In this talk I will use arm and sub-process interchangeably.

Slide 7:

If we fix lambda, and remove the term in the objective, then this formulation is the LP approach to solving the Dynamic program Q(\lambda).

Then we can combine the outer minimization over lambda and obtain this LP (7)

I’m going to give you a super hand wavy explanation of how the proof works.

Assume the fraction of arms over states under the index policy is the same as that under pi\*, at time t, that is, say, this column is the same as that column (point to graph). Likewise, assume that the fraction of arms over state-action pairs under the index policy is the same as that under pi^\*. That is, this column….

Then at time t+1, there will be exactly the same fraction of arms going into each state because they started in the same distribution at time t, and in each state the fraction of arms that get played and don’t get played is the same for index policy and pi\* because of the second assumption. So And that proves the first line in theorem 2

By the property of pi\*, we know that the fraction of arms in the states in which pi\* decide to take active action = alpha\_t+1.

By definition of the index, the state that get to played under pi\* are the states with indices > lambda\_t+1\*. So it is roughly true that the fraction of arms in states with indices > lambda\_t+1\* is alpha\_t+1.

So this means that the fraction of arms in states with indices > lambda\_t+1\* is also alpha\_t+1 under the index policy at t+1 , because we have just shown that the distributions of arms are the same at time t+1 under the two policy. Since the index policy ranks the arms by the indices, and then pick the largest alpha\_{t+1} \* K, we know that the arms get to be played are the ones with indices greater than lambda\_t+1\*. So we have the same set of arms being played under the two policy, and that proved the second line.