# Manuelli & Seshadri: Human Capital and the Wealth of Nations Computing the Household Problem

Econ821

Prof. Lutz Hendricks

April 21, 2016

# Job training problem

Write this as starting at $a = 0$ (shifting the age range)

$$V(h(0), T) = \max \int_0^T e^{-ra} \{wh(a)(1 - n(a)) - p_w x(a)\} da \quad (1)$$

$$\begin{aligned} \dot{h}(a) &= G^w (h(a), n(a), x(a)) - \delta_h h(a) & (2) \\ &= z_h (h(a) n(a))^{\gamma_1} x(a)^{\gamma_2} - \delta_h h(a) & (3) \end{aligned}$$

$$h(0) \text{ given} \quad (4)$$

# Hamiltonian

$$\Gamma = wh(1-n) - p_w x + q\left[G^w(h,n,x) - \delta_h h\right] \tag{5}$$

FOCs:

$$whn = q\gamma_1 G^w \tag{6}$$
$$p_w x = q\gamma_2 G^w \tag{7}$$
$$\dot{q} = rq - q\left\{\gamma_1 G^w/h - \delta_h\right\} - w(1-n) \tag{8}$$
$$q_T = 0 \tag{9}$$

Implied static condition

$$p_w x = whn\gamma_2/\gamma_1 \tag{10}$$

# Solution

Let $t$ be the remaining horizon.

Collect the equations that solve for $q(t), n(t)h(t), x(t), h(t)$.

$$q(t) = (1-\tau)w\frac{m(t)}{r+\delta_h} \tag{11}$$

$$n(t)h(t) = [Qm(t)]^{1/(1-\gamma)} \tag{12}$$

$$x(t) = n(t)h(t)\frac{w}{p_w}\frac{\gamma_2}{\gamma_1} \tag{13}$$

# Solution for $h(a)$

$$h(a) = e^{-\delta_h a} h(0) + C e^{-\delta_h a} \int_0^a e^{\delta_h t} m(T-t)^{\gamma/(1-\gamma)} dt \tag{14}$$

with

$$C = z_h Q^{\gamma/(1-\gamma)} \left( \frac{\gamma_2}{\gamma_1} \frac{w}{p_w} \right)^{\gamma_2} \tag{15}$$

Value function:

$$V(h,T) = q(T) h + w \frac{1-\gamma}{\gamma_1} Q^{1/(1-\gamma)} \int_0^T e^{-rt} m(T-t)^{1/(1-\gamma)} \tag{16}$$

# Useful helper objects

$$m(t) = 1 - \exp((r + \delta_h)t) \tag{17}$$

$$Q = \frac{z_h \gamma_1^{1-\gamma_2} \gamma_2^{\gamma_2}}{r + \delta_h} \left(\frac{w}{p_w}\right)^{\gamma_2} \tag{18}$$

A limitation: this does not handle corner solutions (one could extend it, of course).

# Other properties we need

For calibration, we need

1. age-wage profile: $wh(a)(1-n(a))-p_w x(a)$

For the schooling first-order condition, we need

1. $dV/dT$
2. $-rV+dV/dT$

For testing, we need

1. computing $h(a)$ for arbitrary paths of $n(a), x(a)$

# Organizing the code

All of this is completely self-contained, so we put it into a general purpose Matlab class: `BenPorathContTimeLH`.

The object holds all the parameter values: $T, r, \gamma_1, \gamma_2, ...$

We write a function for each equation (why not?)

Even though the equations are in terms of horizon remaining, it is more natural to return age profiles (e.g. $h(a)$).

# Computing $h(a)$ for given $n(a), x(a)$

We need to solve

$$h(a) = \int_0^a \left[G^w(h(t), n(t), x(t)) - \delta_h h(t)\right] dt \tag{19}$$

How do we do that?

# Testing the code

1. Check first-order conditions that define a solution.
2. Check that value is maximized by perturbing $n(a), x(a)$
3. Check value function against its definition
4. Check $dV/dT$
5. Check $q = dV/dh$

See `BenPorathContTimeTestLH`

# Schooling

# Schooling

The household solves

$$\max_{s,x(a),n(a),h(a),x_e,h_e} -p_e x_e + \int_0^s -p_s x_s(t)\,dt + e^{-rs} V(h_s, R - s - a_0) \quad (20)$$

subject to

- $h(a_0) = h_E = h_B x_E^{\nu}$
- $\dot{h}(a) = G^s(h(a), x(a)) - \delta_h h(a)$; $a \leq a_0 + s$
- $h_s = h(a_0 + s)$

Terminal value function from job training.

# Hamiltonian

$$\Gamma = -p_s x_s + q\left(G^s(h, x_s) - \delta_h h\right) \tag{21}$$

First order conditions (essentially the same as job training without the condition for $n$):

$$\begin{aligned} -p_s + q\partial G^s/\partial x_s &= 0 \quad (22) \\ \dot{q} &= (r+\delta_h)q - q\partial G^s/\partial h \quad (23) \\ q(s) &= \partial V/\partial h \quad (24) \end{aligned}$$

In addition: $x_e$ solves

$$V_0 = \max_{x_e} q_e h_B x_e^v - p_e x_e \tag{25}$$

where $q_e = q(a_0) = \partial W/\partial h$ from schooling problem.

# School solution

Solution is: $x_e, h_E, q_E, h_s, s, q_s, x_s$ that solve:

$q(s)\, h_s^{\beta_1} = q_E h_E^{\beta_1} e^{g(x_s)(1-\beta_2)s}$

$x(a) = (z_h \beta_2 / p_s)^{1/(1-\beta_2)} \left( q(a)\, h(a)^{\beta_1} \right)^{1/(1-\beta_2)}$ for $a = a_0 + s$

$q_e = q(a_0) = \partial W / \partial h$

First-order condition for $s$

From job training: $q(a_0 + s) = dV/dh$

From ChildCareMS: $h_e, x_e$ as a function of $q_e$

# How to solve this numerically?

We could search over the 6 objects, but that would be inefficient.

Easier:

- search over 2 objects (e.g. $s, qE$)
- get $x_e h_e$ from `ChildCareMS`
- get $q_s$ from `BenPorathContTimeMS`
- get $h_s$ from $q(s)\, h_s^{\beta_1} = q_E h_E^{\beta_1} e^{g(x_s)(1-\beta_2)s}$
- get $x_s$ from its FOC.

See method `solve`.

# Complication: corner at $s = 0$

We need to allow for the possibility that $s = 0$.

Steps:

1. solve problem for $s = 0$ (`solve_s0`)
2. check whether the marginal value of $s$ is negative
3. if so: $s = 0$ is optimal
4. if not: solve for interior $s$

So we need a method that computes the marginal value of $s$

- this is $\Gamma - rV + dV/ds$ (recall the FOC for $s$)

# School solution

Once we have these scalars, we get age profiles from:

$$q(a)\,h_a^{\beta_1} = q_E h_E^{\beta_1} e^{g(x_s)(1-\beta_2)(a-a_0)} \tag{26}$$

$$x(a) = (z_h \beta_2 / p_s)^{1/(1-\beta_2)} \left(q(a)\,h(a)^{\beta_1}\right)^{1/(1-\beta_2)} \tag{27}$$

We don't need the age profile for $h(a)$ (though we have a solution for that, too).

See method `age_profile`.

# Code organization

We want a self-contained object that "knows" its parameters.

So we make it a Matlab class: `SchoolProblemMS`.

A trick: we store pointers to the job training class and the child care class inside the object

Benefits:

1. Once the `SchoolProblemMS` object is constructed, its methods can be called without having to pass (job-training) parameters.
2. If we change the job training problem, we just need to pass the new job training class to the constructor. Nothing changes in the school problem code.

# Testing

1. check age profiles against first-order conditions
2. check initial and terminal conditions
3. perturb $s$ and check that value declines
4. check marginal value of $s$ by perturbation

See `SchoolProblemTestMS`