EFFICIENT DIRECT SEARCH ALGORITHMS FOR SOLVING A LARGE-SCALE STOCHASTIC DYNAMIC PROGRAM

N. Saadouli[1]
[1]Gulf University for Science and Technology, Kuwait

Corresponding author: N. Saadouli, Gulf University for Science and Technology
PO Box 7207, Hawally 32093, Kuwait,
saadouli1971@yahoo.fr

Abstract. Many general-purpose decomposition techniques and heuristics have been introduced to minimize the computational cost of solving large models. However, many models tend to remain utterly large and costly prompting the application-oriented practitioner to exploit the structure of the specific problem at hand and devise decomposition techniques and solution algorithms that render efficient solutions. In this paper, a large stochastic dynamic program is formulated to address the problem of managing a water reservoir under several operational constraints and subject to stochastic effects. By inspecting the structure of the problem, and under suitable assumptions on the interdependence among the stochastic variables, and given the finite range within which the variables lie, efficient solution algorithms were developed. The procedure considers the discretization of some of the key variables and searching for values of the decision variables that would maximize the objective set forth in the model. Furthermore, to avoid the need to specify functional forms for the stochastic variables, a simulation routine was embedded in the model's solution procedure, as well as a validation tool of the model's performance through the direct calculation of the system reliabilities and comparing them to the model solution's output.

1 Introduction and Literature Review

To solve large stochastic problems, several techniques have been proposed in the literature. These include heuristics, decomposition techniques, and in the case of stochastic programs, the use of stochastic dynamic programming (SDP). Dynamic programming is largely accredited to Bellman (1957). A useful introduction to dynamic programming can also be found in Nemhauser (1966). For a detailed review of stochastic optimization, heuristics and decomposition techniques, the reader is referred to Fouskakis and Draper (2002), Silver (2005) and Dominguez-Ballesteros et al. (2002), respectively. In the SDP case, modelers often run into the so-called *curse of dimensionality* whereby the problem becomes exponentially large as more stages and evaluations are required. To alleviate this problem, researchers have proposed various aggregation, decomposition, and hybrid simulation-decomposition techniques. Bean et al. (1987) reduce the size of the dynamic program by combining stage aggregation and disaggregation steps in one algorithm to develop bounds on the error from optimality. Higle et al. (1994) develop an algorithm that uses randomly generated (simulated) observations with a Bender's decomposition of the problem. This is especially useful if the distributions of the random variables cannot be specified precisely. Mulvey and Ruszczynski (1995) use a parallel decomposition algorithm to decompose the problem into subproblems that correspond to scenarios. The subproblems are then solved using a nonlinear interior point algorithm. Aggregation can be applied at several levels. Some methods aggregate the search space by reducing it using some limiting assumptions. Other more commonly used methods of aggregation reduce the scenario space whereby several scenarios of the original problem are grouped together as one scenario, see Rockefeller and Wets (1991). For a thorough survey on aggregation in optimization, see Rogers et al.(1991).

Since using scenario trees requires the discretization of the random variables distributions, one typically faces the challenge of determining the appropriate number of scenarios. Towards this end, Hoyland and Wallace (2001) propose a method that generates a limited number of discrete outcomes that satisfy specified statistical properties by minimizing the distance between the statistical properties of the generated outcomes and the specified properties. For systems with a large number of states, Philbrick Jr. and Kitandis (2001) apply accurate interpolation to numerical approximation of continuous cost-to-go functions to reduce the number of discrete states that must be evaluated by incorporating information on first and second order derivatives. Boesel et al. (2003) address the problem of finding the simulated system with the best expected performance when a heuristic search procedure has been applied in a simulation-optimization context by combining statistical subset selection and indifference-zone ranking procedures.

Despite the efficiency introduced by these approaches, many problems remain utterly large, where general-purpose algorithms may not be of the desired efficiency. In such cases, exploiting the structure of the problem may yield insights that render the solution more practical. The case study used to illustrate the procedure is that of the optimal operation of multipurpose water reservoirs. For a state-of-the-art review of the optimal operation of multireservoir systems, the reader is directed to refer to the Labadie (2004) and Yeh (1985) survey papers. This should pro-

vide the reader with the necessary background to appreciate the complexity of the problem and the magnitude of the interdependencies among the various variables. A more specific discussion of the case study is presented in Edirisinghe et al (2000).

In this paper, an integrated simulation-optimization approach is used in conjunction with decomposition techniques that exploit the problem structure to generate very efficient solution algorithms. Under lenient assumptions on the dependence between random variables, the solution cost is significantly reduced. In section 2, the problem is stated and the model is formulated. In section 3, the model's solution is validated through the use of simulation. Concluding remarks and future research outlooks finish the exposition.

2 Problem Statement

To illustrate the approach, we develop a stochastic dynamic programming model for the reservoir problem involving multiple periods representing 12 months of operation, see Saadouli and Edirisinghe (2005) for a detailed description of the development of the objective function and the constraints. The main source of randomness in the reservoir is the monthly water inflow to the reservoir. The downstream demand for irrigation water is prescribed *a priori* and thus it is not random. During any month, the randomness of inflow will be modeled by a sample of discrete outcomes, generated randomly subject to the history of inflows up until that month. In the sequel, we will describe how such samples are generated to develop a *scenario tree* of potential future inflow patterns. The degree of violation of a constraint is considered explicitly and is controlled. The state of the system is completely defined by the state space $(\mathscr{H}_{t-1}, S_{t-1})$. The history of inflows $\mathscr{H}_{t-1}$ provides the information on what scenario of inflows has realized up until period t. The model is to select the release R_t that would minimize the expected penalty cost, less energy benefits, relative to the inflows. The value function, denoted by $\phi_t(.)$, at the node $\mathscr{H}_{t-1}$ of uncertainty resolution is then defined by the following, for a given value of S_{t-1}:

$$\begin{aligned}
\phi_t(\mathscr{H}_{t-1}, S_{t-1}) = \min_{R_t} \quad & E_{I_t|\mathscr{H}_{t-1}}\left[\mathscr{F}(\mathscr{H}_{t-1}, S_{t-1}, I_t, R_t) + \phi_{t+1}(\mathscr{H}_t, S_t)\right] \\
& \text{s.}t. \\
& S_t + R_t = S_{t-1} + I_t \\
& S_t - SD = \delta_t^{SD} \\
& R_t - T_t = \delta_t^{D} \\
& S_t - (k - V) = \delta_t^{F} \\
& \nu - EG_t = \delta_t^{EG} \\
& S_T - S_0 = \delta^{S_0}, \quad \text{if } t = T \\
& R_t \geq 0.
\end{aligned} \tag{1}$$

Where,
$\mathscr{F}(\mathscr{H}_{t-1}, S_{t-1}, I_t, R_t)$ is the net cost function due to the penalties δ_t^{SD}, δ_t^{D}, δ_t^{F}, δ_t^{EG}, and $\delta_T^{S_0}$ and benefit due to the firm energy ν. Also, $E_{I_t|\mathscr{H}_{t-1}}[.]$ denotes the conditional expectation with respect to the random inflow I_t given $\mathscr{H}_{t-1}$.

2.1 Dynamic Programming with Independent Inflows Models

Solving the dynamic program (1) requires the solution of a nonlinear program at each node $\mathscr{H}_{t-1}$, for a specified S_{t-1}. This is an orenous task as the number of such nodes increases exponentially with the addition of periods and/or outcomes. Furthermore, the latter computation needs to be performed for every possible S_{t-1}, as determined by a suitable grid of values for S_{t-1}. In this section we present an efficient version of the above DP under the assumption that the monthly inflows are independent.

Assumption 2.1 *The monthly inflow at period t, namely the random variable I_t, is independent of the history of inflows, $\mathscr{H}_{t-1}$.*

2.1.1 Modeling Inflows in the DP model

Under assumption (2.1), note that random variables Q_{t-1} and I_t are stochastically independent. Therefore, the mean and variance of a distribution of the individual monthly inflows are determined as follows.

$$\begin{aligned}
& Q_t = I_t + Q_{t-1}, \quad t = 1, ..., 12 \\
& E[Q_t] = E[I_t] + E[Q_{t-1}], \quad t = 1, ..., 12 \\
& Var[Q_t] = Var[I_t] + Var[Q_{t-1}], \quad t = 1, ..., 12.
\end{aligned} \tag{2}$$

This yields the following mean and variance for I_t:

$$\begin{aligned} &E[I_t] = E[Q_t] - E[Q_{t-1}], \quad t = 1,...,12 \\ &Var[I_t] = Var[Q_t] - Var[Q_{t-1}], \quad t = 1,...,12. \end{aligned} \tag{3}$$

Note here that we don't have a description of the distribution of the monthly inflows, although lognormal distributions were fitted for Q_t in Edirisinghe et al. (2000). There are two approaches by which one is able to generate outcomes for I_t. The first is to assume a distribution for the inflow with the mean and variance defined by (3). For example one can assume a uniform distribution with 3σ limits, i.e.

$$I_t := U(\max(0, E[I_t] - 3\sqrt{Var[I_t]}), E[I_t] + 3\sqrt{Var[I_t]}),$$

or a lognormal distribution, i.e.

$$I_t := \mathscr{LN}(E[I_t], Var[I_t]).$$

A more convenient approach where we do not need an explicit functional form for the individual monthly inflows is to simulate the distribution of I_t from the distributions of Q_t and Q_{t-1} which we have already determined to be lognormally distributed. A scenario branch at a particular node in period t in this case is generated by considering the lower and upper limits,

$$\begin{aligned} &U_t = E[I_t] + s(Var[I_t])^{1/2} \\ &L_t = Max(0, E[I_t] - s(Var[I_t])^{1/2} \\ &\text{where } s \text{ represents the number of standard deviations around the mean.} \end{aligned} \tag{4}$$

With these limits and the assumption that this is a uniform distribution, we can simply divide this interval $[L_t, U_t]$ into n subintervals Δ_t^n, where n is the number of outcomes in each period. The midpoint of each subinterval Δ_t^n would be the inflow at that particular outcome. If we assume a uniform distribution, then each outcome is equally likely with probability $1/n$. If the lognormal distribution is assumed, then we generate a large number of random variates and place them into appropriate subintervals. A frequency count divided by the total number of random variates generated would yield the needed probabilities. The final approach is to simulate the individual monthly inflows from the cumulative monthly inflows distributions which have been determined to be lognormal. To generate a scenario branch at a particular node of a particular period, we generate a random variate q_{t-1} from Q_{t-1} and a random variate q_t from Q_t, and define variate i_t (for I_t) as $q_t - q_{t-1}$ provided that $q_t > q_{t-1}$. The variates i_t thus generated is placed in appropriate subintervals. A frequency count is made, and the probability of each outcome is computed as the frequency of variates in each subinterval divided by the total number of variates generated.

2.1.2 Model Description

Under Assumption (2.1), the outcomes at period t are determined independently of the historical scenario being followed until period t. Thus, the outcomes and their corresponding probabilities of occurrence in each node in period t are identical. Therefore, the DP model in this can be presented as follows:

$$\begin{aligned} \phi_t(S_{t-1}) = \quad & \min_{R_t} \quad E_{I_t}\left[\mathscr{F}(S_{t-1}, I_t, R_t) + \phi_{t+1}(S_t)\right] \\ & \text{s.}t. \\ & S_t + R_t = S_{t-1} + I_t \\ & S_t - SD = \delta_t^{SD} \\ & R_t - T_t = \delta_t^{D} \\ & S_t - (k - V) = \delta_t^{F} \\ & v - EG_t = \delta_t^{EG} \\ & S_T - S_0 = \delta^{S_0}, \quad \text{for} \quad t = T \\ & R_t \geq 0. \end{aligned} \tag{5}$$

Note that the value function does not depend on $\mathscr{H}_{t-1}$, and hence, for a given beginning storage S_{t-1}, the value function is identical for all nodes $\mathscr{H}_{t-1}$ in period t. This is a very important result since it implies that we only solve the model in (5) once in each period. For instance, for 12 periods with with 10 outcomes, instead of solving $\sum_{i=0}^{12} 10^i$ problems corresponding to the number of nodes, we only have to solve 12 problems of the format (5) for each value of S_{t-1}. We also propose to solve the minimization in (5) by a grid search on R_t, thereby avoiding a possibly difficult nonlinear programming procedure.

2.1.3 Solution Algorithm

Given a set of discretized beginning storages S_{t-1}^n, for $n = 1,\ldots,N$, and a set of discretized releases R_t^j, for $j = 1,\ldots,J$, we compute the value function for one node in period 12 and copy it to all other nodes in the period.
Step 0:Initialization

1. Set reservoir size and maximum firm energy level
2. Obtain inflow data (mean, standard deviation, and number of outcomes per period; determine probability of each outcome if a distribution other than the uniform distribution is used)
3. Set t= T-1, where T= number of periods to be considered in the model

Step 1:DO WHILE n $\leq$ N

1. Set $S_{t-1} = S_{t-1}^n$, Let $\phi^*(S_{t-1}) = M$, where M denotes a large positive number.

Step 2: DO WHILE j $\leq$ J

1. evaluate $\phi_t(S_{t-1}, R_t^j)$ as described in (5)
2. for $\phi^*(S_{t-1}) \quad < \quad \phi_t(S_{t-1}^i, R_t^j)$
   let $\phi^*(S_{t-1}) \quad = \quad \phi_t(S_{t-1}^i, R_t^j)$ and $R_t^*(S_{t-1}) = R_t^j$.

END DO
END DO
Step 3:

1. Set t=t-1.
2. if $t < 0$, STOP,
   else go to Step 1

2.2 Dynamic Programming Model with Restricted dependence

In this section, cumulative monthly inflows are explicitly considered in the decision policy. However, if we were to consider a full scale dependence, we would end up with a problem that is no easier to solve than the original stochastic dynamic program. The restrictive nature of the dependence structure of inflows is described in the following assumption.

Assumption 2.2 *The individual monthly inflow at period t, random variable I_t, is stochastically dependent only on the cumulative inflows up to period t, Q_{t-1}.*

2.2.1 Model Description

Under Assumption (2.2), for a given cumulative inflow Q_{t-1}, a given beginning storage level S_{t-1}, and a given release R_t, the dynamic programming recursive formulation can be rewritten as follows:

$$\begin{aligned}
\phi_t(Q_{t-1}, S_{t-1}) = \quad & \min_{R_t} E_{I_t|Q_{t-1}} [\mathscr{F}(q_{t-1}, S_{t-1}, I_t, R_t) + \phi_{t+1}(Q_t, S_t)] \\
& \text{s.}t. \\
& S_t + R_t = S_{t-1} + I_t \\
& S_t - SD = \delta_t^{SD} \\
& R_t - T_t = \delta_t^D \\
& S_t - (k - V) = \delta_t^F \\
& \nu - EG_t = \delta_t^{EG} \\
& S_T - S_0 = \delta^{S_0}, \quad if \quad t = T \\
& q_t = q_{t-1} + I_t \\
& R_t \geq 0.
\end{aligned} \tag{6}$$

Given a realization q_{t-1} of Q_{t-1}, we have

$$\begin{aligned}
& I_t|q_{t-1} = Q_t - q_{t-1}, \\
& E[I_t|q_{t-1}] = E[Q_t] - q_{t-1}, \\
& Var[I_t|q_{t-1}] = Var[Q_t].
\end{aligned}$$

Thus to generate a scenario tree for I_t in this case, one needs the distribution of $I_t|q_{t-1}$. This is due to the fact that for each instance of q_{t-1}, one needs to generate a set of outcomes from the distribution of $I_t|q_{t-1}$. This will, however,

lead to a DP which is as complex and computationally tedious as that in the general setting. Note that we have a given distribution for Q_t and we need only generate one sample from that distribution. Therefore, if we generate the scenario tree where the outcomes are cumulative monthly inflows, we get the same outcomes for each node in a particular period. However, the individual monthly inflows are dependent on the level of cumulative inflows of the previous period and are thus different in each outcome. This is also evident from the DP recursive formula in (6), as the value function is only dependent on the level of cumulative inflows realized thus far. From these observations, if we generate the scenario tree where the outcomes in each period are based upon the cumulative inflows, then we need only to solve the problem for one node in each period. This is a tremendous advantage because it reduces the computational burden from solving $\sum_{i=0}^{12} 10^i$ problems to only solving 12 such problems.

2.2.2 Solution Algorithm

Given a set of discretized cumulative inflows Q_{t-1}^n, $n = 1,\ldots,N$, a set of discretized beginning storages S_{t-1}^j, $j = 1,\ldots,J$, and a set of discretized releases R_t^l, $l = 1,\ldots,L$, the algorithm can be described as follows.
Step 0:Initialization

1. Set reservoir size and maximum firm energy level
2. Obtain inflow data (mean, standard deviation, and number of outcomes per period)
3. Generate outcomes and determine probabilities for each period from the provided distribution
4. Set t= T-1, where T= number of periods to be considered in the model

Step 1: DO WHILE n≤N and j≤J

1. Set $Q_{t-1} = Q_{t-1}^n$, and $S_{t-1} = S_{t-1}^j$, Let $\phi^*(Q_{t-1}, S_{t-1}) = M$, where M denotes a large positive number.

Step 2: DO WHILE l≤L

1. evaluate $\phi_t(, Q_{t-1}, S_{t-1}, R_t^l)$ as described in (6)
2. if $\phi^*(Q_{t-1}, S_{t-1}) < \phi_t(Q_{t-1}, S_{t-1}, R_t^l)$
   let $\phi^*(Q_{t-1}, S_{t-1}) = \phi_t(Q_{t-1}, S_{t-1}, R_t^l)$ and $R_t^*(Q_{t-1}, S_{t-1}) = R_t^l$.

END DO
END DO
Step 3:

1. Set t=t-1.
2. if $t < 0$, STOP,
   else go to Step 1

3 Simulation and Model Validation

The input parameters input to the model are: reservoir size K, the firm energy level ν, the mean and standard deviation of the cumulative inflows, the water targets, and the release policy $R_t^*(S_{t-1})$. The output is the simulated reliabilities, α_s, θ_s, β_s, γ_s, ρ_s. The simulation model also outputs the average energy generated each period, the average potential energy generated in each period, an average simulated firm energy level, and the firm energy reliability measure. Then a simulation of the release policy is carried out to gain some insight into the complex structure of the problem and to also validate the model's solution.
We do 100,000 simulation runs, by changing the inflow sequence for 12 months. At each iteration, a release is selected from the optimal release policy depending on the period and the beginning storage level of that period, S_{t-1}. The simulated reliabilities α_s, θ_s, β_s, remain consistently above 0.9. This indicates that the model does take into account these secondary objectives and they are all satisfied with very high probabilities.

The potential energy generation is defined as the energy generated if all available water above the dead storage volume is released. Both the actual and potential energy generation reliabilities remain consistently high indicating that the model is indeed maximizing returns for the energy authority through high firm energy levels. More importantly, the actual energy generation and the potential energy generation reliabilities are almost equal, indicating that the release policy chosen by the model maximizes firm energy benefits without bringing the level of water below the dead storage level.

Recall that one of the reasons we decided to use a reliability level on the firm energy constraint was that determination of its distribution was a complex matter as it requires the knowledge of the covariance terms among $EG_t, t = 1,...,12$. In the sequel, we analyze the release policies generated by the restrictive dependence DP model. We first describe how random outcomes of different scenarios are generated and also how the discretized cumulative monthly inflows used in the model are generated. Then a simulation of the release policy is carried out to gain some insight into the complex structure of the problem and to also validate the model's solution. The release

policy of the current model is no longer a function of the beginning storage S_{t-1} only. The release in each period t depends on both the beginning storage S_{t-1} and the cumulative inflow of the previous period Q_{t-1}.

Furthermore, we notice a phenomenon that was not apparent in the independent inflows DP model of Section 2.1. For a fixed beginning storage level, the optimal release level increases up to a certain critical cumulative inflow, Q^c_{t-1} and then starts decreasing as the cumulative inflow level Q_{t-1} increases. This is an interesting remark as it highlights one of the important tasks that the model is accomplishing. This is due to the fact that the model uses the knowledge of the value of the cumulative inflows in deciding which optimal release to choose. Before that critical point Q^c_{t-1}, the releases increase to maximize the energy generation. However, once that level is exceeded, the model realizes that the probability of large inflows in the coming periods decreases and so the releases are chosen to be smaller so as not to hamper energy generation in coming periods.

The simulation routine for this model is not too different form that of the previous model. The inputs consist of the reservoir size, the firm energy level desired, and the optimal release policy obtained from the solution of the model. The outputs include the simulated reservoir reliabilities, the simulated firm energy reliability, and the frequency files for the distribution of L_{fe} and S_{12}. Consider the case where the reservoir size is 2,000(MCM) and the firm energy level is required to be 4,000. The lowest level attained was 0.65 which is still high for a flood storage reliability. Two important observations are due in this regard. The first is that the simulated energy generation reliability $\gamma_{L_{fe}}$ remains very high (> 0.9), in fact in 7 of the 12 monthly periods, $\gamma_{L_{fe}} = 1$. This indicates that the model does cater to the primary concern of the reservoir management which is the generation of energy. The second remark is that the levels of the reliabilities of the potential energy and actual energy generation are close. This indicates that although the release policy is non-anticipative, the information provided by the cumulative inflow, the releases are fairly close to the potential release defined as the release of available water above the dead storage level *SD*. This validates the claim made in the preceding section that the knowledge of the level of the cumulative inflow helps the model in "estimating" the level of inflow in subsequent periods. Recall that in generating the scenario tree, each branch corresponds to a vector of inflows, say $\hat{I}^G$. In the simulation, we generate a vector of equal size $\hat{I}^R$, and that decides which of the nodes of the coming stage will be the node of reference, i.e. which release policy to use for the next stage. Basically, the branch of the scenario tree that most closely resembles the randomly generated vector of inflows is the scenario we follow to get the release policy. There are potentially several methods of measuring the distance between vectors. The most common being the *Euclidean* distance, ED, computed as follows:

$$ED = [\sum_{i=t}^{t+p} (I_i^G - I_i^R)^2]^{\frac{1}{2}}.$$

The computational efficiency of the three stochastic programming with recourse models depend on the number of random inflow outcomes considered in each period, and on the fineness of the discretization of beginning storage and beginning cumulative inflow vectors. This is understandable since these two parameters increase the search space exponentially. Despite this fact, solution of the independent inflows model, for the case of 50 outcomes per period and 50 discrete points for the search space arrays, takes about 12 seconds including generating the scenarios. The restricted dependent inflows model takes about 6 minutes, because we are adding a new search dimension, i.e., cumulative inflows. The solution algorithms prove to be quite efficient and accurate. Essentially, the algorithms reduce the solution burden from solving o^n nonlinear problems to solving only n evaluation problems, where n is the number of periods and o is the number of outcomes considered per period. This amounts to a drastic reduction in the computational time, which is certainly desired in a situation where a solution is required in an operationally feasible time frame. The simulation analysis indicates that the reliabilities are maintained at significantly high levels.

4 Concluding Remarks and Future Research

In situations where a preliminary solution is desired, it can be concluded that the restricted dependent inflows model would be the preferred choice because of its solution efficiency, and its ability to incorporate partial inflow dependence information by considering a large number of outcomes in each period. While the proposed models provide robust results, their focus is limited to monthly decision periods and without assuming the individual inflows dependence which would typically be the case. An operational model would have to take into account a decision period much shorter than a month, and would need to have the flexibility of providing better solutions as random events unfold. Furthermore, it should include the underlying dependence between the inflows. This will make the model much more challenging to solve efficiently; however, using adequate aggregation and decomposition techniques may yield operationally feasible solutions. An other alternative is the *rolling horizon* approach, where the model is re-solved at the end of each operational period, the model being revised with new observations of inflow data. These avenues and others are the subject of future research.

5 References

[1] Bean J. C., Birge J. R.,and Smith R. L.: *Aggregation in Dynamic Programming*. Operations Research, 1987, 35(2): 215-220

[2] Bellman R. E.: *Dynamic Programming*. Princeton University Press, Princeton, N.J., 1957.

[3] Boesel J., Nelson B. L.,and Kim S.: *Using Ranking and Selection to "Clean-up"' After Simulation Optimization*. Operations Research, 2003, 51(5): 814-825

[4] Dominguez-Ballesteros B., Mitra G., Lucas C. ,and Koutsoukis N-S.: *Modeling and Solving Environments for Mathematical Programming (MP): a Status Review and New Directions*. Journal of the Operational Research Society, 2002, 53: 1072-1092

[5] Edirisinghe N.C.P., Patterson I., and Saadouli N.: *Capacity Planning Model for a Multipurpose Water Reservoir with target-Priority Operation*. Annals of Operations Research, 2000, 100, 273-303.

[6] Fouskakis D.,and Draper D.: *Stochastic Optimization: A Review*. International Statistical Review, 2002, 70(3): 315-349

[7] Higle J. L., Lowe W. W.,and Odio R.: *Conditional Stochastic Decomposition: An Algorithm Interface for Optimization and Simulation*. Operations Research, 1994, 42(2): 311-322

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

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

[10] Mulvey J. M. ,and Ruszczynski A.: *A New Decomposition Method or Large-Scale Stochastic Optimization*. Operations Research, 1995, 43(3): 477-490

[11] Nemhauser, G.L.: *Introduction to Dynamic Programming*. John Wiley, New York, 1966.

[12] Philbrick Jr C.R. ,and Kitandis P.K.: *Improved Dynamic Programming Methods or Optimal Control of Lumped-Parameter Stochastic Systems*. Operations Research, 2001, 49(3): 398-412

[13] Rockafeller R. T. , Roger J. ,and Wets J. B.: *Scenario and Policy Aggregation in Optimization Under Uncertainty*. Mathematics of Operations Research, 1991, 16(1): 119-147.

[14] Roger D. F, Plante R. D., Wong R. T. ,and Evans J. R.: *Aggregation Techniques and Methodology in Optimization*. Operations Research, 1991, 39: 553-582.

[15] Saadouli N. ,and Edirisinghe N.C.P.: *Multi-Stage and Dynamic Stochastic Models for the Operation of a Multipurpose Water Reservoir*. WSEAS Transaction on Business and Economics, 2006, 3(1): 1-8

[16] Silver E. A.: *An overview of Heuristic Solution Methods*. Journal of the Operational Research Society, 2004, 55: 936-956

[17] Yeh W. W-G: *Reservoir Management and Operations Models: A State-of-the-Art Review*. Water Resources Research, 21(12), (1985), 1797-1818.