Hydro-Storage Subproblems in Power Generation: An Approach with a Relaxation Method for Network Flow Problems

Holger Heitsch and Werner Römisch

Abstract— **Mathematical models for the electricity portfolio management of a utility that owns a hydro-thermal generation system and trades on the power market often lead to complex stochastic optimization problems. We present a new approach to solving stochastic hydro-storage subproblems that occur when stochastic Lagrangian relaxation is applied to solving such models. The special structure of such hydro-storage subproblems allows the design of a stochastic network flow algorithm. The algorithm represents a stochastic extension of a relaxation method, that algorithmically solves the linear minimum cost flow problem. It is based on the iterative improvement of dual costs. Numerical experience of the new algorithm is reported and its performance is compared with that of standard LP software .**

Index Terms—**Stochastic programming, Lagrangian relaxation, hydro-storage subproblems, network optimization, minimum cost flow problem.**

I. INTRODUCTION

In the last few years large scale multi-stage stochastic programming models for the cost-optimal generation and trading of electric power under uncertainty have been developed. Such optimization problems combine several mathematical challenges, namely, mixed-integer decisions, stochastic data and huge dimensions. The uncertainty consists in electrical load forecasts, generator failures, stream flows to hydro reservoirs, and fuel and electricity prices.

One approach for solving such mixed-integer multi-stage models is the stochastic Lagrangian relaxation of coupling constraints ([11], [15], [18]). Its idea consists in assigning stochastic multipliers to coupling constraints and in solving the Lagrangian dual by subgradient methods. This leads to a successive decomposition into finitely many stochastic (contract, thermal and hydraulic) subproblems, for which efficient solution techniques that take advantage of their special structure are needed. In this paper, we discuss the hydro-storage subproblems and their algorithmic solution. They are still multi-stage stochastic programs but exhibit a specific network flow structure. Various optimization models and solution algorithms for hydro-electric scheduling have been discussed in the literature so far, e.g. [6], [7], [8], [10], [14], [16], [20].

Motivated by the structure of stochastic hydro subproblems, we develop an extended version of a relaxation method for solving stochastic minimum cost flow problems based on the iterative improvement of dual costs. We show how this method

H. Heitsch and W. Römisch are with the Institute of Mathematics, Humboldt–University Berlin, D-10099 Berlin, Germany.

applies to hydro-storage subproblems and discuss its computational performance.

The paper is organized as follows. Section II reviews classical linear minimum cost flow problems and their algorithmic solution. In Section III a detailed description of a stochastic extension is given. In Section IV we discuss the stochastic Lagrangian-based decomposition approach to portfolio optimization models for a power utility that owns a hydro-thermal power generation system and the application of the new algorithm to the hydro-storage subproblems. Finally, in Section V numerical experience of the algorithm is provided. Its performance is compared with that of CPLEX 8.0 on a set of test examples.

II. MINIMUM COST FLOW PROBLEMS

Network flow problems are one of the most important and most frequently encountered classes of optimization problems. They naturally arise in the optimization of large systems, such as communication networks and transportation networks. Many important problems, such as shortest path, assignment, maximum flow, transportation, minimum cost flow, and travelling salesmen, belong to the wide spectrum of network optimization. The exhaustive mathematical treatment of network flow problems started with Ford and Fulkerson in 1962 [9]. Since then numerous papers and books on network optimization have appeared. Recent textbooks on network flows are due to Rockafellar [17] in 1984, Bazaraa, Jarvis and Sherali [2] in 1990, Ahuja, Magnanti and Orlin [1] in 1993 and Bertsekas [3] in 1998. For a history of network flow problems we refer to the recent paper by Schrijver [21].

Network optimization problems typically cannot be solved analytically. Usually they have to be solved computationally with one of the available algorithms. Clearly, general linear or nonlinear programming algorithm could be used. However, specialized network optimization algorithms that exploit the network structure turn out to be much more efficient. In practice, network problems can often be solved much faster than general linear or convex programs of comparable dimension. The main ideas that are fundamental for general mathematical programs are maintained in network optimization.

Often network flow problems are modelled in terms of graph-related notions. In general, network flow problems consist of supply and demand points together with several routes connecting these points. The network is given by a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which consists of a set $\mathcal{V}$ of *nodes* and a set $\mathcal{E}$ of

pairs of distinct nodes from $\mathcal{V}$ called *arcs*. The linear minimum cost flow problem consists in finding a set of arc flows that minimizes a linear cost function subject to the constraints that they produce a given divergence vector and that they lie within some given bounds. Formally, it is given by

$$\min \sum_{(i,j) \in \mathcal{E}} a_{ij} x_{ij} \quad \text{s.t.} \tag{1}$$

$$\sum_{(i,j) \in \mathcal{E}} x_{ij} - \sum_{(j,i) \in \mathcal{E}} x_{ji} = s_i, \quad \forall i \in \mathcal{V}, \tag{2}$$

$$b_{ij} \le x_{ij} \le c_{ij}, \quad \forall (i,j) \in \mathcal{E}. \tag{3}$$

The cost coefficients a_{ij}, the flow bounds b_{ij} and c_{ij} for arcs $(i,j) \in \mathcal{E}$, and the values of supply s_i for nodes $i \in \mathcal{V}$ are given scalars. The constraints (2) and (3) are called flow and capacity constraints, respectively. Figure 1 shows a small example network graph.
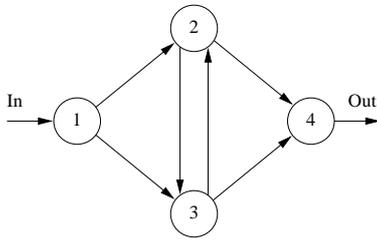
Fig. 1. Example network flow

The available netflow algorithms for solving minimum cost flow problems can be grouped into two main categories:

(1) *Primal cost improvement:* These methods start with a feasible flow vector x and generate a sequence of other feasible flow vectors, each having a smaller primal cost than its predecessor. The main idea is to push the flow along a simple cycle to obtain an improved flow vector as long as the flow vector is not optimal.

(2) *Dual cost improvement:* Such methods solve a dual problem iteratively. A sequence of dual variables (price vectors) is generated such that each new price vector has strictly improved dual cost.

One of the most efficient primal cost improvement methods is the network simplex method. There are several approaches to finding cycles to improve the primal cost, but the most successful ones in practice include specialized versions of the simplex method. An important example for the second group of dual cost improvement or dual ascent methods is the relaxation method. Its main advantage, which distinguishes it from classical primal-dual methods, is that the choice of ascent directions is very simple and, hence, that the computation of dual ascent directions is very fast.

III. STOCHASTIC EXTENSION

Stochastic programming mostly deals with the optimization of decision making under uncertainty over time. The decision to be optimized must not anticipate future outcomes where only probabilistic information on the uncertain data is available. Basic references for the theory, numerical analysis and application of stochastic programming are the monographs [4], [19].

Motivated by the efficiency of current network flow algorithms and, in particular, because of the network structure of some subproblems occurring in power management applications, we have developed a stochastic extension of network optimization models and methods. It applies to stochastic programming models whose underlying (deterministic) optimization problem has network flow structure.

To formulate an extended or stochastic minimum cost flow problem that corresponds to a network $\mathcal{V}$ of nodes we introduce the following terminology. A *multi-arc* is an ordered pair of nonempty disjoint sets whose elements belong to $\mathcal{V}$, i.e., more than one start or end node correspond to a multi-arc. Thus, we refer to a classical arc, i.e., an ordered pair of distinct nodes, as a *single-arc*.
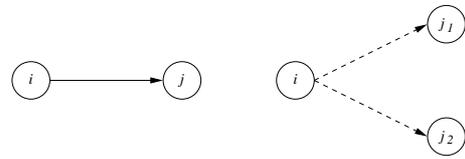
Fig. 2. Example single-arc and multi-arc

Figure 2 shows an example of a single-arc, and an example of a multi-arc that consists of one start node and two end nodes.

Let a pair $(\mathcal{V}, \mathcal{E})$ be given, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ a set of multi-arcs, i.e.,

$$\mathcal{E} \subset \{ (I,J) : I, J \subset \mathcal{V}; I, J \ne \emptyset \text{ and } I \cap J = \emptyset \}. \tag{4}$$

The following optimization problem will be called *extended minimum cost flow problem with multi-arcs*:

$$\min \sum_{(I,J) \in \mathcal{E}} a_{IJ} x_{IJ} \quad \text{s.t.} \tag{5}$$

$$\sum_{\{(I,J) \in \mathcal{E}: i \in I\}} x_{IJ} - \sum_{\{(J,I) \in \mathcal{E}: i \in I\}} x_{JI} = s_i, \forall i \in \mathcal{V}, \tag{6}$$

$$b_{IJ} \le x_{IJ} \le c_{IJ}, \quad \forall (I,J) \in \mathcal{E}. \tag{7}$$

Here, a_{IJ} denote the cost coefficients, b_{IJ} and c_{IJ} the flow bounds for $(I,J) \in \mathcal{E}$, and s_i the supply of node $i \in \mathcal{V}$. The constraints (6) and (7) represent the flow and the capacity constraints, respectively.

A. Duality

There is a well developed duality theory for solving network flow problems (e.g., cf. [3]). The corresponding results can be extended to cover network flow models with multi-arcs. In particular, we need such an extension for dualizing the flow constraints (6) of the extended minimum cost flow problem with multi-arcs. Let μ denote the Lagrange multiplier (or price) with components μ_i for $i \in \mathcal{V}$. Then the Lagrangian function is of

the form

$$
\mathcal{L}(x, \mu) = \sum_{(I,J) \in \mathcal{E}} a_{IJ} x_{IJ}
$$

$$
+ \sum_{i \in \mathcal{V}} \mu_i \left(s_i - \sum_{\{(I,J) \in \mathcal{E}: i \in I\}} x_{IJ} + \sum_{\{(J,I) \in \mathcal{E}: i \in I\}} x_{JI} \right)
$$

$$
= \sum_{(I,J) \in \mathcal{E}} \left(a_{IJ} + \sum_{j \in J} \mu_j - \sum_{i \in I} \mu_i \right) x_{IJ} + \sum_{i \in \mathcal{V}} s_i \mu_i. \quad (8)
$$

The dual function is given by

$$
\mathcal{D}(\mu) = \min_x \{ \mathcal{L}(x, \mu) \, : \, b_{IJ} \le x_{IJ} \le c_{IJ}, \, (I, J) \in \mathcal{E} \}, \quad (9)
$$

and the corresponding dual problem reads

$$
\max \{ \mathcal{D}(\mu) : \mu \in I\!\!R^V \}, \quad (10)
$$

where V is the number of elements in $\mathcal{V}$. As the Lagrangian function $\mathcal{L}(x, \mu)$ is separable in the arc flows x_{IJ}, its minimization decomposes into a separate minimization for each arc $(I, J) \in \mathcal{E}$. $\mathcal{D}(\mu)$ can be written as

$$
\mathcal{D}(\mu) = \sum_{(I,J) \in \mathcal{E}} \mathcal{D}_{IJ} + \sum_{i \in \mathcal{V}} s_i \mu_i, \quad (11)
$$

where

$$
\mathcal{D}_{IJ} = \min_{b_{IJ} \le x_{IJ} \le c_{IJ}} \left(a_{IJ} + \sum_{j \in J} \mu_j - \sum_{i \in I} \mu_i \right) x_{IJ}. \quad (12)
$$

Solving the dual problem provides the correct values of the prices μ_i, which allow to obtain the optimal flow x by minimizing the Lagrangian function.

Now we develop the basic duality results for the extended minimum cost flow problem with multi-arcs. With respect to equation (12) we introduce some helpful terminology. For any price vector μ we say that an arc (I, J) is

$$
\text{inactive} \quad \text{if} \quad \sum_{i \in I} \mu_i < a_{IJ} + \sum_{j \in J} \mu_j,
$$

$$
\text{balanced} \quad \text{if} \quad \sum_{i \in I} \mu_i = a_{IJ} + \sum_{j \in J} \mu_j,
$$

$$
\text{active} \quad \text{if} \quad \sum_{i \in I} \mu_i > a_{IJ} + \sum_{j \in J} \mu_j.
$$

We say that a flow-price vector pair (x, μ) satisfies the *complementary slackness condition* if x satisfies the capacity constraints (7) and if it holds that:

$$
\begin{aligned}
x_{IJ} = b_{IJ} \qquad &\text{for all inactive arcs } (I, J) \in \mathcal{E}, \\
b_{IJ} \le x_{IJ} \le c_{IJ} \qquad &\text{for all balanced arcs } (I, J) \in \mathcal{E}, \\
x_{IJ} = c_{IJ} \qquad &\text{for all active arcs } (I, J) \in \mathcal{E}.
\end{aligned}
$$

The following proposition provides an important duality result, which is analogous to the duality result of the classical minimum cost flow problem (cf. [3, Proposition 4.1]).

Proposition 1: A feasible flow vector x^* and a price vector μ^* satisfy the complementary slackness condition if and only if x^* and μ^* are optimal primal and dual solutions, respectively, and the optimal primal and dual values coincide.

Proof: We first show that for any feasible flow vector x and any price vector μ the primal cost of x is not less than the dual cost of μ. Clearly, for all pairs (x, μ) with a feasible flow x it holds that

$$
\mathcal{L}(x, \mu) = \sum_{(I,J) \in \mathcal{E}} a_{IJ} x_{IJ},
$$

and, thus, due to (9) we obtain

$$
\mathcal{D}(\mu) \le \mathcal{L}(x, \mu) = \sum_{(I,J) \in \mathcal{E}} a_{IJ} x_{IJ}.
$$

If x^* is now feasible and satisfies, together with μ^*, the complementary slackness condition, the equations (9), (11) and (12) imply

$$
\begin{aligned}
\mathcal{D}(\mu^*) &= \min_x \{ \mathcal{L}(x, \mu^*) \, : \, b_{IJ} \le x_{IJ} \le c_{IJ}, \, (I, J) \in \mathcal{E} \} \\
&= \mathcal{L}(x^*, \mu^*) = \sum_{(I,J) \in \mathcal{E}} a_{IJ} x_{IJ}^*.
\end{aligned}
$$

Note that the latter equation is valid because of the feasibility of x^*. Hence, we have shown that for a pair (x^*, μ^*) satisfying the complementary slackness condition, x^* and μ^* are optimal primal and dual solutions, respectively, and the optimal primal and dual values coincide. Conversely, now we show that, if x^* and μ^* are optimal primal and dual solutions, respectively, and if

$$
\mathcal{D}(\mu^*) = \sum_{(I,J) \in \mathcal{E}} a_{IJ} x_{IJ}^*,
$$

holds, the pair (x^*, μ^*) satisfies the complementary slackness condition. By (9) we have

$$
\mathcal{D}(\mu^*) = \min_x \{ \mathcal{L}(x, \mu^*) \, : \, b_{IJ} \le x_{IJ} \le c_{IJ}, \, (I, J) \in \mathcal{E} \}.
$$

Using the Lagrangian expression (8), and due to the feasibility of x^* we obtain

$$
\sum_{(I,J) \in \mathcal{E}} a_{IJ} x_{IJ}^* = \mathcal{L}(x^*, \mu^*).
$$

and, hence,

$$
\mathcal{L}(x^*, \mu^*) = \min_x \{ \mathcal{L}(x, \mu^*) \, : \, b_{IJ} \le x_{IJ} \le c_{IJ}, \, (I, J) \in \mathcal{E} \}.
$$

The latter equation and formula (8) for the Lagrangian imply, in particular, that

$$
x_{IJ}^* \in \arg \min_{b_{IJ} \le x_{IJ} \le c_{IJ}} \left(a_{IJ} + \sum_{j \in J} \mu_j^* - \sum_{i \in I} \mu_i^* \right) x_{IJ}.
$$

holds for all arcs (I, J). Clearly, the latter expression implies that the pair (x^*, μ^*) satisfies the complementary slackness condition. $\square$

B. Dual Ascent Method

Dual ascent methods for the classical minimum cost flow problems (see [3, Chapter 6]) can be extended to models with multi-arcs. Similar to the dual ascent method for the classical problem we are looking for a dual ascent direction in order to change the prices of a connected subset $\mathcal{S}$ of nodes by a certain amount, to improve the dual cost of the dual function (11). However, determining a dual ascent direction is more difficult. To simplify matters we only admit multi-arcs originating at one node, i.e, a multi-arc is of the form (i, J), where $i \in \mathcal{V}$ and $J \subset \mathcal{V}$. Hence, the set of arcs is now restricted to

$$\mathcal{E} \subset \{(i, J) \,:\, i \in \mathcal{V}, \, J \subset \mathcal{V}, \, J \neq \emptyset \text{ and } i \notin J\}. \tag{13}$$

Each iteration of a dual ascent method involves a change of the price vector along a direction of the form $d_{\mathcal{S}} = (d_1, \ldots, d_V)$, where V is the number of nodes in $\mathcal{V}$, d_i, $i \in \mathcal{V}$, is given by

$$d_i = \begin{cases} 1 & \text{if } i \in \mathcal{S}, \\ 0 & \text{if } i \notin \mathcal{S}, \end{cases} \tag{14}$$

and $\mathcal{S}$ is a connected subset of nodes. Such directions $d_{\mathcal{S}}$ and the corresponding sets $\mathcal{S}$ are called *elementary* if $\mathcal{S}$ has the property that, for all arcs $(i, J) \in \mathcal{E}$, the set $J \cap \mathcal{S}$ contains at most one element.

For the decision whether an elementary direction is a direction of dual ascent we have to calculate the directional derivative of the dual cost along $d_{\mathcal{S}}$ and check whether it is positive. It follows from the dual cost expressions (11) and (12) that the directional derivative into an elementary direction is

$$\begin{aligned} D'(\mu; d_{\mathcal{S}}) &= \lim_{\alpha \downarrow 0} \frac{D(\mu + \alpha d_{\mathcal{S}}) - D(\mu)}{\alpha} \\ &= \sum_{\substack{(i,J) \in \mathcal{E}: \\ \text{inactive/balanced} \\ i \notin \mathcal{S}, \, J \cap \mathcal{S} \neq \emptyset}} b_{iJ} + \sum_{\substack{(i,J) \in \mathcal{E}: \\ \text{active} \\ i \notin \mathcal{S}, \, J \cap \mathcal{S} \neq \emptyset}} c_{iJ} \\ &\quad - \sum_{\substack{(i,J) \in \mathcal{E}: \\ \text{inactive} \\ i \in \mathcal{S}, \, J \cap \mathcal{S} = \emptyset}} b_{iJ} - \sum_{\substack{(i,J) \in \mathcal{E}: \\ \text{active/balanced} \\ i \in \mathcal{S}, \, J \cap \mathcal{S} = \emptyset}} c_{iJ} \end{aligned} \tag{15}$$

For a flow vector x let us define the *surplus* g_i of node i as the difference between the total sum of all inflows into i minus the total sum of all outflows from i, i.e.,

$$g_i = \sum_{\{(j,I) \in \mathcal{E} \,:\, i \in I\}} x_{jI} - \sum_{(i,J) \in \mathcal{E}} x_{iJ} + s_i.$$

Note that for a feasible flow vector x the surplus of each node is zero. However, for an elementary direction and corresponding subset of nodes $\mathcal{S} \subset \mathcal{V}$ we obtain

$$\sum_{i \in \mathcal{S}} g_i = \sum_{i \in \mathcal{S}} s_i + \sum_{\substack{(i,J) \in \mathcal{E}: \\ i \notin \mathcal{S}, J \cap \mathcal{S} \neq \emptyset}} x_{iJ} - \sum_{\substack{(i,J) \in \mathcal{E}: \\ i \in \mathcal{S}, J \cap \mathcal{S} = \emptyset}} x_{iJ} \tag{16}$$

To organize the search for an ascent direction and to obtain a suitable set $\mathcal{S}$ with positive directional derivative $D'(\mu; d_{\mathcal{S}})$, it is convenient to maintain a flow vector x satisfying the complementary slackness condition together with μ. For a flow price pair (x, μ) satisfying the complementary slackness condition the term of the directional derivative can be reduced. In this case we obtain for an elementary direction

$$\begin{aligned} D'(\mu; d_{\mathcal{S}}) &= \sum_{i \in \mathcal{S}} g_i - \sum_{\substack{(i,J) \in \mathcal{E}: \\ \text{balanced} \\ i \notin \mathcal{S}, J \cap \mathcal{S} \neq \emptyset}} (x_{iJ} - b_{iJ}) \\ &\quad - \sum_{\substack{(i,J) \in \mathcal{E}: \\ \text{balanced} \\ i \in \mathcal{S}, J \cap \mathcal{S} = \emptyset}} (c_{iJ} - x_{iJ}) \end{aligned} \tag{17}$$

by the equations (15) and (16). It turns out that an elementary node set $\mathcal{S}$ that has positive total surplus is a candidate for generating a direction $d_{\mathcal{S}}$ of dual ascent. The following proposition generalizes an analogous result for the classical minimum cost flow problem (cf. [3, Lemma 6.1]).

Proposition 2: Let x and μ satisfy the complementary slackness condition, and let $\mathcal{S}$ be an elementary subset of nodes. If

$$\sum_{i \in \mathcal{S}} g_i > 0$$

holds, then either $d_{\mathcal{S}}$ is a dual ascent direction, i.e,

$$D'(\mu; d_{\mathcal{S}}) > 0,$$

or there exists a balanced arc $(i, J) \in \mathcal{E}$ with either
(a) $i \in \mathcal{S}$, $J \cap \mathcal{S} = \emptyset$ and $x_{iJ} < c_{iJ}$ or
(b) $i \notin \mathcal{S}$, $J \cap \mathcal{S} \neq \emptyset$ and $x_{iJ} > b_{iJ}$.
    Proof: Follows from equation (17). □

C. Extended Relaxation Method

The extended relaxation method solves the dual problem iteratively. The method starts with a flow-price vector pair (x, μ) satisfying the complementary slackness condition and, maintaining this condition at all iterations, finally terminates with a pair $(\hat{x}, \hat{\mu})$, where $\hat{x}$ is feasible and, due to Proposition 1, $\hat{x}$ and $\hat{\mu}$ are primal and dual optimal, respectively.

At the beginning of each iteration an elementary initial set $\mathcal{S}$ of nodes consisting of one node with positive surplus is chosen. In general, due to Proposition 2 we have the following possibilities for an elementary $\mathcal{S}$ with positive total surplus:
(a) Dual ascent is possible, i.e., $d_{\mathcal{S}}$ defines a dual ascent direction. Then a price change is performed to improve the dual cost.
(b) Due to Proposition 2 an enlargement of $\mathcal{S}$ is possible such that $\mathcal{S}$ can be enlarged by adding a node with non-negative surplus and an elementary direction can be associated with the enlarged set, too.
(c) If dual ascent or an enlargement of $\mathcal{S}$ is not possible, then an *unblocked* path originating at some node of $\mathcal{S}$ with positive surplus can be constructed. Unblocked means that all arcs of the path allow a flow increase in the direction of the path. We refer to such a path as an *augmentation path*.

A flow increasing along an augmentation path can be used to change the surplus of the start node and end node, respectively. Such a flow does not influence the complementary slackness condition of the flow-price vector pair. In particular, a certain amount of flow can reduce the total *absolute* surplus of the network.

Unfortunately, an augmentation path which in general involves multi-arcs is more complicated for extended network problems than in the classical case. For example, a flow along a path originating at a node with positive surplus and ending at a node with negative surplus usually does not reduce the total surplus of the network. The Figures 3-6 illustrate typi-
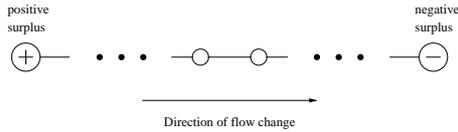
Fig. 3. Augmentation (A): Simple path
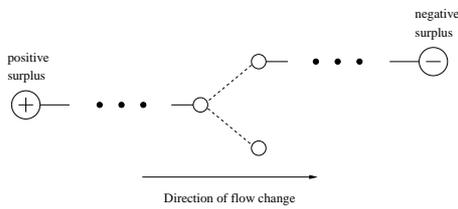
Fig. 4. Augmentation (B): Forward splitting
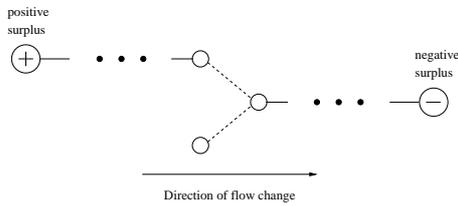
Fig. 5. Augmentation (C): Backward splitting
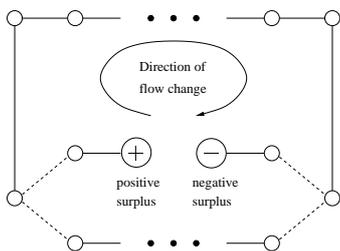
Fig. 6. Augmentation (D): Cycle path

cal augmentation paths that may occur in an extended network. However, all of these augmentation paths possess a common property. A certain amount of flow along such a path reduces a *relative* total surplus of the network. The relative total surplus represents a weighted sum of all surpluses (see [12] for details). It can be shown that if dual ascent or an elementary enlargement of $\mathcal{S}$ is not possible, a so-called *multi-augmentation path*

can be determined in such a way that a increasing flow along this multi-path reduces the total relative surplus of the network.

Hence, within each iteration of the extended relaxation method for solving the extended minimum cost flow problem either a price change with strict improvement of the dual cost or a flow adjustment that reduces the total relative surplus can be performed. Thus, we cannot have an infinite number of price changes and it is impossible to have an infinite number of flow augmentations between two successive price changes. The method terminates if no node i can be found with $g_i > 0$. In this case the current pair (x, p) is optimal if it holds $g_i = 0$ for all nodes i, implying x to be a feasible flow vector, otherwise the problem is infeasible. Thus, if the problem is feasible, the iteration will terminate with an optimal pair of flow, and price $(\hat{x}, \hat{p})$.

To search for an ascent direction comprises a recursive and path wise scanning of nodes. So the extended relaxation iteration contains several subroutines that are used recursively. For an efficient implementation special features should be used. For example, the value of dual ascent can be efficiently updated using a labeling scheme. Moreover, special data structures introduced for the classical relaxation iteration (cf. [3, Chapter 6]) can be adapted to the extended method, too.

IV. APPLICATION TO POWER MANAGEMENT

In this section we discuss the stochastic Lagrangian relaxation approach to power management in a hydro-thermal system under uncertainty (cf. [5], [11], [15]).

A. Stochastic Model

We consider a power utility that owns a generation system comprising thermal units and hydro storage plants and describe a model for the optimal generation and trading of electric power under uncertainty on the electrical load, market prices of fuel and electricity and stream flows to hydro reservoirs. Contracts for delivery and purchase are modeled as special thermal units. Let T denote the number of time intervals obtained from a discretization of the operation horizon. Let I and J be the number of thermal and hydro storage units in the system, respectively. The decision variables for the thermal units are the binary variables $\mathbf{u}_i^t$ for on/off decisions and the bounded variables $\mathbf{p}_i^t$ for the production levels of the thermal unit i during the time period t. The variables $\mathbf{v}_j^t$ and $\mathbf{w}_j^t$ denote the generation and pumping levels, respectively, of the (pumped) hydro storage plant j during the period t. Further, by $\mathbf{l}_j^t$ and $\mathbf{s}_j^t$ we denote the storage level in the upper reservoir and the stream flow (or supply) to the upper reservoir of plant j at the end of period t, respectively.

By $\xi = \{\xi^t = (\mathbf{d}^t, \mathbf{c}^t, \mathbf{s}^t)\}_{t=1}^T$ we denote the stochastic data process whose components are the electrical load $\mathbf{d}$, a vector $\mathbf{c}$ of relevant prices and the vector $\mathbf{s}$ of supplies. We assume that ξ^1 is deterministic and that ξ nonanticipative. The latter means that ξ^t does not depend on future realizations of ξ at any t. In case of a process having finitely many scenarios, these requirements lead to a tree structure of the process. A *scenario tree* may be represented be a finite number of nodes. It starts from the root node at period $t = 1$ and eventually branches into several nodes at the next period. The branching continues
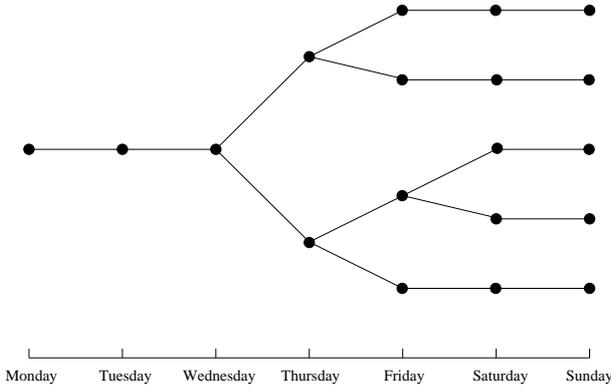
Fig. 7.   Example of a scenario tree

eventually up to nodes at the final time period. Figure 7 provides an example of a scenario tree on a weekly time horizon with nodes at each day and 5 scenarios. Let $\mathcal{N} = \{1, \ldots, N\}$ denote the finite set of nodes of the tree. The root node $n = 1$ stands for the period $t = 1$. Every other node n has a unique predecessor node n_- and a transition probability π_{n/n_-}, which describes the probability of n being a successor of n_-. The probability π_n of each node n is given recursively by $\pi_1 = 1$, and $\pi_n = \pi_{n/n_-} \pi_{n_-}$ for $n > 1$. By $\mathcal{N}_+(n)$ we denote the set of successors to node n and by $\mathrm{path}(n)$ the set $\{1, \ldots, n_-, n\}$ of nodes from the root to node n. Let $t(n)$ denote the number of its elements and $\mathcal{N}_t$ the set of all nodes n with $t(n) = t$. All nodes belonging to $\mathcal{N}_T$ are the leaves of the tree. A scenario corresponds to a path from the root to some leaf, i.e., to $\mathrm{path}(n)$ for some $n \in \mathcal{N}_T$.

Clearly, the decision process $\{(\mathbf{u}^t, \mathbf{p}^t, \mathbf{v}^t, \mathbf{w}^t)\}_{t=1}^T$ corresponding to the data process ξ in scenario tree form has the same tree structure. By ξ^n and (u^n, p^n, v^n, w^n) we denote the data and the decision, respectively, at node n. Then the scenario tree formulation of the stochastic power management model is of the form

$$\min \sum_{n \in \mathcal{N}} \pi_n \sum_{i=1}^I [C_i^n(p_i^n, u_i^n) + S_i^n(u_i)], \qquad (18)$$

subject to

$$\begin{aligned}
&u_i^n \in \{0, 1\}, \quad i = 1, \ldots, I, \, n \in \mathcal{N}, \\
&p_i^{\min} u_i^n \leq p_i^n \leq p_i^{\max} u_i^n, \quad i = 1, \ldots, I, \, n \in \mathcal{N}, \\
&0 \leq v_j^n \leq v_j^{\max}, \quad j = 1, \ldots, J, \, n \in \mathcal{N}, \\
&0 \leq w_j^n \leq w_j^{\max}, \quad j = 1, \ldots, J, \, n \in \mathcal{N}, \\
&0 \leq l_j^n \leq v_j^{\max}, \quad j = 1, \ldots, J, \, n \in \mathcal{N},
\end{aligned} \qquad (19)$$

$$\begin{aligned}
&u_i^{n-(\tau-1)} - u_i^{n-\tau} \leq 1 - u_i^n, \quad \tau = 1, \ldots, \underline{\tau}_i, \, n \in \mathcal{N}, \\
&u_i^{n-(\tau-1)} - u_i^{n-\tau} \leq u_i^n, \quad \tau = 1, \ldots, \overline{\tau}_i, \, n \in \mathcal{N},
\end{aligned} \qquad (20)$$

$$\begin{aligned}
&l_j^n = l_j^{n-} - v_j^n + \eta_j w_j^n + s_j^n, \quad n \in \mathcal{N}, \\
&l_j^0 = l_j^{in}, \quad j = 1, \ldots, J, \\
&l_j^n = l_j^{end}, \quad j = 1, \ldots, J, \, n \in \mathcal{N}_T;
\end{aligned} \qquad (21)$$

$$\sum_{i=1}^I p_i^n + \sum_{j=1}^J (v_j^n - w_j^n) \geq d^n, \quad n \in \mathcal{N}, \qquad (22)$$

$$\sum_{i=1}^I (p_i^{\max} u_i^n - p_i^n) \geq r^n, \quad n \in \mathcal{N}, \qquad (23)$$

where C_i^n is the piecewise linear convex cost function for operating unit or contract i at node n and S_i^n represents the piecewise constant start-up costs for getting unit i online at node n. The constraints (19) are operational constraints representing unit output limits and reservoir capacities, the inequality constraints (20) represent minimum up- and down-times for thermal units, (21) are dynamic constraints of storage levels of the hydro units, where $\eta_j \in (0, 1)$ denotes the pumping efficiency of unit j, (22) and (23) are load and (spinning) reserve constraints at all nodes n in $\mathcal{N}$.

B. Stochastic Lagrangian Relaxation

The approach by Lagrangian relaxation consists in dualizing the coupling constraints, i.e, the load and reserve constraints, by stochastic multipliers λ_1 and λ_2, and in solving the stochastic dual by some subgradient-type method. The dual optimum serves as a lower bound for the optimal cost and as a starting point for Lagrangian heuristics to determine a nearly optimal scheduling decision $(\bar{u}, \bar{p}, \bar{v}, \bar{w})$. The dual problem has the form

$$\max \{ D(\lambda) : \lambda = (\lambda_1, \lambda_2) \in \mathbb{R}_+^{2N} \},$$

where the dual function D is of the form

$$D(\lambda) = \sum_{i=1}^I D_i(\lambda) + \sum_{j=1}^J \hat{D}_j(\lambda_1) + \sum_{n \in \mathcal{N}} \pi_n (\lambda_1^n d^n + \lambda_2^n r^n)$$

and $D_i(\lambda)$ and $\hat{D}_j(\lambda_1)$ represent the optimal values of the stochastic thermal and hydro-storage subproblems, respectively (see e.g. [11], [15] for details). Hence, this procedure leads to a successive decomposition into stochastic single (thermal and hydro) unit subproblems.

C. Hydro-Storage Subproblems

Let us take a closer look at the Lagrangian hydro-storage subproblem for unit j in scenario tree form. It is of the form

$$\hat{D}_j(\lambda_1) = \min_{(v_j, w_j)} \left\{ \sum_{n \in \mathcal{N}} \pi_n \lambda_1^n (w_j^n - v_j^n) : 0 \leq v_j^n \leq v_j^{\max}, \right.$$

$$0 \leq w_j^n \leq w_j^{\max}, \, 0 \leq l_j^n \leq l_j^{\max}, \, l_j^n = l_j^{n-} - v_j^n + \eta_j w_j^n + s_j^n$$

$$\left. \text{for } n \in \mathcal{N}, \, l_j^0 = l_j^{in}, \, l_j^n = l_j^{end} \text{ for } n \in \mathcal{N}_T \right\}. \qquad (24)$$

The stochastic hydro-storage problem (24) can be formulated in terms of an extended network flow optimization problem, i.e., it corresponds to an extended linear minimum cost flow problem.

To obtain a network formulation with graph $(\mathcal{V}, \mathcal{E})$, we choose $\mathcal{V}$ such that it contains all nodes of $\mathcal{N}$ and a number of artificial nodes, which represent the (possibly also artificial) lower reservoir. If the tree would consist of only one scenario,

$\mathcal{V}$ contains the nodes of $\mathcal{N}$ and one artificial node. In general, we consider parts of scenarios that consist of nodes where no branching occurs. For each such part of a scenario we include one artificial node into $\mathcal{V}$. The set $\mathcal{E}$ contains the multi-arcs $(n, \mathcal{N}_+(n))$ for each $n \in \mathcal{N} \setminus \mathcal{N}_T$ and additional arcs to connect each node in $\mathcal{N}$ with the corresponding artificial node in both directions. In addition, we add to $\mathcal{E}$ all multi-arcs, which connect artificial nodes and have the same successor structure as the corresponding nodes in $\mathcal{N}$.

We associate to each element of $\mathcal{E}$ a component of the flow vector x. More precisely, we associate the storage level l_j^n to the arc $(n, \mathcal{N}_+(n))$ for $n \in \mathcal{N} \setminus \mathcal{N}_T$ and the variables v_j^n and $\eta_j w_j^n$ to the arcs connecting nodes in $\mathcal{N}$ with the corresponding artificial nodes. Furthermore, we introduce auxiliary components of the flow vector that correspond to arcs connecting artificial nodes. The capacities of all components coincide with the bounds in the operational constraints (19) and correspond to the reservoir capacities in case of the auxiliary components, respectively. For each node i in $\mathcal{V}$ we assign its supply s_i as follows. For the root node $n = 1$ we set $s_1 := s_j^1 + l_j^{\text{in}}$, $s_n := s_j^n - l_j^{\text{end}}$ for $n \in \mathcal{N}_T$, and $s_n := s_j^n$ for the remaining nodes in $\mathcal{N}$. For an artificial node i that corresponds to the nodes $n_1, \dots, n_k$ in $\mathcal{N}$, we set $s_i := \sum_{\nu=1}^{k}(l_{n_\nu} - s_{n_\nu})$, where l_n is set to $l_n := l_j^{\max}$ for the root node $n = 1$, $l_n := -l_j^{\max}$ for all leaves $n \in \mathcal{N}_T$ and $l_n := 0$ for all other nodes in $\mathcal{N}$. The cost coefficients for all components of the flow vector vanish except for those corresponding to the components v_j^n and $\eta_j w_j^n$ for $n \in \mathcal{N}$. The cost coefficients of v_j^n and $\eta_j w_j^n$ are $-\pi_n \lambda_1^n$ and $\frac{\pi_n \lambda_1^n}{\eta_j}$, respectively.

To explain this we want to consider an example. Let a small scenario tree consisting of four time periods, two stages and two scenarios be given (see Figure 8). Since there are two stages
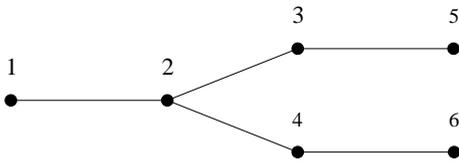
Fig. 8.   Tree structure of the example

and one branching point of degree two, there are three different parts of the given scenarios where no branching occurs. Hence, altogether we have to add three additional nodes to the given node set $\mathcal{N} = \{1, \dots, 6\}$. Figure 9 illustrates the correspond-
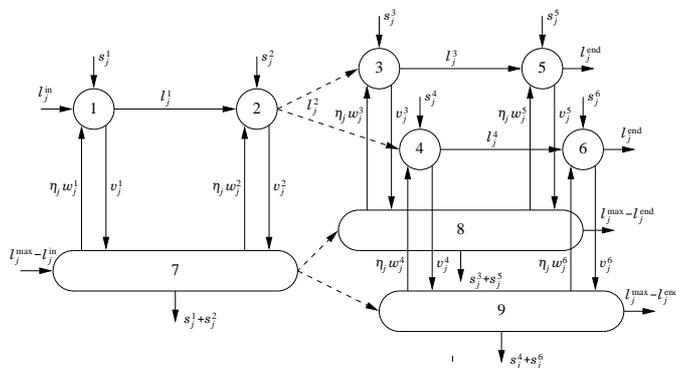
Fig. 9.   Example network for hydro-storage plant j

ing network of the example for one hydro-storage plant j. According to the branching structure of the scenario tree there are two multi-arcs involving each two end nodes (dashed lines).

V. NUMERICAL RESULTS

The extended relaxation method developed in Section III has been completely implemented in C. For testing the implementation we have randomly generated a bunch of hydro-storage test problems of the form (24). All test problems are based on realistic data. The stochastic data process has been modelled by binary scenario trees of varying dimensions ranging up to more than $130\,000$ scenarios. The test runs have been performed on a PC Pentium III with 700 MHz frequency and 128 MByte main memory under SuSE Linux 8.0.

TABLE I
NUMERICAL RESULTS FOR HYDRO SUBPROBLEMS

Scen	Nodes	Arcs	Time	Ascent	Augment
2048	8190	12284	0.49	3213	9002
2048	8190	12284	0.52	3199	9331
2048	8190	12284	0.55	3178	9722
4096	16382	24572	1.17	6275	15777
4096	16382	24572	1.22	6160	15930
4096	16382	24572	1.28	6085	16847
8192	32766	49148	3.38	16005	34062
8192	32766	49148	1.80	12801	33290
8192	32766	49148	1.90	13019	34135
16384	65534	98300	4.63	26509	71195
16384	65534	98300	4.20	26074	70852
16384	65534	98300	5.11	27245	72398
32768	131070	196604	12.32	58817	141076
32768	131070	196604	11.61	58207	142150
32768	131070	196604	19.60	71205	155335
65536	262142	393212	33.30	130478	267622
65536	262142	393212	34.01	129028	270890
65536	262142	393212	46.35	133098	274844
131072	524286	786428	87.77	279919	521973
131072	524286	786428	99.07	274566	534538
131072	524286	786428	86.66	276725	537894

Table I shows numerical results of the code DualAscent for a couple of test problems containing $2\,048$ up to $131\,072$ scenarios. The first three columns describe the problem size, i.e, the number of scenarios, the number of nodes, and the total number of arcs in the extended network. The last three columns report the computing time (in seconds), and the number of performed ascent and augmentation steps, respectively. The table shows that also very large network models containing more than $500\,000$ nodes can be solved in less than 100 seconds.

Furthermore, Figure 10 shows that the computing time for solving the test problems grows approximately linearly with respect to the number of scenarios.

The performance of DualAscent has been compared with the standard linear programming solver CPLEX 8.0. The results
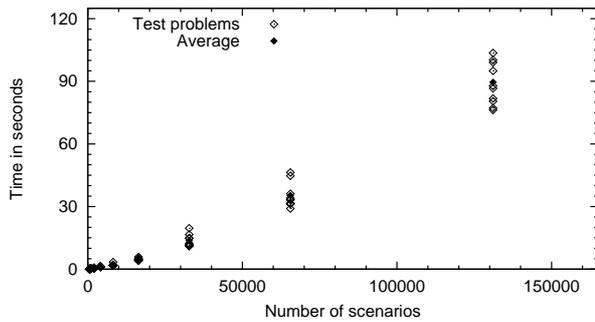
Fig. 10.   Computing time of DualAscent

TABLE II

COMPUTING TIME OF DUALASCENT (DA) COMPARED WITH CPLEX

Scen	CPLEX 8.0				DA
	primal	dual	baropt	netopt	
8192	18.07 s	11.42 s	18.49 s	10.37 s	1.64 s
8192	19.03 s	11.81 s	17.49 s	10.47 s	1.92 s
8192	19.59 s	11.38 s	18.53 s	9.54 s	1.56 s
32768	129.78 s	111.03 s	120.02 s	96.82 s	12.64 s
32768	114.84 s	118.21 s	92.83 s	103.21 s	14.40 s
32768	147.53 s	112.28 s	104.60 s	95.32 s	9.46 s
131072	881.0 s	1131.2 s	674.0 s	1088.4 s	89.6 s
131072	961.3 s	1160.7 s	820.0 s	1100.8 s	81.6 s
131072	902.2 s	1079.9 s	644.3 s	1013.1 s	63.2 s

displayed in Table II show that the extended relaxation method is the fastest algorithm for all test problems. It outperforms all of the tested CPLEX methods, i.e., the primal simplex method, the dual simplex method, the barrier method and the network optimization method.

VI. CONCLUSIONS

It has been shown that a stochastic extension of the relaxation method for solving the linear minimum cost flow problem can be applied to solve stochastic hydro-storage subproblems in power management. Because of the special structure of these problems network flow algorithms represent an efficient alternative to standard linear programming software. Our test runs show the promising performance of the new approach.

REFERENCES

[1] R.K. Ahuja, T.L. Magnanti and J.B. Orlin: Network flows - Theory, Algorithms and Applications, Prentice-Hall, Englewood Cliffs, New Jersey 1993.
[2] M.S. Bazaraa, J.J. Jarvis and H.D. Sherali: Linear programming and network flows, John Wily & Sons, 2nd Ed., N.Y. 1990.
[3] D.P. Bertsekas: Network optimization - Continuous and discrete models, Athena Scientific, Belmont, Massachusetts 1998.
[4] J.R. Birge and F. Louveaux: Introduction to Stochastic Programming, Springer, New York, 1997.
[5] D. Dentcheva and W. Römisch: Optimal power generation under uncertainty via stochastic programming, in: *Stochastic Programming Methods and Technical Applications* (K. Marti and P. Kall Eds.), Lecture Notes in Economics and Mathematical Systems Vol. 458, Springer, Berlin 1998, 22–56.
[6] L. Escudero, J.L. de la Fuente, C. Garcia and F.J. Prieto: Hydropower generation management under uncertainty via scenario analysis and parallel computation, *IEEE Transactions on Power Systems* 11 (1996), 683–689.
[7] S. Feltenmark and P.O. Lindberg: Network methods for head-dependent hydro-power scheduling, in: *Network Optimization* (P.M. Pardalos, D.W. Hearn and W.W. Hager Eds.), Lecture Notes in Economics and Mathematical Systems Vol. 450, Springer, Berlin 1997, 249–264.
[8] S.-E. Fleten, S.W. Wallace and W.T. Ziemba: Hedging electricity portfolios via stochastic programming, in: *Decision Making under Uncertainty: Energy and Power* (C. Greengard and A. Ruszczynski Eds.), IMA Volumes in Mathematics and its Applications Vol. 128, Springer, New York 2002, 71–93.
[9] L.R. Ford and D.R. Fulkerson: Flows in Networks, Princeton University Press, Princeton, NJ 1962.
[10] A. Gjelsvik, T.A. Rotting and J. Roynstrand: Long-term scheduling of hydro-thermal power systems, *Hydropower* '92, E. Broch, D.K. Lysne, N. Flatabo and E. Helland-Hansen, eds., Balkema, Rotterdam, 1992, pp. 539–546.
[11] N. Gröwe-Kuska, K.C. Kiwiel, M.P. Nowak, W. Römisch and I. Wegner: Power management in a hydro-thermal system under uncertainty by Lagrangian relaxation, in: *Decision Making under Uncertainty: Energy and Power* (C. Greengard and A. Ruszczynski Eds.), IMA Volumes in Mathematics and its Applications Vol. 128, Springer, New York 2002, 39–70.
[12] H. Heitsch: A relaxation method for network flow problems applied to hydro-storage subproblems in power generation, in preparation.
[13] H. Heitsch and W. Römisch: Scenario Reduction Algorithms in Stochastic Programming, *Computational Optimization and Applications* 24 (2003), 187–206.
[14] J. Jacobs, G. Freeman, J. Grygier, D. Morton, G. Schultz, K. Staschus and J. Stedinger: SOCRATES: A system for scheduling hydroelectric generation under uncertainty, *Annals of Operations Research* 59 (1995), 99–133.
[15] M.P. Nowak and W. Römisch: Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty, *Annals of Operations Research* 100 (2000), 251–272.
[16] M.V.F. Pereira: Optimal stochastic operation scheduling of large hydroelectric systems, *Electrical Power & Energy Systems* 11 (1989), 161–169.
[17] R.T. Rockafellar: Network Flows and Monotropic Programming, Wiley, New York, 1984.
[18] W. Römisch and R. Schultz: Multistage stochastic integer programs: An introduction, *Online Optimization of Large Scale Systems*, Springer-Verlag, Berlin 2001, 579–598.
[19] A. Ruszczyński and A. Shapiro (Eds.): *Stochastic Programming*. Handbooks in Operations Research and Management Science, Vol. 10, North-Holland, Amsterdam 2003 (to appear).
[20] T.A. Røtting and A. Gjelsvik: Stochastic dual programming for seasonal scheduling in the Norwegian power system, *IEEE Transactions on Power Systems* 7 (1992), 273–279.
[21] A. Schrijver: On the history of the transportation and maximum flow problem, *Mathematical Programming* 91 (2002), 437–445.

VII. BIOGRAPHIES

Holger Heitsch is a research assistant at the Institute of Mathematics of the Humboldt-University Berlin, Germany, where he graduated in 2001.
E-mail: heitsch@mathematik.hu-berlin.de
Web: www.mathematik.hu-berlin.de/~heitsch/
Werner Römisch is a Full Professor at the Institute of Mathematics of the Humboldt-University Berlin. His current research interests are in theory and solution methods for mixed-integer stochastic programming problems. He is actively working on several applications, especially in the electric power industry. He is co-editor of the *Stochastic Programming E-Print Series* (www.speps.info).
E-mail: romisch@mathematik.hu-berlin.de
Web: www.mathematik.hu-berlin.de/~romisch/