Abstract Number: 020-0226

Risk-Reward Analysis in Stochastic Dynamic Programming

Preetam Basu
Assistant Professor, Operations Management Group
Indian Institute of Management Calcutta
Diamond Harbor Road, Kolkata 700104, India
preetamb@iimcal.ac.in; Tel.: +91-33-2467-8300

Suresh K. Nair
Professor, Operations and Information Management
School of Business, University of Connecticut, Storrs, CT 06269
suresh.nair@business.uconn.edu; Tel.: +1-860-486-1727

Abstract: Stochastic dynamic programming models are extensively used for sequential decision making when outcomes are uncertain. These models have been widely applied in different business contexts such as inventory control, capacity expansion, cash management, etc. The objective in these models is to deduce optimal policies based on expected reward criteria. However, in many cases, managers are concerned about the risks or the variability associated with a set of policies and not just the expected reward. Considering risk and reward simultaneously in a stochastic dynamic setting is a cumbersome task and often difficult to implement for practical purposes. Here we develop heuristics that systematically track the variance and the average reward for a set of policies, which are then utilized to construct efficient frontiers. We apply our heuristics to the inventory control model. Our heuristics perform creditably in providing efficient risk-reward curves.

(Keywords: Stochastic Dynamic Programming, Risk-Reward frontiers, Mean-Variance Trade-offs, Heuristics)
1: Introduction

Stochastic dynamic programming models have been extensively used in solving problems that have uncertainty as well as temporal aspects. These models provide sequential optimal decisions where present actions are taken in consideration of future outcomes. The models are characterized by decision epochs, states, actions, rewards and probability functions. At a specified point in time, which is referred to as a decision epoch, the decision maker observes the state of the system and based on that chooses an action. The action produces an immediate reward and the system moves to a different state in the next time-period according to a probability distribution. Stochastic dynamic programming models find wide applications amongst various problems that arise in different sections of an organization such as inventory control, capacity planning, advertising expenditures, cash management to name just a few.

The usual optimization criteria for stochastic dynamic programming models are to maximize the expected value of the limiting average reward per unit time or the sum of discounted rewards in a finite or infinite-horizon setting. The focus of majority of these models is on optimization of average reward criteria. However, using expected total reward criteria may yield optimal policies that are unacceptable to a risk-sensitive decision maker. In many cases, managers are concerned about the risks or the variability associated with a set of policies and not just the expected reward. Instead of identifying one policy to achieve the stated objective, managers are often interested in obtaining set of policies at different levels of risk sensitivity. Then based on their risk-tolerance level decisions could be taken that provide the maximum expected rewards.

Risk-reward trade-offs are an essential component of many managerial decisions. Variance of the outcomes about the expected value is a widely used measure of risk in portfolio theory. Investors use variance to measure the risk of a portfolio of stocks. The basic idea is that variance is a measure of volatility and the more a stock’s returns vary from the stock's average return, the more volatile is the stock. Portfolios of financial instruments are chosen to minimize the variance of the returns subject to a level of expected return or vice versa to maximize expected return subject to a level of variance of the return. This paradigm was first introduced by Markowitz’s (1959) mean-variance analysis for which contribution he was honored with the Nobel Prize in Economics. In many operational decisions also managers are interested in the mean-variance trade-offs. They do not want to focus on a single criterion rather they want to deduce a set of policies that give them optimal solutions at different levels of risk-tolerance. In start-up operations it is extremely important to ascertain the risks associated with a set of policies be its inventory strategies or advertising ventures and not just the expected return. Decisions such as capital investments or capacity expansion that have long-term repercussions can be detrimental if the focus is only on the expected returns.

Risk-sensitive optimization in stochastic dynamic programming setting has been subjects of some limited research. Markov Decision Process (MDP) models are one of the most widely used sequential decision models where the set of available actions, rewards and
transition probabilities depend only on the current state and not on states occupied in the past. MDP models have found wide applications as they can efficiently model most realistic sequential decision-making problems. The usual optimization criteria for MDPs are to maximize the expected value of the sum of discounted rewards. There is a small stream of research that employs risk-sensitive optimization criteria in MDPs. White (1974) used Lagrange multipliers to include probabilistic constraints on variances in the optimization of mean performance. Howard and Mathesoon (1972) considered the maximization of certain equivalent reward generated by MDP with constant risk sensitivity. Kawai (1987) analyzed randomized policies that minimize the variance of the reward in a transition among the policies that lead to a certain level of expected reward for undiscounted MDPs. Filar et al. (1989) studied mean-variance trade-offs in MDPs with both expected limiting average and discounted total return criteria. They formulated appropriate nonlinear programs in the space of state-action frequencies to analyze the corresponding variance penalized MDP models. Baykal-Gursoy and Ross (1992) considered time-average MDPs with finite state and action spaces and introduced two definitions of variability, viz., expected time-average variability and time-average expected variability. The authors used these criteria to penalize the variance in a stream of rewards. Sobel (1994) analyzed undiscounted MDPs to generate Pareto optima in the sense of high mean and low variance of the stationary probability distribution of the reward. White (1988) provides a survey on risk-sensitive optimization in finite MDPs. In the stream of literature discussed above the variance-penalized MDPs are formulated as mathematical programs with linear constraints and nonlinear objective function. However, as Filar et al. point out these formulations lead to formidable mathematical difficulties and are difficult to implement for practical purposes.

Here we come up with a simple easy-to-use heuristic to construct near-optimal risk-reward curves. Using our heuristic managers can come up with policies that lead to maximum average reward at a particular level of risk. The heuristic can be easily applied to any finite-horizon stochastic dynamic programming model and gives a manager an useful tool with which he can figure out a set of decisions that maximizes profit at a certain level of risk. In a standard finite-horizon dynamic programming problem, information about the expected reward of a set of decisions is carried along and is used in solving the problem by backward induction algorithm. In our heuristic, we carry information about the variance of a set of decisions in addition to expected rewards and this information about the variance is utilized in selecting the optimal policies. Using our heuristic a risk-reward curve can be easily drawn which can be used to ascertain risk exposure of a set of actions. Once the management identifies the risk exposure of a set of policies they can choose their actions based on the level of risk they are willing to accept.

2: Stochastic Dynamic Programming Formulation

Here we consider a finite-horizon stochastic dynamic programming model with \( T \) time-periods. At each time-period the system occupies a state denoted by \( s \). Let the set of possible system states be \( \mathcal{S} \) and the allowable actions at any state be \( \mathcal{A}_s \). At each time-period the decision-
maker chooses an action \( a(a \in A_s) \) and based on that receives a reward \( r_t(s,a) \) and the system state moves to the next time-period according to the probability distribution \( p(.|s,a) \). Let the terminal reward be given by \( r_T(s) \) and \( \beta \) be the discount factor. The following recursive functional equations specify the model:

\[
 f_t^T(s) = \max_{a \in A} r_t(s,a) + \beta \sum_{j \in S} p(j|s,a) f_{t+1}^T(j)
\]

The boundary condition is given by:

\[
 f_T^T(s) = r_T(s)
\]

In most of the stochastic dynamic programming models the objective is to maximize the expected sum of rewards. As we have pointed out before, in many cases, managers are interested in not only the expected return but also on the risk or the variability of the returns. Normally in a finite-horizon discrete-time dynamic programming problem, the information about the average reward for a set of optimal actions is carried at each state. This information is then used to compute the average reward at a subsequent state. Backward induction is an efficient solution methodology for these kinds of dynamic programming problems. It is the process of reasoning backwards in time, from the end of a problem to determine a sequence of optimal actions. It proceeds by first choosing the optimal action at the last decision epoch or the boundary of the problem. Then using this information, the optimal action at the second-to-last time of decision is determined. This process continues backwards until one has determined the best action for every possible situation at every point in time. In our heuristics, we use the backward induction algorithm to solve the dynamic programming model and in addition to carrying information about the average reward, we also keep track of the variance of a set of policies at each state. This variance information is utilized to obtain risk-reward trade-offs. Next, we explain the two heuristics in detail that provide us with an easy-to-implement mechanism to construct risk-reward curves.

3: Heuristic for Risk-Reward Curves

Here we develop a heuristic utilizing the variance information at each state for the possible actions. In the first heuristic, which we call RiskTrackr_MaxEfficiency, the focus is on identifying solutions that have high values of the ratio of expected reward to variance.

3.1: The RiskTrackr_MaxEfficiency Heuristic

In the RiskTrackr_MaxEfficiency heuristic our focus is on identifying solutions that have high values of the ratio of expected reward to variance and then picking the solutions that have high expected reward values. The average reward at any state \( s \), for a set of policies is captured by the functional value \( f_t^T(s) \). Here we need additional information about the risk associated with
those set of policies. We define risk as the variance of the possible outcomes at any state. We
denote the variance of the possible outcomes for a set of action $a$, at any state $s$, by $\gamma_a(s)$.

$\gamma_a(s)$ is given as follows:

$$\gamma_a(s) = \sum_{j \in S} p(j \mid s,a) \beta^2 v_{T+1}^T(j) + \sum_{j \in S} p(j \mid s,a)(r_j(s,a) + f_{i+1}^T(j))^2$$

$$- \sum_{j \in S} p(j \mid s,a)(r_j(s,a) + f_{i+1}^T(j)) \mu_{T+1}(s)$$  \hspace{1cm} (3)

where,

$$\mu_{T+1}(s) = r_j(s,a) + \sum_{j \in S} p(j \mid s,a)f_i^T(j)$$

The variance at any state $s$, is denoted by $v_i^T(s)$ and it is the variance corresponding to the
optimal action $a^*$.

In the RiskTrackr_MaxEfficiency heuristic, we use the backward induction algorithm to
solve the dynamic programming model and in addition to carrying information about the average
reward, we also keep track of the variance of a set of policies at each state. At $t = T$, i.e., at the
boundary, the variance at any state is zero since no actions are taken. Then at any preceding
decision epoch ($t = T-1, T-2, \ldots, 0$) the variance at any state for each of the possible actions is
calculated based on equation (3). This variance information is then used to evaluate the optimal
action at each state. Next, we explain how we determine the optimal action $a^*$ at each decision
epoch.

At each state of the problem, we develop a ratio based on average reward to variance
for each of the possible actions. We call this ratio “criteria” and denote it by $K_a(s)$.

$$K_a(s) = \frac{r_j(s,i) + \sum_{j \in S} p(j \mid s,a)f_{i+1}^T(j)}{\gamma_a(s)}$$  \hspace{1cm} (4)

We then solve the dynamic programming model repeatedly for $j = 1, 1 + y, 1 + 2y, \ldots, X$
where $j$ is the collection of actions that give the top $j$ criteria values from the set of all possible
actions at any state denoted by $A$. Here $X$ is the total number of elements in the action set and
$y$ is the step size used in incrementing $j$. At any state we rank all the actions descending order
based on the criteria value using (4). We then pick the top \( j \) actions. Then out of the top \( j \) actions, we pick the action that gives the maximum reward.

We first solve the model for \( j = 1 \), i.e., at each state we only consider the actions that give the highest criteria value and then from those actions pick the one that gives the maximum average reward. As we solve the models repeatedly for different values of \( j \) starting from \( j = 1 \), we keep track of the average reward earned in each state and store that information in \( G^T_t[s] \). For a subsequent value of \( j, (j = j + y, j + 2y, \ldots) \), we check that the average reward earned at each state is at least as large as that has been earned for a lower value of \( j \). The average reward and variance for the optimal actions are stored in \( f^T_t(s) \) and \( v^T_t(s) \) respectively. At \( t = 0 \), for a particular value of \( j \), \( f^T_0(s) \) and \( v^T_0(s) \) gives the average reward and variance for a sequence of optimal actions executed at each of the time periods based on the top \( j \) criteria values. We store the average reward and variance value for each \( j \) in \( E_j(j) \) and \( Var_j(j) \) respectively. We continue the above process until \( j = X \). \( E_j(j) \) and \( Var_j(j) \) \( \forall j = 1, 1 + y, \ldots, X \), obtained from the RiskTrackr_MaxEfficiency heuristic is to construct the efficient frontier. Solving the stochastic dynamic programming model repeatedly for \( j = 1, 1 + y, 1 + 2y, \ldots, X \) using the RiskTrackr_MaxEfficiency heuristic gives us a set of policies that consider both the expected reward and the associated variance. Since at each state we pick actions that lead to highest value of the ratio of average reward to variance, we are in turn considering the actions that are the most efficient at each state. Then we are picking the action that has the highest average reward. Therefore, out of all the possible actions we are picking the ones that are the most efficient, i.e., the ones that give the highest average reward with respect to variance. Hence as we increase \( j \), we are increasing the risk exposure of the model and at a higher risk-exposure we are again picking the action that has the maximum average reward. This process in turn leads to outcomes starting from very low risk-exposure and increasing to the maximum risk-exposure.

Next, we apply RiskTrackr_MaxEfficiency to one of the most widely used applications of stochastic dynamic programming modeling, namely, the inventory control model to check the performance of our heuristics in constructing efficient risk-reward curves. We do so by comparing the risk-reward frontier obtained by our heuristic against the risk-reward solutions from all enumerated outcomes of the inventory models.

### 3.2: Application RiskTrackr_MaxEfficiency: Inventory Control Model

Stochastic dynamic programming has been extensively used in inventory control models. These models are widely applicable in a variety of business problems such as determining order-sizes in a supply chain set-up and even for managing cash balances. Next we present a simplified version of that model.
Each time period the manager looks at the current inventory or stock-on-hand and decides whether or not to order additional stock from a supplier. We look at a finite-horizon model with \( T \) time-periods. The state of the system at any time-period is given by \( z \) which denotes the stock-on-hand. We assume demand for the product at each time-period, \( d \), is random and independent and follows a probability distribution \( p(d) \). Let \( q \) be the units of stock ordered at each time-period and \( \hat{q} \) be the maximum possible order-size. The unit selling price is given by \( s \) and the unit purchase cost of the product is \( u \) and the unit holding cost is \( h \). The salvage value for unit unsold inventory is given by \( \lambda \). \( f_t^T(z) \) is the maximal net present value of being in state, \( z \), when optimal actions are taken in each time period from period \( t \) to \( T \). The following recursive functional equations specify the model:

\[
f_t^T(z) = \max_q \sum_d p(d)(sd - uq - h\tilde{z} + f_{t+1}^T(z - \tilde{d} + q))
\]

where \( \tilde{d} = \text{Minimum}(d, z) \) and \( \tilde{z} = \text{Maximum}(0, z - d) \)

The boundary condition is given by:

\[
f_T^T(z) = \lambda z
\]

Next we explain the steps in applying the RiskTrackr\_MaxEfficiency to the inventory control model (5)-(6). At each state of the system, we need to calculate the average reward and the variance for each order-size. The average reward at any state \( z \) is captured by the functional value \( f_t^T(z) \). The variance of the possible outcomes for any order-size \( q \), is given by \( \gamma_q(z) \) which is calculated as follows using equation (3):

\[
\gamma_q(z) = \sum_d p(d) \gamma_{t+1}^T(z - \tilde{d} + q) + \sum_d p(d)(sd - uq - h\tilde{z} + f_{t+1}^T(z - \tilde{d} + q))^2 - \sum_d p(d)(sd - uq - h\tilde{z} + f_{t+1}^T(z - \tilde{d} + q)) \mu_{t+1}(z)
\]

where, \( \mu_{t+1}(z) = sd - uq - h\tilde{z} + \sum_d p(d) f_t^T(z - \tilde{d} + q) \)

At each state of the problem, we also need to calculate the “criteria” value for any order-size which is denote it by \( K_q(z) \) and is obtained using equation (4) as follows:
We solve the inventory control model repeatedly for \( j = 1, 1 + y, 1 + 2y, \ldots, \hat{q} \), where \( j \) is the collection of actions that give the top \( j \) criteria values from the set of all possible order-sizes at any state. After ranking the order-sizes in a descending order based on the criteria value using (8) we pick the top \( j \) actions. Then out of the top \( j \) actions, we pick the action that gives the maximum reward. As we solve the models repeatedly for different values of \( j \) starting from \( j = 1 \), we keep track of the average reward earned in each state and store that information in \( G^T_{j} [z] \). For a subsequent value of \( j \), \( (j = j + y, j + 2y, \ldots) \), we check that the average reward earned at each state is at least as large as that has been earned for a lower value of \( j \). The average reward and variance for the optimal actions are stored in \( f^T_{j} (z) \) and \( v^T_{j} (z) \) respectively. At \( t = 0 \), for a particular value of \( j \), \( f^T_{0} (z) \) and \( v^T_{0} (z) \) gives the average reward and variance for a sequence of optimal actions executed at each of the time periods based on the top \( j \) criteria values. We store the average reward and variance value for each \( j \) in \( E_{j} (j) \) and \( Var_{j} (j) \) respectively. We continue the above process until \( j = \hat{q} \).

Now, we need to check the performance of our heuristic in constructing efficient risk-reward curves. We do so by comparing the risk-reward frontier obtained by our heuristic against the risk-reward associated with all enumerated outcomes of the inventory control model. For the performance evaluation, we picked a 6-period problem with maximum order-size equal to 20 in each state. Here the unit cost \( u = \$3 \) and the unit selling price \( s = \$10 \). We assume that the salvage value for unsold inventory at the boundary is zero. We compare the performance of our heuristic against the efficient frontier of the enumerated solutions for different demand distributions. We use discretized Beta distribution to generate different kinds of demand distributions. Table 1 presents the performance evaluation results of the efficient frontier constructed by RiskTrackr_MaxEfficiency compared to the enumerated efficient frontier for various demand distributions. Here we look at the Mean Percentage Deviation of the RiskTrackr_MaxEfficiency efficient frontier from the enumerated efficient frontier. We also analyze the Hit-Rate of the RiskTrackr_MaxEfficiency efficient frontier. Hit-Rate of the RiskTrackr_MaxEfficiency heuristic is defined as the percentage of risk-reward solutions with deviations from the enumerated efficient frontier. We calculated the reward at various levels of risk along the two efficient frontiers and computed the Mean Percentage Deviation and Hit-Rate of the RiskTrackr_MaxEfficiency heuristic at <1%, <2%, <3%, <5% and <10% deviations from the enumerated efficient frontier.
We find that on an average the Mean Percentage Deviation between the heuristics and the enumerated efficient frontiers were less than 0.04% and the Hit-Rate of the heuristics efficient frontier is 100% at the 5% level. Based on the above accuracy measures we can conclude that our heuristics performs creditably in constructing efficient frontiers.

4: Conclusions and Future Research Directions

In this paper, we come up with a simple easy-to-use heuristic to construct near-optimal risk-reward curves in stochastic dynamic programming models. In many cases, managers are concerned about the risks or the variability associated with a set of policies and not just the expected reward. Instead of identifying one policy to achieve the stated objective, managers are often interested in obtaining set of policies at different levels of risk sensitivity. Using our heuristic a risk-reward curve can be easily drawn which can be used to ascertain risk exposure of a set of actions. The heuristic can be easily applied to any finite-horizon stochastic dynamic programming model. The heuristic gives a manager a useful tool with which he can figure out a set of decisions that maximizes return at a certain level of risk.

Here we have applied our heuristic to a finite-horizon stochastic dynamic programming model, namely, the inventory control model. There are other important problems that fall under the umbrella of optimal stopping rule models such as equipment replacement, options exercising and secretary problems. Coming up with efficient heuristics that can be used to construct risk-reward curves for these models could be an interesting future research endeavor. Risk-sensitive optimization in stochastic dynamic setting is an important area that has seen very limited research. In the future, we could also look at developing heuristics that provide risk-reward solutions for infinite horizon stochastic dynamic programming models.

<table>
<thead>
<tr>
<th>Beta Parameters</th>
<th>Shape of the Density Function</th>
<th>Mean % Deviation</th>
<th>Hit Rate (% occurrence of solutions with deviations from enumerated eff frontier)</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>β</td>
<td>Right Skewed</td>
<td>0.00%</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>Uniform</td>
<td>0.02%</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Bell-Shaped</td>
<td>0.02%</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>Left Skewed</td>
<td>0.04%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Beta Parameters</th>
<th>Shape of the Density Function</th>
<th>Mean % Deviation</th>
<th>Hit Rate (% occurrence of solutions with deviations from enumerated eff frontier)</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>β</td>
<td>Right Skewed</td>
<td>0.00%</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>Uniform</td>
<td>0.02%</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Bell-Shaped</td>
<td>0.02%</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>Left Skewed</td>
<td>0.04%</td>
</tr>
</tbody>
</table>
References:


