Optimization of Medicare Advantage Healthcare Plans

Zachary Sunberg, Evan Mast

***Abstract*—This project focuses on the optimization of Medicare Advantage healthcare plans from an insurer's point of view. The insurance company seeks to maximize profits by increasing demand for the healthcare plans it offers. The problem is an infinite horizon dynamic programming problem on continuous state and action domains. Fitted value iteration and traditional value iteration with simple discretization are used to find approximate solutions. In this scenario, simple discretization outperforms the fitted value iteration approach.**

I. Introduction

HEALTHCARE is one of the largest sectors of the American economy, accounting for nearly 20% of the United States' gross domestic product [1]. Understanding of the healthcare market may lead to better regulation of the healthcare system and more efficient allocation of resources. One aspect of this market that economists are seeking to characterize is the behavior of healthcare insurance companies. Some recent working papers [2], [3] have commented on this question in a limited scope. In this project, we will use several approaches to determine healthcare insurance plans that are optimal from the insurance company's perspective in a regulated subsidized market. This may eventually lead to a better understanding of why insurance companies make decisions.

The insurance companies seek to maximize their profit over all time with future profits slightly discounted. This goal leads to an infinite-horizon discounted value planning problem (note that the word "plan" refers to a healthcare insurance plan, while "planning" refers to determining future actions). Profits are increased when plans with a higher demand are offered, so plans are chosen according to a demand model. The method for solving such a planning problem is known as Dynamic Programming and was popularized by Bellman as early as the 1950s [6]. It is based on the Principle of Optimality which states that (informally) every tail-end subsequence of an optimal action sequence is also optimal. In the infinite-horizon case, this means that if the optimal cost to go from every state can be estimated, then a greedy choice to take the action that results in moving to the state with the lowest cost to go (of the states that can be reached with the available actions) is optimal. Dynamic Programming is used widely in Economics research [4], [5].

Z. Sunberg is a graduate student in the Department of Aeronautics and Astronautics, Stanford University. E. Mast is graduate student in the Department of Economics, Stanford University.

The most basic method for realizing this solution approach is known as value iteration. Each step of this value iteration involves solving an optimization subproblem. Value iteration is a very straightforward algorithm to implement for discrete state and action spaces, and is guaranteed to converge under mild conditions [7], but it becomes more complicated in continuous domains. The purpose of undertaking this particular project is not to immediately make a new contribution to economics or advance the state of the art of optimization, but rather to apply existing techniques to a problem for the sake of learning. That being said, the techniques explored in this project may be very important in Medicare Advantage research that will be conducted in the near future.

II. Medicare Advantage

The Affordable Care Act aims to create a competitive market for healthcare coverage that will be significantly subsidized by the government. Another program that has been in place since 1997, known as Medicare Advantage (MA), has some similarities with the new system. MA was designed to provide an alternative to traditional Medicare (TM) that would allow private insurance providers to offer improved care at lower cost. Within MA, private insurers offer bids to provide Medicare-like coverage to consumers in a geographic region. The Centers for Medicare and Medicaid Services (CMS) provides detailed data on current and past plans which provides a rich set of material for research.

The basic structure of a bid can be seen in Figure 1. The government initially sets a benchmark approximating the expected cost per patient for TM ($700 in the figure). Insurance companies submit bids in the amount that they will use to provide TM-equivalent coverage ($600 in the figure) and extract profits. Of the gap between the bid and the benchmark, 75% is given to the insurer as a rebate, and 25% is returned to the government. The rebate, however, is restricted to only be used for improvements to coverage. It cannot be retained as profit. For our investigation, we have simplified the choices available to the insurance providers. The distribution of the rebate in each plan will be referred to as θ. Under our model, they can use the rebate to 1) increase cost-sharing contributions (c_θ), 2) reduce Part B (outpatient care) premiums (b_θ), 3) reduce Part D (prescription drug coverage) premiums (d_θ), and 4) provide other services (s_θ). When represented as a vector, θ is ordered as follows:

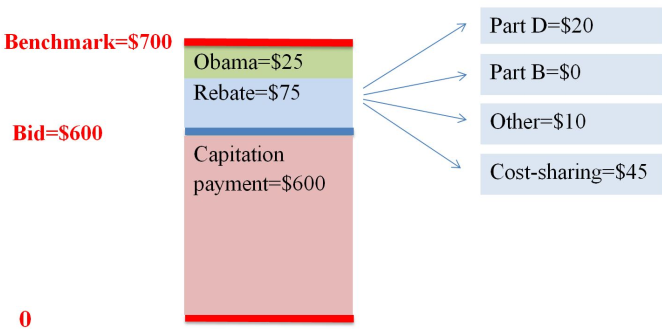

Fig. 1. Breakup of a Medicare Advantage Plan Bid

$$\theta = [c_\theta, b_\theta, d_\theta, s_\theta]$$

Since the rebate money cannot be retained for profit, the distribution of the rebate can only be used to increase the desirability of the plan and thus the demand for it. Since profit only comes from the bid portion, the profit per person covered is directly proportional to the demand for the plan. This of course neglects second-order effects that the insurance companies might try to take advantage of, for example offering coverage that might be more attractive to healthier beneficiaries that will cost less.

One would expect different components of the plan to have different effects on the demand. For example, Part B premiums are taken from social security funding. Since beneficiaries don't have to deliberately write a check for this, it seems natural that this parameter would have a diminished effect on the demand. In contrast, Part D premiums are paid directly by the consumer, so high demand for increased Part D contributions is expected. Indeed the data seems to support some of these hypotheses. For example Part D premium contributions have risen or stayed constant through rebate reductions more consistently than any of the other categories in recent years. This can be clearly seen in Figure 2

III. Problem Formulation

The problem of finding the optimal plan will be formulated and solved as a DP problem.

A. Dynamic Programming Problem

The planning problem that we seek to solve requires the definition of several parameters:

- The **state** of the system at time stage t is the current configuration of the world as it relates to the problem. It lies within the state space $\mathcal{S}$. In this case, the state consists of the current plan that is in place, θ_{t-1}, and the number of beneficiaries on that plan, NB_{t-1}.

$$x_t = \{\theta_{t-1}, NB_{t-1}\}$$

- The **action** taken by the insurance company at time t is the plan to be offered in the upcoming time stage. It lies in the action space $\mathcal{A}$.

$$u_t = \theta_t$$

- The stagewise **utility** function maps a state and action to the corresponding immediate desirability of the state and action, that is $g : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Specifically in this case, the utility is the profit, which is proportional to the demand for the upcoming time stage.

$$g(x_t, u_t) = \pi D(\theta_{t-1}, NB_{t-1}, \theta_t)$$

where π is the profit per beneficiary.
- The **system dynamics** govern the transition of the state between stages. Abstractly, this is represented by

$$x_{t+1} = f(x_t, u_t)$$

In this specific problem, the control from the previous time stage makes up part of the new state, and the number of beneficiaries evolves according to the demand function.

$$NB_t = D(\theta_{t-1}, NB_{t-1}, \theta_t)$$

- The **discount factor**, β, is a number between 0 and 1 that determines the relative importance of the immediate utility and future utility of an action. $\beta = 1$ indicates that utility at any time is equally important. As β decreases, the future utility becomes less important.

Using these definitions, the problem can be stated as

$$\begin{aligned} \underset{u_t \in \mathcal{A}}{\text{maximize}} \quad & \sum_{t=0}^{\infty} \beta^t g(x_t, u_t) \\ \text{subject to} \quad & x_{t+1} = f(x_t, u_t);\ t = 0 \ldots \infty \end{aligned} \tag{1}$$

It should be noted that the class of problems to which this problem belongs is a subclass of the Markov Decision Process (MDP) class. This subclass has a deterministic state transition mapping while the general MDP problem class allows the use of probabilistic transitions. If noise is added to the dynamics, e.g.

$$NB_t = D(\theta_{t-1}, NB_{t-1}, \theta_t) + w_t \tag{2}$$

where w_t is a sample from a Gaussian white noise process, then the problem becomes a traditional MDP with probabilistic state transitions.

B. Demand Function Estimation

Since the profit per beneficiary is independent of the rebate distribution, the stagewise utility is simply the demand, D, generated by a new plan given the current state. This can be further broken up into two parts. D_{old} is the demand for the new plan generated by beneficiaries who were already covered by the previous plan. D_{new} is the demand generated by people

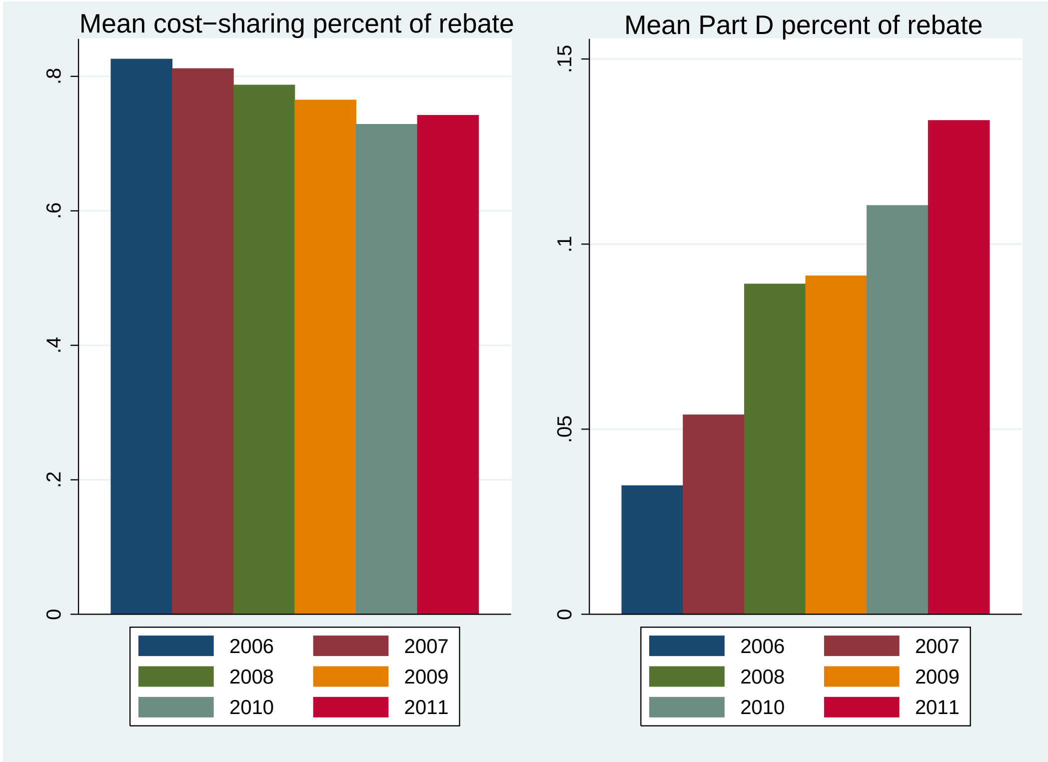

Fig. 2. Growth of Part D contributions in MA plans

not enrolled in the previous plan. The total plan is the sum of these two values.

$$D(\theta_{t-1}, NB_{t-1}, \theta_t) = D_{old}(\theta_{t-1}, NB_{t-1}, \theta_t) + D_{new}(NB_{t-1}, \theta_t) \quad (3)$$

Neither the structure nor the parameter values for these functions are well-known, so, for this project, we must pick some reasonable demand functions. These functions could be rigorously estimated by fitting them to the large amount of data that is available. One method of accomplishing this would be to use a genetic algorithm on a population of basis functions and parameter values. Unfortunately, the limit on the amount of time that I had to devote to this project prevented this from being realized.

Because of the limited scope of the project, demand functions were chosen based only on their *qualitative* resemblance to the actual data. D_{old} is simply a linear function of the new plan

$$\begin{aligned} D_{old}(\theta_{t-1}, NB_{t-1}, \theta_t) = \\ NB_{t-1}\big(0.75 + .15(d_{\theta_t} - d_{\theta_{t-1}}) \\ + .1(s_{\theta_t} - s_{\theta_{t-1}}) + .5(c_{\theta_t} - c_{\theta_{t-1}})_- \\ + .2(s_{\theta_t} - s_{\theta_{t-1}})_-\big) \end{aligned} \quad (4)$$

where $(\cdot)_-$ is the minimum of the quantity and zero, i.e. only a negative change has an effect. The most important property of this function is that it is concave with respect to θ_t, meaning that it can be easily maximized. One feature that is not well expressed with this function is consumer ”inertia”. If the plan stays exactly the same from year to year, it is natural to expect that a large number of beneficiaries would stick to the plan even if the terms are not otherwise attractive. For example the function

$$D'_{old} = \begin{cases} .95\, NB_{t-1} & \textbf{if } \theta_t = \theta_{t-1} \\ D_{old} & \textbf{else} \end{cases}$$

could be used to express this. This function is quasiconcave, so its use in this case is feasible, but the concave function in (4) was used for this project.

D_{new} is designed to express the concept of diminishing returns. Each factor of the plan has a weight, but the values are raised to the $1/4$ power, so that the utility diminishes.

$$\begin{aligned} D_{new}(NB_{t-1}, \theta_t) = (P - NB_{t-1}) \\ \left(.1 + .2c_{\theta_t}^{1/4} + .2d_{\theta_t}^{1/4} + .1s_{\theta_t}^{1/4}\right) \end{aligned} \quad (5)$$

This function is also concave with respect to θ_t facilitating rapid maximization.

In order to validate this demand function, the results of maximizing the demand of plans were qualitatively compared to actual data. Figure 3 shows the actual evolution of a plan over time. Figure 4 shows simulation data with the cost function given in (3) using the discretized solver. The actual number of dollars spent per beneficiary is simply a Gaussian random variable with a mean of \$125 and a standard deviation

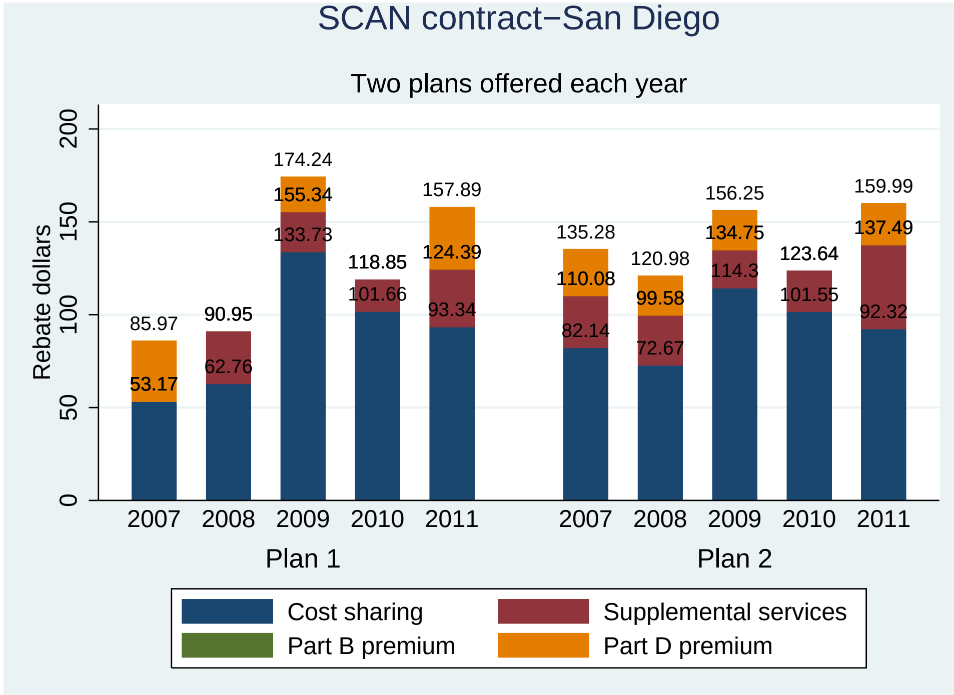

Fig. 3. Actual evolution of a Medicare advantage plan. The numbers are dollars spent per beneficiary.

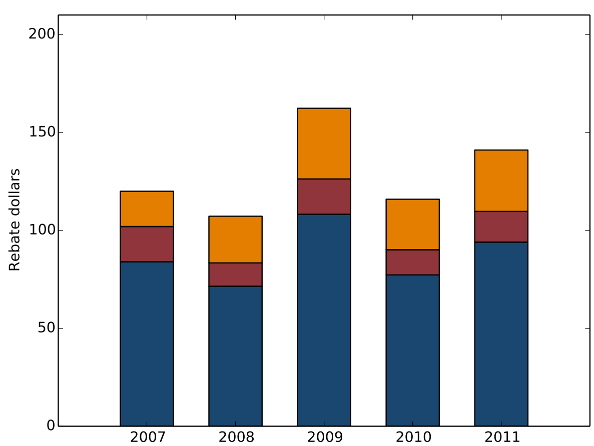

Fig. 4. Simulation results for comparison with Figure 3

of \$35. Qualitatively, these figures look similar.

There are a great deal of changes that could be made to the demand model to make it match the data. For example a discrete indicator could be applied so that the demand contribution of a component below a certain level is zero, e.g. if d_θ is less than 0.5, the contribution to demand is zero. This might draw out some behavior seen in the data like the lack of Part D premium contributions seen in 2010. Such changes would however make the demand function quasiconcave, meaning that a solution would be practical to obtain, but the time constraints on the project prevented it from being implemented.

IV. Solution Approaches

Solution of the infinite horizon dynamic programming problem (1) is accomplished through an algorithm known as value iteration. This method leads to a value function, $V : \mathcal{S} \rightarrow \mathbb{R}$ that is an estimate of the cost to go from any state. The value function corresponds to a policy $\mu : \mathcal{S} \rightarrow \mathcal{A}$ which optimally maps every state to an action according to V. The true value function is the solution to the fixed point equation

$$V^*(x) = g(x, \mu(x)) + \beta V^*(f(x, \mu(x)) \tag{6}$$

V^* can be estimated iteratively by repeatedly applying the Bellman operator T

$$\begin{aligned} V_{k+1}(x) &= \\ &T[V_k(x)] = \max_u \left(g(x,u) + \beta V_k(f(x,u))\right) \\ &\qquad \forall x \in \mathcal{S} \end{aligned} \tag{7}$$

Under mild conditions, it can be shown that the Bellman operator is a contraction and the fixed point equation (6) has a single optimal solution that Equation 7 will converge to[7]. The optimal policy is then given by

$$\mu(x) = \arg\max_u \left(g(x,u) + \beta V^*(f(x,u))\right) \tag{8}$$

Value iteration is trivial to implement for discrete state and action spaces. However (1) lies on a continuous domain. It is impossible to solve the general dynamic programming problem over continuous spaces because the state and/or control spaces contain an infinite number of members (although solutions to some special cases such as the LQG problem are known). Thus, approximate solution techniques must be used. Unfortunately, with the approximation, the strong convergence guarantees for value iteration are lost, although good performance is often obtained in practice. Three approximate techniques are introduced here, and two were implemented for testing as part of this project.

A. Simple Discretization

The simplest method for finding an approximate solution is to simply discretize the state and action spaces In this case, the value function returns the value calculated for the nearest member of the discrete state set. This is the most straightforward way of dealing with continuous domains, and limits on the inaccuracies produced by this discretization are discussed in [4] (although the results of that paper are not directly applicable to this work because the cost function is not differentiable). This approach was implemented in simulation using a set of 217 plans evenly sampled from the space of possible plans and 11 possible values of NB. The value iteration algorithm converged to within machine precision in 280 steps, although the level of convergence reached at 10-15 steps was usually used.

B. Interpolation

Interpolation is a more sophisticated approach to handle continuous problems. Instead of only calculating the value function for a discrete set of points, the value function is defined for the entire state space by using multilinear interpolation between the discrete points. The primary advantage of this is that actions not in the set of discrete actions are considered, which will likely lead to an improved control law when compared to the law derived using simple discretization as described above. Nonlinear spline interpolation may also be used, sometimes with significant speed improvements as discussed in [8]. Unfortunately, time did not allow for implementation of this approach. The difficulty does not lie in the interpolation, but rather in that a nonconvex continuous optimization problem must be solved each time the Bellman operator is used in (7) and optimal actions are determined in (8). A fast and reliable solver for this was too difficult to implement and debug in the time alloted to this project.

C. Fitted Value Iteration

Fitted value iteration is another method for approximating the value function over a continuous domain. In this approach, each time the Bellman operator is applied, a function is fit to a randomly selected set of states and the corresponding optimal actions with respect to the current value function. Any type of function fit can be used, but the simplest approach is to use linear regression. This is described in more detail in [9]. The value function is approximated as

$$V_k(x) = \alpha_k^T \phi(x) \tag{9}$$

where ϕ is a possibly nonlinear set of basis functions. α is chosen using least squares based on m randomly sampled points in the state space according to

$$\alpha_{k+1} = \arg\min_{\alpha} \sum_{i=1}^{m} \Big(\alpha^T \phi(x^{(i)}) - g(x^{(i)}, \mu(x^{(i)})) - \alpha_k^T \phi(f(x^{(i)}, \mu(x^{(i)}))) \Big)^2 \tag{10}$$

For this specific problem, several different functions were tested as ϕ. The one that seemed to work the best is

$$\phi(x) = [\theta, \sqrt{\theta}, NB, 1] \tag{11}$$

This includes linear and diminishing returns (square root) terms for each of the plan parameters, a linear term for the number of beneficiaries, and a constant. A very important feature of this set of functions is that it is concave, allowing for very fast maximization when calculating $\mu(x)$ (which must be done m times every time the Bellman operator is applied).

All of the simulation was written in using the Julia programming language, except for the convex optimization step of calculating μ. This was completed using cvxpy. Unfortunately, the overhead involved in translating the problem using cvxpy makes this method relatively slow, limiting the amount of experimentation that could be done, and preventing direct execution time comparison with the simple discretization approach. In a more advanced implementation, the convex optimization problem would already be encoded in a form that can be quickly solved instead of using the cvxpy translation at each step. Convergence of this algorithm to a reasonable value function estimate requires around 10 steps.

V. Results

In this section, the performance of the simple discretization and fitted value iteration methods are compared. Figure 5 shows the number of beneficiaries as a function of time for both of the implemented methods. The simple discretization method outperforms the fitted value iteration approach. Even though fitted value iteration considers a continuous set of states and actions rather than a finite discrete set when optimizing, the set of basis functions used for the fit was not rich enough to capture the important features of the value function.

The same trend can be seen in Figure 6, although it is not as pronounced. In this case, Gaussian noise with a standard deviation of 50 was added to the demand as described in (2). The plan breakdown history for the noisy case is shown in Figure 7 for fitted value iteration and in Figure 8 for simple discretization. There is very little change in the policy over time. In fact, the simple discretization method chooses the same policy at every time after the initial step. This is due to the fact that the demand and fitted value iteration basis functions are relatively simple.

VI. Conclusions

Two methods for solving an infinite horizon dynamic programming problem were tested on a Medicare Advantage healthcare plan optimization problem. Both approaches converged to a near-optimal solution. However, in terms of solution quality, the less complicated method, simple discretization, performed stronger. Simple discretization also seemed

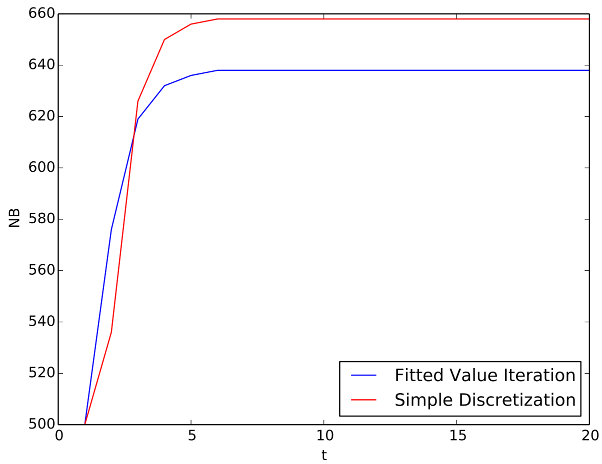

Fig. 5. Demand time history

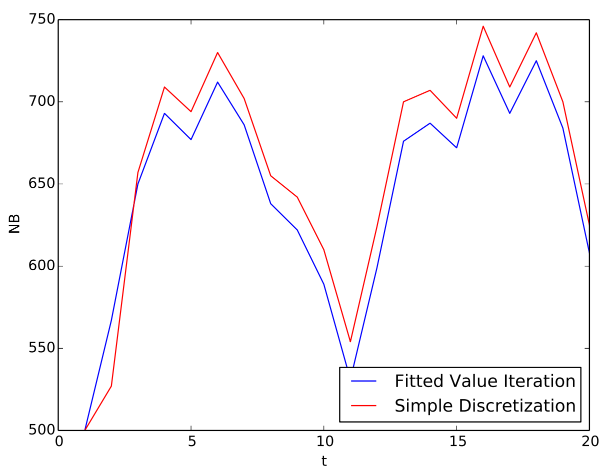

Fig. 6. Demand time history with noise added

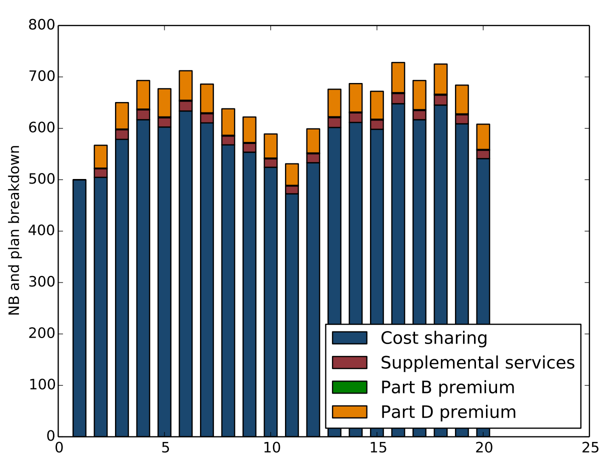

Fig. 7. Plan breakdown time history for the fitted value iteration approach

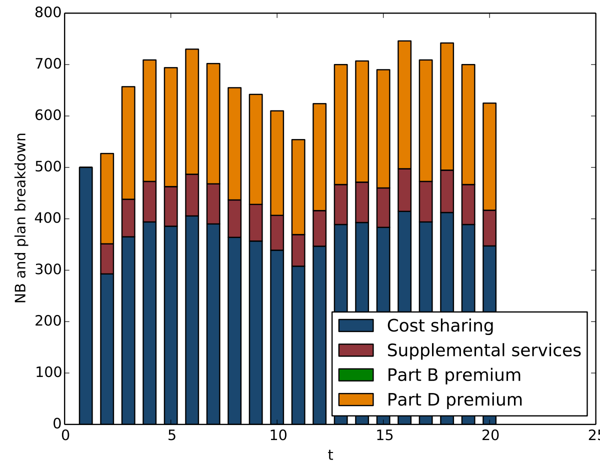

Fig. 8. Plan breakdown time history for the simple discretization approach

to use less computation, however a direct comparison is not possible because the fitted value iteration approach utilized a computationally inefficient package for translating (at every iteration) small inner optimization problems to a form that can be quickly solved.

The methods used in this project are not complicated, but developing the demand functions and getting the optimizations to work consistently did require a significant amount of work. The most interesting future extension of this short study would be to remove the restriction that the demand and basis functions be concave. This would allow for more complex and interesting demand behavior and basis functions that better characterize the true cost to go. Moreover, if these functions are quasiconcave, then they can still be solved very efficiently.

Work Distribution

The instructor requested that I specifically note the work breakup and specific contributions of the two authors. Evan essentially only provided the background for the project (and created Figures 1 and 3). I took the problem that he described, formulated in the style of [7], and proposed and implemented the solutions.

References

[1] "Health Expenditure". World Health Organization National Health Account database, The World Bank. 02 June 2014. http://data.worldbank.org/indicator/SH.XPD.TOTL.ZS .

[2] Duggan, M., Starc, A., and Vabson, B. "Who Benefits when the Government Pays More? Pass-Through in the Medicare Advantage Program". Unpublished working paper. 2014.

[3] Cabral, M., Geruso, N., and Mahoney, N. "Does Privatized Medicare Benefit Patients or Producers? Evidence from the Benefits Improvement and Protection Act". Unpublished working paper. 2014.

[4] Santos, M., and Vigo, J. "Error bounds for a numerical solution for dynamic economic models". *Applied Mathematics Letters*. 1996, 9(4), 41-45.

[5] Santos, M., and Vigo-Aguiar, J. "Analysis of a numerical dynamic programming algorithm applied to economic models". *Econometrica*. 1998, 66(2), 409-426.

[6] Bellman, R. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[7] Bertsekas, D. *Dynamic Programming and Optimal Control, Vols. 1 and 2*. Athena Scientific, Nashua, NH, 2005.

[8] Johnson, S., Stedinger, J. R., Shoemaker, C., Li, Y., and Tejada-Guibert, J. "Numerical Solution of Continuous-State Dynamic Programs Using Linear and Spline Interpolation". *Operations Research*. 1993, 41(3), 484500. doi:10.1287/opre.41.3.484

[9] Ng, Andrew. "Reinforcement Learning and Control". CS-229 course notes. Stanford University, 2013.