International Institute for
Applied Systems Analysis
IIASA www.iiasa.ac.at

A Learning-by-Doing Energy Model on Dynamic Programming

Tze-Chao Ng, C.

IIASA Interim Report
November 2003

International Institute for
Applied Systems Analysis
Schlossplatz 1
A-2361 Laxenburg, Austria

Tel: +43 2236 807 342
Fax: +43 2236 71313
E-mail: publications@iiasa.ac.at
Web: www.iiasa.ac.at

Interim Report **IR-03-048**

A Learning-by-Doing Energy Model Based on Dynamic Programming

Charles Tze-Chao Ng (tzechao@stanfordalumni.org)

Approved by

Leo Schrattenholzer (leo@iiasa.ac.at)
Environmentally Compatible Energy Strategies Project (ECS)

November 3, 2003

Contents

Abstract

The concept of learning by doing (LBD) rests on the assumption that the more we do something, the more efficient we become at it. The inclusion of this phenomenon in our models results in a non-convex formulation and the possibility of multiple local optimal solutions. In this paper, we present a dynamic programming formulation of a model with learning-by-doing. The main advantage of this formulation is the guarantee of a global optimal solution, as conventional nonlinear solvers generally return local optimal solutions with no guarantee of global optimality. We also present two nonlinear extensions to the model that are not easily solved with some other heuristics. We conclude by running the model based on three carbon tax cases and a discussion of the results.

Acknowledgments

I would like to thank all the members of the ECS department at IIASA for their help and support. I would especially like to acknowledge my supervisor, Leonardo Barreto, for his constructive comments and insights.

I am particularly grateful to my thesis advisor, Prof. Alan Manne at Stanford University, for his invaluable guidance.

Finally, thanks to all my fellow YSSPers for an unforgettable summer.

About the Author

Charles Tze-Chao Ng is currently a PhD candidate in the Management Science and Engineering Department at Stanford University. He is working on solving non-convex learning-by-doing models using dynamic programming.

A learning-by-doing energy model based on dynamic programming

Charles Tze-Chao Ng

1 Introduction

The concept of learning by doing (LBD) rests on the assumption that the more we do something, the more efficient we become at it. This phenomenon can be seen in many aspects of everyday life. Typing skills, for example, can not be acquired without doing. We can memorize the placement of every key on the keyboard, but in end, it takes hours and hours of actual practice to type with a reasonable degree of accuracy and speed. Similarly, no one has ever learned to swim without entering the water, nor has anyone ever learned to ride a bicycle without the bike. Finding our way to a new location for the very first time almost always takes more time than subsequent trips, no matter how many times we look at the map beforehand. These are but a few examples where implementation is not only an important, but absolutely essential part, of the learning process.

In recent years, there has been a trend towards including the LBD phenomenon into energy models. From the energy modeling point of view, efficiency shows up as savings in cost of production. The concept of costs going down with cumulative production, or experience, was first brought up by Wright (1936) in his study of costs in airframe manufacturing. He observed that the direct labor hours needed to build an airframe decreased as more airframes were produced.

These feedback effects from learning are generally nonlinear. In Wright's airframe study, the number of direct labor hours required decreased by a fixed percentage with every doubling of cumulative experience. In other words, it is easy to become moderately knowledgeable in an area, but to get from mediocrity to perfection takes a lot more effort. The first few units of experience will contribute greatly to learning, but as we gain more experience, it becomes more and more difficult to extract lessons from implementation procedures. In many models, we preserve Wright's method of modeling learning, and assume that learning costs will go down with every doubling of experience. In the beginning, when experience is low, it does not take much effort to double experience, and the learning process is fast. As we accumulate experience, however, it becomes more and more difficult to double our experience, and the learning process slows down.

1.1 Non-convexity

This nonlinear behavior of cost with respect to experience results in a non-convex formulation of the model and the possibility of multiple local optimal solutions when we try to solve the model. For a minimization problem, a ***local optimal solution***, loosely defined, is a point where the objective function at that point is lower than the objective function for the points in its immediate vicinity. A ***global optimal solution***, on the other hand, is defined as a point where the objective function is the lowest among all feasible points.

For an analogy in three-dimensional space, we can think of solving a model as finding the lowest point on an uneven surface. If we pour water onto this surface, it will flow in a downwards direction, i.e. where the objective function is decreasing. A local optimal solution is a point where water accumulates. If the surface is convex, water will always flow towards the lowest point on the surface – the global optimal solution. For non-convex surfaces, however, it is possible that the water will accumulate in places other than the global optimum, like a lake on a mountaintop. Conventional nonlinear solvers generally use techniques that can be likened to water flowing on a surface, and they return local optimal solutions with no guarantee of global optimality.

In the right context, local optimal solutions represent different trajectories that the energy system may follow if decisions are made myopically. Different trajectories often lead to different technological configurations of the energy system with different environmental impacts. Sometimes, system costs can be very similar for different locally optimal configurations; at other times, a myopic policy could result in a much higher system cost than the global optimal configuration. Therefore, although local optimal solutions are interesting in their own right, it is important to have a global optimal solution in which to compare them against. In this paper, we focus on finding the global optimal solution.

Because of the nature of learning, LBD technologies are prone to being “locked-out”. We can think of this as a result of a myopic strategy – because an LBD technology is relatively expensive at the beginning, we do not use it; because we do not use it, its cost does not go down. This becomes a vicious cycle, and eventually, we do not use it altogether. If a technology is not locked out, there is also a question of “when” it comes in. In a locally optimal solution, a technology might be introduced either later or earlier than its optimal debut time. Empirically, this is a rarer occurrence than the lockout effect, but the modeler should still be aware of this possibility.

In this paper, we present a dynamic programming formulation of a model with learning-by-doing. The main advantage of this formulation is the guarantee of a global optimal solution. We first give an overview of the literature in Section 2, focusing on the various heuristics that have been used to solve LBD models. In Section 3, we describe our model and the dynamic programming formulation in detail. In Section 4, we present two nonlinear extensions to the model that are not easily solved with some other heuristics, in particular, the MIP approach described in Section 2. In Section 5, we run the model based on three carbon tax cases and discuss the results. Conclusions and directions for further work are presented in Section 6.

2 An Overview of the Literature

As mentioned in the previous section, the concept of LBD itself goes all the way back to Wright’s work (1936) on airframe manufacturing. Arrow (1962) was one of the first to apply learning curves in economics, suggesting an “endogenous theory of the changes in knowledge which underlie intertemporal and international shift in production functions.” Since then, many studies have been done on the incorporation of LBD into economic models.

In the realm of energy modeling, recent work include papers by Barreto and Kypreos (1999); Kypreos *et al*. (2000); Gritsevskyi and Nakićenović (2000); Manne and Barreto (2002).

Several heuristics have been used to overcome the problem of local optimality, with varying degrees of success. Mattsson and Wene (1997) use a “multiple starting point” approach that provides a “best known solution” instead of a global optimum solution.

Messner (1997) and Kypreos *et al*. (2000) both approximate the learning-by-doing cost curve with a piecewise linear one, and solve the problem with mixed integer programming (MIP) solvers. This method does guarantee a global optimal solution, but does not work readily when we have other nonlinear components in the model.

Manne and Barreto (2002) use a simple heuristic that forces the introduction of a technology by the end of the learning horizon. This approach eliminates local optima due to initial “lockout effects”, but does not deal with local optimality that might occur elsewhere. They also bring up the possibility of using the global optimization solver called BARON. This approach calls for much larger commitments in computing time and memory, and is at this point impractical for large-scale models.

3 Description of the Model

In this section, we will give a brief description of our model, and the formulation as a dynamic program. We have a system with a single region and a number of energy producing technologies. The objective is to satisfy all exogenous electric energy demands at the cheapest cost. There are growth and decline constraints on the technologies. A carbon tax is imposed on the amount of carbon produced.

3.1 Cost Function

Let $X_{j,t}$ be the level of production of technology j in time period t. The unit cost of production using technology j at time t is a function of experience $Y_{j,t}$.

$$unit\_cost_{j,t}(Y_{j,t}) = sc_j + incl_j \; \frac{Y_{j,t}}{acc_j}^{-lrn} \qquad (1)$$

Experience $Y_{j,t}$ is defined by the following equation 2.

$$Y_{j,t} = acc_j + \sum_{\tau=1}^{t-1} X_{j,\tau} \tag{2}$$

The parameter *sc* represents the portion of the unit cost that is static and not affected by the learning phenomenon. The parameter *incl* represents the initial portion of the unit cost that is affected by learning. The learning parameter *lrn* is greater than zero. Thus, as experience $Y_{j,t}$ increases, the learning portion of the unit cost decreases. For non-learning technologies, the parameter *incl* is zero. The parameter *acc* represents initial experience and the learning exponent *lrn* determines how fast costs decrease with time.

To calculate the learning exponent for a technology with learning costs, we assume that learning costs go down by a fixed percentage x with every doubling of experience. We then solve for *lrn* in the following equation.

$$2^{-lrn} = 1 - x$$

When x is 20%, the learning exponent is 0.32.

3.2 Dynamic Programming

As mentioned in Section 1, global optimality is not guaranteed when conventional nonlinear solvers are used. By formulating and solving this problem as a dynamic program, we are assured of a global optimum.

Dynamic programming is an extensive field. A thorough and detailed explanation of the theory and applications of dynamic programming can be found in Bertsekas (2000). In the following paragraphs, we provide a rough description with an emphasis on how it applies to our model.

A dynamic program is defined by the following components – ***state variables***, ***action variables*** and a ***recursive formula*** with terminal conditions. State variables describe the state of the world at a certain time period, or the information that affects our decisions. In our case, experience is a state variable because it denotes the unit cost of an LBD technology. Also, previous production is a state variable because it defines the critical levels for expansion and decline.

Action variables are the choices available to the decision maker. In our model, the action variables are the production levels for all available technologies. Different actions may result in different states. Clearly, different levels of production will result in different levels of experience in the next time period.

The recursive formula provides the connection between one time period and the next. The main idea behind a dynamic program is to first find the optimal actions for each possible state at the last time period T – the terminal stage. We then store the information in a ***cost-to-go function*** and pass it down to the decision maker at time period T-1. Using the recursive formula and the cost-to-go function, we can again find the optimal action for each possible state at time period T-1. This is again passed down to time period T-2 in the form of a cost-to-go function, and the recursion continues until we get to the first time period.

At the first time period, we know our initial conditions and we can easily find the optimal production portfolio by doing a forward recursion. That is, starting from the first period onwards.

3.3 Equations

We now present the dynamic program in mathematical terms. Let TECH be the set of technologies in the system. We define the sets X_{t-1} and Y_t as state variables, where

$$X_{t-1} = \{X_{j,t-1} \mid j \in TECH\}$$

and

$$Y_t = \{Y_{j,t} \mid j \in TECH\}\,.$$

X_{t-1} represents the levels of production in the previous time period for all technologies, and Y_t represents the experience levels.

The set $Y_t + X_t$ represents the updated experience due to production X_t and is defined as

$$Y_t + X_t = \{Y_{j,t} + X_{j,t} \mid Y_{j,t} \in Y_t, X_{j,t} \in X_t, j \in TECH\}\,.$$

Let the cost-to-go function

$$COST_t(X_{t-1}, Y_t)$$

be the minimum present value of all costs from time period t to the time horizon T, given the levels of the state variables X_{t-1} and Y_t. This is a continuous function over the state variables X_{t-1} and Y_t and we approximate it by calculating the values of the function at a pre-determined number of (X_{t-1},Y_t) grid-points.

The rest of the function is approximated via linear interpolation. MATLAB (2003) is used because it has a built-in interpolation function called *interpn*. This function does not require that the grid-points be evenly spaced. The main advantage of this approximation over a straightforward discretization is that it allows us more control over the resolution of the solution and the size of the problem.

The calculation of the cost-to-go function at the final time period T is given below. This is also known as a terminal condition because it does not depend on the cost-to-go function from a later time period.

$$COST_T(X_{T-1}, Y_T) = \min_{X_T \in A(X_{T-1})} \sum_{j \in TECH} unit\_cost(Y_{j,T}) \cdot X_{j,T} \tag{3}$$

The feasible region resulting from expansion and decline constraints is denoted by A(˙) and is known from the problem specification.

When we are in a time period t < T, the present value of total cost is the sum of two parts – the cost of production in this time period, and the discounted total cost of future time periods. A decision at this time period affects not only accumulated experience, but also upper and lower bounds for production in the next period due to expansion and decline constraints.

When t < T, the recursive formula for COST is given in 4) below.

$$COST_t(X_{t-1}, \mathrm{Y}_t) = \min_{X_T \in A(X_{t-1})} \{ \sum_{j \in TECH} \text{unit\_cost}(Y_{j,T}) \cdot X_{j,T} \\ + \beta \cdot COST_{t+1}(X_t, \mathrm{Y}_\mathrm{t} + X_t)\} \tag{4}$$

where β denotes the discount factor.

For readability, the equations 3 and 4 assume no carbon taxes. The modifications required to include carbon taxes are straightforward and will not be covered explicitly in this paper.

3.4 Complexity

The running time of a dynamic program is non-polynomial. It grows exponentially with respect to the number of state variables in the model. It also depends on the resolution used. The more grid-points we calculate, the closer our linear approximation will be to the true cost-to-go function. However, since these calculations take up the bulk of the running time, running time increases with the number of grid-points we calculate.

Besides time complexity, we are also interested in the space complexity of the algorithm – the amount of computer memory it takes to run the model. The main memory requirement is the storage of the cost-to-go function. If we had used a straightforward discretization of the cost-to-go function, the amount of memory required would have depended on the parameters. With the linear interpolation approximation, the amount of memory required is proportional to the number of grid-points we choose to approximate the cost-to-go function.

4 Extensions

With this dynamic programming formulation, we are able to incorporate two different extensions into the model. The extensions described below both require nonlinear terms, and are not easily modeled using an MIP approach.

4.1 Expansion and Decline Penalties

Instead of “hard bounds” for expansion and decline constraints, we impose a nonlinear penalty for large growth and large contractions in production. This extension serves two purposes. First of all, it gives a more subtle representation of growth and decline dynamics. Secondly, if we use “hard bounds” with the linear interpolation approximation, some state variable combinations will be infeasible and have a cost of infinity. Because of the sudden jump from a real number to infinity, linear interpolation is not a good approximation for the cost-to-go function when the states are close to the boundary of the feasible region. With nonlinear penalties for growth and decline and allowing all production levels to be feasible, we remove the problem of having to interpolate between real numbers and infinity.

When production in one period does not deviate too much from previous production, the penalty is zero. But when the deviation becomes too large, the penalty grows exponentially. The penalty function that we use is given in equation (5).

$$penalty(X_t, X_{t-1}) = \begin{cases} b\left(e^{d_t(X_{t-1})-X_t} - 1\right) & if\ X_t < d_t(X_{t-1}) \\ 0 & if\ d_t(X_{t-1}) \le X_t \le c_t(X_{t-1}) \\ a\left(e^{X_t - c_t(X_{t-1})} - 1\right) & if\ X_t > c_t(X_{t-1}) \end{cases} \qquad (5)$$

The critical levels after which the penalties kick in are $c_t(X_{t-1})$ and $d_t(X_{t-1})$. These depend on the level of previous production X_{t-1}. The parameters a and b determine how fast the penalties increase once we pass the critical level.

4.2 Cost Range Estimates

We can also use range estimates for unit cost instead of the point estimates described in the previous sections.

The concept of a single parameter for unit cost is a simplification. We live in a non-homogeneous world, and variation in costs is common. In their paper, Strubegger and Reitgruber (1995) give an interesting discussion on cost distributions of investment costs for various technologies.

Petersik (1999) provides an example of cost differentiation in US wind power generation. In the Northern states, wind plants require reinforcement to cope with the winter storms. In Texas, violent wind storms and tornadoes are common. The terrain in New England and the Northwest is steep and heavily vegetated. In the Midwest, wind turbine blades are often coated with insects. All these factors result in variations in production costs.

By using range estimates, we attempt to capture not only the variation in production costs due to geographical, political or cultural reasons, but also the optimization behavior that occurs with this variation. Imagine a cinema with a small number of people. Almost all of the people will be seated in the middle, since that is where the best view of the screen is. As more people enter, they will start sitting in the more unattractive seats, since the good ones will have already been taken. We assume that the decision maker is rational like the people in a cinema, and production always starts from the cheapest unit available.

With range estimates, we assume that different potential sites have different unit costs of production for the same technology. These differences could result from many different factors, ranging from geographical to political to cultural. Given the minimum and maximum values of the range of unit costs, we assume that potential production follows a uniform distribution with respect to cost. In other words, we have the same number of potential locations for each cost category between the maximum and minimum values. Figure 1 shows the uniform distribution of potential sites with respect to cost. The area under the rectangle is 1.

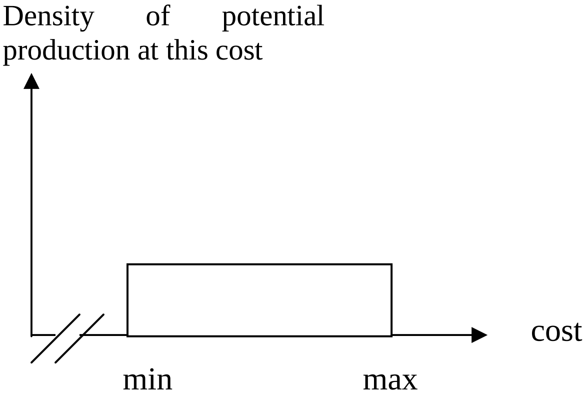

Figure 1: Uniform distribution of potential production over cost.

Integrating the function in Figure 1 gives us something conceptually equivalent to a cumulative distribution function in probability. Figure 2 gives the percentage of potential production below a given cost.

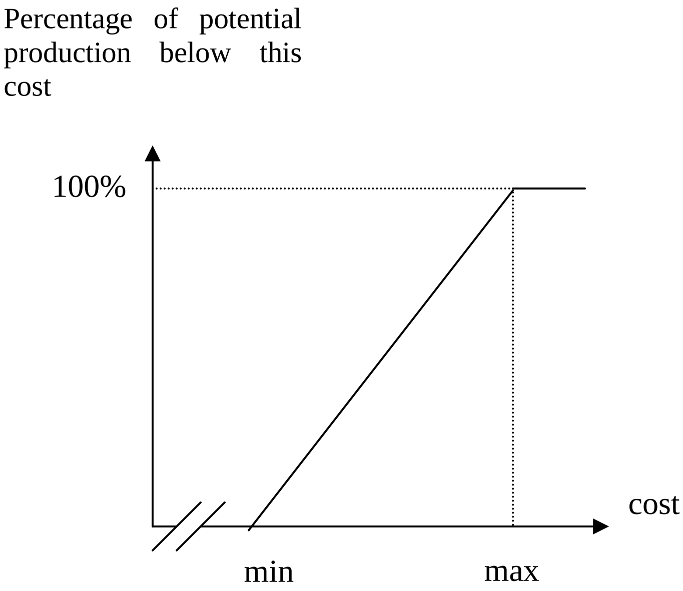

Figure 2: Percentage of potential production below a given cost.

Taking the inverse of the cumulative function gives the cost of the nth cheapest percentile of potential production. This is shown in Figure 3. We also show how the unit cost is divided into ranged cost, learning cost and static cost.

If we produce at x% of total potential production, the total cost of production is the area under the curve from 0 to x in Figure 3. With the uniform assumption for cost, this area can be calculated geometrically. If a different assumption on the cost distribution is made, total cost can still be calculated via integration.

It is also possible to assume a cost distribution over absolute production instead of a percentage of potential production. The discussion will be similar to the one presented in this section with slight modifications.

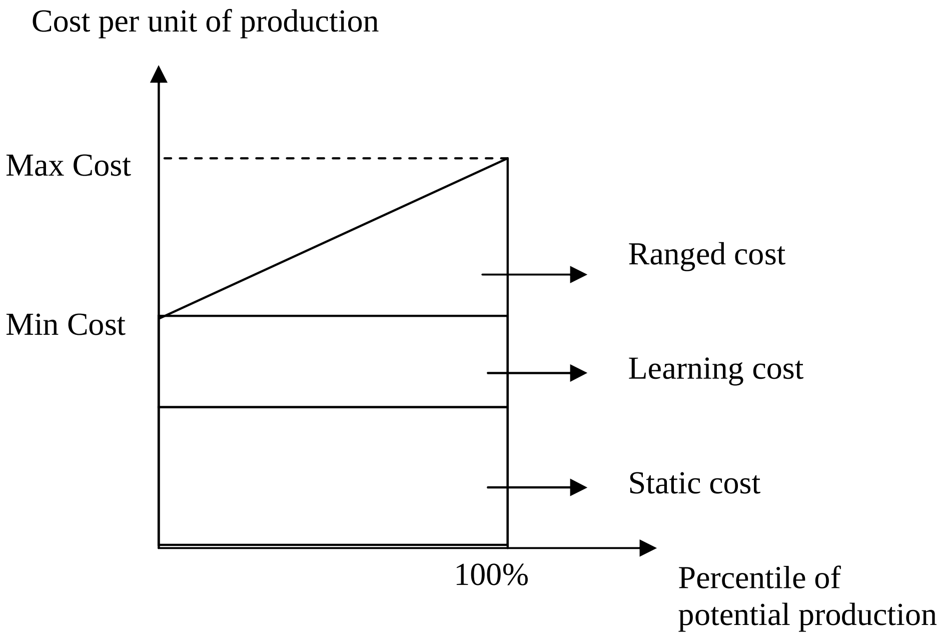
Cost per unit of production
Max Cost
Min Cost
Ranged cost
Learning cost
Static cost
100%
Percentile of potential production

Figure 3: Unit cost as a function of production.

5 Numerical results

5.1 Parameters

We use a time horizon of 2100, with ten years per time period. A discount rate of 5% is used.

For world electric energy demands in 2000, we use statistics from the International Energy Outlook 2003 published by the Energy Information Administration (2003). Predictions for 2010 and 2020 are also taken from the IEO 2003. For demands after 2020, we extrapolate by assuming a fixed percentage annual growth rate of 1.8%.

We consider three aggregate technologies in our model. The "mature" technology is low cost, with relatively high carbon emissions. The unit cost of production with the "mature" technology is low. The "med" technology has slightly higher initial costs than the "mature" one. It has a moderate amount of experience, but cost reductions are still possible through learning. It also produces carbon emissions, albeit with a much lower carbon coefficient. The variation in unit costs is moderate. The "new" technology has higher initial unit production costs, higher cost variations, and a higher learning potential. The "new" technology is carbon free. In year 2000, the "mature" technology satisfies 75% of all electricity demands, while the "med" and "new" technologies take up 20 and 5% respectively. Learning exponents of 0.32 are used for both learning technologies.

Figure 4 shows how the unit cost for each technology is divided into its individual ranged, learning and static parts.

For all technologies, we assume that the critical levels for both decline and growth are 5% a year. Growth or contraction exceeding 5% a year will be penalized according to the penalty function given in the previous section. The penalty parameters a and b are chosen so that the penalty is US$100 billion if we exceed expansion and decline critical levels by 10% of previous production.

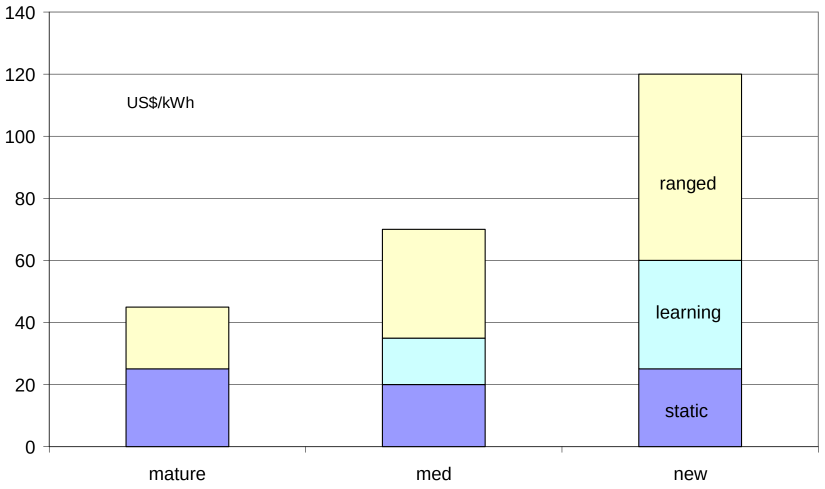

Figure 4: Unit cost for each technology for 2000, in US$/kWh.

We consider three cases with different levels of carbon tax. These cases are a subset of the cases studied at the EMF19 forum (2001).

- BAU – a business as usual case with no carbon tax
- LTAX – a +10 US$/ton per decade carbon tax increase case which starts with a 10 US$/metric ton carbon tax in 2010, which increases by US$10 per decade from then on.
- HTAX – a + 25 US$/ton per decade carbon tax increase case which starts with a 25 US$/metric ton carbon tax in 2010, increasing by 25 US$/metric ton per decade until 2040, and then held at the 100 US$/metric ton level through the end of the century.

A grid size of 10 x 10 x 10 x 10 is used for the approximation of the cost-to-go function, and the model is implemented in MATLAB. The bulk of the running time is devoted to the calculation of the cost-to-go functions in each time period. A single run of the model takes approximately one hour on a 1.8 GHz machine with 256 MB of RAM.

5.2 Analysis

In Figure 5, we give the globally optimal production paths of the three technologies for the BAU case. Because of the overlapping cost ranges and expansion/decline penalties, all three technologies are present in the optimal portfolio. The "mature" technology, however, maintains cost superiority for the first half of the century. As we enter the second half of the century, unit cost for the "med" technology becomes competitive, and the "med" technology starts to come in. Although the "new" technology follows a "minimal" production path, it still accumulates enough experience for its total unit cost to go down by roughly 36% by the year 2100. Even with these cost reductions due to learning, the "new" technology is still too expensive to be widely used, and it does not play a major role in fulfilling electricity demands over the course of the century.

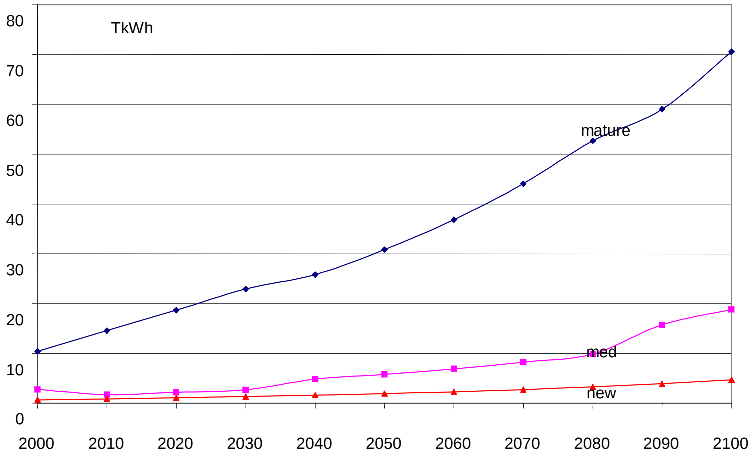

Figure 5: Optimal production portfolio for BAU case, in TkWh.

When carbon taxes are imposed, we have an incentive to steer away from the carbon-intensive "mature" technology. Figure 6a shows the optimal production trajectories when we impose carbon taxes according to the LTAX case. Since the "med" technology has a lower carbon coefficient than the "mature" one, it becomes more attractive and overtakes the "mature" technology at the end of the century. The "new" technology also plays a larger role, especially in the final two decades, when carbon taxes are US$90 and US$100 a ton respectively.

Figure 6b shows the optimal production trajectories when we impose carbon taxes according to the HTAX case. Because of the higher carbon tax, the carbon-free "new" technology has higher market penetration, supplying up to 20% of the demands in the last few decades. In order to achieve this, expansion for the "new" technology starts earlier in the century compared to the LTAX case.

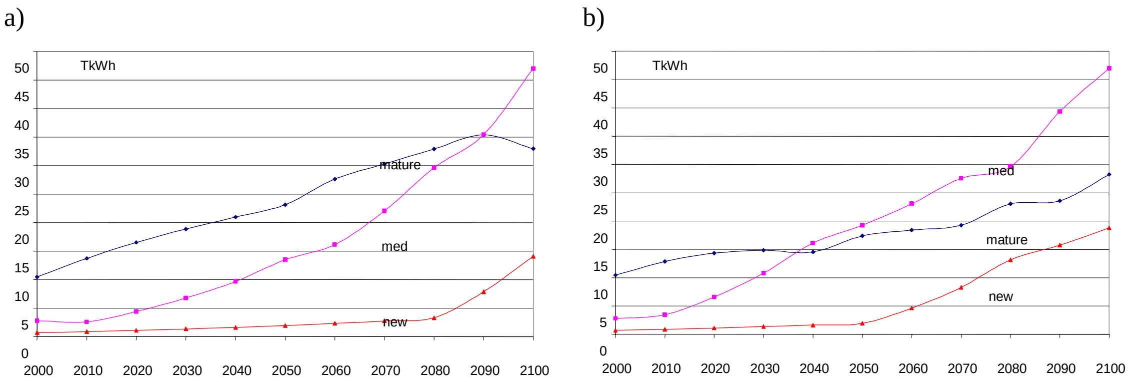

Figure 6: Optimal production portfolio for a) LTAX case, and b) HTAX case, in TkWh.

Figure 7 shows the amount of experience accumulated for the “new” technology over the century and Figure 8 shows the corresponding unit costs. We see that the unit costs at the end of the century for the “new” technology are slightly different, but not significantly, with 37.6 US$/kWh for the LTAX case versus 35 US$/kWh for the HTAX case. The reason for this small difference in cost is due to the fact that the “new” technology already satisfies 5% of the demand in year 2000. If we just maintain this minimal market share throughout the entire century, as in the BAU case, it is still possible to double experience four times over by the time horizon. If carbon taxes are imposed, experience accumulates faster, but the cost reductions from the extra experience are relatively small.

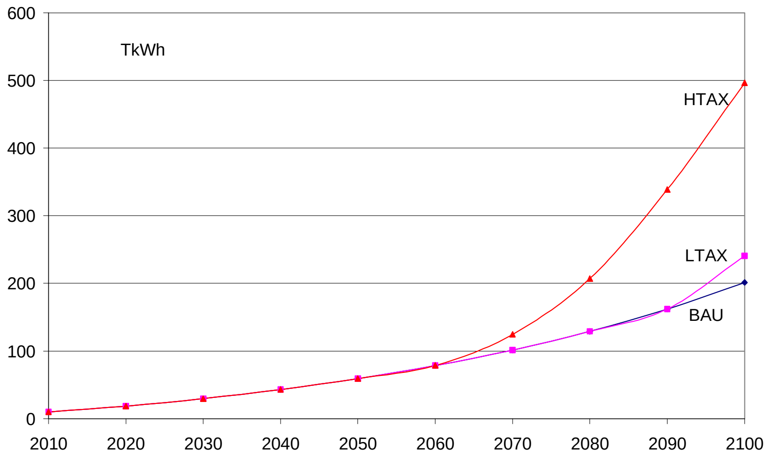

Figure 7: Experience accumulated for “new” technology, in TkWh.

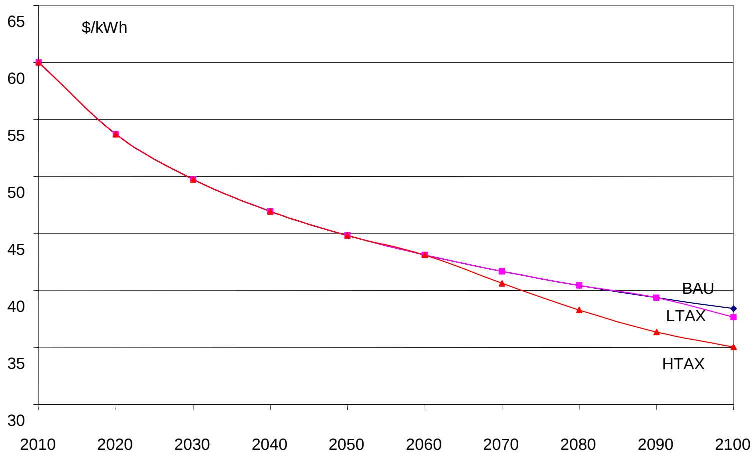

Figure 8: Unit cost for the cheapest unit of the “new” technology, in US$/kWh.

Thus, most of the difference in the “new” technology market penetration behavior results from the carbon tax with some contribution from the difference in unit costs due to learning.

Figure 9 shows the carbon emissions for each case. In all three cases, carbon emissions maintain an upwards trend, although they do level off at the time horizon for the LTAX case. This trend is due to the increasing demand and the fact that the carbon-free technology is not yet cost-effective. With other cost and demand parameters, it is possible that carbon emissions could decline over time. It is also clear from the figure that scenarios with higher carbon taxes result in lower carbon emissions.

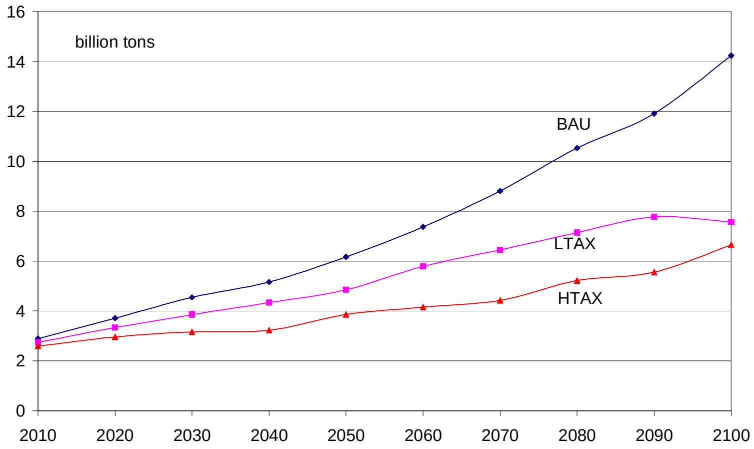

Figure 9: Carbon emissions under different tax scenarios, in billon tons.

6 Conclusions and Further Work

We have shown how dynamic programming can be used to solve nonlinear, non-convex models that include learning-by-doing. By using dynamic programming, we are guaranteed a global optimal solution. In addition, we have been able to incorporate features that cannot be easily handled by other approaches.

The running time of a dynamic program is non-polynomial and grows exponentially with the number of state variables. Since each new technology requires two new state variables, dynamic programming is currently not suitable for models with a large number of technologies. Thus, finding heuristics or algorithms that will reduce the running time of the dynamic program is an important direction for future research.

6.1 Parallel Processing

An interesting and promising direction for further work is the use of parallel processing. With the linear approximation of the cost-to-go function, we calculate the value of the function at a predetermined number of grid points. These calculations only depend on the cost-to-go function of the next time period, and are independent of one another. Therefore, it is theoretically possible to do these calculations on multiple processors simultaneously. We conjecture that since the bulk of the running time is due to the calculation of the cost-to-go function, the running time will be inversely proportional to the number of processors used as long as the number of processors does not exceed the number of predetermined grid points we calculate. If number of processors exceeds the number of predetermined grid points, the extra processors will not contribute towards decreasing running time.

6.2 Other Extensions

Another interesting extension to the model will be to have the variation of costs go down with learning. Mature technologies tend to have a smaller variation in unit costs than newer technologies, and it would be reasonable to assume that the variation will go down as we learn more.

Assuming that the some of the technologies in the model are based on the consumption of finite resources like oil or coal), we can also assume that the costs of these technologies go up with experience as these resources become scarce. This assumption should result in a shift away from these technologies as time goes by.

6.3 Conclusions

As computer processing power becomes increasingly available, we believe that the dynamic programming approach is a promising one. We would like to conclude with the observation that the dynamic programming approach itself exhibits learning by doing behavior. With the accumulation of experience, the running time of the algorithm has the potential to be reduced even further. But this potential, like the learning potential of our technologies, can not be realized unless we “do”.

References

Arrow, K.J. 1962. The Economic Implications of Learning by Doing, *Review of Economic Studies*, 29:155-173.

Beecy, D., Kydes, A, Richels, R, and Weyant, J. 2001. A Report on the EMF 19 Study on Technology and Global Climate Change Policies. Paper presented to the First National Conference on Carbon Sequestration, National Energy Technology Laboratory, U.S. Department of Energy. Available: http://www.netl.doe.gov/publications/proceedings/01/carbon_seq/2c1.pdf. [20 August 2003]

Bertsekas, D.P. 2000. *Dynamic Programming and Optimal Control, Vol I,* Athena Scientific.

Barreto, L., and Kypreos, S. 1999. Technological Learning in Energy Models: Experience and Scenario Analysis with MARKAL and the ERIS Model Prototype, PSI Bericht Nr. 99-08. Paul Scherrer Institute, Villigen, Switzerland. Sep, 1999.

Gritsevskyi, A., and Nakicenovic, N. 2000. Modeling Uncertainty of Induced Technological Change, *Energy Policy*. Vol. 28. pp 907-921.

Energy Information Administration 2003. *International Energy Outlook, 2003*, Washington, DC.

Kypreos, S. 2000. The MERGE Model with Endogenous Technological Change and the Climate Change, working paper, Paul Scherrer Institute. Presented at "Economic Modeling of Environmental Policy and Endogenous Technological Change Workshop", Amsterdam, November 16-17, 2000.

Kypreos, S., Barreto, L., Capros, P., and Messner, S. 2000 ERIS: A Model Prototype with Endogenous Technological Change, *Int. J. Of Global Energy Issues*. Vol. 14, 1-4. pp 374-397.

Manne, A.S. and Barreto, L. 2002. Learn-by-doing and Carbon Dioxide Abatement, presented at Energy Modeling Forum 19, February 12-14, 2002.

The MathWorks: Online Documentation, Release 13 2003. Available: http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml [20 August 2003].

Mattsson, N., and C-O. Wene 1997. Assessing New Energy Technologies using an Energy System Model with Endogenized Experience Curves, *International Journal of Energy Research.* Vol. 21, pp 385-393.

Messner, S. 1997. Endogenised Technological Learning in an Energy Systems Model. *Journal of Evolutionary Economics.* Vol. 7, pp 291-313.

Nash, S.G., and Sofer, A. 1996. *Linear and Nonlinear Programming*, McGraw-Hill.

Petersik, T. 1999. Modeling the Costs of U.S. Wind Supply. *Issues in Midterm Analysis and Forecasting*. pp 19 – 30.

Strubegger, M., and Reitgruber, I., 1995. Statistical Analysis of Investment Costs for Power Generation Technologies, WP-95-109. International Institute for Applied Systems Analysis, Laxenburg, Austria. November 1995.

Wright, T. 1936. Factors Affecting the Cost of Airplanes, *Journal of Aeronautical Science* Vol. 4 No. 4, pp 122-128.