Model Description (updated 05/04/2016)

Base Case: decision at every time epoch

- **Index and notation**

  N^A, N^B = number of ALS and BLS servers in the system
  m = total number of call types
  λ_t^{ip} = arrival rates for calls of type i and priority p that arrive per unit time at time epoch t, $i \in \{1, \ldots, m\}$, $p \in \{1, 2, 3\}$, $t \in \{1, \ldots, T\}$
  μ^A = service rate of ALSs
  μ^B = service rate of BLSs
  α^{ip} = probability that the call of type i, priority p needs ALS server
  $p \in \{1, 2, 3\}$ = reported priority
  $q \in \{H, L\}$ = realized true priority
  $k \in \{A, B\}$ = classified priority: decide whether to send ALS or BLS

  Every call has its type $i \in \{1, \ldots, m\}$ and reported priority $p \in \{1, 2, 3\}$. Depending on the system state and call type, we seek for the decision vector $a_t(s_t)$ that assigns either ALS or BLS to the arriving call.

- **Assumptions**

  1. We may have stationary or nonstationary arrival rates. For non-stationary case, arrival rate function is modeled as a piecewise linear function constructed by the interpolation between hourly arrival rates(either given by our data or randomly generated). When we assume a non-stationary arrival rate, transition probabilities and rewards are non-stationary as well.
  2. We make a new reprioritization decision at every time epoch. In other words, we make a decision whenever a call arrives.
  3. Priority 1 always need ALS and priority 3 always need BLS. Hence $\alpha^{i1} = 1$ and $\alpha^{i3} = 0$ for every call type i. Hence for notational simplexity, we can omit the priority superscript term in accuracy, and use α^i to demote probability that a call of type i and priority 2 needs ALS.
  4. Probability of reaching in time given a number of busy servers is modelled as monotone non-increasing functions $f^H(\cdot)$ for high priority calls and $f^L(\cdot)$ for low priority calls, which could be a convex/concave/linear function or take any other form.

- **Transition probability**

We set a uniformization factor Λ as

$$\Lambda = \max_t \left(\sum_{i=1}^{m} \sum_{p=1}^{3} \lambda_t^{ip} \right) + n^A \mu^A + n^B \mu^B$$

Let $P_t(j|s,a)$ denote the one-stage transition probabilities from $s = (s^A, s^B)$ to $j = (j^A, j^B)$ under action $a = (a^1, \ldots, a^m)$. We have

$$P_t(j|s,a) = \begin{cases} \sum_{i=1}^{m} (\lambda_t^{i1} + \lambda_t^{i2}(1-a^i))/\Lambda & \text{if } j = (s^A+1, s^B) \\ \sum_{i=1}^{m} (\lambda_t^{i3} + \lambda_t^{i2} a^i)/\Lambda & \text{if } j = (s^A, s^B+1) \\ s^A \mu^A/\Lambda & \text{if } j = (s^A-1, s^B) \\ s^B \mu^B/\Lambda & \text{if } j = (s^A, s^B-1) \\ 1 - (s^A\mu^A + s^B\mu^B + \sum_{i=1}^{m}\sum_{p=1}^{3} \lambda_t^{ip})/\Lambda & \\ +I(s^A = N^A) \sum_{i=1}^{m} (\lambda_t^{i1} + \lambda_t^{i2}(1-a^i))/\Lambda & \\ +I(s^B = N^B) \sum_{i=1}^{m} (\lambda_t^{i3} + \lambda_t^{i2} a^i)/\Lambda & \text{if } j = (s^A, s^B) \\ 0 & \text{otherwise.} \end{cases}$$

- **Reward**

1. Utility of serving a call
   $U_{q,A}$ is a utility gained by successfully serving a call with priority $q \in \{H, L\}$ by an ALS unit. $U_{q,B}$ is defined analogously.

	need ALS (H)	**need BLS (L)**
send ALS (A)	U_{HA}	U_{LA}
send BLS (B)	U_{HB}	U_{LB}

   Obviously We have $U_{HA} > U_{LA}$, $U_{HA} > U_{L,B}$ and $U_{HA} > U_{HB}$. Here we also assume $U_{LA} = U_{LB}$.

2. probability of successfully serving a call
   Given the action for call type i is 0(1) and the realized priority of the call is H, $f^H(s)$ represents the probability of successfully serving a call (in a time threshold) when s ALSs(BLSs) units are busy. $f^L(s)$ is also defined and evaluated analogously for low priority calls.

3. probability of being urgent given that call type is i and priority is p
   This is externally given as α^{ip}.

Using above parameters and functions, reward function is given as

$$
\begin{aligned}
R(s^A, s^B, a^1, \ldots, a^m) = & \sum_i \lambda_t^{i1}(\alpha^{i1} f^H(s^A) U_{HA} + (1-\alpha^{i1}) f^L(s^A) U_{LA})/\Lambda \\
& + \sum_i \lambda_t^{i2}(1-a^i)\alpha^{i2} f^H(s^A) U_{HA} + (1-\alpha^{i2}) f^L(s^A) U_{LA})/\Lambda \\
& + \sum_i \lambda_t^{i2} a^i (\alpha^{i2} f^H(s^B) U_{HB} + (1-\alpha^{i2}) f^L(s^B) U_{LB})/\Lambda \\
& + \sum_i \lambda_t^{i3}(\alpha^{i3} f^H(s^B) U_{HB} + (1-\alpha^{i3}) f^L(s^B) U_{LB})/\Lambda
\end{aligned}
$$

Advanced Case: distinct decision points

- **Decision points VS time epochs**

  We have a discrete decision point for every $g > 1$ time epochs. The action chosen in a decision point is maintained until the MDP reaches the next decision point.
  To be precise, we want to make a new choice of action for every other hour(or other fixed unit time), and therefore we increase our uniformization factor appropriately so that it evenly divides time difference between two successive decision points.
  Hence we have a set of time epoch $t \in T = \{1, 2, 3, \dots\}$ and a set of decision points $t_d \in T_d = \{1, 1+g, 1+2g, \dots\} \subset T$ which is a subset of time epochs.

- **Generating multi-step transition probability and reward**

  At time epoch t, for a vector of action $a = (a^1, \dots, a^m)$ we have a corresponding transition probability matrix $P_t(a)$ and reward vector $R_t(a)$ such that

$$[P_t(a)]_{ij} = P(j^A, j^B | s^A, s^B, a) \quad \text{where } i = (s^A, s^B), j = (j^A, j^B)$$
$$[R_t(a)]_i = R(s^A, s^B, a) \quad \text{where } i = (s^A, s^B).$$

  Then we need to generate a g-step transition probability matrix from time epoch t to $t+g$, P. Multi-step transition probability matrix is simply gained by multiplying transition probabilities in order.

$$P_t^g(a) = P_t(a) \cdot P_{t+1}(a) \cdot \dots \cdot P_{t+g-1}(a).$$

  We also need a vector of rewards, $R_t^g(a)$, which is accumulated during g time epochs after t. This is the sum of expected reward from each time epoch between decision points. To get the current expected value of reward matrix at time epoch $t+k$, we can apply the k-step transition matrix. Therefore we get the expected accumulated reward as

$$\begin{aligned} R_t^g(a) &= R_t(a) + P_t^1(a) \cdot R_{t+1}(a) + \dots + P_t^{g-1}(a) \cdot R_{t+g-1}(a) \\ &= R_t(a) + P_t(a) \cdot R_{t+1}(a) + \dots \\ &\quad + (P_t(a) P_{t+1}(a) \dots P_{t+g-2}(a) P_{t+g-2}(a)) \cdot R_{t+g-1}(a) \\ &= R_t(a) + \sum_{k=1}^{g-1} \left(\prod_{l=0}^{k-1} P_{t+l}(a) \right) R_{t+k}(a) \end{aligned}$$

  Then we can solve

$$u_t(s) = \max_{a \in \{0,1\}^m} \left([R_t^g(a)]_s + [P_t^g(a)]_{s\cdot} \, u_{t+g}(s) \right)$$

  for every $s = (s^A, s^B)$ and $t \in T_d$ for backward induction.

- **Action Space**

  As we use multiple-step transition probability matrix, now there may be more than one call arrival between decisions. Naturally, separability between call types does not hold any longer. The decision maker should enumerate and compare expected future rewards from all possible action combinations.