

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Risk-Constrained Forward Trading Optimization by Stochastic Approximate Dynamic Programming

Miguel Gil-Pugliese and Fernando Olsina

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/57466>

1. Introduction

Since the mid-twentieth century, Dynamic Programming (DP) has proved to be a flexible and powerful approach to address optimal decisions problems. Nevertheless, a decisive drawback of the conventional DP is the need for exploring the whole state space in order to find the optimal solution. The immense amount of mathematical operations involved to solve real-scale problems, constrained the application of DP to small or highly simplified cases. Indeed, state space grows exponentially with the number of variables when considering multivariate optimization. The curse of dimensionality is a well-known limitation of conventional DP algorithms for tackling large-scale problems ubiquitous in real science and engineering applications.

In the last decades, many new algorithms emerged in different branches of science to overcome the inherent limitations of conventional DP. Unlike conventional DP, these algorithms avoid enumerating and calculating every possible state of a system during the optimization process. Instead, they estimate relevant features of the state space. This approach circumvents the dimensionality limitations of the conventional DP while retaining many of its advantages.

In this chapter, the application of advanced stochastic dynamic programming techniques to the optimization of the forward sell strategy of a power generator subjected to delivery risk is considered. The proposed approach allows rebalancing the portfolio during the period of analysis. In electricity markets, a power generator can sell in advance part or all its future energy production at a fixed price, hedging against the high price volatility of the spot market. The strategy of eliminating the price risk by selling in advance the entire production in the forward market to a fixed price is often thought as the minimum-risk trading policy. Nonetheless, it can be proven that this is not the case for most generators. The outages of the generation units and transmission lines, as well as unforeseen limitations in the primary energy

supply expose generators to delivery risk [1]. Delivery risk considerably modifies the probability distribution of profits, shifting the optimal trading strategy toward a portfolio mixing forward contracts and power sold in the spot market. Because of the size of the probability state space and the limited computing capabilities, the problem of the optimal trading strategy has not a closed form solution and thus, its determination is matter of current study. The increase in computing power and recent developments in Operational Research has brought new insights into the solution of such problems.

In the past decade and by virtue of the ever increasing computational power, many methods emerged in different scientific fields with several different names: Reinforced Learning, Q-Learning, Neuro-Dynamic Programming, etc. All these methods were later brought together in what is currently known as Approximated Dynamic Programming (ADP) [2],[3]. These algorithms resign the exhaustive enumeration and calculation of the space-state typically performed by conventional DP. Instead, they iteratively approximate a function of the space state through stochastic simulation and statistical regression techniques, circumventing the dimensionality problem of DP.

Although ADP algorithms are being used in several other fields of science, the application to design optimal trading strategies in power markets has not been proposed so far. In this chapter, ADP techniques are exploited to optimize the selling strategy of a power generator trading in a frictional market with transaction costs. Three available products are considered: selling in the spot market, and/or get involved in quarterly and one-year forward contracts. The objective of the generator is to maximize the expected profit while limiting financial risk. Decisions can be made only at the beginning of each month. At each decision stage, the current trading position can be changed at a cost in order to rebalance the portfolio.

2. Approximate dynamic programming

In the field of decision making, it is often useful to assess what could be expected from each possible decision given the information available. After evaluating the outcome of each alternative decision, a simple comparison is enough to take the optimal course of action. This approach is straightforward but also naive. Real problems often present simply too many possible options to evaluate. Moreover, if the problem involves sequential decision stages, the number of possible solution paths scales up exponentially. Finally, outcomes are frequently subjected to uncertainty. So, several outcomes could present themselves for each decision, augmenting further the size of the problem.

Therefore, in order to keep the size of the problem within reach, shortcuts and simplifications are often necessary. Dynamic Programming (DP) is a clever way to reduce the number of options based on Bellman's Principle of Optimality: "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision" [4]. This leads to the fact that from any given state there is an optimal decision trajectory that once solved can be used for every other path containing that state. Therefore for each state is only necessary to

hold the information of the best solution until the end, ruling out suboptimal options. This rule prevents the exponential growth in the decision sequence path, scaling the problem only linearly.

Yet, other reasons of explosive dimensionality growth remain, namely the number of states, as well as decision and outcome spaces. For small financial decision problems, conventional DP algorithms are able to find the optimal policy. For real-scale problems, gross simplifications are often necessary to keep tractability. Sometimes, these simplifications render the model unrealistic turning results meaningless.

Finding an appropriate combination of financial instruments in a portfolio can be portrayed as a Markov Decision Problem (MDP). A MDP is defined as a succession of states that are reached through decisions (actions) followed by an outcome (reaction) of the system. After each decision, the system evolves into another state, according to probabilities defined by the previous state and the decision taken. Each transition between states has a cost, usually dependent on the action taken, and a reward produced by the reaction of the system, as depicted in Figure 1. In these processes, a decision maker should choose a sequence of actions so that the expected sum of rewards minus the sum of incurred costs is maximized over a certain period of time.

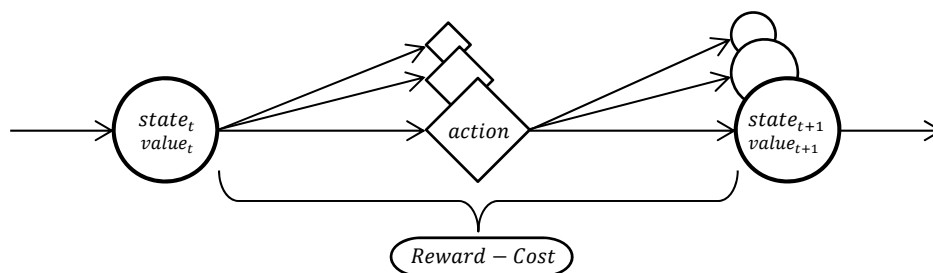


Figure 1. Markov Decision Process depiction

According to the Bellman's Principle of Optimality, for every MDP can be determined a series of Value Functions, which represents the continuation value of each state. The continuation value associated to a given state is the expected sum of rewards that the optimal strategy would yield from that state until the end of the process (or the expected average reward if the MDP is infinite).

It is easy to see how the value functions of a MDP can be found using a classic backwards DP algorithm. Starting from the final states, a DP algorithm exhaustively calculates the continuation value for a discrete number of states. All these continuation values, collected in a lookup table, constitute later the Value Functions which are accurate but not very compact or easy to calculate. After acquiring the Value Functions, it is simple to find an optimal decision for each state as the one that maximizes the sum of the expected reward and the expected continuation value of the next state or states. However, the problems that a DP algorithm can address by this procedure are restricted by the size of their state spaces.

Other forms to represent and/or approximate the Value Functions can then be proposed. These approaches should not require exhaustive calculation of every state. The Value Functions can be interpolated between computed states. Approximate Dynamic Programming algorithms are built on this cornerstone: approximation of the Value Functions in the state space domain. The estimation methods can be linear regressions, artificial neural networks, etc. Several authors make detailed analysis of MDPs and the use of ADP and DP algorithms to solve them [2],[3].

For approximating and updating the Value Functions, the proposed algorithm uses linear regression on Gaussian radial basis functions jointly with Monte Carlo simulations to consider randomness. An interior-point optimization algorithm is implemented to make decisions.

The ADP algorithm starts with a series of approximations of the value functions, usually constant. Then taking a Monte Carlo sample, a simulation of the system is conducted. At each decision stage, the algorithm makes a decision that is optimal regarding to the current state and the available approximations. Finally, after each Monte Carlo simulation, decisions and outcomes are used to refine the estimation of the Value Functions and complementary Risk Functions, denoted by V_t and R_t respectively. The process continues iteratively, until a certain termination criterion is fulfilled. A simple diagram of this approach is illustrated in Figure 2.

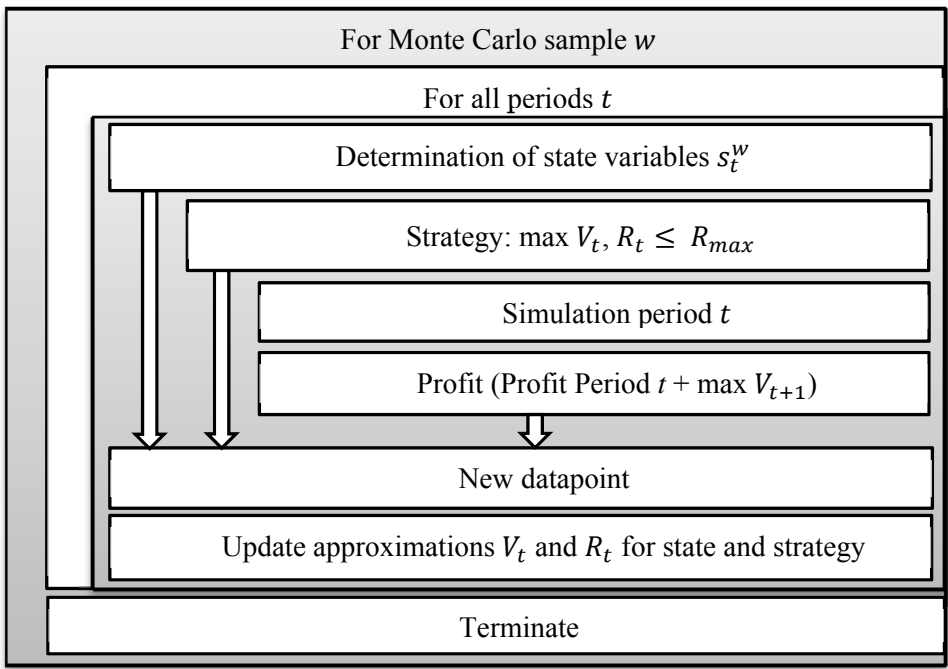


Figure 2. Simulation and estimation update as performed by the proposed ADP algorithm

2.1. Regression technique

An approximation of a system involves finding a function that can predict the output values for a given set of inputs. In this case, the inputs are the state variables and the decision and the outputs are the continuation value and risk.

To approximate the value and risk functions one could use several methods such as linear regressions, artificial neural networks, splines, etc. In the context of electricity trading, it will be discussed how to use linear regression with radial basis functions. The same principles and techniques apply to any regression based on linear parameters.

The main feature of the algorithm is that the approximation is used to make decisions while collecting new data to further improve the currently available approximation. Thus, it is necessary that the regressed function fitting the data can be readily updated as new data is simulated, making the improved approximations immediately available for the next simulation. To fulfill this requirement, a recursive regression technique can be used.

Let be a set of inputs X and a set of collected data from outputs Y

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,g} \\ x_{2,1} & x_{2,2} & \dots & x_{2,g} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \dots & x_{n,g} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}, Y = \begin{bmatrix} y_{1,1} & y_{1,2} & \dots & y_{1,h} \\ y_{2,1} & y_{2,2} & \dots & y_{2,h} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n,1} & y_{n,2} & \dots & y_{n,h} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_n \end{bmatrix} \quad (1)$$

The matrix of inputs X has n data points and g dimensions and the matrix of outputs Y has n data points and h dimensions. Then, the approximation function is:

$$\hat{Y} = \sum_{j=1}^k \varphi_j(X) \cdot \theta^j = V(X) = \begin{bmatrix} \hat{y}_{1,1} & \hat{y}_{1,2} & \dots & \hat{y}_{1,h} \\ \hat{y}_{2,1} & \hat{y}_{2,2} & \dots & \hat{y}_{2,h} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{n,1} & \hat{y}_{n,2} & \dots & \hat{y}_{n,h} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \vdots \\ \hat{\mathbf{y}}_n \end{bmatrix} = \varphi(X) \cdot \theta \quad (2)$$

$$\varphi(X) = \begin{bmatrix} \varphi_1(\mathbf{x}_1) & \varphi_2(\mathbf{x}_1) & \dots & \varphi_k(\mathbf{x}_1) \\ \varphi_1(\mathbf{x}_2) & \varphi_2(\mathbf{x}_2) & \dots & \varphi_k(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{x}_n) & \varphi_2(\mathbf{x}_n) & \dots & \varphi_k(\mathbf{x}_n) \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_n \end{bmatrix} = \mathbf{Z}, \theta = \begin{bmatrix} \theta^1 \\ \theta^2 \\ \vdots \\ \theta^k \end{bmatrix} \quad (3)$$

Where the kernel functions $\varphi(X)$ transform the input variables X into a k -dimensional space and are in general non-linear. Here, it is important to notice that despite the fact the φ functions are nonlinear, the approximation is still linear with respect the parameters θ . Therefore, the regression parameters can easily be found by solving a linear system of equations.

The parameter vector θ that minimizes the mean quadratic error MQE of the estimated outputs is:

$$\theta = [\mathbf{Z}^T \mathbf{Z}]^{-1} \mathbf{Z}^T \mathbf{Y} = \mathbf{A} \mathbf{Z}^T \mathbf{Y} \quad (4)$$

$$\mathbf{A} = [\mathbf{Z}^T \mathbf{Z}]^{-1} \quad (5)$$

The dimension of matrices \mathbf{X} , \mathbf{Y} and \mathbf{Z} increase as new points are simulated, and thus the estimated parameters change as new data is collected. On the other hand, the algorithm needs the approximation to get new data. One approach could be calculating the new parameters after a number of simulations with equation (4) but it soon reveals impractical as an increasing amount of data need to be stored. Instead, by storing only the matrix \mathbf{A} and the parameter vector θ , the matrix itself can recursively be updated jointly with the parameters, following the process described in [2]:

In the i -th iteration, a new data point \mathbf{x}_i , \mathbf{y}_i is simulated and the matrices \mathbf{X} and \mathbf{Y} become:

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{X}_{i-1} \\ \mathbf{x}_i \end{bmatrix}, \mathbf{Y}_i = \begin{bmatrix} \mathbf{Y}_{i-1} \\ \mathbf{y}_i \end{bmatrix} \quad (6)$$

Also the matrix \mathbf{Z} becomes:

$$\mathbf{Z}_i = \begin{bmatrix} \phi(\mathbf{X}_{i-1}) \\ \phi(\mathbf{x}_i) \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{i-1} \\ \mathbf{z}_i \end{bmatrix} \quad (7)$$

Replacing these values in (4):

$$\theta_i = [\mathbf{Z}_i^T \mathbf{Z}_i]^{-1} \mathbf{Z}_i^T \mathbf{Y}_i = \mathbf{A}_i \mathbf{Z}_i^T \mathbf{Y}_i \quad (8)$$

$$\theta_i = \left[\begin{bmatrix} \mathbf{Z}_{i-1}^T & \mathbf{z}_i^T \end{bmatrix} \begin{bmatrix} \mathbf{Z}_{i-1} \\ \mathbf{z}_i \end{bmatrix} \right]^{-1} \begin{bmatrix} \mathbf{Z}_{i-1}^T & \mathbf{z}_i^T \end{bmatrix} \begin{bmatrix} \mathbf{Y}_{i-1} \\ \mathbf{y}_i \end{bmatrix} \quad (9)$$

$$\theta_i = [\mathbf{Z}_{i-1}^T \mathbf{Z}_{i-1} + \mathbf{z}_i^T \mathbf{z}_i]^{-1} [\mathbf{Z}_{i-1}^T \mathbf{Y}_{i-1} + \mathbf{z}_i^T \mathbf{y}_i] \quad (10)$$

Using the Sherman-Morrison formula:

$$[\mathbf{L} + \mathbf{u}^T \mathbf{u}]^{-1} = \mathbf{L}^{-1} - \frac{\mathbf{L}^{-1} \mathbf{u}^T \mathbf{u} \mathbf{L}^{-1}}{1 + \mathbf{u} \mathbf{L}^{-1} \mathbf{u}^T} \quad (11)$$

Where \mathbf{L} is a $k \times k$ invertible matrix and \mathbf{u} is a k -dimensional row vector:

$$\theta_i = \left(\left[\mathbf{Z}_{i-1}^T \mathbf{Z}_{i-1} \right]^{-1} - \frac{\left[\mathbf{Z}_{i-1}^T \mathbf{Z}_{i-1} \right]^{-1} \mathbf{z}_i^T \mathbf{z}_i \left[\mathbf{Z}_{i-1}^T \mathbf{Z}_{i-1} \right]^{-1}}{1 + \mathbf{z}_i \left[\mathbf{Z}_{i-1}^T \mathbf{Z}_{i-1} \right]^{-1} \mathbf{z}_i^T} \right) (\mathbf{Z}_{i-1}^T \mathbf{Y}_{i-1} + \mathbf{z}_i^T \mathbf{y}_i) \quad (12)$$

Replacing $[\mathbf{Z}_{i-1}^T \mathbf{Z}_{i-1}]^{-1} = \mathbf{A}_{i-1}$

$$\theta_i = \left(\mathbf{A}_{i-1} - \frac{\mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{z}_i \mathbf{A}_{i-1}}{1 + \mathbf{z}_i \mathbf{A}_{i-1} \mathbf{z}_i^T} \right) (\mathbf{Z}_{i-1}^T \mathbf{Y}_{i-1} + \mathbf{z}_i^T \mathbf{y}_i) \quad (13)$$

Naming $\rho_i = \frac{1}{1 + \mathbf{z}_i^T \mathbf{A}_{i-1} \mathbf{z}_i}$ and distributing:

$$\theta_i = \mathbf{A}_{i-1} \mathbf{Z}_{i-1}^T \mathbf{Y}_{i-1} + \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{y}_i - \rho_i \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{z}_i \mathbf{A}_{i-1} \mathbf{Z}_{i-1}^T \mathbf{Y}_{i-1} - \rho_i \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{z}_i \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{y}_i \quad (14)$$

Now, simplifying $\mathbf{A}_{i-1} \mathbf{Z}_{i-1}^T \mathbf{Y}_{i-1} = \theta_{i-1}$

$$\theta_i = \theta_{i-1} - \rho_i \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{z}_i \theta_{i-1} - \rho_i \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{z}_i \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{y}_i + \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{y}_i \quad (15)$$

Taking common factor $-\rho_i \mathbf{A}_{i-1} \mathbf{z}_i^T$ of the last 3 terms:

$$\theta_i = \theta_{i-1} - \rho_i \mathbf{A}_{i-1} \mathbf{z}_i^T \left[\mathbf{z}_i \theta_{i-1} + \mathbf{z}_i \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{y}_i - \frac{1}{\rho_i} \mathbf{y}_i \right] \quad (16)$$

Replacing $\frac{1}{\rho_i} = 1 + \mathbf{z}_i^T \mathbf{A}_{i-1} \mathbf{z}_i$:

$$\theta_i = \theta_{i-1} - \rho_i \mathbf{A}_{i-1} \mathbf{z}_i^T \left[\mathbf{z}_i \theta_{i-1} + \mathbf{z}_i \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{y}_i - (1 + \mathbf{z}_i^T \mathbf{A}_{i-1} \mathbf{z}_i) \mathbf{y}_i \right] \quad (17)$$

$$\theta_i = \theta_{i-1} - \rho_i \mathbf{A}_{i-1} \mathbf{z}_i^T \left[\mathbf{z}_i \theta_{i-1} + \mathbf{z}_i \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{y}_i - \mathbf{y}_i - \mathbf{z}_i \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{y}_i \right] \quad (18)$$

Finally, canceling out we get a formula to update the parameter vector:

$$\theta_i = \theta_{i-1} - \rho_i \mathbf{A}_{i-1} \mathbf{z}_i^T [\mathbf{z}_i \theta_{i-1} - \mathbf{y}_i] \quad (19)$$

Note that in order to update the approximations, it is needed to store and update only the values of the matrix A and the parameter vector θ . The complete set of update formulas is then:

$$\rho_i = \frac{1}{1 + \mathbf{z}_i^T \mathbf{A}_{i-1} \mathbf{z}_i} \quad (20)$$

$$\theta_i = \theta_{i-1} - \rho_i \mathbf{A}_{i-1} \mathbf{z}_i^T [\mathbf{z}_i \theta_{i-1} - y_i] \quad (21)$$

$$\mathbf{A}_i = \mathbf{A}_{i-1} - \rho_i \mathbf{A}_{i-1} \mathbf{z}_i^T \mathbf{z}_i \mathbf{A}_{i-1} \quad (22)$$

By using the equations (20), (21) and (22), it is possible to update a regression while using it to take decisions within the ADP algorithm without explicitly storing the entire dataset and performing the matrix inversion after each Monte Carlo simulation. However, an important issue arises at the start of the recursive process, namely the starting values for both the parameter vector θ and the matrix A . One possibility is to carry first a number of simulations with random decisions, and then use the data collected to calculate the starting values for θ and A . Another option is to use a diagonal matrix:

$$\mathbf{A}_i = \begin{bmatrix} a_{1,1} & 0 & \dots & 0 \\ 0 & a_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{k,k} \end{bmatrix} \quad (23)$$

This option is usually simpler despite the fact that setting the absolute values of the elements requires care and it is not always clear in the literature. To set suitable values, a closer look at the regression formulas for matrix A is needed:

$$\mathbf{A}_i = [\mathbf{Z}_i^T \mathbf{Z}_i]^{-1} = [\mathbf{Z}_{i-1}^T \mathbf{Z}_{i-1} + \mathbf{z}_i^T \mathbf{z}_i]^{-1} \quad (24)$$

$$\mathbf{A}_i = [\mathbf{Z}_i^T \mathbf{Z}_i]^{-1} = [\mathbf{Z}_{i-2}^T \mathbf{Z}_{i-2} + \mathbf{z}_{i-1}^T \mathbf{z}_{i-1} + \mathbf{z}_i^T \mathbf{z}_i]^{-1} \quad (25)$$

$$\mathbf{A}_i = [\mathbf{Z}_i^T \mathbf{Z}_i]^{-1} = \left[\sum_{p=1}^i \mathbf{z}_p^T \mathbf{z}_p \right]^{-1} \quad (26)$$

$$\mathbf{A}_i = \left[\sum_{p=1}^i \begin{bmatrix} (z_{p,1})^2 & z_{p,1}z_{p,2} & \cdots & z_{p,1}z_{p,k} \\ z_{p,1}z_{p,2} & (z_{p,2})^2 & \cdots & z_{p,2}z_{p,k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{p,1}z_{p,k} & z_{p,2}z_{p,k} & \cdots & (z_{p,k})^2 \end{bmatrix} \right]^{-1} \quad (27)$$

The matrix \mathbf{A}_i is the inverse of a matrix sum whose diagonal elements are positive and increasing with i . That means that the absolute value of the diagonal elements of \mathbf{A} would in general decrease as i increases. The relation factor between the starting value of all the elements $a_{j,j}$ and the corresponding expected value of $(1/z_{p,j})^2$ controls how new values modify the parameter vector θ .

$$a_{j,j} = \delta \cdot \mathbf{E} \left[1 / (z_{p,j})^2 \right] \quad (28)$$

Setting starting diagonal elements with a small δ value implies a large number of iterations i . This would mean that the starting parameter vector represents already a large amount of “fictional” data. In that case, any new real data sample will have little impact on the parameter vector, and the convergence rate of the approximation might be significantly slowed. On the other hand, setting initial diagonal values calculated with large δ enables the algorithm to large modifications on the parameter vector as new data is collected, which could cause instability of the parameter vector and in the decision computed, again reducing the overall convergence rate. In practice, the factor δ depends on the problem and it must be assessed accordingly. From extensive experimentation, values of δ between 1 and 1000 tend to work most of the time.

For initializing the parameter vector θ_0 , it is usual to set a constant vector. Values obtained from the formula $\hat{\mathbf{Y}}_s = \varphi(\mathbf{E}(\mathbf{X})) \cdot \theta_0$ produce coherent results. It is important to notice that the starting values of the parameter vector are important in cases where δ is small, while for a large δ the starting vector value is lost in the first iterations.

2.2. Kernel functions

The approximations are highly influenced by the type and parameterization of the kernel functions. These functions transform the original inputs x_m , $1 \leq m \leq g$ into a k -dimensional space where a linear correlation between outputs and transformed inputs is more accurate.

There are several kernel functions, such as polynomial, trigonometric, logarithmic, radial basis, etc. The algorithm proposed in this work uses Gaussian radial basis functions, whose general formula is:

$$\varphi_j(\mathbf{x}) = e^{-(\alpha \|\mathbf{x} - \mathbf{c}_j\|)^2} \quad (29)$$

where \mathbf{c}_j is $1 \times g$ vector called a center or centroid, with g being the number of input variables.

The functions measure the distance of each variable in the input state space to k centers and then transform each distance using a Gaussian function. The Euclidean norm is typically used to measure the distance. Nevertheless, any other norm can be considered.

The number and relative position of centers \mathbf{c}_j are important. The number of centers determines the dimension of the matrices in the linear regression, and also the smoothing and overfitting properties of the approximation. There must be enough centers and they must be placed to “cover” the entire state space, avoiding blanks in regions of interest while keeping the number of centers to a minimum. A random setting of the centers in the state space can serve in some cases. However, it is usually more appropriate to place the centers using techniques such as Latin Hyper Cube Sampling, as it is done in the implemented algorithm.

2.3. ADP and linear regressions in the context of distributed computing

In the context of ADP, a large amount of the calculation is attributed to the Monte Carlo simulations. Consequently, distributed computing techniques can be exploited to dramatically improve and speed up the optimization. The optimal point between thread opening and result gathering basically depends on the hardware system, network latency and simulation times. With time-consuming simulations, it is often useful to parallelize the program after each approximation update, leaving several threads gather more data in parallel before any update. But if the simulation time is short, the overhead times to distribute the calculation are usually larger than the speed up achieved through parallel simulation. In such cases, another approach should be used. A practical approach is to run parallel ADP algorithms for the same problem in several threads in order to combine and synchronize all approximations at regular time intervals. The combination of the results of two or more independent threads is based on the same mathematical basis as the recursive update, provided that the non-linear parametric functions are identical for all threads.

As shown in equation (4), the optimal parameter vector is:

$$\boldsymbol{\theta} = [\mathbf{Z}^T \mathbf{Z}]^{-1} \mathbf{Z}^T \mathbf{Y} = \left[\sum_{i=1}^n \mathbf{z}_i^T \mathbf{z}_i \right]^{-1} \cdot \left[\sum_{i=1}^n \mathbf{z}_i^T \mathbf{y}_i \right] \quad (30)$$

$$\boldsymbol{\theta} = [\mathbf{B}]^{-1} \mathbf{C}, \mathbf{B} = \sum_{i=1}^n \mathbf{z}_i^T \mathbf{z}_i, \mathbf{C} = \sum_{i=1}^n \mathbf{z}_i^T \mathbf{y}_i \quad (31)$$

where each \mathbf{z}_i and \mathbf{y}_i represent the transformed inputs and the outputs respectively of the i -th simulation of a total of n data points. As i increases, the matrices \mathbf{B} and \mathbf{C} summarize all the

regression data collected. The data simulated by several threads of ADP can be easily combined using these matrices. As parallel ADP threads a and b gather different sets of data, the parameter vectors that they approximate and update are different

$$\boldsymbol{\theta}^a = [\mathbf{B}^a]^{-1} \mathbf{C}^a = \left[\sum_{i=1}^v \mathbf{z}_i^T \mathbf{z}_i \right]^{-1} \cdot \left[\sum_{i=1}^v \mathbf{z}_i^T \mathbf{y}_i \right] \quad (32)$$

$$\boldsymbol{\theta}^b = [\mathbf{B}^b]^{-1} \mathbf{C}^b = \left[\sum_{i=v}^n \mathbf{z}_i^T \mathbf{z}_i \right]^{-1} \cdot \left[\sum_{i=v}^n \mathbf{z}_i^T \mathbf{y}_i \right] \quad (33)$$

However, saving the matrices \mathbf{B} and \mathbf{C} allows to recombine all the data later as

$$\boldsymbol{\theta} = \left[\sum_{i=1}^v \mathbf{z}_i^T \mathbf{z}_i + \sum_{i=v}^n \mathbf{z}_i^T \mathbf{z}_i \right]^{-1} \cdot \left[\sum_{i=1}^v \mathbf{z}_i^T \mathbf{y}_i + \sum_{i=v}^n \mathbf{z}_i^T \mathbf{y}_i \right] \quad (34)$$

$$\boldsymbol{\theta} = [\mathbf{B}^a + \mathbf{B}^b]^{-1} \cdot [\mathbf{C}^a + \mathbf{C}^b] \quad (35)$$

This result can be generalized to any number of parallel threads.

In general, data gathered from several parallel algorithms can be recombined by updating and saving the matrices \mathbf{B} and \mathbf{C} after each simulation i of each thread j . The update formulas are:

$$\mathbf{B}_i^j = \mathbf{B}_{i-1}^j + \mathbf{z}_i^T \mathbf{z}_i \quad (36)$$

$$\mathbf{C}_i^j = \mathbf{C}_{i-1}^j + \mathbf{z}_i^T \mathbf{y}_i \quad (37)$$

After a fixed simulation time, they can be synchronized and recombined, so all the threads share the same data gathered:

$$\mathbf{B} = \sum_{j=1}^{th} \mathbf{B}^j \quad (38)$$

$$\mathbf{C} = \sum_{j=1}^{th} \mathbf{C}^j \quad (39)$$

Finally, each thread can restart the simulation process using as starting values for θ and A

$$A = [B]^{-1} \quad (40)$$

$$\theta = [B]^{-1} C \quad (41)$$

The final approximation is not as good as the one carried out by a single thread for the same amount of simulations. This is due to the fact that the decisions taken by the single thread algorithm have always all the information gathered up to the decision point while the multithread algorithm lacks the information gathered by other threads since the last synchronization. Nevertheless, with the correct choice of synchronization and simulation cycles, the overall optimization process can run much faster.

3. Electricity trading

3.1. Electricity markets

Since about two decades, the power industry had undergone a major restructuring in many countries. The former vertically-owned and centrally-planned electricity utilities have been unbundled in separate and independent business segments: the generation, transmission and distribution sectors. Unlike the latter two segments, which have remained as natural monopolies under regulation, power generation is now a business subject to competition in the open marketplace [5].

Electricity is a commodity with some very distinctive features. First, modern societies are exceedingly dependent on a continuous delivery of electrical power, placing a very high value to supply reliability. Because electrical energy cannot be economically stored in considerable amounts, production and consumption must be continuously in perfect balance. In addition, power demand is nearly price irresponsive (inelastic) in the short-run. Therefore, power prices often escalate to very high quotes (price spikes) when supply/demand conditions are tight. Most of these circumstances are short-lived, e.g. equipment outages, transmission congestion climatic events, etc., and price rapidly reverse to normal levels [6].

The exceptionally high volatility of electricity prices imposes high financial risks when trading electricity and forces generation companies to make decisions and commitments under high uncertainty. Thus, stochastic modeling and optimal decision making under uncertainty are key tasks in modern power trading and power risk management [7].

3.2. Trading power in spot markets

Currently, electricity is traded forward on bilateral negotiations and on centrally-run electronic platforms like power exchanges. Electricity can be traded in anticipation from one to three

years (mostly OTC) to several months ahead. Shorter term forward markets negotiating electricity with delivery horizon of weeks, or even one day in advance in the so-called day-ahead markets are quite common. Market liquidity is typically higher as the contracting horizon shortens.

Because of technical requirements of real time system balance, spot power markets are always centralized and run by entities in charge of the physical operation of the system to keep reliability and security. Equilibrium spot prices are computed each 5 min, 30 min or in hourly basis according to the realized power demand and the last bid accepted for keeping the system balance in real time. Rigorously, locational marginal prices are set in order to account for transmission constraints. Therefore, spot prices reflect the actual conditions of the system at the time of delivery.

Though spot prices are subject to high uncertainty, liquidity of this market is warranted as the generator can always sell its production at real time prices. For this reason, the spot market is regarded a last-resort market. One advantage of participating only in this market is that unavailability of the generating unit does not have financial consequences for the generator other than the opportunity cost of the lost production.

Let $p_s(t)$ the prevailing price at the t -th time interval in the spot market, $P_s(t)$ the power delivered by the generating unit and P_{\max} its generating capacity. Under the hypothesis the generator is price-taker and its marginal costs of generation, denoted by $MC(t)$, are constant with the rate of production, the optimal operating policy is:

$$P_s(t) = \begin{cases} 0 & \text{if } p_s(t) < MC(t) \\ P_{\max} & \text{if } p_s(t) > MC(t) \end{cases} \quad (42)$$

The operating profit $B_s(t)$ the generator obtains in the spot market by implementing this optimal production policy can be written as:

$$B_s(t) = \max[p_s(t)P_{\max} - MC(t)P_{\max}, 0] = \max[(p_s(t) - MC(t))P_{\max}, 0] \quad (43)$$

This equation shows the operating flexibility of generator to alter its output in response to the spot price in order to avoid operating losses if prevailing spot prices drop below marginal costs.

Figure 3 depicts the discontinuous nature of the profit function $B_s(t)$ when participating in the spot market, i.e. $B_s(t) \geq 0$ for all prices. Indeed, this profit function can be assimilated to a call option with strike price MC . Figure 3 also schematically illustrates the probability density function (pdf) of the spot price of electricity $f(p_s)$. This function is typically highly right-skewed and presents strong leptokurtosis.

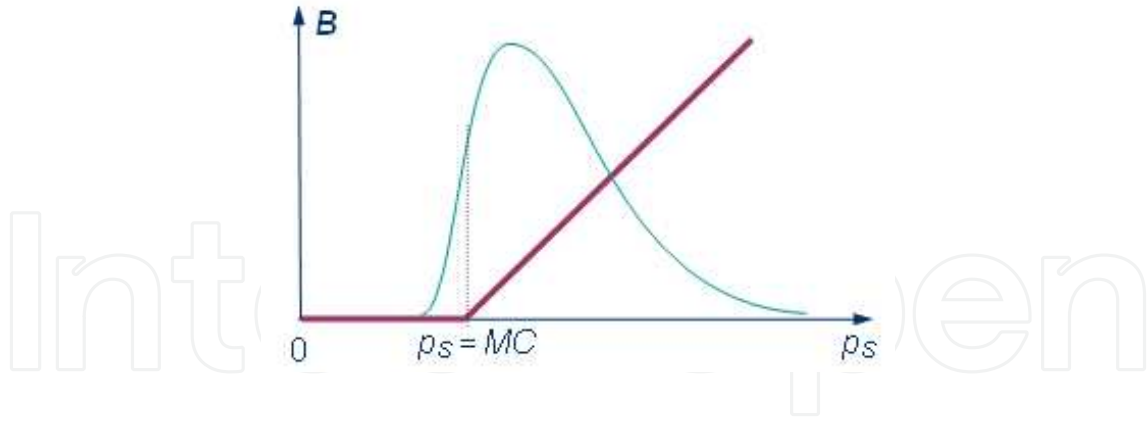


Figure 3. Profit function of selling power in the spot market

The expected value of the profit in the spot market per unit of generating capacity $b_s(t)$ under the optimal operating policy is described by the following equation:

$$E[b_s] = \int_0^{MC} [0 \cdot p_s f(p_s)] dp_s + \int_{MC}^{\infty} [(p_s - MC) f(p_s)] dp_s \quad (44)$$

$$E[b_s] = \int_{MC}^{\infty} [(p_s - MC) f(p_s)] dp_s \geq E[p_s] - MC \quad (45)$$

It is noteworthy to observe that the first term in equation is the probability of obtaining a zero profit in the spot market. Note also that $E[p_s] = MC$ only if $MC = 0$. By selling the production in the spot market, the generator never incurs in operating losses, i.e. $\Pr(b_s < 0) = 0$ as it can immediately stop production if $p_s < MC$.

We consider now the more general case where generating units are unavailable, either for planned or unplanned reasons, during a fraction of the time. Let p the failure probability and $q = 1 - p$ the probability of the unit being available, provided the failure and operating states are the only two mutually exclusive states in which the generator resides. We further assume that the price level and the state of the generator are statistically independent. Under these considerations, the generator cannot always capture the spread $p_s - MC$ and thus the probability of obtaining a positive profit will decrease accordingly. The expected operating profit under these conditions is then:

$$E[b_s] = q \int_{MC}^{\infty} [(p_s - MC) f(p_s)] dp_s \quad (46)$$

The probability of having zero profit $\beta_0 = \Pr(b_s = 0)$ is given by:

$$\beta_0 = p \int_0^{\infty} f(p_S) dp_S + q \int_0^{MC} f(p_S) dp_S = p + q \int_0^{MC} f(p_S) dp_S \quad (47)$$

The resulting probability density function of the hourly operating profit is illustrated in Figure 4. Despite the high variance of the profits, the function clearly shows that the generator cannot lose money when participating in the spot market, even if the unit is technically unavailable.

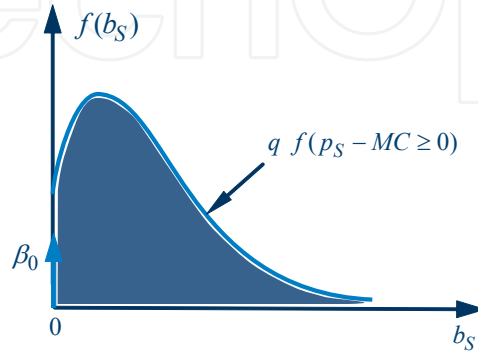


Figure 4. Probability density function of the operating profit in the spot market of electricity

3.3. Trading electricity in forward markets

Given the dramatic volatility of real time electricity prices, a major activity of power trading is structuring hedging strategies by means of tradable derivative instruments like future and option contracts [8]. A power company owning a set of generating units may decide either to sell electricity in advance at a fixed price in a forward market, or wait to the time of delivery and receive the spot price. Deciding on committing production forward or being exposed to volatility of real-time power prices has however a drastic impact on risk.

By selling forward its production, the generator may hedge against a sudden decline of electricity spot prices during the delivery horizon, thereby securing an operating margin. This hedging strategy isolates the generator from the price risk. However, the generator in exchange resigns the opportunity of selling electricity in the spot market if high prices happen.

Electricity markets are typically arranged under a two-settlement system. This approach preserves the economy and efficiency of the physical operation of the power system from any financial commitment the market players have entered into in the past. Under the two-settlement scheme, only deviations from contractual obligations are negotiated in the spot market.

The revenue from the forward contracting is given by the volume sold P_F times the price p_F agreed in the forward contract, i.e. $R_F = p_F P_F$. On the other hand, the revenue captured by selling in the spot market is given by $R_S = p_S \Delta P = p_S (P_S - P_F)$. So, the total revenue R_F from forward contracting and delivering power in the spot market is given as:

$$R_T = R_F + R_S = p_F P_F + p_S (P_S - P_F) = P_F (p_F - p_S) + p_S P_S \quad (48)$$

We can observe the utility of forward contracts by inspecting this equation. If generator delivers in the spot market an amount equal to its contractual obligation, i.e. $P_S = P_F$, the total revenue is set equal to $p_F P_F$ irrespective of the fluctuations of the spot price p_S .

At the time of delivery, and assuming the generator is price-taker, the term $P_F (p_F - p_S)$ is fixed and represents the profit of the forward contract against the spot market. Therefore, the profit-maximizing production policy is the same and given by the spot price, irrespective of the contractual obligations. The profit the generator can make by selling electricity in forward markets is given by the expression:

$$B_F = P_F (p_F - p_S(t)) + \max [P_S (p_S(t) - MC), 0] \quad (49)$$

In Figure 5A, a probability density function of the spot price is depicted. In the following, it is assumed that the forward market price is an unbiased estimator of the spot price at the time of delivery. Therefore, the condition $p_F = E(p_S)$ holds. In the forward contract, the generator makes a profit for unit of capacity $b_F = p_F - MC$, assuming $p_F > MC$. Otherwise, the generator is better by avoiding entering into a forward obligation with negative profit. For realizing this profit, the generator must be able to deliver in the spot market the contracted volume in the exact amount. This profit level is achieved as long as the spot price exceeds the marginal cost, i.e. the probability of making this profit is $\Pr(b_F = p_F - MC) = \Pr(p_S > MC)$.

Graphically, this probability is represented by the dark grey area under the pdf of the spot price (cf. Figure 5A). The generator can make additional profits in the forward contract, $b_F = p_F - p_S > p_F - MC$, each time the spot price drops below the marginal cost, i.e. $p_S < MC$. In fact, the generator is better buying replacement power in the spot market than incurring in fuel costs generating with its own facilities. Figure 5B illustrates the pdf of the profit of a forward contract. When compared with the profit distribution in the spot market (cf. Figure 5), it is easily noticeable the drastic reduction of the profit variance under forward contracting. The forward obligation sets a floor for profits, reducing dramatically dispersion of results and thereby the price risk. In exchange, the generator also foregoes the chance of profiting at times of high power prices in the spot market. The expected profit of a forward contract in terms of the pdf of the spot price $f(p_S)$ can be expressed as:

$$\begin{aligned} E[b_F] &= (p_F - MC) \int_{MC}^{\infty} f(p_S) dp_S + \int_0^{MC} [(p_F - p_S) f(p_S)] dp_S = \\ &= p_F - MC \int_{MC}^{\infty} f(p_S) dp_S - \int_0^{MC} [p_S f(p_S)] dp_S \end{aligned} \quad (50)$$

It can be mathematically demonstrated that under rational expectations and efficiency of forward markets, i.e. the forward price is an unbiased estimator of the spot price $p_F = E(p_S)$,

the condition $E(b_F) = E(b_S)$ holds [1]. This means that for a risk-neutral generator both policies, either selling in the spot market or hedging in forward markets, are entirely equivalent. Nevertheless, for risk-averse players (which is the rule in real market settings), the hedged strategy is clearly preferred as profit expectation remain unaltered while price risk is eliminated.

3.4. Delivery risk in forward contracting

If we consider again unplanned outages of generating units, hedging price risk in the forward markets exposes generating companies to other class of risk, i.e. delivery risk, which also referred as quantity or volume risk. We further examine this important issue. When a generator under a contractual obligation is unable to deliver in the spot market the contracted amount, i.e. $P_S \neq P_F$, the generator is forced to buy replacement power in the spot market at the prevailing price at that time. This may configure a very significant loss if while the generator is down the spot price is considerable higher than its own marginal costs, i.e. $p_S \gg MC$. Under this situation, the generator may be compelled to buy very expensive replacement power to honor the obligation, incurring in a potentially high financial loss. It is interesting to note that if when the unit is unavailable spot prices are lower or equal than the marginal cost, the generator can even make an extra profit $b_F = p_F - p_S > p_F - MC$. The probability density function of the forward position under consideration of positive failure probability and the associated delivery risk is illustrated in Figure 5C.

The expected profit can be computed as the expected value of the contract under the hypothesis of fully reliable unit times the probability of being available:

$$E[b_F] = q \left[(p_F - MC) \int_{MC}^{\infty} f(p_S) dp_S + \int_0^{MC} (p_F - p_S) f(p_S) dp_S \right] \quad (51)$$

As for modern units $q \cong 1$, the change in the expected profit due to unit unavailability is typically negligible. However, downside risk increases substantially.

Assuming statistical independence between the unit's failure and the level of spot prices, the probability of incurring in losses is given by:

$$\Pr(b_F < 0) = p \Pr(p_S > p_F) \quad (52)$$

and the conditional expectation on the value of losses can be written as:

$$L = \int_{p_F}^{\infty} (p_F - p_S) f(p_S) dp_S \mid b_F < 0 \quad (53)$$

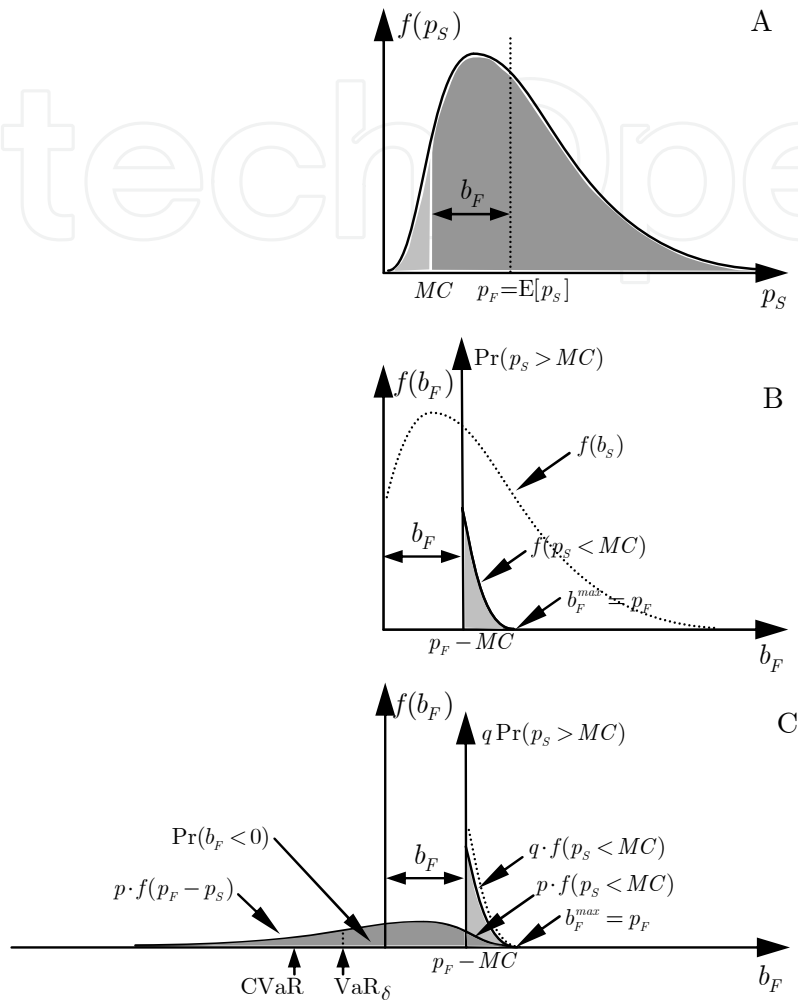


Figure 5. Probability density function of operating profits for a forward contract

From the profit pdf of the forward contract $f(b_F)$, the downside risk metrics, namely the value at risk (VaR) and the conditional value at risk (CVaR) for a δ confidence level can respectively be computed as:

$$\Pr(b_F < -\text{VaR}_\delta) = \int_{-\infty}^{-\text{VaR}_\delta} f(b_F) db_F = \delta \quad (54)$$

$$\text{CVaR}_\delta = \int_{-\infty}^{-\text{VaR}_\delta} b_F f(b_F) db_F \quad (55)$$

4. Problem modeling

4.1. Problem formulation

Let consider a small generation portfolio running in an electricity market. The power company owning the generation portfolio wants to determine the best-selling strategy of the energy production, which would maximize the expected profit while financial risk is constrained. Any trading strategy x is defined by the amount of energy to be sold in each different available selling instrument i in the electricity market, for example, the spot market, day-ahead obligations, annual forward contracts, etc.

It is important to notice that the trading strategy sets the amounts of energy committed in every forward instrument, but only estimates the amount of energy to be actually sold in the spot market. Indeed, actual energy production is stochastic and depends on technical availability of generating units and the spread between spot and fuel prices. Suppose that whatever trading strategy is decided now, it could be changed in future decision stages in order to rebalance the portfolio. Composition of the portfolio can be rebalanced only at a cost however, i.e. the transaction costs. The process is then a sequence of balancing decisions determined by the trading strategy, each followed by a stochastic reward according to the position taken in the market. The process is over after a number of periods n .

The optimization of the trading strategy can mathematically be formulated as a stochastic non-linear problem involving the maximization of the expected profit accrued by the generation portfolio across all instruments i and time intervals t :

$$\max_x \left[\mathbb{E} \left[\sum_t \left[\sum_i I^w(x_{i,t}) - \sum_i T^w(x_{i,t}, x_{i,t-1}) - C_t^w \right] \right] \right] \quad (56)$$

subject to the following constraints:

$$\sum_{i \neq spot} x_{i,h} + x_{spot,h}^w = E_h^w, \quad \forall h \quad (57)$$

$$\sum_{i \neq spot} x_{i,t} \leq E_{max,t}, \quad \forall t \quad (58)$$

$$x_{i,t} \geq 0, \quad \forall t, i \neq spot \quad (59)$$

$$Risk_W \left[\sum_i I(x_{i,t}) - \sum_i T(x_{i,t}, x_{i,t-1}) - C_t^w \right] \leq Risk_{max,t}, \quad \forall t \quad (60)$$

where:

E : Expected value operator

w : Monte Carlo sample path

W : Total number of Monte Carlo samples

t : Time period beginning after a balancing decision

h : Hourly time step

$x_{i,h}$: Energy sold and instrument i in hour h

$x_{spot,h}^w$: Energy sold in spot market in hour h and Monte Carlo sample w

E_h^w : Energy generated in hour h and Monte Carlo sample w

$E_{max,t}$: Maximal energy that can be generated in period t

$x_{i,t}$: Energy to be sold by instrument i in period t

$I^w(x_{i,t})$: Revenue due to energy the sold by instrument i in period t and Monte Carlo sample w

$$I^w(x_{i,t}) = \begin{cases} x_{i,t} \cdot p_{Fi,t}^w, & i = forward \\ \sum_{h \in t} x_{i,h} \cdot p_{Sh}^w, & i = spot \end{cases} \quad (61)$$

$p_{Fi,t}^w$: Effective future price of instrument i in period t and Monte Carlo sample w

p_{Sh}^w : Spot price in hour h and Monte Carlo sample w

C_t^w : Costs of energy in period t and Monte Carlo sample w

$$C_t^w = MC \cdot \sum_{h \in t} E_h^w \quad (62)$$

MC : Constant marginal cost of generation.

$T^w(x_{i,t}, x_{i,t-1})$: Transaction costs due to the change in the amount of instrument i held in the portfolio after the rebalancing decision at the beginning of period t

$$T^w(x_{i,t}, x_{i,t-1}) = \begin{cases} 3\% \cdot p_{Fi}^{w,h0_t} \cdot (x_{i,t} - x_{i,t-1}), & i = forward \\ 0, & i = spot \end{cases} \quad (63)$$

$p_{Fi}^{w,h0_t}$: Forward price of instrument i and Monte Carlo sample w at the beginning of period t .

The constraint (57) represents the hourly energy balance for all Monte Carlo samples and forces the generator to settle in the spot market differences between the energy sold in forward markets and actual production. Constraints (58) and (59) are introduced to avoid financial positions without physical counterpart, i.e. avoid the generator to take speculative positions by selling in forward markets more energy than the generation portfolio can produce. These constraints may be replaced by capital restrictions as regulations often allow financial trading without physical position. Finally, constraint (60) represents the limit to the financial risk associated to the selling strategy within each period t . In order to limit risk over the horizon time, the selected risk metric must be coherent [9] ensuring subadditivity. There are several downside risk measures that fulfill this requirement among which the Conditional Value at Risk (CVaR) is the most widely used (cf. equation 55).

The equation (61) represents the revenue generated by each forward contract and the revenue (or cost) due to selling (or buying) differences between energy sold in futures and the real generation in the spot market forced by restriction (57). The equation (62) represents the costs of the operating policy, which is independent of the forward obligation for a price taker as demonstrated in Section 3. Therefore, the three equations (57), (61) and (62) calculated for the whole Monte Carlo set represent then the profit calculated by equation (49) of Section 3. Finally the transaction costs in equation (63) are assumed to be the 3% of the total transacted value only for forward contracts, and not existent for the spot market.

4.2. Modeling stochastic spot and forward electricity prices

The problem formulation relies on Monte Carlo simulations to represent uncertainty on the future development of key variables. In addition, stochastic simulations are used to confront the algorithm with mapping scenarios for approximating both, the Value and the Risk functions.

A synthetic ensemble of 2000 annual realizations of hourly power prices in the spot market were generated by means of spectral representation techniques [10]. Forward prices and spot prices are not statistically independent. The forward prices corresponding to each spot price sample are calculated considering both, the expected value of the spot price and the mean value of each spot price time series. To simulate the changes in the forward prices accounting for the correlation to each sample of the spot prices a simple model is introduced.

Under perfect competition and rationality on the expectation formation, the price of a forward product should converge to the mean expected spot prices for the delivery period. This is relatively easy to calculate for the first hour of the simulated time. Assuming that the whole Monte Carlo set of spot prices was simulated taking the same price forecast as the market, the price of any forward should be the mean spot prices for the delivery period:

$$p_{FAi}^{w,h} = \frac{1}{\left(H_i - (h_i^d - 1)\right)} \sum_{j=h_i^d}^{H_i} \mathbb{E} \left[p_S^j \right] = \frac{1}{W \left(H_i - (h_i^d - 1)\right)} \sum_{j=h_i^d}^{H_i} \sum_{w=1}^W p_S^{w,j} \quad (64)$$

$$h_i^d = \begin{cases} H0_i & \text{if } h \leq H0_i \\ h & \text{if } h > H0_i \end{cases} \quad (65)$$

where:

H_i : delivery period for instrument i

$H0_i$: first hour of delivery period for instrument i

For the first hour, this model represents the expectation of the market for each delivery period. However, this expectation should change according to the values that the spot prices take in each sample and the information gathered by a *virtual* market taking place in each particular sample path. If perfect foresight is assumed, each virtual market could calculate without uncertainty the hourly forward price for its particular spot prices sample path simply as:

$$p_{FBi}^{w,h} = \frac{1}{H_i - (h_i^d - 1)} \sum_{j=h_i^d}^{H_i} p_S^{w,j} \quad (66)$$

The equation (64) represents a model where any additional information that arrives as the spot price of the particular sample path is different from the forecasted in the first hour is dismissed by the virtual market. Thus, this model is representative of reality only for the first hour of simulation where no additional information could have been gathered by the virtual markets. On the other hand, the equation (66) represents a model where all the additional information is obtained beforehand for each Monte Carlo sample. Likewise, this model is suitable only for the last hour of simulation where all information is already known by the virtual market within each Monte Carlo sample. Finally, the two models can be combined, in order to simulate the forward price dynamics in correlation to the spot prices of each Monte Carlo sample path. In this work, it is assumed that the information gathered by each virtual market grows linearly, augmenting for each hour¹. Then, equations (64)² and (66) are combined by a weighted average:

$$p_{FCi}^{w,h} = \frac{(H - (h_i^d - 1))}{H_i} \cdot p_{FAi}^{w,h} + \frac{(h_i^d - 1)}{H_i} \cdot p_{FBi}^{w,h} \quad (67)$$

$$p_{FCi}^{w,h} = \frac{(H - (h_i^d - 1))}{H_i} \cdot \frac{1}{W(H_i - (h_i^d - 1))} \sum_{j=h_i^d}^{H_i} \sum_{w=1}^W p_S^{w,j} + \frac{(h_i^d - 1)}{H_i} \cdot \frac{1}{H_i - (h_i^d - 1)} \sum_{j=h_i^d}^{H_i} p_S^{w,j} \quad (68)$$

¹ The assumption that the information is linear with time could be replaced with a more complex information model, such as a function of the cumulated difference between the initial forecasted spot price and the particular spot prices of each Monte Carlo sample path.

² The prices for the first hour in a real market situation should consider the real market prices.

Finally, a *contango* situation is considered in the forward market. A risk premium of 8% in excess of expected spot prices is considered for the forward prices in the future market to compensate for the volatility risk. This premium reduces linearly during the delivery period of the future and becomes zero at the last hour of delivery.

The model for future prices for each instrument i , hour h and sample w is thus as follows:

$$p_{Fi}^{w,h} = \left(1 + \beta \frac{H_i - h_i^d}{H_i} \right) \cdot \left[\frac{(H - (h_i^d - 1))}{H_i} \cdot \frac{1}{W(H_i - (h_i^d - 1))} \sum_{j=h_i^d}^{H_i} \sum_{w=1}^W p_S^{w,j} + \frac{(h_i^d - 1)}{H_i} \cdot \frac{1}{H_i - (h_i^d - 1)} \sum_{j=h_i^d}^{H_i} p_S^{w,j} \right] \quad (69)$$

where β is the risk premium paid in excess to the expected spot price and set $\beta=8\%$.

4.3. Reliability model of the generation units

Other relevant source of uncertainty considered is the random failure of the generating units. The stochastic model of generator outages is built considering that the unit can reside in four mutually exclusive states: Operation (required), Reserve (not required), Unavailable (required) and Unavailable (not required), as shown in the diagram of space states in Figure 6 [11].

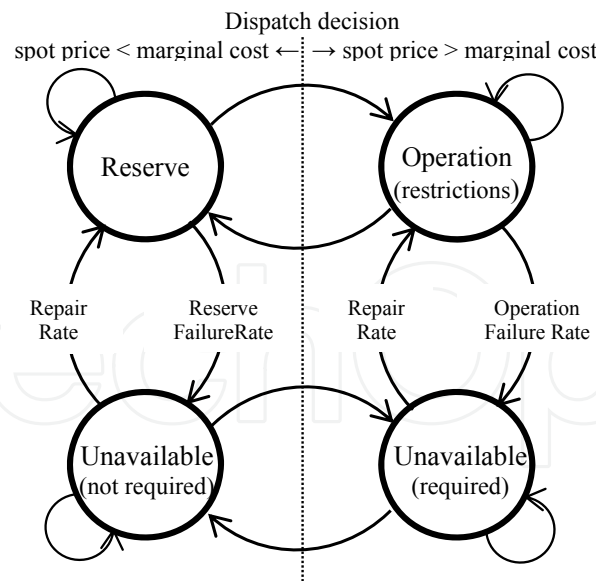


Figure 6. Four-state stochastic model of the generation units

This unit model accounts for the fact that peaking units exhibit higher availability rates. This result is explained by the fact the failure probability is typically very small when the unit is in the stand-by state. A generator is economically called online if its marginal cost of production

is below the prevailing spot prices, following the decision model of equation (42). Variable costs of generation are assumed linear with power output, i.e. marginal costs are constant.

The operation-failure cycles of the generating unit are obtained from a chronological Markovian stochastic simulation. For each spot price sample, a time series of power output is synthesized for every generation unit. The hourly power output is simulated following three steps:

1. Based on failure and repair rates defined by the state the unit resided in the previous hour, a random failure is simulated [12]. If a failure is in place, the output power is set to zero for this particular hour.
2. The dispatch of the unit is simulated, taking into account the marginal cost of generation and the prevailing sample spot price at that time interval. Here perfect foresight of the spot price is assumed in order to decide the dispatch and fulfill the minimal generation times.
3. If dispatched, other unit's technical restrictions are fulfilled, e.g. ramping capabilities.

This chronological stochastic model reproduces with accuracy the dynamics involved in failure and repair cycles of generators, giving the possibility to select different failure rates depending on whether the unit is generating or is in stand-by.

4.4. Risk constraint formulation

As already mentioned before, the financial decision process can be modeled by means of a MDP. Naming profit B^w for each sample the sum of income, cost and transaction cost over all instruments, the objective function (56) becomes:

$$\max_x \left[E \left[\sum_t B^w(x_t, x_{t-1}) \right] \right] \quad (70)$$

Exchanging the order of the expectation operator and the summation, and expanding the summation:

$$\max_x \left[E \left[B_1^w(x_1, x_0) \right] + \sum_{t=2}^n E \left[B_t^w(x_t, x_{t-1}) \right] \right] \quad (71)$$

where n is the total number of time periods considered.

At this point and considering the first term depends only on the initial market position x_0 and the first strategy decision x_1 , the maximization can be decomposed using the Bellman's Principle of optimality as follows:

$$\max_{x_1} \left[\mathbb{E} \left[B_1^w(x_1, x_0) \right] + \max_{x_{t=2 \dots n}} \sum_{t=2}^n \mathbb{E} \left[B_t^w(x_t, x_{t-1}) \right] \right] \quad (72)$$

If x is independent of the Monte Carlo sample, the terms inside the summation over all future periods ($t=2 \dots n$) are simply the expected profit for the period t after a decision $x_{t-1} \rightarrow x_t$. However, the model should take into account that future strategy decisions may be different for each Monte Carlo sample, accounting for adjustments the decision-maker almost certainly would execute to face specific scenarios. Then, the expected profit \bar{B}_t and the decision itself for future decision stages will depend on a set of variables s_t , which represent the variables considered by the decision maker in order to adapt the strategy to a particular situation and the equation (72) becomes:

$$\max_{x_1} \left[\bar{B}_1(x_1, x_0, s_1) + \max_{x_{t=2 \dots n}} \sum_{t=2}^n \bar{B}_t(x_t, x_{t-1}, s_t) \right] \quad (73)$$

Defining the continuation value functions V_t as:

$$V_t(x_t, x_{t-1}, s_t) = \bar{B}_t(x_t, x_{t-1}, s_t) + \max_{x_{t+1}} [V_{t+1}(x_t, x_{t+1}, s_{t+1})] \quad (74)$$

The maximization can be solved by a set of recursive maximizations, each one solving only one decision stage:

$$\max_{x_t} [V_t(x_t, x_{t-1}, s_t)] \quad (75)$$

With this model, the optimization can be decomposed in steps and the dynamic nature of a strategy can be accurately replicated. Despite the fact future trading decisions $x_{t=2 \dots n}$ are considered and optimized, the practical product of this procedure is the new optimal rebalanced state $x_{t=1}$ starting from the previous trading position $x_{t=0}$. The further trading positions are only optimal given the current information available and should be reconsidered later. Therefore, each new trading position ($x_{t=2}, x_{t=3}, \dots, x_{t=n}$) should be the product of a similar optimization incorporating the additional market information available immediately before.

The value functions provide the expected continuation value within a state space defined by the state variables, x_t , x_{t-1} and s_t ³. However, the continuation functions, which are essential to solve the optimization problem, are unknown beforehand. It is here that the ADP approach is introduced to approximate the value and risk functions for the state space.

³ There are several other decomposition methods, some of which exclude the decision as state space variable defining the value functions in a post-decision state space. These approaches make the step maximization sometimes harder but present advantages such as a state space of fewer dimensions. See for example [2].

The set of constraints remains the same as they were already defined for each period t . Nevertheless, special considerations should be made to calculate the risk and to fulfill the risk-constrained optimization. As V_t only account for the expected profit, risk functions R_t should also similarly be calculated for the same state space in order to enforce the risk constraint.

By definition, the linear regression will deliver an approximation that minimizes the mean square error on the entire dataset. In a stochastic setting where the same inputs leads to several different outputs, the regression will accurately estimate the expected value for a given set of inputs, provided the sample size and the approximation order are appropriate. This fits perfect for the case of the value function but for the approximation of the risk function some problems arise.

Let suppose that the CVaR is chosen as risk metric. As the algorithm progresses, new data points are collected, i.e. a set of state variables and its corresponding simulated profits for the period. For approximating the CVaR associated to a particular point in the state space, one approach is to select a subset from the dataset whose input variables are “close” to the point. Then the CVaR is calculated first by sorting the profit values and then taking the mean of those below the specified α -quantile. Now, let suppose that a new data point is simulated and an update of the CVaR approximation is needed. To do so, the process described must be repeated, but now including the new data point. This simple approach has large disadvantages: all the data points must be stored and the mean is not easily updated as old data may be excluded or included of the zone below the α -quantile. These drawbacks are caused by the fact that the CVaR is quantile-based. To solve these difficulties, another solution is envisioned. Instead of approximating directly the CVaR, another risk measure is used to approximate the CVaR within the space state. The risk measure used is called Relative Lower Semideviation (RLS) and it is moment based instead. Hence, it can be updated more easily and without needing to store the entire dataset. These types of risk measures are described in detail in [13-16] and for a stochastic profit P_t have the form:

$$RLS(B_t) = -E[B_t] + a \cdot \sigma_p^-[B_t] \quad (76)$$

$$\sigma_p^-[B_t] = \left(E \left[\max(E[B_t] - B_t, 0)^p \right] \right)^{1/p} \quad (77)$$

where the equation (77) is the negative semideviation of degree p of the stochastic profit P_t .

It can be proven that these moment-based risk measures are coherent if $0 \leq a \leq 1$ and $p \geq 1$. To approximate a CVaR with a 5%-quantile, in this work the parameters used are $a=1$ and $p=9.5$. To compute the approximation, two linear regressions were used to calculate the expectations on the profit and on the negative deviations, which can be updated using the same method proposed for the value functions. In a Monte Carlo scheme, a large amount of data is needed to compute reliable risk estimations. Therefore, the risk constraint should not

be enforced at the early iterations of the ADP algorithm, simply because the sample size of the dataset is small to get a consistent and statistically converged risk value.

5. Numerical case study

5.1. Algorithm validation

With the objective of validating the results of the proposed ADP algorithm, a first simple exemplary case is considered in which a thermal generator sells energy in the spot market and in a future contract. The results of the ADP algorithm were compared with the results of a conventional DP algorithm for which the space-state was discretized appropriately.

The fractions of energy sold in the spot market and in a quarter future contract were optimized considering that the future can be traded during the delivery period. The optimization determines three decision stages during this period, one at the beginning of each month, consisting on sell or buy energy in the future market based on the previous state. The space-state previous to each decision is defined only by the level of future already sold, in order to keep tractable the DP problem.

5.2. Validation results

The results obtained by solving the problem by means of the ADP and DP algorithms are presented in Figure 7. The plots represent the expected profit and the downside risk measured by the CVaR of the optimal strategy as a function of the initial state, i.e. the energy already committed in the future contract at the initial stage. An excellent agreement between the optimal strategies obtained by ADP and DP is evidenced, validating the proposed approach.

It can be noticed that the expected profit rises as the amount of energy sold forward increases. This is caused by the risk premium paid to the generator in the future market, i.e. the mean future prices are higher than the mean spot prices. Additionally, the transaction costs are not compensated by the risk premium; therefore the best trading strategy is to maintain similar involvement in the future market to the initial level without rebalancing the portfolio. This is illustrated in Figure 8, where the optimal decisions for the first month are practically the same when solving with a conventional DP and an ADP approach.

It is noteworthy to observe in Figure 7 that financial risk lessens when forward contracting in the future market increases. This means that for the conventional generator considered, which present a high availability, the delivery risk is lower than the risk of not being dispatched in the spot market. The behavior of the risk curve is closely related to the unit's failure and reparation rates and to the marginal production costs. Generators with low marginal costs are in the first places of the dispatch merit order, and hence the risk of not being dispatched is low. Moreover, high failure rates imply also a higher delivery risk. Out of these relations arise a broad number of risk curves that differ from one generator to another and suggest that considerable risk mitigation by aggregating different generators in a portfolio is possible.

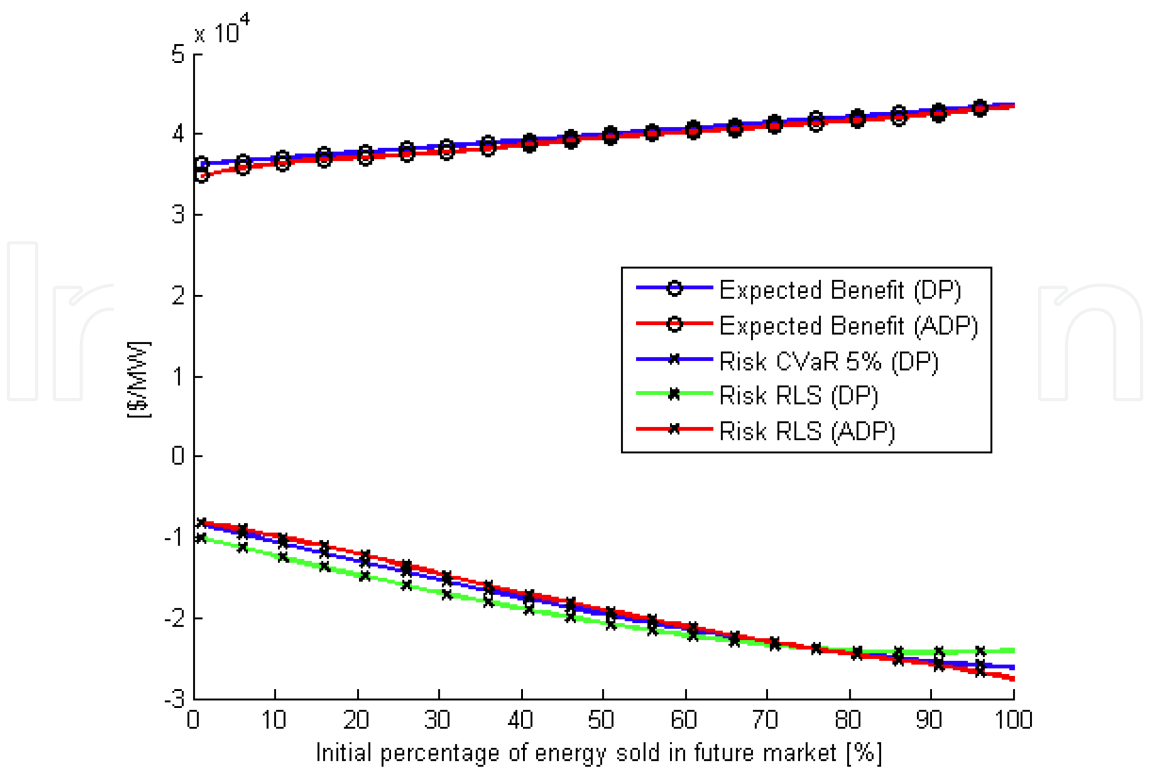


Figure 7. Expected benefit and downside risk of the optimal strategy – Validation case.

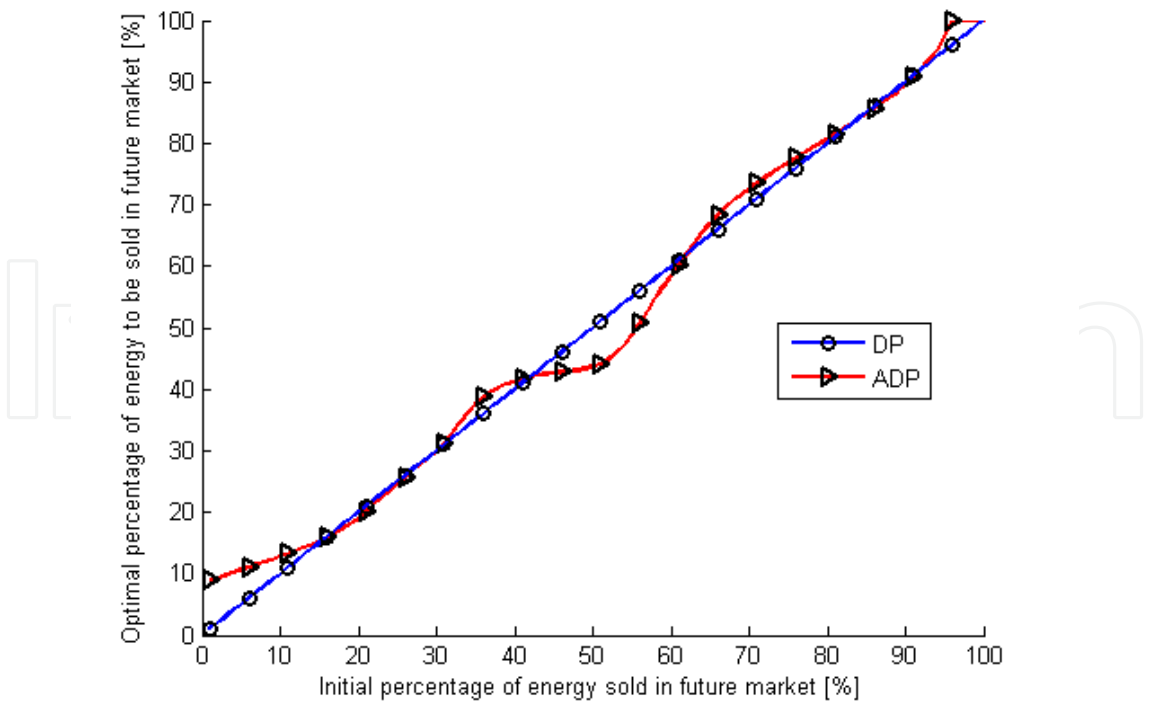


Figure 8. Optimal percentage of energy sold in the future market for the first time period – Validation case

5.3. Optimal policy for portfolio rebalancing

For the base study case, a slightly more complex system was examined. In this case, the generation portfolio comprises five generation units of 2 MW each with a constant marginal generation cost of 50\$/MWh, failure rate of $1/950\text{h}^{-1}$, reserve failure rate of $1/9950\text{h}^{-1}$ and repair rate of $1/50\text{h}^{-1}$. These rates give a failure probability of 5% while the unit is in operation and of 0.5% while in standby.

An annual future, four quarterly futures and the spot market were considered together with 12 monthly trading decision stages for rebalancing the portfolio. The space state for this arrangement is defined by the amount sold forward as well as spot and future prices before each decision point and for each realization. Therefore, the decisions are chosen taking into account additional information about the state of the market for each realization. Even though the simplicity of the example, the additional variables cause a drastic increase in the dimension of the problem and would force an unacceptable coarse discretization of the space-state in order to keep the problem tractable with conventional DP. In this case, the maximum admissible expected loss for the 5%-confidence level is set to $\text{CVaR}_{5\%} = \$20000$ for each decision stage. Since the risk measure is coherent, i.e. sub-additive, the annual risk is less or equal to $\$20000 \cdot 12$ periods = $\$240000$. Transaction costs are set to 3% of the dollar amount contracted in the forward market. The optimization problem was solved on a Beowulf cluster comprising 20 multicore Intel i7 2600K 3.4 GHz processors connected by a Gbit LAN. The 160 available computation nodes were fully exploited and the total computation time was 5 h.

In Figure 9, the results of the ADP algorithm for the optimal strategy on the first rebalancing period with a previous trade equal to zero are illustrated. The amount of energy to be committed in each market is expressed in terms of a fraction of the maximal energy output the generation portfolio would generate without failures in the period, i.e. 10MWh per hour of operation. The prices for the traded futures are also presented in Figure 9 except for the 2nd quarter future which is 45.63\$/MWh and it is not shown as the optimal trade does not include this contract, presumably because the price is too low and it is better to wait for a better price in the spot market and sell in subsequent decisions. Likewise, the expected spot prices for each quarter are displayed, except for the last quarter which is 52.45\$/MWh. Finally, the expected annual profits without considering fixed costs and the risk estimated by the RLS for the first rebalancing period are shown. Note that the expected profit is calculated considering that the following trading decisions are made taking into account the particular sample price realization, capturing the adaptation to the market developments. Thus, the rebalancing decisions for the second up to the last period are not unique.

5.3.1. Sensitivity to unit availability

In order to investigate the sensitivity of the optimal trading strategy to the unit availability, a second case was considered. Under these conditions, all the parameters are identical to the described base case, except for the failure and repair rates. Operation failure rate was set to $1/850\text{h}^{-1}$, a reserve failure rate to $1/9850\text{h}^{-1}$ and repair rate of $1/150\text{h}^{-1}$. With these rates, the failure probability is 15% in operation and 1.5% in standby.

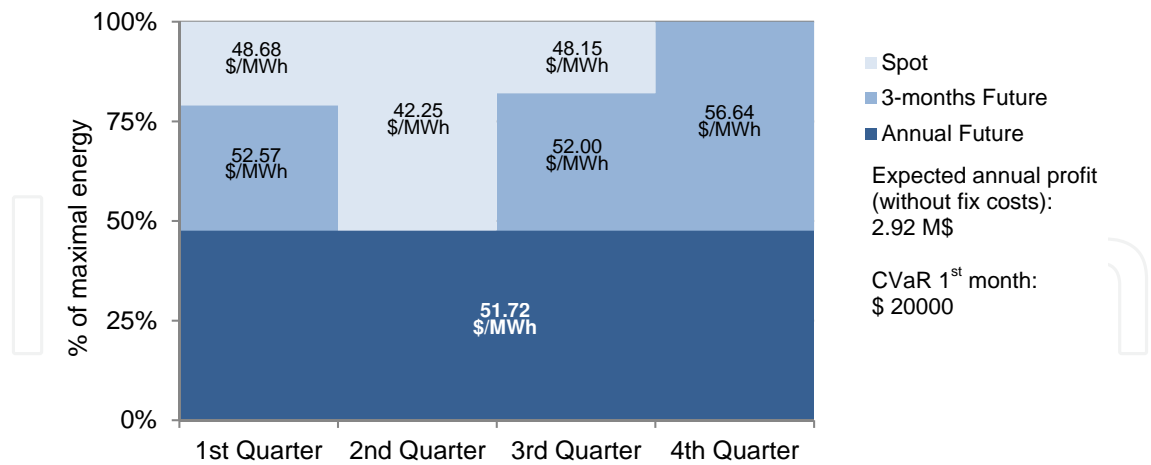


Figure 9. Optimal trading strategy for a 5x2MW generation portfolio

In Figure 10, the optimal trading strategy delivered by the ADP algorithm for the case of decreased unit availability is depicted. The differences on the sell strategy are evident for the annual future and the total amount of energy left to be sold in the spot market. Because of units have more frequent and longer random failures, a long-term commitment is avoided. However, the high prices for the fourth quarter push the sell in future markets up to 100%. In comparison to the base case results, the expected annual profit is slightly lower. The risk for the first month is irrelevant, because the forward commitment is around 50% and the probability of having more than two units (> 40%) unavailable is very low, leading to almost all unit failures can be covered by the remaining available units.

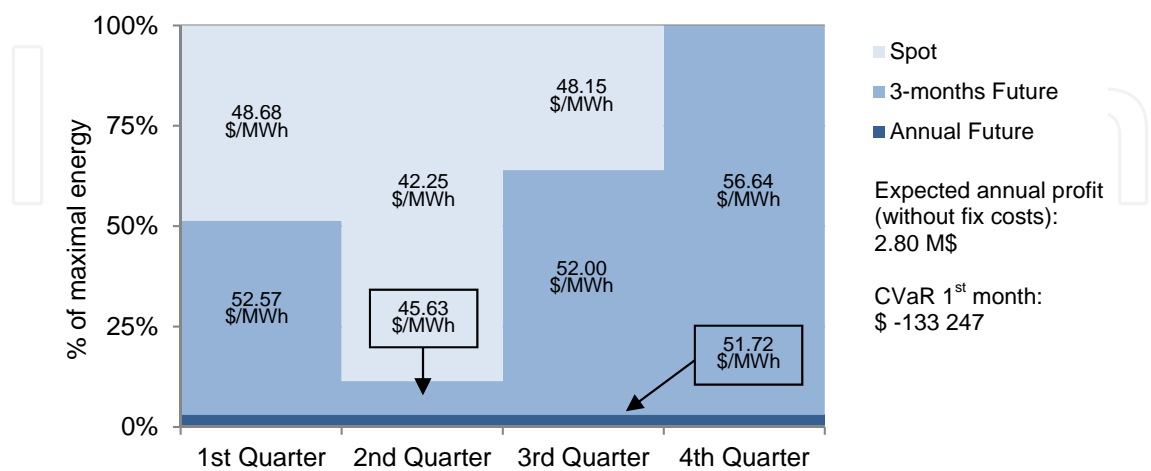


Figure 10. Optimal trading strategy for a 5x2MW generation portfolio with reduced availability

5.3.2. Sensitivity to transaction costs

The third case of study is identical to the base case but considering a transaction cost of 7% of the sold amount instead of 3%. The optimal trading policy for the case of increased transaction costs are shown in Figure 11. In this case, the generator sells only 1.32% of its capacity in an annual future contract due to the irreversibility introduced by the higher transactions costs and the larger contracting volume. Under these circumstances, the risk premium offered in the annual future price render insufficient for attracting the generator to enter in such a long-lasting commitment. On the other hand, the sell volume in quarterly futures is higher than in the base case, staying between 62% and 76% in a rather static trading policy. Under higher transaction fees, it is desirable to be able to rebalance the portfolio in future stages with smaller changes and hence smaller transaction costs. Expectedly, the expected profit is lower due to higher costs. The delivery risk for the first month is negligible, due to the fact that the most likely failures can still be covered by the remaining operative units without buying replacement power in the spot market.

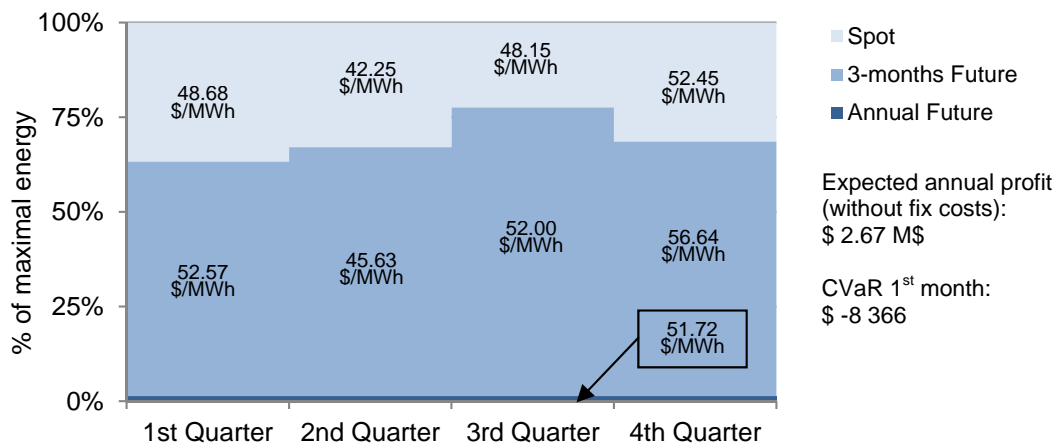


Figure 11. Optimal trading strategy for a 5x2MW generation portfolio with higher transaction costs

6. Conclusions

Optimal decision-making under uncertainty is a field of active research and uppermost relevance in science, engineering and computational finance. Conventional optimization approaches have difficulties and serious limitations for tackling high-dimensional problems often encountered in real world settings. Recent advances in operation research and computation technology opened new possibilities for approaching optimization problems that were considered intractable until recent times. This chapter presents an efficient Approximate Dynamic Programming algorithm for solving complex stochastic optimization problems and amenable for running in a distributed computing environment. The implemented ADP algorithm has been validated against conventional Dynamic Programming for a simple problem.

The proposed algorithm uses Monte Carlo simulation techniques combined with linear regression for successively approximating and refining the continuation and risk functions. A novel and efficient procedure for updating these functions, combining calculations of independent computing threads and without storing the entire datasets, is proposed. This feature enables exploiting the currently widespread multicore processor architectures and deploying the algorithm in large computation clusters.

In order to demonstrate the practicability of the envisioned approach, the proposed algorithm has been applied to find the optimal trading strategy of a power generation portfolio in forward and spot electricity markets. Power trading and risk management is currently a central activity of power companies running in liberalized electricity markets. The probability density functions of the profits a generator would make by participating in either the spot or the forward markets are extremely different. The forms and boundaries of these probability functions have drastic implications for risk when generators get involved in the spot or the forward markets. Generators can hedge price risk of spot markets by contracting forward, but by exposing themselves to delivery risk. Hence, the optimization problem is formulated as the maximization of the expected profit of the trading policy while the downside risk is constrained. For doing so, the generator selects and combines a portfolio of annual and quarterly forward contracts as well as involvement in the spot market. A frictional market with non-negligible transaction costs is considered.

A detailed chronological 4-state reliability model of generating units has been adopted for replicating stochastic behavior of random outages. Large stochastic ensembles of spot prices and forward prices time series have been synthesized for this application. In order to retain subadditivity, downside risk is measured by CVaR. The approximation of CVaR by a moment-based risk metric drastically improves computational efficiency while providing accurate and consistent risk estimations.

Applying ADP-based optimization techniques to electricity markets is a novel undertaking and opens a prospectively fertile avenue for research. In future works, further algorithmic enhancement are foreseen. Application of these methods for designing trading strategies that considers a larger set of available financial contracts as well as generation portfolios comprising renewable resources would provide results and findings of high practical significance.

Acknowledgements

This work was supported in part by the National Scientific and Technical Research Council (CONICET) and the Agency for Promotion of Science & Technology (ANPCyT), Argentina. The financial support of the German Academic Exchange Service (DAAD) is also gratefully acknowledged. The authors also thank the support of the colleagues at the Institute of Power Systems and Power Economics (IAEW), RWTH Aachen, Germany, especially Univ.-Prof. Dr.-Ing. Albert Moser as well as colleagues at Institute of Electrical Energy (IEE), National University of San Juan, Argentina.

Author details

Miguel Gil-Pugliese^{1*} and Fernando Olsina²

*Address all correspondence to: miguel.gil.pugliese@gmail.com

1 Institute of Power Systems and Power Economics (IAEW), RWTH Aachen, Germany

2 Institute of Electrical Energy (IEE), National University of San Juan, Argentina

References

- [1] Olsina F, Larisson C, Garcés F. Hedging and volumen risk in forward contracting [*in Spanish*]. Proceedings of XII ERIAC de CIGRE, Foz do Iguazu, Brazil, May 2007. paper C5.02.
- [2] Powell WB. Approximate Dynamic Programming: Solving the curses of dimensionality. Wiley-Blackwell; 2007.
- [3] Bertsekas DP, Tsitsiklis JN. Neuro-Dynamic Programming (Optimization and Neural Computation Series, 3). Athena Scientific; 1996.
- [4] Bellman RE. Dynamic Programming. Princeton University Press, Princeton, NJ; 1957
- [5] Stoft S. Power System Economics: Designing Markets for Electricity, New York, USA; IEEE Press/Wiley; 2002.
- [6] Hadsell L, Maralhe A, Shawky, HA. Estimating the volatility of wholesale electricity spot prices in the US. The Energy Journal 2004; 25(4) 23-40.
- [7] Bjorgan R, Liu C, Lawarrée J. Financial risk management in a competitive electricity market. IEEE Transactions on Power Systems 1999; 14(4) 1285-1291.
- [8] Stoft S, Belden T, Goldman C, Pickle S. Primer on Electricity Futures and Other Derivatives. Lawrence Berkeley National Laboratory, University of California, LBNL-41098, 1998.
- [9] Artzner P, Delbaen F, Eber JM, Heath D. Coherent measures of risk. Mathematical Finance 1999; 9(3) 203-228.
- [10] Olsina F, Weber C. Stochastic simulation of spot power prices by spectral representation. IEEE Transactions on Power Systems 2009; 24(4) 1710-1719.
- [11] Billinton R, Ge J. A comparison of four-state generating unit reliability models for peaking units. IEEE Transactions on Power Systems 2004; 19(2) 763-768.

- [12] Billinton R, Allan R. Reliability evaluation of power systems. Plenum Press, New York, 1996.
- [13] Rockafellar R, Uryasev S, Zabarankin M. Generalized deviations in risk analysis. Finance and Stochastics 2006; 10(1) 51-74.
- [14] Fischer T. Risk capital allocation by coherent risk measures based on one-sided moments. Insurance: Mathematics and Economics 2003; 32(1) 135-146.
- [15] Pavlo AK. Higher moment coherent risk measures. Taylor & Francis, 2007.
- [16] Rockafellar T, Uryasev S, Zabarankin M. Deviation measures in risk analysis and optimization. University of Florida, Department of Industrial & Systems Engineering Working Paper 2002.