Supplementary Materials for Dynamic Service Migration in Mobile Edge-Clouds

Shiqiang Wang*, Rahul Urgaonkar†, Murtaza Zafer‡¶, Ting He†, Kevin Chan§, and Kin K. Leung*
*Imperial College London, United Kingdom, Email: {shiqiang.wang11, kin.leung}@imperial.ac.uk
†IBM T. J. Watson Research Center, Yorktown Heights, NY, United States, Email: {rurgaon, the}@us.ibm.com
‡Nyansa Inc., Palo Alto, CA, United States, Email: murtaza.zafer.us@ieee.org
§Army Research Laboratory, Adelphi, MD, United States, Email: kevin.s.chan.civ@mail.mil

I. Introduction

Supplementary materials for [1] is provided in this report. The variables, notations, and numberings (for equations, propositions, etc., except for sections and references) mentioned in this report refer to those in [1]. Newly introduced numberings in this report all start with prefix "S".

This report is organized as follows. Section II outlines the difference between service migration and cellular handover. The main notations in [1] are summarized in Section III. Section IV provides some additional discussions on the simulation results in Section IV-D of [1]. The proofs are given in Section V, and a method to approximate a general cost function with a constant-plus-exponential cost function is discussed in Section VI.

II. Relationship with Cellular Handover

A related area of work that is also relevant to the user mobility is handover policies in the context of cellular networks. However, the notion of service migration is quite different from cellular handover. Handover is usually decided by signal strengths from different basestations, and a handover must occur if a user is no longer covered by its original basestation. In the service migration context, a user may continue receiving service from an MEC even if it is no longer associated with that basestation, because the user can exchange data with the remote MEC via its currently associated basestation. As a result, the service for a particular user can potentially be placed on any MEC, and the service migration problem has a much larger decision space.

III. Summary of Notations

The main notations in [1] are summarized in Table S.1.

IV. Additional Discussions on Simulation Results in [1, Section IV-D]

A. Reduction in Computation Time

The computation time reduction is explained as follows. As discussed in Section [1, Section IV-B], for a distance-based MDP with N states (excluding state zero), the 2-D MDP has $M = 3N^2 + 3N$ states (excluding state $(0,0)$). When we ignore the complexity of matrix inversion in the policy iteration procedure, the standard value and policy iteration approaches on the 2-D MDP have a complexity of $O(M^2)$ in each iteration, because the optimal action needs to be found for each state, which requires enumerating through all the states and all possible actions for each state (similarly as in Lines 18–20 of Algorithm 1). In the simulations, $N = 10$, so we have $M = 330$. Recall that the complexity of Algorithm 1 used in the proposed approach is $O(N^2)$, so the ratio of the computational complexities of different approaches can be approximated by $\frac{M^2}{N^2} \approx 10^3$. Therefore, the standard value and policy iteration approaches consume about 10^3 times more computation time compared with the proposed approach.

B. Observations on Cost Value Characteristics

The cost from the optimal policy approaches the cost from the always-migrate policy when $-\beta_l$ is small, and it approaches the cost from the never-migrate policy when $-\beta_l$ is large. This is because, when $-\beta_l$ is small, the migration cost is relatively small, and migration can be beneficial for most cases; when $-\beta_l$ is large, the migration cost is large, and it is better not to migrate in most cases. The myopic policy is the same as the never-migrate policy when $-\beta_l \geq 0.5$, because $c_m(x)$ and $c_d(x)$ are both concave according to the simulation settings and we always have $c_m(x) \geq c_d(x)$ when $-\beta_l \geq 0.5$. It is also because the myopic policy does not consider the future impact of actions. There is an intersection of the costs from never-migrate and always-migrate policies. When $-\beta_l$ takes values that are larger than the value at the intersection point, the optimal cost is close to the cost from the never-migrate policy when γ is small, and the gap becomes larger with a larger γ. The reason is that the impact of migration is more obvious when we look further ahead into the future.

V. Proofs

A. Proof of Lemma 1

Assume that we are given a service migration policy π such that the service can be migrated to a location that is farther away from the user. We will show that, for an arbitrary sample path of the user, we can find a (possibly history dependent)

[¶] Contributions of the author to this work are not related to his current employment at Nyansa Inc.

Table S.1
SUMMARY OF MAIN NOTATIONS IN [1]

Notation	Description
$\mathcal{L}$	Set of locations
l	Location
$\Vert l_1 - l_2 \Vert$	Distance between locations l_1 and l_2
$u(t)$	User location at timeslot t
$h(t)$	Service location at timeslot t
$c_m(x)$	Migration cost
$c_d(x)$	Transmission cost
$s(t)$	Initial state at slot t
$s'(t)$	Intermediate state at slot t
π	Decision policy
$a(s)$	Action taken when system is in state $s(t)$
$C_a(s)$	Sum of migration and transmission costs when taking action $a(s)$ in slot t
$V(s_0)$	Discounted sum cost when starting at state s_0
$P_{s'_0,s_1}$	Transition probability from intermediate state s'_0 to initial state s_1 (where s_1 is in the next slot)
γ	Discount factor of the MDP
$d(t)$	User-service distance in slot t before possible migration (state in the distance-based MDP)
N	Number of states (excluding state zero) in the distance-based MDP
p_0, p, q	Transition probabilities of the distance-based MDP (see Fig. 3)
$\beta_c, \beta_l, \delta_c,$ δ_l, μ, θ	Parameters related to the constant-plus-exponential cost function (see (4) and (5))
$A_k, B_k, D,$ H, m_1, m_2	Parameters related to the closed-form solution of the distance-based MDP (see (7)–(10))
$\{n_k : k \geq 0\}$	Series of migration states (i.e., all n_k such that $a(n_k) \neq n_k$)
r	Transition probability to one of the neighbors in the 2-D model
$e(t)$	Offset of the user from the service as a 2-D vector (state in the 2-D MDP)

policy ψ that does not migrate to locations farther away from the user, which performs not worse than policy π.

In the following, we consider an arbitrary sample path of the user. For this sample path, denote the timeslot t_0 as the *first* timeslot (starting from $t = 0$) in which the service is migrated to somewhere farther away from the user when following policy π. The initial state at timeslot t_0 is denoted by $s(t_0) = (u(t_0), h(t_0))$. When following π, the state shifts from $s(t_0)$ to $s'_\pi(t_0) = (u(t_0), h'_\pi(t_0))$, where $\Vert u(t_0) - h'_\pi(t_0) \Vert > \Vert u(t_0) - h(t_0) \Vert$. The subsequent states in this case are denoted by $s_\pi(t) = (u(t), h_\pi(t))$ for $t > t_0$.

Now, we define a policy ψ such that the following conditions are satisfied for the given sample path of the user:

- The migration actions in timeslots $t < t_0$ are the same when following either ψ or π.
- The policy ψ specifies that there is no migration within the timeslots $t \in [t_0, t_m - 1]$, where $t_m > t_0$ is a timeslot index that is defined later.
- The policy ψ is defined such that, at timeslot t_m, the service is migrated to $h'_\pi(t_m)$, where $h'_\pi(t_m)$ is the service location (after possible migration at timeslot t_m) when following π.
- For timeslots $t > t_m$, the migration actions for policies ψ and π are the same.

For $t > t_0$, the states when following ψ are denoted by $s_\psi(t) = (u(t), h_\psi(t))$.

The timeslot t_m is defined as the *first* timeslot after t_0 such that the following condition is satisfied:

1) $\Vert u(t_m) - h_\psi(t_m) \Vert > \Vert u(t_m) - h'_\pi(t_m) \Vert$, i.e., the transmission cost (before migration at timeslot t_m) when following ψ is larger than the transmission cost (after possible migration at timeslot t_m) when following π.

Accordingly, within the interval $[t_0{+}1, t_m{-}1]$, the transmission cost when following ψ is always less than or equal to the transmission cost when following π, and there is no migration when following ψ. Therefore, for timeslots $t \in [t_0, t_m - 1]$, policy π cannot bring lower cost than policy ψ.

In timeslot t_m, we can always choose a migration action for ψ where the migration cost is smaller than or equal to the sum of the migration costs of π within $[t_0, t_m]$. The reason is that ψ can migrate (in timeslot t_m) following the same migration path as π within $[t_0, t_m]$.

It follows that, for timeslots within $[t_0, t_m]$, policy π cannot perform better than policy ψ, and both policies have the same costs for timeslots within $[0, t_0{-}1]$ and $[t_m{+}1, \infty)$. The above procedure can be repeated so that all the migration actions to a location farther away from the user can be removed without increasing the overall cost, thus we can redefine ψ to be a policy that removes all such migrations.

We note that the policy ψ can be constructed without prior knowledge of the user's future location. For any policy π, the policy ψ is a policy that does not migrate whenever π migrates to a location farther away from the user (corresponding to timeslot t_0), and then it migrates to $h'_\pi(t_m)$ when condition 1 is satisfied (corresponding to timeslot t_m).

The policy ψ is a history dependent policy, because its actions depend on the past actions of the underlying policy π. From [2, Section 6], we know that history dependent policies cannot outperform Markovian policies for our problem (note that the optimal policy found from (2) and (3) are Markovian policies). Therefore, there exists a Markovian policy that does not migrate to a location farther away from the user, which does not perform worse than π. Thus, we have proved the lemma.

B. Proof of Corollary 1

The proof follows the same procedure as the proof of Lemma 1. For any given policy π that migrates to locations other than the user location, we show that there exists a policy ψ that does not perform such migration, which performs not worse than the original policy π. The difference is that t_0 is defined as the first timeslot such that $u(t_0) \neq h'_\pi(t_0)$, and t_m is defined as the first timeslot after t_0 such that $u(t_m) = h'_\pi(t_m)$. Due to the strict inequality relationship of the cost functions given in the corollary, and because $0 < \gamma < 1$, we can conclude that π is not optimal.

C. Proof of Proposition 1

For a given policy, the discounted sum cost $V(d)$ for $0 \leq d \leq N$ can be computed from a set of difference equations [3]. For $0 < d < n_0$ or $n_{k-1} < d < n_k$, we have $a(d) = d$, according to (6), we have

$$V(d) = \delta_c + \delta_l \theta^d + \gamma \sum_{d_1=d-1}^{d+1} P_{dd_1} V(d_1) \tag{S.1}$$

We can rewrite (S.1) as

$$V(d) = \phi_1 V(d-1) + \phi_2 V(d+1) + \phi_3 + \phi_4 \theta^d \tag{S.2}$$

where $\phi_1 \triangleq \frac{\gamma q}{1-\gamma(1-p-q)}$, $\phi_2 \triangleq \frac{\gamma p}{1-\gamma(1-p-q)}$, $\phi_3 \triangleq \frac{\delta_c}{1-\gamma(1-p-q)}$, and $\phi_4 \triangleq \frac{\delta_l}{1-\gamma(1-p-q)}$.

This difference function has characteristic roots

$$m_1 = \frac{1+\sqrt{1-4\phi_1\phi_2}}{2\phi_2}, m_2 = \frac{1-\sqrt{1-4\phi_1\phi_2}}{2\phi_2} \tag{S.3}$$

When $0 < \gamma < 1$, we always have $m_1 \neq m_2$. The homogeneous version of (S.2) has general solution

$$V_h(d) = A_k m_1^d + B_k m_2^d \tag{S.4}$$

To solve the non-homogeneous equation (S.2), we try a particular solution in the form of

$$V_p(d) = \begin{cases} D + H \cdot \theta^d & \text{if } 1 - \frac{\phi_1}{\theta} - \phi_2\theta \neq 0 \\ D + Hd \cdot \theta^d & \text{if } 1 - \frac{\phi_1}{\theta} - \phi_2\theta = 0 \end{cases} \tag{S.5}$$

where D and H are constant coefficients. By substituting (S.5) into (S.2), we get

$$D = \frac{\phi_3}{1-\phi_1-\phi_2} \tag{S.6}$$

$$H = \begin{cases} \frac{\phi_4}{1-\frac{\phi_1}{\theta}-\phi_2\theta} & \text{if } 1 - \frac{\phi_1}{\theta} - \phi_2\theta \neq 0 \\ \frac{\phi_4}{\frac{\phi_1}{\theta}-\phi_2\theta} & \text{if } 1 - \frac{\phi_1}{\theta} - \phi_2\theta = 0 \end{cases} \tag{S.7}$$

Therefore, $V(d)$ can be expressed as (7).

D. Proof of Proposition 2

Recall that among the neighbors (i', j') of a cell (i, j) in the 2-D offset-based MDP $\{e(t)\}$, when $i > 0$, we have two (or, correspondingly, one) cells with $i' = i - 1$, and two (or, correspondingly, three) cells with $i' = i + 1$. To "even out" the different number of neighboring cells, we define a new (modified) MDP $\{y(t)\}$ for the 2-D offset-based model, where the states are connected as in the original 2-D MDP, but the transition probabilities are different, as shown in Fig. S.1.

In the modified MDP $\{y(t)\}$, the transition probabilities starting from state $(0, 0)$ to each of its neighboring cells have the same value r. For all the other states $(i, j) \neq (0, 0)$, they are defined as follows:

- The transition probability to each of its neighbors with the same ring index i is r.
- If state (i, j) has two (correspondingly, one) neighbors in the lower ring $i-1$, then the transition probability to each of its neighbors in the lower ring is $\frac{1.5}{2}r$ (correspondingly, $1.5r$).
- if state (i, j) has two (correspondingly, three) neighbors in the higher ring $i + 1$, then the transition probability to each of its neighbors in the higher ring is $\frac{2.5}{2}r$ (correspondingly, $\frac{2.5}{3}r$).

We denote the discounted sum cost from the original MDP $\{e(t)\}$ by $V(i, j)$, and denote that from the modified MDP $\{y(t)\}$ by $U(i, j)$, where (i, j) stands for the initial state in the discounted sum cost definition.

Part I – Upper bound on the difference between $V(i, j)$ and $U(i, j)$ for a given policy π

Assume we have the same policy π for the original and modified MDPs. Then, the balance equation of $V(i, j)$ is

$$V(i,j) = C_a(i,j) + \gamma \left((1-6r)\, V(a(i,j)) + r \sum_{(i',j') \in \mathcal{N}(a(i,j))} V(i',j') \right) \tag{S.8}$$

where $a(i, j)$ is the new state after possible migration at state (i, j), and $\mathcal{N}(a(i, j))$ is the set of states that are neighbors of state $a(i, j)$.

For $U(i, j)$, we have

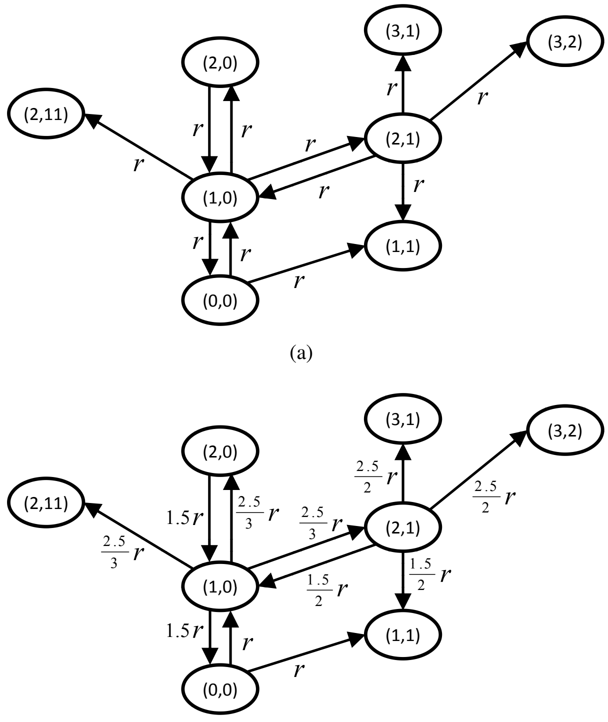
$$\begin{aligned} U(i,j) = C_a(i,j) + \gamma \Bigg(& (1-6r)\, U(a(i,j)) \\ & + \sum_{(i',j') \in \mathcal{N}(a(i,j))} P_{y'(t)=a(i,j), y(t+1)=(i',j')} U(i',j') \Bigg) \end{aligned} \tag{S.9}$$

where $P_{y'(t)=a(i,j), y(t+1)=(i',j')}$ is the transition probability of the modified MDP $\{y(t)\}$ as specified earlier.

In the following, let $i_{a(i,j)}$ denote the ring index of $a(i, j)$. We define sets

$$\mathcal{N}^-(a(i,j)) = \left\{ (i',j') \in \mathcal{N}(a(i,j)) : i' = i_{a(i,j)} - 1 \right\}$$
$$\mathcal{N}^+(a(i,j)) = \left\{ (i',j') \in \mathcal{N}(a(i,j)) : i' = i_{a(i,j)} + 1 \right\}$$

$$
\begin{aligned}
&|U(i,j)-V(i,j)| \\
&=\gamma\Big|\left(1-6r\right)\left(U(a(i,j))-V(a(i,j))\right) \\
&\quad+\sum_{(i',j')\in\mathcal{N}(a(i,j))} P_{y'(t)=a(i,j),y(t+1)=(i',j')}\left(U(i',j')-V(i',j')\right) \\
&\quad\pm 0.5\lambda_{i_{a(i,j)}} r\left(\frac{\sum_{(i',j')\in\mathcal{N}^{+}(a(i,j))}V(i',j')}{|\mathcal{N}^{+}\left(a(i,j)\right)|}-\frac{\sum_{(i',j')\in\mathcal{N}^{-}(a(i,j))}V(i',j')}{|\mathcal{N}^{-}\left(a(i,j)\right)|}\right)\Big| && \text{(S.10)} \\
&\le\gamma\epsilon+0.5\gamma r\left|\frac{\sum_{(i',j')\in\mathcal{N}^{+}(a(i,j))}V(i',j')}{|\mathcal{N}^{+}\left(a(i,j)\right)|}-\frac{\sum_{(i',j')\in\mathcal{N}^{-}(a(i,j))}V(i',j')}{|\mathcal{N}^{-}\left(a(i,j)\right)|}\right| && \text{(S.11)} \\
&\le\gamma\epsilon+0.5\gamma r\max_{i,j,j':(i+1,j)\in\mathcal{N}_2(i-1,j')}\left|\left(V(i+1,j)-V(i-1,j')\right)\right| && \text{(S.12)}
\end{aligned}
$$



(a)

(b)

Figure S.1. Illustration of original and modified 2-D MDPs, only some exemplar states and transition probabilities are shown: (a) original, (b) modified.

to represent the neighboring states of $a(i,j)$ that are respectively in the lower and higher rings. We use $|\cdot|$ to denote the number of elements in a set.

Assume $|U(i,j)-V(i,j)|\le\epsilon$ for all i and j, and the value of ϵ is unknown for now. We subtract (S.8) from (S.9), and then take the absolute value, yielding (S.10), (S.11), and (S.12), where the set $\mathcal{N}_2\left(i,j\right)$ is the set of states that are two-hop neighbors of state (i,j), the variable $\lambda_{i_{a(i,j)}}=0$ when $i_{a(i,j)}=0$, and $\lambda_{i_{a(i,j)}}=1$ when $i_{a(i,j)}>0$.

The first two terms of (S.10) subtract the discounted sum cost of the original MDP $\{e(t)\}$ from that of the modified MDP $\{y(t)\}$, by assuming that both chains have the *same* transition probabilities specified by the modified MDP. The difference in their transition proabilities is captured by the last term of (S.10). There is no difference in the transition probabilities when $i_{a(i,j)}=0$, thus $\lambda_{i_{a(i,j)}}=0$ when $i_{a(i,j)}=0$.

In the following, we consider $i_{a(i,j)}>0$ and further explain the last term of (S.10). We first note that there is difference in the transition probabilities only when moving to the lower or higher ring:

- The *sum* probability of moving to the lower ring in $\{y(t)\}$ is by $0.5r$ smaller (or, correspondingly, greater) than that in $\{e(t)\}$.
- The *sum* probability of moving to the higher ring in $\{y(t)\}$ is by $0.5r$ greater (or, correspondingly, smaller) than that in $\{e(t)\}$.

Therefore, the transition probablity difference for each neighboring state in the lower (or higher) ring is $\pm0.5r$ divided by the number of neighbors in the lower (or higher) ring. Also note that the probablity difference for lower and higher rings have opposite signs. This explains the last term of (S.10), which captures the difference in the transition probabilities and its impact on the discounted sum costs.

The inequality in (S.11) is from the triangle inequality. We note that the subtraction in the last term of (S.10) only occurs on $V(i',j')$ values that are two-hop neighbors, so we have the inequality in (S.12) by replacing the value with the maximum.

From (S.12), we can obtain a balance equation for the upper bound of $|U(i,j)-V(i,j)|$, which is

$$
\epsilon=\gamma\epsilon+0.5\gamma r\max_{i,j,j':(i+1,j)\in\mathcal{N}_2(i-1,j')}\left|\left(V(i+1,j)-V(i-1,j')\right)\right| \tag{S.13}
$$

Because $0<\gamma<1$, the value of $V(i,j)$ converges after a number of iterations according to (S.8) [2, Section 6], so $|(V(i+1,j)-V(i-1,j))|$ also converges, and the value of ϵ can be solved by

$$
\epsilon_V=\frac{\gamma r\max_{i,j,j':(i+1,j)\in\mathcal{N}_2(i-1,j')}\left|\left(V(i+1,j)-V(i-1,j')\right)\right|}{2(1-\gamma)} \tag{S.14}
$$

Note that the above argument also applies when interchanging V and U, which means that an alternative upper bound of the cost is

$$\epsilon_U = \frac{\gamma r \max_{i,j,j':(i+1,j)\in\mathcal{N}_2(i-1,j')} |(U(i+1,j)-U(i-1,j'))|}{2(1-\gamma)} \tag{S.15}$$

The upper bound can also be expressed as $\epsilon = \min\{\epsilon_V, \epsilon_U\}$, but either ϵ_V or ϵ_U may have the smaller value.

Part II – Optimal policy for the modified 2-D MDP $\{y(t)\}$ is equivalent to the optimal policy for the distance-based MDP $\{d(t)\}$

We note that the optimal policy of an MDP can be found from value iteration [2, Section 6]. For the modified MDP $\{y(t)\}$, we initialize with a never migrate policy, which gives the initial value function $U_0(i,j) = c_d(i)$, satisfying $U_0(i,j) = U_0(i,j')$ for all i and $j \neq j'$.

Suppose $U_n(i,j) = U_n(i,j')$ for all i and $j \neq j'$. In each iteration, we use the following equation to obtain the new value function and the corresponding actions for each i and j:

$$U_{n+1}(i,j) = \min_a \Bigg\{ C_a(i,j) + \gamma \sum_{i'} \sum_{j'} P_{y'(t)=a(i,j),y(t+1)=(i',j')} U_n(i',j') \Bigg\} \tag{S.16}$$

From the hexagon model in Fig. 5, we can see that for ring indexes i and i', where $i' < i$, we can always reach from state (i,j) to a state in ring i' with $i - i'$ hops, regardless of the index j. In the $(n+1)$th iteration, if it is optimal to migrate from state (i,j) to a state with ring index i', then the migration destination must be $i - i'$ hops away from origin state, because $U_n(i',j) = U_n(i',j')$ for all i' and $j \neq j'$, it cannot be beneficial to migrate to somewhere further away.

Further, if it is optimal to migrate at a state (i,j) to a state in ring i', it must be optimal to migrate at states (i,j) for all j to a state (which may not be the same state) in ring i', bringing the same cost, i.e. $U_{n+1}(i,j) = U_{n+1}(i,j')$ for $j \neq j'$. This is due to the symmetry of cost functions (in the sense that $U_n(i,j) = U_n(i,j')$ for all i and $j \neq j'$) and symmetry of transition probabilities (in the sense that the sum probability of reaching ring $i-1$ from any state in ring i is the same, and the sum probability of reaching ring $i+1$ from any state in ring i is also the same). Similarly, if it is optimal not to migrate at a state (i,j), then it is optimal not to migrate at states (i,j) for all j, which also brings $U_{n+1}(i,j) = U_{n+1}(i,j')$ for any $j \neq j'$.

Because the value iteration converges to the optimal policy and its corresponding cost as $n \to \infty$, for the optimal policy of the modified MDP $\{y(t)\}$, we have the same discounted sum cost for states in the same ring, i.e. $U^*(i,j) = U^*(i,j')$ for all i and $j \neq j'$. Meanwhile, for a given i, the optimal actions $a^*(i,j)$ and $a^*(i,j')$ for any $j \neq j'$ have the same ring index $i_{a^*(i,j)}$.

Since the optimal actions $a^*(i,j)$ only depend on the ring index i and the ring index of $a^*(i,j)$ does not change with j, the optimal policy for $\{y(t)\}$ can be directly mapped to a policy for the distance-based MDP $\{d(t)\}$. A policy for $\{d(t)\}$ can also be mapped to a policy for $\{y(t)\}$ by considering the shortest path between different states in $\{y(t)\}$, as discussed in [1, Section IV-B]. This implies that there is a one-to-one mapping between the optimal policy for $\{y(t)\}$ and a policy for $\{d(t)\}$, because the optimal policy for $\{y(t)\}$ also only migrates along the shortest path between states.

We now show that the policy for $\{d(t)\}$ obtained from the optimal policy for $\{y(t)\}$ is optimal for $\{d(t)\}$. To find the optimal policy for $\{d(t)\}$, we can perform value iteration according to the following update equation:

$$U_{n+1}(i) = \min_a \Bigg\{ C_a(i) + \gamma \sum_{i'} \Bigg(\sum_{j'} P_{y(t)=a(i),y(t+1)=(i',j')} \Bigg) U_n(i') \Bigg\} \tag{S.17}$$

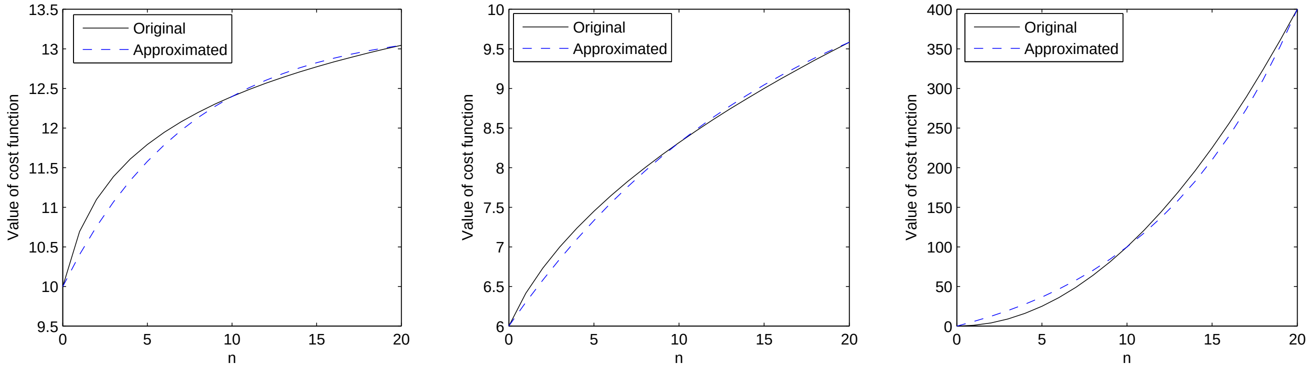
The difference between (S.16) and (S.17) is that (S.17) does not distinguish the actions and value functions with different j indexes. Recall that for the modified 2-D MDP $\{y(t)\}$, we have $U_n(i,j) = U_n(i,j')$ for all n, i and $j \neq j'$, so the index j can be natually removed from the value functions. Further, if it is optimal to migrate at state (i,j) to a state in ring i', it must be optimal to migrate at states (i,j) for all j to a state (which may not be the same state) in ring i'. The migration cost for different j are the same because they all follow the shortest path from state (i,j) to ring i'. If it is optimal not to migrate at state (i,j), then it is optimal not to migrate at states (i,j) for all j. Therefore, we can also remove the j index associated with the actions, without affecting the value function. It follows that the optimal policy for $\{y(t)\}$ is equivalent to the optimal policy for $\{d(t)\}$, both bringing the same value functions (discounted sum costs).

Part III – Error bound for distance-based approximation

By now, we have shown the upper bound on the discounted sum cost difference between the original and modified 2-D MDPs $\{e(t)\}$ and $\{y(t)\}$, when both MDPs use the same policy. We have also shown that the optimal policy for the modified 2-D MDP $\{y(t)\}$ is equivalent to the optimal policy for the distance-based MDP $\{d(t)\}$. Note that the true optimal cost is obtained by solving for the optimal policy for the original 2-D MDP $\{e(t)\}$, and the approximate optimal cost is obtained by applying the optimal policy for $\{d(t)\}$ to $\{e(t)\}$. In the following, we consider the upper bound on the difference between the true and approximate optimal discounted sum costs.

We start with a (true) optimal policy π^*_{true} for $\{e(t)\}$, denote the discounted sum costs from this policy as $V_{\pi^*_{true}}(i,j)$. When using the same policy on $\{y(t)\}$, the difference between the costs $U_{\pi^*_{true}}(i,j)$ and $V_{\pi^*_{true}}(i,j)$ satisfies the upper bound given in (S.14), i.e.

$$U_{\pi^*_{true}}(i,j) - V_{\pi^*_{true}}(i,j) \leq \epsilon_{V_{\pi^*_{true}}} \tag{S.18}$$



(a)

(b)

(c)

Figure S.2. Examples of approximating a general cost function with exponential cost function: (a) $f(n) = \ln(n+1) + 10$, (b) $f(n) = \sqrt{n+1} + 5$, (c) $f(n) = n^2$.

Since $V_{\pi^*_{true}}(i,j)$ are the optimal costs, we have

$$\max_{i,j,j':(i+1,j)\in\mathcal{N}_2(i-1,j')} \left|\left(V_{\pi^*_{true}}(i+1,j) - V_{\pi^*_{true}}(i-1,j')\right)\right| \leq \max_x \{c_m(x+2) - c_m(x)\} \tag{S.19}$$

because, otherwise, there exists at least one pair of states $(i+1,j)$ and $(i-1,j')$ for which it is beneficial to migrate from state $(i+1,j)$ to state $(i-1,j')$, according to the 2-D extension of (11). The cost after performing such migration is upper bounded by (S.19), which contradicts with the fact that π^*_{true} is optimal.

Define

$$\epsilon_c = \frac{\gamma r \max_x \{c_m(x+2) - c_m(x)\}}{2(1-\gamma)} \tag{S.20}$$

From (S.14), (S.18) and (S.19), we have

$$U_{\pi^*_{true}}(i,j) - V_{\pi^*_{true}}(i,j) \leq \epsilon_c \tag{S.21}$$

According to the equivalence of $\{y(t)\}$ and $\{d(t)\}$, we know that the optimal policy π^*_{appr} of $\{d(t)\}$ is also optimal for $\{y(t)\}$. Hence, we have

$$U_{\pi^*_{appr}}(i,j) \leq U_{\pi^*_{true}}(i,j) \tag{S.22}$$

because the cost from the optimal policy cannot be higher than the cost from any other policy.

When using the policy π^*_{appr} on $\{e(t)\}$, we get costs $V_{\pi^*_{appr}}(i,j)$. From (S.15), and because $U_{\pi^*_{appr}}(i,j)$ is the optimal cost for $\{y(t)\}$, we have

$$V_{\pi^*_{appr}}(i,j) - U_{\pi^*_{appr}}(i,j) \leq \epsilon_{U_{\pi^*_{appr}}} \leq \epsilon_c \tag{S.23}$$

From (S.21), (S.22), and (S.23), we get

$$V_{\pi^*_{appr}}(i,j) - V_{\pi^*_{true}}(i,j) \leq 2\epsilon_c \tag{S.24}$$

which completes the proof.

VI. Approximating a General Cost Function with Constant-Plus-Exponential Cost Function

In the following, we only focus on the migration cost function $c_m(x)$, because $c_m(x)$ and $c_d(x)$ are defined in the same way. For a given cost function $f(n)$, where $n = 0,1,2,...$, we would like to approximate it with the constant-plus-exponential cost function given by $c_m(n) = \beta_c + \beta_l \mu^n$. Because $c_m(n)$ includes both a constant term and an exponential term, we cannot obtain an exact analytical expression to minimize a commonly used error function, such as the mean squared error. One way, however, is to solve for the parameters that minimize the error cost function numerically. To reduce the computational complexity compared with a numerical solution, we provide an approximate solution in this section.

We are given an integer w that may be chosen according to practical considerations, and we solve for the parameters β_c, β_l, and μ according to the following system of equations:

$$\beta_c + \beta_l = f(0) \tag{S.25}$$

$$\beta_c + \beta_l \mu^w = f(w) \tag{S.26}$$

$$\beta_c + \beta_l \mu^{2w} = f(2w) \tag{S.27}$$

Subtracting (S.25) respectively from (S.26) and (S.27) and dividing the two results gives

$$\frac{\mu^{2w} - 1}{\mu^w - 1} = \frac{f(2w) - f(0)}{f(w) - f(0)} \tag{S.28}$$

subject to $\mu^w \neq 1$. We can then solve μ^w from (S.28) which gives

$$\mu^w = \frac{R \pm \sqrt{R^2 - 4(R-1)}}{2} \tag{S.29}$$

where $R \triangleq \frac{f(2w)-f(0)}{f(w)-f(0)} \geq 1$, it follows that we always have $R^2 - 4(R-1) \geq 0$ and $\mu^w \geq 0$. To guarantee that $\mu^w \neq 1$, we set $\mu^w = 1 \pm \epsilon_0$ if $|\mu^w - 1| < \epsilon_0$, where ϵ_0 is a small number and the sign is the same as the sign of $\mu^w - 1$. Then, we have

$$\mu = \left(\frac{R \pm \sqrt{R^2 - 4(R-1)}}{2} \right)^{\frac{1}{w}} \tag{S.30}$$

From which we can solve

$$\beta_c = \frac{f(0)\mu^w - f(w)}{\mu^w - 1} \tag{S.31}$$

$$\beta_l = \frac{f(w) - f(0)}{\mu^w - 1} \tag{S.32}$$

According to (S.30), we have two possible values of μ, and correspondingly two possible sets of values of β_c and β_l. To determine which is better, we compute the sum squared error $\sum_{n=0}^{2w} \left(f(n) - (\beta_c + \beta_l \mu^n)\right)^2$ for each parameter set, and choose the parameter set that produces the smaller sum squared error.

Some examples of approximation results are shown in Fig. S.2.

Acknowledgment

This research was sponsored in part by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," in *Proc. of IFIP Networking 2015*, May 2015.

[2] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2009, vol. 414.

[3] S. Elaydi, *An Introduction to Difference Equations, Third Edition*. Springer, 2005.