Tutorial 7: solution sketches

1. The Bellman optimality equations are as follows:

$$
\begin{aligned}
x_6 &= 1 \\
x_5 &= \max\{x_1, x_2\} \\
x_4 &= x_4 \\
x_3 &= \max\{x_2, x_4\} \\
x_2 &= 2x_1/5 + x_4/5 + 2x_6/5 \\
x_1 &= x_2/6 + x_4/6 + x_5/6 + x_6/2
\end{aligned}
$$

Our goal is to compute the unique minimal (least non-negative) solution vector $p^* = (p^*_1, \dots, p^*_6)$, which gives the optimal probabilities of reaching s_6 starting from each state. It is clear that $p^*_4 = 0$, so that at s_3 the node s_2 is always chosen, giving $p^*_3 = p^*_2$. Also, since $p^*_6 = 1$, it only remains to solve for p^*_1, p^*_2, and p^*_5. From the optimality conditions we see that the equations governing these are as follows:

$$
\begin{aligned}
p^*_5 &= \max\{p^*_1, p^*_2\} \\
p^*_2 &= 2p^*_1/5 + 2/5 \\
p^*_1 &= p^*_2/6 + p^*_5/6 + 1/2
\end{aligned}
$$

We can (as shown on the lecture slides) solve this by computing the unique optimal solution for the following linear programming problem:

Minimize: $x_1 + x_2 + x_5$

Subject to:

$x_5 \geq x_1$,

$x_5 \geq x_2$,

$x_2 \geq (2/5)x_1 + (2/5)$,

$x_1 \geq (1/6)x_2 + (1/6)x_5 + (1/2)$,

$x_1 \geq 0, x_2 \geq 0, x_5 \geq 0$.

In this small example, we can also avoid using linear programming, in a somewhat ad hoc way (which is NOT suitable for large models), by enumerating all possible cases of the max equations. There are two cases to consider: (i) $\max\{p_1^*, p_2^*\} = p_2^*$ and (ii) $\max\{p_1^*, p_2^*\} = p_1^*$. In both of these cases we know the value of p_5^*, so we can calculate the rest.

In case (i), the equations reduce to

$$p_2^* = 2p_1^*/5 + 2/5$$
$$p_1^* = p_2^*/3 + 1/2$$

These can be solved to get $p_1^* = 19/26$ and $p_2^* = 18/26$. This contradicts our assuption that $p_2^* = \max\{p_1^*, p_2^*\}$.

In case (ii), the equations reduce to

$$p_2^* = 2p_1^*/5 + 2/5$$
$$p_1^* = p_1^*/6 + p_6^*/6 + 1/2$$

which gives us $p_1^* = 17/23$ and $p_2^* = 16/23$. This gives us the full solution to the original problem: $p^* = (p_1^*, \ldots p_6^*) = (17/23, 16/23, 16/23, 0, 17/23, 1)$. Player 1s optimal strategy is to choose s_2 when at node s_3, and to choose s_1 when at node s_5.

2. As we are working with a congestion game, we can find a pure Nash Equilibrium by starting at any pure strategy profile, and iteratively improving it until we can't. To get a concrete starting point, let's say all players take the route $s \to v_3 \to t$. Then we can do iterative improvements for example[1] as follows:

   (i) Player 1 switches to $s \to v_2 \to v_1 \to t$
   (ii) Player 2 switches to $s \to v_2 \to v_1 \to t$
   (iii) Player 3 switches to $s \to v_1 \to t$.
   (iv) Player 2 switches to $s \to v_1 \to t$

   At (iv) no further improvements can be made, so we reached the following NE:

   Player 1: $s \to v_2 \to v_1 \to t$
   Player 2: $s \to v_1 \to t$
   Player 3: $s \to v_1 \to t$

[1] at many stages there's more than one option on who improves and how

Note that in the above sequence we weren't done at stage (iii), even though every player had switched once. Other starting points will take through other sequences of steps, and they might end up in a different NE, although it turns out that in this game all pure Nash equilibria send two players via the route $s \to v_1 \to t$ and one via $s \to v_2 \to v_1 \to t$, differing only in which player chooses the path $s \to v_2 \to v_1 \to t$.

Note that there are "different" pure NEs in this game for a trivial reason: we can arbitrarily rename the players in any permutation.

However, this is only a different pure NE in a trivial sense, by renaming which player plays which strategy.

It turns out that in this game there is no other "fundamentally different" pure NE, meaning a pure NE consisting of an entirely different set of 3 paths from s to t.

In general however, such an atomic network congestion came can have multiple genuinely different pure NEs, where the players have completely different costs in the different NEs.