8th International IFAC Symposium on
Dynamics and Control of Process Systems
Preprints Vol.1, June 6-8, 2007, Cancún, Mexico

ACCOUNTING RISK IN MULTISTAGE STOCHASTIC PROBLEMS USING APPROXIMATE DYNAMIC PROGRAMMING

Nikolaos E. Pratikakis * **Matthew J. Realff** *
Jay H. Lee *,[1]

* *Chemical and Biomolecular Engineering, Georgia Institute of Technology,311 Ferst Drive, Atlanta GA 30332-0100, USA.*

Abstract:
This work proposes a methodology to generate risk averse policies for Markov Decision Processes(MDPs). This methodology is based on modifying the one stage reward or cost to weigh the trade-off between expected performance and downside risk represented by $(CVaR_\alpha)$. The modified stage-wise utility function is used within dynamic programming to generate a set of policies representing different levels of the trade-off. The approach is demonstrated in a shortest path optimal control problem and a project management problem modeled as constrained MDP. To address a more complex management problem, we utilize the Real Time Approximate Dynamic Programming algorithm.

Keywords: Real Time Dynamic Programming, Risk, Markov Decision Problems , Approximate Dynamic Programming.

1. INTRODUCTION

To analyze and account for risk due to uncertainty in decision-making, the adoption of a quantitative measure for risk is required. Such a measure should not lead to counter-intuitive outcomes. For example diversification should lead to risk reduction, not an increase. Artzner *et al.* (1999) defined the class of 'coherent' risk measures as those that satisfy the four axioms, which are sub-additivity, monotonicity, positive homogeneity, and translation invariance. Commonly used risk measures like standard deviation (σ) or Value at Risk (VaR_α) violate at least one of these properties and therefore can lead to counter-intuitive outcomes in certain situations. Details of why these measures violate a certain axiom can be found in Benfield (2005).

As an alternative to the popular (VaR_α), a coherent risk measure called Conditional Value at Risk $(CVaR_\alpha)$ has been proposed in the recent risk literature. $CVaR_\alpha$ is formally defined for an arbitrary loss distribution L as: $CVaR_\alpha = \mathbb{E}[L|L < VaR_\alpha]$, which represents the mean of the tail of the $(1-\alpha)\times 100$ bottom percentile of the distribution. In the above VaR_α represents the cut-off value for the corresponding percentile. The most attractive characteristics of the $CVaR_\alpha$ measure are: a) consistency with the mean-variance (Markowitz, 1952) approach in one stage problems for normal loss distributions, b) convexity leading to an attractive one stage optimization problem via LP even for non-normal dis-

[1] The authors gratefully acknowledge the financial support from the National Science Foundation (CTS#03019993)

tributions, and c) the capacity to handle fat tails (e.g., Student-T distributions). A simplified description about optimizing $CVaR_\alpha$ can be found in Borca (2004-2005). In the same article, they propose a two-stage LP formulation to minimize $CVaR_\alpha$ via external sampling.

Once such a quantitative measure for risk is adopted, one should be in position to analyze and synthesize decision policies based on it. Our interest is in generating decision policies that reflect varying degrees of risk averseness of decision-maker for general Markov Decision Processes (MDPs). For this, the construction of a utility function that weighs the trade-off between expected performance and risk is proposed, and then the usage the dynamic programming (DP) framework to generate optimal policies. An alternative approach would have been to use the mathematical programming with "chance constraints", but such a formulation presents difficulties in having to derive the multi-variable joint probability distribution of the random quantities. Hence, the approach is practical only for special cases, e.g., when the random variables are normally distributed. For an excellent introduction of such formulation, readers are referred to Henrion (2006). Compared to the mathematical programming approaches the key advantage of approximate DP is that it can be based on a procedural representation of the problem (essentially codes that perform the simulation plus a relatively small amount of book keeping overhead).

In using DP methodologies to solve multi-stage decision problems, the handling of multi-stage risk measures can be problematic, as their evaluation over multiple stages will appear in a nested manner. This means that, unlike profits or losses, risk measures are inherently non-additive stage-wise. In particular, minimizing $CVaR_\alpha$ at each time period will not amount to minimizing the $CVaR_\alpha$ of the actual multistage loss distribution itself, even though it does result in minimizing a coherent risk measure. Recent work has been done on closed form calculation of the $CVaR_\alpha$ statistic for various distributions (Andreev *et al.*, Sep. 2005).

For the well-known multistage newsvendor problem discussed in Ahmed *et al.* (2005), the authors use for the one stage cost a mean risk objective function, with $CVaR_\alpha$ as the dispersion statistic. They derive the optimality equations from scratch, and show that given this objective the structure of the optimal policy corresponds to a policy that solves the myopic problem. The derivation of the optimality equations should be performed for every "new" problem, however

To our knowledge, within the chemical engineering community, the first significant series of papers that combine DP principles and risk measures is attributed to Westerberg and coworkers (Cheng *et al.*, 2004; Cheng *et al.*, 2003; Cheng *et al.*, 2004b). Their first contribution is that they identified the need for separability and monotonicity in order for a risk measure to be able to be decomposed into stage-wise separable functions. They propose the expected downside risk as a risk measure, and augment the state definition with an additional dimension that ensures backward induction with respect to the chosen risk measure. Their second contribution is a simulation based optimization approach that approximates optimal pareto (maximize expectation - minimize risk) solutions for a multi-stage problem. They show that multi-stage stochastic mathematical programming provides higher quality pareto solutions, but their approach circumvents several numerical issues associated with mathematical programming. In their work, the efficient frontier is well represented. Their strategy is generic enough to be applied to MDP instances with large state space. However, their approach suffers from sampling issues as well as the use of global value function approximators. According to the research by Lee and Lee (2004), global approximators cannot guarantee convergence of the value function approximation and can lead to non-smooth behavior.

The proposed strategy is based on formulating a single-stage utility function that reflects a balance between expected reward and risk for the stage. From each system state s_i, given uncertainty ω_t, the expected one-stage cost (reward) for each action a_i denoted as $\mathbb{E}(f(s_i, a_i, \omega_t))$ as well as its one stage conditional value at risk are evaluated. Then one can define the one stage cost (reward) C by weighting these two statistics with a predetermined risk averse parameter λ:

$$C(s_i, a_i, \omega_t) = \lambda \mathbb{E}(f(s_i, a_i, \omega_t)) + (1-\lambda) CVaR_\alpha \quad (1)$$

The formation of a multi-stage loss function can be achieved by adding this cost function over all stages. Such formulation satisfies the requirement of separability and monotonicity, needed for application of dynamic programming.

To illustrate that the proposed modification of the single stage captures the tradeoff between expected profit and risk - and the adjustment of the linear weigh parameter yields risk averse policies of varying degrees, the following two examples are used: a) a shortest path problem with normally distributed costs, and b) a stochastic project-management problem modeled as a constrained MDP. These problems are small enough that exact solutions via dynamic programming can be achieved. Then we expand the state space of the latter problem by relaxing the resource constraint and we address it via an approximate DP method.

1.1 A Shortest Path Problem

In this section dynamic programming is applied via value iteration at a 2D shortest path setting represented at Figure 1. This problem enumerates 77 discrete states. The state space consists of all the positions in the x-y plane, while the action space includes the moves to neighboring positions (including those reachable via diagonal moves). The starting state is $(x, y) = (0, 0)$, while the end state is $(x, y) = (10, 6)$. One incurs a normally distributed cost $C_{(x,y)} \sim N(\mu_{(x,y)}, \sigma_{(x,y)})$ for each state visited. In the case, where the cost follows an unknown distribution, one should use DP with Bayesian updates over the unknown cost distribution.

Our goal is to find the path from (0,0) to (10,6), that minimizes the expected cost and simultaneously minimizes the cost associated with the (1-α) $\times$ 100% worst cases.

If one cares to solve for a risk neutral policy one would include in the one stage cost C only the statistic $\mu_{(x,y)}$. Similarly, if one wants to solve for a risk averse policy, when visiting state (x, y) the one stage cost is modified to $C = \lambda\mu + (1 - \lambda)CVaR_\alpha$ and repeat the value iteration for different values of the risk averse parameter λ. To calculate the $CVaR_\alpha$ for normal distributions, the following formula is used $CVaR_\alpha = \mu + k\sigma$, where $k = \sqrt{2}\exp(erf(2\alpha - 1)^2)^{-1}(1 - \alpha)^{-1}$.

For $\alpha = 0.95$ the resulting optimal policies are shown in Fig. 1. The cumulative numerical results are illustrated at Table 1 for $\alpha = 0.95$ for different values of λ. Theoretically, if α is increased the derived policy is inherently more risk averse for the same values of λ.

The final result for α=0.95 depicted in Fig.1 is as such: For λ=1 the shortest path minimizes the expected cost, and is represented with the solid line in Figure 1. The mean cost of that policy is 202, while its standard deviation is 32.5. The corresponding $CVaR_{0.95} = 269.9$. Parameter value λ=0 results in the most risk averse policy that can be derived with this methodology. That policy is represented with the dashed line in Fig.1. The mean cost is 206, while its standard deviation is 14.5. That results in a $CVaR_{0.95}$ of $= 235.9$. The cumulative results appear at Table 1. $\alpha = 0.99$ has been also tried. This choice of α results in the same policies with slight differences in the range of λ values corresponding to each policy. (The λ value range for the first policy is 0.0–0.3 instead of 0.0-0.1.)

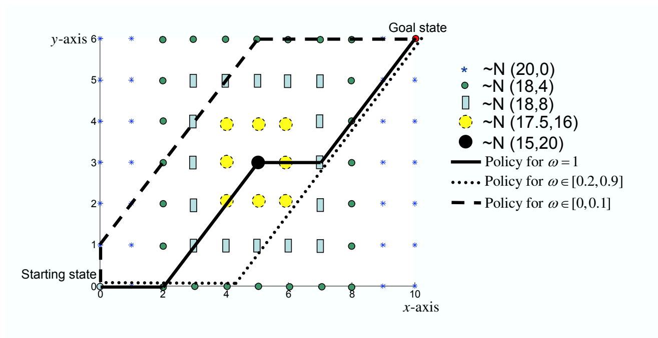

Fig. 1. Schematic illustration of a 2D shortest path problem. It demonstrates how an appropriate tuning of the λ parameter can instruct a different policy. α is set to 0.95

Cost Distribution	$\mu_{0.95}$	$\sigma_{0.95}$	$CVaR_{0.95}$
$\lambda \in$ [0,0.1]	206	14.5	235,9
$\lambda \in$ [0.2,0.9]	205	21	248.2
λ=1	202	32.5	269.9

Table 1. The mean μ and standard deviation σ of the cost distributions associated with the different policies according to λ values. (α=0.95)

1.2 Stochastic Resource Constrained Project Management Problem

Griffin (1997) research indicates that almost half of the resources that U.S. industry devotes to New Product Development (NPD) are spent on products that fail to be launched. Such uncertainty motivates the use of optimal stochastic control to optimize the usage of resources to projects that satisfy the decision makers risk averse criteria. Optimal stochastic control can be applied only via DP or multistage stochastic programming.

1.2.1. Problem Description In the proposed project management problem, the process of new project arrival D is dictated by a first order Markov Chain. In our case only three types of projects can arrive. Each project can be processed via a 5 stage specialized pipeline.

For each project, one can choose between two types of pipe lines, one with high-risk / high-return and the other with lower-risk / lower-return. Hence there are six possible pairings of project type (not up to the choice of the decision-maker) and pipe-line type (up to the choice of the decision-maker). Each pairing is characterized in terms of: **a)** profit distribution (i.e., the profit is realized, if the project is successfully launched), **b)** required resources at each time period while it is in the pipeline, and **c)** probability of project failure at each time period. A further twist is added to this problem, by modeling the progress of each project with a two state Markov Chain. Due to this, at each time period the project will either progress to the next stage or stay static at the same stage according to a given probability.

Also, in order to capture and hedge the risk in this application, the probability of failure is set to be commensurate with the expected profit of each project. Available resources are constrained and projects can be canceled at any time (without any profit). All these features make it a complex decision problem. In order to solve for the optimal policy for this constrained MDP, rules have to be defined to ensure that each action will result in transitions from the original state s_i the feasible states.

1.2.2. State-Action Space The state space $\mathbb{S}$ has 16 variables. $\mathbb{S}$=$\{(D, n_{i,j}), i = 1,2,3 \wedge j = 1,2,3,4,5 | \sum_{i,j} n_{i,j} K_{i,j} \leq \Theta\}$. D is the Markov state that corresponds to the type of the possible arriving product at the next time period and $n_{i,j}$ denotes the number of projects of type i in the j^{th} stage of the i^{th} pipeline. $K_{i,j}$ corresponds to the number of resource units needed to continue a project of type i in the j^{th} stage. Hence, the total number of resource units available is Θ. The action space $\mathbb{A}$ contains a) actions that accept or reject incoming project arrivals, and b) actions that direct a project either to the low risk or the high risk pipeline.

1.2.3. Transition Function The transition function from state s_i to s_j's will undergo three steps.

Step 1 Apply desired control on s_i.

Step 2 Observe the derived state and calculate the automatic transitions. All the possible s_j's correspond to the uncertain Markov process of the progress of the on-going projects given initial state s_i. Each state s_j is associated with a transition probability $P_{s_i \rightarrow s_j}$. Assuming N possible transitions then: $\sum_{j=1}^{N} P_{s_i \rightarrow s_j} = 1$.

Step 3 For each state s_j the next possible transitions correspond to the uncertain process of survival or termination of the individual projects. If one denotes those states by s_{jj}, and with M the possible transitions, for each s_j: then $\sum_{jj=1}^{M} P_{s_j \rightarrow s_{jj}} = P_j$.

1.2.4. Reward Function Using Bayes rule one can calculate the expected profit for each project. The recursive formula for the expected profit for a project at stage i is:

$$\mathbb{E}(\text{i}) = \frac{Pr(\text{i} \rightarrow \text{i}+1)P(s)}{(1 - Pr(\text{i} \rightarrow \text{i}))P(s)} \mathbb{E}(\text{i+1}) \qquad (2)$$

where, $\mathbb{E}$(i) is the expected value of a project at stage i of the pipeline, $P(s)$ is the probability of the project still remaining in the pipeline, and $P(i \rightarrow i+1)$ is the probability of the project progressing to a next stage. If the last stage is successfully reached, a profit value is realized from the given profit distribution.

1.2.5. Comments on Full DP Results The simulation baseline parameters are displayed in Figure 2.

$$P_{PG-HR-1} = \begin{bmatrix} 0.3 & 0.7 & 0 \\ 0 & 0.40 & 0.60 \\ 0 & 0 & 1 \end{bmatrix}, \quad P_{PR-LR-1} = \begin{bmatrix} 0.20 & 0.80 & 0 \\ 0 & 0.2 & 0.80 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P_{PG-HR-2} = \begin{bmatrix} 0.3 & 0.7 & 0 \\ 0 & 0.35 & 0.65 \\ 0 & 0 & 1 \end{bmatrix}, \quad P_{PR-LR-2} = \begin{bmatrix} 0.15 & 0.85 & 0 \\ 0 & 0.15 & 0.85 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P_{PG-HR-3} = \begin{bmatrix} 0.45 & 0.55 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}, \quad P_{PR-LR-2} = \begin{bmatrix} 0.35 & 0.65 & 0 \\ 0 & 0.35 & 0.65 \\ 0 & 0 & 1 \end{bmatrix}$$

$\text{Profit}_{HR-1} \sim N(3 \cdot 10^3, 6 \cdot 10^2), \text{Profit}_{LR-1} \sim N(1.8 \cdot 10^3, 1)$

$\text{Profit}_{HR-2} \sim N(2 \cdot 10^3, 3 \cdot 10^2), \text{Profit}_{LR-1} \sim N(1.2 \cdot 10^3, 1)$

$\text{Profit}_{HR-3} \sim N(10^5, 4 \cdot 10^4), \text{Profit}_{LR-3} \sim N(6 \cdot 10^4, 1)$

$P_{HR-1}(s) = 0.8, P_{HR-2}(s) = 0.85, P_{HR-3}(s) = 0.85,$

$P_{LR-1}(s) = 0.9, P_{LR-2}(s) = 0.95, P_{LR-3}(s) = 0.95,$

$K_1 = 60, K_2 = 45, K_3 = 110,$

$P_{D_1 \rightarrow D_1} = 0.7, P_{D_2 \rightarrow D_2} = 0.8$

$\Theta = 200$

Fig. 2. Simulation Baseline Parameters.

P_{PG-x-i} is the Markov Chain that describes the progress of a type i project via the pipeline x (the type of pipeline can be of low risk (LR) or high risk (HR)). Profit_{x-i} corresponds to the normally distributed uncertain reward of a launched project i via the pipeline x . P_{x-y} is the probability that the type y project does not fail at the current time period and continues to occupy K_y resource units. The probabilities $P_{D_1 \rightarrow D_1}$ and $P_{D_2 \rightarrow D_2}$ define the Markov Chain of the future arriving states. If the arrival state D_1 is experienced there is a 50% chance for a project type 1 to appear in order to be processed and a 5% chance for a project type 3 to appear. Likewise for arrival state D_2.

The feasible state space (due to the resource constraint) for this instance is 892 states out of the possible 26,871. The curse of dimensionality (COD) is mainly attributed to the difficulty in calculating the one step expected reward. Note that, the computational requirement for that operation increases exponentially with the number of projects handled by the pipelines. From the results displayed in Table 3, one can see that the choice of parameter value $\lambda = 0$ gave the most risk averse policy possibly obtained with our methodology. Also, one could not create a more risk averse multistage policy by increasing α for this particular problem. To test the derived strategies we sampled 250 scenarios. As expected the risk-averse strategy sacrifices the expected profit, but increases the $CVaR_\alpha$ and vice versa.

Profit Distr. (*10^4)	$\mu_{0.99}$	$CVaR_\alpha$ α=0.99	$\mu_{0.95}$	$CVaR_\alpha$ α=0.95
$\lambda = 0$	4.2	2.7	4.2	2.7
$\lambda = 1$	6.1	2.4	6.1	2.4

Table 2. Statistics of the profit distributions obtained with the policies for different values of α and ω for the project management problem.

2. SOLVING MORE COMPLEX PROBLEMS

In larger applications exact DP is computationally intractable. Therefore we have to resolve to ADP algorithms. The trick is that ADP algorithms are usually specialized to a particular application and they almost never claim generality. To our knowledge and experience, state of the art ADP algorithms that may produce satisfactory results to a general problem are (Powell, 2006) (Farias and Van Roy, 2003) (Lee and Lee, 2004).

Our proposed ADP strategy is based on Real Time Dynamic Programming (RTDP) (Barto *et al.*, 1995) and is named Real Time Approximate Dynamic Programming (RTADP). In the proposed RTADP strategy, one approximates the value functions in the regions of the state space, visited by effective "risk-sensitive" policies. It adaptively samples portions of the state space as it simulates the system behavior under a greedy or a ϵ-greedy exploration strategy.

In Section 2.1, the RTADP algorithm is analyzed and. Then in section 2.2, RTADP is applied to the same project management problem with an enlarged state space. This is achieved by relaxing the resource constraint. The quality of the derived policy is then evaluated for $\lambda = 0$ and $\lambda = 1$ via Monte Carlo simulations.

2.1 The RTADP Algorithm

The procedure below samples the state space using a greedy policy and constructs a value table denoted as $\mathbb{S}_{sim}$ starting from an empty one by gradually adding entries, as states are encountered in the simulation. The following steps are involved in each iteration of the algorithm.

For iterations $i = 1, 2, ...M$, where M is a sufficiently large integer

Step 1 Start from a *random* state $s_i \in \mathbb{S}_{sim}$.

Step 2 Construct set of actions (denoted by A_{sub}) for s_i. $A_{sub} \subset A$, where A is the set of all possible controls that the decision maker can exercise at any time instance for a given state. Details about the notion of A_{sub} (called 'Adaptive Action Set') and how it is numerically constructed are given in (Pratikakis *et al.*, 2006).

$$J^{\pi}_{i+1}(s_i) = \max_{\alpha \in \mathbb{A}_{sub}} \{\phi(s_i, \alpha, \omega_t) + \gamma \sum_{j=1}^{N} P_{s_j|s_i,\alpha} J^{\pi}_i(s_j)\} \quad (3)$$

, where γ is a discount factor $\gamma \in [0, 1)$.

Step 3 Update the value functions for $J^{\pi}_i(s_i)$ according to Eq.(4).

Every control in A_{sub} is evaluated with respect to the Bellman Equation (Eq.(3) and the decision -maker follows a policy that is greedy with respect to the most recent estimate of the value table (Eq.(4)).

$$\alpha^*(s_i) = \textbf{arg} \max_{\alpha \in A_{sub}} \{\phi(s_i, \alpha, \omega_t) + \gamma \sum_{j=1}^{N} P_{s_j|s_i,\alpha} J^{\pi}_i(s_j)\} \quad (4)$$

This evaluation (Eq.(3) requires knowledge of, if not an estimation of $J^{\pi}_i(s_j)$ $\forall j$. j denotes an index running from $1, ..., N$, where N is the number of possible transitions from the starting state s_i to the successor states s_j. N may vary with the iteration number (e.g., if s_i does not fully communicate with some s_j's). The interested reader can find more insight about the how to circumvent the computational obstacles associated with the optimality or Bellman equation in (Powell, 2006).

Step 4 A state s_j is sampled according to the probability distribution $p(s_j|s_i, \alpha^*)$ as defined from the Markov model of the random variables. If s_j does not belong in $\mathbb{S}_{sim}$, then it is added to the table as a new entry. The sampled s_j is set as s_{i+1}. Set $i = i + 1$ and go back to Step 1.

End

Note that, if the algorithm happens to circulates over a small cyclic graph of states, the algorithm is restarted from a *random* state $s_i \in \mathbb{S}_{sim}$. Empirically, one terminates with : $(\|J^{\pi}_{i+1}(s_i) - J^{\pi}_i(s_i) < \varepsilon\|_\infty \quad \forall s_i \in \mathbb{S}_{sim} \subseteq \mathbb{S}$) like in VI, where ε is a tolerance parameter. The user can apply this termination criterion, only if the state space is saturated and the number of entries does not grow. In Pratikakis *et al.* (2006) it is analyzed how this ADP approach differs from the classical RTDP. Moreover it is emphasized and discussed the importance of initialization of the value function for unseen successive states in balancing between the exploitation and exploration.

2.2 *Resource Constrained Project Scheduling Problem with Enlarged State Space*

By relaxing the resource constraint ($\Theta = 400$) the feasible state space $\mathbb{S}$ compared to the previous example is enlarged by one order of magnitude. Specifically, the feasible state space consists out of 24,871 states. If the above hard constraint was not imposed then the state space would consist out of 2,672,646 states.

The results obtained by applying RTADP are summarized in Table 3. Note that, the tried realizations are only 100.

Profit Distr. (*10^3)	$\mu_{0.95}$	$CVaR_\alpha$ α=0.95	$\|\mathbb{S}\|$
$\lambda = 0$	7.2	-6.2	$\simeq 6100$
$\lambda = 1$	9.1	-5.9	$\simeq 3400$

Table 3. Statistics of the profit distribution, after applying the policy learned by the RTADP from a given initial state for $\lambda = 0$ and $\lambda = 1$ for 30 time periods.

The policy derived using the objective function from Eq.1 with $\lambda = 1$ is superior than the policy with $\lambda = 0$. This result indicates that the proposed methodology is quite sensitive to the choice of the baseline parameters of a given problem. Nonetheless, the RTADP procedure (with $\lambda = 1$) instructs a high quality policy for this complex project management problem.

3. CONCLUSIONS

Our proposal to solve for a multi-stage risk averse policy lies in modifying the expected reward into a pseudo-utility function. The statistics that are weighed, by parameter λ, in this function are the expected profit and the downside risk measured by $CVaR_\alpha$. Our contributions are: **a)** providing a mechanism to solve for policies of varying degrees of risk-averseness for general MDPs; **b)** providing empirical results that demonstrate, that the separable pseudo-utility functions, give converged value functions that lead to risk averse multistage policies directly controlled by the tuning parameter λ; **c)** demonstrating the fact that RTADP can be used to solve combinatorial problems with large state space.

The on-going research involves to couple this RTADP approach with local regression methods to minimize the COD concerning the k-nearest neighbor local value function approximation scheme. Solving problems involving multiple agents is another topic of interest.

REFERENCES

Ahmed, S., U. Cakmak and A. Shapiro (2005). Coherent risk measures in inventory problems. *http://www.optimization-online.org.*

Andreev, A., A. Kanto and P. Malo (Sep. 2005). On closed form calculation of cvar. *Helsinki School of Economics, W-389.*

Artzner, P., F. Delbaen, J.-M. Eber and D. Heath (1999). Coherent measures of risk. *Math. Fin.* **9**, 203–228.

Barto, A., S. Bradtke and S. Singh (1995). Learning to act using real-time dynamic programming. *Art. Int.* **72**, 81–138.

Benfield (2005). *Risk Measurement in Insurance.* http://www.benfieldgroup.com/media+centre/research+and+publications/risk+measurement.pdf. London.

Borca, B. (2004-2005). *Multi-period Constrained Portfolio Optimization Using Conditional Value at Risk.* McS in Banking and Finance. University of Lausanne.

Cheng, L., E. SUbrahmanian and A.W. Westerberg (2003). Design and planning under uncertainty: Issues on problem formulation and solution. *Comp. Chem. Eng.* **27**, 781–801.

Cheng, L., E. SUbrahmanian and A.W. Westerberg (2004). A comparison of optimal control and stochastic programming from a formulationa and computation perspective. *Comp. Chem. Eng.*

Cheng, L., E. SUbrahmanian and A.W. Westerberg (2004b). Multi-objective decisions on capacity planning and inventory control. *Ind. Eng. Chem. Res.* **43**, 2192–2208.

Farias, D. De and B. Van Roy (2003). The programming approach to approximate dynamic programming. *Op. Res.* **51**(6), 850–865.

Griffin, A. (1997). Pdma research on new product development practises: Updating trends and benchmarking best practices. *J. Prod. Innov. Man.* pp. 429–458.

Henrion, Rene (2006). Introduction to chance-constrained programming. *http://stoprog.org/index.html/SPIntro/intro2ccp.html.*

Lee, J.M. and J.H. Lee (2004). Approximate dynamic programming strategies and their applicability for process control: A review and future directions. *Int. J. Cont. Aut. Sys.* **2**(3), 263–278.

Markowitz, H. (1952). Portfolio selection. *J. Fin.* pp. 77–91.

Powell, W.B. (2006). *Approximate Dynamic Programming for Operations Research.* http://www.castlelab.princeton.edu.

Pratikakis, N.E., J.H. Lee and M.J. Realff (2006). A real time adapive dynamic programming approach for planning and scheduling. 16^{th} *ESCAPE and* 9^{th} *PSE.*