CIRRELT
Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport
Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Optimal Scenario Set Partitioning for Multistage Stochastic Programming with the Progressive Hedging Algorithm

Pierre-Luc Carpentier
Michel Gendreau
Fabian Bastin

September 2013

CIRRELT-2013-55

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

UNIVERSITÉ LAVAL UQÀM HEC MONTRÉAL ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

Optimal Scenario Set Partitioning for Multistage Stochastic Programming with the Progressive Hedging Algorithm

Pierre-Luc Carpentier[1,2,*], Michel Gendreau[1,2], Fabian Bastin[1,3]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Mathematics and Industrial Engineering, École Polytechnique de Montréal, P.O. Box 6079, Station Centre-ville, Montréal, Canada H3C 3A7

[3] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

Abstract. The progressive hedging algorithm (PHA) is a classical method for solving stochastic programs (SPs) defined on scenario trees. To use the traditional PHA, all non-anticipativity constraints (NACs) must be formulated explicitly as linear equality constraints to ensure that any feasible solution is scenario in variant at each tree node. An augmented Lagrangian relaxation is applied on NACs. The resulting auxiliary problem is decomposed into smaller scenario subproblems containing a linear and a quadratic penalty term for each NAC. Unfortunately, the number of NACs grows exponentially with the number of branching stages in the tree and applying the PHA is particularly challenging when this number is large. The large number of penalty terms slows down the PHA by increasing the difficulty of individual scenario-subproblems and by increasing the total number of iterations to be performed. In this paper, we propose a new approach to improve the PHA running time when solving multistage SPs (MSPs). Our method builds an optimal partitioning scheme which minimizes the total number inter-subtree NACs that need to be relaxed. Each subproblem is formulated as small MSPs defined a particular scenario subtree and is solved directly using a readily-available solver. Intra-subtree NACs are represented implicitly using a mathematical formulation based on a node-wise index system. The proposed approach is tested on an hydroelectricity generation planning problem with stochastic inflows.

Keywords: Stochastic programming, large scale optimization, augmented Lagrangian, scenario tree, progressive hedging.

Acknowledgements. The authors would like to express their gratitude to Hydro-Québec, the Natural Sciences and Engineering Research Council of Canada (NSERC) and Fonds de recherche du Québec – Nature et technologies (FRQNT) for their financial support.

[*] Corresponding author: Pierre-Luc.Carpentier@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2013

1. Introduction

Practical applications of stochastic programming (SP) methods for solving uncertain optimization problems are quite numerous cover a wide spectrum of domains (Dupačová , 2002; Rusczynzki and Shapiro , 2003). The book by Gassman and Ziemba (2013) presents different applications of these methods in energy, logistics and production planning, finance and telecommunications. Most of these applications contain one or several random parameters (e.g. inflows, prices, interest rates, yield, demand, electrical load, wind/solar generation) that are characterized by a (joint) continuous probability distribution. A popular approach to represent continuously distributed parameters in SP models consists in replacing the original continuous distribution by a discrete distribution possessing a finite number of possible outcomes (scenarios). This type of approximation leads to a scenario tree representation of uncertainty. Fig. 1a shows a simple example of a scenario tree with three stages, four scenarios and seven nodes. Different methods were proposed over the years for constructing a scenario tree from a set of synthetic or historical time series (e.g. Pflug , 2001; Høyland and Wallace , 2001; Latorre et al. , 2007). Heitsch and Römisch (2009) proposed a construction method based the theoretical results of Heitsch et al. (2006). The SCENRED2 package of the General Algebraic Modeling System (GAMS) is a computer implementation of this technique.

Multistage stochastic programs (MSPs) defined on scenario trees can be reformulated into deterministic equivalent programs (DEPs) of finite size (variables, contraints). The number of decision variables and constraints is usually proportionnal to the number of nodes contained in the scenario tree. Therefore, the size of DEPs grows exponentially with the discretization level (number of branching stages, number of branches per stage) used to describe the original probability distribution. In general, most real-world MSPs cannot be solved directly using a commercial solver (e.g. GLPK, Gurobi, XPress-MP, CPLEX) when an accurate representation of random parameter is used. The required amount of random access memory (RAM) is typically the main limiting factor when solving large-scale linear or quadratic programs directly.

Over the past decades, different decomposition methods were proposed in the literature for solving efficiently large-scale SPs (Rusczynzki , 2003; Sagastizábal , 2012). Solution methods based on Benders (1962) decomposition (e.g. Van Slyke and Wets , 1969; Birge , 1985; Birge and Louveaux , 1988; Pereira and Pinto , 1991; Laporte and Louveaux , 1993; Küchler and Vigerske , 2007; Carpentier et al. , 2013b) are widely used for solving linear MSPs. These methods use a stage-wise decomposition scheme and work by constructing convex and piecewise linear recourse functions.

The progressive hedging algorithm (PHA) proposed by Rockafellar and Wets (1991) is another popular method for solving SPs defined on scenario trees. This method was applied successfully in various type of problems including electricity generation planning (Gonçalves et al. , 2011; Carpentier et al. , 2013a), network design (Crainic et al. , 2011), network flow (Mulvey and Vladimirou , 1991), lot-sizing (Haugen et al. , 2001) and resources allocation (Watson and Woodruff

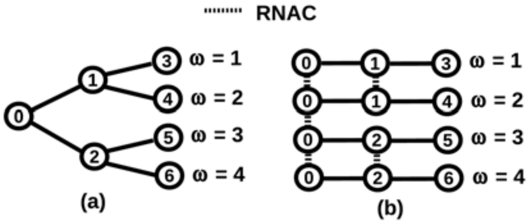

Figure 1: (a) Example of a scenario tree. (b) Illustration of the scenario-decomposition scheme.

, 2011). To use the the traditional version of the PHA, a decision vector $x_{t\omega} \in \mathbb{R}^m$ must be defined at each stage (time period) $t = 1, ..., T$ for all scenarios $\omega \in \Omega$ contained in the tree. All non-anticipativity constraints (NACs) must formulated explicitly as linear equality constraints

$$x_{t(n)\omega} = \hat{x}_n, \qquad \forall \omega \in \Omega, n \in \mathcal{N}^*_\omega : \lambda_{n\omega} \tag{1}$$

to ensure that feasible solutions are scenario-invariant at each tree node. The function $t(n)$ returns the stage index associated with tree node n, $\hat{x}_n \in \mathbb{R}^m$ is the decision vector at node n, $\lambda_{n\omega} \in \mathbb{R}^m$ is the vector of Lagrange multipliers associated for scenario ω at node n, $\mathcal{N}^*_\omega$ is a set that contains all nodes visited by scenario ω and by at least another scenarios. The PHA works by applying a scenario-decomposition scheme on the resulting mathematical formulation and by applying an augmented Lagrangian relaxation on constraints (1). Fig. 1b illustrates an application of the scenario-decomposition scheme on the scenario tree shown on Fig. 1a. In this example, $\mathcal{N}^*_1 = \mathcal{N}^*_2 = \{0, 1\}$ and $\mathcal{N}^*_3 = \mathcal{N}^*_4 = \{0, 2\}$. An augmented Lagrangian relaxation must be applied on the following NACs

$$x_{1,1} = x_{1,2} = x_{1,3} = x_{1,4} = \hat{x}_0, \tag{2}$$
$$x_{2,1} = x_{2,2} = \hat{x}_1, \; x_{2,3} = x_{2,4} = \hat{x}_2. \tag{3}$$

A different version of the PHA was proposed by Crainic et al. (2013) for solving two-stage stochastic network design problems. Instead of applying the traditional scenario-decomposition scheme, these authors applied a multi-scenario decomposition scheme. In this approach, the scenario set Ω is partitioned into disjoint subsets (clusters) Ω_c for $c \in \mathcal{C}$ and a decision vector $\tilde{x}_{nc}$ is defined at each node n contained in each scenario cluster c. The following NAC are formulated as linear equality constraints

$$\tilde{x}_{nc} = \hat{x}_n, \qquad \forall c \in \mathcal{C}, n \in \mathcal{N}^*_c : \; \tilde{\lambda}_{nc} \tag{4}$$

where $\mathcal{C}$ is the set of clusters, $\mathcal{N}^*_c$ is the set of nodes contained in cluster c and $\tilde{\lambda}_{nc}$ is the vector of Lagrange multipliers. Each subproblem corresponds to a small two-stage SP (TSP) defined on a cluster c of scenarios which are

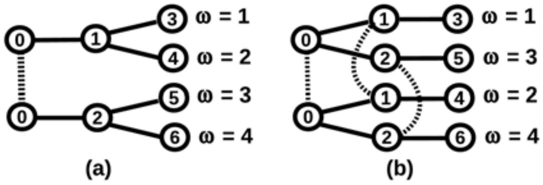

Figure 2: Examples of clustering schemes for a multistage scenario tree.

aggregated according to a similarity (or dissimilarity) level. An AL relaxation is applied on all NAC and the resulting problem is decomposed into cluster-subproblems. Decomposing the TSP into less (and larger) subproblems allows to reduce the number of relaxed NACs (RNACs) which, in turn, reduces the number of linear-quadratic penalty terms that need to be included in subproblems. Doing this makes individual subproblems easier to solve and enhances the algorithm converence rate.

The similarity-based method proposed by Crainic et al. (2013) works well on TSPs because the topology of the underlying scenario tree is quite simple. In two-stage scenario trees, all leaves possess the same ancestor node (the root). Consequently, the total number of RNACs is always equal to the number of clusters no matter which scenarios are grouped together. Therefore, the only relevant metric that can be used for grouping scenarios would be a function of the similarity (or dissimilarity) level between scenarios. However, general multi-stage scenario trees usually possess a complex branching structure and, consequently, the number of RNACs will vary importantly depending on which partitioning scheme is used. Fig. 2 shows a simple illustrative example where two different partitioning schemes are applied to the tree shown on Fig. 1a. For the scheme shown on Fig. 2a, the clusters $c = 1$ and $c = 2$ are defined on scenarios $\Omega_1 = \{1, 2\}$ and $\Omega_2 = \{3, 4\}$, respectively. With this scheme, only the two following NACs need to be formulated explicitly (and relaxed)

$$\tilde{x}_{0,1} = \tilde{x}_{0,2} = \hat{x}_0$$

where $\tilde{x}_{nc}$ is the decision at node n in cluster c. The remaining NACs associated with nodes 1 and 2 are dealt with directly when solving subproblems defined on clusters 1 and 2, respectively. For the scheme shown on Fig. 2b, the clusters $c = 1$ and $c = 2$ are defined on scenarios $\Omega_1 = \{1, 3\}$ and $\Omega_2 = \{2, 4\}$, respectively. The following NACs need to be formulated explicitly

$$\tilde{x}_{0,1} = \tilde{x}_{0,2} = \hat{x}_0,\ \tilde{x}_{1,1} = \tilde{x}_{1,2} = \hat{x}_1,\ \tilde{x}_{2,1} = \tilde{x}_{2,2} = \hat{x}_2.$$

In this paper, we propose an new approach to improve the PHA running time when solving multi-stage SPs (MSPs). Our method aims at building an partition of the scenario set which minimizes the total number inter-subtree NACs (4) that need to be relaxed. Reducing the number of RNACs makes individual subproblems easier to solve (less penalty terms) and improves the PHA's convergence rate (less iterations).

The paper is organized as follows. A general mathematical formulation for MSPs is presented in Section 2. Section 3 presents the progressive hedging algorithm applied in the context of a subtree decomposition scheme. Section 4 describes a numerical experiment performed to evaluate the solution method. Numerical results are presented in Section 5. Comments and conclusions are drawn in Section 6.

2. Problem formulation

2.1. Multistage stochastic program

We consider the following optimization problem defined on T time periods (stages)

$$(\mathcal{P}) \qquad \min \mathbb{E}\left[\sum_{t=1}^{T} g_t(x_t, \xi_t)\right] \tag{5}$$

subject to

$$A_t(\xi_t)x_t + B_t(\xi_t)x_{t-1} = b_t(\xi_t) \quad , \quad \forall t = 1, ..., T \tag{6}$$

$$x_t \in \mathcal{X}_t \quad , \quad \forall t = 1, ..., T \tag{7}$$

$$x_t \in \mathcal{F}_t(\xi_1, ..., \xi_t) \quad , \quad \forall t = 1, ..., T \tag{8}$$

where $\mathbb{E}[\,\cdot\,]$ is the expectation operator, $g_t(\cdot, \cdot)$ is a convex function that represent the operating costs at time period t, $x_t \in \mathbb{R}^m$ is the decision vector at time period t and ξ_t is a random vector at time period t. Each component of ξ_t represents one of the problem's random parameter at time period t. Equations (6) and (7) represent dynamic (e.g. water budget, inventory dynamics, ramping constraints) and static (e.g. system limits, ...) constraints, respectively. Coefficient of technological matrices A_t, B_t and vectors b_t are treated as random variables. The sets of static constraints $\mathcal{X}_t$ are assumed to be non-empty and convex. Non-anticipativity constraints (8) ensure that each x_t is chosen using only past (known) observations $(\xi_1, ..., \xi_t)$ and cannot depend on future (unknown) realizations of random parameters $\xi_{t+1}, ..., \xi_T$. Each set $\mathcal{F}_t$ contains all solutions that meet this criterion at time t.

2.2. Scenario tree

Each random vector ξ_t is characterized by a known probability distribution $\mathbb{P}_t(\,\cdot\,|\,\xi_1, ..., \xi_{t-1})$ which is conditional to previous observations $\xi_1, ..., \xi_{t-1}$. We assume that $\mathbb{P}_t$ possess a finite number of possible outcomes at each $t = 1, ..., T$. We also assume that all random parameters are exogenous to the controlled system. Therefore, $\mathbb{P}_t$ is not influenced by x_t. Making these assumptions enables us to represent the stochastic process $\{\xi_t : t = 1, ..., T\}$ using a finite scenario tree $\mathcal{T}$. Each node $n \in \mathcal{N}$ has an occurence probability of p_n. Each scenario $\omega \in \Omega$ in $\mathcal{T}$ corresponds to a path from the root $0 \in \mathcal{N}$ to a particular leaf $\ell(\omega) \in \mathcal{L}$.

The probability of a given scenario ω correspond to the probability $p_{\ell(\omega)}$ of its leaf $\ell(\omega)$. The branching structure of $\mathcal{T}$ is represented by a function $a(n)$ which returns the ancestor of any node n. The vector ξ_n represents the realization of random parameters at node n. The set $\Omega(n)$ contains all scenarios visiting node n. Each set $\Delta(n)$ contains all child nodes of n. Fig. 1a shows a simple example with $T = 3$, $\mathcal{N} = \{0, 1, 2, 3, 4, 5, 6\}$, $\mathcal{L} = \{3, 4, 5, 6\}$, $\Omega = \{1, 2, 3, 4\}$, $a(1) = a(2) = 0$.

2.3. Node-wise formulation

The problem $\mathcal{P}$ defined on a scenario tree $\mathcal{T}$ can be transformed into the following deterministic equivalent program (DEP)

$$(\mathcal{E}) \qquad \min \sum_{n \in \mathcal{N}} p_n g_{t(n)}(\hat{x}_n, \xi_n) \tag{9}$$

subject to

$$A_n \hat{x}_n + B_n \hat{x}_{a(n)} = b_n \quad , \quad \forall n \in \mathcal{N}, \tag{10}$$

$$\hat{x}_n \in \mathcal{X}_n \quad , \quad \forall n \in \mathcal{N}. \tag{11}$$

The program $\mathcal{E}$ is obtained from $\mathcal{P}$ by replacing all occurences of stage-wise decision vectors x_t by a node-wise decision vector $\hat{x}_n$ at node n. In the objective function, the expectation operator is replaced by a finite sum. Each term is weighted by the probability of the corresponding tree node. Non-anticipativity constraints (8) are represented implicitly with this formulation because we use a node-wise index system.

3. Decomposition method

3.1. Scenario clusters

We partition the scenario set Ω into disjoint subsets (clusters) Ω_c where $c \in \mathcal{C}$. We denote by $\mathcal{T}_c$ the subtree associated with all scenarios in Ω_c. Each set $\mathcal{N}_c$ contains all the nodes that are visited by the scenarios in Ω_c. The occurence probability of cluster c is defined as follow

$$\tilde{p}_c := \sum_{\omega \in \Omega_c} p_{\ell(\omega)}. \tag{12}$$

The probability $\hat{p}_{nc}$ of node n conditional to cluster c is defined as follow

$$\hat{p}_{nc} := p_n / \tilde{p}_c. \tag{13}$$

3.2. Cluster-wise formulation

The program $\mathcal{E}$ defined on scenarios clusters $c \in \mathcal{C}$ is reformulated into the following equivalent program

$$(\tilde{\mathcal{E}}) \qquad \min \sum_{c \in \mathcal{C}} \tilde{p}_c \sum_{n \in \mathcal{N}_c} \hat{p}_{nc} g_{t(n)}(\tilde{x}_{nc}, \xi_n) \tag{14}$$

subject to

$$A_n \tilde{x}_{nc} + B_n \tilde{x}_{a(n)c} = b_n \quad , \quad \forall c \in \mathcal{C},\ n \in \mathcal{N}_c \tag{15}$$

$$\tilde{x}_{nc} \in \mathcal{X}_n \quad , \quad \forall c \in \mathcal{C},\ n \in \mathcal{N}_c, \tag{16}$$

$$\tilde{x}_{nc} = \hat{x}_n \quad , \quad \forall c \in \mathcal{C},\ n \in \mathcal{N}_c^* \ : \ \tilde{\lambda}_{nc}. \tag{17}$$

The formulation $\tilde{\mathcal{E}}$ can be obtained from $\mathcal{E}$ by making the following transformations. The objective function (14) and constraints (15)–(16) are obtained by replacing all occurences of $\hat{x}_n$ at nodes $n \in \mathcal{N}$ by an alternative decision vector $\tilde{x}_{nc}$ defined at nodes $n \in \mathcal{N}_c$ visited by clusters $c \in \mathcal{C}$. In (14), the probability p_n of node n is replaced $\tilde{p}_c \hat{p}_{nc}$ by according to (13). The NACs (17) ensure that any feasible solution of $\tilde{\mathcal{E}}$ is subtree-invariant at all tree nodes and $\tilde{\lambda}_{nc}$ is the vector of Lagrange multipliers associated with node n and cluster c. Each set $\mathcal{N}_c^*$ contains all nodes visited by cluster c and by at least another cluster. For the example shown on Fig. 2a, $\mathcal{N}_1^* = \mathcal{N}_2^* = \{0\}$.

3.3. Augmented lagrangian

We apply an augmented Lagrangian relaxation on constraints (17) and the resulting objective function to be minimized is

$$\mathcal{A}_\rho(\tilde{x}, \hat{x}, \tilde{\lambda}) = \sum_{c \in \mathcal{C}} \tilde{p}_c \left(\sum_{n \in \mathcal{N}_c} \hat{p}_{nc} g_{t(n)}(\tilde{x}_{nc}, \xi_n) + \sum_{n \in \mathcal{N}_c^*} \left(\tilde{\lambda}'_{nc}(\tilde{x}_{nc} - \hat{x}_n) + \frac{\rho}{2} \|\tilde{x}_{nc} - \hat{x}_n\|^2 \right) \right) \tag{18}$$

subject to constraints (15)–(16). The penalty parameter ρ is a positive constant, $\tilde{x} = (\tilde{x}_{nc})$ is the vector of cluster-wise decision vectors, $\hat{x} = (\hat{x}_n)$ is the vector of node-wise decision vectors, $\tilde{\lambda} = (\tilde{\lambda}_{nc})$ is the vector of Lagrange multipliers associated with constraints (17). All vectors are assumed to be column vectors, $(\cdot)'$ is the transpose operator and $\|\cdot\|$ is the euclidian norm.

3.4. Progressive hedging algorithm

The algorithm begins with an initial penalty parameter ρ_0, a suboptimal node-wise solution $\hat{x}^0 = (\hat{x}_n^0)$ and Lagrange multiplier $\tilde{\lambda}^0 = (\tilde{\lambda}_{nc}^0)$. Then, at each iteration $k = 0, 1, 2, \ldots$, the two following steps are performed.

Step 1. Find a new cluster-wise solution $\tilde{x}^{k+1} = (\tilde{x}_{nc}^{k+1})$ by minimizing $\mathcal{A}_{\rho_k}(\tilde{x}, \hat{x}^k, \tilde{\lambda}^k)$ for $\tilde{x}$ subject to constraints (15)–(16). Because we consider

$(\tilde{x}^k, \tilde{\lambda}^k)$ as fixed values, this problem is separable by cluster. Each cluster-subproblem is a relatively small MSP defined on a subtree $\mathcal{T}_c$ associated with a particular cluster c. The subproblem associated with cluster c is

$$(\mathcal{S}_c^k) \qquad \min \tilde{p}_c \sum_{n\in\mathcal{N}_c} \hat{p}_{nc} g_{t(n)}(\tilde{x}_{nc}, \xi_n) + \sum_{n\in\mathcal{N}_c^*} \left((\tilde{\lambda}_{nc}^k)'(\tilde{x}_{nc} - \hat{x}_n^k) + \frac{\rho_k}{2} \|\tilde{x}_{nc} - \hat{x}_n^k\|^2 \right)$$

subject to

$$A_n\tilde{x}_{nc} + B_n\tilde{x}_{a(n)c} = b_n \quad , \quad \forall n \in \mathcal{N}_c, \tag{19}$$

$$\tilde{x}_{nc} \in \mathcal{X}_n \quad , \quad \forall n \in \mathcal{N}_c. \tag{20}$$

Step 2. a) Compute the new cluster-averaged solution

$$\hat{x}_n^{k+1} \leftarrow \sum_{c\in\mathcal{C}(n)} \tilde{p}_c \hat{p}_{nc} \tilde{x}_{nc}^k / \sum_{c\in\mathcal{C}(n)} \tilde{p}_c \hat{p}_{nc}, \qquad \forall n \in \mathcal{N}_*$$

where $\mathcal{C}(n)$ is the set of clusters visiting node n.

b) Update the Lagrange multipliers

$$\tilde{\lambda}_{nc}^{k+1} \leftarrow \tilde{\lambda}_{nc}^k + \rho_k(\tilde{x}_{nc}^{k+1} - \hat{x}_n^{k+1}), \qquad \forall c \in \mathcal{C},\ n \in \mathcal{N}_c^*.$$

c) Update the penalty parameter using

$$\rho_{k+1} \leftarrow \mu\rho_k \tag{21}$$

where $\mu \geq 1$ is a constant. This formula is the traditional update formula used in general augmented Lagrangian methods (Nocedal and Wright , 2006).

d) Stop if

$$\zeta_k \leftarrow \frac{1}{T} \sum_{c\in\mathcal{C}} \tilde{p}_c \sum_{n\in\mathcal{N}_c^*} \|\tilde{x}_{nc}^{k+1} - \hat{x}_n^{k+1}\|^2 < \epsilon. \tag{22}$$

Otherwise, return to step 1. The stopping criterion ϵ is a positive constant and ζ_k the total violation of NAC at iteration k. The constraints (15)–(16) are satisfied by $\tilde{x}_{nc}^k$ at each k. The NACs (17) are violated at early iterations and the feasibility will improve gradually as the number of iteration increases. In this problem, all decision variables are continuous and all constraints and the objective function are convex. Therefore, the PHA is an exact solution method for this problem. Rockafellar and Wets (1991) presented a proof of convergence for the PHA.

3.5. Optimal subtree decomposition

The aim of the optimal subtree decomposition problem (OSDP) is to partition the scenario set Ω into disjoint subsets (clusters) $\Omega_1, ..., \Omega_{|\mathcal{C}|}$ to minimize the PHA running time. Any feasible partitioning scheme should ensure that the size of subproblems $\mathcal{S}_c^k$ is manageable. The total running time depends on the algorithm's convergence rate (number of iterations) and on the difficulty level of individual subproblems $\mathcal{S}_c^k$ (time per iteration).

In this paper, we formulate the OSDP as follow

$$\min \sum_{c \in \mathcal{C}} |\mathcal{N}_c^*| \tag{23}$$

subject to

$$|\Omega_c| \leq N_{\max} \quad , \quad \forall c \in \mathcal{C} \tag{24}$$

$$\bigcup_{c \in \mathcal{C}} \Omega_c = \Omega \quad , \tag{25}$$

$$\Omega_c \cap \Omega_d = \emptyset \quad , \quad \forall c, d \in \mathcal{C}, c \neq d \tag{26}$$

The objective function (23) to be minimized is the total number of RNACs linking clusters $c \in \mathcal{C}$. This performance metric makes sense in the context of MSPs because a linear and a quadratic penalty term must be included in subproblems for each RNAC. By reducing the number of RNAC, we expect that the convergence rate will improve. We also expect subproblems to be easier to solve. The constraints (24) ensures that all clusters cannot contain more than $N_{\max}$ scenarios. The constraints (25) ensure that all scenarios are assigned to a particular cluster. The constraints (25) ensure that all subsets Ω_c are disjoint.

3.6. Heuristic method

We propose a heuristic method which enables to a good solutions to the OSDP. The Algorithm 1 summmarizes all steps of our method. This algorithm receives a general scenario tree $\mathcal{T}$ and the parameter $N_{\max}$ which represent the maximal number of scenarios that can be contained in a single cluster. The proposed method choses how many clusters are required and returns Ω_c, $\tilde{p}_c$ and $\hat{p}_{nc}$ for each $c \in \mathcal{C}$.

The Algorithm 1 builds a finite number of clusters c sequentially for $c = 1, 2, ..., |\mathcal{C}|$ by selecting a different reference node $\tilde{n}$ among the set of candidate nodes $\tilde{\mathcal{N}}$. The algorithm starts by building cluster $c = 1$ and only the root node 0 is contained in $\tilde{\mathcal{N}}$. At each iteration the main `while` loop, the algorithm selects a new reference node $\tilde{n}$ and removes it from $\tilde{\mathcal{N}}$. The node $\tilde{n}$ chosen is the one that is visited by the most scenarios in the tree. If the node $\tilde{n}$ is visited by more scenarios than is allowed in a single cluster $N_{\max}$, then this node is not the reference node of current cluster c to be constructed and all child nodes of $\tilde{n}$ are added to the set $\tilde{\mathcal{N}}$. Otherwise, the node $\tilde{n}$ is the reference node of cluster c, Ω_c is defined by all scenarios visiting node $\tilde{n}$, the probability of cluster c correspond to the occurence probability of n and the conditional probability of each node $n \in \mathcal{N}_c$ visited by $\omega \in \Omega_c$ is computed as follow:

- The conditional probability of tree nodes $n \in \mathcal{D}(\tilde{n})$ that are located downstream of $\tilde{n}$ is computed using equation (13).

- The conditional probability of tree nodes $n \in \mathcal{K}(\tilde{n})$ that are located upstream of $\tilde{n}$ (including $\tilde{n}$ itself) is equal to one.

Algorithm 1 Scenarios clustering heuristic.

```
$c \leftarrow 1$, $\tilde{\mathcal{N}} \leftarrow \{0\}$
while $\tilde{\mathcal{N}} \neq \emptyset$ do
  $\tilde{n} \leftarrow \arg\max\{|\Omega(n)| : n \in \tilde{\mathcal{N}}\}$
  $\tilde{\mathcal{N}} \leftarrow \tilde{\mathcal{N}} - \{\tilde{n}\}$
  if $|\Omega(\tilde{n})| > N_{\max}$ then
    $\tilde{\mathcal{N}} \leftarrow \tilde{\mathcal{N}} \cup \Delta(\tilde{n})$
  else
    $\Omega_c \leftarrow \Omega(\tilde{n})$
    $\tilde{p}_c \leftarrow p_{\tilde{n}}$
    for $n \in \mathcal{D}(\tilde{n})$ do
      $\hat{p}_{nc} \leftarrow p_n/\tilde{p}_c$
    end for
    for $n \in \mathcal{K}(\tilde{n})$ do
      $\hat{p}_{nc} \leftarrow 1$
    end for
    $c \leftarrow c + 1$
  end if
end while
```

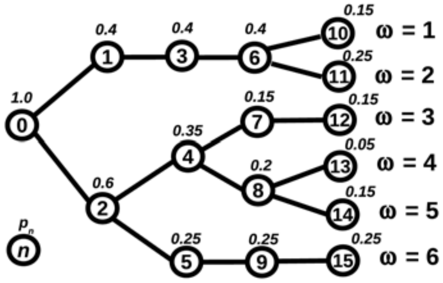

Figure 3: Example of a scenario tree.

For the example shown on Fig. 3, $\mathcal{D}(4) = \{7, 8, 12, 13, 14\}$ and $\mathcal{K}(4) = \{0, 2, 4\}$. The algorithms continues as long as the set $\tilde{\mathcal{N}}$ is non-empty.

We illustrate how the Algorithm 1 works by applying it on the scenario tree shown on Fig. 3 with $N_{\max} = 3$. The algorithm returns three clusters as shown on Fig. 4. The probability of each cluster is $\tilde{p}_1 = 0.35$, $\tilde{p}_2 = 0.4$ and $\tilde{p}_3 = 0.25$. The following NACs must be formulated explicitly

$$\tilde{x}_{0,1} = \tilde{x}_{0,2} = \tilde{x}_{0,3} = \hat{x}_0$$

$$\tilde{x}_{2,1} = \tilde{x}_{2,3} = \hat{x}_2.$$

The following intermediary results are obtained at five iterations of the `while` loop

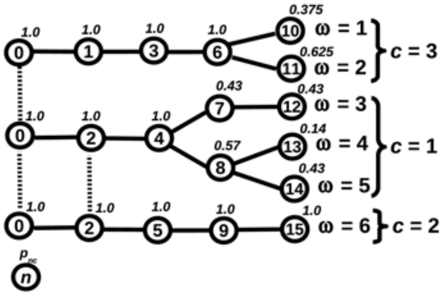

Figure 4: Subtrees.

1. $\tilde{\mathcal{N}} = \{0\}$ and, consequently, the $\tilde{n} = 0$ is selected and removed from $\tilde{\mathcal{N}}$. The selected node is visited by all scenarios ($\Omega(\tilde{n}) = \{1, ..., 6\}$). Because, $|\Omega(\tilde{n})| = 6 > 3$, the child nodes $\Delta(0) = \{1, 2\}$ are added to $\tilde{\mathcal{N}}$.
2. $\tilde{\mathcal{N}} = \{1, 2\}$. The node $\tilde{n} = 2$ is selected because $|\Omega(2)| = 4 > 2 = |\Omega(1)|$. Because $|\Omega(2)| = 4 > 3$, the child nodes $\Delta(2) = \{4, 5\}$ are added to $\tilde{\mathcal{N}}$.
3. $\tilde{\mathcal{N}} = \{1, 4, 5\}$. The node $\tilde{n} = 4$ is selected and removed from $\tilde{\mathcal{N}}$ because $|\Omega(4)| = 3 > 2 = |\Omega(1)| < 1 = |\Omega(5)|$. Because $|\Omega(4)| = 3 \leq 3$, the cluster $c = 1$ is defined by the scenarios $\Omega_1 = \{3, 4, 5\}$ and has a total probability $\tilde{p}_1 = 0.35$. The sets $\mathcal{K}_1 = \{0, 2, 4\}$ and $\mathcal{D}_1 = \{7, 8, 12, 13, 14\}$. The probability of each node in $c = 1$ is $\hat{p}_{0,1} = \hat{p}_{2,1} = \hat{p}_{4,1} = 1$, $\hat{p}_{7,1} = \hat{p}_{12,1} = \hat{p}_{14,1} = 0.15/0.35$, $\hat{p}_{8,1} = 0.2/0.35$ and $\hat{p}_{13,1} = 0.05/0.35$. We now start constructing the cluster $c = 2$.
4. $\tilde{\mathcal{N}} = \{1, 5\}$. The node $\tilde{n} = 1$ is selected and removed from $\tilde{\mathcal{N}}$ because $|\Omega(1)| = 2 > 1\,|\Omega(5)|$. Because $|\Omega(1)| = 2 \leq 3$, the cluster $c = 2$ is defined by the scenarios $\Omega_1 = \{1, 2\}$ and has a total probability $\tilde{p}_2 = 0.4$. The sets $\mathcal{K}_2 = \{0, 1, 3, 6\}$ and $\mathcal{D}_2 = \{10, 11\}$. The probability of each node in $c = 2$ is $\hat{p}_{0,2} = \hat{p}_{1,2} = \hat{p}_{3,2} = \hat{p}_{6,2} = 1$, $\hat{p}_{10,2} = 0.15/0.4$ and $\hat{p}_{10,2} = 0.25/0.4$. We now start constructing the cluster $c = 3$.
5. $\tilde{\mathcal{N}} = \{5\}$. The node $\tilde{n} = 5$ is selected and removed from $\tilde{\mathcal{N}}$. Because $|\Omega(5)| = 1 \leq 3$, the cluster $c = 3$ is defined by the scenarios $\Omega_1 = \{5, 6\}$ and has a total probability $\tilde{p}_3 = 0.25$. The sets $\mathcal{K}_3 = \{0, 2, 5\}$ and $\mathcal{D}_3 = \{15\}$. The probability of each node in $c = 3$ is $\hat{p}_{0,3} = \hat{p}_{2,3} = \hat{p}_{5,3} = \hat{p}_{9,3} = \hat{p}_{15,3} = 1$. $\tilde{\mathcal{N}} = \emptyset$. This is the last iteration because $\tilde{\mathcal{N}} = \emptyset$.

4. Numerical experiment

We apply the PHA described in subsection 3.4 on a typical reservoir management problem (RMP) with stochastic inflows. This problem is formulated as a particular case of the general mathematical program $\mathcal{P}$ and hydrologic uncertainty is modeled by a finite scenario tree. Over the past decades, many stochastic optimization methods were proposed in the literature for solving RMPs (Yeh

, 1985; Labadie , 2004; Rani and Moreira , 2010). Among those methods, only a few can be applied on large reservoir systems (20–40 reservoirs). The PHA was rarely used in this field, but this method is nevertheless a promising alternative to traditional methods such as the Nested Benders decomposition (NBD) algorithm (Birge , 1985) for managing high dimensional reservoir systems. For example, Carpentier et al. (2013a) and Gonçalves et al. (2011) applied the classical version of the PHA on large hydroelectric reservoir systems in Québec and Brazil, respectively.

4.1. Problem statement

We consider an hydroelectricity producer that operates I hydro plants and J interconnected reservoirs over a T-period planning horizon. The objective function to be maximized is the expected value of

$$\sum_{t=1}^{T}\sum_{i=1}^{I} P_{it}\Delta t + \sum_{j=1}^{J} \alpha_j (v_{jT} - \underline{v}_j) \qquad \text{(MWh)} \tag{27}$$

where P_{it} (MW) is the power output of hydro plant i during time period t, Δt (hours) is the time step, v_{jT} (hm^3) is the volume of water stored in reservoir j at the end of time period T, $\underline{v}_j$ (hm^3) is the minimum storage of reservoir j and α_j (MWh/hm^3) is the production factor of reservoir j. The objective function (27) contains two parts. The first part represents the amount energy generated by all hydro plants $i = 1, ..., I$ during time periods $t = 1, ..., T$. The second part represents the amount of potential energy stored in reservoirs $j = 1, ..., J$ at the end time period $t = T$.

We assume that the power output P_{it} of hydro plant i during time period t is a concave and piecewise linear function of the turbined outflow q_{it} (m^3 s^{-1}) at i during t and of the volume of water $v_{j(i)t}$ (hm^3) stored in the reservoir $j(i)$ located immediatly upstream of i at the end of t. This assumption enables us model head and generation efficiency variations at hydro plants. Fig. 5 shows an illustrative example with two pieces $h \in \{1, 2\}$. In the optimization model, the relationship between the P_{it}, q_{it} and $v_{j(i)t}$ is represented by the following linear inequality constraints

$$P_{it} \leq \gamma^0_{ih} + \gamma^1_{ih} q_{it} + \gamma^2_{ih} v_{it}, \qquad \forall i, t, h. \tag{28}$$

where $\gamma^0_{ih}, \gamma^1_{ih}, \gamma^2_{ih}$ are the linear coefficients of piece h.

The volume of water stored in reservoirs $j = 1, ..., J$ at time periods $t = 1, ..., T$ evolves from a known initial state v_{j0} according to

$$v_{jt} = v_{j,t-1} + \left(\sum_{u \in \mathcal{U}(j)} Q_{ut} - Q_{jt} + \mathcal{I}_{jt} \right) \beta \Delta t, \qquad \forall j, t \tag{29}$$

where $\mathcal{U}(j)$ is a set that contains all reservoirs that are located immediatly upstream of reservoir j, Q_{jt} (m^3 s^{-1}) is the controlled outflow of reservoir j

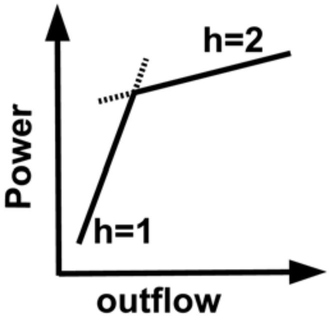

Figure 5: Hydroelectricity generation function.

during t, $\beta = 0.0036$ is a constante used for converting flow units into volumetric units and $\mathcal{I}_{jt}$ (m^3 s^{-1}) is a random parameter representing the intensity of natural inflows in reservoir j during t. The controlled outflow of reservoir j during t is defined as follow

$$Q_{jt} := s_{jt} + \sum_{i \in I(j)} q_{it} \qquad (\text{hm})^3$$

where s_{jt} (m^3 s^{-1}) is the spilled outflow of j during t and $I(j)$ is a set that contains all hydro plants connected directly to reservoir j.

All decision variables should also satisfy the following box constraints

$$\underline{v}_j \leq v_{jt}\overline{v}_j, \qquad \forall j, t \tag{30}$$

$$\underline{s}_j \leq s_{jt}\overline{s}_j, \qquad \forall j, t \tag{31}$$

$$\underline{P}_i \leq P_{it}\overline{P}_i, \qquad \forall i, t \tag{32}$$

$$\underline{q}_i \leq q_{it}\overline{q}_i, \qquad \forall i, t. \tag{33}$$

where $(\underline{v}_j, \underline{s}_j, \underline{P}_i, \underline{q}_i)$ and $(\overline{v}_j, \overline{s}_j, \overline{P}_i, \overline{q}_i)$ are parameters representing the lower and upper bounds on decision variables $(v_{jt}, s_{jt}, P_{it}, q_{it})$, respectively.

The RMP is a particular case of the general mathematical formulation $\mathcal{P}$ with random right-hand side vectors $b_t(\xi_t)$ at $t = 1, ..., T$. Each component of random vectors

$$\xi_t := (\mathcal{I}_{jt})$$

represents the intensity of natural inflows in a particular reservoir j at the corresponding time period t. The matrices A_t, B_t and cost functions g_t are deterministic and each decision vector is defined as follow

$$x_t := (P_{it}, q_{it}, v_{jt}, s_{jt}).$$

The cost functions at $t = 1, ..., T - 1$ are defined as follow

$$g_t(x_t) := -\sum_{i=1}^{I} P_{it}\Delta t.$$

The cost function at $t = T$ is

$$g_T(x_T) := -\sum_{i=1}^{I} P_{iT}\Delta t - \sum_{j=1}^{J} \alpha_j(v_{jT} - \underline{v}_j).$$

The water balance equations (29) corresponds to dynamic constraints (6). The linear inequality (28) and box constraints (30)–(30) corresponds to static constraints (7).

4.2. Experimental set-up

We test our method on a power system containing $I = 4$ hydro plants and $J = 4$ reservoirs. The power system is located in Québec, Canada and has an installed capacity of 1,572 MW. The reservoir system has a total storage capacity of 3,710 hm^3. Fig. 6 shows the structure of the hydrosystem. The characteristics of each hydro plant and reservoir are summarized on Tables 1 and 2, respectively. Three hyperplanes ($h = 1, 2, 3$) are used to describe each concave and piecewise linear hydroelectric generation functions. The planning horizon is discretized in $T = 52$ time periods and a time step of $\Delta t = 168$ hours is used.

We implemented two optimization model. The first model solved the DEP directly and the second model solves the same problem using the PHA with subtree decomposition. Both models were implemented in object-oriented C++ using the ILOG CPLEX/Concert library version 12.5.1 and solved using the barrier solver in parallel mode. The stopping condition used in this experiment is $\epsilon = 10^{-3}$. We used $\rho_0 = 10^{-4}$ and $\mu = 1.1$.

Two different methods were implemented to build scenario subtrees. The first method correspond to Algorithm 1. The second method is based on a random selection of scenarios. Both methods are implemented in MATLAB R2013a.

All the numerical results presented in this paper were obtained using a personal computer running on Ubuntu 12.04 with a AMD Phenom II X6 2.8 GHz processor and 6 GB of RAM.

4.3. Scenario tree

In this experiment, we represent the hydrological stochastic process $\{\xi_t : t = 1, ..., T\}$ using the scenario tree shown on Fig. 7. This tree contains 500 scenarios, and 16,275 nodes. It was generated from 1,000 input synthetic time series using the SCENRED2 package which is part of the General Algebraic Modeling System (GAMS) version 23.9.3. The MPAR(1) stochastic model that was used for generating synthetic time series was tuned using historical data observed during the period 1962–2003. Table 3 shows the decomposition schemes that are considered in this experiment.

Table 1: Characteristics hydro plants

Plant	Maximum power output	Maximum turbined outflow
	(MW)	(m^3 s^{-1})
1	259	315
2	404	375
3	647	467
4	262	485

Table 2: Characteristics reservoirs

Reservoir	Minimum storage	Maximum storage	Initial storage	Production factor
	(hm^3)	(hm^3)	(hm^3)	(MWh/hm^3)
1	952	2,710	1,831	1,091
2	1,403	1,878	1,840	872
3	2,260	3,720	3,645	562
4	129	147	144	158

Table 3: Decomposition schemes.

Scheme	$N_{\max}$	Number of clusters	Number of NAC
A	1	500	13,174
B	10	84	666
C	25	32	147
D	50	17	53
E	100	12	30
F	250	3	3

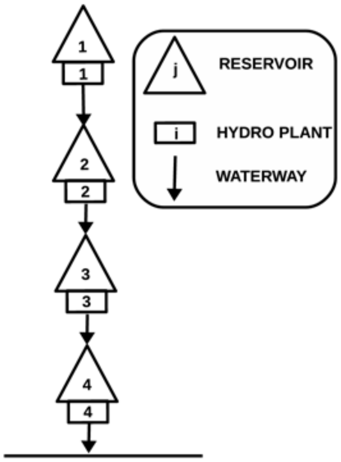

Figure 6: Reservoir system.

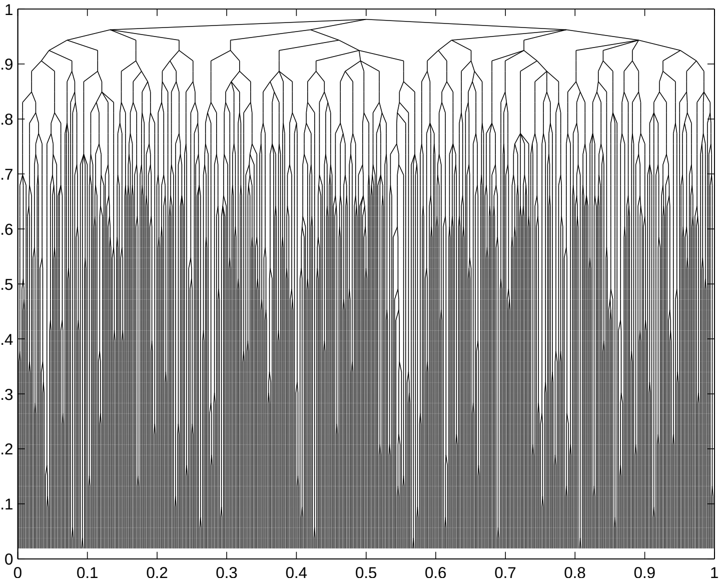

Figure 7: Scenario tree.

5. Results

Tables 4 show the numerical results obtained for decomposition schemes A to F. The best results were obtained using scheme D. This scheme used the least amount of RAM and lead to the fastest running time. For schemes A to D, the amount of RAM and the running time decreases as the subproblems size grow. This is due to the decreased number of Lagrange multipliers that need to be stored. Inversely, the amount of RAM increases from D to F as the subproblems size grow.

Table 4: Numerical results.

Scheme	Iterations	Total time	Time per iteration	RAM
		(seconds)	(seconds)	(MB)
A	51	11,361	222.8	1,489.6
B	34	1,498	44.0	313.6
C	21	468	22.3	168.0
D	1	16.3	16.3	145.6
E	1	17.4	17.4	156.8
F	1	24.7	24.7	240.8

6. Conclusions

In this article, we propose a new approach to enhance the performance of the PHA for solving MSPs defined on scenario trees. Instead of using traditional scenario decomposition scheme which is typically used with the PHA, we apply an multi-scenario decomposition scheme designed to minimize the total number of RNACs. A heuristic algorithm is proposed do achieve this efficiently. We test the proposed decomposition scheme on a RMP with stochastic inflows in Québec, Canada over a 52-period planning horizon with weekly time steps. Numerical results show that minimizing the total number of RNACs decreases the PHA running time substantially and enables to reduce the amount of memory required by the PHA.

References

Archibald, T.W., Buchanan, C.S., McKinnon, K.I.M, & Thomas, L.C. (1999). Nested Benders decomposition and dynamic programming for reservoir optimization. *Journal of the Operational Research Society*, 50, 468–479.

Benders, J.F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4, 238–252.

Birge, J.R. (1985). Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs. *Operations Research*, 33, 989–1007.

Birge, J.R., & Louveaux, F.V. (1988). A multicut algorithm for two-stage stochastic linear programs.*European Journal on Operational Research*, 34, 384–392.

Birge, J.R., & Louveaux, F.V. (2011). *Introduction to Stochastic Programming.* (2nd edition). New York: Springer.

Carpentier, P.-L., Gendreau, M., & Bastin, F. (2013a). Long-term management of an hydroelectric multireservoir system under uncertainty using the progressive hedging algorithm. *Water Resources Research*, 49, 2812–2827.

Carpentier, P.-L., Gendreau, M., & Bastin, F. (2013b). The Extended L-Shaped Method for Mid-Term Planning of Hydroelectricity Generation *IEEE Transactions on Power Systems*, (Submitted).

Crainic, T.G., Fu, X., Gendreau, M., Rei, W., Wallace, S.W. (2011). Progressive hedging-based metaheuristics for stochastic network design. *Networks*, 58, 114–124.

Crainic, T.G., Hewitt, M., Rei, W. (2012). Scenario Clustering in a Progressive Hedging-Based Meta-Heuristic for Stochastic Network Design. *Publication CIRRELT-2013-52*, 1–25.

Dantzig, G.B. (1955). Linear programming under uncertainty. *Management Science*, 3, 197–206.

Dupačová, J. (2002). Applications of stochastic programming: Achievements and questions. *European Journal of Operational Research*, 140, 281–290.

Gassman, H.I., & Ziemba, W.T. (2013). *Stochastic Programming: Applications in Finance, Energy, Planning and Energy.* World Scientific.

Gonçalves, R.E.C., Finardi, E.C., Da Silva, E.L. (2011). Applying different decomposition schemes using the progressive hedging algorithm to the operation planning problem of a hydrothermal system. *Electric Power Systems Research*, 83, 19–27.

Haugen, K.K., Lokketangen, A., & Woodruff, D.L. (2001). Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. *European Journal of Operational Research*, 132, 116–122.

Heitsch, H., Römisch, W., & Strugarek, C. (2006). Stability of multistage stochastic programs. *SIAM Journal on Optimization*, 17, 511–525.

Heitsch, H., & Römisch, W. (2009). Scenario tree modeling for multistage stochastic programs. *Mathematical Programming Series A*, 118, 371–406.

Høyland, K., & Wallace, S.W. (2001). Generating Scenario Trees for Multistage Decision Problems. *Management Science*, 47, 295–307.

Kallrath, J., Pardalos, P., Rebennack, S., Scheidt (2009). *Optimization in the Energy Industry.* Berlin: Springer.

King, A.J., & Wallace, S.W. (2012). *Modeling with Stochastic Programming.* New York: Springer.

Küchler, C., & Vigerske, S. (2007). Decomposition of Multistage Stochastic Programs with Recombining Scenario Trees. *Stochastic Programming E-Print Series.*

Labadie, J.W. (2004). Optimal Operation of Multireservoir Systems: State-of-the-Art Review. *Journal of Water Resources Planning and Management*, 130, 93–111.

Laporte, G., & Louveaux, F.V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13, 133–142.

Latorre, J.M., Cerisola, S., & Ramos, A. (2007). Clustering algorithms for scenario tree generation: Application to natural hydro inflows. *European Journal of Operational Research*, 181, 1339–1353.

Mulvey, J.M., Rosenbaum, D., & Shetty, B. (1997). Strategic financial risk management and operations research. *European Journal of Operational Research*, 97, 1–16.

Mulvey, J.M., & Vladimirou, H. (1991). Applying the progressive hedging algorithm to stochastic generalized networks. *Annals of Operations Research*, 31, 399–424.

Nocedal, J., & Wright, S. (2006). *Numerical Optimization.* (2nd edition). Springer Series in Operations Research and Financial Engineering. New York: Springer.

Pereira, M.V.F, & Pinto, L.M.V.G. (1991). Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52, 359–375.

Pflug, G.C. (2001). Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming Series B*, 89, 251–271.

Powell, W.B., & Topaloglu, H. (2003). Stochastic Programming in Transportation and Logistics. In Rusczynzki, A., & Shapiro, A. (Eds.), *Stochastic Programming* pp. 555–635. New York: Elsevier.

Wallace, S.W., & Ziemba, W.T. (2005). *Applications of Stochastic Programming.* SIAM.

Wallace, S.W., & Fleten, S.-E. (2003). Stochastic Programming Models in Energy. In Rusczynzki, A., & Shapiro, A. (Eds.), *Stochastic Programming* pp. 637-677. New York: Elsevier.

Rani, D., & Moreira, M.M. (2010). Simulation-Optimization Modeling: A Survey and Potential Application in Reservoir Systems Operation. *Water Resources Management*, 24, 1107–1138.

Rockafellar, R.T., & Wets, R.J.B. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16, 119–147.

Rusczynzki, A. (2003). Decomposition methods. In Rusczynzki, A., & Shapiro, A. (Eds.), *Stochastic Programming* pp. 141–211. New York: Elsevier.

Rusczynzki, A., & Shapiro, A. (Eds.), *Stochastic Programming*. New York: Elsevier.

Sagastizábal, C. (2012). Divide to conquer: decomposition methods for energy optimization. *Mathematical Programming Series B*, 134, 187–222.

Yeh, W.W.G. (1985). Reservoir Management and Operations Models: A State-of-the-Art Review. *Water Resources Research*, 21, 1797–1818.

Van Slyke, R.M., & Wets, R.B. (1969). L-shaped linear programs with applications to optimal control and stochastic programming.*SIAM Journal on Applied Mathematics*, 17, 638–663.

Watson, J.P. & Woodruff, D.L. (2011). Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems.*Computational Management Science*, 8, 355–370.

Wolf, C., & Kobertstein, A. (2013). Dynamic sequencing and cut consolidation for the parallel hybrid-cut nested L-shaped methods.*European Journal on Operational Research*, 230, 143–156.