Working Paper

A Fast Descent Method for the Hydro Storage Subproblem in Power Generation

Matthias Peter Nowak

WP-96-109
September 1996

IIASA
International Institute for Applied Systems Analysis □ A-2361 Laxenburg □ Austria
Telephone: +43 2236 807 □ Fax: +43 2236 71313 □ E-Mail: info@iiasa.ac.at

A Fast Descent Method for the Hydro Storage Subproblem in Power Generation

Matthias Peter Nowak

WP-96-109
September 1996

IIASA International Institute for Applied Systems Analysis □ A-2361 Laxenburg □ Austria
Telephone: +43 2236 807 □ Fax: +43 2236 71313 □ E-Mail: info@iiasa.ac.at

Abstract

For many years energy optimization has dealt with large scale mixed integer linear programs. The paper concentrates on programs that are used for controlling an existing generation system consisting of thermal power units and pumped hydro storage plants, therefore they should be solved in real time. The problem can be decomposed into smaller problems using Lagrangian Relaxation. One of these problems is still a large scale multistage problem and it handles with pumped hydro storage plants only. In this paper, this problem is investigated down to the smallest details. The objective function for this problem is a linear function but stochastic. Using the special structure of the constraints, a solution method based on a subset of descent directions was developed. This method was compared with an available standard software for multistage linear programs.

Contents

A Fast Descent Method for the Hydro Storage Subproblem in Power Generation

Matthias Peter Nowak

1 Introduction

The main question of the nineties is not the Question of the Universe, Life and Everything [Ada80] but the question how to save the enviroment.

This paper deals with the optimization of energy systems. Optimization of such systems means reducing air pollution, saving fuel and last but not least saving money. The use of electric energy is very common today. Much is done by reducing the consumption of energy, but the production of energy can be optimized, too. Since the reduction of fuel cost means also reduction of air pollution, the minimization of fuel cost makes it easier to convince utility companies of the advantage of reducing air pollution. The energy optimization problem becomes more complicated, if the uncertainty of the demand is included and the problem has to be solved in real time. However, these assumptions make the problem more realistic in the case of short term planning.

Since the problem is a large scale one, it is necessary to develop an appropriate solution method, that takes advantage of the special structure of the problem.

2 Model

This paper deals with a system consisting of thermal power units and pumped hydro storage plants. The thermal power units are fired by coal, oil and gas. These units can be turn on and off. For getting a thermal power unit to work, one has to spend a certain amount of energy for heating boilers and turbines. These costs are called startup costs and they depend on the duration that unit has been turned off. In the case of coal fired units, these costs are as big as the production cost for 4 up to 9 hours. Because of these costs one has to consider the question whether it is better to replace a thermal unit by a pumped hydro storage plant or not. The pumped hydro storage plant considered in this paper have such a small income of water at the upper dam that the amount of water used for energy production has to be pumped uphill before the use. Therefore, the amount of water at the upper dam is measured in terms of energy. The restricted efficiency of pumped hydro storage plants is taken into account at the time of pumping. The amount of water at the upper dam is limited from below and above. The same is true for the pumping engines, the turbines and of course for the generators of the thermal power plants. Because of the startup costs of the thermal power units and because of the storage function of the pumped hydro storage plants, one cannot optimize the system for one time period only.

2.1 The deterministic model

The model described here is a slight modification of the model in [RS96]. In this paper, T always denotes the number of time periods. I denotes the number of thermal power units. The decisions dealing with thermal power units are described by binary variables u_t^i for ON/OFF-decision and by bounded real valued variables p_t^i for the amount of produced electricity. Such a pair (u_t^i, p_t^i) is assigned to the thermal power unit i and one time period t. $B^i(p_t^i, u_t^i)$ denotes the time independent fuel costs, while $A_t^i(u^i)$ denotes the startup costs. Furthermore, J is the number of pumped hydro storage plants, s_t^j is the produced amount of electricity of the plant j and w_t^j is the power used for pumping water uphill. As in [GRS92], it is assumed that there occur no generation costs for the pumped hydro storage plants. At this point the objective function can be formulated.

$$\min_{(u,p,s,w)} \sum_{i=1}^{I} \sum_{t=1}^{T} B^i(p_t^i, u_t^i) + A_t^i(u^i) \tag{1}$$

The fuel costs are zero, if the unit is turned off ($u_t^i = 0$). Otherwise, they are assumed to be convex with respect to p_t^i. The restrictions are the following. The power produced by thermal power units is bounded from below and above. For simplicity, they are zero, if the corresponding unit is turned off. If a thermal power unit is turned on, then this plant has to produce at least a certain amount of electricity.

$$\forall i = 1 \dots I, t = 1 \dots T : u_t^i p_{min}^i \le p_t^i \le u_t^i p_{max}^i \tag{2}$$

The operation mode of pumped hydro storage plants can be changed continuously from maximum pumping to maximum generating.

$$\forall j = 1 \dots J, t = 1 \dots T \quad : \quad 0 \le s_t^j \le s_{max}^j \tag{3}$$

$$0 \le w_t^j \le w_{max}^j \tag{4}$$

The amount of storaged energy is bounded, too. One can express this fact by using s_t^j and w_t^j only. However, in this paper a new variable L_t^j is introduced instead of only using s_t^j and w_t^j. This variable describes the water level at the upper dam in terms of energy. The introduction of this variable allows a markovian structure of the restrictions. This fact is a basic requirement of the algorithm presented in this paper. At the beginning of the operation cycle the water level at the upper dam is known.

$$\forall j = 1 \dots J : L_0^j = L_{in}^j \tag{5}$$

The water level at the upper dam changes due to generating energy and pumping water uphill. At this point, the restricted efficiency of pumped hydro storage plants is taken into account. The constant η^j describes the efficiency of the pumped hydro storage plant j.

$$\forall j = 1 \dots J, t = 1 \dots T : L_t^j = L_{t-1}^j - s_t^j + \eta^j w_t^j \tag{6}$$

At the end of the operation cycle the water level must have a certain value.

$$\forall j = 1 \dots J : L_T^j = L_{end}^j \tag{7}$$

And the water level is bounded.

$$\forall j = 1 \dots J, t = 1 \dots T : 0 \le L_t^j \le L_{max}^j \tag{8}$$

The produced power must meet the demand.

$$\forall t = 1 \dots T : \sum_{i=1}^{I} p_t^i + \sum_{j=1}^{J} (s_t^j - w_t^j) = d_t \tag{9}$$

However, the demand is not constant during one time period, therefore there has to be a certain reserve to fulfill a slightly changed demand.

$$\forall t = 1 \dots T : \sum_{i=1}^{I} (u_t^i p_{max}^i - p_t^i) \ge r_t \tag{10}$$

This is often called minute reserve. The opportunity of reducing the generated power is given by turning off thermal power plants, hence there are no extra restrictions.

This model is a mixed integer linear optimization problem. For a small number of units and a small number of time periods, this problem can be solved by standard software like *CPLEX*[CPL95]. However, an existing power system configuration is not small, the operation cycle comprises at least one week and the problem should be solved within 5 minutes because the solution is to be used for controlling that system. Another reason, which makes this problem difficult, is the uncertainty of the demand.

2.2 Description of the uncertain demand via scenario trees

First a few definitions will be given.

Let $(\Omega, \mathcal{A}, P)$ be a probability space, let $\mathbf{d} : \Omega \times \{1 \dots T\} \to \mathbb{R}$ be measurable with respect to $\mathcal{A}$, i.e. a random variable. For convenience, the function $\mathbf{d}(\omega, t)$ will be denoted by $\mathbf{d}_t(\omega)$. Let $\mathcal{F}_t$ be the smallest σ-algebra generated by $\{\mathbf{d}_s^{-1}(B) \mid B \in \mathcal{B}(\mathbb{R}),\ s \le t\}$, whereas $\mathcal{B}(\mathbb{R})$ denotes the Borel σ-algebra on $\mathbb{R}$. Then, $\mathcal{F} = (\mathcal{F}_t)_{t=1}^T$ is a filtration as defined in [Tay90], i.e.:

$$\forall s, t, 1 \le s \le t \le T \ : \ \mathcal{F}_s \subseteq \mathcal{F}_t$$

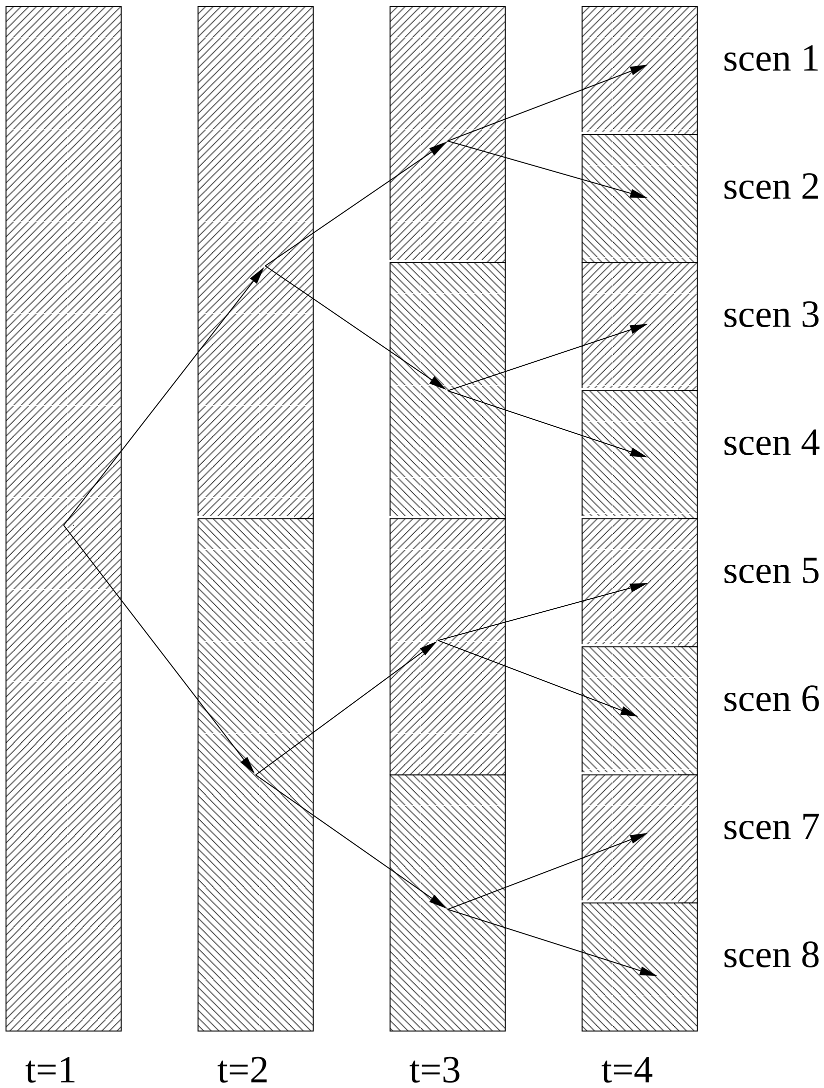


Figure 1: Relation between a scenario tree and the partitions

Under the assumption that the random variable has only finitely many possible values, one is able to compute numerically expectations and functions that are based on expectations. Therefore, in this paper it is assumed:

$$\forall t = 1 \dots T \ : \ \#\mathbf{d}_t(\Omega) < \infty \ . \tag{11}$$

Then, it follows:

$$\forall t = 1 \dots T \ : \ \#\mathcal{F}_t < \infty \ . \tag{12}$$

Finite σ-algebras and partitions are closely related terms, since the power set of the partition and the σ-algebra generated by the sets of a partition are equal. If a function is measurable with respect to a finite σ-algebra, then the function is constant on the sets of a certain partition. Let $[\omega]_t$ denote the equivalence class of ω at time t.

$$[\omega]_t := \{\upsilon \in \Omega \mid \forall s = 1 \dots t \ : \ \mathbf{d}_s(\upsilon) = \mathbf{d}_s(\omega)\}$$

The relation between the partitions and the σ-algebras is:

$$\{[\omega]_t\}_{\omega\in\Omega} = \{A \in \mathcal{F}_t \mid A \neq \emptyset, \forall B \in \mathcal{F}_t, \, B \subseteq A, \, B \neq \emptyset \Rightarrow A = B\} \ .$$

The demand of the first time period is known, therefore it yields:

$$\mathcal{F}_1 = \{\emptyset, \Omega\} \tag{13}$$

and that $\mathbf{d}_1(\Omega)$ is a singleton.

At the first time period, it is impossible to differentiate between several realisations of the random variable for the next time periods, therefore all elements of Ω are in the same set, i.e. at time $t = 1$ the partition consists of only one set Ω. This is to be seen in part $t = 1$ in figure 1. Each filled rectangle denotes a set of a partition. The number of the corresponding time period can be read below the rectangles. In figure 1 the scenario tree branches into 2 parts at each time period . For that reason there are 2 possibilities for the random variable at the time period $t = 2$ and therefore the partition for $t = 2$ consists of 2 sets.

Under the assumptions given above, a scenario tree can be defined:

Definition 1 **A scenario tree** *is a directed graph* $G = (V, E)$ *with:*

$$V \subseteq 2^{\Omega} \times I\!N \tag{14}$$
$$V = \{([\omega]_t, t) \mid \omega \in \Omega, t = 1 \dots T\} \tag{15}$$
$$E \subseteq V \times V \tag{16}$$
$$(([\omega]_s, s), ([v]_t, t)) \in E \Leftrightarrow [v]_t \subseteq [\omega]_s \wedge t = s + 1 \tag{17}$$

For the notion of a directed graph see [Jun94]. This graph is a tree because $[\omega]_1 = \Omega$. The root of the tree is denoted by k_0. Each $\omega \in \Omega$ corresponds to a path from the root k_0 to a leaf. Such a path is called scenario. Let $T(k)$ denote the reachability set, i.e. the set of all nodes, which can be reached from the node k.

$$T(k) := \{l \mid \exists K \in I\!N,\ K > 0,\ \exists \{l_j\}_{j=1}^{K},\ (k, l_1) \in E, \forall j = 1 \dots K-1\ (l_j, l_{j+1}) \in E,\ l_K = l\} \tag{18}$$

The operation cycle of pumped hydro storage plants usually consists of several days. Since the demand is not completely known for this time period, the stochasticity of the demand has to be taken into account. Assuming that the demand can have only finitely many possible values, one can model the stochasticity by a scenario tree, where each scenario describes a possible realization (with a given probability) of the demand. The known demand of the first time period is assigned to the root of the scenario tree. Because there is only one possibility, this node gets the probability 1. The nodes on the second stage of the scenario tree correspond to the possible realizations of the demand at the second time period. The same is true for all further stages of the scenario tree.

2.3 Stochastic Model

Since the demand is a random variable and the generated power has to meet the demand, the variables p_t^i, s_t^j and w_t^j have to be random, too. The operator knows the demand values of the previous and the current hour only, therefore all decisions he can make depend on these values. That means p_t^i, s_t^j and w_t^j are measurable with respect to $\mathcal{F}_t$, i.e. they are nonanticipative ([Wet89]). In this paper, it is assumed that thermal power units can be turned on immediately, which is not true in reality. The preparation time for turning on a thermal power unit depends on the type of the thermal power unit. This time may depend also on the duration the unit was turned off, but this makes the problem even harder to solve. Including preparation times means that u_t^i is measurable with respect to $\mathcal{F}_{t-\tau^i}$. It is also possible to assume that u_t^i does not depend on the random realization as assumed in [GRS95]. This leads to a deterministic plan for the ON/OFF-decision of the thermal power units. As mentioned above, here it is assumed that u_t^i are random but nonanticipative. For simplicity, stochastic variables are denoted by bold letters ($\mathbf{d}_t$) instead of denoting them by functions ($d_t(\omega)$). Then the stochastic version of the model reads:

$$\min_{(\mathbf{u},\mathbf{p},\mathbf{s},\mathbf{w})} \mathbf{E} \sum_{i=1}^{I} \sum_{t=1}^{T} B^i(\mathbf{p}_t^i, \mathbf{u}_t^i) + A_t^i(\mathbf{u}^i) \tag{19}$$

subject to:

$$\forall i = 1 \dots I, t = 1 \dots T : \mathbf{u}_t^i p_{min}^i \leq \mathbf{p}_t^i \leq \mathbf{u}_t^i p_{max}^i \tag{20}$$

$$\forall j = 1 \dots J, t = 1 \dots T \ : \ 0 \leq \mathbf{s}_t^j \leq s_{max}^j \tag{21}$$
$$0 \leq \mathbf{w}_t^j \leq w_{max}^j \tag{22}$$

$$\forall j = 1 \dots J : \mathbf{L}_0^j = L_{in}^j \tag{23}$$

$$\forall j = 1 \dots J, t = 1 \dots T : \mathbf{L}_t^j = \mathbf{L}_{t-1}^j - \mathbf{s}_t^j + \eta^j \mathbf{w}_t^j \tag{24}$$

$$\forall j = 1 \dots J : \mathbf{L}_T^j = L_{end}^j \tag{25}$$

$$\forall j = 1 \dots J, t = 1 \dots T : 0 \leq \mathbf{L}_t^j \leq L_{max}^j \tag{26}$$

$$\forall t = 1 \dots T : \sum_{i=1}^{I} \mathbf{p}_t^i + \sum_{j=1}^{J} (\mathbf{s}_t^j - \mathbf{w}_t^j) = \mathbf{d}_t \tag{27}$$

The requirement of a minute reserve may be superflous because one may include a scenario with a slighly increased demand instead.

$$\forall t = 1 \dots T : \sum_{i=1}^{I} (\mathbf{u}_t^i p_{max}^i - \mathbf{p}_t^i) \geq \mathbf{r}_t \tag{28}$$

This problem is a large scale mixed integer linear problem with a high dimension. One point that makes this problem difficult to solve, is the fact that there are joined restrictions for all units and plants. Because of these constraints, the ON/OFF-decisions of each thermal power unit depend on the ON/OFF-decisions of the others.

In the last 15 years, a dual approach using the Lagrangian Relaxation of the demand and of the minute reserve constraints is suggested. An overview of solution techniques is given in [SF94], where the authors came to the conclusion that a clear consensus is presently tending toward the Lagrangian Relaxation approach over other methodologies. Dentcheva/Römisch[DR96] present a detailed view of this problem and a discussion about some methods for solving the formulated problem.

3 Dual Approach

A dual approach using the Lagrangian Relaxation of the demand and of the minute reserve constraints is proposed for large generation systems and for large time horizons, because the relaxed problem can be split into smaller problems. For these smaller problems efficient fast solution techniques exist. For an increasing number of time periods and of thermal power units the duality gap becomes smaller ([BLSP83]). This may be a justification for using the dual approach.

The approach for the problem given above is described in [RS96]. Here, the contraints are changed by introduction of L_t^j variables.

The dual problem reads:

$$\max_{(\mu \geq 0, \lambda)} \min_{(\mathbf{u},\mathbf{p},\mathbf{s},\mathbf{w})} \mathbf{E} \sum_{t=1}^{T} \left\{ \sum_{i=1}^{I} B^i(\mathbf{p}_t^i, \mathbf{u}_t^i) + A_t^i(\mathbf{u}^i) \right. \tag{29}$$

$$+\lambda_t \left(\mathbf{d}_t - \sum_{i=1}^{I} \mathbf{p}_t^i - \sum_{j=1}^{J} (\mathbf{s}_t^j - \mathbf{w}_t^j) \right) \tag{30}$$

$$\left. +\mu_t \left(\mathbf{r}_t - \sum_{i=1}^{I} (\mathbf{u}_t^i \mathbf{p}_{max}^i - \mathbf{p}_t^i) \right) \right\} \tag{31}$$

subject to:

$$\forall i = 1 \dots I, t = 1 \dots T : \mathbf{u}_t^i p_{min}^i \leq \mathbf{p}_t^i \leq \mathbf{u}_t^i p_{max}^i \tag{32}$$

$$\forall j = 1 \dots J, t = 1 \dots T \quad : \quad 0 \leq \mathbf{s}_t^j \leq s_{max}^j \tag{33}$$

$$0 \leq \mathbf{w}_t^j \leq w_{max}^j \tag{34}$$

$$\forall j = 1 \dots J : \mathbf{L}_0^j = L_{in}^j \tag{35}$$

$$\forall j = 1 \dots J, t = 1 \dots T : \mathbf{L}_t^j = \mathbf{L}_{t-1}^j - \mathbf{s}_t^j + \eta^j \mathbf{w}_t^j \tag{36}$$

$$\forall j = 1 \dots J : \mathbf{L}_T^j = L_{end}^j \tag{37}$$

$$\forall j = 1 \dots J, t = 1 \dots T : 0 \leq \mathbf{L}_t^j \leq L_{max}^j \tag{38}$$

The dual variables μ_t and λ_t are are also measurable with respect to $\mathcal{F}_t$. The objective function can be rewritten in such a way, that the separability structure of the problem becomes more visible.

$$\max_{(\mu \geq 0, \lambda)} \left\{ \sum_{i=1}^{I} \min_{(\mathbf{u}^i, \mathbf{p}^i)} \mathbf{E} \sum_{t=1}^{T} \left[B^i(\mathbf{p}_t^i, \mathbf{u}_t^i) + A_t^i(\mathbf{u}^i) - \lambda_t \mathbf{p}_t^i - \mu_t(\mathbf{u}_t^i \mathbf{p}_{max}^i - \mathbf{p}_t^i) \right] \right. \quad (39)$$

$$+ \sum_{j=1}^{J} \min_{(\mathbf{s}^j, \mathbf{w}^j)} \mathbf{E} \sum_{t=1}^{T} \left[-\lambda_t \left(\mathbf{s}_t^j - \mathbf{w}_t^j \right) \right] \quad (40)$$

$$\left. + \mathbf{E} \sum_{t=1}^{T} \left[\lambda_t \mathbf{d}_t + \mu_t \mathbf{r}_t \right] \right\} \quad (41)$$

Obviously, the minimization problems are dealing with one unit or plant only and they can be solved separately. The part (41) is a simple calculation. The problems (39) might be solved by dynamic programming [NW88], but these (40) are still large scale linear programs. Therefore, these programs (40) were investigated down to the smallest details.

4 Structure of the problem dealing with only one pumped hydro storage plant

The problem dealing with the pumped hydro storage plant j is the following:

$$\min_{(\mathbf{s}^j, \mathbf{w}^j)} \mathbf{E} \sum_{t=1}^{T} -\lambda_t \left(\mathbf{s}_t^j - \mathbf{w}_t^j \right) \quad (42)$$

subject to:

$$\mathbf{L}_0^j = L_{in}^j \quad (43)$$

$$\forall t = 1 \dots T : \mathbf{L}_t^j = \mathbf{L}_{t-1}^j - \mathbf{s}_t^j + \eta^j \mathbf{w}_t^j \quad (44)$$

$$\mathbf{L}_T^j = L_{end}^j \quad (45)$$

$$\forall t = 1 \dots T \quad : \quad 0 \leq \mathbf{s}_t^j \leq s_{max}^j \quad (46)$$

$$0 \leq \mathbf{w}_t^j \leq w_{max}^j \quad (47)$$

$$\forall t = 1 \dots T : 0 \leq \mathbf{L}_t^j \leq L_{max}^j \quad (48)$$

The nonanticipativity of the variables $\mathbf{s}_t^j$, $\mathbf{w}_t^j$ and $\mathbf{L}_t^j$ can be forced in the following way. The scenario tree is built up of the realizations of the demand. Each node of this tree is linked with a set of a certain partition by the definition of a scenario tree. If a random variable should be measurable with respect to the filtration corresponding to that partition, then this variable is constant on all sets of that partition. Therefore, the possible values of the random variable can be assigned to the nodes belonging to the corresponding partition. That means the problem can be rewritten as a graph theoretical problem using the scenario tree instead of partitions and filtrations. Let k_0 denote the root of the scenario tree and $B^\star$ denote the set of its leaves, while $Succ(k)$ denotes the successors of node k.

$$\inf_{\{(s_k, w_k)\}_{k \in V}} \left\{ \sum_{k \in V} \lambda_k (s_k - w_k) p_k \right\} \quad (49)$$

$$\forall k \in V \quad : \quad 0 \leq s_k \leq s_{max} \quad (50)$$

$$0 \leq w_k \leq w_{max} \quad (51)$$

$$0 \leq L_k \leq L_{max} \quad (52)$$

$$\forall k \in V, \forall l \in Succ(k) \quad : \quad L_l = L_k - s_l + \eta w_l \quad (53)$$

$$L_{k_0} = L_{in} \quad (54)$$

$$\forall k \in B^\star \quad : \quad L_k = L_{lev} \quad (55)$$

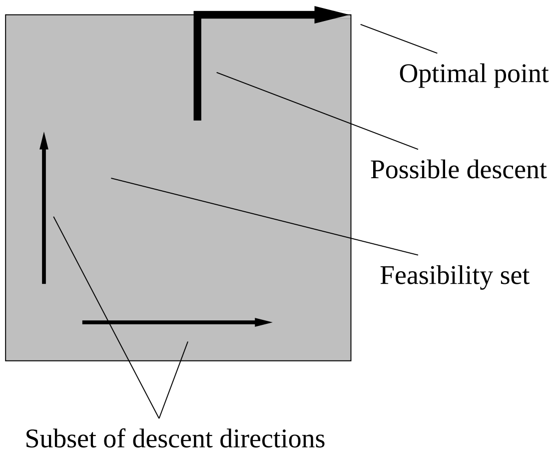


Figure 2: Box constraints and the subset of descent directions

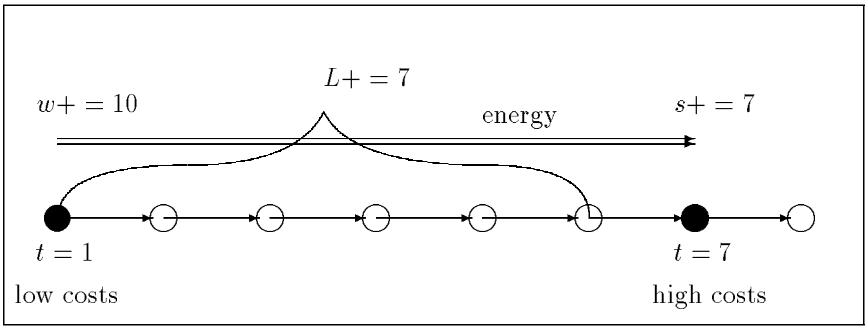


Figure 3: The deterministic case with $\eta = 0.7$

Constraints like (50)-(52) are called box constraints. That means each variable is bounded separately. If there are no other constraints, then it is possible to optimize such a system by optimizing the system with respect to each variable one by one.

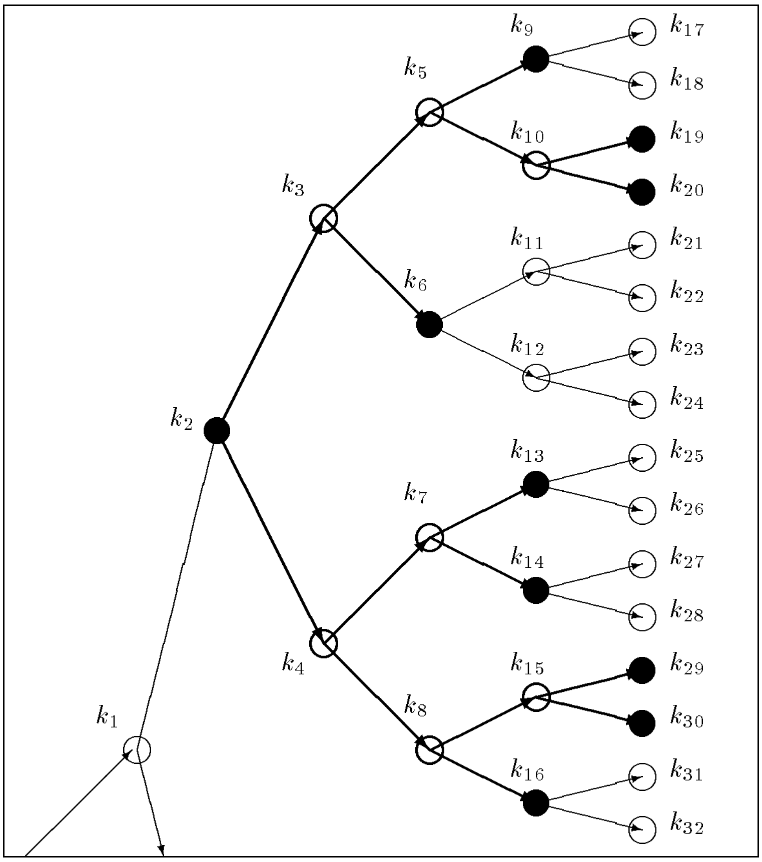
More generally, the idea is to consider a certain subset of descent directions instead of the set of all descent directions. It is necessary for such an approach that the subset is sufficiently large. If the set of all descent direction is not empty, then the subset has to comprise at least one element. This is true for problems with box constraints and in this case the subset of descent direction consists of the directions of the axis and there negatives. A two dimensional example is shown in figure 2.

The constraint (53) defines an intersection of the box of feasible points due to (50)-(52) with hyperplanes. The obtained set has still a geometrically regular structure. A sufficiently large subset of descent directions can be found using this structure.

The constraints (50)(51)(52)(54)(55) correspond to nodes, while the contraint (53) corresponds to an edge. These constraints characterize some capacity bounds for moving energy from one time interval to another. This problem is very closely related to network flow problems, because the deterministic version is a network flow problem, where each node can be a source or a sink. In the deterministic case there exists only one leaf.

In figure (3) one spends cheap energy at time $t = 1$ (that is the root of the considered subtree) for pumping water uphill, so that water can be used for generating energy at time $t = 7$ (the leaf of the subtree), when energy is more expensive. The amount of three units has been payed due to the efficiency.

The stochastic case is more difficult. In the stochastic example k_2 may denote the node with cheap energy. This cheap energy will be transported to several nodes in the future. There exists at least one node in each scenario, if that scenario includes k_2. Therefore, a certain subset of subtrees was considered. This sufficiently large subset of subtrees corresponds to a sufficiently large subset of descent directions.



k_2 is the root of the considered subtree. k_6, k_9, k_{13}, k_{14}, k_{16}, k_{19}, k_{20}, k_{29}, k_{30}, are leaves in the considered subtree. If a node is not a leaf in this tree, then all successors of that node in the scenario tree are nodes of this subtree.

Figure 4: An example of a scenario subtree

4.1 Sufficiency of the subset of descent directions

The sufficiency of the subset depends on the opportunity to reach an arbitrary feasible point from every feasible point using steps corresponding to subtrees (only) like in figure 4. The next proposition shows the existence of such a sequence of steps.

Proposition 1 *Let* $\{s_k, w_k\}_{k\in V}$ *and* $\{\hat{s}_k, \hat{w}_k\}_{k\in V}$ *be feasible points.*
$\Longrightarrow$ *Then, there exists a finite sequence of steps with:*

$$\exists n \in I\!N \cup \{0\}, \{\{\Delta s_k^l, \Delta w_k^l\}_{k\in V}\}_{l=1}^n \quad : \tag{56}$$

$$\forall k \in V \quad : \quad \hat{s}_k = s_k + \sum_{l=1}^{n} \Delta s_k^l \tag{57}$$

$$\forall k \in V \quad : \quad \hat{w}_k = w_k + \sum_{l=1}^{n} \Delta w_k^l \tag{58}$$

$$\tag{59}$$

All steps belong to the subset of directions:

$$\forall k \in V, \forall l = 1 \dots n \quad : \quad \Delta s_k^l \cdot \Delta w_k^l = 0 \tag{60}$$

$$\forall l = 1 \dots n, \exists! k \in V, \exists B \subset T(k) \quad : \tag{61}$$

$$\forall j \in V \setminus (\{k\} \cup B) \quad : \quad \eta \Delta w_j^l - \Delta s_j^l = 0 \tag{62}$$

$$B = B \setminus T(B) \tag{63}$$

$$\forall j \in T(k) \setminus (B \cup T(B)) \quad : \quad T(j) \cap B \neq \emptyset \tag{64}$$

$$\tag{65}$$

All steps satisfy the water balance equations:

$$\forall j \in B \quad : \quad \eta \Delta w_k^l - \Delta s_k^l + \eta \Delta w_j^l - \Delta s_j^l = 0 \tag{66}$$

Proof: *The proof of the finitesness works with the number of elements of following sets of nodes.*

$$N_s^+ = \{k \mid s_k - \hat{s}_k > 0\} \tag{67}$$
$$N_s^- = \{k \mid s_k - \hat{s}_k < 0\} \tag{68}$$
$$N_w^+ = \{k \mid w_k - \hat{w}_k > 0\} \tag{69}$$
$$N_w^- = \{k \mid w_k - \hat{w}_k < 0\} \tag{70}$$
$$N = N_s^+ \cup N_s^- \cup N_w^+ \cup N_w^- \tag{71}$$

The proof is done in a recursive manner. Therefore, the following number is introduced. $m(s, w, \hat{s}, \hat{w}) = \#N_s^+ + \#N_s^- + \#N_w^+ + \#N_w^-$. *If* $m = 0$ *is true, then let* $n = 0$*, the conditions are obviously satisfied. When* m *is positive, then* $N \setminus T(N)$ *comprises at least one element. Let* $k \in N \setminus T(N)$ *be arbitrarily selected.*

Now 2 cases are distinguished.

1. $\eta(w_k - \hat{w}_k) - (s_k - \hat{s}_k) < 0$
2. $\eta(w_k - \hat{w}_k) - (s_k - \hat{s}_k) > 0$

Since $k \in N$ *the case* $\eta(w_k - \hat{w}_k) - (s_k - \hat{s}_k) = 0$ *is impossible. Both cases can be processed in the same way therefore only the first case will be considered. In that case it follows* $k \in N_w^- \cup N_s^+$. *The leaves* (B) *of the subtree have to be elements of the other set due to the water balance equations, i.e.:*

$$\hat{N} = T(k) \cap N_w^+ \cap N_s^- \tag{72}$$

$$B = \hat{N} \setminus T(\hat{N}) \tag{73}$$

The step length is a positive number, computed as follows:

$$d_1 = \min_{\{k\} \cap N_w^-} \eta(\hat{w}_l - w_l) \tag{74}$$
$$d_2 = \min_{\{k\} \cap N_s^+} (s_l - \hat{s}_l) \tag{75}$$
$$d_3 = \min_{B \cap N_w^+} \eta(w_l - \hat{w}_l) \tag{76}$$
$$d_4 = \min_{B \cap N_s^-} (\hat{s}_l - s_l) \tag{77}$$
$$d = \min\{d_1, d_2, d_3, d_4\} \tag{78}$$

Because of the water balance equations there is no scenario comprising node k *but not comprising any node of* B*. Now the step can be constructed. The variables* Δs_k *und* Δw_k *of nodes that are not mentioned here are zero.*

$$\Delta s_k^1 = \begin{cases} -d & s_k > 0 \\ 0 & s_k = 0 \end{cases} \tag{79}$$

$$\Delta w_k^1 = \begin{cases} \frac{d}{\eta} & s_k = 0 \\ 0 & s_k > 0 \end{cases} \tag{80}$$

$$\forall l \in B\ \Delta s_l^1 = \begin{cases} d & w_l = 0 \\ 0 & w_k > 0 \end{cases} \tag{81}$$

$$\forall l \in B\ \Delta w_l^1 = \begin{cases} -\frac{d}{\eta} & w_l > 0 \\ 0 & w_l = 0 \end{cases} \tag{82}$$

The variables $(\Delta s^1, \Delta w^1)$ *satisfy the constraints (60) - (66). Let* $\bar{s}_j^1 = s_j + \Delta s_j^1$, $\bar{w}_j^1 = w_j + \Delta w_j^1$. *The point* $(\bar{s}^1, \bar{w}^1)$ *is feasible since:*

$$\forall j \in V : \bar{s}_j^1 \in conv(s_j, \hat{s}_j), \bar{w}_j^1 \in conv(w_j, \hat{w}_j) \tag{83}$$

The water balance equations are satisfied due to (66).

The same procedure is applied repeatedly. One gets a sequence of $(\Delta s^l, \Delta w^l)$ *and* $(\bar{s^l}, \bar{w^l})$*. Since* $m(\bar{s}, \bar{w}, \hat{s}, \hat{w}) \leq m(s, w, \hat{s}, \hat{w}) - 1$ *the method stops after finitely many steps.* #

The sequence of steps constructed in proposition 1 is further investigated. What happens if $(\Delta s^1, \Delta w^1)$ is omitted?

Proposition 2 *Let* $\{s_k, w_k\}_{k\in V}$ *and* $\{\hat{s}_k, \hat{w}_k\}_{k\in V}$ *be feasible points , the sequence* $\{(\Delta s^l, \Delta w^l)\}_{l=1}^n$ *is constructed as in proposition 1.*

$$\forall j \in V \quad : \quad \bar{s}_j = s_j + \sum_{l=2}^{n} \Delta s_j^l \tag{84}$$

$$\bar{w}_j = w_j + \sum_{l=2}^{n} \Delta w_j^l \tag{85}$$

$\Longrightarrow$ *The point* $\{\bar{s}, \bar{w}\}_{k\in V}$ *is feasible.*
Proof: *Since* $\{s_k, w_k\}_{k\in V}$ *was a feasible point and the sequence* $(\Delta^j s, \Delta^j w)$ *constructed in proposition 1 satisfy the equation (66), the point* $(\bar{s}, \bar{w})$ *also satisfies the water balance equations. Only the box constraints are left to be checked.*

Let k *denote the root and* B *denote the leaves of the subtree corresponding to* $\{\Delta^1 s, \Delta^1 w\}_{k\in V}$. *The following sets of nodes are introduced:*

$$V_k = \{k\} \cup T(k) \setminus T(B) \tag{86}$$

$$T_k = V \setminus V_k \tag{87}$$

V_k *denotes the sets of all nodes of the subtree and* T_k *denote the complementary set. Then, it yields:*

$$\forall j \in T_k \quad : \quad \bar{s}_j = \hat{s}_j \tag{88}$$

$$\bar{w}_j = \hat{w}_j \tag{89}$$

The investigations of the values of the nodes in V_k *remains. The definitions of the node sets* N_s^+, N_s^-, N_w^+, N_w^- *are as in proposition 1.*

$$N_s^+ = \{k \,|\, s_k - \hat{s}_k > 0\} \tag{90}$$

$$N_s^- = \{k \,|\, s_k - \hat{s}_k < 0\} \tag{91}$$

$$N_w^+ = \{k \,|\, w_k - \hat{w}_k > 0\} \tag{92}$$

$$N_w^- = \{k \,|\, w_k - \hat{w}_k < 0\} \tag{93}$$

$$N = N_s^+ \cup N_s^- \cup N_w^+ \cup N_w^- \tag{94}$$

At this point, only the case $k \in N_s^+$ *is considered, since all other cases can be proved in a similar way. In this case, more energy will be generated at node* k *of the point* $\{s_j, w_j\}_{j\in V}$ *as at the same node of the points* $\{\hat{s}_j, \hat{w}_j\}_{j\in V}$ *and* $\{\bar{s}_j, \bar{w}_j\}_{j\in V}$.

Because of (83), the variables $\bar{s}$ *and* $\bar{w}$ *satisfy the box constraints.*

Now, the fulfillment of the constraint (52) will be proved. The corresponding L *variables for the points* $\{\bar{s}, \bar{w}\}_{j\in V}$ *are denoted by* $\bar{L}$. *Having in mind the definition of* B *one gets:*

$$(N_s^- \cup N_w^+) \cap (T(k) \setminus T(B)) = B \tag{95}$$

Therefore:

$$\forall l \in T(k) \setminus (T(B) \cup B) \,:\, L_l \leq \bar{L}_l \leq \hat{L}_l \tag{96}$$

Since the points $\{s_j, w_j\}_{j\in V}$ *und* $\{\hat{s}_j, \hat{w}_j\}_{j\in V}$ *are feasible, the variables* $\bar{L}$ *are also feasible and the points* $\{\bar{s}, \bar{w}\}_{j\in V}$ *are feasible, too.* #

If one applies the proposition 2 repeatedly, it follows that all points $\{\bar{s}_j^k, \bar{w}_j^k\}_{j\in V}$

$$\forall j \in V \quad : \quad \bar{s}_j^k = s_j + \sum_{l=k}^{n} \Delta s_j^l \tag{97}$$

$$\bar{w}_j^k = w_j + \sum_{l=k}^{n} \Delta w_j^l \tag{98}$$

are feasible, too.

Proposition 3 *Let* $\{s_k, w_k\}_{k\in V}$ *and* $\{\hat{s}_k, \hat{w}_k\}_{k\in V}$ *be feasible points.* $\{\hat{s}_k, \hat{w}_k\}_{k\in V}$ *is an optimal point. The sequence* $\{(\Delta s^l, \Delta w^l)\}_{l=1}^n$ *is constructed as in proposition 1.*
$\Longrightarrow$ *The first step is a descent step.*

$$\sum_{k\in V} \lambda_k (\Delta s_k^1 - \Delta w_k^1) P_k \leq 0 \tag{99}$$

Proof: *From the construction of the sequence, it follows:*

$$\sum_{k\in V} \lambda_k(\hat{s}_k - \hat{w}_k)P_k = \sum_{k\in V} \lambda_k(s_k - w_k)P_k + \sum_{l=1}^{n}\sum_{k\in V} \lambda_k(\Delta s_k^l - \Delta w_k^l)P_k \tag{100}$$

This can be expressed using $\{\bar{s}_k, \bar{w}_k\}_{k\in V}$*, which are defined as in proposition 2.*

$$\sum_{k\in V} \lambda_k(\hat{s}_k - \hat{w}_k)P_k = \sum_{k\in V} \lambda_k(\Delta s_k^1 - \Delta w_k^1)P_k + \sum_{k\in V} \lambda_k(\bar{s}_k - \bar{w}_k)P_k \tag{101}$$

Since $\{\bar{s}_k, \bar{w}_k\}_{k\in V}$ *are feasible and* $\{\hat{s}_k, \hat{w}_k\}_{k\in V}$ *is optimal, it follows:*

$$\sum_{k\in V} \lambda_k(\bar{s}_k - \bar{w}_k)P_k \geq \sum_{k\in V} \lambda_k(\hat{s}_k - \hat{w}_k)P_k \tag{102}$$

Therefore:

$$\sum_{k\in V} \lambda_k(\Delta s_k^1 - \Delta w_k^1)P_k \leq 0 \tag{103}$$

#

The points $\{\Delta s_k, \Delta w_k\}_{k\in V}$ satisfying strictly the inequality (103) are descent steps. Starting at a feasible point one can look for descent directions.

4.2 Conditions for descent directions

Next, a fixed arbitrary sufficient subset of nodes of the scenario tree is considered. The current iteration is l. There are several cases for each node:

1. pumping more $w^{l+1} = w^l + d$, if $s^l = 0$
2. generating less $s^{l+1} = s^l - \eta d$, if $s^l > 0$
3. generating more $s^{l+1} = s^l + \eta d$, if $w^l = 0$
4. pumping less $w^{l+1} = w^l - d$, if $w^l > 0$

The first two cases are valid when energy is stored, i.e. water is pumped uphill, the two other are valid when pumped water is used for generating energy. Let B denote the set of leaves, k_1 denote the root. F_i denotes the set of nodes for which case i is valid. δ_{F_i} is the characteristic function of the set F_i.

The following inequalities guarantee a descrease of the objective function. The left sides of the inequalities denote the slope of the objective function with respect to the considered direction.

The condition for storing energy before using energy:

$$-\lambda_{t_1}(\omega_{k_1}))P([\omega_{k_1}])\,(\delta_{F_1}(k_1) + \eta\delta_{F_2}(k_1)) + \sum_{k\in B} \lambda_{t_k}(\omega_k)P([\omega_k])\,(\delta_{F_4}(k) + \eta\delta_{F_3}(k)) < 0 \tag{104}$$

This set of descent directions satisfying (104) will be denoted by $A_{\uparrow\downarrow}$. This means a flow of energy forward in time.

The condition for using energy before storing energy:

$$+\lambda_{t_1}(\omega_{k_1})P([\omega_{k_1}])\,(\delta_{F_4}(k_1) + \eta\delta_{F_3}(k_1)) - \sum_{k\in B} \lambda_{t_k}(\omega_k)P([\omega_k])\,(\delta_{F_1}(k) + \eta\delta_{F_2}(k)) < 0 \tag{105}$$

This set of descent directions satisfying (105) will be denoted by $A_{\downarrow\uparrow}$. This means a flow of energy backward in time.

An example: Pumping at node t_{k_1} and generating at nodes $\{t_k\}_{k\in B}$:

$$-\lambda_{k_1}P([\omega_{k_1}]) + \eta\sum_{k\in B} \lambda_{t_k}(\omega_k)P([\omega_k]) < 0 \tag{106}$$

With the conditions (104) and (105), one can decide to pump or generate.

The restrictions for the amount of energy transported from k_1 to $\{t_k\}_{k\in B}$ or conversely, i.e. the step length in the descent algorithm are given by the following inequalities:

$$\begin{aligned} \forall \omega \in \Omega,\ t = 1\ldots T:\quad 0 &\le L_t^{l+1}(\omega) \le L_{max} && (107)\\ 0 &\le w_t^{l+1}(\omega) \le w_{max} && (108)\\ 0 &\le s_t^{l+1}(\omega) \le s_{max} && (109)\end{aligned}$$

It makes sense to consider subtrees with $d_{max} > 0$ only.
The conditions for $A_{\uparrow\downarrow}$:

$$\begin{aligned} \forall k \in V\setminus B &\quad \eta d \le L_{max} - L_k^l && (110)\\ \forall k \in W &\quad w_k^l + \delta_{F_1}(k)d \le w_{max} && (111)\\ &\quad s_k^l - \eta\delta_{F_2}(k)d \ge 0 && (112)\\ \forall k \in B &\quad s_k^l + \eta\delta_{F_3}(k)d \le s_{max} && (113)\\ &\quad w_k^l - \delta_{F_4}(k)d \ge 0 && (114)\end{aligned}$$

The conditions for $A_{\downarrow\uparrow}$:

$$\begin{aligned} \forall k \in V\setminus B &\quad \eta d \le L_k^l && (115)\\ \forall k \in W &\quad w_k^l - \delta_{F_4}(k)d \ge 0 && (116)\\ &\quad s_k^l + \eta\delta_{F_3}(k)d \le s_{max} && (117)\\ \forall k \in B &\quad s_k^l - \eta\delta_{F_2}(k)d \ge 0 && (118)\\ &\quad w_k^l + \delta_{F_1}(k)d \le w_{max} && (119)\end{aligned}$$

4.3 The stochastic network flow algorithm

All subtrees considered above have exactly one root, auxilary variables are connected with them. The following variables take part in the calculation of the slope of the objective function. The decrease is the product of a weighted sum of some auxilary variables and the step length.

$$r_k^{up} = \begin{cases} \lambda_k(\omega_k)P([\omega_k]) & \text{if } s_k^l = 0\\ \lambda_k(\omega_k)P([\omega_k])\eta & \text{if } s_k^l > 0\end{cases} \tag{120}$$

$$r_k^{down} = \begin{cases} \lambda_k(\omega_k)P([\omega_k]) & \text{if } w_k^l > 0\\ \lambda_k(\omega_k)P([\omega_k])\eta & \text{if } w_k^l = 0\end{cases} \tag{121}$$

These variables express an upper bound for the step length with respect to the pumps or turbines of that node.

$$d_k^{down} = \begin{cases} \frac{s_{max}-s_k^l}{\eta} & \text{if } w_k^l = 0\\ w_k^l & \text{if } w_k^l > 0\end{cases} \tag{122}$$

$$d_k^{up} = \begin{cases} \frac{s_k^l}{\eta} & \text{if } s_k^l > 0\\ w_{max} - w_k^l & \text{if } s_k^l = 0\end{cases} \tag{123}$$

If the objective function is a piecewise linear function, the different slopes influence the variables r_k^{up} and r_k^{down}. The distance to the next segment of the piecewise linear objective function reduces the step lengths d_k^{up} and d_k^{down}. Binary variables b_k^{up} und b_k^{down} are introduced for each node . $b_k^{down} = 1$ means that node k is a leaf in the set $A_{\uparrow\downarrow}$. Let $Succ(k)$ denote the set of successors. The first two variables denote the weighted sums of the corresponding values of the leaves of that subtree starting at node k, while the other two denote the minima of the corresponding step lengths.

$$\hat{r}_k^{up} = \begin{cases} r_k^{up} & \text{if } b_k^{up} = 1\\ \sum_{l\in Succ(k)} \hat{r}_l^{up} & \text{if } b_k^{up} = 0\end{cases} \tag{124}$$

$$\hat{r}_k^{down} = \begin{cases} r_k^{down} & \text{if } b_k^{down} = 1\\ \sum_{l\in Succ(k)} \hat{r}_l^{down} & \text{if } b_k^{down} = 0\end{cases} \tag{125}$$

$$\hat{d}_k^{up} = \begin{cases} d_k^{up} & \text{if } b_k^{up} = 1\\ \min\{\frac{L_k}{\eta}, \min_{l\in Succ(k)} \hat{d}_l^{up}\} & \text{if } b_k^{up} = 0\end{cases} \tag{126}$$

$$\hat{d}_k^{down} = \begin{cases} d_k^{down} & \text{if } b_k^{down} = 1 \\ \min\{\frac{L_{max}-L_k}{\eta}, \min_{l\in Succ(k)} \hat{d}_l^{down}\} & \text{if } b_k^{down} = 0 \end{cases} \quad (127)$$

Using these variables, one can write the conditions for descent directions in the following manner:

- Case $A_{\uparrow\downarrow}$:
$$\left(-r_k^{up} + \hat{r}_k^{down}\right) \min\left\{d_k^{up}, \hat{d}_k^{down}\right\} < 0 \quad (128)$$
- Case $A_{\downarrow\uparrow}$:
$$\left(r_k^{down} - \hat{r}_k^{up}\right) \min\left\{d_k^{down}, \hat{d}_k^{up}\right\} < 0 \quad (129)$$

If b_k^{up} and b_k^{up} are set correct setting for b_k^{up} and b_k^{up} without paying attention to all subtrees. The decision for one node can be made by considering the values of the following nodes only.

4.4 Steps of the stochastic network flow algorithm

1. **Reading of Input**
   - global constants: η, L_{max}, s_{max}, w_{max}, L_{in}, L_{end}
   - probabilities: $P([\omega_k])$
   - dual variables (here: -cost gradients) $\lambda_k(\omega)$
2. **Computing of a feasible solution** (w, s)
3. **Computing of auxiliary variables**
   - L, r_k^{up}, r_k^{down}, d_k^{up}, d_k^{down}
   - Let $b_k^{up} = 1$, if
   $$\left(\min\{L_k, \min_{l\in Succ(k)} \hat{d}_l^{up}\} = 0\right) \vee (d_k^{up} > 0) \wedge \left(r_k^{up} \geq \sum_{l\in Succ(k)} \hat{r}_l^{up}\right)$$
   else $b_k^{up} = 0$.
   - Let $b_k^{down} = 1$, if
   $$\left(\min\{L_{max} - L_k, \min_{l\in Succ(k)} \hat{d}_l^{down}\} = 0\right) \vee \left(d_k^{down} > 0\right) \wedge \left(r_k^{down} \leq \sum_{l\in Succ(k)} \hat{r}_l^{down}\right)$$
   else $b_k^{down} = 0$.
   - $\hat{r}_k^{up}$, $\hat{r}_k^{down}$, $\hat{d}_k^{up}$, $\hat{d}_k^{down}$
   - Since a scenario tree is considered, all variables b_k^{up} und b_k^{down} are well defined.
4. **Choice** of a descent direction
   Determination of $\min\{(\hat{r}_k^{down} - r_k^{up}), (r_k^{down} - \hat{r}_k^{up})\}$. Node k, yielding the minimum, becomes the root of the considered subtree.
   If $\min_k\{(\hat{r}_k^{down} - r_k^{up}), (r_k^{down} - \hat{r}_k^{up})\} \geq 0$, then **STOP**
5. **Update**
   - for root k and leaves $B(k)$: s, w, r und d
   Case $A_{\uparrow\downarrow}$:

$$\begin{aligned} d &= \min\{d_k^{up}, \hat{d}_k^{down}\} && (130)\\ s_k &:= s_k - \eta d, \text{ if } s_k > 0 && (131)\\ w_k &:= w_k + d, \text{ if } s_k = 0 && (132)\\ \forall l \in B(k) & && (133)\\ s_l &:= s_l + \eta d, \text{ if } w_l = 0 && (134)\\ w_l &:= w_l - d, \text{ if } w_l > 0 && (135)\\ & && (136) \end{aligned}$$

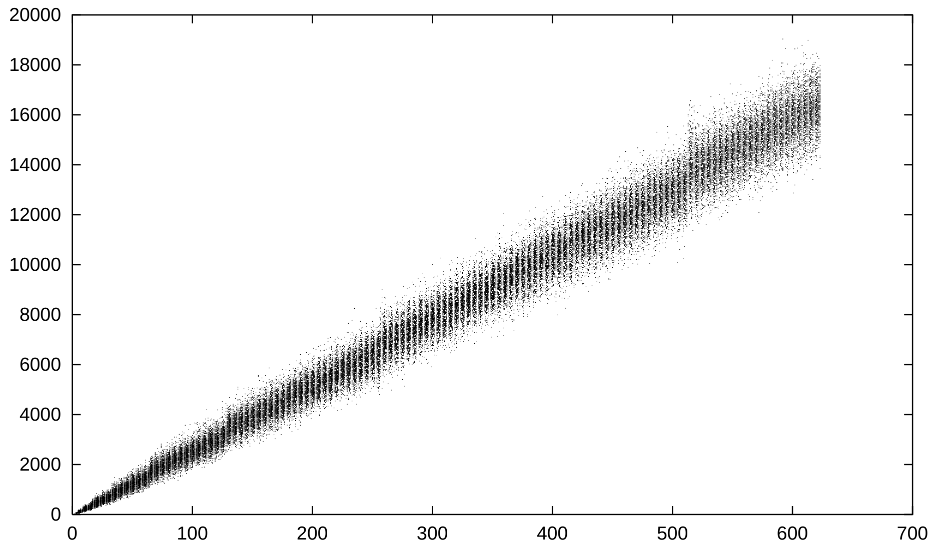


Figure 5: The relation between the number of iterations of MSLiP and the number of scenarios

Case $A_{\downarrow\uparrow}$:

$$
\begin{aligned}
d &= \min\{d_k^{down}, \hat{d}_k^{up}\} && (137)\\
s_k &:= s_k + \eta d, \text{ if } w_k = 0 && (138)\\
w_k &:= w_k - d, \text{ if } w_k > 0 && (139)\\
\forall l \in B(k) & && (140)\\
s_l &:= s_l - \eta d, \text{ if } s_l > 0 && (141)\\
w_l &:= w_l + d, \text{ if } s_l = 0 && (142)\\
& && (143)
\end{aligned}
$$

6. GOTO 3

4.5 Comparisons of MSLiP with StochTrans

StochTrans is the implementation of the algorithm described in section 4.3 in *C++*. All problems were randomly generated and each problem was solved by MSLiP and StochTrans.

The generation of the scenario trees has some specifics. In order to generate trees with a number of scenarios other than 2^i, a part of binary trees was cutted.

The program MSLiP [Gas90] is the version "MSLiP 8.3, version of April 7, 1995." [Gas90]. This is a general purpose program for solving MultiStage Linear Programs using the L-shaped Method [Bir85]. Therefore, the comparison is made with respect to the question, how much is the advantage of using an adapted algorithm instead of a general purpose algorithm. MSLiP consider nonstochastic objective functions only, thus additional variables were introduced.

The examples were computed on a HP-apollo Workstation Model 715/75 with HP-UX 9.03 and 64 Mbyte Memory. Figure 5 shows, how many iterations MSLiP needs. The implemention of the L-shaped Method ([vSW67]) is very tricky. Subproblems are solved by a modified simplex method. Pivot steps for different subproblems at the same stage are done together. Thus the code MSLiP is very fast, as shown in figure 6.

In figure 7, the number of iterations done by StochTrans is small, but the amount of computations for each iteration done by StochTrans is greater than the amount done by MSLiP.

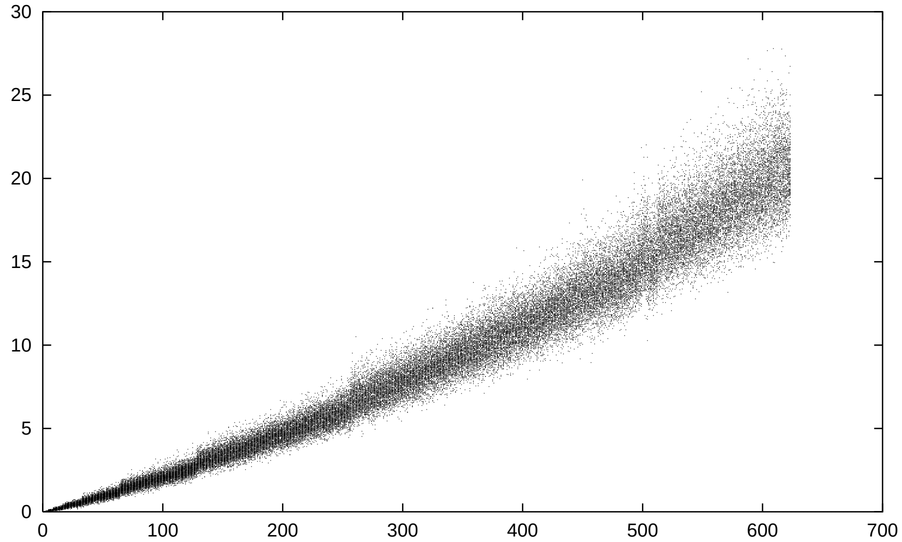


Figure 6: The relation between the computing time of MSLiP in seconds and the number of scenarios

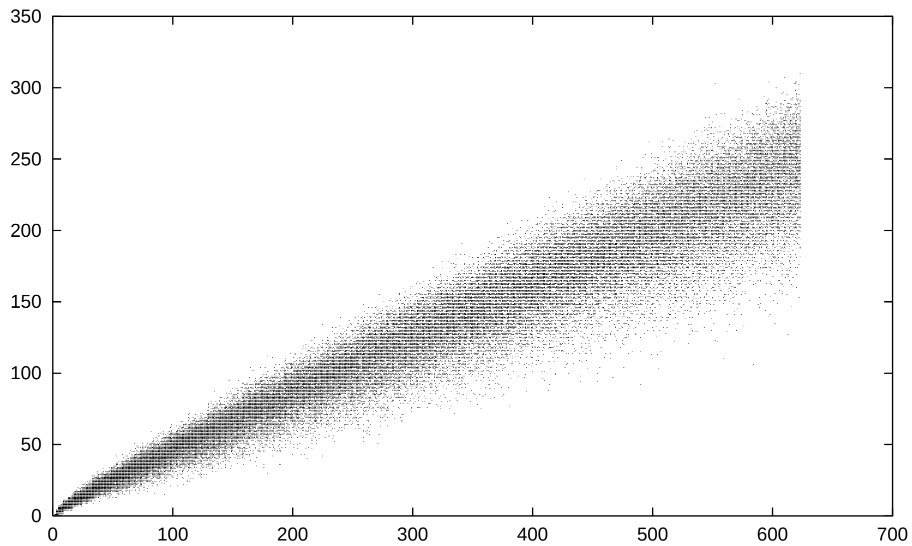


Figure 7: The relation between the number of iterations of StochTrans and the number of scenarios

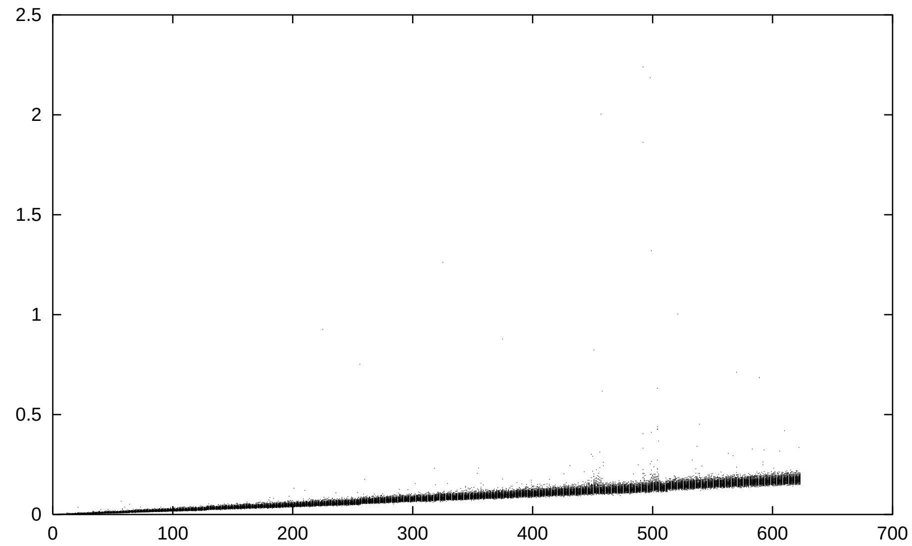


Figure 8: The relation between the computing time of StochTrans in seconds and the number of scenarios

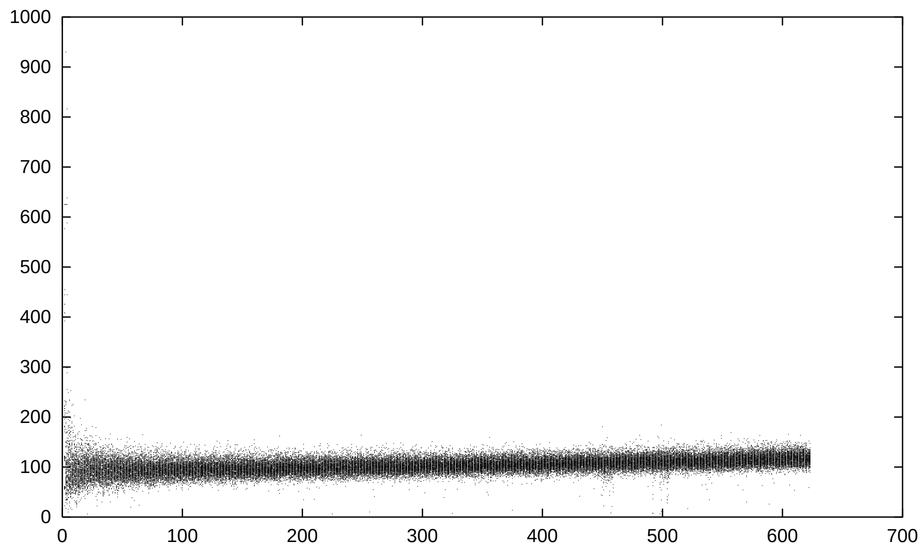


Figure 9: The relation between (the ratio of MSLiP and StochTrans computing times) and the number of scenarios

Figure 8 shows the time taken by StochTrans. The runaway points do not denote problems, which take much time to solve, but moments with a big system load. The time in figure 8 is the time between the end of input and the start of output. The additional load of other programs have to be taken into account, because the comparison was started by the command *nice -17 do_comp.sh*.

Figure 9 shows that it is usefull to develope an adapted algorithm for a special structured problem, since it shows that StochTrans is about 100 times faster.

5 Conclusions

Further examples were computed with up to 200 000 scenarios and 19 stages. The number of variables was about 4 000 000. The solution of such large problems took ca 60 seconds. Problems with more than 230 000 scenarios exceed the memory (64 Mbyte). Then, solving such large problems took more than 2000 seconds swapping time included.

The crucial point in that algorithm is the application of a subset of descent directions to that problem. This set was obtained by analysis of the problems special structure.

The problem dealing with thermal power units only can be solved by dynamic programming as described in [ZG88]. The dual program itself can be solved by a Bundle Method ([Kiw85][SZ92][TBL96][FKL96]).

The efficiency of the described method is an essential contribution, which determine to great extend the effective application of the dual approach.

References

[Ada80] D. Adams. *The Hitch Hiker's Guide to the Galaxy.* Pan Books Ltd., 1980.

[Bir85] J.R. Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Operation Research*, 33:989–1007, 1985.

[BLSP83] D.P. Bertsekas, G.S. Lauer, N.R. Sandel, and T.A. Posbergh. Optimal short-term scheduling of large-scale power systems. *IEEE Transactions on AC*, 28(1):1–11, January 1983.

[CPL95] CPLEX Optimization, Inc. *Using the CPLEX Callable Library*, 1989-19995.

[DR96] D. Dentcheva and W. Römisch. Optimal power generation und uncertainty via stochastic programming. Manuscript, Humboldt-University Berlin (in preparation), 1996.

[FKL96] S. Feltenmark, K. Kiwiel, and P.-O. Lindberg. Solving unit commitment problems in power production. *Handout at SOR-Conference 1996*, 1996.

[Gas90] H. I. Gassmann. Mslip: A computer code for the multistage stochastic linear programming problem. *Mathematical Programming*, 47:407–423, 1990.

[GRS92] J. Guddat, W. Römisch, and R. Schultz. Some applications of mathematical programming techniques in optimal power dispatch. *Computing*, 49:193–200, 1992.

[GRS95] N. Gröwe, W. Römisch, and R. Schultz. A simple recourse model for power dispatch under uncertain demand. *Annals of Operations Research*, 59:135–164, 1995.

[Jun94] D. Jungnickel. *Graphen, Netzwerke und Algorithmen.* Wissenschaftsverlag, 1994.

[Kiw85] K.C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization*, volume 1133. Springer Berlin, 1985.

[NR89] G.L. Nemhauser and A.H.G. Rinnooy Kan, editors. *Optimization*, volume 1 of *Handbooks in Operations Research and Management Science.* North–Holland, Amsterdam, 1989. edited by G.L. Nemhauser, A.H.G. Rinnooy Kan and M.J. Todd.

[NR90] G.L. Nemhauser and A.H.G. Rinnooy Kan, editors. *Stochastic Models*, volume 2 of *Handbooks in Operations Research and Management Science.* North Holland, 1990. edited by D.P. Heyman and M.J. Sobel.

[NW88] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley and Sons, 1988.

[RS96] W. Römisch and R. Schultz. Decomposition of a multi-stage stochastic program for power dispatch. *ZAMM (to appear)*, 1996.

[SF94] G.B. Sheble and G.N. Fahd. Unit commitment literature synopsis. *IEEE Transactions on Power Systems*, 9(1):128–135, February 1994.

[SZ92] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2:121–152, 1992.

[Tay90] H.M. Taylor. *Martingales and Random Walks*, pages 136–141. Volume 2 of Nemhauser and Rinnooy Kan [NR90], 1990. edited by D.P. Heyman and M.J. Sobel.

[TBL96] S. Takriti, J.R. Birge, and E. Long. A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems*, 1996. to appear.

[vSW67] R. van Slyke and R.J.-B. Wets. L-shaped linear programs with application to optimal control and stochastic optimization. *SIAM Journal on Applied Mathematics*, 17:638–663, 1967.

[Wet89] R.J.-B. Wets. *Stochastic Programming*, pages 573–629. Volume 1 of Nemhauser and Rinnooy Kan [NR89], 1989. edited by G.L. Nemhauser, A.H.G. Rinnooy Kan and M.J. Todd.

[ZG88] F. Zhuang and F.D. Galiana. Towards a more rigorous and practical unit commitment by lagrangian relaxation. *IEEE Transactions on Power Systems*, 3:763 – 773, 1988.