Fast computation of partial index and application to an AoI minimization problem

Haoqian Xue

Department of Information Engineering
The Chinese University of Hong Kong
1155205422@link.cuhk.edu.hk

Xiaojun Lin
Department of Information Engineering
The Chinese University of Hong Kong
xjlin@ie.cuhk.edu.hk

Abstract

In this paper, we study efficient algorithms for computing the partial index. We focus on an AoI (Age-of-Information) minimization problem under the generate-at-will setting such that multiple sources/agents transmit information updates to the base-station over multiple heterogeneous and unreliable wireless channels. While the partial index has been proposed to solve this otherwise exponential-complexity MDP problem, computing the partial index for each source still incurs significant complexity. Existing fast computation algorithms for Whittle index cannot be applied to this setting due to the multiple heterogeneous channels. Instead, we identify a number of general structural conditions for the per-agent MDP, based on which we develop a fast algorithm that can compute the partial index more efficiently. We then verify that the AoI problem under the generate-at-will setting satisfies these general conditions and our algorithm can compute the partial index for all states and all channels with a complexity of $\mathcal{O}(M^3K^3)$, where K denotes the number of per-source states and M denotes the number of channel types. Our numerical results confirm that our proposed algorithm is significantly faster in computing the partial index than standard methods based on binary search.

1 Introduction

Index policy has become an important tool for solving wireless scheduling problems minimizing the AoI (Age of Information) [10]. AoI, defined as the elapsed time since the last received packet was created at the source, is particularly suitable for measuring the freshness of information at the receiver [9]. Compared to the classical metric of latency, AoI can capture the application requirement that

(oftentimes) only information with the latest timestamp is valuable to the receiver, and outdated data packets are of little value. When information updates from multiple sources/agents are transmitted over unreliable wireless channels, the resulting AoI minimization problem can be modeled as a Markov Decision Process (MDP) [7, 8]. However, this MDP is known to suffer the curse of dimensionality, i.e., its complexity grows exponentially with the number of sources/agents. Index policies, including both the Whittle index [18] and the recently-developed partial index [19], can potentially help to decompose this large-scale MDP into per-agent problems, which can lead to solutions with much lower complexity.

For single-channel systems, the Whittle index [18] has been successfully applied to AoI minimization [6, 14, 16]. To understand how to compute the Whittle index, first, for every price λ of using the wireless channel, the agent needs to solve its own MDP minimizing the AoI plus the cost of using the channel. Then, the Whittle index is defined as the largest λ such that the agent still prefers to use the channel at the current state. Then, the Whittle index policy only needs to pick the agents with the highest indices for transmission. In this way, the complexity of the Whittle index policy does not grow with the number of agents [17]. In some simpler cases, the value function of the per-agent MDP can be solved in closed-form, based on which one can derive a closed-form expression of the Whittle index [15]. For more complex settings, however, neither the value function nor the Whittle index can be derived in closed-form. For these cases, a brute-force method for calculating the Whittle index is to conduct a binary search over different λ values, which involves solving the corresponding per-agent MDP repeatedly using value iteration or policy iteration [1]. This repeated solution of per-agent MDPs will incur high complexity [2, 12]. Thankfully, this computational difficulty for Whittle index has been addressed in [11] and [3], which propose fast algorithms that can compute the Whittle index of a restless bandit with K states in $\mathcal{O}(K^3)$ complexity.

When the system has multiple heterogeneous channels, each source will have more choices than simply "transmitting" or "not transmitting". Then, the Whittle index cannot be defined any more. To address this difficulty, a recent work [19] introduces the new notion of partial index, which extends the Whittle index to systems with multiple heterogeneous channels. The partial index leads to a low-complexity policy as well [19]. However, computing the partial index for each channel m at each agent state is also a highly non-trivial task. Like the Whittle index, the standard binary search will again incur high complexity due to the need to repeatedly solve the per-agent MDP. Unfortunately, the idea of fast Whittle-index computation in [11] and [3] cannot be directly generalized to the partial index. Note that in [11] and [3], the key idea is that, as λ increases, the passive set (i.e., the set of states that the optimal decision of the per-agent MDP does not transmit) expands sequentially. Thus, in order to obtain the Whittle index values for all of the states one by one, we can simply search which state should be added next to the passive set. Since every passive set corresponds to a unique policy, we only need to explore a small number of policies to find the correct next state to add. In contrast, for partial index, even if we fix the passive set for a given channel m, we still do not know how the passive sets for other channels will vary. Since there could be an exponential number of

possibilities for the other passive sets, using the idea of [11] and [3] alone is insufficient for lowering the complexity.

To overcome this difficulty, in this paper we identify a number of new and general structural conditions for the per-agent MDP, based on which we develop a fast algorithm for calculating the partial index. Our structural conditions define an ordering of the policies for the per-agent MDP, as well as a set of operations that can produce one policy after another policy. Once certain conditions for such ordering and operations are met, we can then design a fast algorithm that can compute the partial index with reduced complexity. We then verify that the AoI minimization problem under the generate-at-will setting satisfies these structural conditions, and our algorithm can compute the partial index for all states and channels with $\mathcal{O}(M^3K^3)$ complexity, where M is the number of different channel-types and K is the number of states. To the best of our knowledge, this is the first algorithm in the literature that can compute the partial index with complexity that grows cubically with the number of states (see detailed comparison with binary search in Section 4.1). Further, thanks to our general structural conditions, our fast algorithm can potentially be applied to other problems involving multi-agent MDPs sharing multiple heterogeneous resources.

The rest of the paper is structured as follows. In Section 2, we define both a specific system model for AoI minimization and a more general model that can potentially be used for other problems, and introduce the problem of computing the partial index. Our fast algorithm is presented in Section 3 for the general model, and then in Section 4 for the AoI model. Numerical results are presented in Section 5, and then we conclude.

2 System Model and Partial Index

2.1 System Model for AoI Minimization

In this section we will review the AoI-minimization model from [19] under the generate-at-will setting. Slightly different from [19], we use the discounted-cost MDP rather than the average-cost MDP. Nonetheless, the results from [6] can be easily extended to discounted cost.

We consider a wireless system where N data sources transmit information to the base-station (BS) over multiple heterogeneous channels in the uplink. Assume that time is slotted, i.e., each transmission takes exactly one unit of time. We denote the heterogeneous channel types as $\mathcal{M} = \{1, 2, ..., M\}$ and each channel type $m \in \mathcal{M}$ has C channel instances. Thus, on each type of channels, at most C sources can be scheduled for transmission at each time slot. We assume that $CM < N$. Due to the uncertainty of the wireless channel, each packet transmission of source n through a channel of type m succeeds with probability p_m^n, independently of other transmissions. Let $u_n(t) \in \mathcal{U} \triangleq \{0\} \cup \mathcal{M}$ be the scheduling decision for source n at time t, such that $u_n(t) = m$, if source n is scheduled to transmit on channel type m, and $u_n(t) = 0$, if the source will be passive and not transmit through any of the channels.

We then collect all the decisions into the action $\bar{U}(t) = [u_1(t), ..., u_N(t)]$. Note that the action $\bar{U}$

must respect both the resource constraints and the constraint that each source can be scheduled on at most one channel, i.e.,

$$\sum_{n=1}^{N} 1_{\{u_n(t)=m\}} \leq C, \quad \forall m \in \mathcal{M}, t = 1, 2, ..., \tag{1a}$$

$$\sum_{m=1}^{M} 1_{\{u_n(t)=m\}} \leq 1, \quad \forall n = 1, ..., N, t = 1, 2, ... \tag{1b}$$

We denote $h_n(t)$ as the AoI of source n at time t, which is the time elapsed from the generation time of the last received packet from the data source n to the current time t.

We assume the generate-at-will setting [19], i.e., whenever a data source is scheduled for transmission, it can generate a fresh packet immediately. Further, for technical reasons, we assume that the maximum age is bounded by K. Thus, the state evolution for source n can be written as

$$h_n(t+1) = \begin{cases} 1, & \text{transmission success} \\ \min\{h_n(t)+1, K\}, & \text{otherwise.} \end{cases}$$

Let $P_u^n(h_n, h_n')$ be the state transition probability of source n when it takes the action u. Thus, if source n is passive (i.e., $u = 0$), we have

$$P_0^n(h_n, \min\{h_n + 1, K\}) = 1. \tag{2}$$

If source n is scheduled on channel type m, we have,

$$P_m^n(h_n, 1) = p_m^n; \; P_m^n(h_n, \min\{h_n + 1, K\}) = 1 - p_m^n. \tag{3}$$

We now model our AoI minimization problem as an MDP. Specifically, we collect the AoI of all sources as the system state

$\bar{S}(t) \triangleq [h_1(t), h_2(t), \ldots, h_N(t)] \in \mathbb{N}_+^N$ at time t. A policy $\bar{\pi}$ then maps from the system state $\bar{S}(t)$ to the action $\bar{U}(t)$. Our goal is then to minimize the total discounted AoI of all sources starting from the initial states $\bar{S}(0)$, i.e.,

$$\min_{\bar{\pi}} \sum_{t=0}^{+\infty} \sum_{n=1}^{N} \beta^t \mathbb{E}_{\bar{S}(0)}^{\bar{\pi}} [h_n(t)] \tag{4}$$

subject to (1a) and (1b), where $0 < \beta < 1$ is the discount factor. Without loss of generality, we assume that the initial states are all $h_n(0) = 1$.

Unfortunately, this MDP is known to suffer the curse of dimensionality. Below, we will review the partial index introduced in [19], which leads to an asymptotically optimal solution to problem (4) with lower complexity.

2.2 The Relaxed Problem and Partial Index

Similar to the development of the Whittle index [18], we relax constraint (1a) to a discounted-sum constraint, and obtain the relaxed problem from (4):

$$\begin{aligned} &\min_{\bar{\pi}} \sum_{t=0}^{\infty} \sum_{n=1}^{N} \beta^t \mathbb{E}^{\bar{\pi}}_{\bar{S}_0} \left[h_n(t)\right] \\ &\text{s.t } \mathbb{E}^{\bar{\pi}}_{\bar{S}(0)} \left[\sum_{t=0}^{\infty} \sum_{n=1}^{N} \beta^t 1_{\{u^{\bar{\pi}}_n(t)=m\}}\right] \leq \frac{C}{1-\beta}, \forall m \\ &\quad \sum_{m=1}^{M} 1_{\{u^{\bar{\pi}}_n(t)=m\}} \leq 1, \quad \forall n \leq N, t = 1, 2, ... \end{aligned} \tag{5}$$

Next, we associate a Lagrange dual cost λ_m to each constraint with respect to m in (5), and form the Lagrangian. Minimizing the Lagrangian over π

can then be decoupled into N sub-problems, one for each source n:

$$\begin{aligned} &\min_{\bar{\pi}} \sum_{t=0}^{\infty} \beta^t \mathbb{E}^{\bar{\pi}}_{s_n(0)} \left[h_n(t) + \sum_{m=1}^{M} \lambda_m 1_{\{u^{\bar{\pi}}_n(t)=m\}}\right] \\ &\text{s.t.} \quad \sum_{m=1}^{M} 1_{\{u^{\bar{\pi}}_n(t)=m\}} \leq 1, \quad \forall t = 1, 2, ..., \end{aligned} \tag{6}$$

We refer to sub-problem (6) as the per-agent problem. Note that (6) is also an MDP. It is known that when $\vec{\lambda} = [\lambda_m]_{m \in \mathcal{M}}$ is optimally chosen, there exist optimal solutions to the per-agent MDP that will also provide an optimal solution to the relaxed problem (5). See [4] (Ch 3, Theorem 3.6).

However, the solution to the relaxed problem (5) does not respect the original "hard" constraint (1a). The goal of the partial index introduced in [19] is precisely to produce a policy that still satisfies the "hard" constraint (1a). Below, we will focus on the per-agent MDP (6) of a particular source n, and for ease of exposition, we will drop the index n whenever there is no source of confusion.

Let $V^*(s, \vec{\lambda})$ and $Q^*(s, u, \vec{\lambda})$ denote the optimal value function and optimal state-action value function, respectively, for (6). Thanks to the Bellman equation, they are related by:

$$V^*(h, \vec{\lambda}) = \min_{u \in \mathcal{U}} \{Q^*(h, u, \vec{\lambda})\} \tag{7}$$

$$Q^*(h, u, \vec{\lambda}) = h + \lambda_u + \beta \sum_{h'} P_u(h, h') V^*(h', \vec{\lambda}). \tag{8}$$

We can now define the partial indexability and partial index through the concept of the passive set [19]. Specifically, we focus on a given channel-type m. Let $\mathcal{S} = \{0, ..., K\}$ denote the per-source state-space.

Definition 1. (Passive Set) Given the dual cost $\vec{\lambda}$, the set of passive states for action $m \in \mathcal{M}$ is

$$\mathcal{P}_m(\vec{\lambda}) \triangleq \left\{ h \in \mathcal{S} \mid Q^*(h, m, \vec{\lambda}) > \min_{u \neq m, u \geq 0} Q^*(h, u, \vec{\lambda}) \right\}.$$

Given the current vector $\vec{\lambda} = [\lambda_1, ..., \lambda_M]$ of the dual costs, we now form a new $\vec{\lambda}' = [\lambda'_m, \vec{\lambda}_{-m}]$ by varying only λ_m to λ'_m but keeping all other dual costs $\vec{\lambda}_{-m}$ unchanged. For certain problems (including the AoI minimization problem under the generate-at-will setting [19]), the passive set can be shown to only expand as λ'_m increases. In that case, the partial indexability is said to hold, as defined below.

Definition 2. (Partial indexability) Given the cost vector $\vec{\lambda}$, the sub-problem (6) is partially indexable if, for all $m \in \mathcal{M}$, the size of the passive set $\mathcal{P}_m(\vec{\lambda}')$ increases monotonically to the entire state space $\mathcal{S}$ as λ'_m increases from 0 to ∞ (while fixing other channels' costs $\vec{\lambda}_{-m}$).

Definition 3. (Partial index) Given the dual cost $\vec{\lambda}$, the partial index $I_m(h, \vec{\lambda}_{-m})$ of state $h \in \mathcal{S}$ for action $m \in \mathcal{M}$ is defined as the supremum of costs λ'_m such that h is not in the passive set $\mathcal{P}_m(\vec{\lambda}')$, i.e.,

$$I_m\left(h, \vec{\lambda}_{-m}\right) \triangleq \left[\sup\left\{\lambda'_m \mid Q^*(h, m, \vec{\lambda}') \leq Q^*(h, u, \vec{\lambda}'), \forall u\right\}\right]^+. \tag{9}$$

We also define the partial index for the passive action $u = 0$ as $I_0\left(h, \vec{\lambda}\right) \triangleq \left[\min_{m \in \mathcal{M}} Q^*(h, m, \vec{\lambda}) - Q^*(h, 0, \vec{\lambda})\right]^-$.

Note that the partial index is independent of the number C of channel instances per type, because it is derived from the per-agent MDP (6). In [19], the authors proposed the Sum Weighted Index Matching (SWIM) algorithm, which assigns sources to all channel instances according to a max-weight matching (MWM) based on the partial indices. This algorithm not only respects the "hard" constraint (1), but is also asymptotically optimal. However, a major bottleneck

in the SWIM policy is how to compute the partial index efficiently, as the dual costs $\vec{\lambda}$ change constantly. As we discussed in the Introduction, a brute-force approach for computing the partial index is through a binary search on the value of λ'_m. This approach involves repeated solutions to the value functions (7), which also involves high complexity. Thus, the main goal of our paper is to design a fast algorithm to compute the partial index.

2.3 The More-General Setting

Before we proceed, we note that the AoI minimization problem in Section 2.1 can be seen as a special case of a general multi-agent MDP. We can simply replace each source n by an agent, replace its AoI $h_n(t)$, action $u_n(t)$, and AoI cost $h_n(t)$ by a general state $s_n(t)$, action $u_n(t)$, and cost-function $c_n(s_n, u_n)$.

Further, we can replace the transition probabilities $P_m^n(h_n, h'_n)$ in (2) and (3) by more general state transition probabilities. We can then formulate the multi-agent MDP that minimizes the total discounted cost subject to the constraints (1a) and (1b). This multi-agent MDP can again be decomposed using the approach in Section II.B, resulting into per-agent MDP in the form of (omitting the

agent index n):

$$\begin{aligned} &\min_{\pi} \sum_{t=0}^{\infty} \beta^t \mathbb{E}^{\pi}_{s(0)} \left[c(s(t), u(t)) + \sum_{m \in \mathcal{M}} \lambda_m 1_{\{u^{\pi}(t)=m\}} \right] \\ &\text{s.t.} \sum_{m \in \mathcal{M}} 1_{\{u^{\pi}(t)=m\}} \leq 1, \quad \forall t = 1, 2, ... \end{aligned} \tag{10}$$

The partial index can then be defined analogously based on this per-agent MDP. We will assume that partial indexability holds for this per-agent MDP (as stated in Assumption 1 below), and our goal is that the fast partial-index computation algorithm that we develop in this paper will work not only for the AoI minimization problem, but also for other problems with similar structures.

Assumption 1. *We assume that the problem (10) satisfies the partial indexability.*

However, as we mentioned in the Introduction, fast computation is inherently more difficult to develop for partial index than for Whittle index. Thus, in the rest of the paper we will identify additional structural conditions on the types of problems that we can solve. In fact, the partial indexability of the AoI minimization problem in Section 2.1 was shown in [19] based on also some additional structural conditions. Therefore, below we will first state similar conditions for our general model.

Condition 1. *(1) There is a partial order $<$ on the state space $\mathcal{S}$ of each agent n. When $s_1 < s_2$, we say that state s_2 is "higher" than state s_1, and state s_1 is "lower" than state s_2;*

(2) There is a total order $<$ on the action set $\mathcal{U}$ of each agent n. When $u_1 < u_2$, we say that action u_2 is "stronger" than action u_1, and action u_1 is "weaker" than action u_2. We use $u_1 \leq u_2$ if either $u_1 < u_2$ or $u_1 = u_2$;

(3) For any given $\vec{\lambda}$, the optimal policy π^* to the per-agent MDP (6) will satisfy: if $s_1 < s_2$, then $\pi^*(s_1) \leq \pi^*(s_2)$.

In other words, part (3) of the condition states that, for higher states, the optimal policy tends to use stronger actions. To better understand Condition 1, let us use the per-agent problem (6) of the generate-at-will AoI setting as an example.

We can simply use the natural order of the AoI to order the states. Further, we can order the actions by their success probabilities. That is, assume without loss of generality that the channel types are numbered such that $p^n_1 < p^n_2 < ... < p^n_M$. Then, we can again use the natural order of these channel-type numbers to order the actions. Finally, part (3) of Condition 1 is then equivalent to the MTT (Multi-Threshold-Type) property shown in Lemma 4.2 of [19].

However, Condition 1 and Assumption 1 are still insufficient for developing a fast partial-index computation algorithm. Next, we will identify additional structural conditions that will enable fast computation.

3 Towards fast partial-index computation

In this section, we study the general model in Section 2.3. We will focus on the per-agent MDP (10) for a given agent n and develop a fast method for computing the partial index for each possible action and state. For ease of exposition, we will again drop the agent index n throughout Sections III and IV. According to the definition of the partial index (9), next we will fix a possible action m and the dual costs λ_{-m} for all actions other than m, and study the structures of the per-agent MDP as λ_m varies.

3.1 Geometric Structure of the Value Function

Let Π denote the set of all feasible policies. Based on part (3) of Condition 1, we define Π' as the subset of Π that contains all the policies π such that $\pi(s_1) \leq \pi(s_2)$ if $s_1 < s_2$. We call Π' the set of *potential optimal policies*. Given a policy π and the initial state $s(0) = 1$, we can write its value function as

$$\begin{aligned} V^{\pi}(s(0), \vec{\lambda}) &= \sum_{t=0}^{\infty} \beta^t \mathbb{E}^{\pi}_{s(0)} \left[c(s(t), u^{\pi}(t)) + \sum_{m=1}^{M} \lambda_m 1_{\{u^{\pi}(t)=m\}} \right] \\ &= \sum_{t=0}^{\infty} \mathbb{E}^{\pi}_{s(0)} \left[\beta^t \left(c(s(t), u^{\pi}(t)) + \sum_{k \neq m} \lambda_k 1_{\{u^{\pi}(t)=k\}} \right) \right] \\ &\quad + \lambda_m \mathbb{E}^{\pi}_{s(0)} \left[\sum_{t=0}^{\infty} \beta^t 1_{\{u^{\pi}(t)=m\}} \right] \\ &= D^{\pi}_m + \lambda_m T^{\pi}_m \end{aligned}$$

where $D^{\pi}_m \triangleq \sum_{t=0}^{\infty} \mathbb{E}^{\pi}_{s(0)} [\beta^t (c(s(t), u^{\pi}(t)) + \sum_{k \neq m} \lambda_k 1_{\{u^{\pi}(t)=k\}})]$ and $T^{\pi}_m \triangleq \mathbb{E}^{\pi}_{s(0)} \left[\sum_{t=0}^{\infty} \beta^t 1_{\{u^{\pi}(t)=m\}} \right]$.

Note that both D^{π}_m and T^{π}_m are independent of λ_m. Therefore, the value function can be regarded as a linear function of λ_m, with the slope T^{π}_m. In particular, T^{π}_m can be interpreted as the total discounted sum of the number of times that the policy π uses action m. Thus, we refer to it as the *active time* of policy π for action m.

Thanks to Condition 1, the optimal value function satisfies

$$V^*(s, \vec{\lambda}) = \min_{\pi \in \Pi'} V^{\pi}(s, \vec{\lambda}). \tag{11}$$

In other words, V^* is the point-wise minimum of a finite number of linear and non-decreasing functions. We then immediately obtain the following lemma.

Lemma 1. *$V^*(s, \vec{\lambda})$ is a continuous, piece-wise linear, non-decreasing, and concave function in λ_m.*

This structure is illustrated in Fig. 1. Each of the linear pieces corresponds to a different policy attaining the minimum in (11). We will refer to them as *supporting optimal policies.*

In Fig. 1, suppose that there are in total $L + 1$ supporting optimal policies. Let $\tilde{\pi}_i$ be the i-th supporting optimal policy for $\lambda_m \in [\tilde{\lambda}_{i-1}, \tilde{\lambda}_i]$, $i = 1, 2, ..., L, L + 1$, where $\tilde{\lambda}_0 = 0$ and $\tilde{\lambda}_{L+1} = +\infty$.

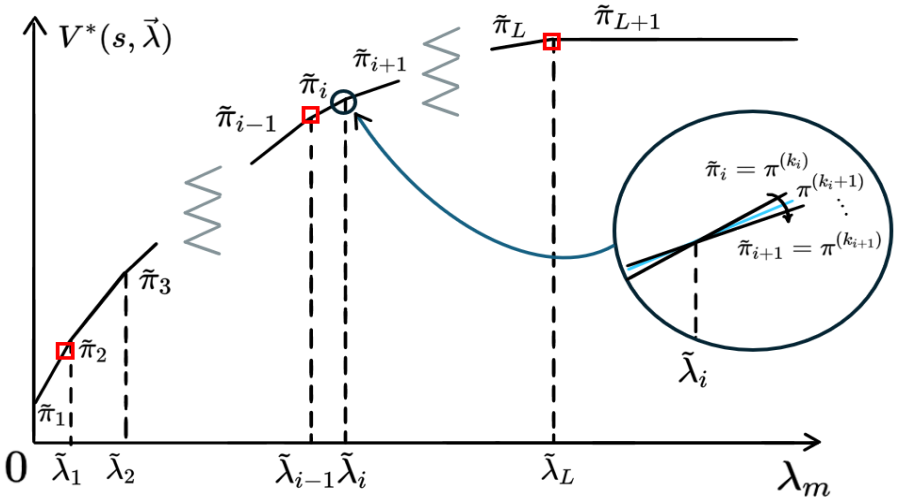


Figure 1: An illustration of the supporting optimal policy

Among them, the last supporting optimal policy $\tilde{\pi}_{L+1}$ corresponds to a passive set equal to the whole state space $\mathcal{S}$. This happens when λ_m is very large, and no optimal policies will use the channel m.

Recall Assumption 1 that partial indexability holds. Thus, once the passive set becomes $\mathcal{S}$ at $\lambda_m = \tilde{\lambda}_L$, at larger λ_m, the passive set will still be $\mathcal{S}$. As a result, the active time will be zero, and the value function will correspond to a line parallel to the x-axis for $\lambda_m \geq \tilde{\lambda}_L$. In contrast, the passive sets of $\tilde{\pi}_1, ...\tilde{\pi}_L$ are strict subsets of $\mathcal{S}$. The corresponding slopes (i.e., active times) are positive and decreasing in i as stated in Lemma 1. Among the transition points $\tilde{\lambda}_1, ..., \tilde{\lambda}_L$, some of them (the squares in Fig. 1) correspond to a change/expansion of the passive set for action m (while others do not). Each such λ_m is then the partial index for the state that was newly added to the passive set.

We note that a geometric structure similar to Fig. 1 has also been used in [3] to derive the fast Whittle-index computation algorithm. The idea in [3] is to find the passive sets for these supporting optimal policies one after another, which will then produce all the index values. However, for partial index, there arises a significant difficulty. That is, unlike the situation for the Whittle index where each passive set corresponds to a unique policy, in Fig. 1 the passive set for action m alone does not determine a policy anymore. Indeed, between the two squares in Fig. 1, the passive set for action m does not change, but the supporting optimal policies do change: they change in the states that use actions other than m. Since in the worst case there are exponentially many possibilities for the states that use actions other than m, we no longer have an efficient way to search for the supporting optimal policies. Clearly, new solutions are needed.

3.2 Additional Structural Conditions

To overcome the above difficulty, our key idea below is to introduce a new "ordering" of the policies (and the corresponding structural conditions), so that we can still search for the supporting optimal policies efficiently. Towards this end, we first define an *operation* on policy.

Definition 4. (Operation on a policy) For any $\pi \in \Pi'$, the operation ψ takes the policy π, changes its action at just one state $\bar{s}$, and produces another policy $\psi(\pi)$.

Since we want to search for the supporting optimal policies in Fig. 1 one-by-one, we want ψ to produce a new policy $\psi(\pi)$ that is to the right of π, i.e., with a smaller active time. This is the concept of *valid operation*.

Definition 5. (Valid operation on a policy) For any $\pi \in \Pi'$, a valid operation ψ on π is an operation such that $T_m^{\psi(\pi)} < T_m^{\pi}$.

Remark: By the above definition, the valid operation ψ must always operate on a potentially-optimal policy $\pi \in \Pi'$, but it may produce a policy outside Π'.

Note that there will be many valid operations, but it suffices for our algorithm to use only a subset. Let Ψ denote a subset of $\{(\psi, \pi) \mid \psi \text{ is a valid operation on } \pi\}$. If $(\psi, \pi) \in \Psi$, we will say that ψ is a Ψ-*valid operation* on π. Given such a set Ψ of valid operations, we can then obtain a partial order of the policies.

Definition 6. For two policies $\pi_1, \pi_2 \in \Pi'$, we say $\pi_1 \prec_\Psi \pi_2$, if π_2 can be obtained from π_1 by performing a sequence of Ψ-valid operations. When $\pi_1 \prec_\Psi \pi_2$, we will say that π_1 is "before (earlier than)" π_2, and π_2 is "after (later than)" π_1. Whenever Ψ is clear from the context, we simply write $\pi_1 \prec \pi_2$.

In other words, a Ψ-valid operation can transform a given policy π to another policy "after" it. Thus, by applying Ψ-valid operations repeatedly to various policies $\pi \in \Pi'$, we can get a partial order of these policies.

Remark 1. For example, later in Section 4, we will use the following set Ψ, which contains all valid operations of the following form: for each $(\psi, \pi) \in \Psi$, ψ will take one state $\bar{s}$, and change the action for this state by one of the following:

(i) if $\pi(\bar{s}) < m$, then $\pi(\bar{s}) < \psi(\pi)(\bar{s}) < m$;

(ii) if $\pi(\bar{s}) > m$, then $\pi(\bar{s}) > \psi(\pi)(\bar{s}) > m$;

(iii) if $\pi(\bar{s}) = m$, then $\psi(\pi)(\bar{s}) \neq m$ and $\psi(\pi) \in \Pi'$.

This set Ψ will again completely determine our partial order of the policies in the generate-at-will case. We will show in Section 4 that ψ is indeed a valid operation on π, i.e., the new policy $\psi(\pi)$ will have a smaller active time than π (see Theorem 6 and Theorem 7).

Our fast computation algorithm (presented soon in Section 3.3) will, from the current supporting optimal policy π, search for the next supporting optimal policy among policies of the form $\psi(\pi)$, where $(\psi, \pi) \in \Psi$. However, at this point, it is unclear why this strategy will work. First, even though the supporting optimal policy on the right in Fig. 1 is known to have a smaller active time, it is unclear why it can be obtained through a sequence of Ψ-valid operations from the previous supporting policy. Second, even if there exists such a sequence of Ψ-valid operations from one supporting optimal policy

to another, finding this sequence with low complexity may not be easy. Thus, it is important to introduce additional conditions below.

Some of these conditions use the following concept of 'cross-over'.

Definition 7. A policy π is a cross-over between π_1 and π_2 if, for all states s, either $\pi(s) = \pi_1(s)$ or $\pi(s) = \pi_2(s)$.

Lemma 2. *Suppose that π_1 and π_2 are both optimal at λ_m. Then, any cross-over policies between π_1 and π_2 are also optimal at the same λ_m.*

The intuition for Lemma 2 is quite straight-forward. Since both π_1 and π_2 are optimal, they must have the same value function. Further, both $\pi_1(s)$ and $\pi_2(s)$ must satisfy the Bellman equation with the common value function. Since any cross-over policy uses either $\pi_1(s)$ and $\pi_2(s)$, it must also satisfy the Bellman equation and is thus optimal. Detailed proof can be found in Appendix A.

We can now state our additional conditions on the set Ψ of valid operations.

Assumption 2. *For any policy $\pi \in \Pi'$ such that $\pi(\bar{s}) = m$ for some state $\bar{s}$, there must exist a Ψ-valid operation ψ on π such that $\psi(\pi) \in \Pi'$. Moreover, if an operation ψ on such policy π satisfies: (1) $\psi(\pi)(\bar{s}) \neq \pi(\bar{s})$; and (2) $\psi(\pi) \in \Pi'$, then it must be a Ψ-valid operation.*

Assumption 3. *For any two policies $\pi_1 \prec \pi_2 \in \Pi'$, there must exist a Ψ-valid operation ψ on π_1, such that $\psi(\pi_1)$ is a cross-over between π_1 and π_2.*

Assumption 4. *Fix $\vec{\lambda}_{-m}$. Suppose that two policies π_1 and π_2 are optimal when $\lambda_m = \mu_1$ and $\lambda_m = \mu_2$ ($\mu_1 < \mu_2$), respectively. Further, suppose that there exists some state s such that $\pi_1(s) = \pi_2(s) = m$. If $\pi_1 \prec \pi_2$ does not hold (i.e., either $\pi_2 \prec \pi_1$, or π_1 and π_2 are not comparable), then there exists a Ψ-valid operation ψ on π_2 such that $\psi(\pi_2)$ is a cross-over between π_1 and π_2.*

Roughly speaking, Assumption 4 addresses the first difficulty and ensures that one supporting optimal policy can always reach another supporting optimal policies on the right of Fig. 1 through a sequence of Ψ-valid operations (see Theorem 3). Assumption 3 further ensures that we can find this sequence of Ψ-valid operations through a greedy procedure (see Theorem 4). Finally, Assumption 2 covers a few corner cases in Theorem 4.

Next, we will develop our fast computation algorithm based on these assumptions.

3.3 The Proposed Algorithm

The first important consequence of the above assumptions is the following theorem, which states that the sequence of supporting optimal policies must follow the partial order defined by the set of valid operations in Ψ.

Theorem 3. *Suppose that Assumptions 1 and 4 hold. Then, we must have $\tilde{\pi}_i \prec \tilde{\pi}_{i+1}$ for $i = 1, 2, ..., L-1$.*

Proof. We prove this by contradiction. Recall that $\tilde{\pi}_i$ and $\tilde{\pi}_{i+1}$ are both optimal policies. Further, $\tilde{\pi}_i$ is optimal when $\lambda_m = \mu_1 \in [\tilde{\lambda}_{i-1}, \tilde{\lambda}_i]$, $\tilde{\pi}_{i+1}$ is optimal when $\lambda_m = \mu_2 \in (\tilde{\lambda}_i, \tilde{\lambda}_{i+1}]$, and we have $\mu_1 < \mu_2$. Assume that $\tilde{\pi}_i \prec \tilde{\pi}_{i+1}$ does not hold. Since $i \leq L-1$, the passive set of $\tilde{\pi}_{i+1}$ is not the whole state space $\mathcal{S}$ yet, and thus there exists state s that make $\tilde{\pi}_{i+1}(s) = m$. According to Assumption 1, we must then have $\tilde{\pi}_i(s) = m$. Thus, we can apply Assumption 4, and infer that there exists a Ψ-valid operation ψ on $\tilde{\pi}_{i+1}$ such that $\psi(\tilde{\pi}_{i+1})$ is a cross-over between $\tilde{\pi}_i$ and $\tilde{\pi}_{i+1}$. As $\tilde{\pi}_i$ and $\tilde{\pi}_{i+1}$ are both optimal at the transition point $\lambda_m = \tilde{\lambda}_i$ (see Fig. 1), according to Lemma 2, $\psi(\tilde{\pi}_{i+1})$ must also be optimal at $\lambda_m = \tilde{\lambda}_i$. Since ψ is a Ψ-valid operation, we must have $\tilde{\pi}_{i+1} \prec \psi(\tilde{\pi}_{i+1})$ and therefore, $T_m^{\psi(\tilde{\pi}_{i+1})} < T_m^{\tilde{\pi}_{i+1}}$. This means that $\tilde{\pi}_i$ and $\psi(\tilde{\pi}_{i+1})$ are both optimal at $\lambda_m = \tilde{\lambda}_i$, but the slope of the line corresponding to $\psi(\tilde{\pi}_{i+1})$ is smaller, which implies that $\tilde{\pi}_{i+1}$ cannot be the supporting optimal policy for $\lambda_m \in (\tilde{\lambda}_i, \tilde{\lambda}_i + \epsilon)$, which is a contradiction. Thus, we must have $\tilde{\pi}_i \prec \tilde{\pi}_{i+1}$. □

Let us denote

$$\eta_{\pi_1,\pi_2} = \frac{D_m^{\pi_2} - D_m^{\pi_1}}{T_m^{\pi_1} - T_m^{\pi_2}} \tag{12}$$

as the abscissa (i.e., x-coordinate) of the intersection point between the lines corresponding to π_1 and π_2 (which is exactly the value of λ_m such that $V^{\pi_1}(s(0), \vec{\lambda}) = V^{\pi_2}(s(0), \vec{\lambda})$.

Define $\Gamma_1(\pi) = \{\psi(\pi) \mid (\psi, \pi) \in \Psi \text{ and } \psi(\pi) \in \Pi'\}$. Inspired by Theorem 3, our proposed algorithm will therefore search for the next supporting optimal policy from $\Gamma_1(\pi)$, where π is the current policy. This idea leads to Algorithm 1 below. During initialization, Line 2 finds the first supporting optimal policy $\tilde{\pi}_1$ (at $\lambda_m = 0$), and Line 3 finds the last supporting optimal policy $\tilde{\pi}_{L+1}$ (at very large λ_m). Line 4-14 iteratively find all of the other supporting optimal policies $\tilde{\pi}_2$ to $\tilde{\pi}_L$. Specifically, Line 5 computes the abscissa values (12) between the current policy π and all policies $\pi' \in \Gamma_1(\pi)$. Line 6 picks the π' with the smallest abscissa value η^*, which will become the next policy π in Line 13. If the new policy changes action at a state $\bar{s}$ whose original action is m (Lines 7 and 8), the passive set for action m has changed, and a new partial index value is then assigned (Lines 9 and 10). Line 12 computes the abscissa value (12) between π and $\tilde{\pi}_{L+1}$ (whose active time is 0). If $\eta^* \geq \eta_{stop}$, the next supporting optimal policy should be $\tilde{\pi}_{L+1}$. The algorithm can terminate and the partial index for all remaining states in the old passive set is assigned to η_{stop} (Line 15). Otherwise, the iteration should continue (Line 14).

Theorem 4. *Suppose that Assumptions 2, 3, and 4 hold. Algorithm 1 can obtain the partial index of all states.*

Proof. We prove this by induction. Before that, let us define some additional notations. For the k-th iteration of Lines 4-14, we denote $\pi^{(k)}$ as the new policy $\psi^*(\pi)$ obtained in Line 6, $\eta^{(k)}$ as the η^* obtained in Line 6, and $\eta_{stop}^{(k)}$ as the value of η_{stop} in Line 12. Our induction hypothesis is: for each $i = 1, ..., L$, there exists an iteration number k_i of Lines 4-14 such that at the k_i-th iteration, the algorithm can output $\tilde{\pi}_i$ with $\eta^{(k_i)} = \tilde{\lambda}_{i-1}$.

The base step $i = 1$ is obvious because the initial policy found by the algorithm (with $\lambda_m = 0$) must be $\tilde{\pi}_1$ and we have $\tilde{\lambda}_0 = 0$ by default. Now we prove the induction step, which we state as a

Algorithm 1

Input: m, $\vec{\lambda}_{-m}$, $\mathcal{S}$; **Output:** $I_m\left(s, \vec{\lambda}_{-m}\right)$ for all state $s \in \mathcal{S}$;

Initialization: Given $\vec{\lambda}_{-m}$ and $\lambda_m = 0$, find the optimal policy π with the smallest active time T_m^{π}, get the corresponding passive set $\mathcal{P}_m(\vec{\lambda})$ and let $I_m\left(s, \vec{\lambda}_{-m}\right) = 0$ for $s \in \mathcal{P}_m(\vec{\lambda})$;

Given the same $\vec{\lambda}_{-m}$, find a large enough λ_m that can make $\mathcal{P}_m(\vec{\lambda}) = \mathcal{S}$ and compute the optimal value function $V^*(s_0, \vec{\lambda}) = D_m^{\tilde{\pi}_{L+1}}$;

repeat

  Compute $\eta_{\pi,\pi'} = \frac{D_m^{\pi'} - D_m^{\pi}}{T_m^{\pi} - T_m^{\pi'}}$ for all $\pi' \in \Gamma_1(\pi)$;

  Compute $\eta^* = \min_{\pi' \in \Gamma_1(\pi)} \{\eta_{\pi,\pi'}\}$ and choose one corresponding $\psi^*(\pi) \in \operatorname{argmin}_{\pi' \in \Gamma_1(\pi)} \{\eta_{\pi,\pi'}\}$;

  Check the state $\bar{s}$ that $\pi(\bar{s})$ and $\psi^*(\pi)(\bar{s})$ have different actions;

  if $\pi(\bar{s}) = m$ **then**

    $I_m\left(\bar{s}, \vec{\lambda}_{-m}\right) = \eta^*$;

    $\mathcal{P}_m(\vec{\lambda}) \longleftarrow \mathcal{P}_m(\vec{\lambda}) \cup \{\bar{s}\}$;

  end if

  Compute $\eta_{stop} = \frac{D_m^{\tilde{\pi}_{L+1}} - D_m^{\pi}}{T_m^{\pi}}$;

  $\pi \longleftarrow \psi^*(\pi)$;

until $\eta^* \geq \eta_{stop}$

For all the state $s \in \mathcal{S} \backslash \mathcal{P}_m(\vec{\lambda})$, let $I_m\left(s, \vec{\lambda}_{-m}\right) = \eta_{stop}$.

lemma below.

Lemma 5. *Assume that for $i \leq L-1$ the algorithm can output $\tilde{\pi}_i$ after the k_i-th iteration and $\eta^{(k_i)} = \tilde{\lambda}_{i-1}$. Then, through a finite number of iterations, i.e., after the k_{i+1}-th iteration, the algorithm can output $\tilde{\pi}_{i+1}$ with $\eta^{(k_{i+1})} = \tilde{\lambda}_i$.*

Based on Lemma 5, as long as the algorithm does not terminate, i.e., $\eta^{(k)} < \eta^{(k)}_{stop}$, the algorithm will continue producing the next supporting optimal policy from $\tilde{\pi}_2$ to $\tilde{\pi}_L$. We will later show that, after the algorithm produces $\tilde{\pi}_L$, it must get $\eta^{(k)} = \eta^{(k)}_{stop}$ in the next iteration. The algorithm will then terminate. This would complete the proof of Theorem 4.

Proof of Lemma 5. We use another induction and the induction hypothesis is: for each iteration k such that $k \geq k_i$ and $k \leq k_{i+1}$, (1) $\pi^{(k)}$ is optimal at $\lambda_m = \tilde{\lambda}_i$ and there is at least a common state s that makes $\pi^{(k)}(s) = \tilde{\pi}_{i+1}(s) = m$; (2) either the policy $\tilde{\pi}_{i+1}$ has already been outputted, or we must have $\pi^{(k)} \in \Pi'$ and $\pi^{(k)} \prec \tilde{\pi}_{i+1}$; and (3) $\eta^{(k)} < \eta^{(k)}_{stop}$.

To see why this induction hypothesis implies Lemma 5, note that if the iteration k of Algorithm 1 returns $\tilde{\pi}_{i+1}$, then $k_{i+1} = k$ and the result of Lemma 5 holds trivially. If not, part (2) of our induction hypothesis ensures that $\pi^{(k+1)}$ is still earlier than $\tilde{\pi}_{i+1}$. Since there are only a finite number of policies that are after $\tilde{\pi}_i$ and before $\tilde{\pi}_{i+1}$, eventually Algorithm 1 must output $\tilde{\pi}_{i+1}$. (See the inlet of Fig. 1 for illustration.)

To prove our induction hypothesis, we skip the base step ($k = k_i$) as it follows from the assumption of the lemma. Next, we prove the induction step. Assume that $\pi^{(k)}$ is optimal at $\lambda_m = \tilde{\lambda}_i$ (which implies that $\pi^{(k)} \in \Pi'$), and $\pi^{(k)} \prec \tilde{\pi}_{i+1}$. We wish to show that our induction hypothesis also holds for $k+1$. We first verify part (1) of the induction hypothesis. Towards this end, we first show that the lines corresponding to $\pi^{(k+1)}$ and $\pi^{(k)}$ must intersect at $\tilde{\lambda}_i$. According to Assumption 3, as $\pi^{(k)} \prec \tilde{\pi}_{i+1}$, there exists a Ψ-valid operation ψ_1 on $\pi^{(k)}$ such that $\psi_1(\pi^{(k)})$ is a cross-over policy between $\pi^{(k)}$ and $\tilde{\pi}_{i+1}$. Then, by Lemma 2, since $\pi^{(k)}$ and $\tilde{\pi}_{i+1}$ are both optimal at $\lambda_m = \tilde{\lambda}_i$, $\psi_1(\pi^{(k)})$ must also be optimal at $\lambda_m = \tilde{\lambda}_i$. Thus, it implies that $\psi_1(\pi^{(k)}) \in \Pi'$ and further, $\psi_1(\pi^{(k)}) \in \Gamma_1(\pi^{(k)})$. Therefore, $\Gamma_1(\pi^{(k)})$ cannot be empty. Note that by Lines 5 and 6, we must have $\pi^{(k+1)} \in \Pi'$. According to the analysis above, we have shown that $\psi_1(\pi^{(k)})$ is optimal at $\lambda_m = \tilde{\lambda}_i$, and therefore, the line corresponding to $\psi_1(\pi^{(k)})$ will intersect with the lines corresponding to $\pi^{(k)}$ at $\lambda_m = \tilde{\lambda}_i$. Then, since in Line 6 we pick the policy with the smallest abscissa value, we must have $\eta^{(k+1)} = \min_{\pi' \in \Gamma_1(\pi^{(k)})}\{\eta_{\pi^{(k)},\pi'}\} \leq \eta_{\pi^{(k)},\psi_1(\pi^{(k)})} = \tilde{\lambda}_i$. It only remains to show that $\eta^{(k+1)} < \tilde{\lambda}_i$ cannot happen. We prove this by contradiction. Assume that $\eta^{(k+1)} < \tilde{\lambda}_i$, which implies that there is another policy $\psi^*(\pi^{(k)})$ in $\Gamma_1(\pi^{(k)})$ whose line intersects with the line corresponding to $\pi^{(k)}$ at $\lambda_m = \eta^{(k+1)} < \tilde{\lambda}_i$. Then, since $T_m^{\psi^*(\pi^{(k)})} < T_m^{\pi^{(k)}}$ due to $\pi^{(k)} \prec \psi^*(\pi^{(k)})$, that will make $\pi^{(k)}$ no longer the optimal policy at $\lambda_m = \tilde{\lambda}_i$, which is a contradiction to our assumption. Therefore, $\eta^{(k+1)}$ must be equal to $\tilde{\lambda}_i$, and the lines corresponding to $\pi^{(k+1)}$, $\pi^{(k)}$ and $\tilde{\pi}_{i+1}$ must all intersect at $\lambda_m = \tilde{\lambda}_i$. Hence, $\pi^{(k+1)}$ is also optimal at $\lambda_m = \tilde{\lambda}_i$.

Next, we show that there is at least a common state s that makes $\pi^{(k+1)}(s) = \tilde{\pi}_{i+1}(s) = m$. We prove it by contradiction. Notice that according to our induction hypothesis for $\pi^{(k)}$, there is a common state $\bar{s}$ such that $\pi^{(k)}(\bar{s}) = \tilde{\pi}_{i+1}(\bar{s}) = m$. Assume on the contrary that $\pi^{(k+1)}(\bar{s}) \neq m$. We can construct a new policy π' such that $\pi'(\bar{s}) = \pi^{(k+1)}(\bar{s})$ and $\pi'(s) = \tilde{\pi}_{i+1}(s)$ for $s \neq \bar{s}$. It implies that π' is a cross-over policy between $\pi^{(k+1)}$ and $\tilde{\pi}_{i+1}$. Because $\pi^{(k+1)}$ and $\tilde{\pi}_{i+1}$ are both optimal at $\lambda_m = \tilde{\lambda}_i$, π' is also optimal at $\lambda_m = \tilde{\lambda}_i$ and hence, $\pi' \in \Pi'$. Consider the operation $\psi_1(\tilde{\pi}_{i+1}) = \pi'$. According to Assumption 2, ψ_1 must be a Ψ-valid operation. Then, the active time of $\psi_1(\tilde{\pi}_{i+1})$ will be smaller than $\tilde{\pi}_{i+1}$, which implies that $\tilde{\pi}_{i+1}$ cannot be the next supporting optimal policy, which is a contradiction! Therefore, we conclude that there is at least a common state $\bar{s}$ that makes $\pi^{(k+1)}(\bar{s}) = \tilde{\pi}_{i+1}(\bar{s}) = m$. Part (1) of the induction step for $i+1$ is then verified.

To verify part (2) of the induction step, if $\pi^{(k+1)} = \tilde{\pi}_{i+1}$, then $\tilde{\pi}_{i+1}$ is outputted and the induction step is trivially complete. Further, from our earlier proof for part (1) of the induction step, we have shown that $\eta^{(k+1)} = \tilde{\lambda}_i$. If $\tilde{\pi}_{i+1}$ is not outputted, we now show $\pi^{(k+1)} \prec \tilde{\pi}_{i+1}$ by contradiction. Assume on the contrary that $\pi^{(k+1)} \prec \tilde{\pi}_{i+1}$ is not true. We already know from part (1) of the induction step that there is at least a common state s that makes $\pi^{(k+1)}(s) = \tilde{\pi}_{i+1}(s) = m$. Further, as shown in the proof of part (1) above, $\pi^{(k+1)}$ is optimal when $\lambda_m = \mu_1 = \tilde{\lambda}_i$, $\tilde{\pi}_{i+1}$ is optimal when $\lambda_m = \mu_2 \in (\tilde{\lambda}_i, \tilde{\lambda}_{i+1}]$, and we have $\mu_1 < \mu_2$. According to Assumption 4, there must be a Ψ-valid operation ψ on $\tilde{\pi}_{i+1}$ such that $\psi(\tilde{\pi}_{i+1})$ is a cross-over between $\pi^{(k+1)}$ and $\tilde{\pi}_{i+1}$. As $\pi^{(k+1)}$ and $\tilde{\pi}_{i+1}$ are both optimal at $\lambda_m = \tilde{\lambda}_i$, $\psi(\tilde{\pi}_{i+1})$ must also be optimal at $\lambda_m = \tilde{\lambda}_i$. However, $\tilde{\pi}_{i+1} \prec \psi(\tilde{\pi}_{i+1})$ leads to $T_m^{\psi(\tilde{\pi}_{i+1})} < T_m^{\tilde{\pi}_{i+1}}$, which implies that $\tilde{\pi}_{i+1}$ is no longer the supporting optimal policy when $\lambda_m \in (\tilde{\lambda}_i, \tilde{\lambda}_i + \epsilon)$, which is a contradiction! Therefore, we must have $\pi^{(k+1)} \prec \tilde{\pi}_{i+1}$.

Finally, we prove part (3) of the induction hypothesis, i.e., $\eta^{(k+1)} < \eta_{stop}^{(k+1)} = \frac{D_m^{\tilde{\pi}_{L+1}} - D_m^{\pi^{(k)}}}{T_m^{\pi^{(k)}}}$. We prove by contradiction. Assume on the contrary that $\eta^{(k+1)} \geq \eta_{stop}^{(k+1)}$. As we have already known from part (1) of the induction step that $\eta^{(k+1)} = \tilde{\lambda}_i$, we must then have $\eta_{stop}^{(k+1)} \leq \tilde{\lambda}_i$. Further, we have shown that $\tilde{\lambda}_i$ is the x-oordinate of the intersection point between the lines corresponding to $\pi^{(k)}$ and $\tilde{\pi}_{i+1}$ ($i < L$). Meanwhile, $\eta_{stop}^{(k+1)}$ is the x-coordinate of the intersection point between the lines corresponding to $\pi^{(k)}$ and $\tilde{\pi}_{L+1}$. Further, the line corresponding to $\tilde{\pi}_{L+1}$ is parallel to the x-axis, which has the smallest slope of 0, while the line corresponding to $\tilde{\pi}_{i+1}$ has a slope greater than 0. Since we have $\eta_{stop}^{(k+1)} \leq \tilde{\lambda}_i$, it then implies that $\tilde{\pi}_{L+1}$ will be the next supporting optimal policy on $\lambda_m \in (\eta_{stop}^{(k+1)}, +\infty)$. In other words, $\tilde{\pi}_{i+1}$ cannot be the next supporting optimal policy on $\lambda_m \in (\eta_{stop}^{(k+1)}, \tilde{\lambda}_i) \subset (\eta_{stop}^{(k+1)}, +\infty)$, which is a contradiction! The result of Lemma 5 then follows.

□

We can now return to the proof of Theorem 4. Thanks to Lemma 5, we can conclude that our algorithm will produce $\tilde{\pi}_1, \tilde{\pi}_2, ..., \tilde{\pi}_L$ and obtain $\tilde{\lambda}_1, \tilde{\lambda}_2, ..., \tilde{\lambda}_{L-1}$. It only remains to show that our algorithm can also obtain $\tilde{\lambda}_L$.

Now, we suppose that during the q-th iteration, we obtain $\pi^{(q)} = \tilde{\pi}_L$. We will prove that after the next iteration, we must have $\eta^{(q+1)} \geq \eta_{stop}^{(q+1)}$. In other words, the algorithm will stop. Towards this

end, note that $\tilde{\pi}_L \in \Pi'$ and the passive set is not the whole state space yet. Thus, there is some state s that makes $\tilde{\pi}_L(s) = m$. According to Assumption 2, there exists at least one Ψ-valid operation ψ on $\pi^{(q)}$ and $\psi(\pi^{(q)}) \in \Pi'$. Hence, we can continue to get $\eta^{(q+1)}$ during the next $(q+1)$-th iteration. We now show that $\eta^{(q+1)} \geq \eta_{stop}^{(q+1)}$. Towards this end, we first show that $\eta_{stop}^{(q+1)} = \tilde{\lambda}_L$. To see this, note that according to Line 12, we have $\eta_{stop}^{(q+1)} = \frac{D_m^{\tilde{\pi}_{L+1}} - D_m^{\pi^{(q)}}}{T_m^{\pi^{(q)}}}$. Since $\pi^{(q)} = \tilde{\pi}_L$ and $T_m^{\tilde{\pi}_{L+1}} = 0$, we must have $\eta_{stop}^{(q+1)} = \frac{D_m^{\tilde{\pi}_{L+1}} - D_m^{\tilde{\pi}_L}}{T_m^{\tilde{\pi}_L} - T_m^{\tilde{\pi}_{L+1}}} = \tilde{\lambda}_L$, where the last equality follows from the definition of $\tilde{\lambda}_L$, i.e., $\tilde{\lambda}_L$ is the x-coordinate of the intersection point between the lines corresponding to $\tilde{\pi}_L$ and $\tilde{\pi}_{L+1}$. It then only remains to prove that $\eta^{(q+1)} \geq \tilde{\lambda}_L$. We prove this by contradiction. Assume on the contrary that $\eta^{(q+1)} < \tilde{\lambda}_L$. As $\pi^{(q+1)}$ is obtained by performing a Ψ-valid operation from $\pi^{(q)}$ (see the definition of Γ_1), we must have $T_m^{\pi^{(q+1)}} < T_m^{\pi^{(q)}} = T_m^{\tilde{\pi}_L}$. In other words, the line corresponding to $\pi^{(q)} = \tilde{\pi}_L$ intersects with a line with a smaller slope (that corresponds to $\pi^{(q+1)}$) at a point $\eta^{(q+1)}$ earlier than $\tilde{\pi}_L$. It implies that the line corresponding to $\pi^{(q+1)}$ must be lower than the line corresponding to $\pi^{(q)} = \tilde{\pi}_L$ in the interval $\lambda_m \in (\eta^{(q+1)}, \tilde{\lambda}_L]$. This contradicts our assumption that $\tilde{\pi}_L$ is the supporting optimal policy when $\lambda_m \in [\tilde{\lambda}_{L-1}, \tilde{\lambda}_L]$. As a result, we must have $\eta^{(q+1)} \geq \tilde{\lambda}_L = \eta_{stop}^{(q+1)}$. In summary, not only can we obtain $\tilde{\lambda}_L$, but also the algorithm will exit the loop from Line 3 to Line 13 after the $(q+1)$-th iteration and go to Line 14.

From Line 14, the partial index $I_m(s, \vec{\lambda}_{-m})$ for all the states $s \in \mathcal{S}/\mathcal{P}_m(\vec{\lambda})$ will be assigned to $\eta_{stop}^{(q+1)} = \tilde{\lambda}_L$. This assignment is correct because when $\lambda_m < \tilde{\lambda}_L$, the supporting optimal policy $\tilde{\pi}_L$ will choose action m for those states, but when $\lambda_m > \tilde{\lambda}_L$, the optimal policy becomes $\tilde{\pi}_{L+1}$ and it chooses action m for no state at all. In summary, our algorithm can obtain the correct partial index for all of the states.

□

3.4 Complexity of the Algorithm

To find the complexity of Algorithm 1, note that the most expensive part of the iteration in Lines 4-14 is Line 5, where we need to compute $D_m^{\pi'}$ and $T_m^{\pi'}$ for every $\pi' \in \Gamma_1(\pi)$. Fortunately, using the fact that π' and π differ in the decision at only one state, there is an efficient formula that can compute these expressions from D_m^{π} and T_m^{π} with $\mathcal{O}(K^2)$ complexity (see [3], Section 4). Then, let A upper-bound the total number of iterations of Lines 4-14, and let B upper-bound the size of $\Gamma_1(\pi)$. The total complexity of all iterations is then $\mathcal{O}(ABK^2)$. The initialization step can be solved by linear programming, with complexity $\mathcal{O}(K^3)$. Finally, since we have M possible actions, the total complexity to compute the partial index for all states and all actions is then $\mathcal{O}(M(K^3 + ABK^2))$. Later in Section 4.1, we will show that, for the generate-at-will case, using the proposed algorithm leads to a complexity of only $\mathcal{O}(M^3K^3)$.

4 Return to the generate-at-will AoI problem

Given the results in Section 3, we only need to verify that the generate-at-will AoI minimization problem satisfies all the conditions/assumptions introduced earlier. As we mentioned in Section 2, Assumption 1 and Condition 1 have been verified in [19]. (Note that although [19] studies an average-cost MDP, the analysis can be easily extended to discounted MDP.) Thus, here we will focus on proving Assumptions 2, 3, and 4. But firstly, we need to verify that the operations defined in Remark 1 are valid operations. We have the following two theorems.

Theorem 6. *Consider two policies π and π' in Π' with the following properties: there exists exactly one state $\bar{s}$ such that $\pi(\bar{s}) \neq \pi'(\bar{s})$ and $\pi(\bar{s}) = m$; for all other states $s \neq \bar{s}$, the actions chosen by the two policies are the same. Then, we must have $T_m^{\pi} > T_m^{\pi'}$.*

The intuition of Theorem 6 is that, since π' uses action m at fewer states, its active time for action m should also be smaller. The proof is shown in Appendix B.

Theorem 7. *Consider two policies $\pi \in \Pi'$ and π' with the following properties: there exists exactly one state $\bar{s}$ along with channel u and $u+1$ such that $\pi(\bar{s}) = u$ and $\pi'(\bar{s}) = u+1$. For all other states $s \neq \bar{s}$, the actions chosen by the two policies are the same, i.e., $\pi(s) = \pi'(s)$. The following statements must holds:*

(i) if $u+1 < m$, then $T_m^{\pi} > T_m^{\pi'}$;

(ii) if $u > m$, then $T_m^{\pi} < T_m^{\pi'}$.

Sketch. Next, we will sketch the proof of Theorem 7, part (i). Let us define two Markov chains, chain 0 and chain 1. The two chains have exactly the same initial state $s(0) = 1$ and dual cost of channels $\vec{\lambda}$, but chain 0 follows policy π and chain 1 follows policy π'. This means that their state transition probabilities are also almost the same, except for state $\bar{s}$ where their decisions differ. To compare their active times, we perform the following stochastic coupling on the random transitions. From time 0 onwards,

(1) For each $s \neq \bar{s}$, the l-th transition from state s of chain 1 will have the same channel success (i.e., to state $s_0 = 1$) or failure (i.e., to state $s+1$) event as the l-th transition from state s of chain 0.

(2) If the l-th transition from state $\bar{s}$ of chain 0 has the success event, then the l-th transition from state $\bar{s}$ of chain 1 will also have the success event. If the l-th transition from state $\bar{s}$ of chain 0 has the failure event, then the l-th transition from state $\bar{s}$ of chain 1 will have the success event with probability $\frac{p_{u+1}-p_u}{1-p_u}$ and will have the failure event (transmit to state $\bar{s}+1$) with probability $\frac{1-p_{u+1}}{1-p_u}$. (Recall that we have assumed $p_{u+1} > p_u$ in our channel ordering.)

Define S_+ as the subset of $\mathcal{S}$ that contains all the states that are higher than $\bar{s}$, and define S_- as the subset of $\mathcal{S}$ that contains all the states that are lower than $\bar{s}$. Thus, $\mathcal{S} = \{\bar{s}\} \cup S_+ \cup S_-$. Let $t_+^0(k)$ be the k-th time when chain 0 transitions from $\mathcal{S} \backslash S_+$ to S_+, and $t_-^0(k)$ be the k-th time when chain 0 transitions from $\mathcal{S} \backslash S_-$ to S_-. Define $t_+^1(k)$ and $t_-^1(k)$ similarly for chain 1. We have the following lemma.

Lemma 8. *Under our coupling, we have* $t_+^0(k) \le t_+^1(k)$ *and* $t_-^0(k) \ge t_-^1(k)$ *for all* k.

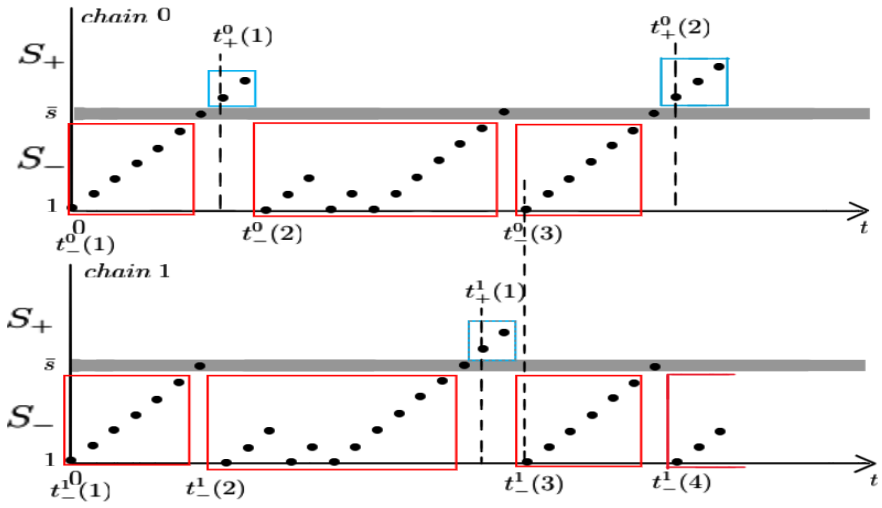


Figure 2: The illustration for lemma 8

The result of the lemma can be understood from Fig. 2, which illustrates an example of the state evolution of the two coupled chains. Each blue rectangle represents a period of time (i.e., an episode) that the chain stays in S_+, and each red rectangle represents an episode that the chain stays in S_-. Note that due to our coupling rule (1), the sequence of blue (correspondingly, red) rectangles/episodes of the two chains must have exactly the same transitions. The only difference is at $\bar{s}$ (the horizontal gray bar), where chain 1 will have a larger probability to go down to state 1 (due to choosing channel $k+1$) than chain 0. As a result, the red episodes of chain 1 tend to appear earlier than that of chain 0, and the blue episodes of chain 1 tend to appear later, which leads to Lemma 8.

To complete the proof sketch of Theorem 7, recall that we are interested in the active time of choosing channel m. Consider first the case $k+1 < m$. This implies that the states using channel m (which are common for both π and π') are in S_+. As we have seen in Fig. 2 (Lemma 8), every episode of chain 0 in S_+ can only appear earlier than that of chain 1, and they have exactly the same sequence of transitions, including the steps that they use channel m. It is then not hard to show that $T_m^\pi > T_m^{\pi'}$. Part (i) of Theorem 7 then follows. The case of $k > m$ can be shown similarly. See the detailed proof in Appendix C. □

Combining Theorems 6 and 7, we conclude that all the operations in Ψ given in Remark 1 are valid operations. Next, we will prove that Ψ given in Remark 1 will satisfy Assumptions 2, 3 and 4. As Assumption 2 can be verified relatively directly, we set it as a corollary. Finally, Assumptions 3 and 4 are verified by Theorem 9 below.

Corollary 1. Ψ *satisfies Assumption 2.*

The proof is provided in Appendix D.

Theorem 9. *Ψ satisfies Assumption 3 and Assumption 4.*

The proof is provided in Appendix E.

4.1 Complexity Analysis and Comparison to Binary Search

From Section 3.4, we only need to analyze the upper bound A on the number of iterations and the upper bound B of the number of operations in $\Gamma_1(\pi)$.

We can show that B is upper-bounded by $2M$ and A is upper-bounded by $MK+1$. The detailed proof is provided in Appendix F.

Based on the upper bounds, the complexity of our algorithm (for all states and all channels) will be

$$\mathcal{O}(M(K^3 + (A-1)BK^2)) = \mathcal{O}(M^3K^3). \tag{13}$$

Comparison with binary search: We now compare the complexity in (13) with a brute-force binary search method. Note that our algorithm computes each partial index precisely. While the binary search can only compute the partial index with some precision of ϵ. Assume that the search range for λ_m is $[C_{min}, C_{max}]$. Then, the total number of binary searches is $\log_2((C_{max} - C_{min})/\epsilon)$. For each step of the binary search, we should calculate the optimal policy for a given λ_m, and see whether the optimal policy chooses channel m for a given state s. We could use either policy iteration or value iteration to do so. Between them, policy iteration is usually of lower complexity, as it can give the exact optimal policy in a finite number of steps. Its complexity is $N^{PI}(\beta)\mathcal{O}(K^\omega + MK^2)$, where $N^{PI}(\beta)$ is the number of the policy iterations and is of the order $\mathcal{O}(MK)$, and ω can be taken as 2.807 by Strassen's algorithm [5, 13]. Putting these together, the complexity of binary search to compute a partial index *for one state* and all channels is $\mathcal{O}(M^2K^{3.807} + M^3K^3)$, which is higher than our complexity (13) to compute the index *for all states* and all channels when K is large.

5 Numerical Results

In this section, we present MATLAB simulation results of our proposed algorithm. We first verify the correctness of our method. We focus on the per-source MDP (6) with $M = 3, K = 8$, and channel success probabilities given by $\vec{p} = [0.3, 0.6, 0.9]$. We compute the partial index of every channel m and state s, and then compare the results by our algorithm and by binary search. For binary search with policy iteration, we set the precision level as $\epsilon = 0.001$, and the search range $[C_{min}, C_{max}] = [0, 100]$ for the dual cost λ_m. In all the experiments, we find that the partial indices computed by our algorithm and by binary search are all within ϵ, which verifies that our algorithm computes the correct partial index. Table I show a representative example for $m = 3$ and $\vec{\lambda} = [1, 1.5, 2]$, where the partial indices obtained by the two algorithms match each other for all states.

Next, we compare the running times of the two algorithms. Note that there are two ways to use our fast computation algorithm. The first is to use it at a *pre-computation* stage. That is, we first

Table 1: The partial index computed by binary search/our method

$I_3(1, \vec{\lambda}_{-3})$	$I_3(2, \vec{\lambda}_{-3})$	$I_3(3, \vec{\lambda}_{-3})$	$I_3(4, \vec{\lambda}_{-3})$
0.782 / 0.783	2.129 / 2.129	2.560 / 2.560	2.907 / 2.908
$I_3(5, \vec{\lambda}_{-3})$	$I_3(6, \vec{\lambda}_{-3})$	$I_3(7, \vec{\lambda}_{-3})$	$I_3(8, \vec{\lambda}_{-3})$
3.259 / 3.259	3.609 / 3.609	3.933 / 3.933	3.933 / 3.933

sample some values for $\vec{\lambda}$ (e.g., in a grid) and pre-calculate the partial index at these $\vec{\lambda}$ values. Then, in real time execution of the SWIM policy, we use interpolation to approximate the partial index for the current $\vec{\lambda}$, which is much faster. For this pre-computation stage to work, it is important to pre-calculate the partial index for *all* states. The second approach is to directly use our algorithm in real time, in which case only the partial index of the *current* state needs to be computed. Note that our algorithm always produces the partial index for *all* states.

In contrast, binary search by default produces only the partial index for *one* state, and it needs to be executed K times to compute the partial indices for all states.

Below, we vary M between 3 and 6, and vary K between 10 and 20. For $M = 3$, the channel success probabilities are $\vec{p} = [0.3, 0.6, 0.9]$, and for $M = 6$ we use $\vec{p} = [0.1, 0.2, 0.3, 0.5, 0.7, 0.9]$. In Fig. 3, we show the running times of the two algorithms (in milliseconds) to compute the partial index for one $\vec{\lambda}$ and for all channels. When these algorithms are used for pre-computation (i.e., the partial indices for all states must be computed), we can compare the first and second bars within each group in Fig. 3. We can clearly see that our algorithm reduces the running time by orders of magnitude. For example, the reduction is more than 30 times (1.5 vs 50.6ms) for $M = 6$ and $K = 10$. The gap is even larger (almost 100 times) at $M = 6$ and $K = 20$. On the other hand, when these algorithms are used in real time (i.e., only the partial index for the current state needs to be computed), the gap between the first and the third bars is smaller. Nonetheless, our algorithm still shows significant speed-up (between 2 to 5 times) depending on the setting.

6 Conclusion

In this paper, we study how to efficiently compute the partial index. While we focus on the AoI minimization problem under the generate-at-will setting [19], we also identify general structural conditions under which our fast algorithm will work. These general conditions can potentially be applied to other multi-agent MDPs where agents share multiple heterogeneous resources, which we will explore in the future.

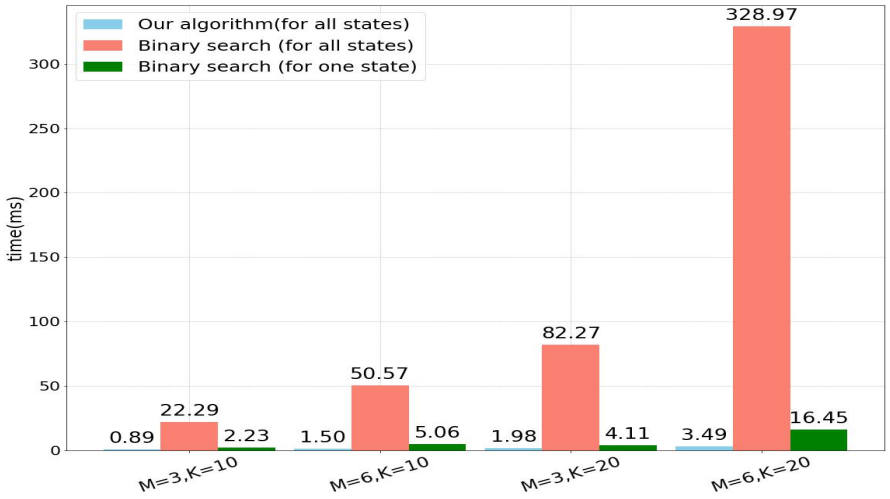


Figure 3: The running times for partial index computation

References

[1] N. Akbarzadeh and A. Mahajan. Dynamic spectrum access under partial observations: A restless bandit approach. In *2019 16th Canadian Workshop on Information Theory (CWIT)*, pages 1–6. IEEE, 2019.

[2] N. Akbarzadeh and A. Mahajan. Restless bandits with controlled restarts: Indexability and computation of Whittle index. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7294–7300, 2019.

[3] N. Akbarzadeh and A. Mahajan. Conditions for indexability of restless bandits and an $\mathcal{O}(k^3)$ algorithm to compute Whittle index. *Advances in Applied Probability*, 54(4):1164–1192, 2022.

[4] E. Altman. *Constrained Markov decision processes*. Routledge, 2021.

[5] E. A. Feinberg and G. He. Complexity bounds for approximately solving discounted mdps by value iterations. *Operations Research Letters*, 48(5):543–548, 2020.

[6] Y.-P. Hsu. Age of information: Whittle index for scheduling stochastic arrivals. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2634–2638, 2018.

[7] Z. Jiang, B. Krishnamachari, S. Zhou, and Z. Niu. Can decentralized status update achieve universally near-optimal age-of-information in wireless multiaccess channels? In *2018 30th International Teletraffic Congress (ITC 30)*, volume 01, pages 144–152, 2018.

[8] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano. Scheduling policies for minimizing age of information in broadcast wireless networks. *IEEE/ACM Transactions on Networking*, 26(6):2637–2650, 2018.

[9] Kahraman, A. Köse, M. Koca, and E. Anarim. Age of information in Internet of Things: A survey. *IEEE Internet of Things Journal*, 11(6):9896–9914, 2024.

[10] A. Maatouk, S. Kriouile, M. Assad, and A. Ephremides. On the optimality of the Whittle's index policy for minimizing the age of information. *IEEE Transactions on Wireless Communications*, 20(2):1263–1277, 2020.

[11] J. Niño-Mora. Dynamic priority allocation via restless bandit marginal productivity indices. *Transactions in Operations Research*, 15:161–198, 2007.

[12] Y. Qian, C. Zhang, B. Krishnamachari, and M. Tambe. Restless poachers: Handling exploration-exploitation tradeoffs in security domains. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 123–131, 2016.

[13] B. Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. *Advances in Neural Information Processing Systems*, 26, 2013.

[14] B. Sombabu, A. Mate, D. Manjunath, and S. Moharir. Whittle index for AoI-aware scheduling. In *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*, pages 630–633, 2020.

[15] J. Sun, Z. Jiang, B. Krishnamachari, S. Zhou, and Z. Niu. Closed-form Whittle's index-enabled random access for timely status update. *IEEE Transactions on Communications*, 68(3):1538–1551, 2020.

[16] V. Tripathi and E. Modiano. A Whittle index approach to minimizing functions of age of information. pages 1160–1167, 2019.

[17] R. R. Weber and G. Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27(3):637–648, 1990.

[18] P. Whittle. Restless bandits: activity allocation in a changing world. *Journal of Applied Probability*, 25(A):287–298, 1988.

[19] Y. Zou, K. T. Kim, X. Lin, and M. Chiang. Minimizing age-of-information in heterogeneous multi-channel systems: A new partial-index approach. In *Proceedings of the twenty-second international symposium on theory, algorithmic foundations, and protocol design for mobile networks and mobile computing*, pages 11–20, 2021.

Appendix

A Proof for Lemma 2.

As π_1 and π_2 are both the optimal policies, we have $V^{\pi_1}(s,\vec{\lambda}) = V^{\pi_2}(s,\vec{\lambda}) = V^*(s,\vec{\lambda})$. According to the Bellman equation, for any state s,

$$V^{\pi_1}(s,\vec{\lambda}) = s + \lambda_{\pi_1(s)} + \beta\left(p_{\pi_1(s)}V^{\pi_1}(1,\vec{\lambda}) + (1-p_{\pi_1(s)})V^{\pi_1}(s+1,\vec{\lambda})\right), \forall s \in \mathcal{S} \tag{14}$$

and

$$V^{\pi_2}(s,\vec{\lambda}) = s + \lambda_{\pi_2(s)} + \beta\left(p_{\pi_2(s)}V^{\pi_2}(1,\vec{\lambda}) + (1-p_{\pi_2(s)})V^{\pi_2}(s+1,\vec{\lambda})\right), \forall s \in \mathcal{S} \tag{15}$$

has the same solution $V^*(s,\vec{\lambda})$. Notice that for any cross-over policy π between π_1 and π_2, for any state s, either $\pi(s) = \pi_1(s)$ or $\pi(s) = \pi_2(s)$ will hold. Then, the system of Bellman Equation

$$V^{\pi}(s,\vec{\lambda}) = s + \lambda_{\pi(s)} + \beta\left(p_{\pi(s)}V^{\pi}(1,\vec{\lambda}) + (1-p_{\pi(s)})V^{\pi}(s+1,\vec{\lambda})\right), \forall s \in \mathcal{S} \tag{16}$$

will also have the solution $V^*(s,\vec{\lambda})$ for all s. Thus, π is also the optimal policy with that given $\vec{\lambda}$.

B Proof for Theorem 6.

Recall that for the AoI minimization problem under the generate-at-will setting, the states are simply the AoI. Further, in order to better represent the state, we use the notation s_i to denote AoI state i. We also note that Π' contains all the policies that choose stronger actions at higher states. We will repeatedly use this property in all the following proofs.

Before we start to prove Theorem 6, we state a lemma, which can be easily verified by algebra.

Lemma 10. *For $a, b, c > 0$ and $a < b$, there is $\frac{a}{b} < \frac{a+c}{b+c}$.*

Going back to Theorem 6, we use A_m^π to denote the subset of states that contains all the states choosing channel m under policy π. This set can be viewed as the "active set" for choosing channel m, which is opposite to the passive set. We also denote the lowest state and the highest state in A_m^π as s_q and s_p, respectively. As $\pi \in \Pi'$, according to the definition of Π', we know that all the states between s_p and s_q will belong to A_m^π. Therefore, A_m^π can be expressed as $A_m^\pi = \{s_p, s_{p+1}, ..., s_q\}$.

Denote $\mathcal{T}_{s_i}^\pi(s_1) \triangleq \mathbb{E}^\pi\left[\sum_{t=0}^\infty \beta^t 1_{\{s(t)=s_i\}} | s(0) = s_1\right]$ as the expected discounted visiting time of state s_i. According to the generate-at-will setting [19], state s_{i+1} can only be reached from state s_i ($i \geq 1$) with probability $1 - p_{\pi(s_i)}$. Using this fact, we first show that $\mathcal{T}_{s_2}^\pi(s_1) = \beta\left(1 - P_{\pi(s_1)}\right)\mathcal{T}_{s_1}^\pi(s_1)$. To see this, note that,

$$
\begin{aligned}
\mathcal{T}_{s_2}^{\pi}(s_1) &= E^{\pi}\left[\sum_{t=0}^{\infty}\beta^t 1_{\{s(t)=s_2\}} \mid s(0)=s_1\right] \\
&= \sum_{t=0}^{\infty} E^{\pi}\left[\beta^t 1_{\{s(t)=s_2\}} \mid s(0)=s_1\right] \\
&= \sum_{t=0}^{\infty} \beta^t P_{\pi}\left\{s(t)=s_2 \mid s(0)=s_1\right\} \\
&= \sum_{t=1}^{\infty} \beta^t P_{\pi}\left\{s(t)=s_2 \mid s(0)=s_1\right\} \qquad (\because s(0)\neq s_2) \\
&\overset{(a)}{=} \sum_{t=1}^{\infty} \beta^t \sum_{s} P_{\pi}\left\{s(t)=s_2 \mid s(t-1)=s\right\} P_{\pi}\left\{s(t-1)=s \mid s(0)=s_1\right\} \\
&\overset{(b)}{=} \sum_{t=1}^{\infty} \beta^t P_{\pi}\left\{s(t)=s_2 \mid s(t-1)=s_1\right\} P_{\pi}\left\{s(t-1)=s_1 \mid s(0)=s_1\right\} \\
&= \sum_{t=1}^{\infty} \beta\left(1-p_{\pi(s_1)}\right)\beta^{t-1} P_{\pi}\left\{s(t-1)=s_1 \mid s(0)=s_1\right\} \\
&= \beta\left(1-p_{\pi(s_1)}\right)\sum_{t=1}^{\infty} E^{\pi}\left[\beta^{t-1} 1_{\{s(t-1)=s_1\}} \mid s(0)=s_1\right] \\
&= \beta\left(1-p_{\pi(s_1)}\right)\sum_{t=0}^{\infty} E^{\pi}\left[\beta^{t} 1_{\{s(t)=s_1\}} \mid s(0)=s_1\right] \\
&= \beta\left(1-p_{\pi(s_1)}\right)\mathcal{T}_{s_1}^{\pi}(s_1).
\end{aligned}
$$

where equality (a) holds by using total probability and the Markov property. And equality (b) holds since $P_{\pi}\left\{s(t)=s_2 \mid s(t-1)=s\right\}=0$, if $s\neq s_1$.

Using the same method, we can obtain, inductively for $i=1,...,K-1$, that

$$
\begin{aligned}
\mathcal{T}_{s_3}^{\pi}(s_1) &= \beta(1-p_{\pi(s_2)})\mathcal{T}_{s_2}^{\pi}(s_1) = \beta^2(1-p_{\pi(s_1)})(1-p_{\pi(s_2)})\mathcal{T}_{s_1}^{\pi}(s_1) \\
&\quad ... \\
\mathcal{T}_{s_i}^{\pi}(s_1) &= \beta^{i-1}\prod_{j=1}^{i-1}(1-p_{\pi(s_j)})\mathcal{T}_{s_1}^{\pi}(s_1).
\end{aligned}
$$

Further, for S_K, we have,

$$\begin{aligned}
\mathcal{T}_{s_K}^{\pi}(s_1) &= \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\beta^t 1_{\{s(t)=s_K\}} \mid s(0)=S_1\right] \\
&= \sum_{t=1}^{\infty}\beta^t\left[P_\pi\left\{s(t)=s_K \mid s(t-1)=s_{K-1}\right\}P_\pi\left\{s(t-1)=s_{K-1} \mid s(0)=s_1\right\}\right. \\
&\qquad \left. +P_\pi\left\{s(t)=s_K \mid s(t-1)=s_K\right\}P_\pi\left\{s(t-1)=s_K \mid s(0)=s_1\right\}\right] \\
&= \beta\left(1-p_{\pi(s_{K-1})}\right)\mathcal{T}_{s_{K-1}}^{\pi}(s_1)+\beta\left(1-p_{\pi(s_K)}\right)\mathcal{T}_{s_K}^{\pi}(s_1) \\
&= \beta^{K-1}\prod_{j=1}^{K-1}\left(1-p_{\pi(s_j)}\right)\mathcal{T}_{s_1}^{\pi}(s_1)+\beta\left(1-p_{\pi(s_K)}\right)\mathcal{T}_{s_K}^{\pi}(s_1).
\end{aligned}$$

We then get,

$$\begin{aligned}
T_{s_K}^{\pi}(s_1) &= \frac{1}{1-\beta\left(1-p_{\pi(s_K)}\right)}\cdot\beta^{K-1}\prod_{j=1}^{K-1}\left(1-p_{\pi(s_j)}\right)\mathcal{T}_{s_1}^{\pi}(s_1) \\
&= \left[1+\beta\left(1-p_{\pi(s_K)}\right)+\beta^2\left(1-p_{\pi(s_K)}\right)^2+\cdots\right]\beta^{K-1}\prod_{j=1}^{K-1}\left(1-p_{\pi(s_j)}\right)\mathcal{T}_{s_1}^{\pi}(s_1) \\
&= \beta^{K-1}\prod_{j=1}^{K-1}\left(1-p_{\pi(s_j)}\right)\mathcal{T}_{s_1}^{\pi}(s_1)+\beta^{K}\prod_{j=1}^{K}\left(1-p_{\pi(s_j)}\right)\mathcal{T}_{s_1}^{\pi}(s_1) \\
&\quad +\beta^{K+1}\prod_{j=1}^{K+1}\left(1-p_{\pi(s_j)}\right)\mathcal{T}_{s_1}^{\pi}(s_1)+\cdots.
\end{aligned}$$

In other words, we can rethink $\mathcal{T}_{s_K}^{\pi}(s_1)$ as the sum of the contributions of many "imaginary" states $t \geq K$, where the contribution from each $s_t, t \geq K$, follows the same form as (*), with $P_{\pi(s_t)} = P_{\pi(s_K)}$. This use "imaginary" states greatly simplifies the presentation below. Specifically, denote $C_i = \beta^{i-1}\Pi_{j=1}^{i-1}(1-p_{\pi(s_j)})$ for all $i = 1, 2, ...,$ (specifically, we have $C_1 = 1$). We then have,

$$\begin{aligned}
\mathcal{T}_{s_k}^{\pi}(s_1) &= C_k\mathcal{T}_{s_1}^{\pi}(s_1), \quad k=1,2,...,K-1, \\
\mathcal{T}_{s_K}^{\pi}(s_1) &= \sum_{k=K}^{\infty}C_k\mathcal{T}_{s_1}^{\pi}(s_1).
\end{aligned} \tag{17}$$

Notice that,

$$\begin{aligned}\sum_{k=1}^{K}\mathcal{T}_{s_k}^{\pi}(s_1) &= \sum_{k=1}^{K}\mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\beta^t 1_{\{s(t)=s_k\}}|s(0)=s_1\right]\\ &= \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\beta^t\sum_{k=1}^{K}1_{\{s(t)=s_k\}}|s(0)=s_1\right]\\ &= \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\beta^t\cdot 1|s(0)=s_1\right]\\ &= \sum_{t=0}^{\infty}\beta^t = \frac{1}{1-\beta}.\end{aligned} \tag{18}$$

At meanwhile, $\sum_{i=1}^{K}\mathcal{T}_{s_i}^{\pi}(s_1) = \mathcal{T}_{s_1}^{\pi}(s_1)\sum_{i=1}^{\infty}C_i$. We then obtain,

$$\begin{aligned}&\mathcal{T}_{s_k}^{\pi}(s_1) = \frac{1}{1-\beta}\frac{C_k}{\sum_{i=1}^{\infty}C_i},\ k=1,...,K-1;\\ \text{and}\quad &\mathcal{T}_{s_K}^{\pi}(s_1) = \frac{1}{1-\beta}\frac{\sum_{i=K}^{\infty}C_i}{\sum_{i=1}^{\infty}C_i}.\end{aligned} \tag{19}$$

We now use (19) to compare the active times for channel m under policy π and π'. Recall that π and π' only differ at a single state $\bar{s}$ and $\pi(\bar{s}) = m$. Based on the different value of $\pi'(\bar{s})$, we divide into two cases, which are $\pi'(\bar{s}) > m$ or $\pi'(\bar{s}) < m$.

Case 1: $\pi'(\bar{s}) > m$. Since $\pi' \in \Pi'$, $\bar{s}$ must be s_q. We then have $\pi'(s_q) > m = \pi(s_q)$ and $p_{\pi'(s_q)} \geq p_{\pi(s_q)} = p_m$. We further divide into two sub-cases to compare T_m^{π} and $T_m^{\pi'}$.

Case 1.1: $q \neq K$. Then, the active times for channel m under policy π and π' can be expressed by $T_m^{\pi}(s_1) = \sum_{i=p}^{q}\mathcal{T}_{s_i}^{\pi}(s_1)$ and $T_m^{\pi'}(s_1) = \sum_{i=p}^{q-1}\mathcal{T}_{s_i}^{\pi'}(s_1)$. For policy π, we have

$$\begin{aligned}T_m^{\pi}(s_1) = \sum_{i=p}^{q}\mathcal{T}_{s_i}^{\pi}(s_1) &= \frac{1}{1-\beta}\frac{\sum_{i=p}^{q}C_i}{\sum_{i=1}^{\infty}C_i}\\ &= \frac{1}{1-\beta}\frac{\sum_{i=p}^{q-1}C_i + C_q}{\sum_{i=1}^{p-1}C_i + \sum_{i=p}^{q-1}C_i + C_q + \sum_{i=q+1}^{\infty}C_i}\\ &= \frac{1}{1-\beta}\frac{b+c}{a+b+c+cde};\end{aligned}$$

where $a = \sum_{i=1}^{p-1}C_i$, $b = \sum_{i=p}^{q-1}C_i$, $c = C_q$, $d = \beta\ (1-p_{\pi(s_q)})$ and $e = 1 + \sum_{i=1}^{\infty}\beta^i\prod_{j=q+1}^{i+q}(1-p_{\pi(s_j)})$. Here we have written the last term of the denominator as

$$\begin{aligned}\sum_{i=q+1}^{\infty}C_i =&\beta^q\ \prod_{j=1}^{q}(1-p_{\pi(s_j)})\left[1+\sum_{i=1}^{\infty}\beta^i\prod_{j=q+1}^{i+q}(1-p_{\pi(s_j)})\right]\\ =&C_q\cdot\beta\ (1-p_{\pi(s_q)})\cdot\left[1+\sum_{i=1}^{\infty}\beta^i\prod_{j=q+1}^{i+q}(1-p_{\pi(s_j)})\right] = cde;\end{aligned}$$

Similarly, for policy π', if we denote $d' = \beta(1 - p_{\pi'(s_q)}) < d$, then $T_m^{\pi'}(s_1)$ can be expressed as

$$T_m^{\pi'}(s_1) = \sum_{i=p}^{q-1} \mathcal{T}_{s_i}^{\pi'}(s_1) = \frac{1}{1-\beta} \frac{b}{a+b+c+cd'e}.$$

Therefore, proving $T_m^{\pi}(s_1) > T_m^{\pi'}(s_1)$ is equivalent to proving $\frac{b+c}{a+b+c+cde} > \frac{b}{a+b+c+cd'e}$. From Lemma 10, we have $\frac{b+(d-d')ce}{a+b+c+cde} > \frac{b}{a+b+c+cd'e}$. Thus it suffices to prove

$$\begin{aligned} \frac{b+c}{a+b+c+cde} &> \frac{b+(d-d')ce}{a+b+c+cde} \\ \iff 1 &> (d-d')e \\ \iff e &< \frac{1}{d-d'}. \end{aligned}$$

Recall that $\pi' \in \Pi'$. Then, we have $\pi'(s_j) \geq \pi'(s_q)$ for all the states $s_j \geq s_q$ $(j \geq q)$. Further, we have $p_{\pi'(s_j)} \geq p_{\pi'(s_q)}$ for all $j \geq q$. Therefore,

$$e = 1 + \sum_{i=1}^{\infty} \beta^i \prod_{j=q+1}^{i+q} (1 - p_{\pi(s_j)}) \leq 1 + \sum_{i=1}^{\infty} \beta^i (1 - p_{\pi'(s_q)})^i < \sum_{i=0}^{\infty} (d')^i = \frac{1}{1-d'} < \frac{1}{d-d'}.$$

The last step is because $d < 1$. Thus the result of the theorem for $\pi'(\bar{s}) > m$ and $q \neq K$ then follows.

Case 1.2: $q = K$. Then, the active times for channel m under policy π and π' can be expressed by $T_m^{\pi}(s_1) = \sum_{i=p}^{K} \mathcal{T}_{s_i}^{\pi}(s_1)$ and $T_m^{\pi'}(s_1) = \sum_{i=p}^{K-1} \mathcal{T}_{s_i}^{\pi'}(s_1)$. For policy π, we have

$$\begin{aligned} T_m^{\pi}(s_1) &= \sum_{i=p}^{K} T_{s_i}^{\pi}(s_1) = \frac{1}{1-\beta} \frac{\sum_{i=p}^{\infty} C_i}{\sum_{i=1}^{\infty} C_i} \\ &= \frac{1}{1-\beta} \frac{\sum_{i=p}^{K-1} C_i + C_K + \sum_{i=K+1}^{\infty} C_i}{\sum_{i=1}^{p-1} C_i + \sum_{i=p}^{K-1} C_i + C_K + \sum_{i=K+1}^{\infty} C_i} \\ &= \frac{1}{1-\beta} \frac{b + c + c\frac{d}{1-d}}{a+b+c+c\frac{d}{1-d}} \\ &= \frac{1}{1-\beta} \frac{b + \frac{c}{1-d}}{a+b+\frac{c}{1-d}}, \end{aligned} \tag{20}$$

where $a = \sum_{i=1}^{p-1} C_i$, $b = \sum_{i=p}^{K-1} C_i$, $c = C_K$, $d = \beta\ (1 - p_{\pi(s_K)})$ and $\sum_{i=K+1}^{\infty} C_i$ have been written as

$$\begin{aligned}
\sum_{i=K+1}^{\infty} C_i &= \sum_{i=K}^{\infty} \beta^{i-1} \prod_{j=1}^{i-1} \left(1 - p_{\pi(s_j)}\right) \\
&= \beta^K \prod_{j=1}^{K} \left(1 - p_{\pi(s_j)}\right) \left[1 + \sum_{i=1}^{\infty} \beta^i \prod_{j=K+1}^{i+K} \left(1 - p_{\pi(s_j)}\right)\right] \\
&= C_K \cdot \beta \left(1 - p_{\pi(s_K)}\right) \left[1 + \sum_{i=1}^{\infty} \beta^i \prod_{j=K+1}^{i+K} \left(1 - p_{\pi(s_j)}\right)\right] \\
&= C_K \cdot \beta \left(1 - p_{\pi(s_K)}\right) \left[1 + \sum_{i=1}^{\infty} \beta^i \left(1 - p_{\pi(s_K)}\right)^i\right] \\
&= C_K \cdot \beta \left(1 - p_{\pi(s_K)}\right) \frac{1}{1 - \beta\left(1 - p_{\pi(s_K)}\right)} \\
&= c\frac{d}{1-d}.
\end{aligned} \tag{21}$$

Similarly, for policy π', if we denote $d' = \beta(1 - p_{\pi'(s_K)}) < d$, then $T_m^{\pi'}(s_1)$ can be expressed as

$$T_m^{\pi'}(s_1) = \sum_{i=p}^{q-1} \mathcal{T}_{s_i}^{\pi'}(s_1) = \frac{1}{1-\beta} \frac{b}{a + b + \frac{c}{1-d'}}.$$

Therefore, proving $T_m^{\pi}(s_1) > T_m^{\pi'}(s_1)$ is equivalent to proving $\frac{b+\frac{c}{1-d}}{a+b+\frac{c}{1-d}} > \frac{b}{a+b+\frac{c}{1-d'}}$. In fact, we have,

$$\frac{b + \frac{c}{1-d}}{a + b + \frac{c}{1-d}} > \frac{b + \frac{c}{1-d} - \frac{c}{1-d'}}{a + b + \frac{c}{1-d}} > \frac{b}{a + b + \frac{c}{1-d'}} \tag{22}$$

where the second inequality uses Lemma 10. Thus, the result of the theorem for $\pi'(\bar{s}) > m$ and $q = K$ then follows.

Case 2: $\pi'(\bar{s}) < m$. Since $\pi' \in \Pi'$, $\bar{s}$ must be s_p. We then have $\pi'(s_p) < m = \pi(s_p)$ and $p_{\pi'(s_p)} \leq p_{\pi(s_p)} = p_m$.

Case 2.1: $p, q \neq K$. Then, the active times for channel m under policy π and π' can be expressed by $T_m^{\pi}(s_1) = \sum_{i=p}^{q} \mathcal{T}_{s_i}^{\pi}(s_1)$ and $T_m^{\pi'}(s_1) = \sum_{i=p+1}^{q} \mathcal{T}_{s_i}^{\pi'}(s_1)$.

$$\begin{aligned}
T_m^{\pi}(s_1) &= \sum_{i=p}^{q} \mathcal{T}_{s_i}^{\pi}(s_1) = \frac{1}{1-\beta} \frac{\sum_{i=p}^{q} C_i}{\sum_{i=1}^{\infty} C_i} \\
&= \frac{1}{1-\beta} \frac{C_p + \sum_{i=p+1}^{q} C_i}{\sum_{i=1}^{p-1} C_i + C_p + \sum_{i=p+1}^{q} C_i + \sum_{i=q+1}^{\infty} C_i} \\
&= \frac{1}{1-\beta} \frac{b + bcd}{a + b + bc(d+e)};
\end{aligned}$$

where $a = \sum_{i=1}^{p-1} C_i$, $b = C_p$, $c = \beta\ (1 - p_{\pi(s_p)})$, $d = \sum_{i=p+1}^{q} \beta^{i-p-1} \prod_{j=p+1}^{i-1}(1 - p_{\pi(s_j)})$, $e = \sum_{i=q+1}^{\infty} \beta^{i-p-1} \prod_{j=q+1}^{i-1}(1 - p_{\pi(s_j)})$. Here, we have written the last two terms on the denominator as,

$$\begin{aligned}
\sum_{i=p+1}^{q} C_i =& \beta^p \prod_{j=1}^{p}(1 - p_{\pi(s_j)}) \left[\sum_{i=p+1}^{q} \beta^{i-p-1} \prod_{j=p+1}^{i-1} (1 - p_{\pi(s_j)}) \right] \\
=& C_p \cdot \beta\ (1 - p_{\pi(s_p)}) \cdot \left[\sum_{i=p+1}^{q} \beta^{i-p-1} \prod_{j=p+1}^{i-1} (1 - p_{\pi(s_j)}) \right] = bcd; \\
\sum_{i=q+1}^{\infty} C_i =& \beta^p \prod_{j=1}^{p}(1 - p_{\pi(s_j)}) \left[\sum_{i=q+1}^{\infty} \beta^{i-p-1} \prod_{j=p+1}^{i-1} (1 - p_{\pi(s_j)}) \right] \\
=& C_p \cdot \beta\ (1 - p_{\pi(s_p)}) \cdot \left[\sum_{i=q+1}^{\infty} \beta^{i-p-1} \prod_{j=p+1}^{i-1} (1 - p_{\pi(s_j)}) \right] = bce;
\end{aligned}$$

Similarly, for policy π', if we denote $c' = \beta(1 - p_{\pi'(s_p)}) > c$, then $T_m^{\pi'}(s_1)$ can be expressed as

$$T_m^{\pi'}(s_1) = \sum_{i=p+1}^{q} \mathcal{T}_{s_i}^{\pi'}(s_1) = \frac{1}{1-\beta} \frac{bc'd}{a + b + bc'(d+e)}. \tag{23}$$

Therefore, proving $T_m^{\pi}(s_1) > T_m^{\pi'}(s_1)$ is equivalent to proving $\frac{b+bcd}{a+b+bc(d+e)} > \frac{bc'd}{a+b+bc'(d+e)}$.

$$\begin{aligned}
& \frac{b + bcd}{a + b + bc(d+e)} > \frac{bc'd}{a + b + bc'(d+e)} \\
\Longleftrightarrow & 1 - \frac{b + bcd}{a + b + bc(d+e)} < 1 - \frac{bc'd}{a + b + bc'(d+e)} \\
\Longleftrightarrow & \frac{a + bce}{a + b + bc(d+e)} < \frac{a + b + bc'e}{a + b + bc'(d+e)}
\end{aligned} \tag{24}$$

From Lemma 10, we have $\frac{a+bce}{a+b+bc(d+e)} < \frac{a+bce+b(c'-c)(d+e)}{a+b+bc'(d+e)}$. Thus it suffices to prove

$$\begin{aligned}
& \frac{a + bce + b(c'-c)(d+e)}{a + b + bc'(d+e)} < \frac{a + b + bc'e}{a + b + bc'(d+e)} \\
\Longleftrightarrow & (c'-c)d < 1 \\
\Longleftrightarrow & d < \frac{1}{c'-c}.
\end{aligned}$$

Recall that $\pi \in \Pi'$. Then, we have $\pi(s_j) \geq \pi(s_p)$ for all the states $s_j \geq s_p$ $(j \geq p)$. Further, we have $p_{\pi(s_j)} \geq p_{\pi(s_p)}$ for all $j \geq p$. Therefore,

$$d = \sum_{i=p+1}^{q} \beta^{i-p-1} \prod_{j=p+1}^{i-1} (1 - p_{\pi(s_j)}) \leq \sum_{i=0}^{q-p-1} \beta^i (1 - p_{\pi(s_p)})^i < \sum_{i=0}^{\infty} c^i = \frac{1}{1-c} < \frac{1}{c'-c}.$$

The last step is because $c' < 1$. Thus the result of the theorem for $\pi'(\bar{s}) < m$ and $p, q \neq K$ then follows.

Case 2.2: $q = K$ and $p \neq K$. This case is very similar to case 2.1. The active times for channel m under policy π and π' can be expressed by $T_m^{\pi}(s_1) = \sum_{i=p}^{q} \mathcal{T}_{s_i}^{\pi}(s_1)$ and $T_m^{\pi'}(s_1) = \sum_{i=p+1}^{q} \mathcal{T}_{s_i}^{\pi'}(s_1)$.

$$\begin{aligned} T_m^{\pi}(s_1) &= \sum_{i=p}^{q} \mathcal{T}_{s_i}^{\pi}(s_1) = \frac{1}{1-\beta} \frac{\sum_{i=p}^{\infty} C_i}{\sum_{i=1}^{\infty} C_i} \\ &= \frac{1}{1-\beta} \frac{C_p + \sum_{i=p+1}^{\infty} C_i}{\sum_{i=1}^{p-1} C_i + C_p + \sum_{i=p+1}^{\infty} C_i + \sum_{i=q+1}^{\infty} C_i} \\ &= \frac{1}{1-\beta} \frac{b + bcd}{a + b + bcd}; \end{aligned}$$

where $a = \sum_{i=1}^{p-1} C_i$, $b = C_p$, $c = \beta\ (1 - p_{\pi(s_p)})$, $d = \sum_{i=p+1}^{\infty} \beta^{i-p-1} \prod_{j=p+1}^{i-1} (1 - p_{\pi(s_j)})$.

Then, if we denote $c' = \beta(1 - p_{\pi'(s_p)}) > c$, $T_m^{\pi'}(s_1)$ can be formulated as,

$$T_m^{\pi'}(s_1) = \sum_{i=p+1}^{q} \mathcal{T}_{s_i}^{\pi'}(s_1) = \frac{1}{1-\beta} \frac{bc'd}{a + b + bc'd}.$$

We can prove $\frac{b+bcd}{a+b+bcd} > \frac{bc'd}{a+b+bc'd}$ by using the same way as in case 2.1.

Case 2.3: $p = K$. As state s_K is the largest state and $\pi'(s_K) < m$, according to $\pi' \in \Pi'$, it implies π' will not choose channel m for any state. Therefore, the active time for channel m under policy π' is zero, which is smaller than the active time for channel m under policy π.

Combining all these cases, we complete the whole proof.

C Proof for Theorem 7.

Let us define two Markov chains, chain 0 and chain 1. The two chains have exactly the same initial state $s(0) = s_1$ and dual cost $\vec{\lambda}$ of channels, but chain 0 follows policy π and chain 1 follows policy π'. This means that their state transition probabilities are also almost the same, except for state $\bar{s}$ where their decisions differ. To compare their active times, we perform the following stochastic coupling on the random transitions. From time 0 onwards,

(1) For each $s \neq \bar{s}$, the l-th transition from state s of chain 1 will have the same channel success (i.e. to state $s(0) = s_1$) or failure (i.e. to state $s+1$) event as the l-th transition from state s of chain 0.

(2) If the l-th transition from state $\bar{s}$ of chain 0 has the success event, then the l-th transition from state $\bar{s}$ of chain 1 will also have the success event. If the l-th transition from state $\bar{s}$ of chain 0 has the failure event, then the l-th transition from state $\bar{s}$ of chain 1 will have the success event (i.e., to state $s(0) = s_1$) with probability $\frac{p_{u+1}-p_u}{1-p_u}$ and will have the failure event (i.e., to state $\bar{s}+1$) with probability $\frac{1-p_{u+1}}{1-p_u}$. (Recall that we have assumed $p_{u+1} > p_u$ in our channel ordering.)

Define S_+ as the subset of $\mathcal{S}$ that contains all the states higher than $\bar{s}$, and S_- as the subset of $\mathcal{S}$ that contains all the states lower than $\bar{s}$. Thus, $\mathcal{S} = \{\bar{s}\} \cup S_+ \cup S_-$. Let $I_+^0(k)$, $n = 1, 2, ...$ denote the k-th contiguous time-interval such that the state of chain 0 is in S_+. We call this interval the k-th S_+-episode of chain 0. Analogously, we define $I_-^0(k)$, $I_+^1(k)$, and $I_-^1(k)$.

Lemma 11. *Under our coupling, for each k, the sequence of states visited in the k-th S_+-episode (i.e., the time-interval $I_+^0(k)$) of chain 0 is the same as the sequence of states visited in the k-th S_+-episode (i.e., the time-interval $I_+^1(k)$) of chain 1. Similarly, the sequence of states visited in the k-th S_--episode (i.e., the time-interval $I_-^0(k)$) of chain 0 is also the same as the sequence of states visited in k-th S_--episode (i.e., the time-interval $I_-^1(k)$) of chain 1.*

Proof. We prove this by induction. Our hypothesis is just the conclusion of the lemma. As state $\bar{s}$ can be any state in the state space, S_- or S_+ may become an empty set when $\bar{s}$ is the smallest state s_1 or the largest state s_K, respectively. In our following proof, we only discuss the more common case that both S_- and S_+ are not empty sets, i.e., $s_1 < \bar{s} < s_K$. The other two corner cases can be verified in a very similar way.

Before we use the induction method, we note that the first state of each S_--episode for both chain 0 and chain 1 must be s_1; and the first state of each S_+-episode for both chain 0 and chain 1 must be $\bar{s} + 1$.

We now prove the base step $k = 1$. As $s_1 < \bar{s} < s_K$, we can directly have $s_1 \in S_-$. Consider the first S_--episodes of chain 0 and chain 1. As the states in S_--episode all belong to $S_-(\not\ni \bar{s})$, according to our coupling (1), from the same first state s_1, the sequence will choose the same channel for transmission, have the same success/failure event and will enter the same state one by one, until the states of chain 0 and chain 1 leave S_- simultaneously. Hence, the first S_--episode in $I_-^0(1)$ of chain 0 and the first S_--episode in $I_-^1(1)$ of chain 1 must be the same. Next, we consider the first S_+-episodes of chain 0 and chain 1. As the states of these two S_+-episodes of chain 0 and chain 1 all belong to $S_+(\not\ni \bar{s})$, we can also use coupling (1). According to the statement of the last paragraph, we know that the first state of the sequence in $I_+^0(1)$ of chain 0 and the first state of the sequence in $I_+^0(1)$ of chain 1 are the same, i.e., $\bar{s}+1$. From the same first state of both S_+-episodes, the sequences must choose the same channel for transmission, have the same success/failure event and will enter the same next state one by one, until the states of chain 0 and chain 1 leave S_+ simultaneously. Therefore, the first S_+-episode in $I_+^0(1)$ of chain 0 and the first S_+-episode in $I_+^1(1)$ of chain 1 must also be the same.

We now prove the induction step. That is, assume that the induction hypothesis holds for $1, ..., k$, we now prove that it also holds for $k + 1$. We first consider the $(k + 1)$-th S_--episode of chain 0 and the $(k + 1)$-th S_--episode of chain 1. To begin with, according to the induction assumption that the i-th ($i \leq k$) S_--episodes of both chain 0 and chain 1 are the same, we know that for any state s in S_-, the total number of times that chain 0 reaches state s in its first k S_--episodes is the same as that by chain 1. In other words, for any state $s \in S_-$, the same numbers of transitions from state s for both chain 0 and chain 1 are the same. Next, we know that the first state of the $(k + 1)$-th S_--episode

for chain 0 is the same with the first state of the $(k+1)$-th S_--episode for chain 1. Then we can use coupling (1) to compare the $(k+1)$-th S_--episode of chain 0 and the $(k+1)$-th S_--episode of chain 1. That is, from the same first state of both episodes, the two sequences will choose the same channel for transmission, have the same success/failure event and will enter the same next state, until they simultaneously leave S_-. Therefore, the $(k+1)$-th S_--episode in $I_-^0(k+1)$ of chain 0 and the $(k+1)$-th S_--episode in $I_-^1(k+1)$ of chain 1 must also be the same. Using the same method, we can also show that the $(k+1)$-th S_+-episode in $I_+^0(k+1)$ of chain 0 and the $(k+1)$-th S_+-episode in $I_+^1(k+1)$ of chain 1 must be the same. Thus, the result of Lemma 11 follows. □

Let $\tau_k^0(\bar{s})$ be the k-th time that the chain 0 hits state $\bar{s}$. Let $n_0^+(k)$ be the number of S^+-episodes of chain 0 before time $\tau_k^0(\bar{s})$, and let $n_0^-(k)$ be the number of S_--episodes before time τ_k^0. Analogously, we define $\tau_k^1(\bar{s})$, $n_1^+(k)$, $n_1^-(k)$ for chain 1. We will prove the following two lemmas below. (For simplicity, we use the short-hand notation $\tau_k^0 = \tau_k^0(\bar{s})$ and $\tau_k^1 = \tau_k^1(\bar{s})$.)

Lemma 12. *Before time τ_k^0 (including τ_k^0), chain 0 has k S_--episodes. Before time τ_k^1 (including τ_k^1), chain 1 also has k S_--episodes.*

Proof. We only prove the result for chain 0, and the proof for chain 1 is exactly the same.

Notice that before the chain 0 enters an S_+-episode, the chain must reaches state $\bar{s}$ first. After the chain leaves an S_+-episode, chain 0 must hit state $s_1 \in S_-$ next. Similarly, before chain 0 enters an S_--episode, the chain will either hit $\bar{s}$ or in S_+. After the chain leaves an S_--episode, the chain must hit $\bar{s}$. Note that the initial state is $s_1 \in S_-$.

Suppose that there are a_k number of S_+-episodes before τ_k^0. Before each of these episodes, the chain must hit state $\bar{s}$. Therefore, among the $(k-1)$ time-instants (before τ_k^0, not including τ_k^0) that the state of chain 0 is $\bar{s}$, a_k of them will transit to $\bar{s}+1 \in S_+$. For the remaining $(k-a_k-1)$ time-instants that the state of chain 0 is $\bar{s}$, it should transit to $s_1 \in S_-$. Notice that after the state of the chain leaves each of the above a_k of S_+-episodes, it will also enter S_-. Further, the initial episode is the S_--episode. Putting these together, the number of the S_-episodes before τ_k^0 is then $(k-1-a_k)+a_k+1 = k$. □

The next lemma shows that each S_+-episode of chain 1 occurs no earlier than that of chain 0, and each S_--episode of chain 1 occurs no later than that of chain 0.

Lemma 13. *For each k, we have (i) $n_1^+(k) \le n_0^+(k)$ and (ii) $n_1^-(k) \ge n_0^-(k)$.*

Proof. We prove the lemma by induction. The induction hypothesis is just the conclusion of the lemma. The base step is trivial because by our coupling, the two chains should have the same evolution in their first S_--episodes until $t = \tau_k^1 = \tau_k^0$. Hence, we will have $n_1^+(1) = n_0^+(1) = 0$ and $n_1^-(1) = n_0^-(1) = 1$.

Next, we will prove the induction step. Suppose that the induction hypothesis holds for k. We next prove that $n_1^+(k+1) \le n_0^+(k+1)$ and $n_1^-(k+1) \ge n_0^-(k+1)$. Consider the evolution of the two

chains starting from time-instants τ_k^1 and τ_k^0, respectively. Through our coupling, either they have the same succuss/failure event, or chain 0 fails but chain 1 succeeds. Therefore, we divide into several cases.

If they have the same success event, they both start a new S_--episode. Thus, we must have $n_1^+(k+1) = n_1^+(k)$, $n_1^-(k+1) = n_1^-(k)+1$, $n_0^+(k+1) = n_0^+(k)$ and $n_0^-(k+1) = n_0^-(k)+1$. According to the assumption of our induction hypothesis that $n_1^+(k) \le n_0^+(k)$ and $n_1^-(k) \ge n_0^-(k)$, we will then obtain $n_1^+(k+1) \le n_0^+(k+1)$ and $n_1^-(k+1) \ge n_0^-(k+1)$.

If they have the same failure event, they both start a new S_+-episode. Thus, we must have $n_1^+(k+1) = n_1^+(k)+1$, $n_1^-(k+1) = n_1^-(k)$, $n_0^+(k+1) = n_0^+(k)+1$ and $n_0^-(k+1) = n_0^-(k)$. According to the assumption of our induction hypothesis that $n_1^+(k) \le n_0^+(k)$ and $n_1^-(k) \ge n_0^-(k)$, we will also obtain $n_1^+(k+1) \le n_0^+(k+1)$ and $n_1^-(k+1) \ge n_0^-(k+1)$.

If chain 0 fails and chain 1 succeeds, then chain 0 starts a new S_+-episode, and chain 1 starts a new S_--episode. Thus, we must have $n_1^+(k+1) = n_1^+(k)$, $n_1^-(k+1) = n_1^-(k)+1$, $n_0^+(k+1) = n_0^+(k)+1$ and $n_0^-(k+1) = n_0^-(k)$. According to the assumption of our induction hypothesis that $n_1^+(k) \le n_0^+(k)$ and $n_1^-(k) \ge n_0^-(k)$, we can obtain that $n_1^+(k+1) = n_1^+(k) \le n_0^+(k) < n_0^+(k+1)$ and $n_1^-(k+1) > n_1^-(k) \ge n_0^-(k) = n_0^-(k+1)$.

Combining these three cases, we can conclude that the induction hypothesis will hold for $k+1$. The result of the lemma then follows. □

Using Lemma 13 and Lemma 12, we now show Lemma 8 holds, i.e., $t_+^0(k) \le t_+^1(k)$ and $t_-^0(k) \ge t_-^1(k)$ for all k, where $t_+^0(k)$ is denoted as the k-th time when chain 0 transitions from $\mathcal{S}\backslash S_+$ to S_+, and $t_-^0(k)$ is the k-th time when chain 0 transitions from $\mathcal{S}\backslash S_-$ to S_-. $t_+^1(k)$ and $t_-^1(k)$ are defined analogously for chain 1.

Proof of Lemma 8. We first compare $t_+^0(k)$ and $t_+^1(k)$. According to the definition of $t_+^0(k)$, chain 0 must be at state $\bar{s}$ at time $t_+^0(k)-1$. Suppose that this is the k'-th time that chain 0 is at state $\bar{s}$, i.e., $\tau_{k'}^0 = t_+^0(k) - 1$. By lemma 13, we have

$$n_1^+(k') \le n_0^+(k') \tag{25}$$

$$n_1^-(k') \ge n_0^-(k'). \tag{26}$$

By definition of $t_+^0(k)$, chain 0 must have completed $k-1$ of S_+-episodes. Thus, we have $n_0^+(k') = k-1$. Meanwhile, we also know that $n_0^-(k') = k'$ by Lemma 12. According to (25), it implies that,

$$n_1^+(k') \le k-1, \quad n_1^-(k') \ge k'. \tag{27}$$

In other words, at the k'-th time that chain 1 enters $\bar{s}$ (which is $\tau_{k'}^1$), the number of S_+-episodes before it is no larger than $k-1$. Hence, the k-th time that chain 1 transits from $\bar{s}$ to S_+ must be at or after $\tau_{k'}^1$, i.e.,

$$t_+^1(k) - 1 \ge \tau_{k'}^1. \tag{28}$$

Suppose that at $t_+^1(k)-1$, it is also the k''-th time that chain 1 enters state $\bar{s}$, i.e., $\tau_{k''}^1 = t_+^1(k)-1$. Then, we must have

$$k'' \geq k'.$$

We thus have

$$\begin{aligned} &n_1^+(k'') = k-1, \quad \text{(by defition of } t_+^1(k)) \\ &n_1^-(k'') = k'' \geq k' \end{aligned}$$

In other words, we have shown the following. At time $\tau_{k'}^0$, chain 0 has $(k-1)$ of S_+-episodes, k' of S_--episodes, and reaches $\bar{s}$ for k' times hitting $\bar{s}$. At time $\tau_{k''}^1$, chain 1 has $(k-1)$ of S_+-episodes, $k''(\geq k')$ of S_--episodes, and reaches $\bar{s}$ for $k''(\geq k')$ times. By Lemma 11, the length of the n-th S_+-episode (or S_--episode) is the same for both chain 0 and chain 1. Thus, we must have,

$$\begin{aligned} &\tau_{k''}^1 \geq \tau_{k'}^0 \\ \Rightarrow &t_+^1(k) - 1 \geq t_+^0(k) - 1 \\ \Rightarrow &t_+^1(k) \geq t_+^0(k). \end{aligned} \tag{29}$$

Next, we compare $t_-^0(k)$ and $t_-^1(k)$. By Lemma 11, both chain 0 and chain 1 will start their k-th S_--episode at $t_-^0(k)$ and $t_-^1(k)$, respectively, and these two S_--episodes should have the same sequence. We denote the length of this k-th S_--episode as l_k. As both chain 0 and chain 1 will hit state $\bar{s}$ right after they leave their k-th S_--episode, chain 0 and chain 1 must be at state $\bar{s}$ at $t_-^0(k)+l_k$ and $t_-^1(k)+l_k$, respectively. Suppose that this is the k'-th time that chain 0 is at state $\bar{s}$, i.e., $\tau_{k'}^0 = t_-^0(k)+l_k$. By definition of $t_-^0(k)$ and $\tau_{k'}^0$, chain 0 must have completed k of S_--episodes before $\tau_{k'}^0$. Thus we have $n_0^-(k') = k$. According to Lemma 12, $n_0^-(k')$ also equals to k'. This implies that, at time $t_-^0(k)+l_k$, it is exactly the k-th time that chain 0 is at state $\bar{s}$. Therefore, we have $k'=k$, $\tau_k^0 = t_-^0(k)+l_k$ and $n_0^-(k) = k$. For the same reason, we can know that at time $t_-^1(k)+l_k$, it is also the k-th time that chain 1 is at state $\bar{s}$, i.e., $\tau_k^1 = t_-^1(k)+l_k$ and we must have $n_1^-(k) = k$.

Meanwhile, denote $a_k \stackrel{\triangle}{=} n_0^+(k)$ and $b_k \stackrel{\triangle}{=} n_1^+(k)$. According to Lemma **??**, we have $b_k = n_1^+(k) \leq n_0^+(k) = a_k$. In other words, we have shown the following. At time τ_k^0, chain 0 has a_k of S_+-episodes, k of S_--episodes, and hit $\bar{s}$ for k times. On the other hand, at time τ_k^1, chain 1 has $b_k(\leq a_k)$ of S_+-episodes, k of S_--episodes, and hit $\bar{s}$ for k times. By Lemma 11, the length of the n-th S_+-episode (or S_--episode) is the same for both chain 0 and chain 1. Thus, we must have,

$$\begin{aligned} &\tau_k^1 \leq \tau_k^0 \\ \Rightarrow &t_-^1(k) + l_k \geq t_-^0(k) + l_k \\ \Rightarrow &t_-^1(k) \leq t_-^0(k) \end{aligned} \tag{30}$$

The result of Lemma 8 then follows.

□

Recall that π and π' differ at state $\bar{s}$, with $\pi(\bar{s}) = u$ and $\pi(\bar{s}) = u+1$. Thus, either when $u+1 < m$ or when $u > m$, chain 0 and chain 1 should use action m on the same set of states. We denote this state set as A_m. We then have the following lemma.

Lemma 14. *Denote $\tau_k^0(s)$ and $\tau_k^1(s)$ as the k-th time that chain 0 and chain 1 enter state s, respectively. For any state $s \in A_m$,*

(I) if $u+1 < m$, then $\tau_k^0(s) \leq \tau_k^1(s)$;

(II) if $u > m$, then $\tau_k^0(s) \geq \tau_k^1(s)$.

Proof. (I) If $u+1 < m$, it implies that channel $u+1$ has lower priority than channel m. Notice that $\pi \in \Pi'$, which implies that a higher state will use a channel with higher priority. Further, notice that $\pi(s) = m$ and $\pi(\bar{s}) = u < m$. Therefore, we must have $\bar{s} < s$, and thus $A_m \subseteq S_+$. Using Lemma 8, each S_+-episode of chain 1 occurs no earlier than that of chain 0. Further, the k-th S_+-episode of chain 1 has the same state evolution as the k-th S_+-episode of chain 0, and thus both episodes enter state s with the same sequence. Combining these facts, we therefore have $\tau_k^0(s) \leq \tau_k^1(s)$.

(II) Similar to (I). If $u > m$, it implies that $\pi(\bar{s}) = u > m = \pi(s)$. By $\pi \in \Pi'$, we must have $s < \bar{s}$, and thus $A_m \subseteq S_-$. Using Lemma 8, each S_--episode of chain 1 occurs no later than that of chain 0. Further, the k-th S_--episode of chain 1 has the same state evolution as the k-th S_--episode of chain 0, and thus both episodes enter state s with the same sequence. Combining these facts, we therefore have $\tau_k^0(s) \geq \tau_k^1(s)$. □

We can now complete the proof of theorem 7. If $u+1 < m$, according to Lemma 14,

$$\begin{aligned} T_m^{\pi}(s_0) &= \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \beta^t 1_{\{u^{\pi}(t)=m\}} \mid s_0\right] \\ &= \mathbb{E}^{\pi}\left[\sum_{k=1}^{\infty} \sum_{s\in A_m} \beta^{\tau_k^0(s)} \mid s_0\right] \\ &< \mathbb{E}^{\pi'}\left[\sum_{k=1}^{\infty} \sum_{s\in A_m} \beta^{\tau_k^1(s)} \mid s_0\right] \\ &= T_m^{\pi'}(s_0), \end{aligned} \tag{31}$$

where in the first inequality, we use $<$ instead of $\leq$ because there is a non-zero probability that the S_--episodes of chain 1 occurs strictly earlier than that of chain 0, and thus the there is a non-zero probability that $\sum_{k=1}^{\infty}\sum_{s\in A_m}\beta^{\tau_k^0(s)} < \sum_{k=1}^{\infty}\sum_{s\in A_m}\beta^{\tau_k^1(s)}$.

(ii) Similar to (i). We have,

$$\begin{aligned} T_m^{\pi}(s_0) &= \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \beta^t 1_{\{u^{\pi}(t)=m\}} \mid s_0\right] \\ &= \mathbb{E}^{\pi}\left[\sum_{k=1}^{\infty} \sum_{s\in A_m} \beta^{\tau_k^0(s)} \mid s_0\right] \\ &> \mathbb{E}^{\pi'}\left[\sum_{k=1}^{\infty} \sum_{s\in A_m} \beta^{\tau_k^1(s)} \mid s_0\right] \\ &= T_m^{\pi'}(s_0). \end{aligned} \tag{32}$$

where in the first inequality, we use $>$ instead of $\geq$ because there is a non-zero probability that the S_+-episodes of chain 0 occurs strictly earlier than that of chain 1, and thus the there is a non-zero probability that $\sum_{k=1}^{\infty}\sum_{s\in A_m}\beta^{\tau_k^0(s)} > \sum_{k=1}^{\infty}\sum_{s\in A_m}\beta^{\tau_k^1(s)}$.

D Proof for Corollary 1.

We first verified the second part of Assumption 2. Considering policy $\pi \in \Pi'$ such that $\pi(\bar{s}) = m$ for some state $\bar{s}$, and consider an operation ψ on π that satisfies $\psi(\pi)(\bar{s}) \neq \pi(\bar{s}) = m$ and $\psi(\pi) \in \Pi'$. By Theorem 6, this operation ψ must be a valid operation. From Remark 1, (ψ, π) must belong to Ψ because ψ satisfies statement (iii) of Remark 1. Therefore, ψ is a Ψ-valid operation. The second part of Assumption 2 thus holds.

It remains to verify the first part of Assumption 2. That is, for any policy $\pi \in \Pi'$ such that $\pi(\bar{s}) = m$ for some state $\bar{s}$, we will show that there must exist a Ψ-valid operation ψ on π and $\psi(\pi) \in \Pi'$. Note that we cannot simply change the decision at state $\bar{s}$ to an action different from m, because the new policy may not satisfy part (3) of condition 1, and thus it may not belong to Π'. Instead, we look at the set A_m^π of all the states choosing channel m under policy π. A_m^π is not empty because there exists some state $\bar{s}$ such that $\pi(\bar{s}) = m$. Since we assume that $\pi \in \Pi'$, this set of states must be contiguous due to part (3) of Condition 1. Thus, we can denote $A_m^\pi = \{a, a+1, ..., b\}$. Further, we must have $\pi(b+1) \geq m+1$ if $b+1$ is a valid state, and $\pi(a-1) \leq m-1$ if $a-1$ is a valid state. We then construct $\psi(\pi)$ as follows. If $m < M$, we then let $\psi(\pi)(b) = m+1$. Otherwise (i.e., if $m = M$, we then let $\psi(\pi)(a) = m-1$. It is easy to show that the resulting $\psi(\pi)$ also satisfies part (3) of Condition 1, and thus $\psi(\pi) \in \Pi'$. According to Theorem 6, ψ is a valid operation. Moreover, (ψ, π) must belong to Ψ defined in Remark 1 because ψ satisfies statement (iii) of Remark 1. Therefore, ψ is a Ψ-valid operation. The result of Corollary 1 then follows.

E Proof for Theorem 9.

To begin with, we need to prove two lemmas.

Lemma 15. *Consider two policies $\pi_1 \in \Pi'$ and $\pi_2 \in \Pi'$, such that $\pi_1 \prec \pi_2$. Then, the following will hold:*

(1) for any state s such that $\pi_1(s) < m$, we must have $\pi_1(s) \leq \pi_2(s) < m$;

(2) for any state s such that $\pi_1(s) > m$, we must have $\pi_1(s) \geq \pi_2(s) > m$.

Proof. Since $\pi_1 \prec \pi_2$, it implies that π_2 can be obtained from π_1 by performing a sequence of Ψ-valid operations, with Ψ defined by Remark 1. We denote this sequence of Ψ-valid operations as $\psi^{(j)}, 1 \leq j \leq J$, and denote the intermediate policies as $\pi^{(j)} = \psi^{(j)}(\pi^{(j-1)}), 1 \leq j \leq J$, with $\pi^{(0)} = \pi_1$ and $\pi^{(J)} = \pi_2$. This sequence of operations is illustrated below:

$$\pi_1 \triangleq \pi^{(0)} \xrightarrow{\psi^{(1)}} \pi^{(1)} \xrightarrow{\psi^{(2)}} \pi^{(2)} \xrightarrow{\psi^{(3)}} \ldots \xrightarrow{\psi^{(J)}} \pi^{(J)} \triangleq \pi_2. \tag{33}$$

To prove part (1), we will use induction. The induction hypothesis is that, for any state $\hat{s}$ such that $\pi_1(\hat{s}) < m$, we will have $\pi_1(\hat{s}) \leq \pi^{(j)}(\hat{s}) < m$ for $0 \leq j \leq J$. The base step ($j = 0$) trivially holds because $\pi^{(0)}(\hat{s}) = \pi_1(\hat{s})$. For the induction step, we assume that $\pi_1(\hat{s}) \leq \pi^{(j)}(\hat{s}) < m$ holds for $j \leq J - 1$. We wish to show that the induction hypothesis also holds for $j + 1$, i.e., $\pi_1(\hat{s}) \leq \pi^{(j+1)}(\hat{s}) < m$. As $\pi^{(j+1)}$ is obtained from $\pi^{(j)}$ through a Ψ-valid operation, there will be only one state s on which $\pi^{(j)}$ and $\pi^{(j+1)}$ have different actions. If $s \neq \hat{s}$, then $\pi^{(j)}$ and we then have $\pi^{(j+1)}$ will have the same action and $\pi_1(\hat{s}) \leq \pi^{(j)}(\hat{s}) = \pi^{(j+1)}(\hat{s}) < m$. If $s = \hat{s}$, then $\pi^{(j+1)}(\hat{s})$ will not be equal to $\pi^{(j)}(\hat{s})$. Since $\pi^{(j)}(\hat{s}) < m$, the Ψ-valid operation ψ on $\pi^{(j)}$ must satisfy the statement (i) of Remark 1. Hence, $\pi^{(j)}(\hat{s}) < \psi(\pi^{(j)})(\hat{s}) = \pi^{(j+1)}(\hat{s}) < m$. We then also have $\pi_1(\hat{s}) \leq \pi^{(j)}(\hat{s}) < \pi^{(j+1)}(\hat{s}) < m$. Combining these two cases, we conclude that the induction hypothesis must also hold for J. It implies that our induction hypothesis must also hold for $j = J$, and thus $\pi_1(\hat{s}) \leq \pi^{(J)}(\hat{s}) = \pi_2(\hat{s}) < m$.

We can prove part (2) in a similar way. □

Lemma 16. *For two policies $\pi_1 \in \Pi'$ and $\pi_2 \in \Pi'$ such that $\pi_1(\bar{s}) = \pi_2(\bar{s}) = m$ for some state $\bar{s}$, $\pi_1 \prec \pi_2$ will hold if π_1 and π_2 satisfy both of the following conditions:*

(1) For any $s \in \mathcal{S}$ such that $\pi_1(s) < m$, π_2 satisfies $\pi_1(s) \leq \pi_2(s) < m$;

(2) For any $s \in \mathcal{S}$ such that $\pi_1(s) > m$, π_2 satisfies $\pi_1(s) \geq \pi_2(s) > m$.

Proof. Suppose that π_1 and π_2 satisfy the two conditions. In order to show $\pi_1 \prec \pi_2$, we only need to find a sequence of Ψ-valid operations (with Ψ defined in Remark 1), which when performed from π_1, will finally obtain π_2. Let us define state sets $S_{\neq}, A, B, C$ and D, where $S_{\neq} = \{s \mid \pi_1(s) \neq \pi_2(s)\}$, $A = \{s \mid \pi_1(s) < \pi_2(s) < m\}$, $B = \{s \mid \pi_1(s) > \pi_2(s) > m\}$, $C = \{s \mid \pi_2(s) < \pi_1(s) = m\}$, $D = \{s \mid \pi_1(s) = m < \pi_2(s)\}$. We first show that $S_{\neq} = A \cup B \cup C \cup D$. Towards this end, note that for any state $s \in A \cup B \cup C \cup D$, we must have $\pi_1(s) \neq \pi_2(s)$. Thus, $s \in S_{\neq}$, and we then have $S_{\neq} \supseteq A \cup B \cup C \cup D$. It remains to show that $S_{\neq} \subseteq A \cup B \cup C \cup D$. We divide into several cases. If $\pi_1(s) < m$, then by Condition (1), we will have $\pi_1(s) < \pi_2(s) < m$, which implies that $s \in A$. If $\pi_1(s) > m$, then by Condition (2), we will have $\pi_1(s) > \pi_2(s) > m$, which implies that $s \in B$. If $\pi_1(s) = m$, then $\pi_2(s)$ is either larger than or smaller than m, which implies that $s \in C \cup D$. Combining the three cases above, we must have $s \in A \cup B \cup C \cup D$. Then, we can conclude that $S_{\neq} \subseteq A \cup B \cup C \cup D$.

Since we know that $S_{\neq} = A \cup B \cup C \cup D$, a natural way to find the sequence of Ψ-valid operations that converts π_1 to π_2 is to have each Ψ-valid operation changes the action of one state in $S_{\neq}$ under policy π_1 to the action of that state under policy π_2. If we can make sure that each step still produces a new policy in Π', then each new policy will have fewer states whose action differs from π_2. Thus, this sequence of Ψ-valid operations will eventually produce π_2. Towards this end, we will first find a sub-sequence of Ψ-valid operations that change the action of the states in C, until we obtain a policy that has the same actions as π_2 on all states in C. Then, we will find a sub-sequence of Ψ-valid operations that change the action of the states in A, followed by D and B. Finally, we simply combine these four sub-sequences into one sequence. After performing this sequence of Ψ-valid operations from

π_1, we can obtain π_2. In the following proof, we only show how we find the sub-sequences for C and A, as finding the sub-sequences for D and B is quite similar.

Part I. Construct the sub-sequence of Ψ-valid operations that change the states in C.

Recall that $C = \{s \mid \pi_2(s) < \pi_1(s) = m\}$. We write the set C as $\{c_1, ...c_k\}$, with $c_1 < ... < c_k$. We then construct the first subsequence of operations ψ^{c_i} and policies $\pi^{c_i}, i = 1, ...$, as follows,

$$\pi_1 \stackrel{\triangle}{=} \pi^{c_0} \xrightarrow{\psi^{c_1}} \pi^{c_1} \xrightarrow{\psi^{c_2}} \pi^{c_2} \xrightarrow{\psi^{c_3}} \dots \xrightarrow{\psi^{c_k}} \pi^{c_k}, \tag{34}$$

where $\pi^{c_i} = \psi^{c_i}(\pi^{c_{i-1}})$. Specifically, we construct π^{c_i} such that $\pi^{c_i}(c_i) = \pi_2(c_i)$ and $\pi^{c_i}(s) = \pi^{c_{i-1}}(s)$ for $s \neq c_i$. We can see that each time after performing an operation, the new policy will have one fewer state on which its action differs from π_2 (i.e., $\pi^{c_{i-1}}(c_i) = m = \pi_1(c_i) \neq \pi_2(c_i)$ but $\pi^{c_i}(c_i) = \pi_2(c_i)$). Note that the order with which we changed the actions of states goes from the lowest state c_1 to the highest state c_k in C. Thus, we know that π^{c_i} must have the same actions as π_2 on $c_1, c_2, ..., c_i \in C$ and must have the same actions as π_1 on $c_{i+1}, ..., c_k \in C$. Further, π^{c_i} has the same actions as π_1 for state $s \notin \{c_1, c_2, ..., c_i\}$. We will next show that each intermediate policy π^{c_i} will belong to Π' and thus, each operation will satisfy the statement (iii) of remark 1. Therefore, these operations are all Ψ-valid operations. Part I is then completed.

We now show that π^{c_i} belongs to Π'. Recall that we have shown that $\pi^{c_i}(s) = \pi_2(s)$ for states $s \in \{c_1, ..., c_i\}$ and $\pi^{c_i}(s) = \pi_1(s)$ for all the states $s \notin \{c_1, ..., c_i\}$. To show $\pi^{c_i} \in \Pi'$, we just need to compare its actions at different states and verify part (3) of Condition 1. Below, we will focus on comparing the action at all possible states s with the action at states in $\{c_1, ..., c_k\}$. (The comparison for other state pairs can be done with another induction on i. We divide into several cases.

(case 1) $s < c_i$ and $s \in \{c_1, ..., c_{i-1}\}$: We can verify that $\pi^{c_i}(s) = \pi_2(s) \stackrel{(a)}{\leq} \pi_2(c_i) = \pi^{c_i}(c_i)$. Here, Step (a) is because $\pi_2 \in \Pi'$.

(case 2) $s < c_i$, $s \notin \{c_1, ..., c_{i-1}\}$ and $s \in A$: We can verify that $\pi^{c_i}(s) = \pi_1(s) \stackrel{(b)}{<} \pi_2(s) \stackrel{(a)}{\leq} \pi_2(c_i) = \pi^{c_i}(c_i)$. Here, Step (a) is because $\pi_2 \in \Pi'$ and Step (b) is because $s \in A$.

(case 3) $s < c_i$, $s \notin \{c_1, ..., c_{i-1}\}$ and $s \notin A$: We can verify that $\pi^{c_i}(s) = \pi_1(s) \stackrel{(d)}{=} \pi_2(s) \leq \pi_2(c_i) = \pi^{c_i}(c_i)$; Here, Step (d) and be explained as follows. We first show that in case 3 we must have $\pi_1(s) < m$. To see this, note that since $s < c_i$ and $\pi_1 \in \Pi'$, we must have $\pi_1(s) \leq \pi_1(c_i) = m$. To show that $\pi_1(s) < m$, we prove by contradiction. Assume on the contrary that $\pi_1(s) = m$. As $\pi_2 \in \Pi'$, we must have $\pi_2(s) \leq \pi_2(c_i) < m$ (since $c_i \in C$). Hence, we must have $s \in C$. Since $s < c_i$, s must be one of the states among $c_1, ..., c_{i-1}$. However, case 3 assumes that $s \notin \{c_1, ..., c_{i-1}\}$, which is a contradiction. Therefore, we must have $\pi_1(s) < m$. Next, according to Condition (1) of Lemma 16, we will have $\pi_1(s) \leq \pi_2(s) < m$. Further, notice that case 3 assumes $s \notin A$. Then, we must have $\pi_1(s) = \pi_2(s)$, and step (d) holds.

(case 4) $c_i < s < \bar{s}$: We can verify that $\pi^{c_i}(c_i) = \pi_2(c_i) < m = \pi_1(c_i) \stackrel{(e)}{\leq} \pi_1(s) = \pi^{c_i}(s)$. Here, Step (e) is because $\pi_1 \in \Pi'$.

(case 5) $s \geq \bar{s}$: We can verify that $\pi^{c_i}(c_i) = \pi_2(c_i) < m = \pi_1(\bar{s}) \overset{(a)}{\leq} \pi_1(s) = \pi^{c_i}(s)$. Here, Step (e) is because $\pi_1 \in \Pi'$.

Combining all of these cases, we conclude that $\pi^{c_i} \in \Pi'$ for $i = 0, ..., k$. In other words, the operation ψ^{c_i} satisfies the statement (iii) of Remark 1 and it is a Ψ-valid operation for all i. As a special case, we have $\pi^{c_k} \in \Pi'$.

Part II. Construct the sub-sequence of Ψ-valid operations that changes the states in A.

Recall that $A = \{s \mid \pi_1(s) < \pi_2(s) < m\}$. We write the set A as $\{a_1, ...a_k\}$, with $a_1 < ... < a_k$. From the analysis in Part I, we know that π^{c_k} takes the same actions as π_2 on states in C and $\pi^{c_k} \in \Pi'$. We also know that $\pi^{c_k} \neq \pi_2$ on A, B, D. Next, we construct the second subsequence of operations ψ^{a_i} and policies π^{a_i}:

$$\pi^{c_k} \triangleq \pi^{a_{k+1}} \xrightarrow{\psi^{a_k}} \pi^{a_k} \xrightarrow{\psi^{a_{k-1}}} \pi^{a_{k-1}} \xrightarrow{\psi^{a_{k-2}}} \dots \xrightarrow{\psi^{a_1}} \pi^{a_1}, \tag{35}$$

where $\pi^{a_i} = \psi^{a_i}(\pi^{a_{i+1}})$. Specifically, we construct π^{a_i} such that $\pi^{a_i}(a_i) = \pi_2(a_i)$ and $\pi^{a_i}(s) = \pi^{a_{i+1}}(s)$ for $s \neq a_i$. We can also see that each time after performing an operation, the new policy will have one fewer state on which it differs from π_2 (i.e., $\pi^{a_{i+1}}(a_i) = \pi_1(a_i) \neq \pi_2(a_i)$ but $\pi^{a_i}(a_i) = \pi_2(a_i)$). Note that the order with which we changed the actions of states goes from the highest state a_k to the lowest state a_1 in A. Thus, we know that π^{a_i} must have the same action as π_2 on $a_i, a_{i+1}, ..., a_k \in A$ and must have the same actions as π_1 on $a_1, ..., a_{i-1} \in A$. Further, combining the result in Part I, we know that π^{a_i} has the same actions as π_2 on $C \cup \{a_i, a_{i+1}, ..., a_k\}$ and π^{a_i} has the same actions as π_1 on the states that are not in $C \cup \{a_i, a_{i+1}, ..., a_k\}$.

Now, we will show that each intermediate policy also satisfies $\pi^{a_i} \in \Pi'$ for $i = 1, ..., k, k+1$. Before that, we first prove a lemma below.

Lemma 17. *For the intermediate policy π^{a_i}, a_{i-1} is the highest state among these states s such that $\pi^{a_i}(s) \neq \pi_2(s)$ and $s < \bar{s}$.*

Proof. We first show that the states in A and C must be smaller than $\bar{s}$. For any $s \in A$, we have $\pi_1(s) < \pi_2(s) < m$. As $\pi_1(s) < m = \pi_1(\bar{s})$, according to $\pi_1 \in \Pi'$, we must have $s < \bar{s}$. For any $s \in C$, we have $\pi_2(s) < m$. As $\pi_2(s) < m = \pi_2(\bar{s})$, according to $\pi_2 \in \Pi'$, we must have $s < \bar{s}$.

With the same method above, we can also show that the states in B and D must be larger than $\bar{s}$. Notice that $S_{\neq} = A \cup B \cup C \cup D$. That implies that A and C contain all the states on which π_1 has different actions with π_2 before $\bar{s}$.

We have already known that π^{a_i} has the same action as π_2 on $C \cup \{a_i, a_{i+1}, ..., a_k\}$ and has the same actions as π_1 on the complement. Hence, $a_1, ..., a_{i-1} \in A$ correspond to all the states $s < \bar{s}$ at which π^{a_i} and π_2 have different actions. Therefore, a_{i-1} is the highest state among these states s such that $\pi^{a_i}(s) \neq \pi_2(s)$ and $s < \bar{s}$. □

Now, we go back to show $\pi^{a_i} \in \Pi'$, we divide into several cases.

(case 1) $s < a_i$ and $s \in \{a_1, ..., a_{i-1}\} \subset A$. We can verify that $\pi^{a_i}(s) = \pi_1(s) < \pi_2(s) \leq \pi_2(a_i) = \pi^{a_i}(a_i)$;

(case 2) $s < a_i$ and $s \notin \{a_1, ..., a_{i-1}\}$. We can verify that $\pi^{a_i}(s) \overset{(a)}{=} \pi_1(s) \overset{(b)}{=} \pi_2(s) \leq \pi_2(a_i) = \pi^{a_i}(a_i)$. Here, to see why equality (a) holds, we first show that $a_k < c_1$, which means that the largest state in A is still smaller than the smallest state in C. Notice that $\pi_1(a_k) < m = \pi_1(c_1)$, as $\pi_1 \in \Pi'$, we must have $a_k < c_1$. Therefore, if $s < a_i$, s must not belong to $C \cup \{a_i, a_{i+1}, ..., a_k\}$. In other word, $\pi^{a_i}(s)$ remains the same with $\pi_1(s)$, thus equality (a) holds. Step (b) can be explained as follows. We have known that $A \cup C = \{a_1, ..., a_k\} \cup \{c_1, ..., c_k\}$ contains all the states which make π_1 and π_2 different before $\bar{s}$. Further, as $s < a_i$ and $s \notin \{a_1, ..., a_{i-1}\}$, which implies that $s < \bar{s}$ and $s \notin A \cup C$, then we must have $\pi_1(s) = \pi_2(s)$. Therefore, equality (b) holds.

(case 3) $a_i < s < \bar{s}$. We can verify that $\pi^{a_i}(a_i) = \pi_2(a_i) \leq \pi_2(s) \overset{(c)}{=} \pi^{a_i}(s)$. Here, equality (c) is based on Lemma 17. To see this, according to Lemma 17, a_{i-1} is the largest state that makes π^{a_i} and π_2 different before $\bar{s}$. Therefore, $\pi_2(s) = \pi^{a_i}(s)$ for $a_i < s < \bar{s}$.

(case 4) $s \geq \bar{s}$. We can verify that $\pi^{a_i}(a_i) = \pi_2(a_i) < m \leq \pi_1(s) \overset{(d)}{=} \pi^{a_i}(s)$. Here, equality (d) holds because π^{a_i} remains the same with π_1 on the states which are not in $C \cup \{a_i, ..., a_k\}$, including all of the states after $\bar{s}$.

Combining all of these cases, we conclude that, $\pi^{a_i} \in \Pi'$. Further, the operation ψ^{a_i} satisfies the statement (i) of Remark 1 and it is a Ψ-valid operation for all i. Also as a special case, we have $\pi^{a_1} \in \Pi'$.

Part III. Construct the sub-sequence of Ψ-valid operations that changes the states in D and B, and then combine the four sub-sequences to one sequence.

Now, π^{a_1} only has different actions with π_2 on states in D and B. Using similar methods, we can construct the third and the fourth subsequences of operations:

$$\begin{aligned} &\pi^{a_1} \triangleq \pi^{d_{k+1}} \xrightarrow{\psi^{d_k}} \pi^{d_k} \xrightarrow{\psi^{d_{k-1}}} \pi^{d_{k-1}} \xrightarrow{\psi^{d_{k-2}}} \dots \xrightarrow{\psi^{d_1}} \pi^{d_1}. \\ &\pi^{d_1} \triangleq \pi^{b_0} \xrightarrow{\psi^{b_1}} \pi^{b_1} \xrightarrow{\psi^{b_2}} \pi^{b_2} \xrightarrow{\psi^{b_3}} \dots \xrightarrow{\psi^{b_k}} \pi^{b_k}. \end{aligned} \tag{36}$$

where π^{d_1} only has different actions with π_2 on state sets B and π^{b_k} will have the same action with π_2 for all the states. In other words, $\pi_2 = \pi^{b_k}$! We can also show in a similar way that all of the intermediate policies will belong to Π' and all of these operations are Ψ-valid operations. Combining all of this, we can have,

$$\pi_1 = \pi^{c_0} \xrightarrow{\psi^{c_1}} \dots \xrightarrow{\psi^{c_k}} \pi^{c_k} \xrightarrow{\psi^{a_k}} \dots \xrightarrow{\psi^{a_1}} \pi^{a_1} \xrightarrow{\psi^{d_k}} \dots \xrightarrow{\psi^{d_1}} \pi^{d_1} \xrightarrow{\psi^{b_1}} \dots \xrightarrow{\psi^{b_k}} \pi^{b_k} = \pi_2. \tag{37}$$

which implies π_2 can be obtained by performing a sequence of Ψ-valid operations from π_1, thus $\pi_1 \prec \pi_2$. □

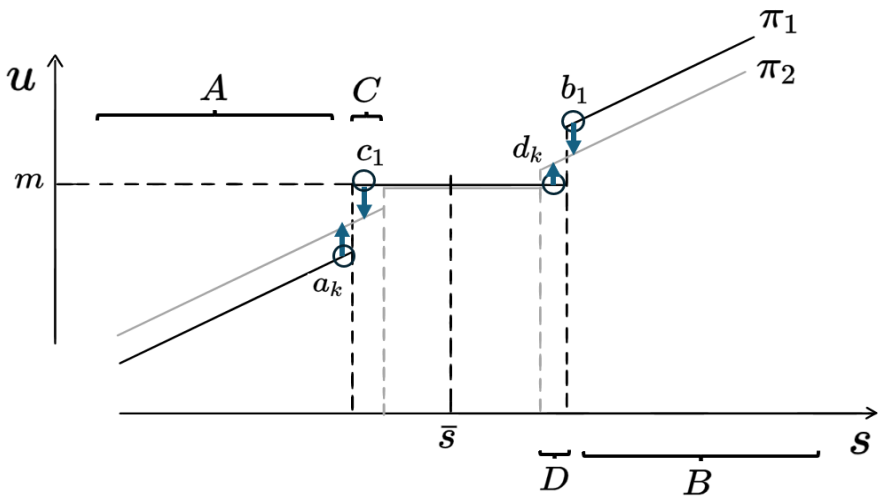


Figure 4: The operation for each part

E.1 Proof that Ψ satisfies Assumption 3

Suppose $\pi_1 \prec \pi_2$ and both π_1 and π_2 are in Π'. By definition, π_1 and π_2 are different, and hence $S_{\neq} = \{s \mid \pi_1(s) \neq \pi_2(s)\} \neq \emptyset$. Denote $A = \{s \mid \pi_1(s) < m, s \in S_{\neq}\} = \{a_1, ..., a_{k_A}\}$, $B = \{s \mid \pi_1(s) > m, s \in S_{\neq}\} = \{b_1, ..., b_{k_B}\}$ and $C = \{s \mid \pi_1(s) = m, s \in S_{\neq}\} = \{c_1, ..., c_{k_C}\}$. Clearly, we have $S_{\neq} = A \cup B \cup C$. We can assume that $a_1 < ... < a_{k_A}$, $b_1 < ... < b_{k_B}$ and $c_1 < ... < c_{k_C}$. As $S_{\neq}$ is not empty, it implies that A, B, and C cannot all be empty sets. We now divide our proof based on the sets A, B, and C.

If $A \neq \emptyset$, we construct a policy π' such that $\pi'(a_{k_A}) = \pi_2(a_{k_A})$ and $\pi'(s) = \pi_1(s)$ for $s \neq a_{k_A}$. In other words, we change the action of π_1 at the highest state a_{k_A} of A to that of π_2. As π' and π_1 only differ at one state a_{k_A}, π' can be regarded as being performed by an operation ψ on π_1, i.e., $\pi' = \psi(\pi_1)$. We can see that $\psi(\pi_1) = \pi'$ is a cross-over policy between π_1 and π_2. It only remains to show that ψ is a Ψ-valid operation. As $\pi_1(a_{k_A}) < m$ and $\pi_1 \prec \pi_2$, according to part (1) of Lemma 15, we have $\pi_1(a_{k_A}) \leq \pi_2(a_{k_A}) < m$. Further, $\pi_1(a_{k_A}) \neq \pi_2(a_{k_A})$ since $a_{k_A} \in S_{\neq}$, so we must have $\pi_1(a_{k_A}) < \pi_2(a_{k_A}) < m$. Therefore, $\pi_1(a_{k_A}) < \psi(\pi_1)(a_{k_A}) < m$. As a result, ψ satisfies item (i) of Remark 1 and is a Ψ-valid operation.

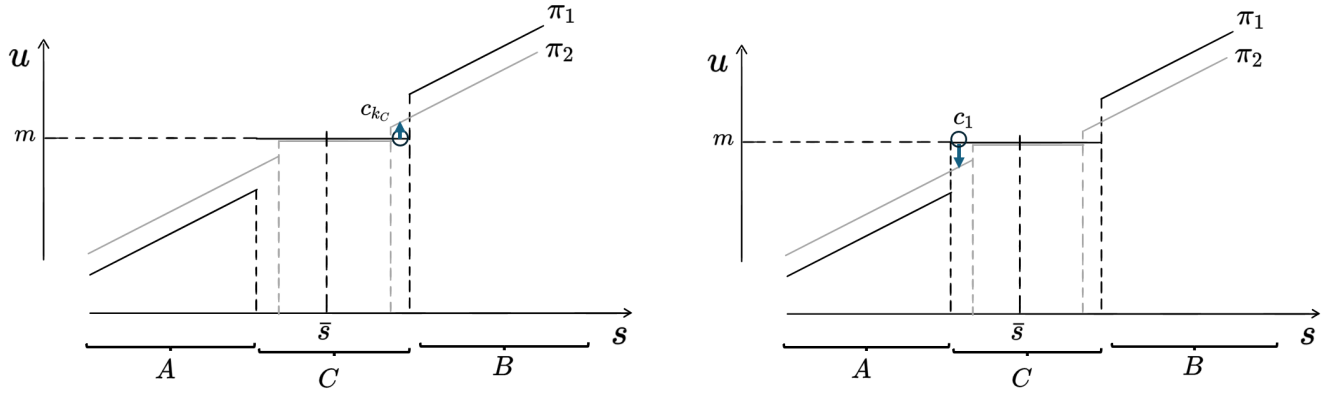
If $B \neq \emptyset$, the construction of the cross-over operation $\psi(\pi_1)$ through a Ψ-valid operation ψ is similar. (We just need to change the action of π_1 at state b_1 to that of π_2.)

If $C \neq \emptyset$, we further divide into two cases depending on the value of $\pi_2(c_{k_C})$ as follows. To begin with, we note that $\pi_2(c_{k_C})$ cannot be equal to m since $c_{k_C} \in S_{\neq}$.

Case 1: $\pi_2(c_{k_C}) > m$. We construct an operation $\psi(\pi_1) = \pi'$ such that $\pi'(c_{k_C}) = \pi_2(c_{k_C})$ and $\pi'(s) = \pi_1(s)$ for $s \neq c_{k_C}$. π' is clearly a cross-over between π_1 and π_2. We next prove that this operation ψ satisfies item (iii) of Remark 1, and thus is a Ψ-valid operation. Towards this end, it suffices to prove $\pi' \in \Pi'$, i.e., it satisfies part (3) of the Condition 1. Recall that π_1 and π_2 both belong to Π'. Consider any other states $s \neq c_{k_C}$. If $s < c_{k_C}$, then we have $\pi'(s) = \pi_1(s) \leq \pi_1(c_{k_C}) =$

$m < \pi_2(c_{k_C}) = \pi'(c_{k_C})$. If $c_{k_C} < s$, we first prove that $\pi_1(s) = m$ cannot happen. We prove by contradiction. Suppose on the contrary that $\pi_1(s) = m$. Note that we have already assumed that c_{k_C} is the largest state such that $\pi_1(c_{k_C}) = m$ and $\pi_2(c_{k_C}) \neq \pi_1(c_{k_C})$. Since $\pi_1(s) = m$ and $s > c_{k_C}$, we must have $\pi_1(s) = \pi_2(s)$. As $\pi_2 \in \Pi'$, we will get $\pi_1(s) = m < \pi_2(c_{k_C}) \leq \pi_2(s) = \pi_1(s)$, which is a contradiction. Therefore, $\pi_1(s) = m$ cannot happen. Further, since $\pi_1 \in \Pi'$ and $s > c_{k_C}$, we must have $\pi_1(s) > m$. Since $\pi_1 \prec \pi_2$, according to part (2) of Lemma 15, we must have $\pi_1(s) \geq \pi_2(s) > m$. Therefore, $\pi'(c_{k_C}) = \pi_2(c_{k_C}) \leq \pi_2(s) \leq \pi_1(s) = \pi'(s)$. In summary, $\psi(\pi_1) = \pi'$ must belong to Π'. We then confirm that ψ satisfies item (iii) of Remark 1, and hence ψ is a Ψ-valid operation.

Case 2: $\pi_2(c_{k_C}) < m$. We construct an operation $\psi(\pi_1) = \pi'$ such that $\pi'(c_1) = \pi_2(c_1)$ and $\pi'(s) = \pi_1(s)$ for $s \neq c_{k_C}$. We next also prove that this operation will satisfy item (iii) of Remark 1, and thus is a Ψ-valid operation. Towards that end, it suffices to prove $\pi' \in \Pi'$. As $\pi_2 \in \Pi'$ and we have already assumed $c_1 < c_2 < ... < c_{k_C}$, we must have $\pi_2(c_1) \leq \pi_2(c_{k_C}) < m$. Moreover, we recall that $\pi_1 \in \Pi'$. Consider any other states $s \neq c_1$. If $c_1 < s$, then $\pi'(c_1) = \pi_2(c_1) < m = \pi_1(c_1) \leq \pi_1(s) = \pi'(s)$. If $s < c_1$, we first prove that $\pi_1(s) = m$ cannot happen. We prove by contradiction. Suppose on the contrary that $\pi_1(s) = m$. Note that we have already assumed that c_1 is the lowest state such that $\pi_1(c_1) = m$ and $\pi_2(c_1) \neq \pi_1(c_1)$. Since $s < c_1$ and $\pi_1(s) = m$, we must have $\pi_1(s) = \pi_2(s)$. As $\pi_2 \in \Pi'$, we will get $\pi_1(s) = \pi_2(s) \leq \pi_2(c_1) < m = \pi_1(s)$, which is a contradiction. Therefore, $\pi_1(s) = m$ cannot happen. Further, since $\pi_1 \in \Pi'$ and $s < c_1$, we must have $\pi_1(s) < m$. As $\pi_1 \prec \pi_2$, according to Part (1) of Lemma 15, we must have $\pi_1(s) \leq \pi_2(s) < m$. Therefore, $\pi'(s) = \pi_1(s) \leq \pi_2(s) \leq \pi_2(c_1) = \pi'(c_1)$. In summary, $\psi(\pi_1) = \pi'$ must belong to Π'. We can then confirm that ψ satisfies item (iii) of Remark 1, and hence ψ is a Ψ-valid operation.



(a) The case of $\pi_2(c_{k_C}) > m$

(b) The case of $\pi_2(c_{k_C}) < m$

Figure 5: The illustration for E.1

E.2 Proof that Ψ satisfies Assumption 4

Fix $\vec{\lambda}_{-m}$. Suppose that $\pi_1 \in \Pi'$ and $\pi_2 \in \Pi'$ are optimal when $\lambda_m = \mu_1$ and $\lambda_m = \mu_2$ respectively, with $\mu_1 < \mu_2$. Further, suppose that $\pi_1(\bar{s}) = \pi_2(\bar{s}) = m$ for some state $\bar{s}$, and $\pi_1 \prec \pi_2$ is not true.

Let $A = \{s \mid \pi_2(s) < \pi_1(s) < m\}$, $B = \{s \mid m < \pi_1(s) < \pi_2(s)\}$, $C = \{s \mid \pi_1(s) < m \text{ and } \pi_2(s) = m\}$ and $D = \{s \mid m < \pi_1(s) \text{ and } \pi_2(s) = m\}$. As $\pi_1 \prec \pi_2$ is not true, we first show that A, B, C and D cannot all be empty. We prove by contradiction.

Assume on the contrary that sets A, B, C, and D are all empty sets. We first consider states s such that $\pi_1(s) < m$. By our assumption, we have $\pi_1(\bar{s}) = m$. Since $\pi_1 \in \Pi'$, i.e., it satisfies part (3) of Condition 1, we must have $s < \bar{s}$. Since $\pi_2(\bar{s}) = m$ and $\pi_2 \in \Pi'$, using Condition 1 again, we must have $\pi_2(s) \leq m$. Moreover, since set A is assumed to be an empty set, $\pi_2(s)$ must be larger or equal to $\pi_1(s)$; since C is also assumed to be an empty set, we must have $\pi_2(s) \neq m$. Therefore, $\pi_2(s)$ must satisfy $\pi_1(s) \leq \pi_2(s) < m$. Using a similar argument, for the states s such that $\pi_1(s) > m$, we can show that, if B and D are empty sets, then it implies that $\pi_2(s)$ must satisfy $\pi_1(s) \geq \pi_2(s) > m$. According to Lemma 16, π_1 must be before (earlier) than π_2, which contradicts to our assumption that $\pi_1 \prec \pi_2$ is not true. Therefore, we conclude that the sets A, B, C, and D cannot all be empty.

Next, we show that the sets C and D must be empty. We first prove by contradiction that C must be empty. Assume on the contrary that $C \neq \emptyset$. There must exist at least one state $c \in C$, such that $\pi_1(c) < m$ and $\pi_2(c) = m$. As π_2 is optimal when $\lambda_m = \mu_2$ and $\pi_2(c) = m$, we can know that $c \notin \mathcal{P}_m(\vec{\lambda}'')$ at $\vec{\lambda}'' = [\vec{\lambda}_{-m}, \mu_2]$. Note that if we can show that $c \in \mathcal{P}_m(\vec{\lambda}')$ at $\vec{\lambda}' = [\vec{\lambda}_{-m}, \mu_1]$, it will then contradict Assumption 1 that the passive set should only expand. We can then conclude that C must be empty. Unfortunately, we cannot draw the conclusion of $c \in \mathcal{P}_m(\vec{\lambda}')$ at $\vec{\lambda}' = [\vec{\lambda}_{-m}, \mu_1]$ directly from the assumption that π_1 is optimal when $\lambda_m = \mu_1$ and $\pi_1(c) \neq m$. That is because there may be another optimal policy when $\lambda_m = \mu_1$ whose action for state c is channel m. To overcome this difficulty, we consider two cases to construct a contradiction.

Case 1: π_1 is only optimal at a single point $\lambda_m = \mu_1$ and π_1 is not optimal at any other points. In this case, there must be two supporting optimal policies $\tilde{\pi}_{i-1}$ and $\tilde{\pi}_i$ that are optimal when $\lambda_m \in [\mu_1 - \epsilon, \mu_1]$ and $\lambda_m \in [\mu_1, \mu_1 + \epsilon]$, respectively, where $\epsilon > 0$ is a small amount that also satisfies $\mu_1 + \epsilon < \mu_2$. As $c \notin \mathcal{P}_m(\vec{\lambda}'')$ and $\mu_1 + \epsilon < \mu_2$, according to Assumption 1 (the partial indexability holds), we must have $c \notin \mathcal{P}_m(\vec{\lambda}''')$ at $\vec{\lambda}''' = [\vec{\lambda}_{-m}, \mu_1 + \epsilon]$. Thus $\tilde{\pi}_i(c)$ must be m. Then, we construct a policy π' such that $\pi'(c) = \pi_1(c) \neq m$ and $\pi'(s) = \tilde{\pi}_i(s)$ for $s \neq c$. It is easy to see that π' is a crossover between policy π_1 and $\tilde{\pi}_i$. As π_1 and $\tilde{\pi}_i$ are both optimal when $\lambda_m = \mu_1$, according to Lemma 2, π' is also optimal when $\lambda_m = \mu_1$. Further, as π' is an optimal policy, π' must belong to Π'. According to Theorem 6, we must have $T_m^{\pi'} < T_m^{\tilde{\pi}_i}$, which implies that $\tilde{\pi}_i$ can no longer be the supporting optimal policy when $\lambda_m \in [\mu_1, \mu_1 + \epsilon]$, which is a contradiction. Therefore, under Case 1, we can conclude that C must be empty.

Case 2: π_1 is optimal at more than one point. According to Lemma 1, π_1 must be one of the supporting optimal policies. We suppose that π_1 is optimal when $\lambda_m \in [\tilde{\lambda}_{i-1}, \tilde{\lambda}_i]$ and μ_1 is a point in this closed interval. Thus, we have $\tilde{\lambda}_{i-1} \leq \mu_1 < \mu_2$. Note that π_1 may not be the only supporting optimal policy when $\lambda_m \in [\tilde{\lambda}_{i-1}, \tilde{\lambda}_i]$, and there may be other policies which are also optimal in this interval. We first show by contradiction that there must be at least one other policy π_3 such that π_3 is also an supporting optimal policy when $\lambda_m \in [\tilde{\lambda}_{i-1}, \tilde{\lambda}_i]$ and $\pi_3(c) = m$. Assume on the contrary that

this statement is not true, which implies that for any other supporting optimal policies at $[\tilde{\lambda}_{i-1}, \tilde{\lambda}_i]$, the optimal action for state c is not m. Therefore, $c \in \mathcal{P}_m(\vec{\lambda}''')$ at $\vec{\lambda}''' = [\vec{\lambda}_m, \mu_3]$, where μ_3 is any point between $(\tilde{\lambda}_{i-1}, \min(\mu_2, \tilde{\lambda}_i))$. Combined with $c \notin \mathcal{P}_m(\vec{\lambda}'')$ at $\vec{\lambda}'' = [\vec{\lambda}_{-m}, \mu_2]$, Assumption 1 is then violated, which is a contradiction. Therefore, there must be another supporting optimal policy π_3 that is also optimal when $\lambda_m \in [\tilde{\lambda}_{i-1}, \tilde{\lambda}_i]$ and meanwhile $\pi_3(c) = m$. Using a similar method as that in case 1, we can also construct a policy π' such that $\pi'(c) = \pi_1(c) \neq m$ and $\pi'(s) = \pi_3(s)$ for $s \neq c$. π' is a crossover policy between policy π_1 and π_3. As π_1 and π_3 are both optimal policies for $\lambda_m \in [\tilde{\lambda}_{i-1}, \tilde{\lambda}_i]$, according to Lemma 2, π' is also optimal when $\lambda_m \in [\tilde{\lambda}_{i-1}, \tilde{\lambda}_i]$. Therefore, $\pi' \in \Pi'$. According to Theorem 6, $T_m^{\pi'} < T_m^{\pi_3}$. Then π_1 and π_3 cannot be supporting optimal policies anymore, which is also a contradiction. Thus, under Case 2, we conclude that $C \neq \emptyset$ as well.

Combining these cases together, we conclude that C must be the empty set. Using a similar argument, D must also be an empty set.

Now that we have shown that C and D are both empty sets, we conclude that A and B cannot both be empty. If $A \neq \emptyset$, we pick a state $a \in A$. We construct an operation $\psi(\pi_2) = \pi'$ such that $\pi'(a) = \pi_1(a)$ and $\pi'(s) = \pi_2(s)$ for $s \neq a$. Then, π' is a cross-over policy between π_1 and π_2. Further, we have $\pi_2(s) \leq \pi'(s) < m$ for all the states such that $\pi_2(s) < m$. According to statement (i) of Remark 1, $\psi(\pi_2) = \pi'$ is then the Ψ-valid operation required in Assumption 4. The case of $B \neq \emptyset$ can be verified similarly.

F Some details about computing the complexity

We first show that B is upper-bounded by $2M$, i.e., the number of operations in $\Gamma_1(\pi)$ is at most $2M$ during any iteration of our Algorithm 1. Denote the active set for each channel $u, u = 0, 1, ..., M$ under policy π as A_u^π . A_m^π is non-empty because, if A_m^π is empty, which means that $\mathcal{P}_m = \mathcal{S}$, then we should have terminated Algorithm 1. Suppose that other than $A_m^\pi \neq \emptyset$, the non-empty active sets are $A_{c_1}^\pi, ..., A_{c_p}^\pi, A_{d_1}^\pi, ..., A_{d_q}^\pi$, with $c_1 < ... < c_p < m < d_1 < ... < d_q$. Notice that, for $\psi(\pi)$ to be in $\Gamma_1(\pi) \subset \Pi'$, it must differ from π at only one state $\bar{s}$. Further, the state $\bar{s}$ must be either the largest state in $A_{c_1}^\pi, ..., A_{c_p}^\pi, A_m^\pi$ or the smallest state in $A_m^\pi, A_{d_1}^\pi, ...A_{d_q}^\pi$. Otherwise, $\psi(\pi)$ cannot be in Π'. If we pick state $\bar{s}$ being the largest state in $A_{c_i}^\pi$, $i = 1, ..., p-1$, there are $c_{i+1} - c_i$ different Ψ-valid operations ψ corresponding to $\psi(\pi)(\bar{s})$ being $c_i + 1, ..., c_{i+1}$. If $\bar{s}$ is the largest state in $A_{c_p}^\pi$, there are $m - c_p - 1$ Ψ-valid operations. Due to similar reasons, there are $d_i - d_{i-1}$ different Ψ-valid operations if $\bar{s}$ is the smallest state in $A_{d_i}^\pi$ for $i = 2, ..., q$, and there are $d_1 - m - 1$ Ψ-valid operations if $\bar{s}$ is the smallest state in $A_{d_1}^\pi$. If state $\bar{s}$ is in A_m^π, $\bar{s}$ must be either the largest or the smallest state in A_m^π. We denote them as state a and state b, respectively. $\psi(\pi)(a)$ can be $c_p, ..., m-1$ and $\psi(\pi)(b)$ can be

$m+1, ..., d_1$. Thus, B can be bounded by

$$\begin{aligned} B &\leq \sum_{i=1}^{p-1}(c_{i+1}-c_i) + (m-c_p-1) + \sum_{i=2}^{q}(d_i-d_{i-1}) + (d_1-m-1) + (m-c_p) + (d_1-m) \\ &= (m-1-c_1) + (d_q-m-1) + (m-c_p) + (d_1-m) \\ &= (d_q-c_1) + (d_1-c_p) \\ &< M+M = 2M. \end{aligned}$$

Next, we show that A is upper-bounded by $MK+1$. i..e, the total number of iterations is upper-bounded by $MK+1$. After the A-th iteration, the passive set is the whole state space $\mathcal{S}$ and we stop the algorithm. For $\pi^{(1)}, ..., \pi^{(A-1)}$, there still exists some state for which the optimal action is channel m. We define a function $f(k) = \sum_s [\pi^{(k)}(s) 1_{\{\pi^{(k)}(s)<m\}} + (M-\pi^{(k)}(s)) 1_{\{\pi^{(k)}(s)>m\}}]$, which is a function of the iteration number k. We prove that $f(k+1) > f(k)$, and therefore $f(k)$ is a strictly increasing function. We will then be able to bound the number of iterations by the maximum value of $f(\cdot)$. Towards this end, Suppose that $\pi^{(k+1)} = \psi(\pi^{(k)})$. Due to the definition of Ψ-valid operation in Remark 1, there are three cases:

(Case 1) ψ satisfies (i) of Remark 1. Then, there is a state $\bar{s}$ such that $\pi^{(k)}(\bar{s}) < \pi^{(k+1)}(\bar{s}) < m$. Thus, we have $\pi^{(k+1)}(\bar{s}) 1_{\{\pi^{(k+1)}(\bar{s})<m\}} > \pi^{(k)}(\bar{s}) 1_{\{\pi^{(k)}(\bar{s})<m\}}$, and other terms in $f(k+1)$ coincide with those of $f(k)$. Therefore, $f(k+1) > f(k)$.

(Case 2) ψ satisfies (ii) of Remark 1. Then, there is a state $\bar{s}$ such that $\pi^{(k)}(\bar{s}) > \pi^{(k+1)}(\bar{s}) > m$. Thus, we have $(M-\pi^{(k+1)}(\bar{s})) 1_{\{\pi^{(k+1)}(\bar{s})>m\}} > (M-\pi^{(k)}(\bar{s})) 1_{\{\pi^{(k)}(\bar{s})>m\}}$, and other terms in $f(k+1)$ coincide with those of $f(k)$. Therefore, $f(k+1) > f(k)$. Therefore, $f(k+1) > f(k)$.

(Case 3) ψ satisfies (iii) of Remark 1. Then, there is a state $\bar{s}$ such that $\pi^{(k)}(\bar{s}) = m$ and $\pi^{(k+1)}(\bar{s}) \neq m$. Thus, we have $\pi^{(k)}(\bar{s}) 1_{\{\pi^{(k)}(\bar{s})<m\}} + (M-\pi^{(k)}(\bar{s})) 1_{\{\pi^{(k)}(\bar{s})>m\}} = 0$, but one of these two items is positive for $\pi^{(k+1)}$. Again, other terms of $f(k+1)$ coincide with those of $f(k)$. Therefore, $f(k+1) > f(k)$.

In summary, no matter which Ψ-valid operation is chosen in Line 6 of Algorithm 1, we have $f(k+1) > f(k)$, i.e., $f(k)$ is a strictly increasing function. Since $f(k) \geq 0$ for all k, the total number of iterations can thus be bounded by the maximum value of $f(\cdot)$ plus 1. For each state s, $\pi^{(k)}(s) 1_{\{\pi^{(k)}(s)<m\}} + (M-\pi^{(k)}(s)) 1_{\{\pi^{(k)}(s)>m\}}$ is less than M. Since there are K states, we must have $f(k) < MK$. Therefore, we obtain $A-1 < MK$ and $A < MK+1$.