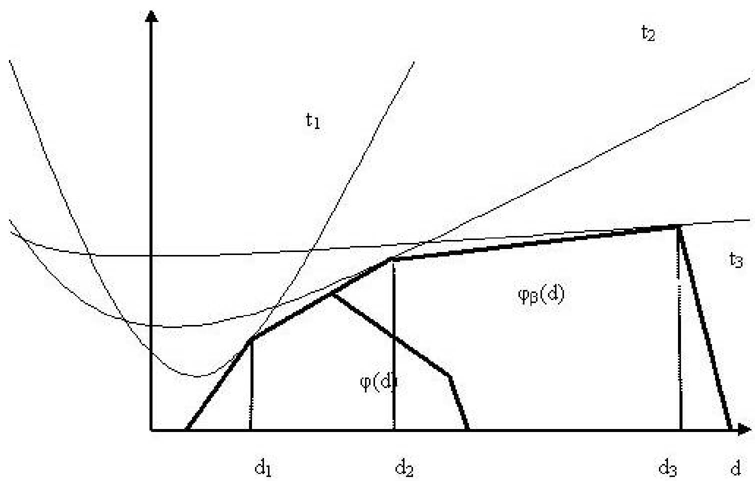

Figure 3.4: Effects of the stabilizing term: for $t_1 < t_2 < t_3$ and given a set β, d_i, i=1,..,3 are the optimal solutions for φ_β

the functions φ_β and $1/2t \parallel t \parallel^2$, that is the point of contact among the two functions, as reported in figure 3.4. The choice of the parameter t (t-strategy), is a critical part of all bundle algorithms. We can prove that the quadratic dual of $(D_{\beta t})$:

$(\Pi_{\beta t})$

$$\max_{d,v}\{v - 1/(2t) \parallel d \parallel^2 : v \le dg_j + \alpha_j, j \in \beta\}$$

is equivalent to $(D_{\beta t})$ in the sense that, being θ the optimal solution of $(D_{\beta t})$. Proof: $\Pi_{\beta t}$ is equivalent to the problem: $(\Pi_{\beta t_1})$

$$\min_{d,v}\{1/(2t)d^T d - v : v - d^T g_j \le \alpha_j, j \in \beta\}$$

the corresponding lagrangian relaxation is given by:

$$\text{L(d,v,}\theta) = 1/(2t)d^T d - v + \textstyle\sum_{j\in\beta}\theta_j[v - d^T g_j - \alpha_j]$$

for the Karush-Khun-Tucker condition, we have the following relations:

$$\nabla_d L(d, v, \theta) = 0 \quad \Rightarrow \quad (1/t)d - \textstyle\sum_{j\in\beta}\theta_j g_j = 0$$

$\nabla_d L(d, v, \theta) = 0 \quad \Rightarrow \quad \sum_{j\in\beta}\theta_j - 1 = 0$ where, from the first one we have $d = t\sum_{j\in\beta}\theta_j g_j$ and from the second $\sum_{j\in\beta}\theta_j = 1$.

Hence, the dual lagrangian problem can be formulated as:

Model $(DL_{d,v,\theta})$

$$max_{d,v,\theta} L(d, , v, \theta) \tag{3.1}$$

$$d = t\sum_{j\in\beta}\theta_j g_j \tag{3.2}$$

$$\sum_{j\in\beta}\theta_j = 1 \tag{3.3}$$

$$\theta \ge 0 \tag{3.4}$$

By making some simple algebra in $DL_{d,v,\theta}$ it follows that:

$$L(d, , v, \theta) = -1/(2t)d^T d + 1/(t)d^T d - v + \sum_{j\in\beta}\theta_j[v - d^T g_j\alpha_j] = \tag{3.5}$$

$$= -1/(2t)d^T d + d[(1/t)d - \sum_{j\in\beta}\theta_j g_j] + v[\sum_{j\in\beta}\theta_j - 1] + \sum_{j\in\beta}\theta_j\alpha_j = \tag{3.6}$$

$$= -1/(2t)t^2\|\sum_{j\in\beta}g_j\theta_j\|_2^2 - \sum_{j\in\beta}\alpha_j\theta_j \tag{3.7}$$

From the second and third eqaulity by using Karush-Khun-Tucker conditions and by substituting d with $t\sum_{j\in\beta}g_j\theta_j$, the problem $DL_{d,v,\theta}$ can be reformulated as:

$$max_\theta - (1/2)t\|\sum_{j\in\beta}g_j\theta_j\|_2^2 - \sum_{j\in\beta}\alpha_j\theta_j \tag{3.8}$$

$$\sum_{j \in \beta} \theta_j = 1 \tag{3.9}$$

$$\theta \geq 0 \tag{3.10}$$

that it is equivalent, except for the sign, to the primal quadratic problem:

$$max_\theta (1/2)t \| \sum_{j \in \beta} g_j \theta_j \|_2^2 + \sum_{j \in \beta} \alpha_j \theta_j \tag{3.11}$$

$$\sum_{j \in \beta} \theta_j = 1 \tag{3.12}$$

$$\theta \geq 0 \tag{3.13}$$

The optimal solution d and v of $(\Pi_{\beta t})$ can be computed as:

$d = tz$ where z=$\sum_{j \in \beta} g_j \theta_j$

v=$t \|z\|^2 + \sigma$ where

$\sigma = \sum_{j \in \beta} \alpha_j \theta_j$

the new tentative point y is found along the direction z (convex combination of the currently available sub-gradient) , using t as predefined step-size. In this approach, called "trust region approach", t is increased or decreased depending on the outcome of the previous step: increasing t can be preferred to move to y whenever $\varphi_\beta(.)$ is a good approximation of φ also outside the current trust region. In case of null step, decreasing t can be preferred respect to inserting the new g, if g is not believed to return new useful information about φ in a neighbourhood of $\bar{y}$ (i.e. due to a large α). As a conseguence, cutting plane and conjugate sub-gradient algorithms represent the limit cases respectively for $t \to 0$ and $t \to \infty$ using $(\Pi_{\beta t})$. If we formulate:

Model $(\zeta_{\beta t})$

$$max_d \{\varphi_\beta(d) - 1/(2t) \parallel d \parallel^2\} \tag{3.14}$$

(3.15)

as:

Model ($\zeta_{\beta t}$)

$$max_{d,v}\ v - 1/(2t) \parallel d \parallel^2 \tag{3.16}$$

$$v \le dg_i + \alpha_i,\ j\ \in\ \beta$$

we can prove that:

$$max_{d,v}\ v - t/2 \parallel \sum_{j\ \in\ \beta} g_j\theta_j \parallel^2 \tag{3.17}$$

$$t \parallel \sum_{j\ \in\ \beta} g_j\theta_j \parallel g_k + \alpha_k - v \ge 0,\ \forall\ k\ \in\ \beta$$

Proof: At first the two problems differ in the space of the admissible solutions. The second problem is a max problem in θ but the objective function depends on a combination of the g_j. The first problem is a max problem in the vector d in the wider region of the possible solutions. A possible solution of the first problem can be written as $d^* = d^{(1)*} + d^{(2)*}$ where $d^{(1*)}\ \in\ Conv\{g_i :\ i\ \in\ \beta\}$ and $d^{(2)*}$ is the ortogonal component of d^* ortogonal to $d^{(1)*}$. Hence, the optimal value is the sum of vectors both belonging to the admissible region of the dual lagrangian problem or ortogonal to it (the admissible region). Now we show that the component ortogonal to the admissible region of the dual lagrangian problem is null in the optimum of the problem of the first problem. Hence the two problems coincide. It decomposes a generic admissible vector in the ortogonal components $d = d^{(1)} + d^{(2)}$ with $d^{(1)} \perp d^{(2)}$ in a such a way that:

$$d^{(1)} = \sum_{j \in \beta} g_j\theta_j \tag{3.18}$$

$$d^{(2)T} g_j = 0,\ \forall\ j\ \in\ \beta$$

hence, $d^{(1)}$ belongs to the admissible region of the dual lagrangian problem and $d^{(2)}$ is ortogonal to it. Substituting the first problem becomes:

$$max_{d,v}\ v - 1/(2t) \parallel d^{(1)} + d^{(2)} \parallel^2 \tag{3.19}$$

$$v \le (d^{(1)} + d^{(2)})g_i + \alpha_i, \; j \; \in \; \beta$$

hence, the constraint can be expressed as:

$$v \le d^{(1)}g_i + \alpha_i, \; j \; \in \; \beta \tag{3.20}$$

If a feasible solution exists, considering the objective function:

$$\max_{d,v} \; v - 1/(2t) \parallel d^{(1)} \parallel + \parallel d^{(2)} \parallel$$

the maximum value is obtained for $\parallel d^{(2)} \parallel = 0$.

The dual lagrangian problem has been obtained except for a factor t that can be introduced by considering $d = d^{(1)} = t \sum_{j \in \beta} g_j \theta_j$. Hence, the bundle algorithm can be viewed as a stabilized version of the cutting plane algorithm, in other word we can obtain a new point $\bar{y} + d^*$ far from y, also if y is close to the solution. The stabilizing term introduces a measure of the distance from the proximal point $\bar{y}$, discouraging directions leading too away. A scheme of a bundle algorithm can be resumed as:

Procedure bundle

let $t^* > 0$ and $\epsilon > 0$ be fixed, initialize $\bar{y}$, t and β

do

solve problems $(D_{\beta t})$ and $(\Pi_{\beta t})$ and $(\sigma, \|z\|)$;

find a direction d, move along d, generating some new g_i and a trial point y ;

if (a large enough improvement has been obtained)

then $\bar{y} = y$;   make a serious step

update β, (possibly) including some new g_i, deleting old ones and/or adding convex combination of the present ones (typically z); $(\beta - strategy)$

update t, depending on the previous iterations; $(t - strategy)$

while $(\sigma + t^{*}\|z\|^2 > \epsilon\ )$;

For any θ the vector $z = \sum_j g_j \theta j$ belongs to the $\sigma - subdifferential$ of the current point $\bar{y}$, and the following relation holds $\|z\| = 0 \Rightarrow \bar{y}$ is $\sigma - optimal$. This can be used in order to obtain the following "standard" stopping condition $\|z\|^2 \leq \varepsilon_Q$ and $\sigma \leq \varepsilon_L$ where ε_Q and ε_L are two fixed tollerance parameters. In fact, the stopping criteria should be $\varphi(y*) - \varphi(\bar{y}) \leq \varepsilon$, but it is not implementable and if y* was known, it could be substituted by $\sigma + (y* - \bar{y})z \leq \varepsilon$, because $z \in \delta_\alpha \varphi(\bar{(y)})$, in other words $\varphi y* \leq \varphi(\bar{y}) + (\bar{y} - y*) + \sigma$. An even stronger condition could be $\sigma + t * (\bar{y}, z)\|z\|^2 \leq \varepsilon$ that can be turned in the "standard" stopping criteria by letting $\varepsilon_L = \varepsilon/2$ and choosing at each step some $\varepsilon_Q \leq \varepsilon_L / t * (\bar{y}, z)$, if the function $t * (\bar{y}, z) = |y * - y| \setminus |z|$ is known. Hence, $t * (\bar{y}, z)$ can substitute by a fixed parameter $t*$, that represents un upper approximation of the function $t * (.)$.

3.2 The primal viewpoint

Bundle algorithms can be applied to all problems having a set of constraints simple to deal with and a set of complicating constraints, such as in scenario analysis where the complicating constraints are represented by linking constraints that make the problem more difficult to solve. Generally, if we regard a primal viewing to such problems, they have the following primal structure:

Model (H)

$$min\ \{cx \mid x \in X,\ Ax = b\} \tag{3.21}$$

where X is a set closed and bounded, $Ax = b$ are the complicant constraints. X could be a structured polyhedron, a convex set, a discrete set or a combination of them, so it is possible substituting X with $\bar{X} = Conv(x_1, .., x_p)$, where each x_i is an extreme point in X. The problem (H) is then equivalent to: Model (P)

$$min\{cx \mid x \in \bar{X},\ Ax = b\} \tag{3.22}$$

$$\tag{3.23}$$

at the place of (H).

Hence, if we apply the Dantzig-Wolfe structured problem at P :

Model (M)

$$\begin{aligned} min\ c(\sum_{j=1,..,p} x_j\theta_j) \\ \sum_{j=1,..,p} x_j\theta_j = b \\ \sum_{j=1,..,p} \theta_j = 1 \end{aligned} \tag{3.24}$$

where $\Theta = \{\sum_{j=1,..,p} \theta_j = 1\}$ is the unitary simplex and each $x \in X$ is described as a convex combination of the vertices x_j using multipliers θ. The fundamental strategy of the (DW) algorithm is to solve (M) by generating subset β of the vertices, and solve the master problem:

(MDW)

$$\begin{aligned} min\ c(\sum_{j\in\beta} x_j\theta_j) \\ \sum_{j\in\beta} x_j\theta_j = b \\ \sum_{jin\beta} \theta_j = 1 \end{aligned} \tag{3.25}$$

Which is equal to:

(MDW)

$$\begin{aligned} min\ cx \\ x \in |X_\beta| \\ Ax = b \end{aligned} \tag{3.26}$$

the linear dual problem of (MDW) is:

Model (DDW)

$$max\ yb + v \tag{3.27}$$

$$v \leq (c - yA)x_i, \ \ x_i \ \in \ X_\beta$$

$$Ax = b$$

and its optimal solution (v,y) is used in the DW approach for the so-called "mechanized pricing", for obtaining N optimal solution x=x(y) of the pricing problem:

Model (P_{c-yA})

$$yb + min\ (c - yA)x \tag{3.28}$$

$$x \in X$$

that gives a new tentative optimal vertex $x_i (= x(y_i)) \in |X|$ which updates the inner approximation X_β. If the corresponding constraint $v \leq (c - yA)x_i, \ \ i \ \in \ \beta$, is satisfied, then we have an optimal dual solution for the starting problem, otherwise a new column $[Ax_i(y)\ 1]^T$ is added to Dantzig-Wolfemaster model and the process is repeated. As example, we report in figure 3.5 the sequence of x_i obtained at each iteration of the Dantzig-Wolfe algorithm.

Now, rewriting the problem in the equivalent lagrangian dual form we have:

$$max_y\ cx_i + (b - Ax_i)(y) \tag{3.29}$$

For any dual vector y, $\varphi(.)$ can be written as $\varphi(y) = cx(y) + yg(y)$ and the linearizzation error α_j of $\varphi(.)$, with respect to any point $\bar{y}$, is:

$$\alpha_j = \varphi(y_j) - (\bar{y} - y_j)g - \varphi(\bar{y}) = (c - \bar{y}A)(x_j - x(\bar{y})) = \tag{3.30}$$

$$= (c - \bar{y}A)x_j - \varphi(\bar{y}) + \bar{y}b = cx_j + \bar{y}g_j - \varphi(\bar{y}) \tag{3.31}$$

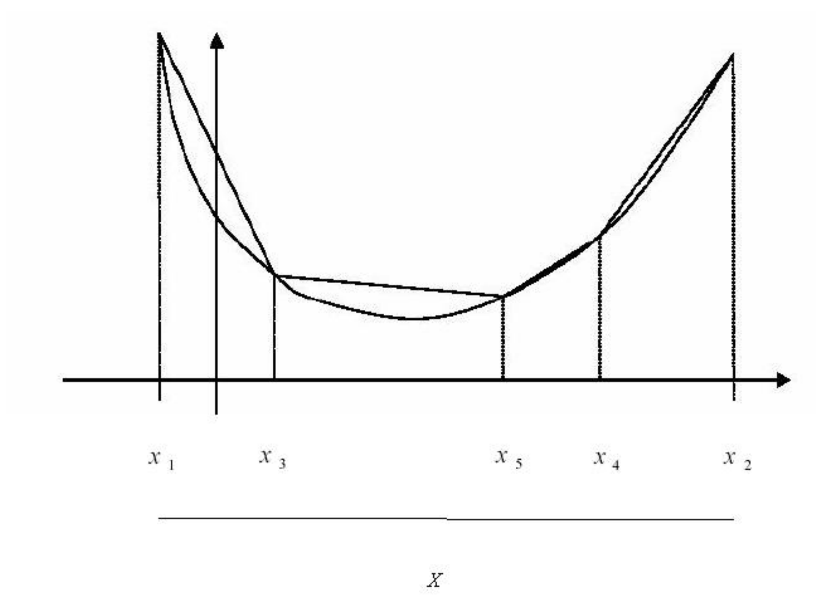

Figure 3.5: Sequence of the optimal solutions $x_i(y)$ obtained in Dantzig-Wolfe algorithm

As a conseguence the problem (L_β) can be expressed as:

Model (L_β)

$$max_{y,v} v : v \leq (c - yA)x_j - \varphi(\bar{y}) + \bar{y}b\} + \varphi(\bar{y})$$

the above relations on the α_i, the definition $y = \bar{y} + d$ and some simple algebra yields:

Model (D_β)

$$max_{y,v}\{v : v \leq dg_j + \alpha_j,\ j \in \beta\} + \varphi(\bar{y})$$

we can show that (L_β), with respect to the current $\bar{y}$ point, is equivalent to the following model:

Model (M_β)

$$min_\theta\{\alpha\theta : \sum_{j\in\beta} g_j\theta_j = 0, \sum_{j\,\in\,\beta} \theta_j = 1\}$$

(M_β) has a non empty feasible region, any feasible solution θ returns an upper bound $\sigma = \alpha\theta + \varphi(\bar{y})$ to the optimal value $\varphi(y*)$.

Moreover, by setting $\psi(\rho) = min_x(c - \rho)x : x \in X = z((Pc - \rho))$ we have $\varphi(\rho) = \psi(yA) + yb$ and it is possible to proof the relation between the ε-optimality and the ε-subgradient for $\psi(.)$. There are a set of important theorems involving the quality of the primal solution x with the subgradient information.

Theorem: let $\psi(\gamma) = \{(c - \gamma)x : x \in X\}$, x is ε-optimal for problem P_{c-yA} $[\Leftrightarrow]$ -x is a ε-subgradient for $\psi(.)$ in γ.

proof: $[\Rightarrow]$ By ε-optimality of x, $\psi(\gamma) + \varepsilon \geq (c - \gamma)x$, clearly for any γ', x is in general suboptimal i.e. $(c - \gamma') \geq \psi(\gamma')$, hence:

$\psi(\gamma') \underset{\Leftarrow}{\leq} (c - \gamma')x + \gamma x - \gamma x = (c - \gamma)x + (\gamma - \gamma')x \leq \psi(\gamma) + (\gamma' - \gamma)(-x) + \varepsilon$

Conversely, assuming $\forall\gamma'\psi(\gamma') \leq \psi(\gamma) + (\gamma' - \gamma)(-x) + \varepsilon$ holds, but $\psi(\gamma) < (c - \gamma)x - \varepsilon)$ (x is not ε-optimal); then by taking $\gamma' = c$ in the previous relation, we get:

$0 = \psi(c) \leq \psi(\gamma) + (c-\gamma)(-x) + \varepsilon < [(c-\gamma)x - \varepsilon] - (c-\gamma) + \varepsilon = 0.$

The linearizzation error $\alpha_i = \varphi(y_i) + g_i(\bar{y} - y_i) - \varphi(\bar{y})$ of the i-th subgradient $g_i = b - Ax_i$ with respect to the current point $\bar{y}$

$\alpha_i = \varphi(y_i) + g_i(\bar{y} - y_i) - \varphi(\bar{y}) =$

$= [\psi(y_i A) + y_i b] + (\bar{y} - y_i)(b - Ax_i) - [\psi(y_i A) + \bar{y}b] =$

$= \psi(y_i A) + (-x_i)(\bar{y}A - y_i A) - \psi(\bar{y}A) = \varphi(\gamma_i) + (\bar{\gamma} - \gamma_i)(-x_i) - \psi(\bar{\gamma})$

(where $\bar{\gamma} = \bar{y}A$ and $\gamma_i = y_i A$), i.e. it is as well the linearizzation error of $-x_i$ (as a sub-gradient of $\psi(.) \in \gamma_i$) with respect to $\bar{\gamma}$: in other words, x_i is an $\alpha_i - optimal$ solution for the pricing subproblem $(P_{c-\bar{\gamma}})$ corresponding to the current point $\bar{y}$. This immediatly implies that for any $\theta \in \Theta, \bar{x} = \sum_{j \in \beta} x_j \theta_j$ is a σ-subgradient of $\psi(.)$ in $\bar{\gamma}$, that is σ-optimal for $(P_{c-\bar{\gamma}})$. Hence, it is possible to measure the "quality" of the current point $\bar{y}$.

Lemma: let $x*$ be the optimal solution for the Dantzig-Wolfe problem and $z = b - A$, $\bar{x} = \sum_{(j \in \beta)} g_i \theta_i$ (for any $\theta \in \Theta$); hence, $cx* \geq c\bar{x} + \bar{y}z - \sigma$.

Proof: σ-optimality of $\bar{x}$ implies $(c - \bar{y}A)x * + \sigma \geq (c - \bar{y}A)\bar{x}$, that is $cx* \geq c\bar{x} + \bar{y}(Ax * - A\bar{x}) - \sigma$, but $x*$ is feasible, in other word $Ax* = b$, this complete the proof.

Lemma: for any $\theta \in \Theta$, $c\bar{x} - \psi(\bar{y}) \leq \sigma - \bar{y}z$.

Proof: By σ-optimality of $\bar{x}$, $\varphi\bar{y} = \psi(\bar{y}A) + \bar{y}b \geq (c - \bar{y}A)\bar{x} - \sigma + \bar{y}b$.

However, the above theorems are not exploited in order to use them as stopping criteria, such rules are a generalizetion of Dantzig-Wolfe stopping conditions, since they hold when the constraint $Ax = b$ are slightly violated $\|b - Ax\| \leq \varepsilon_Q$ and $\bar{x}$ is almost optimal for the current pricing problem. Hence ε_Q is a measure of the maximum mass balance constraints violation accepted for the termination rules. A primal interpretation of $(\delta_{\beta t})$ can be obtained from:

(M_β^ε)

$$min_\theta \ \alpha\theta \tag{3.32}$$

$$|G_\beta\theta|^2 \leq \varepsilon, \ \theta \in \Theta \tag{3.33}$$

Where G_β is the matrix having $\{g_i : \ i \in \beta\}$ as columns, where the mass balance constraints are relaxed as $||b - Ax||^2 \leq \varepsilon$, where M_β is a langrangian relaxation of $\delta_{\beta t}$. In a practical approach using (M_β^ε) instead of (M_β) is profitable because it

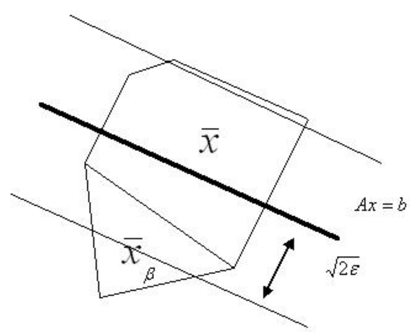

Figure 3.6: (M_β) can be empty while (M_β^ε) it is not

extends the region $\{x| \ Ax = b\}$ to a cylinder of ray $2\varepsilon^{1/2}$ around it, as reported in figure 3.6. It is always possible to find a large enough ε which make (M_β^ε) feasible, in order to mitigate the instability problems due to Dantzig Wolfe algorithm, while if M_β is used the algorithm can return unbounded as result until X_β is enough large.

Proposition: φ_β is bounded if and only if exists $\theta \in \Theta$ such that $G_\beta\theta = 0$

Proof: $G_\beta\theta = 0$ implies that (M_β) has a feasible solution, therefore its dual (L_β)

(equivalent to the maximization of φ_β) cannot be unbounded; in other words if it exists a vector $0 \in \delta_\varepsilon$ then φ_β is bounded. Proposition: if the stopping condition $\sigma + t * |z|^2 \leq \varepsilon$ holds, and $|G\Theta *| = 0$ for any optimal solution of (Δ_{t*}), then $\varphi(y*) - \varphi(\bar{y}) = z(\delta_{t*}) \leq \varepsilon$.

Proof: $|G\Theta *| = 0$ implies $v* = \varphi(\bar{v} + d*) = z(\Pi_{t*}) = z(\Delta_{t*}) = \alpha\theta*$ with $\theta*$ is optimal to (Δ_t) for any $t \geq t*$ and $z(\Delta_t) > z(\Delta_{t*})$ $(z(\Delta_t) = min_\theta\{t|G\theta|^2 + \alpha\theta\}$ is a non decreasing function). It is also true that $z(\Delta_t \leq t0 + \alpha\theta*) = z\Delta_{t*}$, since $\theta*$ is a feasible solution for solution (Δ_t) the model is bounded, so that $\varphi(\bar{y}) + \varphi_\beta(d*) = \varphi(\bar{y} + d*)$ is at the same time an upper bound and a lower bound on $\varphi(y*)$, hence $y* = \bar{y} + d*$ is an optimal solution for $max\{\varphi(y) : y \in Y\}$. Since $\sigma + t * |z|^2 \geq z(\Delta_{t*})$, then $\varphi(y*) - \varphi(\bar{y}) = \varphi_\beta(d) = z(\Delta_{t*}) \leq \sigma + t * |z|^2 \leq \varepsilon$.

Chapter 4

First application: an application of scenario analysis to water system management

4.1 Introduction

Water Resources dynamic management (WR) problems with a multi-period feature are associated with mathematical optimization models that handle thousands of constraints and variables depending on the level of detail required to reach a significant representation of the system ([53], [100]). One optimization approach is to model the WR problem as a dynamic multi-period network flow problem, where all data are fixed and no level of uncertainty is considered ([83],[55]). Efficient optimization algorithms have been used to solve this kind of problem [84]. But, WR problems are typically characterized by a level of uncertainty regarding, among other things, the value of hydrological exogenous inflows and demand patterns. Assigning inaccurate values to them could well invalidate the results of the study. Consequently, deterministic models are inadequate for the representation of these problems where the most crucial parameters are either unknown or are based on an uncertain future.Many dynamic planning and management problems are typically characterized by a level of uncertainty regarding the value of data input such as supply and demand patterns [44]. The traditional stochastic approach gives a probabilistic description of the unknown parameters on the basis of historical data. This is a very efficient approach when a substantial statistical base is available and reliable probabilistic laws can adequately describe parameters, uncertainty and their possible outcome ([52],[54],[81]). It is well known that stochastic optimization approaches cannot be used when there is insufficient statistical information on data estimation to support the model, when probabilistic rules are not available, and/or when it is necessary to take into account information not derived from historical data. In these cases, the scenario analysis technique could be an alternative approach ([30], [77]). The aim of this paper is to apply the scenario analysis framework approach to WR problems and investigate its effectiveness with respect to traditional approaches. A WR model is usually defined in a dynamic planning horizon in which management decisions have to be made sequentially or globally decided as a decision strategy referred to a predefined scenario, where a scenario represents a possible realization of some sets of uncertain data in the time horizon examined. One common approach is to carry out a set of experiments on a number of generated series (parallel scenarios) followed by a simulation-testing phase of each scenario in order to validate the solutions under investigation. All the solutions (each one is a sequence of decisions) are completely independent one from the other because they are obtained from scenarios analyzed separately. As a consequence, the decisions adopted are closely related to the scenario selected at the end of the simulation and the study must start all over again if a different scenario comes true. To overcome the above difficulties, in this paper we analyzed the scenario approach for WR offering some general rules for organizing a predefined set of scenarios into the scenario tree and for identifying a complete set of decision variables relative to all the scenarios under investigation. The scenario tree is obtained by aggregate common portions of scenarios; the aggregation condition guarantees that the solution (that is, the decisions) in any given period is independent of the information not yet available. Scenario analysis approach for WR was proposed in [31] and in Wam-Me EU project [85], and tested on some real physical systems.

4.2 Water Resource Dynamic Model

A successfully applied approach is to model the problem by an optimisation network flow model supported by a multi-period dynamic graph where all data are fixed and no level of uncertainty is considered [83]. Network flow models allow adopting highly efficient computational algorithms even when thousands of variables and constraints are required [6]. In WR management problems, some author explored the possibility of maintaining network flow structure even if non-network constraints are present in the model [58]. Referring to a "static" or single-period situation, we can represent the physical system by a direct network (basic graph), derived from the physical sketch.

Figure 4.1 shows a physical sketch of a simple water system. In the figure, nodes maintain the shape of the common hydraulic notation in order to recall the different function of system components. Nodes could represent sources, demands, reservoirs, groundwater, diversion canal site, a hydro-power station site, etc. Arcs represent the activity connections between them. Physical components corresponding to nodes and arcs can be in the project stage (work planned) and/or operational (existing works with a known dimension). Nodes corresponding to reservoirs represent the system memory since they can store the resource

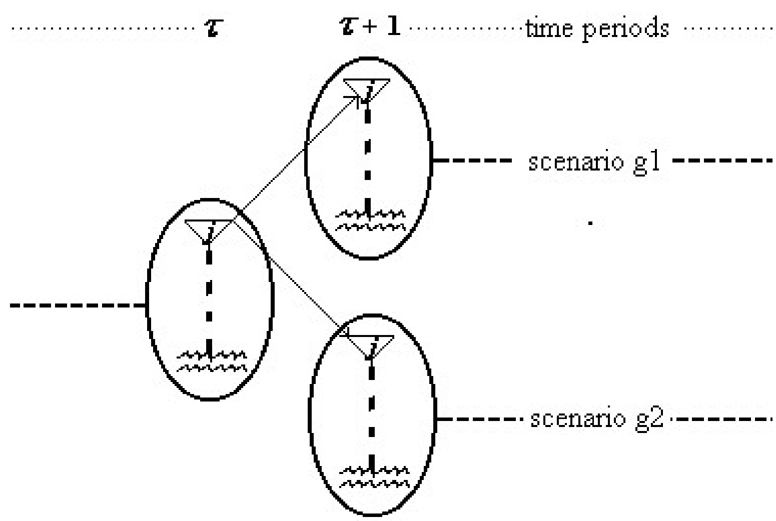

Figure 4.1: Physical sketch

in one period and transfer it in a successive period. A dynamic multi-period network is generated by replicating the basic graph for each period t. We then connect the corresponding reservoir nodes for different consecutive periods by additional arcs carrying water stored at the end of each period. We call these inter-period arcs.

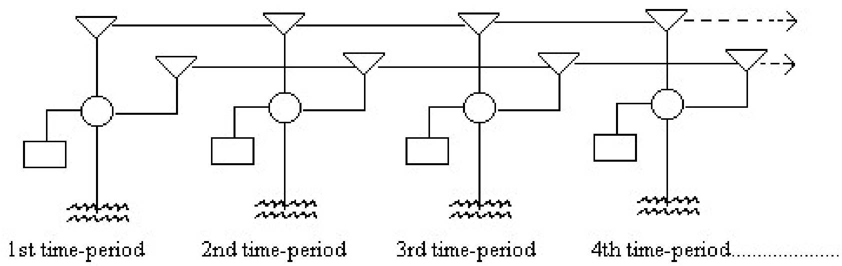

Figure 4.2: Dynamic network sketch generated by simple basic graph

Figure 4.2 shows a segment of a dynamic network generated by the simple basic graph of figure 4.1. Reservoir-nodes are supply-nodes that store and supply the resource. Demand-nodes use and consume the resource. Junction-nodes allow resource passage without consumption. It may be convenient to add "dummy" nodes and arcs to represent not only physical components but also events that may occur in the system.

Figure 4.3 shows the dynamic multi-period network, corresponding to that of figure 4.2, including dummy nodes and arcs marked with a dot. The basic graph is in the frame. The dummy node, U, represents a possible "external system" acting as a supposed source or demand of flow. In this way each arc (i, U) represents a spillway from reservoir-nodes i, each arc (U, i) represents a supposed additional flow in case of shortage in order to meet request in the demand-nodes i and prevent solutions which are not feasible. Flow on arcs (U, i) highlights possible system deficits and the need to modify the dimensions of the works or, alternatively, to make recourse to external water resources. In this paper at first

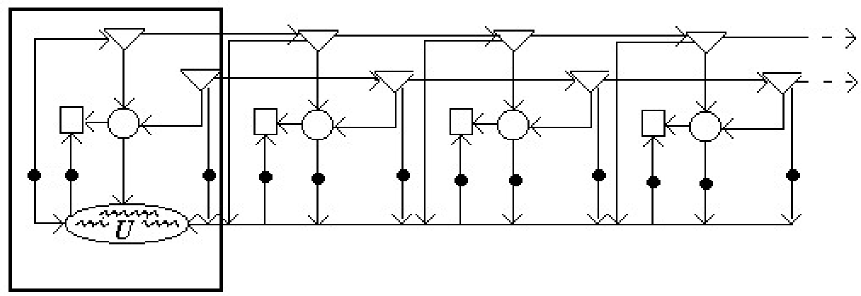

Figure 4.3: Dynamic network generated by simple basic graph

we identifies the components of the deterministic mathematical model and then we provide a formulation for a reduced model that can be adopted to formalize uncertainty in water resources management.

4.2.1 Definition of Water Resources Optimization Model Components

Though it is almost impossible to define a general mathematical model for water resources planning and management problems, our model makes it possible to take into account all possible general system components based on the most typical characterization of these types of models. Different components can be added or deleted updating constraints and objectives. In this paper, we describe only a few of them. A more detailed description of the model supported by a Decision Support System (DSS) can be found in [67] and [84]. Hereafter, we refer to the dynamic network R $= (N, A)$, where N is the set of nodes and A is the set of arcs. T represents the set of time-steps t. Sets of nodes (subsets of N) can represent reservoir nodes, demand nodes (such as civil, industrial, irrigation, etc.), hydroelectric nodes associated with hydroelectric plants, confluence nodes (such as river confluence, withdraw connections for demands satisfaction), etc. Sets of arcs (subsets of A) can represent conveyance work arcs, artificial channels, transfer arcs, spilling arcs, etc. Variables considered in the LP model can be divided into operation and project variables. Operation variables can refer to different types of water transfer (flow on arcs) such as water-transfer in space along an arc connecting different nodes at the same time, water transfer in an arc connecting similar nodes at different times and so on. Project variables are associated to the dimension of future works: reservoir capacities, pipe dimensions, irrigation areas, etc. Constraints in the LP model can represent mass balance equations, demands for the centres of water consumption, evaporation at reservoirs, relations between flow variables and project works, upper and lower bounds on decision variables. To illustrate this concept, we provide hereunder, some details of possible variables, constraints and related data referred to a reservoir-node and a demand node. For a reservoir-node, y^t, represents the portion of water stored in the reservoir at the end of period t that can be used in subsequent periods. A corresponding constraint, for each time period t is:

$$r^t_{min} Y_{max} < y^t < r^t_{max} Y_{max} \tag{4.1}$$

Where Y_{max} represents the max storage volume for inter-period transfer arcs and r^t_{max} (r^t_{min}) represents the ratio between max (min) stored volume in each period t and reservoir capacity. These constraints ensure that, in each period, used volume y^t of the reservoir is in the prescribed range. In an operational state Y_{max} is known while in a project state it is a decision variable. In the latter case it is bounded by:

$$m < Y_{max} < M \tag{4.2}$$

Where M (m) represents the *max* (*min*) allowed capacity. For a demand-node, e.g. a civil demand, p^t represents the water demand at the civil demand centre in period t. A corresponding constraint, for each time period t is:

$$p^t = \eta^t d^t P \tag{4.3}$$

Where P represents the size of the population whose demand can be fulfilled, η^t the request program in each period t and d^t the water requirement for unit of population. These constraints ensure the fulfillment of the demand in each period, no matter if it comes from the system or from dummy resources. In an operational state, P is known while in a project state it is a decision variable. In the latter case it is bounded by:

$$P_{min} < P < P_{max} \tag{4.4}$$

Where $P_{max}(P_{min})$ represents the *max* (*min*) estimated population. Moreover, mass balance constraints are introduced, involving all flows that are going in or out the reservoir or demand node, including hydrological input to the reservoir in each period. The objective function considers costs, benefits, and penalties associated with flow and project variables as well as dummy costs or benefits associated with the dummy components of the multi-period dynamic network.

4.3 A simple system

To illustrate the scenario analysis approach we refer to a sample water system with a reservoir and a demand centre (e.g. civil demand). The supply centre can deliver a resource or store it to deliver in a successive time-period. We assume that dimensions of the reservoirs and the demand centre are known, that is the system is in an operational state. We want to determine the resource management policy over a time horizon such that the known resource demand is satisfied (as much as possible) and the total cost is minimized. Objective function and constraints will be analytically expressed on the basis of the feature of the examined system. Variables of the optimization problem are referred to stored water y_g^t flows transferred from reservoir to demand centre z_g^t, deficits u_g^t in each period t and in each scenario g. Water demand p_g^t is assigned as population P is known. We adopt, as historical data, a hydrological series of 48 monthly time-periods (4 years) reproducing typical behaviour of the Mediterranean system: a wide range of inflow variability between humid and drought periods. Inflows average is adopted as civil demand by the demand centre assuming that the water system is balanced. Moreover, in order to facilitate result interpretation, we assume that the volume, Y_{max}, of the reservoir is large enough to prevent spillage and that evaporation losses are negligible. We generate two scenarios, $g1$ and $g2$, assuming that uncertain parameters correspond to hydrological inflows, inp_g^t, i.e., supplies in reservoir node in period t in scenario g. Costs and bounds associated to variables are considered without uncertainty and, as a consequence, are the same in both scenarios. We generate a scenario-tree with two stages and one branching-time τ. Figure 4.4 shows the scenario-tree for this simple

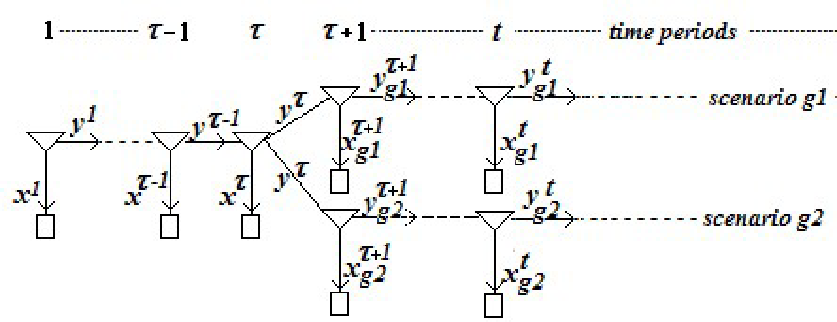

Figure 4.4: Scenario tree

example. In the figure, variables up to branching time t are reported without scenario subscribe because they are the same in the two scenarios, as required by congruity constraints. Dummy node U and dummy arcs, corresponding to deficit u_g^t , are not reported in the figure. The two scenarios are both identical to the historical data up to $\tau = 12^{th}$ time-period. Scenario g_2 follows the historical data from represent costs per unit of flow x_g^t and u_g^t respectively, for $g = g_1, g_2$. The objective is to minimize operative total costs, that is, transfer costs from reservoir to demand centre plus deficit costs. The chance model, for this simple example, can be written as follows:

$$min\ (c^t x_{g1}^t + c^t x_{g2}^t + a^t u_{g1}^t + a^t u_{g2}^t) \tag{4.5}$$

$$r_{min}^t Y_{max} < y_g^t < r_{max}^t Y_{max} \tag{4.6}$$

$$y_g^{t-1} - y_g^t - x_g^t = inp_g^t \tag{4.7}$$

$$x_g^t + u_g^t = p_g^t \tag{4.8}$$

$$y_{g1}^t = y_{g2}^t \tag{4.9}$$

$$x_{g1}^t = x_{g2}^t \tag{4.10}$$

$$u_{g1}^t = u_{g2}^t \tag{4.11}$$

Where constraint (1.8) represents the bounds on stored volumes, (1.9) represents the mass balance in reservoir-node and (1.10) the mass balance in demand-node, with t=2,..,48 and $g = g_1, g_2$. Moreover (1.11), (1.12) and (1.13) represent the congruity constraints (set S in the chance model) with $t = 2,..,\tau$. Hereunder we illustrate some results concerning stored volumes in reservoir, y_g^t, and transferred water to demand centre, x_g^t, obtained by scenario analysis, solving the above optimisation chance model.

Figure 4.5 shows stored volumes with $\tau = 12^{th}$ time-period. Moreover the figure shows stored volumes obtained by an optimisation deterministic model when the scarce scenario g_1 is assumed as database. We call s_1 this independent scenario. The graph referred as scenario s_1 represents decisions, in water trans-

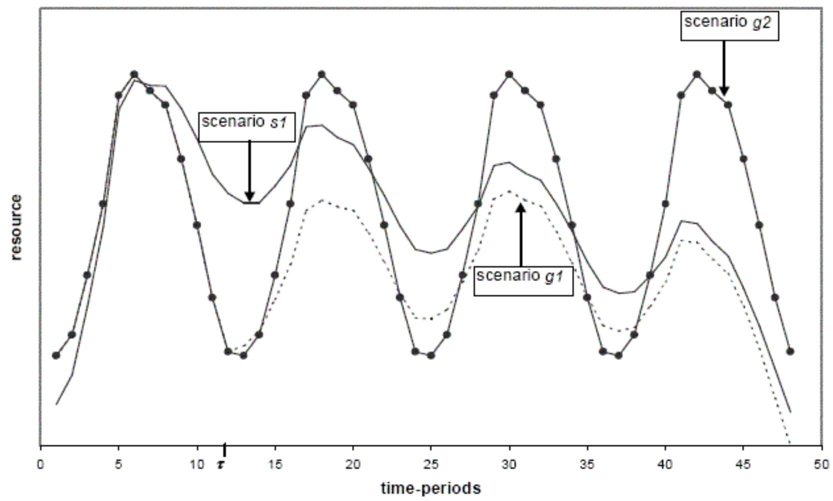

Figure 4.5: Stored volumes

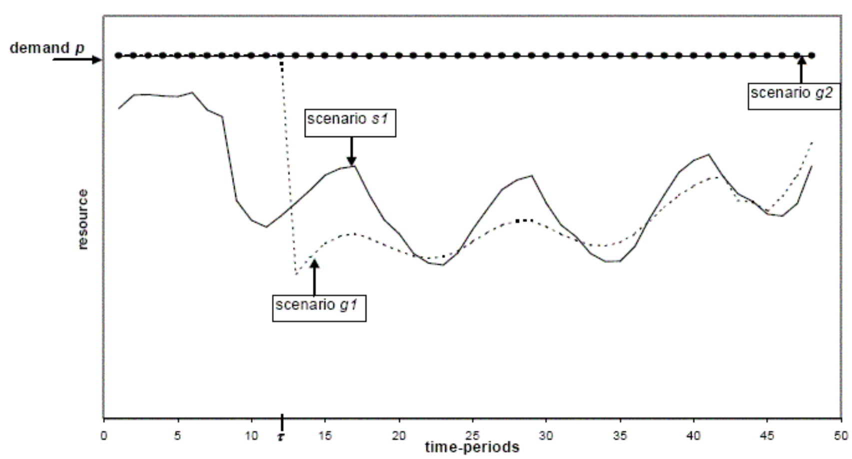

Figure 4.6: Transferred water

fer, in a deterministic optimisation process. The zone (band) between the two graphics of the aggregated scenarios, g_1 and g_2, represents the possible decisions that can be taken for stored volumes. Then we can say that the part of s_1 that does not stay between g_1 and g_2 represents the error that the manager would have made if he had adopted decision s_1. Figure 4.6 shows the transferred water, x_g^t, from reservoir to the demand center in aggregated scenarios, and in the independent scenario. The behaviour of these flows shows that in the scenario g_2 demand is fulfilled while in scenario g_1 deficits are present after branching time τ. But, comparing this with results in deterministic optimization under scenario s_1, we can see that as regards the scarcity of resources conditions, scenario optimization gives a smoother distribution, i.e., with a lower variance of resource distribution in scenario g_1 even though the average is almost the same as scenario s_1. Thus, when planning for scarce resources, scenario analysis provides less dramatic and more easily implementable results then using deterministic optimization to determine management policy.

4.4 A barycentric chance reoptimisation model

In the previous section we showed how scenario analysis could be more useful than the deterministic approach in deciding water management policy. This can be crucial if scarce water resources events occur and a rationing policy must be adopted. But, an effective management policy must be able to establish a target value for delivering resources to the demand centre. The community suffers less from resource rationing if it has been forewarned of a possible shortage. This target value should take into account the entire range of possible scenarios of resource availability, neither too pessimistic in case of abundance, nor too optimistic in case of scarcity of resources. In other words, a target value should be sufficiently barycentric in respect to the different possible scenarios that could take place. Establishing the resource demand level at this target value would permit notifying the resource users (the community) in a timely fashion. As a consequence, preventive measures could be adopted in order to avoid, at least in part, damages derived from an unexpected drastic cut in water resources.

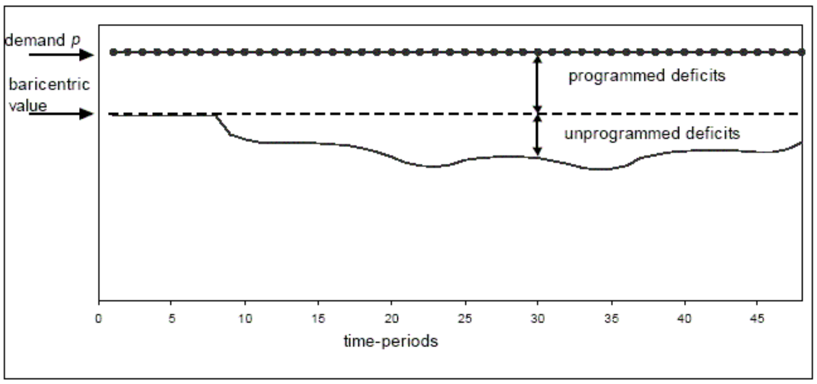

Figure 4.7: Delivered resource

Figure 4.7 shows the resources delivered to demand centre in deterministic reoptimisation, if $\hat{x}_g^t$ are the decision variables representing the resources that can be delivered to a demand centre in time-period t under scenario g, we want to determine a target demand as the value x^b that is barycentric with respect to all $\hat{x}_g^t$. To obtain this value we introduce in the objective function of problem (Pc) a function measuring the weighted distance from x^b to $\hat{x}_g^t$ for all g and t. If we adopt the Euclidean norm to measure this distance, the chance barycentric model can be expressed as:

$$min(\sum_g w_g c_g x_g + \sum_g \sum_t \lambda_g(\hat{x}_g^t - x^b)^2) \tag{4.12}$$
$$A_g x_g = b_g, \qquad \forall\, g \,\in\, G$$
$$l_g < x_g < u_g, \qquad \forall\, g \,\in\, G$$
$$x \,\in\, S$$

where λ_g is the weight associated to the norm. Once the value x^b is determined, a re-optimization process can be adopted in order to identify the sensitivity of the examined system with respect to deficit programming. We construct a deterministic dynamic model in which the predefined demand is settled equal to the barycentric value x^b and adopting as data input, those corresponding to the most crucial scenario (e.g. what the manager considers the most risky for the system). The difference between the new configuration of delivered resources in each time-period t and the value x^b, identifies the set of programmed deficits for the system. In the sample system illustrated in the previous section we determine a value z^b in such a way that it is barycentric with respect to all z_g^t. then, we reoptimize the system solving a deterministic model assigning to the demand centre the obtained value z^b as target value and adopt, as data input, those corresponding to scarce scenario. Figure 4.7 shows the resources delivered to the demand centre in the reoptimization phase together with the programmed deficits (difference between the new configuration of delivered resources in each time-period t and the value x^b) and unprogrammed deficits (difference between the original resource demand and the value x^b). Moreover, comparing the behaviour of delivered resources with that showed in figure 4.6, we observe that management policy is even better than the policy corresponding to scenario g_2. The programming of deficits makes it possible to set up adequate preventive measures, which permit a notable reduction in the event of resources scarcity.

4.5 Test Case II: A Real Physical System

Scenario analysis was performed on the FCC (Flumendosa, Campidano, Cixerri) system in Sardinia, Italy.

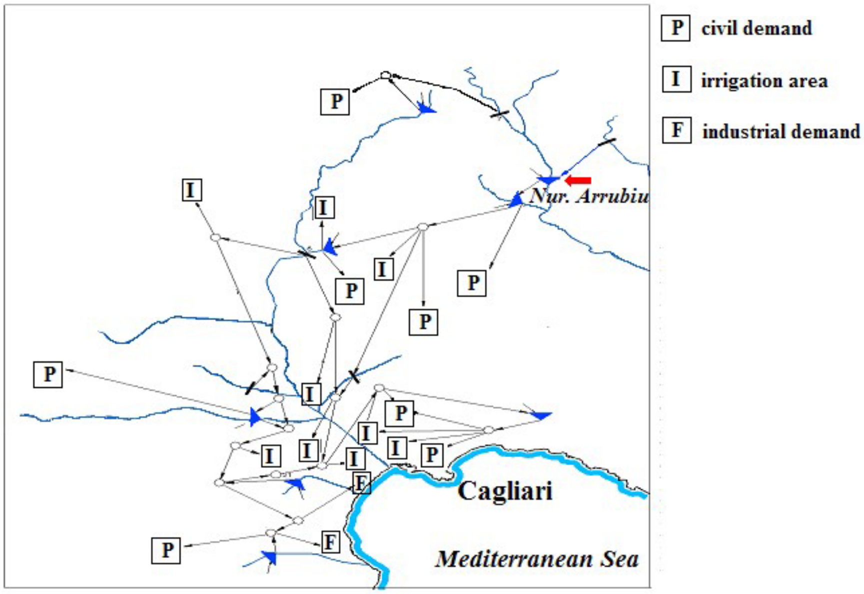

Figure 4.8: System Flumendosa-Campidano-Cixerri

Figure 4.8 shows a sketch of the system. Since 1987, the Sardinia-Water-Plan has highlighted the necessity of defining an optimal water works assessment and the urgency of defining optimal management rules for the water system. Correct evaluation of system performances and requirements became increasingly urgent, as system managers were obliged to face serious resource deficits caused by the drought events of the past decade accompanied by an almost total uncertainty in hydrological inflows. The main infrastructures were built in the mid 50's and supply most of southern Sardinia. The main water supply source of the system is represented by the three reservoirs with a total storage capacity of 666.4 million cubic meters (Mm3). Gravity galleries connect the reservoirs. Total yearly average distributed volume in the period examined is 225 Mm3 for civil, industrial and agricultural demands. No significant aquifers are present in the system. The principal works can be identified in 13 water supply sources (dams and weirs), 3 water diversion and communication galleries, 5 main water supply and distribution networks, 2 hydroelectric power stations, 1 irrigation distribution network, 11 pumping stations, and 2 drinking water plants. A number of synthetic series is generated and adopted as the set of predefined scenarios. The basic hydrological data is derived from the report in RAS (2003) and different scenario generation techniques have been compared. Starting from a database with a time-horizon up to 75 years, corresponding to 900 monthly time-periods, a set of 30 scenarios was submitted to statistical validation and selected. Scenario analysis was performed on a scenario-tree of 2 and 3 stages up to 30 leaves. Since each scenario involves about 3,000 variables, the chance model supports several thousands of variables and constraints. In this paper, I report some results selected among a wide set of output information. In particular, we report some results obtained adopting a time-horizon of 48 time-periods and a branching time in the 12th time period as in the sample system. Illustrated results are referred to the "Nuraghe Arrubiu" reservoir (the red arrow in figure 4.8) that is considered one of the main pivots of the system as it can control water transfers to the principal demand centres. Two scenarios are deduced from the last 4 years of hydrological inflows reported in [70]. We adopt these data as scenario g_2 while scenario g_1 is derived from it assuming that a reduction of 50 percent will occur after the branching-time.

Figure 4.9 shows the behaviour of stored volumes obtained by scenario analysis (aggregation of g_1 and g_2) and the behaviour obtained by deterministic optimisation using the reduced independent scenario s_1. The figure shows that decision policy, corresponding to deterministic optimization, induces an early empty reservoir with respect to the decision policy given by scenario analysis

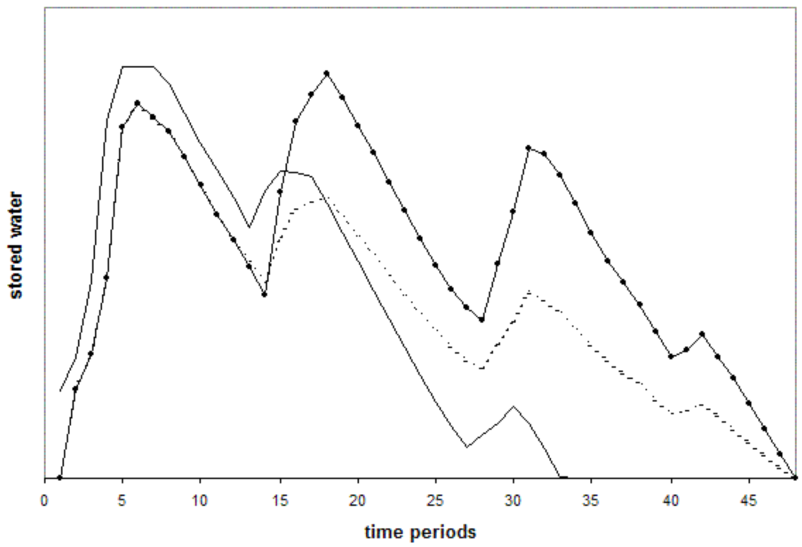

Figure 4.9: Stored volumes

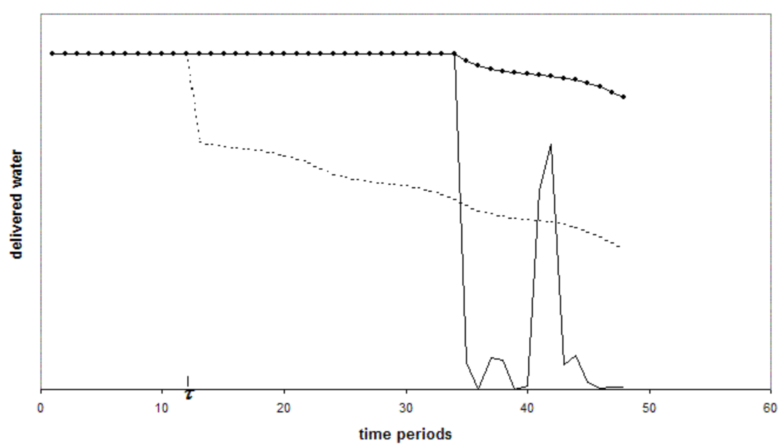

Figure 4.10: Transferred water

in the equivalent scenario g_1. This corresponds to the behaviour of transferred water to demand centres as illustrated in figure 4.10. As in the sample system, decision policy g_1, resulting from scenario analysis, exhibits smoother resources distribution and a lower variance with respect to deterministic policy s_1. The analysis of the FCC system by barycentric chance reoptimisation is in progress.

4.6 Conclusions

In this chapter I showed how scenario analysis can be more useful than the deterministic approach in deciding water system management policy when a level of uncertainty affects data input such as supply and demand patterns. Decision policy under uncertainty condition can be crucial if scarce water resources events occur and a rationing program must be adopted. The scenario analysis approach considers a set of statistically independent scenarios, and exploits the inner structure of their temporal evolution in order to obtain a "robust" decision policy, in the sense that the risk of wrong decisions is minimized. This can be done by a re-optimization deterministic process using a barycentric value derived from a previous scenario optimization. Finally, this make it possible to identify programmed deficits to control the negative consequences deriving from wrong decisions allowing the system manager to adopt preventive measures avoiding, at least in part, damages derived from an unexpected drastic cut in water resources.

Chapter 5

Second application: an application of scenario analysis to air traffic delay management

5.1 Introdution

Air Traffic Delays occur for a variety of reasons ranging from bad weather conditions, such as runway closure, de-icing, cross-wind limitation (nobody's responsability), to technical/operational problems with flight preparation (that is the aircraft operator's responsability), customer service issues, air traffic control system decisions, equipment failures, airport congestion and lack of Air Traffic Management (ATM) capacity (i.e.: the inability of certain Air Traffic Control Units to handle all the flights wanting to cross the airspace at a certain time). Air Traffic Management System must provide Air Navigation Services to a certain volume of air traffic, in line with the targeted high level of safety and without imposing significant operational, economic or environmental penalties under normal circumstances.

5.2 Air traffic management outline

The aim of the ATM is to maintain efficient the flow in the air traffic system and the safety measures, protecting air traffic services from over-deliveries, while at the same time enabling aircraft operators to carry out their intended flight operations with the minimum penalty. Examples of operational penalties are: ground delay, assignment of a cruise level different from the optimum one during a long period of flight, extra mileage, holding pattern. Both Unpredictability and predicability events responsible of airports congestion and lack of ATM capacity can be classified as:

- Catastrophic events: relatively rare, major events typically related to factors external to system operation, such as: terroristic attacks (e.g. 11/09/00) or labor strikes that disrupt operations on a large scale, computer fault (e.g. 24/08/05, 03/06/04, the system produces paper slips which tell air traffic controllers each individual aircraft's height, route, destination and contact information, allowing them to direct the planes correctly. When it is not working, controllers have to produce these forms by hand so it cause delays and backlogs).

- Complicatedness: Many factors interact and affect system behaviour, making it difficult to predict what will happen, a typical manifestation is the interacting of traffic flow in congested air airspace and around congested airports.

- Criticality: system elements are often "near the edge" so behavior is sensitive to small perturbations, it this case demand can be near or above capacity for some NAS resource, making the system sensitive to small changes.

- Distributed adaptive decision making: airspace users (multiple decision-makers) acting in self-interest adapt to changing circumstances, but may over-congesting system resources making the system hard to predict.

- "Shaky-hand" effects: this is when actions taken produce large errors (deliberate or accidental) in execution such as non compliance to initiatives or ineffective communication of an initiative that produce large variance in actual aircraft arrival time compared to scheduled times.

- "blurred vision" effects: decision are made based on imperfect information, this can generate for example problem in assigning airline priorities, weather and demand imperfect forecast. Negative economic conjuncture can produce an important impact in the number of delays, as well. This was observed in the United States airports during 2001, when the decline of delays was far greater than the decline in the number of operations because both the system and many of the largest airports had been operating at or near their theoretical capacity. In these cases, the decrease in the number of operations has had a disproportionate impact on delays. A related response was observed in 1999 and 2000, when a relatively small increase in the number of operations produced a large increase in the number of delays. For many years, in Europe the delay was due to the fact that the air traffic control system was not adapt to manage the continuous growth of the civil air traffic, this fact produced yearly a large quantity of delay. Figure 5.1 reports the portion of each cause of delay on the total delay in 2005. The goal of Air Traffic Flow Managment (ATFM) is to apportion capacity to minimize disadvantageous effects when capacity is reduced. The flow management problem in ATFM occurs when flights arriving at an airport must be delayed because that

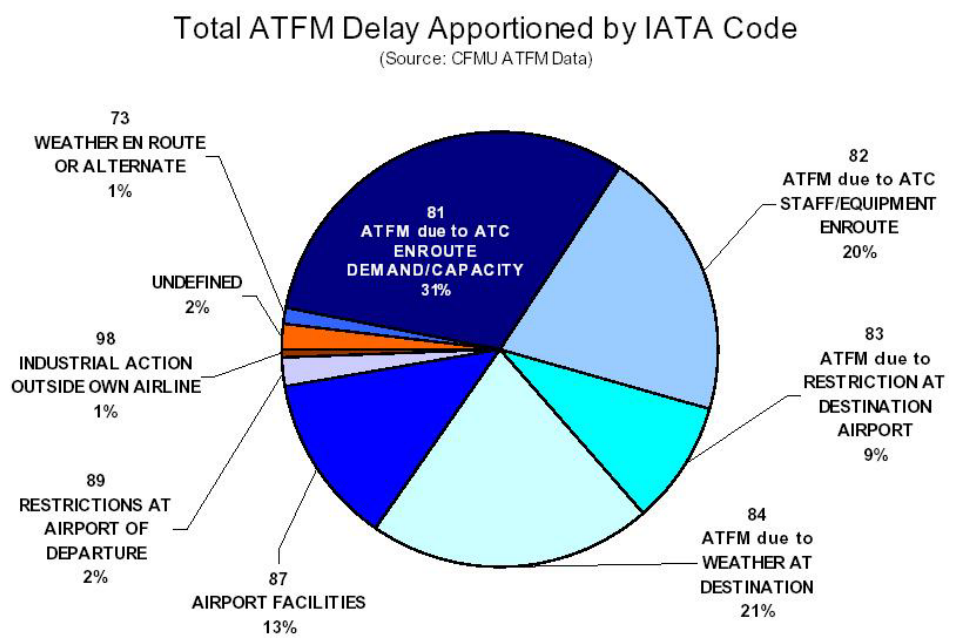

Figure 5.1: Total delay

airport suffers of reduced capacity. Hence, ATFM try to reduce congestion delay effects as-needed while optimizing capacity. The CFMU is now evolving towards air traffic flow and capacity management, where the emphasis is on optimizing available capacity whilst maintaining the safety levels required by European air navigation service providers (ANSPs). These objectives are achieved by assessing the planned traffic load and, where necessary, acting upon this planned traffic data through the implementation of ATFM measures. In Europe the ATFM is covered by the CFMU and in the USA Air Traffic Control System Command Center (ATCSCC), in Europe this service is paid by passengers. Facility construction, fundamental procedural changes, improvement in navigational equipment, new transport aircraft, new landing procedures are long term solutions. In Europe, an important event that has given a significant contribute to reduce the air delay is the implementation of Reduced Vertical Separation Minimal (RVSM), the most important structural change to the European air traffic control system in 50 years, allowing more aircraft to fly in the same volume of airspace, allowing more aircraft to operate at fuel-efficient altitudes, helping decrease aircraft emissions by between 1 and 2 percent. Thanks to this initiative in these years the delay due en-route has been reduced consistently. This solution has been activated from 2002 and has increased capacity in upper airspace by 14 percent. An other important contribution towards the effectiveness of the CFMU's role will be the collaborative decision-making between all the players. This inclusive and transparent process has enabled trust to be built between the CFMU, aircraft operators and ANSPs. In 2002, this process in USA was used extensively to address areas of congestion. These developments allow greater efficiency to be achieved throughout the entire flow management process. They will also provide the aircraft operator with the ability to interact with these processes and use them to his advantage without requiring detailed knowledge of what has already become a very complex system.

Unfortunately, the delays reduction involve only en-route ATFM delays. As result, the proportion of the airports delay, was in the last five years was around 22 percent of the ATFM delays, it has increased significantly reaching 33 percent in 2002, 45 percent in 2003 and 48 percent in 2004, as reported in figure 5.2. In order to reduce both delay and operational costs in short time and improve the congestioned airport capacity, a tactical procedure can suggest a model based on both ground holding and holding pattern assigments. However, a tactical optimization is a difficult process which require a practical implementation of the model and the solutions strictly depends on the airport capacity uncertainty. Methodological approaches used to deal with unpredictability are: scenarios analysis, agent-based modelling, sensitivity analysis, game-theoretic modelling, bayesian networks. In a scenarios analysis context a tactical strategy which explores all the possibly scenarios returning a quickly solution is important to aid the controllers to take tactical decisions. In this case, the solutions strictly depends on the uncertainty parameters regarding the future airport capacity due to some congestion cause. In this thesis, I introduce a tactical flow management procedure for the Ground Holding Problem (GHP) which optimize the arrival/departure daily plan, based on scenario analysis, trying to manage the uncertain future events that could reduce the runway capacity, congestioning the airport. The tactical flow and capacity management model minimizes the costs associated

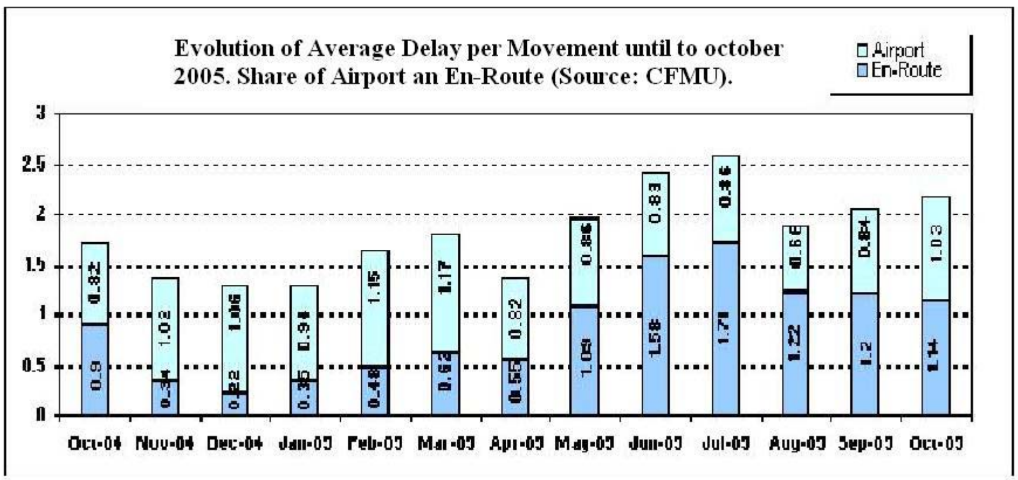

Figure 5.2: airport/enroute delay distribution

to delays and reduces the domino effects. This model reacts to unpredicabilty negative future events, minimizing disruptions impact and/or taking benefit of opportunities.

5.3 Literature review

If airport capacity was deterministic, we would be able to precisely determine the Airport Arrival Rate (AAR) for each time period, and could use the exact AAR to make optimal decisions regarding a new flight schedule. However, Airport Arrival Rate in case specific events is difficult to calculate. GHP is a specific example of the generic flow management problem where the objective is to minimize the expected total cost of delay by assigning delays to aircraft. The GHP is stochastic since the airport capacities are random and is dynamic since the weight associated to capacities change with time. Literature reports several decision support for traffic flow management, in [66] general flow management is posed. [66] provides a comprehensive examination of the flow management problem, this paper outlines the stochastic and dynamic problem characteristic and proposes several methods for approaching the problem. [66] proposes that the generic flow management problem can use any objective function related to ground and air delays and further suggests that the natural objective function is that which minimizes the total cost of delays. Although GHP is a stochastic and dynamic problem, deterministic formulations are often used as a first step in solving the GHP. In this case, the capacity for each time period is assumed to be known. Andreatta and Romanin-Jacur [4] gave a linear program formulation and propose a polynomial algorithm which finds an optimal ground delay decision for the single airport GHP. [72] provides a deterministic linear program formulation for the single airport GHP. In [86] is formulated the GHP for a single airport as an integer program and solve the integer program as a minimum cost flow problem. The single airport GHP has been expanded to consider multiple airports as well. Bertsimas and Stock Patterson [11] propose an integer program formulation of the multi-airport GHP and solve this problem using a linear program relaxation. They are able to solve realistic size problems in an acceptable amount of time since most solutions to the linear program relaxation were integral. Navazio and Romanin-Jacur [64] formulate the deterministic multi-airport GHP with banking constraints as an integer program. Due to high computation time, they provide an alternative heuristic algorithm. Vranas, et al. [13] propose an integer program formulation and solve the multi-airport GHP using a linear program relaxation. They solved the problem for six airports and 3000 flights. Several stochastic models have been proposed to solve the GHP. Andreatta and Romanin-Jacur [4] adapt their polynomial solution algorithm to solve a single airport GHP for one period where the capacity is stochastic with a known distribution. Terrab and Odoni [86] explore using a dynamic program to find the exact solution to the probabilistic formulation, but were constrained by time and memory since they consider each flight individually. They further suggest using several heuristic algorithms to solve the stochastic GHP. Their heuristic algorithms were able to provide reasonable solutions. Several models also include dynamic updates of weather information which may be used to modify ground holding decisions periodically. Vranas, et al. [13] extend their research from [12] by modifying their linear program relaxation of the integer program to include dynamic updating. To solve the stochastic dynamic version of GHP, they select either the most likely or worst case policy and then evaluate that policy against the stochastic capacities. Furthermore, they give a dynamic first-come first-serve heuristic for the stochastic problem but this heuristic was highly inefficient. Panayiotou and Cassandras [68] use a finite perturbation analysis to solve the single airport GHP dynamically using an event-driven model to assign ground delays as opposed to a time-driven model used in integer, linear, and dynamic programming approaches. Since their approach is scalable, they were able to solve a realistic size problem. Unlike the previous papers which consider flights individually, Richetta [71] and Richetta and Odoni ([72], [73]) make decisions on a collection of flights. They first solve the probabilistic GHP deterministically by selecting the most likely capacity profile and solving the deterministic linear program. This formulation is the foundation of the policy-based approach discussed in Chapter 3. Next, Richetta and Odoni provide a stochastic linear program formulation of the single airport multi-period dynamic stochastic GHP. The resulting stochastic linear program is easily solved by transforming the problem to a minimum cost network flow model. [68], this formulation requires time discretization. Both [40] and [4] give ground delay model based on dynamic programming that incorporate uncertainty in forecasting capacity. Unfortunately, these model grow exponentially complex with the number of time periods, making them virtually impossible to solve for real-sized problems. Moreover,[73], [13], [12] are based essentially on assignment models, they assign a departure time to each fly. Next, [42] focus on the ground delays problem to find a good trade-off between the ground delays introducing the airborne hold. [42] proposes a lagrangian decomposition based algorithm. Given a flight schedule and a probability distribution of AAR profiles, the decision makers must decide how to alter the flight schedule by instituting both ground holding and holding patterns to accommodate a probable reduction in capacity. If I deal all the scenarios separately, a possible policy decision is an AAR for each time period. As an example, the decision may be to do nothing and keep the AAR at the original value. Another decision may be to select the most likely AAR or the expected AAR based on the AAR profile distribution. In this thesis I present a model that consider the scenario analysis throughout the scenario tree generation.

5.4 The air traffic delay model

Given a congestioned airport, I afford the GHP on a given horizon time, typically one day. I propose a methodological approach in order to reduce the negative effects due to unpredicability events, minimizing delays and maximizing air system components efficiency. When an airport is congestioned for a interval of time, the runway future capacity is reduced, but is not known in practice. In these cases the scenario analysis can be used to model the problem, in order to obtain a robust solution for each fan of considered scenarios. Scenario analysis model supports the decision maker to decide how much time to assign, in terms of ground holding at departures airports and airborne holding at arrival airport, assuring an high level of safety. Assuming the airplanes have all the same priority, a simple high cost solution can be obtained by assigning the slots with a FIFO (First In First Out) politic. Available slots are assigned to first airplane arrived in the aerodrome airspace and by assigning, in some manner, a ground holding to all airplanes waiting the take off at departure airport. this politic a possible drawback is that the delay costs could not be well distributed among the airplanes, without to exploit in optimal manner the residual runway capacity. Both ground holding and airborne holding can be distributed in optimal manner assigning the delays among the airplanes, minimizing the total cost of the system. Assuming the horizon time T divided in equidistant instants of times (1,2,...), typically in air traffic this interval is taken as 15 minutes, this because $[t_{arrival},\ t_{arrival}+15]$, is the interval of time the fly is considered in time, where $t_{arrival}$ represents the scheduled time of landing. We model the assignment of the airplane delay by using a direct multiperiod graph, representing a temporal network, where a number Ω of airplanes are in competition among them in order to obtain a future slot either to land or to take off. Each arc contains a set of information that is the linking cost, the upper bound capacity. Given a set of starting airports and a destination airport, a tactical approach permits to allocate the airplane departures, by assigning, for each fly, a possible both ground holding and airborne holding, in the right manner, minimizing the total delay cost of the system. As slots are assigned to airplane having the minimum cost, airplanes are in competition among them to obtain the resource. Such approach permits to decision maker to use in optimal manner the resources of landing airport in case of congestion. Information returned by scenario tree optimization is useful to:

* identify the "critical time". The airplane take off before the critical

time have not delay assigned and the landing time is that scheduled. Airplanes taking off after the "critical time" are subjected to delay, the number of airplane to deal requires to utilize extra-resources of the air-traffic system. This is important in ATF management because it is an exact information about which set of airplanes are involved in the restriction. In a collaborative decision making environment, this information permits to controllers to know, rapidly, the other airports involved in the the ground delays restrictions. Once known the airplanes involved in delays restriction pointed out by optimization process, departure airport controllers must be involved in a process of collaborative decision making, to decide if the solution is effectively applicable. The "negotiation" among decision makers (controllers) involved in the control regards of other available airspace system components;

∗ view both which airplanes will be in airborne holding and the segment of time the runway works near the maximum capacity. This is very important because is known that runways working near their maximum capacity generate large scale delays and an overwork of the air traffic control resources. Moreover, in such situation there is a risk to produce domino effects, that is delays are distributed in other areas of the airspace system.

As shown in figure 5.3, we model the determistic problem with a graph $G = (N, A)$ with $|N| = n$ nodes and $|A| = m$ arcs. The model at each starting airport associates a column of nodes representing sequentially different periods of time, nodes at same levels represent same instants of time called periods. Arrival airport is modeled by utilizing two temporal columns, such columns are linked among them by direct diagonal arcs, in order to put in evidence the arcs model the runway and its actual capacity, vertical nodes in the first columns model the airborne holding, in the second columns model landing. Given two instants of time, time(i) and time(j), with time(j)>time(i) since the graph is direct, we model various types of arcs. Because the optimization model have to decide a path, it is introduced a decision variables $x_{i,j}$, it values 1 if the airplane runs through the arc (i,j) and 0 otherwise. In order to better understand the meaning of the arcs (i,j), I introduce the notation $(i,j)^{o,d}$, where o means a departure airport and d a destination airport, if $o = d$, then it means a temporal transfer, that is a delay assignment. The meaning of the principal arcs is the following:

* $(i,j)^{o,d}$, $j = i + 1$, $o = d$, d is a departure airport: ground holding arc;
* $(i,j)^{o,d}$, $j = i + 1$, $o = d$, d is the arrival airport: airborne holding arc;
* $(i,j)^{o,d}$, o is a departure airport, d is the arrival airport: linking arc between airport, a linking arc exists among two nodes if the time difference between the node covers the fly duration;
* $(i,j)^{o,d}$, $j = 1 + 1$, o and d are node in the first and the second column of the arrival airport: runway arc. The capacity associated depends on the AAR value.

In this model the associated costs for both airborne and ground holding are kept high while the other arc costs are kept low. Each airplane is represented by a certain commodity and the take off is scheduled at a certain time, in terms of network flow this event is model as an entering flow in the node that represents the take off time. Since it is important to know the path of each airplane, for each commodity r, an arc $x_{i,j}^{o,d,r}$, simplified as $x_{i,j}^{r}$, models the decision binary variable regarding the fly starting from node i and ending to node j. Fixed an arc (i, j) and a scenario r, the parameter $u_{i,j}^{r}$ represents the upper limit of mutual capacity. In our model runway uncertainty is described changing the value of mutual capacity for all arcs modelling the runway. Each particular realization of the future runway capacity is modeled as an independent scenario. As example, in figure 5.4 is reported a bundle of direct temporal networks, where each direct temporal network represents a different scenario. Let K a set of different scenarios, $u_{i,j}^{k}$, $k = 1, .., |K|$, the mutual capacity associated to scenario k, it is defined as $\tau(k, k') = max\{t|\ u_{i,j}^{k} = u_{i,j}^{k'},\ \ k, k' \in K\}$ the last time the mutual capacity is identical for all scenarios. In other words for each pair of scenarios (k, k'), the time $\tau(k, k')$ characterises the branching time among two scenarios. The instant of time $min\{\tau(k, k')\}$ divides two different stages. We assume the model contains a set Ω of commodities, and a set K of different scenarios. Fixed an arc $(i, j) \in A$, a commodity h and a scenario k, we adopt the notation: $x_{i,j}^{h,k}$, $u_{i,j}^{h,k}$, $c_{i,j}^{h,k}$ to describe respectively, the decision variable, the single capacity, the cost. Now, we define for each pair (k, k') of scenarios, $(k, k') \in K$ the set $S_1 = \{(i, j)|\ time(i) < \tau(k, k')\ and\ time(j) \leq \tau(k, k')\}$, $S_2 = \{(i, j)|\ time(i) \geq \tau(k, k')\}$, the set of arcs in A whose both tail node and head node belong to either first stage or second stage. Moreover, we define as $S_{(1,2)} = \{(i, j)|\ time(i) < \tau(k, k')\ and\ time(j) \geq \tau(k, k')\}$, the set of arcs whose tail node belongs to first stage and the head node belongs to second stage. We adopt temporal direct network scenarios with a topology as in figure 5.4, where the grouping process produces two-stage scenario trees, as reported in figure 5.5.

The air traffic delay problem under uncertainty capacity can be modeled in the following way:

Model (AR)

$$min_x \sum_{(i,j)\,\in\, A} \sum_{h=1}^{|\Omega|} \sum_{k=1}^{|K|} w_k c_{i,j}^{h,k} x_{i,j}^{h,k}$$

$$\sum_{j:\,(i,j)\in A} x_{i,j}^{h,k} - \sum_{j:\,(j,i)\in A} x_{j,i}^{h,k} = b_i^{h,k},\ i = 1, ..., |N|,\ \forall h \in \Omega,\ \forall k \in K$$

$$\sum_{h\in\Omega} x_{i,j}^{h,k} \leq u_{i,j}^{k},\ \forall (i,j) \in A,\ \forall k \in K$$

$$0 \leq x_{i,j}^{h,k} \leq u_{i,j}^{h,k},\ \forall (i,j) \in A,\ \forall k \in K,\ \forall h \in \Omega$$

$$x_{s,t}^{h,k} = x_{s,t}^{h,k+1},\ \forall\ (s,t) \in S_1,\ k = 1, .., |K-1|,\ \forall h \in \Omega$$

$$x_{s,t}^{h,k} = x_{s,t}^{h,k+1},\ \forall\ (i,j),(s,t) \in S_{1,2},\ k = 1, .., |K-1|,\ \forall h \in \Omega$$

$$x_{i,j}^{h,k}\ \in\ \{0,1\},\ \forall\ (i,j) \in A,\ \forall k \in K, \forall h \in \Omega$$

Where the vector $b_i^{h,k}$, of proper dimension, in the right side of the mass balance constraints, for each fixed scenario k and commodity h is the vector of the supply/deficit parameters. Besides the mass balance constraints, we have to satisfy both the single commodity capacity and the mutual capacity constraints. In the model, each scenario defers from the others, for the mutual capacity parameters $u_{i,j}^{k}$, assigned to runway arcs and the parameter w_k describing the "subjective" probability associated to each scenario. Non anticipativity constraints involve the set of arcs belonging to S_1 and $S_{1,2}$. Assuming we dualize the non anticipativity constraints for arcs in $S_{(1,2)}$, using a new vector y, with $D = (|\Omega||S_{(1,2)}||K-1|)$ dimension, we obtain:

Model (ASR)

$$max_{y \in \Re^{|D|}} min_x \sum_{(i,j) \in A} \sum_{h=1}^{|\Omega|} \sum_{k=1}^{|K|} w_k^{h,k} c_{i,j}^{h,k} x_{i,j}^{h,k} +$$

$$+ \sum_{(s,t) \in S_{1,2}} \sum_{k=1}^{|K-1|} \sum_{h=1}^{|\Omega|} y(x_{s,t}^{h,k} - x_{s,t}^{h,k+1})$$

$$\sum_{j:\,(i,j) \in A} x_{i,j}^{h,k} - \sum_{j:\,(j,i) \in A} x_{j,i}^{h,k} = b_i^{h,k},\ i = 1, ..., |N|,\ \forall\, h \in \Omega,\ \forall\, k \in K$$

$$\sum_{h \in \Omega} x_{i,j}^{h,k} \leq u_{i,j}^k,\ \forall\, (i,j) \in A,\ \forall\, k \in K$$

$$0 \leq x_{i,j}^{h,k} \leq u_{i,j}^{h,k},\ \forall\, (i,j) \in A,\ \forall\, h \in \Omega,$$

$$x_{s,t}^{h,k} = x_{s,t}^{h,k+1},\ \forall (s,t) \in S_1,\ \forall\, k = 1, .., |K-1|, \forall\, h \in \Omega,\ \forall\, k \in K$$

$$x_{i,j}^{h,k} \in \{0,1\},\ \forall\, (i,j) \in A,\ \forall\, k \in K,\ \forall h \in \Omega$$

Supposing a problem having a single-commodity and two linking-arcs (s_1, t_1), (s_2, t_2), and three commodities, the components of y associated to each arc are:

$$y_1(x_{s_1,t_1}^{1,1} - x_{s_1,t_1}^{1,2}) = 0$$

$$y_2(x_{s_1,t_1}^{1,2} - x_{s_1,t_1}^{1,3}) = 0$$

$$y_3(x_{s_1,t_1}^{1,3} - x_{s_1,t_1}^{1,1}) = 0$$

$$y_4(x_{s_2,t_2}^{1,1} - x_{s_2,t_2}^{1,2}) = 0$$

$$y_5(x_{s_2,t_2}^{1,2} - x_{s_2,t_2}^{1,3}) = 0$$

$$y_6(x_{s_2,t_2}^{1,3} - x_{s_2,t_2}^{1,1}) = 0$$

Hence, the dimension of the dual vector y depends on the number of arcs in $S_{1,2}$. If we relax the congruity constraints for the arcs in $S_{1,2}$, the congruity constraints for arcs in S_1 become redundants. Now, if we fix y, it is possible to solve a set of independent multicommodity subproblems of reduced dimension as reported in figure 5.6.

5.5 A decomposition algorithm

When we deal with complex models, the choice of a good methodological approach is a critical phase, especially if a fast solution is needed. Many practical problems need of a general directive, about the best approach in terms of quantitative methods. In order to evaluate different resolutive approaches, I have developed a framework to handle scenario trees, composed of GHP instances using different resolution approaches. The framework has been developed in C++ language. It has been implemented following the open source philosophy and standardization rules adopted in deterministic multicommodity source codes ([60]). I exploited the encapsulation and polymorphism features inhering C++ languange developing an articulated code, where the core of the data structure is represented by a set of abstract classes. In order to compare results obtained by using the decomposition solver, the source code of MMCFCplex (a solver for deterministic multicommodity problems downloadble on sorsa.unica.it) has been adapted to solve scenario tree instances. MMCFCplex defines an interface between the solver Cplex and the multicommodity instances and accepts instances written in a standard format. The original source code has been modified and extended by introducing new methods, such methods permit to manage a given set of scenarios and to insert the non anticipativity set of constraints for each given set of scenarios, in a standard format. I called this package MMCFSCCplx. In MMCFSCCplx, both network topology and data parameters about multicommodity instances are maintained in a object of type MMCFCplex. The class MMCFCplex derived from basis class MMCFClass and inherits from it most of its data and code.

The derived methods (publics, private and protected) are redefined in part exploiting the mechanism of the virtual functions. The methods included in MMCFClass permit to:

* read the input data of each associated multicommodity flow problem;
* change the input data of the problem without to reset the proceed obtained solutions (re-optimization);
* choice one of the multicommodity solver available, at compile time;
* read the results obtained from optimization process.

The decomposition based code called MMCFSCDcmp and it is organized in different independent modules. Modular programming permits to include, if available, new optimization computer code in the software package MMCFSCDcmp, implementing a new class derived from basis class. New available modules can be included in MMCFSCDcmp writing in C++ language the original code following standardization rules, rewriting associated both virtual and private methods and updating the makefile in the proper manner. Hence, this package permits to decide at compile time which solver to use, choicing among a set of different available solvers. The goal is to assess the solver effectiveness comparing together, different both non-smooth function and multicommodity solvers, that it is not known in practice. The architecture of MMCFSCDcmp is composed of different directories:

* Main: it contains the file Main.C, this file reads $k+1$ input instances. The first instance describes the first stage parameters, the other instances describe the second stage parameters associated to

scenarios. Assuming k (the number of scenarios) object of Class MMCFClass (the abstract Class MMCFClass implements a wrapper class for the multicommodity problems) are generated.

* SCNRBundle: it contains the file SCNRBundle.C implementing an interface between multicommodity and non differentiable function solvers. The class SCNRBundle is defined as Bundle derived class. The public methods SetGi(.) and Fi(.) represent the only "point" to load the results (lagrangian dual vector) coming from bundle solver and to return to the bundle the results obtained from optimization process of the multicommodity instances. As reported in figure 5.7, an object of type SCNRBundle works as coordinator, sending, updating and receiving both data and results with the other objects. At each iteration, everyone of the $k+1$ objects of class MMCFClass receive from the coordinator process the new dual vector y, update their own costs, re-solve the multicommodity instance and resend the obtained solution x to the coordinator. Then, the coordinator calculates the new subgradient and send them to the bundle object. At each iteration, the best found value of the dual lagrangian function is maintained and if necessary updated.

* Graph: it contains the Graph.C, this file implements the methods used to read the files containing the input parameters in different standard format.

* Bundle: this class contains the basis bundle class and methods which implement the non-smooth functions solver.

* MMCFCplex: this directory contains the definition of the class MMCFCplex derived from MMCFClass, representing a basis class for

the linear multicommodity solver, this class contains the description of a set of "pure" virtual methods proper, the code (data) of each available solver is included in a file derived from basis class.

Starting from the set of input instances, it has been implemented a procedure to pprn format, in order to obtain a easily readable format of the scenario tree structure. This format is used as input instance in the MMCFSCCplex solver. All basis source codes are available on-line.

5.6 Computational experience

Once, I decided the standard to write instances and implemented the various modules about the decomposition algorithm, I wrote a converter from a quasi-pprn format to a pprn standard format and a generator of GHP instances. The goal is to understand, through the computational experience, if an instance regarding the GHP can be solved either using directly a scenario tree or the given decomposition approach. The advice to return depends on the characteristic of the problem. The dimension of each instance depends on the number of departure airports, the number of airplanes, the number of periods, the number of scenarios. In this thesis I decided to report only the results about the instances that can have the same dimension of possible real instances. The experiments were executed on a Pentium IV 1.7 GHz 256 Mb computer with a Linux RedHat 7.2 operative system installed. I have built instances up to one million of arcs.

Results about the instances reported in figure 5.8 show that, GHP instances by using MMCFSCCplex can be solved very quickly. In this case, the decomposition algorithm using a bundle approach returns a good "speed up", only if the solver converges at the optimal solution in less then 5 iterations. The fact that there are only binary variables seems to advantage the MMCFSCCplex solver, this is interesting because it is not obvious to obtain quickly results by using instances of such dimension. Assuming we have solved the first stage instance and the optimal result returns that the airplanes land before the branching time, then the instances can be solved separately, stage for stage.

5.7 Conclusions

Scenario analysis applied to air traffic delay problems permits us to solve large dimension real-size instances in a short time. Both model and solver packages can be adopted as an operational tool to control both landing and departure procedures. I have implemented two computer codes included in a framework, permitting us to know when flights are delayed, where they are delayed and which flights are delayed. This framework can be adopted as a DSS to evaluate the maximum level of air traffic capacity supported by aerodrome utilizing its own resources, in a tactical approach. Decision makers can generate a set of different scenarios, based on their own experience. Quality about obtained results must be evaluated in a collaborative decision making environment, because other arrival/departure airports could have future capacity problems, as well. Whatever, such model can be effectively adopted in facing all those problems asking for assignment of a set of J jobs to a set of M machines, having a bounded work-capacity to a given temporal horizon t. A new interface among scenario tree problem and the solver Cplex has been implemented. This interface permits us to solve scenario tree instances with multicommodity network topology. A new framework has been implemented handling scenario tree problems. The framework permits us to:

* decide the set of scenarios involved in the optimization process;
* extend the set of available solvers rewriting the associated basis classes;
* build a readable scenario tree instance starting from deterministic instances in pprn format.
* read the optimal value of the decisional variables;
* change the topology of the network;

This work can be extended following different paths. It should be interesting to:

* interface the package with new non-smooth convex optimization solvers. For example, interior point based multicommodity solvers;
* test the model with instances coming from real world data;
* extend the model with a network of arrival airports and the constraints concerning the runway capacity in the departure airports;
* permit to airplanes taking off before the scheduled time;
* evaluate the potential benefits in terms of money saving in air traffic costs returned by scenario analysis model;
* evaluate instances with hundreds of scenarios in a parallel environment.

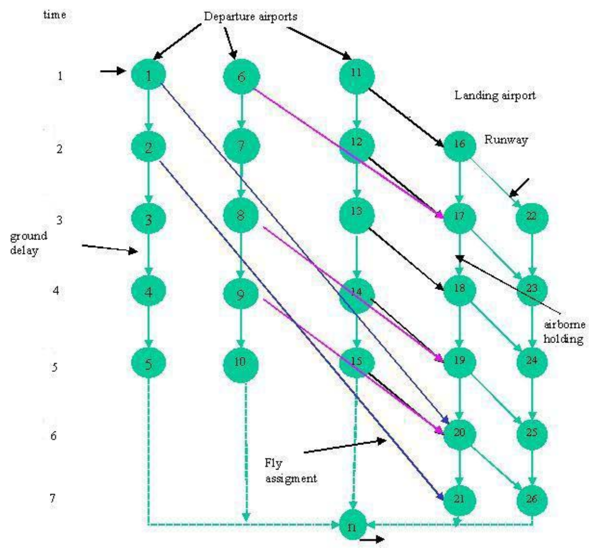

Figure 5.3: Direct temporal network for air traffic delay management

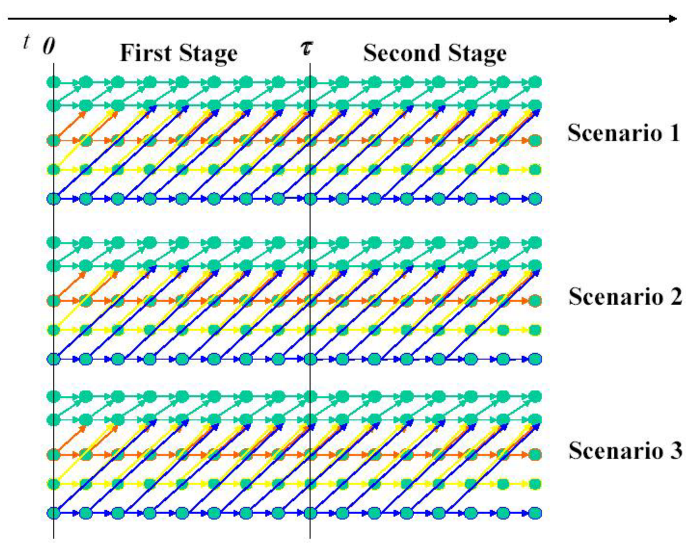

Figure 5.4: A bundle of scenarios generated for the GHP

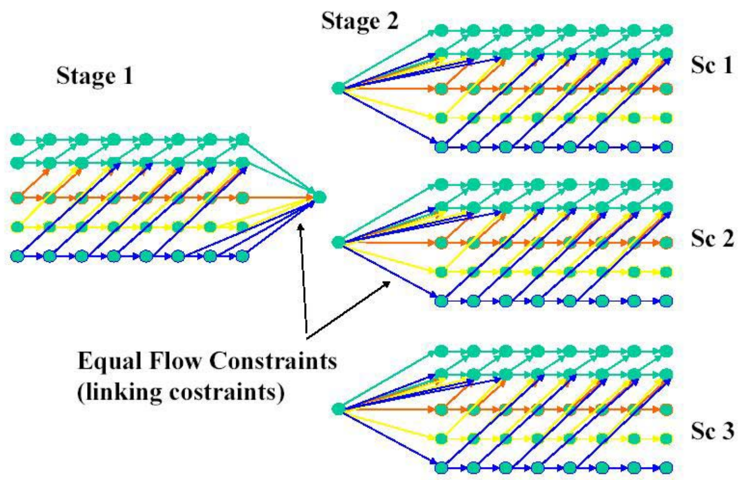

Figure 5.5: Scenario tree resulting after the grouping process

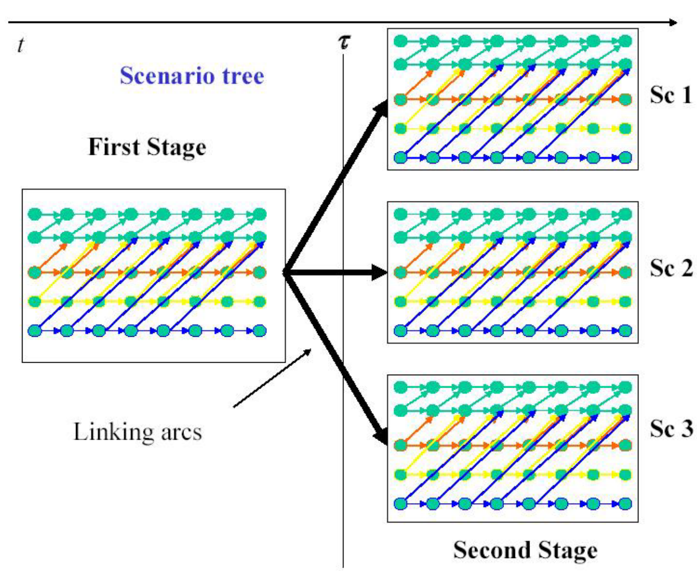

Figure 5.6: Scenario tree decomposed in independent instances

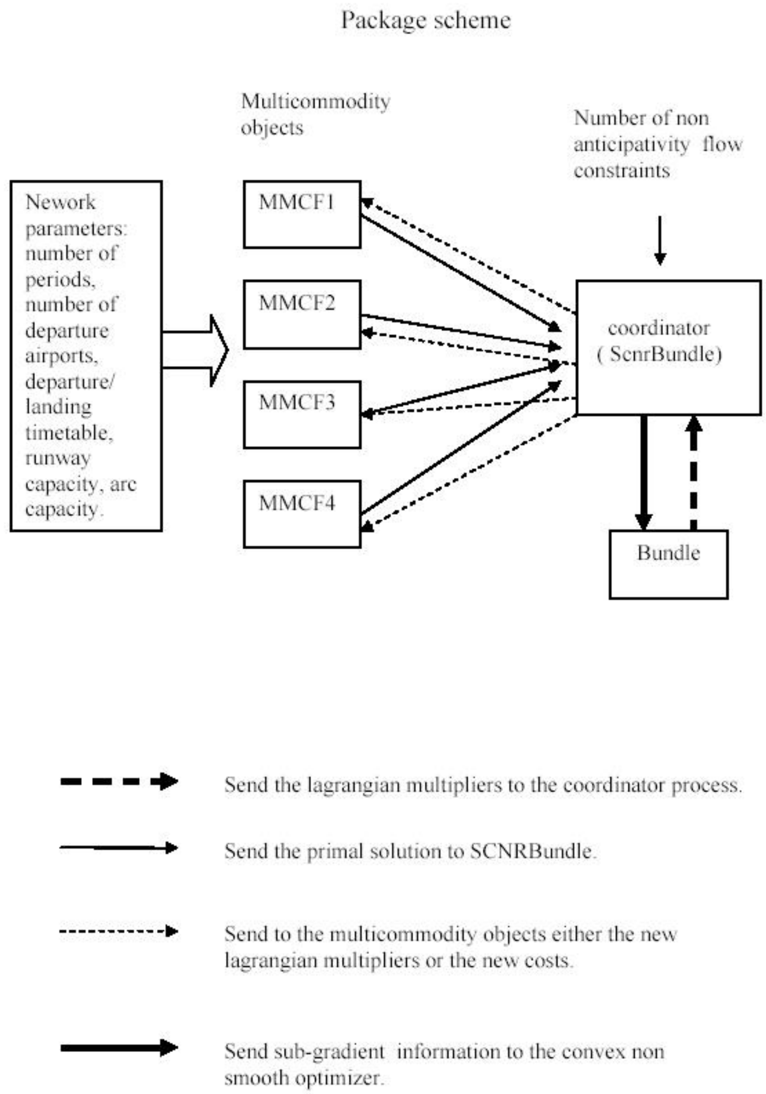

Figure 5.7: A scheme of the decomposition algorithm

File name	System time Scenario_tree Cplex (in seconds)	System time Decomposition Base approach (in seconds)	Number of periods	Number of scenarios	Number of departure airports	Number of airplanes
Net_1	36.08	14.66 (2)	80	10	10	22
Net_2	0.31	2.74 (21)	20	3	10	10
Net_3	0.31	0.67 (5)	20	3	10	10
Net_4	15.9	654 (125)	80	10	10	22
Net_5	171	199 (5)	80	10	10	60
Net_6	15.9	770 (116)	40	10	11	70
Net_7	19.9	66.91 (9)	40	6	10	60
Net_8	15.28	145 (23)	40	5	10	60
Net_9	39.66	360 (27)	40	10	10	60
Net_10	6.96	14.54 (5)	32	7	10	30
Net_11	20.97	222 (30)	32	10	10	30
Net_12	4.07	9.65 (7)	40	5	10	20
Net_13	39.09	14.86 (2)	80	10	10	20

Figure 5.8: Results obtained for real size instances

Bibliography

[1] C. Andrade, A. Lisser, N. Maculan, G. Plateau, Simulation on the Integer Capacity Planning under Uncertain Demand Problem in Telecommunication Networks. *In EUROSIM 2002*, Delft - The Netherlands, pages in CD, (2002).

[2] A. Alonso-Ayuso, L.F. Escudero A. Garin, M.T.Ortuno, G. Perez, An approach for Strategic Supply Chain under Uncertainty based on Stochastic 0-1 Programming, *Journal of Global Optimization*, 26, 97-124, 2003.

[3] V.Albornoz, F. Benario, M. Rojas, A two Stage Stochastic Integer Programming Model for a Thermal Power System Expansion, *International Transaction in Operational Research*, 11, 243-257, 2004.

[4] G. Andreatta, G Romanin-Jacur, Aircraft Flow Management Under Congestion, *Tranportation Science*, 21, 249-253, 1987.

[5] R. K. Ahuja, T. L.Magnanti, J. B. Orlin, *Prentice Hall, Englewood Cliff, NJ*, Birches. J., 35, 123-126, 1993.

[6] R.C. Baker, Network Flows: theory, algorithms and applications, *Prentice Hall*, Englewood Cliff, NJ, 1993.

[7] J.F. Benders, Partitioning Procedures for Solve Mixed Variable Programming Problems. *Numerische Mathematik*, 4:238-252, 1962.

[8] A. Ben-Tal, A. Nemiroski, Robust Solutions of Linear Programming Problems Contaminated with Uncertainty Data, Programming, 88, 411-424, 2000.

[9] D. Bertsimas, M. Sim, The Price of Robustness. *Operations Research*, 52, 35-53, 2004.

[10] D. Bertsimas, M. Sim, *Robust Discrete Optimization and Netwotk Flow* . Operations Research 52, 35-53, 1998.

[11] D. Bertsimas and S. Stock Patterson, The Traffic Flow Management Rerouting Problem in Air Traffic Problem, a Dynamic Network Flow Approach, *Tranportation Science*,34, 249-253, 1987.

[12] D. Bertsimas, A. Odoni, P.B. Vranas, Dynamic Ground Holding Policies for a Network of Airports, *Tranportation Science*, 28, 4, 275-291, 1994.

[13] D. Bertsimas, A. Odoni, P.B. Vranas, The Multi-Airport Ground Holding Problem in Air Traffic Control, *Operations Research*, 42, 249-261, 1994.

[14] J.R. Birge, F.V. Louveaux, A Multicut Algorithm for Two-Stage Stochastic Linear Programs, *European Journal of Operational Research*, 34, 384-392, 1988.

[15] J.R. Birge, C.J. Donhoue , D.F. Holmes ,O.G. Swintsiski. A Parallel Implementation of the Nested Decomposition Algorithm for Multistage Stochastic Linear Problem,*Mathematical Programming*, 75:327-352, 1996.

[16] J.R. Birge and F. Louveaux, Introduction to Stochastic Programming,*Springer*, New York, 1997.

[17] J. R. Birge and C. H. Rosa, Modeling Investment Uncertainty in the Costs of Global CO2 Emission Policy,*European Journal of Operational Research*, 83(3):466-488, 1995.

[18] H. Bjrstad, D. Haugland, and R. Helming, A Stochastic Model for Gasoline Blending, *In M. Breton and G. Zaccour, editors, Advances in Operations Research in the Oil and Gas Industry*, 137-142, 1991.

[19] Jorgen Blomvall and Per Olov Lindberg, A Riccati-Based Primal Interior Point Solver for Multistage Stochastic Programming, *European Journal of Operational Research* , 143(2):452-461, 2002.

[20] C.C. Caroe, J. Tind, L-shaped Decomposition of Two Stage Stochastic Programs with Integer Recourse,University of Copenaghen, {it work paper, 1997.

[21] P. Carraresi, A. Frangioni, M. Nonato, Applying Bundle Methods to Optimization and Polyhedral Functions: An applications-Oriented Development,{it Ricerca Operativa, 27, 5-49, 1996.

[22] S. Cerisola, A.Ramos, Benders Decomposition for Mixed-Integer Hydrothermal Problems by Lagrangean Relaxation, *Proceedings 14th Power Systems Computation Conference (PSCC'02)*, ISBN 84-89673-25-X Session 05, paper 6, Pages 1-8, Seville, Spain, 2002.

[23] CPLEX Optimisation, Inc, Using the CPLEX Callable Library and CPLEX Mixed Integer Library, *CPLEX Optimisation, Inc*,Incline Village, Nevada, 2002.

[24] Consiglio, Cocco, Zenios, Scenario Optimization Asset and Liability Modelling for Individual Investors, University of Cyprus, *Working Paper*, 1993.

[25] T. Crainic, M. Gendreau, P. Dejax, Dynamic Stochastic Models for Allocation of Empty Containers, Dynamic Stochastic Models for the Allocation of Empty Containers, *Operations Research*, 41 102-126, 1993.

[26] T. Crainic, A.G. Lium, S. Wallace, Stochastic and Service Network Design, {it TRISTAN V: The Fifth Triennal Symposium on Trasportation Analisys, Guadalupe, 2004.

[27] G. Dantzig, Linear Programming Under Uncertainty,*Management Science*, 1,197-206, 1955.

[28] G. B. Dantzig and G. Infanger, Intelligent Control and Optimization under Uncertainty with Application to Hydro Power, *European Journal of Operational Research*, 97(2):396-407, 1997.

[29] R. De Leone, L. Romagnoli, Algoritmi di Otimizzazione non Differenziabile con Applicazioni alla programmazione intera, University of Camerino, 1999.

[30] R. Dembo, Scenario Optimisation, *Annals of Operations Research* ,30, 63-80, 1991.

[31] R.Dembo, A Robust Approach for Water Resources Planning under Uncertainty, *Annals of Operations Research*,95, 313-339, 1995.

[32] N. Di Domenica, George Birbilis, Gautam Mitra, Patrick Valente, Stochastic Programming and Scenario Generation within a Simulation Framework: an Information System Perspective, *Technical Report*, CARISMA, london, 2003.

[33] M. Di Francesco, A. Manca, A. Olivo, P. Zuddas. An operational model for empty container management, Maritime Economics Logistics, Palgrave Macmillan (ed), Vol. 7, 2005.

[34] Y. Ermoliev, A. Gaivoronski, Stochastic Quasi-Gradient Methods for Optimization of Discrete Event Systems, *Annals of Operations Research*, 39(1-4),1-39(1993),1992.

[35] L.F. Escudero, Robust Optimization as a Decision Making Aid Under Uncertainty, Universidad Complutense de Madrid, UITESA, Madrid, 1993.

[36] L. F. Escudero, C. Garcia, J. L. de la Fuente, and F. J. Prieto, Hydropower Generation Management under Uncertainty Via Scenario Analysis and Parallel Computation, *IEEE Transactions on Power Systems*, 11(2):683-689, 1996.

[37] Eurocontrol, Delay to Air Transport in Europe, *Annual Report*, 2004.

[38] A. Fournie, From Air Traffic Flow Management to Air Traffic Flow Capacity Management, Eurocontrol, Head of User Relations Bureau, 2004.

[39] A. Frangioni, Dual-Ascent Methods and Multicommodity Flow Problems,*PhD Thesis*, University of Pisa, 1997.

[40] G.D. Glockner, MIDCORT: Minimizing Delay Cost in Real Time, *Technical Report*, Federal Aviation Administration, AOR-200, Washington, 1991.

[41] G.D. Glockner, Dynamic Network Flow with Uncertain Arc Capacities, *PhD thesis*, Georgia Institute of Technology, Atlanta GA, 1997.

[42] G.D. Glockner, Effects of Air Traffic Congestion Delays Under Several Flow Management Policies,*Transportation Research Record*, 1517, 29-36, 1996.

[43] G.D. Glockner and G.L. Nemhauser, Dynamic Network Flow with Uncertain Arc Capacities: Formulation and Problem Structure,*Technical Report*, 96-08, Logistics Engineering Center, Georgia Institue of Technology, Atlanta GA.

[44] Glockner, G. D. and G.L. Nemhauser, A Dynamic Network Flow Problem with Uncertain Arc Capacity: Formulation and Problems Structure,*Operations Research*, 48(2), 233-242, 2000.

[45] J. Gondzio, O. Du Merle, Analitic Center Cutting Plane Method User Guide for the Library, *technical Report*, 1995.

[46] L.Grignon, Analysis of Delays in an Air Traffic System with Weather Uncertainty, , degree thesis, University of Washington, 2002.

[47] R. Henrion, Perturbation ananlysis of chance-constrained programs under variation of all constraint data, *Stochastic Programming E-Print Series*, http://www.speps.info, 2003.

[48] R. Henrion and A. Mller, Optimization of a continuous distillation process under random inflow rate, *Computers Mathematics with Applications*, 45:247-262, 2003.

[49] J. Higle, S.W. Wallace, Sensitivity Analysis and Uncertainty in Linear Programming, *Interfaces*, 33, 53-60, 2003.

[50] J. Higle, S. Wallace, Managing Risk in the New Power Business, *IEEE Computer Applications in Power*, 12th-19th April, 2002.

[51] Hoyland, K. and S. W. Wallace, Generating Scenario Trees for Multistage Decision Problems, *Management Science*, 47(2), 295-307, 2001.

[52] G. Infanger, Planning under Uncertainty: Solving Large-scale Stochastic Linear Programming, *Boyd and Fraser Publishing Company*, Danvers, MA, 1994.

[53] D. P. Loucks, J. R. Stedinger, D. A. Haith, Water resource systems planning and analysis, *Prentice Hall*,New Jork,1981.

[54] P. Kall, and S. W.Wallace, Stochastic Programming, *John Wiley and Sons*, New York, 1994.

[55] G.Kuczkera, Network Linear Programming Codes for Water-Supply Head-Works Modelling, *Journal of Water Resources Planning and Management* , 3, 412-417,1992.

[56] M. Kaur, S.W. Wallace, Evaluation of Scenario-generation Methods for Stochastic Programming, Molde University, work paper,2003.

[57] J. Linderoth, S. Wright, Decomposition Algorithms for Stochastic Programming on a Computatinal Grid, preprint Mathematics and Computer Science Division Argonne National Laboratory, 2001.

[58] A. Manca, A.Podda, G. M. Sechi, P. Zuddas, The network simplex equal flow algorithm in dynamic water resources management, *Proceedings of 6th Congress SIMAI*, Chia Laguna,2002.

[59] A. Manca, G.M. Sechi, P. Zuddas. Scenario Reoptimization under Data Uncertainty. Pahl-Wostl, C., Schmidt, S., Rizzoli, A.E. and Jakeman, A.J.(eds), Complexity and Integrated Resources Management. iEMSs Transactions, Vol. 1, 771-776, 2004.

[60] A. Manca, A. Frangioni. A Computational Study of Cost Reoptimization for Min Cost Flow Problems. INFORMS JOC (Journal On Computing), Vol. 18, No. 1, 2006.

[61] A. Manca, G.M. Sechi, A.Sulis, P. Zuddas. Complex Water Resources System Optimization Tool Aided By Graphical Interface. 6th international conference on hydroinformatic, Liong, Phoon Babovic (eds). World Scientific Publishing Company, 2004.

[62] J.M. Mulvey, A New Scenario Optimization Method for Large Scale Stochastic Optimization, *Operations Research*, 43:477-490, 1995.

[63] J. M. Mulvey, H. Vladimirou, Stochastic Network Optimisation Models for Investment Planning, *Annals of Operations Research* , 20, 187-217,1989.

[64] L. Navazio, G Romanin-Jacur, The Multiple Connections Multi-Airport Ground Holding Problem: Models and Algorithms,*Transportation Science*, 32, 3, 268-276, 1998.

[65] A. Nilim, L. El Ghaoui V. Duong, Algorithms for Multi Aircraft Routing Under Uncertainty, *International Conference RIVF*, Hanoi, 2004.

[66] A. Odoni, The Flow Management Problem in Air Traffic Problem, in *Flow Control of Congested Network*, A. Odoni, L. Bianco, G. Szego. Eds. Springer-Verlag, 269-288, 1987.

[67] L.Onnis, G. M. Sechi, P. Zuddas, Optimisation Processes under Uncertainty, *A.I.C.E.*, Milano, 238-244,1999.

[68] C.G. Panayiotou, C.G. Cassandras, A Sample Path Approach for Solving the Ground Holding Policy Problem in Air Traffic Control. IEEE transaction on Control System Technology 9, 3, 510-523, 2001.

[69] J.W. Pepper, K.R. Mills, L.A. Wojcick, Predictability and Uncertainty in Air Traffic Flow Management, *5th USA/Europe Air Traffic Management RD Seminar*, Budapest, 23-27 June 2003.

[70] RAS-DIT-EAF, *Convention RAS-DIT-EAF, Regional Report, Piano dAmbito della Regione Sardegna*, Cagliari,2003.

[71] O. Richetta, Ground Holding Strategies for Air Traffic Control Under Uncertainty, Techincal Report 198, M.I.T., *Operations Reasearch Center*, 1991.

[72] O. Richetta,A. Odoni, Solving Optimally the Static Ground Holding Policy Problem in Air Traffic Control, *Transportation Science*, 27:3, 228-238, 1993.

[73] O. Richetta,A. Odoni, Dynamic Solution to the Ground Holding in Air Traffic Control, *Transportation Science*, 28:4, 275-291, 1994.

[74] R.T. Rockafellar and R.J-B Wets, A Dual Strategy for the Implementation of the Aggregation Principle in Decision Making under Uncertaint, *Applied Stochastic Models and Data Analysis*, 8:245-255, 1992.

[75] R.T. Rockafellar and R.J-B Wets, Stochastic convex programming: Kuhn-tucker conditions, *J. of Mathematical Economics*, 2:349-370, 1975.

[76] R.T. Rockafellar and R.J-B Wets, Nonanticipativity and stochastic optimization problems, *Mathematical Programming Study*, 6:170-187, 1976.

[77] R.T. Rockafellar and R.J-B Wets, Scenarios and policy aggregation in optimization under uncertainty, *Mathematics of Operations Research*, 16:119-147, 1991.

[78] R.T. Rockafellar, Convex Analysis, *Princeton University Press*, Princeton 1970.

[79] A. Ruszczynski, Parallel Decomposition of Multistage Stochastic Programming Problems, *Mathematical Programming*, 58:201-228, 1993.

[80] A. Ruszczynski, A Regularized Decomposition for Minimizzing a Summ of Polyhedral functions, *Mathematical Programming*, 35:309-333, 1986.

[81] A.Ruszczynski, Decomposition Methods in Stochastic Programming, *Mathematical Programming*, 79, 333-353,1997.

[82] A.Ruszczynsky, A. Swietanoski, On the Regularized Decomposition Method for two stage Stochastic linear problem, *WP-96-014*, 1996.

[83] G. M. Sechi, P. Zuddas, Structure Oriented Approaches for Water System Optimisation, *Conference on Coping with Water Scarcity*, Hurgada, Egypt,1998.

[84] G. M.Sechi, P. Zuddas, WARGI: Water Resources System Optimisation Aided by Graphical Interface In: Blain, W. R., Brebbia, C. A., *C. A., (Eds.), Hydraulic Engineering Software, WIT-PRESS, 109-120*, 2000.

[85] G. M. Sechi, P. Zuddas, *INCO-MED EU Project, Water Resources Management under Drought Conditions (Wam-me), Work Package 5.*, 2000.

[86] M. Terrab and A. Odoni, Strategic Flow Management for Air Traffic Control, Operations Research, 41, 138-152, 1997.

[87] Van Slyke J.B. Wets, L-shaped Linear Programs with application to Optimal Control and Stochastic Programming, *SIAM Journal of Applied Mathematics*, 17, 638-663.

[88] L.A. Wolsey, A Resource Decomposition Algorithm for general Mathematical programs, *Mathematical Programming Study*, 14, 244-257, 1981.

[89] P. Kall and S. Wallace, *Stochastic Programming.* Wiley, New York, 1994. Online available.

[90] A. Prkopa and T. Szntai, Flood control reservoir system design using stochastic programming, *Mathematical Programming Study*, 9:138-151, 1978.

[91] A. Ruszczynski and A. Shapiro, Stochastic Programming, *Handbooks in Operations Research and Management Science*, Vol. 10. Elsevier, Amsterdam, 2003.

[92] S. Sem, R.D. Doverspike and S. Cosares, Network Planning with Random Demand, *Telecommunications Systems*, 3:11-30, 1993.

[93] R.J-B Wets, Stochastic programs with fixed recourse: the equivalent deterministic problem, *SIAM Review*, 16:309-339, 1974.

[94] R.J-B Wets, Stochastic Programming, In G. Nemhauser, A. Rinnooy Kan, and M. Todd, editors, *Handbook for Operations Research and Management Sciences*, Vol 1, pages 573-629. Elsevier Science Publishers B.V. (North Holland), 1989.

[95] R.J-B Wets, Stochastic programs with Fixed Recourse: the Equivalent Deterministic Program, *SIAM Review*, 16,309-339, 1974.

[96] R.J-B Wets, On the relation between stochastic and deterministic optimization. In A. Bensoussan and J.L. Lions, editors, *Control Theory, Numerical Methods and Computer Systems Modelling, Lecture Notes in Economics and Mathematical Systems*, 107, pages 350-361. Springer, 1975.

[97] R.J-B. Wets, The aggregation principle in scenario analysis and stochastic opti-mization, *In S. Wallace, editor, Algorithms and Model Formulations in Mathemat- ical Programming, NATO ASI Vol.51*, 91-113. *Springer*, NATO ASI Vol.51, 1989.

[98] R.J-B. Wets, Stochastic programming models: Wait-and-see versus here-and-now, In Francois Auzerais, R. Burridge, C. Greengard, and A. Ruszczynski, editors, Making under Uncertainty: Energy and Environmental Models, Springer, 2001.

[99] R.J-B Wets, Large-scale linear programming techniques in stochastic program- ming, In Y. Ermoliev and R. Wets, editors, *Numerical Techniques for Stochastic Optimization*, 61-89, Springer, 1988.

[100] W.G. Yeh, Reservoir Management and Operations Models: a State-of-the-art Review, *Water Resources Research*, 25(12), 1795-1818, 1985.