Two user multiple access broadcast

Aditya Mahajan

April 20, 2010

This is `maxima` source code to verify the calculations presented in the appendix of

Aditya Mahajan, "Optimal decentralized transmission policies for two-user multiple access broadcast", in Proceedings of the 2010 conference on decision and control (CDC).

Preliminaries

In Proposition 1, we define a transformation A_i. Since we are only concerned with symmetric arrival rates, we only need to work with

```
Ap(n) := 1 - (1-p)^(n+1);
```

In Definition 3, a function $f_n(x)$ is defined as:

```
f(n,x) := 1 + (1-x)^2 - (3+x)*(1-x)^(n+1) ;
```

s_n denotes the root of $f_n(x)$ that is between $[0, 1]$. Thus,

```
s(n) := find_root( f(n,x), x, 0, 1 )$
```

τ is the root of $x = (1-x)^2$ that lies in $[0, 1]$. Thus,

```
tau : find_root( (1-x)^2 = x, x, 0, 1 )$

display(tau)$
```

$$\tau = .3819660112501052$$

Dynamic program

Although the reachable state space is countable, the value function can be written succiently. We use different variable names for each form of the value function:

- $v(p, A^n p)$ = `v(n)`
- $v(p, 1)$ = `v_p1`
- $v(1, 1)$ = `v_11`
- $v(p, p)$ = `v_pp`

For reason that will become apparent later, we use `v_init` for the value of `v(1)`.

Similar to the value function, the differential reward function has four forms.

- $w_{01}(p, A^n p)$ = `w01(n)`
- $w_{01}(p, 1)$ = `w01_p1`

- $w_{01}(1,1) =$ `w01_11`
- $w_{01}(p,p) =$ `w01_pp`

These differential rewards are given by

```
w01(n) := r*Ap(n) + 'v_init    $
w01_p1 :  r       + 'v_init    $
w01_11 :  r       + 'v_p1      $
w01_pp :  r*p     + 'v_init    $
```

Similar interpretations hold for w_{10}

```
w10(n) := r*p + 'v(n+1)    $
w10_p1 :  r*p + 'v_init    $
w10_11 :  r   + 'v_p1      $
w10_pp :  r*p + 'v_init    $
```

and w_{11}

```
w11(n) := r*(p + Ap(n) - 2*p*Ap(n)) + p*Ap(n)*'v_11 +
          (1-p*Ap(n))*'v_pp                          $
w11_p1 :  r*(1+p-2*p) + p*'v_11 + (1-p)*'v_pp        $
w11_11 :  'v_11                                      $
w11_pp :  r*(2*p - 2*p^2) + p^2*'v_11 + (1-p^2)*'v_pp $
```

We need a few helper functions to display the results of intermediate calculations.

```
show(label, arr, size) := for i : 1 thru size do
    print(label, i, ":",
      arr[i], " = ", ev(arr[i], nouns, eval, eval, ratsimp)) $

show_diff(label, arr1, arr2, size) := for i : 1 thru size do
    print(label, i, ":",
      arr1[i],
      if ev(arr1[i] - arr2[i], nouns, eval, eval, ratsimp) = 0
        then "=" else error(arr1[i], "!=", arr2[i]) ,
       arr2[i])
$

array(check, 4) $
array(diff,  2) $
array(value, 2) $
```

Case 1 : $p \geq \tau$

```
print("Case 1: p ≥ τ")$ print("--------------") $
```

For this case, we find it more convinient to work with $1-p$ rather than p. So,

```
kill(p) $
p : 1-q $
```

We claim that

```
J    :  r*(1-q^2)       $
v(n) := r*(1-q^(n+1))   $
v_p1 :  r               $
v_11 :  r*(1+q^2)       $
v_pp :  r*p             $

v_init : v(1)           $

print("value functions")$ display(J, v(n), v_11, v_pp, v_p1)$
```

value functions

$$J = (1 - q^2)\, r$$
$$v(n) = (1 - q^{n+1})\, r$$
$$v_{11} = (q^2 + 1)\, r$$
$$v_{pp} = q^2\, r$$
$$v_{p1} = r$$

To verify the optimal policy, we need to check two things. First, we have to verify fixed point equations:

```
check[1] : '(v(n) + J - w01(n)) $
check[2] : '(v_p1 + J - w01_p1) $
check[3] : '(v_11 + J - w01_11) $
check[4] : '(v_pp + J - w01_pp) $

show("check", check,4)          $
```

Second, we have to verify that the chosen action gives a larger reward than other actions. We treat the four cases separately:

For $(\pi_1, \pi_2) = (p, A^n p)$, we have

```
diff[1] : '(w01(n) - w10(n)) $
diff[2] : '(w01(n) - w11(n)) $

value[1] : '(r*p*q*(1-q^n))            $
value[2] : '(r*p^2*(1+q*(1-(2+q)*q^n)))$

show_diff("diff(n)", diff, value, 2) $
```

For $(\pi_1, \pi_2) = (p, 1)$, we have

```
diff[1] : '(w01_p1 - w10_p1)  $
diff[2] : '(w01_p1 - w11_p1)  $

value[1] : r*q         $
value[2] : r*p^2*(1+q) $

show_diff("diff_p1", diff, value, 2) $
```

For $(\pi_1, \pi_2) = (1, 1)$, we have

```
diff[1] : '(w01_11 - w10_11)  $  value[1] : 0         $
diff[2] : '(w01_11 - w11_11)  $  value[2] : r*p*(1+q) $

show_diff("diff_11", diff, value, 2) $
```

For $(\pi_1, \pi_2) = (p, p)$, we have

```
diff[1] : '(w01_pp - w10_pp)  $
diff[2] : '(w01_pp - w11_pp)  $

value[1] : 0            $
value[2] : r*p^2*(p-q^2) $

show_diff("diff_pp", diff, value, 2) $
```

Case 2 : $s_1 \leq p < \tau$

```
print("Case 2: s_1 ≤ p < τ")$ print("------------------") $
```

As before, it is more convinient to work with $1-p$ rather than p. So,

```
kill(p) $
p : 1-q $
```

In this case, only the value function for $v(p,p)$ changes. The rest are the same as before

```
v_pp :  r*q^2 $

print("value functions")$ display(J, v(n), v_11, v_pp, v_p1)$
```

To verify the optimal policy, we need to check two things. First, we verify the fixed point equations. The first three equations remain as before, so we only modify the fourth check equation.

```
check[4] : '(v_pp + J - w11_pp) $

show("check", check,4)          $
```

Second, we have to verify that the chosen action gives a larger reward than other actions. We treat the four cases separately:

For $(\pi_1, \pi_2) = (p, A^n p)$, we have

```
diff[1] : '(w01(n) - w10(n)) $
diff[2] : '(w01(n) - w11(n)) $

value[1] : '(r*p*q*(1-q^n))  $
value[2] : r* '(f(n,p) - 3*q^2*(1-q^(n-1)))$

show_diff("diff(n)", diff, value, 2) $
```

For $(\pi_1, \pi_2) = (p, 1)$, we have

```
diff[1] : '(w01_p1 - w10_p1)  $
diff[2] : '(w01_p1 - w11_p1)  $
```

```
value[1] : r*q        $
value[2] : -r*f(0,q)  $

show_diff("diff_p1", diff, value, 2) $
```

For $(\pi_1, \pi_2) = (1, 1)$, we have

```
diff[1] : '(w01_11 - w10_11)  $
diff[2] : '(w01_11 - w11_11)  $

value[1] : 0         $
value[2] : r*p*(1+q) $

show_diff("diff_11", diff, value, 2) $
```

For $(\pi_1, \pi_2) = (p, p)$, we have

```
diff[1] : '(w11_pp - w10_pp)  $
diff[2] : '(w11_pp - w01_pp)  $

value[1] : r*(q^2 - p) $
value[2] : r*(q^2 - p) $

show_diff("diff_pp", diff, value, 2) $
```

Case 3 : $s_{m+1} \le p < s_m$

```
print("Case 3: s_(m+1) ≤ p < s_m")$ print("------------------") $
```

In this case, it is more convinient to work with p rather than $1-p$. So,

```
kill(p) $
q : 1-p $
```

In many ways, this is the most difficult case. Part of the difficulty arises from the fact that the form of the value functions are more complicated.

```
D : 1 + p^2 + p^3            $
J : r*p*(1-ratsimp(f(0,p))/D) $

v_p1 : J            $
v_11 : r            $
v_pp : r*f(1,p)/D   $

c_low(n)  := q*(1-q^n)*J/p + r*q^(n+1) - r*q + v_pp $
c_high(n) := r*(1-q^(n+1)) + c_low(1) - J           $

v_init : c_low(1) $
```

As before, to verify the optimal policy, we need to check two things. First, we verify the fixed point equations.

```
kill (check)  $ array(check, 5) $
```

```
check[1] : '(c_low(n) + J - w11(n))  $
check[2] : '(c_high(n) + J - w01(n)) $
check[3] : '(v_p1 + J - w01_p1) $
check[4] : '(v_11 + J - w01_11) $
check[5] : '(v_pp + J - w11_pp) $

show("check", check,5)          $
```

For $(\pi_1, \pi_2) = (p, A^n p)$, and $n \le m$, we have

```
v(n) := c_low(n)  $
diff[1] : '(w11(n) - w10(n)) $
diff[2] : '(w11(n) - w01(n)) $

value[1] : -r*p*(1-q^(n+1))*'f(0,p)/D   $
value[2] : -r*p^2*f(n,p)/D              $

show_diff("diff(n)", diff, value, 2) $
```

For $(\pi_1, \pi_2) = (p, A^n p)$, and $n > m$, we have

```
v(n) := c_high(n)  $
diff[1] : '(w01(n) - w10(n)) $
diff[2] : '(w01(n) - w11(n)) $

value[1] : r*p*(- f(0,p)/D - q^(n+1))   $
value[2] : r*p^2*f(n,p)/D               $

show_diff("diff(n)", diff, value, 2) $
```

For $(\pi_1, \pi_2) = (p, 1)$, we have

```
diff[1] : '(w01_p1 - w10_p1)  $
diff[2] : '(w01_p1 - w11_p1)  $

value[1] : r*q        $
value[2] : r*p^2*(1+q^2)/D    $

show_diff("diff_p1", diff, value, 2) $
```

For $(\pi_1, \pi_2) = (1, 1)$, we have

```
diff[1] : '(w01_11 - w10_11)  $
diff[2] : '(w01_11 - w11_11)  $

value[1] : 0                    $
value[2] : r*p*(1+p)*(1+q^2)/D $

show_diff("diff_11", diff, value, 2) $
```

For $(\pi_1, \pi_2) = (p, p)$, we have

```
diff[1] : '(w11_pp - w10_pp)  $
diff[2] : '(w11_pp - w01_pp)  $
```

```
value[1] : r*p^2*(1-2*p^2)/D  $
value[2] : r*p^2*(1-2*p^2)/D  $

show_diff("diff_pp", diff, value, 2) $
```

Output

```
Maxima 5.20.1 http://maxima.sourceforge.net
using Lisp SBCL 1.0.37
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1)                    batchload(value-function.mac)
                            tau = .3819660112501052
```

Case 1: $p \geq \tau$

```
--------------
value functions
                                        2
                              J = (1 - q ) r

                                        n + 1
                           v(n) = (1 - q      ) r

                                      2
                             v_11 = (q  + 1) r

                             v_pp = (1 - q) r

                                 v_p1 = r

check 1 : J - w01(n) + v(n)  =  0
check 2 : J - w01_p1 + v_p1  =  0
check 3 : J - w01_11 + v_11  =  0
check 4 : J - w01_pp + v_pp  =  0
                                          n
diff(n) 1 : w01(n) - w10(n) = p q (1 - q ) r
                               2          n
diff(n) 2 : w01(n) - w11(n) = p  (q (1 - q  (q + 2)) + 1) r
diff_p1 1 : w01_p1 - w10_p1 = q r
                                     2
diff_p1 2 : w01_p1 - w11_p1 = (1 - q)  (q + 1) r
diff_11 1 : w01_11 - w10_11 = 0
diff_11 2 : w01_11 - w11_11 = (1 - q) (q + 1) r
diff_pp 1 : w01_pp - w10_pp = 0
                                     2     2
diff_pp 2 : w01_pp - w11_pp = (1 - q)  (- q  - q + 1) r
```

Case 2: $s\_1 \leq p < \tau$

```
------------------
value functions
                                        2
                              J = (1 - q ) r
```

$$v(n) = (1 - q^{n + 1})\, r$$

$$v\_11 = (q^2 + 1)\, r$$

$$v\_pp = q^2\, r$$

$$v\_p1 = r$$

$$\text{check 1} : J - \text{w01}(n) + v(n) = 0$$
$$\text{check 2} : J - \text{w01\_p1} + \text{v\_p1} = 0$$
$$\text{check 3} : J - \text{w01\_11} + \text{v\_11} = 0$$
$$\text{check 4} : J - \text{w11\_pp} + \text{v\_pp} = 0$$

$$\text{diff}(n)\ 1 : \text{w01}(n) - \text{w10}(n) = p\, q\, (1 - q^n)\, r$$
$$\text{diff}(n)\ 2 : \text{w01}(n) - \text{w11}(n) = (f(n, p) - 3\, q^2\, (1 - q^{n - 1}))\, r$$
$$\text{diff\_p1 1} : \text{w01\_p1} - \text{w10\_p1} = q\, r$$
$$\text{diff\_p1 2} : \text{w01\_p1} - \text{w11\_p1} = - ((1 - q)^2 + (- q - 3)\, (1 - q) + 1)\, r$$
$$\text{diff\_11 1} : \text{w01\_11} - \text{w10\_11} = 0$$
$$\text{diff\_11 2} : \text{w01\_11} - \text{w11\_11} = (1 - q)\, (q + 1)\, r$$
$$\text{diff\_pp 1} : \text{w11\_pp} - \text{w10\_pp} = (q^2 + q - 1)\, r$$
$$\text{diff\_pp 2} : \text{w11\_pp} - \text{w01\_pp} = (q^2 + q - 1)\, r$$

Case 3: $s_{(m+1)} \leq p < s_m$

$$\text{check 1} : J - \text{w11}(n) + \text{c\_low}(n) = 0$$
$$\text{check 2} : J - \text{w01}(n) + \text{c\_high}(n) = 0$$
$$\text{check 3} : J - \text{w01\_p1} + \text{v\_p1} = 0$$
$$\text{check 4} : J - \text{w01\_11} + \text{v\_11} = 0$$
$$\text{check 5} : J - \text{w11\_pp} + \text{v\_pp} = 0$$

$$\text{diff}(n)\ 1 : \text{w11}(n) - \text{w10}(n) = - \frac{f(0, p)\, (1 - (1 - p)^{n + 1})\, p\, r}{p^3 + p^2 + 1}$$

$$\text{diff}(n)\ 2 : \text{w11}(n) - \text{w01}(n) = - \frac{((- p - 3)\, (1 - p)^{n + 1} + (1 - p)^2 + 1)\, p^2\, r}{p^3 + p^2 + 1}$$

$$- (1 - p)^2 - (- p - 3)\, (1 - p) - 1$$

$$\text{diff(n) 1 : w01(n) - w10(n)} = p\left(\frac{}{p^3 + p^2 + 1} - (1 - p)^{n + 1}\right) r$$

$$\text{diff(n) 2 : w01(n) - w11(n)} = \frac{((- p - 3)\,(1 - p)^{n + 1} + (1 - p)^2 + 1)\, p^2\, r}{p^3 + p^2 + 1}$$

$$\text{diff\_p1 1 : w01\_p1 - w10\_p1} = (1 - p)\, r$$

$$\text{diff\_p1 2 : w01\_p1 - w11\_p1} = \frac{((1 - p)^2 + 1)\, p^2\, r}{p^3 + p^2 + 1}$$

$$\text{diff\_11 1 : w01\_11 - w10\_11} = 0$$

$$\text{diff\_11 2 : w01\_11 - w11\_11} = \frac{((1 - p)^2 + 1)\, p\, (p + 1)\, r}{p^3 + p^2 + 1}$$

$$\text{diff\_pp 1 : w11\_pp - w10\_pp} = \frac{p^2\, (1 - 2 p^2)\, r}{p^3 + p^2 + 1}$$

$$\text{diff\_pp 2 : w11\_pp - w01\_pp} = \frac{p^2\, (1 - 2 p^2)\, r}{p^3 + p^2 + 1}$$

(%o1)   /home/adityam/Projects/queueing-mac/verify/value-function.mac