Network Planning Under Uncertainty with an Application to Hydropower Generation

M. Alvarez

*Dept. of Applied Mathematics, Industrial Engineering School,
Polytechnical U., Madrid*

C.M. Cuevas

Iberdrola, Spain

L.F. Escudero

*UITESA and Dept. of Statistics and Operations Research,
Univ. Complutense, Madrid*

J.L. de la Fuente

*Iberdrola and Dept. of Applied Mathematics,
Industrial Engineering School, Polytechnical U., Madrid*

C. García

Iberdrola, Spain

F.J. Prieto

Dept. of Statistics and Econometrics, Univ. Carlos III, Madrid

SUMMARY

We present a general modeling framework for the *robust optimization* of linear network problems with uncertainty in the values of the right-hand side. In contrast to traditional approaches in mathematical programming, we use scenarios to characterize the uncertainty. Solutions are obtained for each scenario and these individual scenarios are aggregated to yield a nonanticipative or implementable policy that minimizes the regret of wrong decisions. A given solution is termed robust if it minimizes the sum over the scenarios of the weighted upper difference between the objective function value for the solution and the objective function value for the

optimal solution for each scenario, while satisfying certain nonanticipativity constraints. This approach results in a huge model with a network submodel per scenario plus coupling constraints.

Several decomposition approaches are considered, namely Dantzig-Wolfe decomposition, various types of Benders decomposition and different quadratic network approaches for approximating Augmented Lagrangian decomposition. We present computational results for these methods, including two implementation versions of the Lagrangian based method: a sequential implementation and a parallel implementation on a network of three workstations.

1. DETERMINISTIC NETWORKS

Consider the following deterministic *network problem* (LN):

$$\begin{aligned} z = \quad & \min \quad c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{1.1}$$

where c is an n -column vector, A is an $m \times n$ node-arc incidence matrix for some directed graph $G = (V, E)$, where V denotes the set of nodes with $m \equiv |V|$, E denotes the set of arcs with $n \equiv |E|$, b is an m -column vector and x is the n -column vector of the nonnegative unknowns.

The system $Ax = b$ takes the form

$$\sum_{j \in \delta_i^+} x_{ij} - \sum_{j \in \delta_i^-} m_{ji} x_{ji} = b_i \quad \text{for } i \in V, \tag{1.2}$$

where x_{ij} is the flow over the arc (i, j) , m_{ji} is the multiplier on arc (j, i) , b_i is the supply or demand for node i , $\delta_i^+ \subset V$ is the set of head nodes for the arcs leaving node i and $\delta_i^- \subset V$ is the set of tail nodes for the arcs entering node i .

If all the multipliers m_{ji} are equal to one, we refer to model LN as a *pure network* problem; otherwise, it is termed a *generalized network* problem. This feature is important in modeling situations with gains or losses on the flow through the arcs (e.g., in energy transmission) or whenever a change of units takes place in the flow between two nodes (e.g., in currency arbitrage, cash flow management). There are

many instances of problems that can be represented as networks, some of which are water/gas distribution, electricity, input-output tables in economics, communication, road networks, etc. See [Dembo (1991)], [Escudero (1986)], [Escudero (1990)], [Glover et al. (1992)], [Sartenaer (1991)], [Toint and Tuytens (1990)] among many others.

Let us assume that we can partition the node set V into so-called stages (the term is taken from multiperiod applications). Thus Q_r will denote the set of nodes that belong to stage r , such that $Q_r \cap Q_{r'} = \emptyset$ for any pair of stages $r \neq r'$. For illustrative purposes, assume that there are three stages; then, $V = Q_1 \cup Q_2 \cup Q_3$. Similarly, let us partition the vector b such that b_1 , b_2 and b_3 are the subvectors of b for the node sets Q_1 , Q_2 and Q_3 , respectively. Obviously, there will be coupling variables between any two stages; let $Q_r \neq \emptyset$, $r = 1, 2, 3$, and as another simplification, assume that the coupling variables appear only between stages 1 and 2 and between stages 2 and 3. We will also use the following notation:

R_r , set of arcs whose head and tail nodes belong to set Q_r , $R_r \equiv \{(i, j) \in E : i, j \in Q_r\}$.

R_{12} , set of coupling arcs between stages 1 and 2, $R_{12} \equiv \{(i, j) \in E : i \in Q_1, j \in Q_2\}$.

R_{23} , set of coupling arcs between stages 2 and 3, $R_{23} \equiv \{(i, j) \in E : i \in Q_2, j \in Q_3\}$.

Similarly, let the matrix A be partitioned accordingly, so that A_1 and A_{12}^1 are the submatrices corresponding to the node set Q_1 and the arc sets R_1 and R_{12} , respectively; A_{12}^1 , A_2 and A_{23}^2 are the submatrices corresponding to the node set Q_2 and the arc sets R_{12} , R_2 and R_{23} , respectively; and A_{23}^3 and A_3 are the submatrices corresponding to the node set Q_3 and the arc sets R_{23} and R_3 , respectively. We also assume that the x -variables have been partitioned corresponding to the arc sets R_1 , R_{12} , R_2 , R_{23} and R_3 .

Then, the system $Ax = b$ can be written:

$$\begin{aligned} A_1 x_1 + A_{12}^1 x_{12} &= b_1 \\ A_{12}^2 x_{12} + A_2 x_2 + A_{23}^2 x_{23} &= b_2 \\ A_{23}^3 x_{23} + A_3 x_3 &= b_3 \end{aligned} \quad (1.3)$$

The goal of this work is to present a motivation for using decomposition techniques to solve problem LN under uncertainty, with special

attention to hydropower generation management applications (see Section 2). Section 3 outlines the type of uncertainty to be dealt with. Section 4 presents a model for scenario tracking and anticipating variables. Section 5 gives our implementable policies and presents mathematically equivalent models to deal with the uncertainty under the nonanticipativity principle. Section 6 describes several approaches for the dualization of the coupling constraints (i.e., an Augmented Lagrangian approach). Section 7 presents our Benders decomposition approaches. Section 8 presents our Wolfe-Dantzig decomposition approach. Section 9 presents implementation results obtained for a set of test problems using the different decomposition schemes. Two implementation versions are tested for the Lagrangian based method: a sequential implementation and a parallel implementation on a network of three workstations.

2. PROBLEM DESCRIPTION: HYDROPOWER GENERATION MANAGEMENT

The problem we will use as a reference in the rest of the paper is that of maximizing the hydropower generated along a time horizon (usually, one year) by a multireservoir power system.

In this case, the basic network is constructed from a river where the nodes represent the reservoirs and the arcs correspond to the sections of the river that connect the reservoirs. This basic network is then replicated on a number of segments, corresponding to the time periods (usually, weeks) included in the planning horizon; these segments are linked by special arcs on some of the nodes. The decision variables x are the amount of water to be released from each reservoir to its direct downstream reservoirs in a given period (corresponding to the arcs in the basic network), or the amount of water to be stored in the reservoirs from one period to the next, that becomes available to be released or stored in the next period (the special arcs linking the segments).

The constraints are of two kinds: linear equations that ensure flow balance in each reservoir, and simple bounds on the variables. These simple bounds have the following purposes: first, to ensure that the water released serves the desired flood control, irrigation and navigational purposes; second, to ensure that the amount of water released from a given reservoir to any of its directly downstream reservoirs does not exceed the canal capacity; and third, to penalize the amount of stored water that exceeds a safety capacity for any given reservoir. These constraints have the form indicated above for a network problem.

The objective function consists of the maximization of a nondifferentiable nonlinear function, typically the generation of electricity over the planning horizon.

2.1. Problem formulation

We now introduce a formal model for the problem described above. Let J denote the set of reservoirs, T be the time periods set, $W \subset J$ correspond to the water storage reservoirs (so that J/W are the run-of-the-river plants), $E \subset J$ the set of reservoirs that are not used for hydropower generation, and P_j (Q_j) the set of reservoirs directly upstream (downstream) from reservoir j . The decision variables will be denoted by r_{tji} , the amount of water released from reservoir j to reservoir $i \in Q_j$ in period t , and s_{tj} , the amount of water stored in reservoir j at the beginning of period t .

The flow balance equations in each reservoir for each time period are as follows

$$-\sum_{i \in P_j} r_{tji} - s_{tj} + \sum_{i \in Q_j} r_{tji} + s_{t+1,j} = b_{tj} \quad \forall t \in T, j \in J, \quad (2.1)$$

where b_{tj} is the exogenous inflow to reservoir j in time period t , that could be negative if we have an outflow (evaporation loss, for example) during period t ; s_{tj} is fixed for $t = 1$ and $t = |T| + 1$. If we denote by x the vector of all decision variables, and b is the vector of exogenous inflows, system (2.1) can be written

$$Ax = b,$$

where A is the node-arc incidence matrix.

The bounds on the arcs of the replicated network are given by

$$l_{tji} \leq r_{tji} \leq u_{tji} \quad \forall i \in Q_j, j \in J, t \in T \quad (2.2)$$

$$m_{tj} \leq s_{tj} \leq M_{tj} \quad \forall j \in W, t \in T/\{1\}. \quad (2.3)$$

The important property of the hydropower generation function is its spatial and temporal quasi-separability. The hydropower generated in reservoir j at time period t can be expressed as follows

$$h_{tj} = K_{tj} \sum_{i \in Q_j} c_{tji} r_{tji} \quad \forall t \in T, j \in J/E, \quad (2.4)$$

where K_{tj} may be a constant or a nonlinear function f_{tj} of the s -variables for the reservoirs in set J/E . Let j be termed a linear reservoir if K_{tj} is a constant, otherwise, it is termed a nonlinear reservoir, and K_{tj} can be written as

$$K_{tj} = f_{tj}(s_{tj}, s_{t+1,j}). \quad (2.5)$$

Without penalizing any amount of water stored or released, the objective function can be written

$$\max \sum_{t \in T} \sum_{j \in J/E} h_{tj}. \quad (2.6)$$

Let R_{ji} denote the power generation capacity of canal (j, i) for $j \in J/E$, where $l_{ji} \leq R_{ji} \leq u_{ji}$, and let T_{tj} be an upper bound on the amount of water that can be safely stored in reservoir j for $j \in W$ at time period t . Since the water overflow $r_{tji} - R_{ji}$ cannot be used for hydropower generation, and we would be interested in penalizing the excess water stored $s_{tj} - T_{tj}$, the objective function can be modified to

$$\max \left(\sum_{t \in T} \sum_{j \in J/E} h_{tj} - \sum_{t \in T} \sum_{j \in W} P_{tj} \max(0, s_{tj} - T_{tj}) \right), \quad (2.7)$$

where

$$h_{tj} = K_{tj} \sum_{i \in Q_j} \min(r_{tji}, R_{ji}),$$

and P_{tj} is either a unit penalty or a nonlinear penalty function of the excess water stored.

The objective function (2.7) is temporally quasi-separable and spatially separable. Also, the nondifferentiability introduced in this function can be treated without great difficulty, see [Escudero (1990)].

Since the focus of this paper is on different methodologies to deal with the uncertainty on the right-hand side of (1.1), we will simplify problem (2.1)-(2.7), and assume a linear objective function.

3. UNCERTAINTY

In many situations we have a problem that fits the linear network model introduced in Section 1, but where some elements of the matrix A and vector b may not be known with certainty. Typical examples are exchange rates for currency arbitrage for non-deterministic multipliers m_{ji} [Mulvey and Vladimirou (1989)], product demand b for production planning [Escudero et al. (1993)], or exogenous water inflow (i.e., raining) for the hydroelectric power generation problem introduced in Section 2. See also [Dembo (1987)], [Escudero (1986)] and [Toint and Tuytens (1990)], among others.

In all these cases, one needs to consider two additional features. In the first place, one must model the availability of information over time and state what sort of decisions can be made at each of the various stages. Secondly, to compute an optimizer any proposed solution will have to be compared with other candidate solutions. But in the stochastic setting, the criteria by which this comparison can be performed are much less clear. Thus, we need an approach to model the uncertainty in the problem data. The traditional approach is to make distributional assumptions, estimate the parameters from historical data and then develop a stochastic model to take the uncertainty into account. Such approach may not be appropriate if only limited information is available. On the other hand, in many applications it is often necessary and possible to take into account information that is not reflected in the historical data. In many such cases we may employ a technique called “scenario analysis”, where the uncertainty is modelled via a set of scenarios, say S .

For example, in our network model the right-hand side vector may take different values for different scenarios, b^s for $s \in S$; in the hydropower generation management problem these values would correspond to different exogenous water inflows (rain) and different values of the demand for electricity and water. We also introduce weights w^s representing the likelihood that the decision maker (modeller) associates with each scenario $s \in S$.

4. SCENARIO IMMUNIZATION

Formally, let us denote by z_*^s the optimal solution value for the model (1.1) under scenario s for $s \in S$, given as

$$\begin{aligned} z_*^s = \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b^s \\ & x \geq 0. \end{aligned} \tag{4.1}$$

One way to deal with the uncertainty is to obtain the solution x that best tracks each of the scenarios—see [Dembo (1991)]. We can perform this by obtaining a solution that minimizes the weighted upper difference between the proposed solution value and the optimal solution value for each scenario. In order to model the scenario tracking, let r_+ denote the positive part of r , such that

$$r_+ \equiv \max(0, r). \tag{4.2}$$

Our scenario immunization (SI) model is as follows

$$\begin{aligned} \min \quad & \sum_{s \in S} w^s (c^T x - \hat{z}_*^s)_+^d + \alpha H(x_a) \\ \text{s.t.} \quad & Ax + x_a^s = b^s \quad \forall s \in S \\ & x \geq 0 \end{aligned} \tag{4.3}$$

where $\hat{z}_*^s$ denotes the optimality tolerance—see [Jensen and King (1991)],

$$\hat{z}_*^s = z_*^s + \epsilon \quad \text{for } \epsilon \geq 0, \tag{4.4}$$

d is a positive parameter, α is a nonnegative weighting factor, $H(x_a)$ is a function of the infeasibility at x_a , and x_a^s is an m -column vector that compensates for the infeasibility of the solution x under scenario s . The motivation for using $(c^T x - \hat{z}_*^s)_+$ instead of $(c^T x - \hat{z}_*^s)$ is that a better solution value than the optimal one should not be penalized—note that the related solution infeasibility is already penalized.

Several possible forms for $H(x_a)$ are—see [Mulvey et al. (1991)]:

$$\begin{aligned} H(x_a) : & \max_{s \in S} \{w^s \xi^s\} \\ & : \sum_{s \in S} w^s (\xi^s)^2, \end{aligned} \quad (4.5)$$

where

$$\xi^s = \|x_a^s\|. \quad (4.6)$$

Model SI (4.3) does not increase the number of variables (as the variables x_a^s do not need to be taken explicitly into account) but now there are $m|S|$ constraints. Unfortunately, this representation of the model does not have the nice properties of network models (i.e., two nonzero constraint entries for each variable, at most, and one of them is +1). The next section analyzes some procedures to overcome this difficulty.

Depending upon the value of the parameter d , the function $\sum(\cdot)_+^d$ can be nonlinear. However, given an appropriate value for d , we can approximate the solution of SI through the solutions to linear problems. Specifically, a hierarchical 2-step approach can be used: the first step minimizes the ℓ_∞ -norm, and the second step minimizes the ℓ_1 -norm subject to the ℓ_∞ -norm value that has been obtained from the first step. The details are as follows:

1. Minimize the so-called maximum regret,

$$\begin{aligned} \min \quad & \max_{s \in S} \left(w^s (c^T x - \hat{z}_*^s)_+ \right) + \alpha \max_{s \in S} w^s \xi^s \\ \text{s.t.} \quad & Ax + x_a^s = b^s \quad \forall s \in S \\ & x \geq 0, \end{aligned} \quad (4.7)$$

where $\xi^s = \|x_a^s\|_1$, and $\|\cdot\|_1$ denotes the ℓ_1 -norm. Define $\bar{\xi}$ and $\bar{v}$ from the solution of (4.7) as

$$\bar{\xi} \equiv \max_{s \in S} w^s \xi^s, \quad \bar{v} \equiv \max_{s \in S} w^s (c^T x - \hat{z}_*^s)_+.$$

2. Minimize the total weighted regret within the bounds $\bar{\xi}$ and $\bar{v}$,

$$\begin{aligned}
\min \quad & \sum_{s \in S} \left(w^s (c^T x - \hat{z}_*^s)_+ + \alpha \xi^s \right) \\
\text{s.t.} \quad & w^s (c^T x - \hat{z}_*^s) \leq \bar{v} \quad \forall s \in S \\
& w^s \xi^s \leq \bar{\xi} \quad \forall s \in S \\
& Ax + x_a^s = b^s \quad \forall s \in S \\
& x \geq 0.
\end{aligned} \tag{4.8}$$

As an alternative to model (4.3) we may also use

$$\begin{aligned}
\min \quad & \sum_{s \in S} w^s (c^T x - z_*^s)^d + \alpha \sum_{s \in S} w^s \left(\|x_a^s\| - \sum_{s' \in S} w^{s'} \|x_a^{s'}\| \right)^2 \\
\text{s.t.} \quad & Ax + x_a^s = b^s \quad \forall s \in S \\
& x \geq 0,
\end{aligned} \tag{4.9}$$

5. IMPLEMENTABLE POLICIES

Model SI (4.3) and related formulations do anticipate decisions in x that for multistage applications may not be needed at stage $r = 1$. Frequently, the decisions for stage $r = 1$ are the only decisions to be made, since at stage $r = 2$ one may realize that some of the data has been changed, some scenarios vanish, etc. In this case, the models will usually be reoptimized in a rolling planning horizon mode. When only spot decisions (i.e., decisions for stage $r = 1$) are to be made, the information about future uncertainty is only taken into account for a better spot decision making. This type of approach is termed *full recourse*. Model SI has been designed with a *simple recourse* philosophy in mind. An intermediate situation (i.e., the decisions are not to be changed for the first $\bar{r}$ stages, for $\bar{r} > 1$) is termed *partial recourse*.

Let x^s denote the arc flows under scenario s , for $s \in S$. The following implementable or nonanticipativity principle has been stated in [Rockafellar and Wets (1991)].

“If two different scenarios s and s' are identical up to stage r on the basis of the information available about them at stage r , then the values for the x -variables must be identical up to stage r ”.

This condition guarantees that the solution obtained by the model is not dependent at stage r on information that is not yet available. It is best visualized as follows: each node in figure 1 represents a point in time where a decision can be made; once a decision is made, one of several

contingencies can happen (in this example, there are three contingencies are three), and this information is available at the beginning of the next stage. Each root-to-leaf path in figure 1 represents one scenario and it is one realization of the right-hand-side b . In this case we have three stages. The elements of the subvector b_1 have only one value each, but the elements of b_2 and b_3 have three potential outlooks each for each realization of the uncertainty in the preceding period, and the outlooks in b_3 are assumed to be independent from the outlooks in b_2 . Note that in this case there are $|S| = 9$ scenarios.

Let N denote the set of solutions that satisfy the so-called *implementability* or *nonanticipativity* constraints. That is,

$$x \in N \equiv \{x^s \mid x^s = x^{s'} \text{ for all } s \text{ and } s' \text{ that are identical up to stage } r\}. \quad (5.1)$$

For illustrative purposes, condition (5.1) has the following expression for the case with 3 stages and 9 scenarios described above (see also figure 2).

$$\begin{aligned} x_e^1 &= x_e^2 = \dots = x_e^9 & e \in R_1 \cup R_{12} \\ x_e^1 &= x_e^2 = x_e^3 & e \in R_2 \cup R_{23} \\ x_e^4 &= x_e^5 = x_e^6 & e \in R_2 \cup R_{23} \\ x_e^7 &= x_e^8 = x_e^9 & e \in R_2 \cup R_{23} \end{aligned} \quad (5.2)$$

Our *full recourse* (FR) model with scenario immunization is as follows

$$\begin{aligned} \min \quad & \sum_{s \in S} w^s (c^T x^s - z_*^s)^d \\ \text{s.t.} \quad & Ax^s = b^s \quad \forall s \in S \\ & x \in N \\ & x^s \geq 0 \quad \forall s \in S. \end{aligned} \quad (5.3)$$

Note that in (5.3) we do not need to penalize the solution infeasibility. Both models SI and FR have the same number of constraints, apart from the nonanticipativity constraints (5.1), but FR has $n|S|$ variables instead of n . Note also that $c^T x^s \geq z_*^s$ for feasible x^s .

Model FR (5.3) has a nice structure that we may exploit. Two approaches can be used to represent the nonanticipativity constraints (5.1)—see [Escudero et al. (1993)]. Approach #1 is based on a *compact* representation, where the equalities (5.2) are used for eliminating

some variables. This approach reduces the model size, but the network structure of the constraints is destroyed. Approach #2 is based on a *splitting variable* representation where condition (5.2) is part of the model. Dualizing (5.2) results in independent network submodels.

For illustrative purposes consider the case with 3 stages and 9 scenarios defined above. The following additional notation is required for the x -variables.

x -variables in (1.3)	x -variables in (5.6)
x_1	x_1
x_{12}	x_{12}^0, x_{12}^k
x_2	x_2^k
x_{23}	x_{23}^{k0}, x_{23}^{kl}
x_3	x_3^{kl}

where $k, l = 1, 2, 3$.

As an additional simplification, let $w^s(c^T x^s - z_*^s)^d$ be defined as the linear function $(h^s)^T x^s$ corresponding to any of the steps in the 2-step hierarchical approach presented in this section, where h^s is the appropriate column vector of objective function coefficients for scenario s , such that

$$h^s \equiv (h_1, h_{12}^0, h_2^k, h_{23}^{k0}, h_3^{kl}), \quad (5.4)$$

for $s = 3(k - 1) + l$ and $k, l = 1, 2, 3$.

Finally, for the sake of simplicity we will omit in what follows the infeasibility term $\alpha w^s \xi^s$ from (4.7) and (4.8).

The compact representation of model FR (5.3) is as follows

$$\min h_1^T x_1 + (h_{12}^0)^T x_{12}^0 + \sum_k (h_2^k)^T x_2^k + \sum_k (h_{23}^{k0})^T x_{23}^{k0} + \sum_{k,l} (h_3^{kl})^T x_3^{kl}$$

$$\begin{aligned}
\text{s.t. } A_1 x_1 + A_{12}^1 x_{12}^0 &= b_1 \\
A_{12}^2 x_{12}^0 + A_2 x_2^k + A_{23}^2 x_{23}^{k0} &= b_2^k \quad \forall k \\
A_{23}^3 x_{23}^{k0} + A_3 x_3^{kl} &= b_3^{kl} \quad \forall k, l \\
x_1, \quad x_{12}^0, \quad x_2^k, \quad x_{23}^{k0}, \quad x_3^{kl} &\geq 0.
\end{aligned} \tag{5.5}$$

The computational procedures described in [Escudero et al. (1993)] (a mixture of a crash approach and a Simplex implementation) take advantage of this structure. One of our two Benders decomposition approaches also takes advantage of the structure.

The splitting variable representation of model FR (5.3) is as follows

$$\begin{aligned}
\min \quad & h_1^T x_1 + (h_{12}^0)^T x_{12}^0 + \sum_k (h_2^k)^T x_2^k + \sum_k (h_{23}^{k0})^T x_{23}^{k0} + \sum_{k,l} (h_3^{kl})^T x_3^{kl} \\
\text{s.t. } \quad & A_1 x_1 + A_{12}^1 x_{12}^0 = b_1 \\
& x_{12}^0 - x_{12}^k = 0 \quad \forall k \\
& A_{12}^2 x_{12}^k + A_2 x_2^k + A_{23}^2 x_{23}^{k0} = b_2^k \quad \forall k \\
& x_{23}^{k0} - x_{23}^{kl} = 0 \quad \forall k, l \\
& A_{23}^3 x_{23}^{kl} + A_3 x_3^{kl} = b_3^{kl} \quad \forall k, l \\
& x_1, x_{12}^0, x_{12}^k, x_2^k, x_{23}^{k0}, x_{23}^{kl}, x_3^{kl} \geq 0.
\end{aligned} \tag{5.6}$$

[Lusting et al. (1991)] explored the advantages of using splitting variable formulations for solving deterministic equivalent models by using Interior Point algorithms. [Escudero and Kamesan] report on extensive computational tests comparing different strategies for Interior Point methods, the Simplex method and their crash procedure plus the Simplex method. The results of both studies do not go in the same direction, but neither of them analyzes decomposition approaches.

6. FULL RECOURSE MODEL DUALIZATION

6.1. General framework

The nonanticipativity constraints (5.1) constitute the major obstacle to the decomposition of model (5.6). An Augmented Lagrangian method partially overcomes this difficulty by moving the constraints (5.2) into the objective function. The resulting problem becomes

$$\max_{\pi} D_{\rho}(\pi), \quad (6.1)$$

where the function $D_{\rho}(\pi)$ is defined as

$$\begin{aligned} D_{\rho}(\pi) = \min \sum_{s \in S} w^s & \left((h^s)^T x^s + (\pi_{12}^k)^T (x_{12}^0 - x_{12}^k) + (\pi_{23}^{kl})^T (x_{23}^{k0} - x_{23}^{kl}) \right. \\ & \left. + \frac{\rho}{2} (\|x_{12}^0 - x_{12}^k\|^2 + \|x_{23}^{k0} - x_{23}^{kl}\|^2) \right) \\ \text{s.t.} \quad & x^s \in X \end{aligned} \quad (6.2)$$

where we have used the notation

$$x^s \equiv (x_1, x_{12}^0, x_2^k, x_{23}^{k0}, x_3^{kl}), \quad (6.3)$$

the feasible set X is defined as

$$\begin{aligned} X \equiv \{x^s \mid & A_1 x_1 + A_{12}^1 x_{12}^0 = b_1 \\ & A_{12}^2 x_{12}^k + A_2 x_2^k + A_{23}^2 x_{23}^{k0} = b_2^k \quad \forall k \\ & A_{23}^3 x_{23}^{kl} + A_3 x_3^{kl} = b_3^{kl} \quad \forall k, l \\ & x_1, x_{12}^0, x_{12}^k, x_2^k, x_{23}^{k0}, x_{23}^{kl}, x_3^{kl} \geq 0 \quad \} \end{aligned} \quad (6.4)$$

$\pi \equiv (\pi_{12}^k, \pi_{23}^{kl})$ is the vector of dual variables for the constraints (5.2), and $\rho > 0$ is a penalty parameter.

A rough algorithm to solve (6.1) is as follows:

Algorithm A-1

1. For a given multiplier vector π_i available at iteration i , solve problem (6.2). Let x_i^s be the solution of this problem.
2. If for some tolerance parameter $\epsilon > 0$,

$$\|(x_{12}^0)_i - (x_{12}^k)_i\| \leq \epsilon \quad \text{and} \quad \|(x_{23}^{k0})_i - (x_{23}^{kl})_i\| \leq \epsilon \quad \forall k, l,$$

holds, then stop; the optimal solution for (6.2), and for the original problem (5.6), has been found.

3. Otherwise, reduce by an adequate amount the penalty parameter ρ , and update the dual multipliers π according to

$$\begin{aligned} \pi_{i+1}^k &= \pi_i^k - \beta(x_{12}^0 - x_{12}^k) \quad k = 1, 2, 3, \\ \pi_{i+1}^{kl} &= \pi_i^{kl} - \beta(x_{23}^{k0} - x_{23}^{kl}) \quad k, l = 1, 2, 3, \end{aligned} \tag{6.5}$$

where $\beta \geq 0$ is an appropriate steplength.

It is well known that if (6.1) has a solution, then algorithm **A-1** converges in a finite number of iterations—see [Bertsekas (1992)], for example.

A potential disadvantage of the Augmented Lagrangian approach presented above is that the objective function in (6.2) is not linear but quasi-separable quadratic. Note that any coupling constraints of the form (5.2) have two nonzero elements taking the values $(+1, -1)$; also, the constraints in (6.2) are decomposable into network-like subsets. In the following sections we present several frameworks to decompose the objective function in (6.2).

6.2. Iterative fixed variables approach

Let $\bar{x}$ denote a given value for vector x (usually obtained from the optimal solution for (6.2) in a previous iteration). An iterative procedure may compute a solution for (6.1) by solving at each iteration the following set of independent submodels:

$$\min h_1^T x_1 + (h_{12}^0 - \pi_{12}^0)^T x_{12}^0 + \frac{\rho}{2} \sum_k \|x_{12}^0 - \bar{x}_{12}^k\|^2$$

$$\begin{aligned}
\text{s.t. } & A_1 x_1 + A_{12}^1 x_{12}^0 = b_1 \\
& x_1, \quad x_{12}^0 \geq 0
\end{aligned} \tag{6.6}$$

$$\begin{aligned}
\min & (\pi_{12}^k)^T x_{12}^k + (h_2^k)^T x_2^k + (h_{23}^{k0} - \pi_{23}^{k0})^T x_{23}^{k0} \\
& + \frac{\rho}{2} (\|\bar{x}_{12}^0 - x_{12}^k\|^2 + \Sigma_l \|x_{23}^{k0} - \bar{x}_{23}^{kl}\|^2) \\
\text{s.t. } & A_{12}^2 x_{12}^k + A_2 x_2^k + A_{23}^2 x_{23}^{k0} = b_2^k \\
& x_{12}^k, \quad x_2^k, \quad x_{23}^{k0} \geq 0
\end{aligned} \tag{6.7}$$

for $k = 1, 2, 3$.

$$\begin{aligned}
\min & (\pi_{23}^{kl})^T x_{23}^{kl} + (h_3^{kl})^T x_3^{kl} + \frac{\rho}{2} \|\bar{x}_{23}^{k0} - x_{23}^{kl}\|^2 \\
\text{s.t. } & A_{23}^3 x_{23}^{kl} + A_3 x_3^{kl} = b_3^{kl} \\
& x_{23}^{kl}, \quad x_3^{kl} \geq 0
\end{aligned} \tag{6.8}$$

for $k, l = 1, 2, 3$.

The dual parameters π_{12}^0 and π_{23}^{k0} can be obtained as

$$\begin{aligned}
\pi_{12}^0 &= \Sigma_k \pi_{12}^k \\
\pi_{23}^{k0} &= \Sigma_l \pi_{23}^{kl}, \quad k = 1, 2, 3.
\end{aligned} \tag{6.9}$$

Note that the models are separable QP networks—see [Dembo (1987)], [Escudero (1986)], [Sartenaer (1991)], [Toint and Tuytens (1990)].

6.3. Separable quadratic approximation

If the quadratic terms of the form $\|x_i - x_j\|^2$ in (6.2) are expanded, and the cross-product terms $x_i^T x_j$ are approximated using the suggestion by [Stephanopoulos and Westerberg (1975)]—see also [Mulvey and Ruszczyński (1992)], [Ruszczyński (1993)], for values $x_i^T x_j$ and $\bar{x}_i^T \bar{x}_j$ we can write

$$\begin{aligned}
\|x_i - x_j\|^2 &= \|x_i\|^2 + \|x_j\|^2 - 2x_i^T x_j \\
x_i^T x_j &= -\bar{x}_i^T \bar{x}_j + \bar{x}_i^T x_j + x_i^T \bar{x}_j + (x_i - \bar{x}_i)^T (x_j - \bar{x}_j) \\
\|x_i - x_j\|^2 &\approx \|x_i\|^2 + \|x_j\|^2 + 2\bar{x}_i^T \bar{x}_j - 2\bar{x}_i^T x_j - 2x_i^T \bar{x}_j,
\end{aligned}$$

where the terms of the form $(x_i - \bar{x}_i)^T(x_j - \bar{x}_j)$ have been omitted.

Using this result, the objective function in (6.2) can be approximated by

$$\begin{aligned}
F(x, \pi, \rho) = & g_1^T x_1 + (g_{12}^0)^T x_{12}^0 + \sum_k \left((g_{12}^k)^T x_{12}^k + (g_2^k)^T x_2^k \right. \\
& \left. + (g_{23}^{k0})^T x_{23}^{k0} \right) + \sum_{k,l} \left((g_{23}^{kl})^T x_{23}^{kl} + (g_3^{kl})^T x_3^{kl} \right) \\
& + \frac{\rho}{2} \left(\|x_{12}^0\|^2 + \sum_k (\|x_{12}^k\|^2 + \|x_{23}^{k0}\|^2) + \sum_{k,l} \|x_{23}^{kl}\|^2 \right) + f,
\end{aligned} \tag{6.10}$$

where

$$\begin{aligned}
g_1 &= h_1 \\
g_{12}^0 &= h_{12}^0 - \pi_{12}^0 - \rho \Sigma_k \bar{x}_{12}^k \\
g_{12}^k &= \pi_{12}^k - \rho \bar{x}_{12}^0 \\
g_2^k &= h_2^k \\
g_{23}^{k0} &= h_{23}^{k0} - \pi_{23}^{k0} - \rho \Sigma_l \bar{x}_{23}^{kl} \\
g_{23}^{kl} &= \pi_{23}^{kl} - \rho \bar{x}_{23}^{k0} \\
g_3^{kl} &= h_3^{kl} \\
f &= \rho (\bar{x}_{12}^0 \Sigma_k \bar{x}_{12}^k + \Sigma_k \bar{x}_{23}^{k0} \Sigma_l \bar{x}_{23}^{kl}).
\end{aligned} \tag{6.11}$$

Now the problem

$$\begin{aligned}
\min_x \quad & F(x, \pi, \rho) \\
\text{s.t.} \quad & x \in X,
\end{aligned} \tag{6.12}$$

where X is defined as in (6.4), can be decomposed into quadratic sub-problems with network constraints.

Let

$$\bar{x}^s \equiv (\bar{x}_1, \bar{x}_{12}^0, \bar{x}_{12}^k, \bar{x}_2^k, \bar{x}_{23}^{k0}, \bar{x}_{23}^{kl}, \bar{x}_3^{kl}).$$

Our rough algorithm for solving (6.1) via separable quadratic approximations is as follows:

Let i be a given iteration of algorithm **A-1** and π_i and $\bar{x}_i$ be the dual multipliers and current point, respectively.

Algorithm A-2

1. Let $\pi = \pi_i$, $\bar{x}_{im} = \bar{x}_i$ and $m = 1$.

2. Solve (6.12) with $\bar{x} = \bar{x}_{im}$, to obtain a new point x_{im} .
3. If $\|x_{im} - \bar{x}\| \leq \epsilon$, then stop. Otherwise, set

$$x_{i,m+1} = x_{im} + \gamma(x_{im} - \bar{x}_{im}),$$

where γ is an appropriate steplength, increase m by one and go to step 2.

7. BENDERS DECOMPOSITION

7.1. Splitting variable approach

The splitting variable model (5.6) is a natural candidate for Benders decomposition. Basically, we follow the decomposition methodology given by [Benders (1962)]; we assume that the feasible set is a non-empty and bounded set.

The dual of (5.6) has the form

$$\begin{aligned}
& \max \quad b_1^T \lambda_1 + \sum_k (b_2^k)^T \lambda_2^k + \sum_{k,l} (b_3^{kl})^T \lambda_3^{kl} \\
& \text{s.t.} \quad A_1^T \lambda_1 \leq h_1 \\
& \quad (A_{12}^1)^T \lambda_1 + \sum_k \pi^k \leq h_{12}^0 \\
& \quad -\pi^k + (A_{12}^2)^T \lambda_2^k \leq 0 \quad \forall k \\
& \quad A_2^T \lambda_2^k \leq h_2^k \quad \forall k \\
& \quad (A_{23}^2)^T \lambda_2^k + \sum_l \pi^{kl} \leq h_{23}^{k0} \quad \forall k \\
& \quad -\pi^{kl} + (A_{23}^3)^T \lambda_3^{kl} \leq 0 \quad \forall k, l \\
& \quad A_3^T \lambda_3^{kl} \leq h_3^{kl} \quad \forall k, l
\end{aligned} \tag{7.1}$$

where $\lambda \equiv (\lambda_1, \lambda_2^k, \lambda_3^{kl})$ and $\pi \equiv (\pi^k, \pi^{kl})$ are the vectors of dual variables associated with the constraints in (5.6).

Note that for fixed values of π we obtain independent submodels in (7.1). Let us treat the π -variables as complicating variables. See that by partitioning (7.1) and then obtaining the dual of the resulting problem,

we have that the solution for the preceding problem can be obtained from

$$\max_{\pi} D_B(\pi), \quad (7.2)$$

where

$$\begin{aligned} D_B(\pi) &= \max_{\lambda} b_1^T \lambda_1 + \sum_k (b_2^k)^T \lambda_2^k + \sum_{k,l} (b_3^{kl})^T \lambda_3^{kl} \\ \text{s.t.} \quad A_1^T \lambda_1 &\leq h_1 \\ (A_{12}^1)^T \lambda_1 &\leq h_{12}^0 - \sum_k \pi^k \\ (A_{12}^2)^T \lambda_2^k &\leq \pi^k \quad \forall k \\ A_2^T \lambda_2^k &\leq h_2^k \quad \forall k \\ (A_{23}^2)^T \lambda_2^k &\leq h_{23}^{k0} - \sum_l \pi^{kl} \quad \forall k \\ (A_{23}^3)^T \lambda_3^{kl} &\leq \pi^{kl} \quad \forall k, l \\ A_3^T \lambda_3^{kl} &\leq h_3^{kl} \quad \forall k, l, \end{aligned} \quad (7.3)$$

and the primal of (7.3) would then have the form

$$\begin{aligned} D_B(\pi) &= \min_{x \in X} h_1^T x_1 + (h_{12}^0 - \sum_k \pi^k)^T x_{12}^0 + \sum_k \left((\pi^k)^T x_{12}^k + (h_2^k)^T x_2^k \right. \\ &\quad \left. + (h_{23}^{k0} - \sum_l \pi^{kl})^T x_{23}^{k0} \right) + \sum_{k,l} \left((\pi^{kl})^T x_{23}^{kl} + (h_3^{kl})^T x_3^{kl} \right), \end{aligned} \quad (7.4)$$

where once again X denotes the feasible set defined in (6.4).

The solution for the minimization problem in (7.4) is always attained at some extreme point. Let $\bar{X}$ denote the set of extreme points

of X , $\bar{x}_j \in \bar{X}$ for any extreme point $\bar{x}_j$, and let J be index set corresponding to a complete enumeration of these extreme points. Define

$$\begin{aligned} \bar{F}_j \equiv & h_1^T(\bar{x}_1)_j + (h_{12}^0)^T(\bar{x}_{12}^0)_j + \sum_k \left((h_2^k)^T(\bar{x}_2^k)_j + (h_{23}^{k0})^T(\bar{x}_{23}^{k0})_j \right) \\ & + \sum_{k,l} (h_3^{kl})^T(\bar{x}_3^{kl})_j \quad \forall j \in J, \end{aligned} \quad (7.5)$$

so that we have the equivalent problem to (7.2),

$$\begin{aligned} \max_{z, \pi} \quad & z \\ \text{s.t.} \quad & z \leq \bar{F}_j + \sum_k \left((\bar{x}_{12}^k)_j - (\bar{x}_{12}^0)_j \right)^T \pi^k \\ & + \sum_{k,l} \left((\bar{x}_{23}^{kl})_j - (\bar{x}_{23}^{k0})_j \right)^T \pi^{kl} \quad \forall j \in J. \end{aligned} \quad (7.6)$$

Problem (7.6) is not easy to solve, given the potentially large number of constraints in its formulation. Hence, we use a relaxation methodology, where only a few of the constraints in (7.6) are considered. Let us call this relaxation of (7.6) the *master problem*, and let $(\bar{z}, \bar{\pi})$ denote its optimal solution. Note that $\bar{z}$ is an upper bound on the optimal value of z , as $(\bar{z}, \bar{\pi})$ is optimal for (7.6) if it is feasible for all constraints in (7.6). To check the feasibility of $(\bar{z}, \bar{\pi})$, we may solve the *auxiliary problem*,

$$\begin{aligned} z = \min_{x \in X} \quad & h_1^T x_1 + (h_{12}^0 - \sum_k \bar{\pi}^k)^T x_{12}^0 + \sum_k \left((\bar{\pi}^k)^T x_{12}^k + (h_2^k)^T x_2^k \right) \\ & + (h_{23}^{k0} - \sum_l \bar{\pi}^{kl})^T x_{23}^{k0} + \sum_{k,l} \left((\bar{\pi}^{kl})^T x_{23}^{kl} + (h_3^{kl})^T x_3^{kl} \right), \end{aligned} \quad (7.7)$$

and if $\bar{z} = z$ then we stop, as the optimal solution for the full recourse model (5.3) has been found. Note that $\bar{z} < z$ is mathematically impossible, although some computational care should be taken. If $\bar{z} > z$, then the optimal value for (7.7), say $\bar{x}_j$, is the new extreme point whose related constraint in (7.6) must be generated and appended to the master problem. This enlarged problem must then be reoptimized, and the procedure must be repeated. Since there is only a finite number of extreme points, the process will terminate in a finite number of iterations.

So, our rough algorithm is as follows:

Algorithm A-3

1. Start with an initial value for the variables

$$x = (x_1, x_{12}^0, x_{12}^k, x_2^k, x_{23}^{k0}, x_{23}^{kl}, x_3^{kl}).$$

Set $i = 1$.

2. With the best values available for x we solve the master problem (7.6) to obtain the corresponding multiplier vector $\bar{\pi}$ and the solution value $\underline{z}$.
3. Using the vector $\bar{\pi}$, we form and solve the network submodels that constitute the auxiliary problem (7.7) to obtain a new extreme point $\bar{x}_i$ and a related solution value $\bar{z}$.
4. If $\bar{z} > \underline{z} + \epsilon$ for a given positive tolerance parameter ϵ , then let $i = i + 1$, and go to step 2.
5. Otherwise, stop as the computationally optimal solution for model FR (5.3) has been found. The solution can be obtained as

$$x^* = \sum_{j=1, \dots, i} \bar{x}_j \lambda_j, \quad (7.8)$$

where λ_j is the Lagrange multiplier associated with the j -th constraint in problem (7.6).

7.2. The compact formulation approach

An alternative procedure to solve problem FR (5.3) via a decomposition approach starts with the compact representation given in (5.5). This formulation has the property that if the values for the variables x_{12}^0 and x_{23}^{k0} are fixed, then the optimal values for the remaining variables can be obtained from the solution of separable subproblems—with network constraints.

To introduce an scheme for problem (5.5) that computes first the values of the complicating variables, and then the values of all other variables, we rewrite the problem as

$$\begin{aligned} \min \quad & (h_{12}^0)^T x_{12}^0 + \sum_k (h_{23}^{k0})^T x_{23}^{k0} + \varphi(x_{12}^0, x_{23}^{k0}) \\ \text{s.t.} \quad & x_{12}^0, \quad x_{23}^{k0} \geq 0, \end{aligned} \quad (7.9)$$

where we have isolated the complicating variables, and all other variables are treated through the function φ . This function is defined as

$$\begin{aligned}
& \varphi(\bar{x}_{12}^0, \bar{x}_{23}^{k0}) = \\
& \min \quad h_1^T x_1 + \sum_k (h_2^k)^T x_2^k + \sum_{k,l} (h_3^{kl})^T x_3^{kl} \\
& \text{s.t.} \quad A_1 x_1 = b_1 - A_{12}^1 \bar{x}_{12}^0 \\
& \quad \quad A_2 x_2^k = b_2^k - A_{12}^2 \bar{x}_{12}^0 - A_{23}^2 \bar{x}_{23}^{k0} \quad \forall k \\
& \quad \quad A_3 x_3^{kl} = b_3^{kl} - A_{23}^3 \bar{x}_{23}^{k0} \quad \forall k, l \\
& \quad \quad x_1, \quad x_2^k, \quad x_3^{kl} \geq 0.
\end{aligned} \tag{7.10}$$

This problem has the properties mentioned above: once the values of the variables x_{12}^0 and x_{23}^{k0} are fixed, we obtain a model that is separable in the variables x_1 , x_2^k and x_3^{kl} .

The function φ can also be defined in terms of the dual of (7.10),

$$\begin{aligned}
& \varphi(\bar{x}_{12}^0, \bar{x}_{23}^{k0}) = \\
& \max_{\lambda} \quad (b_1 - A_{12}^1 \bar{x}_{12}^0)^T \lambda_1 + \sum_k (b_2^k - A_{12}^2 \bar{x}_{12}^0 - A_{23}^2 \bar{x}_{23}^{k0})^T \lambda_2^k \\
& \quad \quad + \sum_{k,l} (b_3^{kl} - A_{23}^3 \bar{x}_{23}^{k0})^T \lambda_3^{kl} \\
& \quad \quad A_1^T \lambda_1 \leq h_1 \\
& \quad \quad A_2^T \lambda_2^k \leq h_2^k \quad \forall k \\
& \quad \quad A_3^T \lambda_3^{kl} \leq h_3^{kl} \quad \forall k, l.
\end{aligned} \tag{7.11}$$

From this definition we can derive a simple characterization for $\varphi(\bar{x}_{12}^0, \bar{x}_{23}^{k0})$ to be used in (7.9).

As in the preceding formulation, we may observe that the solutions for the optimization problems in (7.10) and (7.11) will be attained at some extreme point. More specifically, let $\bar{\Lambda}$ denote the set of extreme points of the feasible region in (7.11), $\bar{\lambda}_i \in \bar{\Lambda}$ for any extreme point $\bar{\lambda}_i$ where

$$\bar{\lambda} = (\bar{\lambda}_1, \bar{\lambda}_2^k, \bar{\lambda}_3^{kl}),$$

and let I be index set corresponding to a complete enumeration of these extreme points. Then (7.11) is equivalent to

$$\begin{aligned}
& \min_z z \\
& \varphi(\bar{x}_{12}^0, \bar{x}_{23}^{k0}) = \\
& \text{s.t. } z \geq (b_1 - A_{12}^1 \bar{x}_{12}^0)^T \bar{\lambda}_{1i} + \sum_k (b_2 - A_{12}^2 \bar{x}_{12}^0 - A_{23}^2 \bar{x}_{23}^{k0})^T \bar{\lambda}_{2i}^k \\
& \quad + \sum_{k,l} (b_3^{kl} - A_{23}^3 \bar{x}_{23}^{k0})^T \bar{\lambda}_{3i}^{kl} \quad \forall i \in I, \quad (7.12)
\end{aligned}$$

and this characterization can be introduced in (7.9) to obtain the following equivalent problem

$$\begin{aligned}
& \min (h_{12}^0)^T x_{12}^0 + \sum_k (h_{23}^{k0})^T x_{23}^{k0} + \theta \\
& \text{s.t.} \quad \begin{aligned} & \left(\bar{\lambda}_{1i}^T A_{12}^1 + (\sum_k \bar{\lambda}_{2i}^k)^T A_{12}^2 \right) x_{12}^0 \\ & + \left((\sum_k \bar{\lambda}_{2i}^k)^T A_{23}^2 + (\sum_{k,l} \bar{\lambda}_{3i}^{kl})^T A_{23}^3 \right) x_{23}^{k0} + \theta \geq \zeta_i \quad \forall i \in I \\ & x_{12}^0, \quad x_{23}^{k0} \geq 0. \end{aligned} \quad (7.13)
\end{aligned}$$

where $\zeta_i \equiv \bar{\lambda}_{1i}^T b_1 + \sum_k (\bar{\lambda}_{2i}^k)^T b_2^k + \sum_{k,l} (\bar{\lambda}_{3i}^{kl})^T b_3^{kl}$.

As the number of extreme points for problem (7.11) can be very large, we must introduce some procedure that requires the use of a limited number of them in (7.13).

The solution process for problem (5.5) using Benders decomposition would proceed through the following rough algorithm:

Algorithm A-4

1. Start with an initial value for the variables x_{12}^0 and x_{23}^{k0} . Set $i = 1$.
2. With the last values for x_{12}^0 and x_{23}^{k0} we solve (7.11), and we obtain the corresponding multiplier vector $\bar{\lambda}_i$.
3. We add a constraint of the form (7.12) to the master problem (7.13), and we solve this problem to obtain a new value for the variables x_{12}^0 , x_{23}^{k0} and θ .

4. If the value of θ lies within a prespecified tolerance ϵ of the best value of $\varphi(x_{12}^0, x_{23}^{k0})$, then terminate. Otherwise, let $i = i + 1$ and go to step 2.

The procedure is guaranteed to terminate in a finite number of iterations, as there is only a finite number of extreme points for problem (7.11).

8. DANTZING-WOLFE DECOMPOSITION

One last decomposition scheme considered in this work is based on the splitting variable representation (5.6) for model FR (5.3). To simplify the formulation of the models, we will assume that the feasible region is bounded.

Let X be the feasible set defined in (6.4). The set X is convex, and as we assume it is bounded, any point in the feasible set for (5.6) can be written as a convex combination of extreme points $\bar{x}_j$ for (6.4). Using this characterization, problem (5.6) can be reformulated as

$$\begin{aligned}
\min_{\omega} \quad & \sum_j \omega_j \left(h_1^T \bar{x}_{1j} + (h_{12}^0)^T \bar{x}_{12j} + \sum_k \left((h_2^k)^T \bar{x}_{2j} + (h_{23}^{k0})^T \bar{x}_{23j} \right) \right. \\
& \quad \left. + \sum_{k,l} (h_3^{kl})^T \bar{x}_{3j} \right) \\
\text{s.t.} \quad & \sum_j \omega_j (\bar{x}_{12j}^0 - \bar{x}_{12j}^k) = 0 \quad \forall k \\
& \sum_j \omega_j (\bar{x}_{23j}^{k0} - \bar{x}_{23j}^{kl}) = 0 \quad \forall k, l \\
& \sum_j \omega_j = 1 \\
& \omega_j \geq 0 \quad \forall j \in J.
\end{aligned} \tag{8.1}$$

Again, the number of extreme points $\bar{x}_j$ is potentially very high, implying that any efficient algorithm must solve the problem using only a reduced number of extreme points. The Dantzig-Wolfe decomposition scheme [Dantzing and Wolfe (1960)] introduces a new extreme point if it can be “priced-out” into the basis, that is, if the reduced cost of the corresponding weight ω_i is negative, and so the variable would enter the basis.

The reduced cost for an extreme point $\bar{x}_j$ will be given by

$$\begin{aligned} \sigma(\bar{x}_j) \equiv & h_1^T \bar{x}_{1j} + (h_{12}^0)^T \bar{x}_{12j} + \sum_k ((h_2^k)^T \bar{x}_{2j} + (h_{23}^{k0})^T \bar{x}_{23j}) \\ & + \sum_{k,l} (h_3^{kl})^T \bar{x}_{3j} - \sum_k (\pi^k)^T (\bar{x}_{12j}^0 - \bar{x}_{12j}^k) - \sum_{k,l} (\pi^{kl})^T (\bar{x}_{23j}^{k0} - \bar{x}_{23j}^{kl}) - \theta, \end{aligned} \quad (8.2)$$

where $(\pi^k, \pi^{kl}, \theta)$ is the set of multipliers associated with the equality constraints in (8.1).

The best candidate—from a reduced-cost point of view—to enter the basis would then be the extreme point $x \in X$ that would give the smallest value for $\sigma(x)$, that is, the solution of the problem

$$\begin{aligned} \min_x \quad & h_1^T x_1 + (h_{12}^0 - \sum_k \pi^k)^T x_{12}^0 + \sum_k ((\pi^k)^T x_{12}^k + (h_2^k)^T x_2^k \\ & + (h_{23}^{k0} - \sum_l \pi^{kl})^T x_{23}^{k0}) + \sum_{k,l} ((\pi^{kl})^T x_{23}^{kl} + (h_3^{kl})^T x_3^{kl}) \end{aligned} \quad (8.3)$$

$$\text{s.t.} \quad x \in X$$

Note again that (8.3) is separable into independent linear subproblems with network constraints.

The solution process using the Dantzig-Wolfe approach can be obtained by a procedure based on the following rough algorithm:

Algorithm A-5

1. Start with an initial value for the dual variables π^k and π^{kl} . Set $i = 1$.
2. With the last values for π^k and π^{kl} we solve (8.3), and we obtain an extreme point for X , say $\bar{x}_i$.
3. If $\sigma(\bar{x}_i) \geq 0$, terminate. Otherwise, update $i = i + 1$, add a new column to problem (8.1), and solve it to obtain a vector of multipliers $(\pi^k, \pi^{kl}, \theta)$. Go to step 2.

Once again, the finite termination of this procedure is a consequence of the finite number of extreme points for the set X (6.4).

9. IMPLEMENTATION RESULTS. PARALLEL IMPLEMENTATION

In this section we present the results obtained from the implementation of some of the algorithms described above. In particular, we consider the codes corresponding to Algorithms **A-4** for the Benders decomposition scheme, and **A-1** and **A-2** for the Augmented Lagrangian approach. The code corresponding to algorithm **A-5** was also developed and tested, but the results we obtained were much worse than those for the other algorithms; these results have been omitted from the tables. Regarding algorithm **A-3**, note that problem (7.6), the master problem, is the dual of problem (8.1), the master problem of algorithm **A-5**, and the subproblems to solve, (7.4) and (8.3) respectively, are the same in both cases, implying that if both master problems are solved exactly, the results obtained will be identical for both algorithms, except perhaps for numerical errors.

A relevant property of the decomposition approach is that all the algorithms described above include some step, normally the most computationally expensive one, in which it is necessary to solve a problem that can be trivially decomposed into the solution of a collection of independent subproblems, see step 2 in Algorithm **A-2**, or step 2 in Algorithm **A-4**, for example. In these cases a reasonable procedure to improve the efficiency of the algorithms is to solve these subproblems in parallel. Along this line, we have produced a C code, SPNET, implementing the Augmented Lagrangian algorithm **A-1** and **A-2**, to run on a distributed computation environment. The choice of the Augmented Lagrangian algorithm was motivated by the fact that it is the algorithm with the largest amount of work that can be performed in parallel; note for example that step 3 in Algorithm **A-4** cannot be easily (or efficiently) performed in a distributed computation environment, with high transmission costs between processors.

Note that while algorithm **A-4** requires the solution of linear network subproblems, algorithm **A-1** and **A-2** must solve quadratic network subproblems. To solve the quadratic networks in (6.6)–(6.8), we have developed a C code, QPNET, based on the preconditioned variable reduction truncated Newton approach described in [Dembo (1987)] and [Escudero (1986)].

The computational experiments were conducted on a network of two HP 9000/720 workstations and one HP 9000/730 workstation running HP-UX 8.05. These workstations are rated at 17 Mflops (HP 720) and

22 Mflops (HP 730). The codes were written in C using the HP-UX “cc” compiler, and the parallel code was prepared using PVM version 3.1.3.

The problems in the set are of the form (5.6) where, in keeping with the form of the problem described in Section 2, the matrices A_k correspond to the node-arc incidence matrix of a given basic network (representing the reservoir network of some river basin), that is repeated for a certain number of time periods, and the coefficients h and b have been randomly generated. The sizes of the basic and replicated networks for the test problems are shown in Table 1. In a subsequent step we introduced the uncertainty in the exogenous water supply, via scenarios. The number of scenarios, the total number of subproblems and the size of the full problem corresponding to each test problem are indicated in Table 2. All subproblems have been defined to include just one time period.

Table 1. *Deterministic problem specifications*

	Basic net.			Replicated net.	
Problem	m	n	T	m	n
red4-7	4	3	7	28	45
red4-10	4	3	10	40	66
red4-30	4	3	30	120	206
red10-6	10	9	6	60	104
red10-7	10	9	7	70	123
red10-30	10	9	30	300	560
red10-50	10	9	50	500	940
red25-5	25	41	5	125	305
red25-6	25	41	6	150	371
red25-7	25	41	7	175	437
red25-10	25	41	10	250	635
red35-5	35	60	5	175	440
red35-6	35	60	6	210	535
red35-7	35	60	7	245	630
red35-10	35	60	10	350	915

Legend : n : number of variables (arcs)
 m : number of constraints (nodes)
 T : number of periods

Finally, Table 3 shows the computational results both for the Benders decomposition algorithm A-4 and for the Augmented Lagrangian

algorithm **A-1** and **A-2** in its sequential and distributed computation versions. The termination criterion for all problems was to stop when at any given iteration the sum of the normalized infeasibility and the normalized difference between primal and dual objective values for problem (5.6) was less than 10^{-4} .

For comparison purposes, the running times using the Interior Point code OB1, [Lusting et al. (1992)], to solve the whole hydropower generation problem have also been included for some cases; note that higher problem dimensions imply poorer performances for codes solving the full model. All sequential codes were run on a 720 workstation, except for the OB1 code, due to the limited memory available in the 720 workstations (16 MB); in this last case the running times shown in the table are for a 730 workstation. The ratio of execution times in both workstations is approximately equal to 1.3.

Table 2. *Complete problem specifications*

Problem	$ S $	# Subp.	m	n	nel
red4-7	233	385	1540	1765	4454
red4-10	3625	6027	24108	27689	69874
red4-30	154	3848	15392	26320	53252
red10-6	96	152	1520	1928	4806
red10-7	233	385	3850	4985	12290
red10-30	154	3848	38480	71572	144674
red10-50	154	6928	69280	130920	261714
red25-5	34	56	1400	2846	6517
red25-6	96	152	3800	7632	17639
red25-7	233	385	9625	19585	44970
red25-10	564	1131	28275	60546	135167
red35-5	34	56	1960	4130	9415
red35-6	96	152	5320	11080	25485
red35-7	233	385	13475	28420	64960
red35-10	564	1131	39585	87705	195115

Legend : $|S|$: number of scenarios
 $\#Subp.$: total number of subproblems
 n : number of variables
 m : number of constraints
 nel : number of nonzero elements in the
constraint matrix

The times in Table 3 are given in the format h:min:sec. The blank entries in the table correspond to problems whose size exceeded the memory available in the 720 (16 Mb) or 730 (32 MB) workstations; note that the Augmented Lagrangian code had no difficulty fitting into this reduced amount of memory.

Table 3. *Computational results*

	OB1		SBD		SLD			PLD	
Problem	Iter.	Time	Iter.	Time	Iter.	QP iter.	Time	Iter.	Time
red4-7	21	0:07	11	1:30	1010	230250	1:55	1010	1:03
red4-10					906	3311299	27:46	906	10:14
red4-30	29	1:51			122	369359	2:49	122	1:13
red10-6	22	0:09	22	2:45	910	140983	2:07	910	1:05
red10-7	25	0:19			726	288569	4:30	726	1:57
red10-30					322	1426657	21:10	322	9:06
red10-50					337	2679177	42:52	337	16:49
red25-5	27	0:18	69	13:42	214	16497	0:37	214	0:17
red25-6	28	0:57			468	83892	3:13	468	1:11
red25-7	33	3:22			266	135442	5:09	266	1:54
red25-10					321	468872	18:20	321	8:43
red35-5	28	0:25	>80	>1:00:00	167	15775	0:45	167	0:20
red35-6	31	1:41			407	79387	4:04	407	1:37
red35-7	36	6:31			319	166053	8:22	319	3:13
red35-10					463	663010	37:11	463	16:14

Legend : *SBD* : Benders decomposition
SLD : Sequential Augmented Lagrangian decomposition
PLD : Parallel Augmented Lagrangian decomposition
QP iter. : Total number of QP iterations

From these results it is clear that the Augmented Lagrangian procedure is the method of choice as a decomposition algorithm, with sequential times very close (or better) to those of a good LP code; again, note that the times are not directly comparable, as they correspond to slightly different machines. Also, the improvement in the running times obtained when running the parallel version of the code (even accounting for the non-homogeneous computing environment) is reasonably close to the number of workstations used (3) for all problems, again showing

the excellent properties of the Augmented Lagrangian algorithm in a parallel computation environment.

Figure 1. *Relation between scenarios and parameter's outlooks*

10. CONCLUSIONS

In this paper we have presented a formulation of a general hydropower generation management problem as an instance of the network flow problem with uncertainty. We have proposed a methodology based on “scenario analysis” to deal with the uncertainty. It results in a huge model with network-like constraints and complicating constraints. Several frameworks are proposed to solve the problem: an augmented Lagrangian approach based on the dualization of the complicating constraints in the splitting variable formulation, a Benders decomposition

scheme applied to the complicating variables in the compact formulation and to the complicating constraints in the splitting variable formulation, and a Dantzig-Wolfe decomposition applied to the complicating constraints.

Figure 2. *Implementable (Nonanticipative) Policies.*

Given the huge dimensions of the problem, we have chosen to approximate the hydropower generation function by solving a sequence of linear approximations to the original stochastic problem. As a result, we have separable quadratic network submodels for the first decomposition scheme, and pure linear independent network submodels for the other cases. Comprehensive computing experimentation is under way to assess the potential of the different approaches.

The numerical results obtained so far show clearly that the best decomposition scheme is the one based on the use of an Augmented Lagrangian approach, with running times comparable to those for good LP codes solving the whole problem, at least for problems with sizes in the range of the ones in the test set (tens to hundreds of thousands of variables); also, this approach requires far less memory to run than the full LP codes. Furthermore, the decomposition algorithm lends itself easily to being implemented in a parallel computation environment, with reductions in time very close to the optimal values.

Future lines of research include improving certain aspects of the Augmented Lagrangian algorithm, with particular emphasis in the study of more efficient methods for the update of the different parameters in the algorithm, such as the penalty parameter, left unmodified at the moment, the procedure indicated in (6.5) to update the multipliers, or the procedure in step 2 of Algorithm **A-2** to update the reference values $\bar{x}$.

REFERENCES

- Benders, J.F. (1962). Partitioning procedures for solving mixed variables programming problems, *Numerische Mathematik* **4**, 238–252.
- Bertsekas, D.P. (1992). *Constrained Optimization and Lagrange Multipliers*. Academic Press.
- Dantzig, G.B. and P. Wolfe (1960). A decomposition principle for linear programs, *Operations Research* **8**, 101–111.
- Dembo, R.S. (1987). A primal truncated Newton algorithm with application to large-scale nonlinear network optimization, *Mathematical Programming Study* **31**, 43–72.
- Dembo, R.S. (1991). Scenario optimization, *Annals of Operations Research* **30**, 63–80.

- Escudero, L.F. (1986). Performance evaluation of independent superbasic sets, *European Journal of Operational Research* **23**, 343–355.
- Escudero, L.F. (1990). On solving a nondifferentiable transshipment problem, *Investigación Operativa* **1**, 199–211.
- Escudero, L.F. and P.V. Kamesam, P.V. (to appear). On solving stochastic production planning problems via scenario modelling.
- Escudero, L.F., P.V. Kamesam, A.J. King and R. J.-B. Wets (1993). Production planning via scenario modelling, *Annals of Operations Research* **43**, 311–335.
- Glover, F., D. Klingman and N.V. Phillips (1992) *Network models in optimization and their applications*. John Wiley.
- Jensen, D. and A.J. King (1991) Linear-quadratic efficient frontiers for portfolio optimization, Report RC-17156. IBM T.J. Watson Research Center, Yorktown Heights, N.Y.
- Lustig, I.J., R.E. Marsten and D.F. Shanno (1992). On implementing Mehrotra predictor-corrector interior point method for linear programming, *SIAM Journal on Optimization* **2**, 435–449.
- Lustig, I.J., J.M. Mulvey and T.J. Carpenter (1991). *Formulating stochastic programs for interior point methods*, *Operations Research* **39**, 757–769.
- Mulvey, J.M. and A. Ruszczyński (1992). A diagonal quadratic approximation method for large-scale linear programs, *Operations Research Letters* **12**, 205–221.
- Mulvey, J.M., R.J. Vanderbei and S.A. Zenios (1991). Robust optimization of large-scale systems: general modelling framework and computations, Report 91-06-04, The Wharton School, University of Pennsylvania.
- Mulvey, J.M. and H. Vladimirov (1989). Stochastic networks, optimization models for investment planning, *Annals of Operations Research* **20**, 187–217.
- Rockafellar, R.T. and R.J.-B. Wets (1991). Scenario and policy aggregation in optimization under uncertainty, *Mathematics of Operations Research* **16**, 119–147.
- Ruszczyński, A. (1993). Interior point methods in stochastic programming, Report WP-93-8, IIASA, Vienna.
- Sartenaer, A. (1991). On some strategies for handling constraints in nonlinear optimization, Ph.D. Thesis, Dept. of Mathematics, FUNDP, Namur, Belgium.

- Stephanopoulos, G. and W. Westerberg (1975). The use of Hestenes method of multipliers to resolve dual gaps in engineering system optimization, *Journal of Optimization Theory and Applications* **15**, 285–309.
- Toint, Ph.L. and D. Tuytens (1990). On large-scale nonlinear network optimization, *Mathematical Programming* **48**, 125–159.