Neural Networks, Stochastic Dynamic Programming and a Heuristic for Valuing Flexible Manufacturing Systems

Markus Feurstein
Martin Natter

Working Paper No. 17
September 1998
September 1998

SFB
‘Adaptive Information Systems and Modelling in Economics and Management Science’

Vienna University of Economics
and Business Administration
Augasse 2–6, 1090 Wien, Austria

in cooperation with
University of Vienna
Vienna University of Technology

http://www.wu-wien.ac.at/am

This piece of research was supported by the Austrian Science Foundation (FWF) under grant SFB#010 (‘Adaptive Information Systems and Modelling in Economics and Management Science’).
Neural Networks, Stochastic Dynamic Programming and a Heuristic for Valuing Flexible Manufacturing Systems

Markus Feurstein  
Martin Natter  

September 14, 1998

Abstract

We compare the use of stochastic dynamic programming (SDP), Neural Networks and a simple approximation rule for calculating the real option value of a flexible production system. While SDP yields the best solution to the problem, it is computationally prohibitive for larger settings. We test two approximations of the value function and show that the results are comparable to those obtained via SDP. These methods have the advantage of a high computational performance and of no restrictions on the type of process used. Our approach is not only useful for supporting large investment decisions, but it can also be applied in the case of routine decisions like the determination of the production program when stochastic profit margins occur.

Keywords

Real Options, Neural Networks, Capital Budgeting, Simulated Annealing, Flexible Manufacturing Systems, Dynamic Programming

1 Motivation

While in finance the importance of option valuation is well established, option valuation techniques have recently gained significance in capital budgeting, too. A research direction called "Real Options" deals with the use of such techniques for valuating the possibilities to alter the mode of operation of an irreversible investment project in response to — a priori stochastic — changes of the economic environment. For projects that embed only one of several simple types of "flexibility" and involve traded commodities, analytical option valuation techniques developed in finance can be used directly. However, most real-world investments permit several modifications (e.g., when to invest, to abandon or temporarily shut down the project) whose values are usually not independent (see e.g., [Kulatilaka (1995b), Kulatilaka (1988)]). The assumption that the value underlying the option follows a Geometric Brownian Motion permitting
analytical solutions to financial options is much less established for Real Options and often a multivariate stochastic process is necessary to describe the relevant environment.

In the case of such general Real Options valuation problems it seems, therefore, necessary to resort to approximate, numerical methods. In finance such methods are well-known. [Geske and Shastri (1985)] group them into methods that approximate the underlying process directly (Monte Carlo Simulation and binomial process) and those that solve the partial differential equation derived as partial equilibrium condition resulting from no-riskless-arbitrage (finite difference methods and numerical integration). Special numerical methods for valuing Real Options were developed by [Kulatilaka (1995a)] and [Kamrad (1995)]. Both are finite, discrete time dynamic programming formulations, where the latter considers only the case of a trinomial lattice approximation of Geometric Brownian Motion. All these methods assume that the type of the underlying stochastic process is given and that its parameters are estimated. Furthermore, in the case of a multivariate stochastic process, the state space grows combinatorially so that the backward recursion procedure used to solve the dynamic program quickly becomes computationally demanding.

In this paper, we investigate two approximate methods that are based on observed trajectories of the stochastic process only. For the assessment of these methods, we derive the optimal solutions also via stochastic dynamic programming. The main advantages of such approximate methods are their computational tractability and the fact that there are no restrictions on the type of process used. The first approximate method we discuss estimates a single (threshold) parameter that can be interpreted as the proportion of switching costs to be subtracted from the stochastic variable in order to determine the outcome (mode). The basic idea of the second approach is to make use of a specific class of (feed forward) Neural Networks with powerful theoretical approximation properties and to learn the value function of the dynamic program.

2 Methods for Real Option Valuation

2.1 Stochastic Dynamic Programming

Given that the owner of a project has the possibility to change its operation mode in a persistent way, it is clear that the value of such a project and the optimal decision rule that specifies an optimal action for each possible time/environment combination have to be determined simultaneously via the solution found by a dynamic program. In particular, our model describes a flexible manufacturing system (FMS) that can produce one of $M$ possible products at any time $t$ ($t = 0, \ldots, T$). The payoffs $Z_i(t)$ ($i = 1, \ldots, M$) for the $M$ products follow $M$ independent Brownian motions with drift $\eta_i$ and volatility $\xi_i$:

$$Z_i(t + 1) = Z_i(t) \ast (1 + \eta_i) + \xi_i \ast \gamma,$$

with $\gamma$ being a normally distributed random variable with zero mean and unit variance. Given at any time $t$ a realization of the random variables $Z_i(t)$ and the mode of operation $m(t - 1)$ of the previous time, the task is to find the optimal mode $m(t)$ in order to maximize the discounted sum of profits. Without switching costs the solution of the optimization problem is simply given by
producing at any time $t$ the product with the highest payoff. However, there is a cost of switching between two modes $j, k$, denoted by $c_{jk}$, with $c_{ii} = 0 \ \forall i$. In [Kulatilaka (1995a)], the SDP has been formulated for the case of a one-dimensional exogenous uncertainty in terms of backward recursion using Bellman’s equation of dynamic programming. The generalization to $M$ stochastic processes yields:

$$F(T, m, \tilde{Z}) = \max_i Z_i(T) - c_{mi}$$

$$F(T - 1, m, \tilde{Z}) = \max_i Z_i(T - 1) - c_{mi} + \rho E_{T-1}[F(T, l, \tilde{Z})]$$

$$F(t, m, \tilde{Z}) = \max_i Z_i(t) - c_{mi} + \rho E_t[F(t + 1, l, \tilde{Z})]$$

$$OV = F(0, 0, \tilde{Z}(0)) = \max_i Z_i(0) - c_{0i} + \rho E_0[F(1, l, \tilde{Z})]$$

Starting at the terminal time $T$, we compute at each time both the value of the project and the optimal mode of operation given the current payoffs and the previous operation mode. Finally, at time $t = 0$ the option value of the project, OV, is determined.

The computation of the expected value $E_t[\cdot]$ depending on the state at time $t$ is done numerically. The random variables $Z_i(t)$ are discretized into 101 bins in the range: $(\mu(T) - 2.5\sqrt{\sigma(T)}/(1 + \eta_T^{-1}) \leq Z_i(t) \leq (\mu(T) + 2.5\sqrt{\sigma(T)})/(1 + \eta_T^{-1})$. The probability that the random variables are drawn from these intervals is larger than 0.9996. Special care has to be taken with the drift terms $1 + \xi_i$. As indicated in the formula for the discretisation, the ranges of the random variables have to be rescaled with respect to the drift terms at every time step. The discretized expression for the expected value depending on the $M$ state variables $(i_1, \frac{\Delta Z}{1 + \eta_1}, \ldots, i_M, \frac{\Delta Z}{1 + \eta_M})$, the previous mode of operation $l$ and time $t$ can be written as:

$$E_l[F(t + 1, l, i_1, \frac{\Delta Z}{1 + \eta_1}, \ldots, i_M, \frac{\Delta Z}{1 + \eta_M})] =$$

$$= \sum_{j_1, \ldots, j_M = -\infty}^{\infty} F(t + 1, j_1 \Delta Z) \times P_{i_1 - j_1} \ldots P_{i_M - j_M}$$

The transition probabilities are given by:

$$P_{i_k - j_k} = \frac{\Delta z(j_k + 1/2)}{\Delta z(j_k - 1/2)}$$

$$\varphi_{i_k \Delta z, \sigma_i(t)}(x_k)$$

2.2 Approximation via $\alpha$-rule

Although theoretically, SDP is the appropriate solution to our model, in practice it is impossible to evaluate the integrals numerically if there is a large number of modes $M$. Furthermore, often the underlying stochastic processes are not known
exactly but there is just a time series of realizations available. For this reason, we were looking for a numerically simple decision rule that can be evaluated directly from a set of realizations of the stochastic process.

The simplest decision rule one can think of is to choose at any time \( t \) the mode of operation with the highest profit. If switching costs are high and volatilities are low, this rule will yield poor results, because the switching costs are not taken into account. To overcome this problem, one could extend the simple rule by choosing the mode with the highest profit minus the switching costs. However, in the case of low volatilities and high switching costs the probability of changing the mode of operation will be too low.

From these heuristic arguments we derived a simple rule which we refer to as \( \alpha \)-rule. At time \( t \), we choose the mode that maximizes the expression

\[
\max_t Z_t(t) - \alpha c_{mt},
\]

where \( \alpha \) is a free parameter between zero and one. Given a time series of the underlying process, only a single parameter must be optimized. This rule estimates an optimal penalty factor for switching production modes dependent on the volatility of the underlying processes and switching costs.

For a configuration with \( M \) products and a planning horizon of \( T \) periods, application of the \( \alpha \)-rule requires \( MT \) additions and \( T \) calculations of the maximum of the \( M \) payoffs. Assuming that an \( if \)-statement needs the same amount of time as an addition-operation, \( 2MT \) operations per configuration are necessary. Therefore, the computational costs only grow linearly with \( T \) and \( M \). To learn the \( \alpha \)-rule, we discretize the interval of \( \alpha = \{0,1\} \) with \( N_\alpha \) (typically 100) values and evaluate the payoffs for the \( N_{\text{config}} \) (=50) configurations; i.e., in total, \( 2MTN_\alpha N_{\text{config}} \) operations are required.

### 2.3 Neural Networks Approximation

Neural Networks can be used to approximate the value function of the stochastic dynamic program described above. Based on the payoffs of the different modes and the switching costs given, a feedforward Neural Network model (for an introduction into Neural Network models, see e.g. Ripley (1993)) can be applied to learn to decide which of the given modes it should select (or activate) to maximize the expected value of the sum of the discounted payoffs over all periods.

In a learning (estimation) phase, the network weights (parameters) are determined on the basis of a large number of different payoff time series for the different modes (e.g., payoffs of different types of products which can be produced by the FMS) to maximize the sum of the discounted payoffs. In this phase, the network is trained to map the inputs through a feedforward network onto the outputs in such a way that the output unit (every output unit representing a mode) receiving maximum activation makes the highest contribution to the sum of all discounted payoffs.

In the validation phase, the rules learnt by the network model are applied to a sample of new payoff series of the different modes, and the expected value is calculated. For every mode (product) a separate network model is defined consisting of the following inputs: (1) the payoffs, \( Z_m(t) \), of all modes \( m \) in period \( t \), (2) the period \( t \), (3) the mode of the previous period \( t - 1 \), and a
constant (bias). Information about the status of the modes are dummy-coded.

Each of the networks has one output unit representing the profitability of their
"activation" given the inputs and network weights. The "softmax" operation
applied over all output units determines the production mode.

We start with a 2-layer feedforward Neural Network with logistic activation
functions because of its property as a universal approximator ([White (1989)]).
The number of hidden units is then varied between zero (reducing the system to
a logistic model without hidden units) and ten. With an increasing number of
hidden units, more complex functions can be mapped from inputs to outputs.
To avoid overfitting of the network models, we apply the models to a validation
sample and choose the network architecture based on this sample.

Network weights should be set in such a way that they produce optimal
strategies for any series of payoffs generated. Here, backpropagation cannot
be used because the correct outputs are not known a priori, i.e., we are faced
with an unsupervised learning problem. In our approach, we use Simulated
Annealing ([Kirkpatrick et al. (1983), Aarts and Korst (1989)]), a global opti-
mization method with a stochastic element, to estimate the network weights.
The starting value for the control parameter (temperature) of Simulated An-
nealing was chosen (by trial and error) to be 200, and after every iteration the
control parameter was multiplied by $0.99994$ until the system stabilized.

3 Application to Capital Budgeting for FMS

Typically, the introduction of an FMS into industry must be done on the basis
of cost justification ([Lint (1992)]. [Mitchell (1990)] has pointed out that there
is an adverse impact of short-term financial perspectives on technical strategy
and that the creation of future flexibility can be valued rather by option analysis
than by classical capital budgeting methods.

For the presentation of our methodology, we choose a very simple setting
considering only one type of flexibility ("switch option"). In the following, we
apply the methods discussed in the previous section to determine the production
program of an FMS for different switching costs, volatilities and payoff series.
An FMS which can be used to produce 3 different types of products (modes)
is evaluated. Only one product can be produced at a given time. Depending
on the payoffs $Z_i(t)$ of the 3 products $i = 1, \ldots, 3$, the different methods have
to select the mode of production in each period $t$. By "mode of production",
we denote the type of product that is produced by the FMS. The payoff of a
product is given by its sales price minus its production costs.

Using a Monte Carlo Simulation, payoffs are maximized for 50 different
payoff series generated by function 1 and validated on 10000 new series generated
by the same function. This holds for all methods, except for the SDP. The
decision about which product will be manufactured in a given period is based
on the payoffs generated in the case of its production. The expected value of
the discounted payoffs calculated over all different series represents the value of
the investment (FMS) when production modes are chosen by using the optimal
(stochastic DP) or the approximate rules ($\alpha$-rule, Neural Network). The sum
of the maximized discounted payoffs over all planning periods may be used as a
limit for the maximum price of the FMS. In our study, we apply an interest rate
of 10% per year. The lifetime of the FMS is assumed to be 50 periods (months
Table 1: Parameters for generating payoffs

<table>
<thead>
<tr>
<th>Product</th>
<th>( Z_i(0) )</th>
<th>( \eta_i )</th>
<th>( \xi_{i,A} )</th>
<th>( \xi_{i,B} )</th>
<th>( \xi_{i,C} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>0.02</td>
<td>20</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>0.015</td>
<td>30</td>
<td>60</td>
<td>150</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>0.01</td>
<td>40</td>
<td>80</td>
<td>200</td>
</tr>
</tbody>
</table>

\( t = 0, \ldots, 49 \). The valuation of the FMS is performed for different volatilities \( \xi_i \) of the payoffs with starting payoffs \( Z_i(0) \) and trend parameters \( \eta_i \) as given in Table 1. Table 2 shows switching costs \( c_{ij} \) which have to be considered when in period \( t \) the FMS is used to produce a different product than in the previous period.

Table 2: Switching costs

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_{0,1} )</td>
<td>27</td>
<td>135</td>
<td>270</td>
<td>640</td>
</tr>
<tr>
<td>( c_{0,2} )</td>
<td>13</td>
<td>65</td>
<td>130</td>
<td>260</td>
</tr>
<tr>
<td>( c_{0,3} )</td>
<td>18</td>
<td>90</td>
<td>180</td>
<td>360</td>
</tr>
<tr>
<td>( c_{1,2} )</td>
<td>24</td>
<td>120</td>
<td>240</td>
<td>480</td>
</tr>
<tr>
<td>( c_{1,3} )</td>
<td>16</td>
<td>80</td>
<td>160</td>
<td>320</td>
</tr>
<tr>
<td>( c_{2,1} )</td>
<td>25</td>
<td>125</td>
<td>250</td>
<td>500</td>
</tr>
<tr>
<td>( c_{2,2} )</td>
<td>29</td>
<td>145</td>
<td>290</td>
<td>580</td>
</tr>
<tr>
<td>( c_{3,1} )</td>
<td>27</td>
<td>135</td>
<td>270</td>
<td>540</td>
</tr>
<tr>
<td>( c_{3,2} )</td>
<td>35</td>
<td>195</td>
<td>390</td>
<td>780</td>
</tr>
</tbody>
</table>

**Simulation Results:** Figure 2 compares the results of the discounted payoffs of four different methods for solving the maximization problem. The stochastic dynamic program is the optimal solution to the problem. In the case of up to three different products it can be computed numerically and serves as a benchmark for the approximate methods. By way of comparison, we also calculated the expected value of the discounted payoffs for the case that in each period we chose the operation mode yielding the highest payoff of that period regardless of switching costs. This strategy is referred to as naive rule.

Figure 2 presents different levels of switching costs on the x-axes and average discounted payoffs on the y-axes. Each of the 3 subfigures illustrates the results for different volatility levels. The figure indicates that the payoffs increase considerably with higher volatilities and decrease with higher switching costs. The subfigure at the top also includes a line (the lowest of the 4 lines) depicting the payoffs of a production system that only makes the product with the highest expected value (i.e., the unflexible manufacturing system). Of course, these expected payoffs are independent of the volatilities and switching costs. While with low switching costs all methods lead to similar results, it can be seen that the naive rule clearly takes less advantage of flexibility in the case of higher switching costs than the other approaches investigated.

Within a deviation of 1% the simple \( \alpha \) rule yields the same results as the stochastic dynamic program for all volatilities and switching costs chosen. The network without hidden units turned out to generalize best. In the worst case of highest switching costs and lowest volatility, the deviation with respect
Figure 1: Expected values of the discounted payoffs for low (top), middle (middle) and high (bottom) volatilities for four different switching costs as a function of parameter $\alpha$. 

---

**Note:**

- **Volatility = 1:**
  - Cost 5
  - Cost 10
  - Cost 20
  - Cost 40

- **Volatility = 2:**
  - Cost 5
  - Cost 10
  - Cost 20
  - Cost 40

- **Volatility = 5:**
  - Cost 5
  - Cost 10
  - Cost 20
  - Cost 40
Figure 2: Comparison between the stochastic dynamic program (fat dashed), the network solution (dashed), the solution using the $\alpha$ rule (dash-dotted) and the naive rule (dotted) as a function of the switching costs. The horizontal line displays the discounted payoff of an unflexible manufacturing system.
to the SDP is 4 %. The discounted payoffs computed via the naive rule drop drastically as a function of the switching costs. For a switching cost value of 40 and volatility 1 the payoff equals that of an unflexible manufacturing system (indicated by a straight line at the top of Figure 2). Therefore, the performance of the decision rule chosen is essential for computing the value of flexibility.

The relative time requirements (on a DEC-Alpha) for this analysis are 8 hours for the stochastic dynamic program, 2 hours for the Neural Network model and 2 minutes for the \( \alpha \)-rule. Taking into account the above computing times and the performance in terms of deviations from the optimum, the \( \alpha \)-rule seems recommendable for more complex practical problems where the stochastic dynamic program cannot be applied.

Figure 1 shows the expected values of the discounted payoffs averaged over 10000 independent configurations for the \( \alpha \) rule as a function of parameter \( \alpha \). When switching costs are low, the discounted payoffs hardly depend on the choice of \( \alpha \). Hence, the results hardly depend on the switching rule applied to maximize the discounted payoffs. With increasing switching costs, however, the payoff functions become more and more convex, and an optimal \( \alpha \) can be identified for a given volatility and specified switching costs. In particular, for high switching costs and low volatilities the payoff function has the sharpest maximum. There, the difference between the payoff for the optimal \( \alpha \) and for \( \alpha = 0 \) is approximately a factor of three.

4 Conclusion

The traditional investment analysis leads to wrong results, if the investment object can adapt to a changing environment. In our model, the option value of flexibility, which is neglected in NPV analysis, is up to 15 times as high as for an unflexible production system. In this contribution, we computed the option value of a flexible manufacturing system by means of different solution techniques. The traditional approach is represented by stochastic dynamic programming (SDP), which is characterized by long computing times and its inherent complexity of modelling the underlying stochastic process. Therefore, the applicability of SDP is limited to the evaluation of an FMS that can produce three types of products. We use SDP as a benchmark for the approximate techniques. The simulation results for both the Neural Network and the \( \alpha \)-rule perform considerably well (in the worst case - high switching costs and low volatility - the deviation from the SDP is 4 % for the Neural Network and 1 % for the \( \alpha \)-rule, respectively).

As a pilot application, we evaluated an FMS with the option to change the production mode (3 different products). Changing the production mode means causing switching costs. It was shown that the value of the switching option can be very high - especially in case of high volatility - and that the rate of volatility had a higher influence on the value of the FMS than the switching costs. Thus, the choice of appropriate decision rules is of high relevance for low volatilities and high switching costs. The results obtained indicate that — as compared to a production system which is restricted to only one mode of production — the option of flexibility (as given with an FMS) reduces the risk of obtaining very poor results.
References


