Networked Restless Bandits with Positive Externalities

Christine Herlihy, John P. Dickerson

Department of Computer Science
University of Maryland, College Park
College Park, MD, USA
cherlihy@umd.edu, johnd@umd.edu

Abstract

Restless multi-armed bandits are often used to model budget-constrained resource allocation tasks where receipt of the resource is associated with an increased probability of a favorable state transition. Prior work assumes that individual arms only benefit if they receive the resource directly. However, many allocation tasks occur within communities and can be characterized by positive externalities that allow arms to derive partial benefit when their neighbor(s) receive the resource. We thus introduce *networked restless bandits*, a novel multi-armed bandit setting in which arms are both restless and embedded within a directed graph. We then present GRETA, a graph-aware, Whittle index-based heuristic algorithm that can be used to efficiently construct a constrained reward-maximizing action vector at each timestep. Our empirical results demonstrate that GRETA outperforms comparison policies across a range of hyperparameter values and graph topologies. Code and appendices are available at https://github.com/crherlihy/networked_restless_bandits.

1 Introduction

We study the planning task of allocating budget-constrained indivisible resources so as to maximize the expected amount of time that members of a cohort will spend in a desirable state (e.g., adherent to a prescribed exercise regimen). Restless multi-arm bandits (RMABs) are well-suited for such tasks, as they represent each individual as a Markov decision process (MDP) whose stochastic state transitions are governed by an action-dependent transition function.

Conventionally, an arm must receive the resource at time t to derive any benefit from it, where benefit takes the form of an increased probability of transitioning to the desirable state at time $t + 1$ (i.e., relative to non-receipt). However, many resource allocation tasks *occur within communities* and can be characterized by *positive externalities* that allow arms to derive partial, indirect benefit when their neighbor(s) receive the resource. We consider chronic disease management programs as a motivating example. These programs often combine resource-constrained physician support with less cost-intensive, more scalable peer support to encourage participants to make lifestyle modifications. To this end, we introduce *networked restless bandits*, a novel multi-armed bandit setting in which arms are both restless and embedded within a directed graph. We then present a graph-aware, Whittle-based heuristic algorithm that is constrained reward-maximizing in this setting. Our core contributions include:

(i) Our networked restless bandit model, which lets us represent topological relationships between arms, and associate arm i's receipt of a pull with positive externalities for its neighbors.

(ii) GRETA, a graph-aware, Whittle index-based heuristic algorithm that lets us efficiently construct a constrained reward-maximizing mapping from arms to actions at each timestep.

(iii) Empirical results which demonstrate that GRETA outperforms comparison policies across a range of hyperparameter values and graph topologies.

1.1 Related Work

Restless bandits: The restless multi-armed bandit (RMAB) framework was introduced by Whittle (1988) as a way to model the sequential allocation of a budget-constrained, indivisible resource over a population of N dynamic arms, where: (1) at most $k \ll n$ arms can receive the resource (i.e., a pull) at any given timestep; and (2) the state of each arm evolves over time, regardless of whether or not it is pulled. We provide a formal description in Section 2.1.

Indexability: In the general case, it is PSPACE-hard to pre-compute the optimal policy for a given cohort of restless arms (Papadimitriou and Tsitsiklis 1994). However, as conjectured by Whittle (1988) and proven by Weber and Weiss (1990), when each arm is indexable, a tractable solution exists that is provably asymptotically optimal: we can decouple the arms and consider a Lagrangian relaxation of the original problem. In this context, the Whittle index can be thought of as the infimum subsidy required to make an arm indifferent between a pull and passivity, given its current state. Whittle-index based policies use these index values to rank arms when selecting which k arms to pull.

Proving indexability can be difficult and often requires the problem instance to satisfy specific structural properties, such as the optimality of threshold policies (Liu and Zhao 2010). Additionally, much of the foundational work in this space focuses on the two-action setting, and cannot be directly extended to the multi-action setting that we consider.

Glazebrook, Hodge, and Kirkbride (2011) do consider the multi-action setting, but for divisible rather than indivisible resources; they also require an arm to consume this resource at a level that is decreasing in the resource charge. Killian, Perrault, and Tambe (2021) study multi-action restless bandits and do not make any of the structural assumptions required to verify indexability, but instead develop a Lagrangian bound-minimization approach; however, they do not consider relationships among arms.

Mate et al. (2020) introduce the collapsing bandit model, and demonstrate that this problem is indexable when forward or reverse threshold policies are optimal. They also introduce an efficient, closed-form approach to computing the Whittle index called THRESHOLD WHITTLE (TW), and empirically demonstrate that this approach performs well even when optimality conditions are not satisfied. We leverage TW as a subroutine within GRETA.

Bandits and graphs: Prior work at the intersection of multi-armed bandits and graphs has tended to focus on *stochastic*, rather than restless arms, and on *graph-structured feedback* (i.e., rewards), rather than the embedding of arms within a directed graph, and/or the spillover effects associated with allocation in the face of adjacency. For example, Valko (2016) examines a graph structure among *actions* in *stochastic* bandits, and Lu, Hu, and Zhang (2021) examines a graph structure over *rewards*. However, we examine a graph structure among *arms* in the restless bandit setting.

In recent work, Ou et al. (2022) look at a mobile intervention setting. Similarly to our model, they combine the traditional restless bandit setting with network externalities; however, their model and goal are fundamentally different. Their arms represent locations on a network with pulls impacting a mixture of subpopulations that are located at or near that pull, probabilistically. In contrast, in our model, vertices represent individual arms, and our algorithm exploits—when advantageous—the propensity for allocating a high-cost, high-benefit resource to one arm to unlock potential lower-cost, intermediate-benefit resources for the arm's neighbors.

2 Model Preliminaries

2.1 Restless Multi-arm Bandits

The restless bandit (RMAB) setting features an agent with $n \in \mathbb{N}$ arms. The state of each arm evolves over time and in response to the agent's actions, in a way that is governed by the arm's underlying Markov decision process (MDP). Each MDP is defined by a state space, $\mathcal{S}$, an action space, $\mathcal{A}$, a cost function, $C : \mathcal{A} \to \mathbb{R}$, a local reward function, $r : \mathcal{S} \to \mathbb{R}$, and a transition function, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$. The objective is to find a policy, $\pi : \mathcal{S} \to \mathcal{A}$, that maximizes total expected discounted reward over a finite time horizon, T—i.e., $\pi^* = \arg\max_\pi E_\pi [R(\cdot)]$. The agent must select exactly one action per arm at each timestep, and the associated costs must not exceed the per-timestep budget, $B \in \mathbb{R}_{\geq 0}$.

2.2 Motivating Example

For ease of exposition, we ground our networked restless bandit model in a *motivating example*: let arms represent patients striving to adhere to a chronic disease management program, such as an exercise regimen. A patient's "state" on any given day is thus determined by whether they adhere (i.e., exercise), or fail to adhere to their regimen. To encourage adherence, many such programs feature a combination of resource-constrained physician- and peer support (Fisher et al. 2017). Examples include, but need not be limited to, a reminder call from a physician, a supportive message from a fellow participant, or the provision of awareness-raising outreach materials. Thus, a coordinator seeking to maximize the number of patients who exercise over the program's duration might select a small subset of patients each day to receive a call from a physician, and ask these people to message a handful of their peers in turn, or pass along an educational pamphlet to their caregiver(s). In each case, the lower-cost, easier-to-scale information dissemination option amplifies physician outreach, allowing a broader subset of individuals to receive partial benefit.

2.3 Networked Restless Bandits

With this motivating example in mind, we now introduce our networked restless bandit model, which allows us to model directed relationships among arms. Given a set of n arms, let $G = (V, E)$ be a directed graph, and let there exist a bijective mapping φ from arms to vertices — i.e., $\forall v \in V,\ \exists!\ i \in [n]$ s.t. $\varphi(i) = v$. Let a directed edge, $e \in E$, exist between arms u and v if it is possible for v to benefit indirectly when u receives a pull. Let $\mathcal{N}_{\text{in}}(u) = \{v \in V \mid \exists e_{v,u} \in E\}$ and $\mathcal{N}_{out}(u) = \{v \in V \mid \exists e_{u,v} \in E\}$ represent u's one-hop indegree and outdegree neighborhoods, respectively. This graph is assumed to be constructed or operated by the agent; as such, it is assumed to be observable. Real-life examples with this property include mentoring programs and online social networks.

State space: We consider a discrete state space, $\mathcal{S} := \{0, 1\}$, where the states admit a total ordering by desirability, such that state 1 is more "desirable" than state 0. In our example, state 0 represents non-adherence to the exercise regimen, while state 1 represents adherence. We assume each arm's state is observable (e.g., via fitness tracker data).

Action space: The traditional restless bandit setting considers a binary action space, $\mathcal{A} := \{0, 1\}$, where 1 (or 0) represents the decision to pull (or not pull) arm i at time t. To model positive externalities, we define an extended action space, $\mathcal{A} := \{0 : \texttt{no-act}, 1 : \texttt{message}, 2 : \texttt{pull}\}$. Here, actions 0 and 2 correspond to the actions *don't pull* and *pull* respectively. We note that our `message` action need not represent a literal message. Instead, it represents an intermediate action with respect to desirability that gets "unlocked" as an available action for vertex v at time t only when some vertex $u \in \mathcal{N}_{\text{in}}(v)$ receives a pull at time t.

Transition function: For each arm $i \in [n]$, let $P^{a,i}_{s,s'}$ represent the probability that arm i will transition from state s to s' given action a. In the offline setting, these transition matrices are assumed to be static and known to the agent at planning time. This assumption is reasonable when historical data from the same or similar population(s) provides a source for informative priors, as is common in many domains, in-

cluding healthcare and finance (Steimle and Denton 2017; Pasanisi, Fu, and Bousquet 2012). Extension to the online setting where transition matrices must be learned is possible via Thompson sampling (Thompson 1933; Ortner et al. 2012; Jung and Tewari 2019; Jung, Abeille, and Tewari 2019).

We assume nonzero transition matrix entries, and impose two sets of domain-motivated **structural constraints** (Mate et al. 2020): (i) $\forall a \in \mathcal{A}, P^a_{0,1} < P^a_{1,1}$ and (ii) $\forall (a, a') \in \mathcal{A} \times \mathcal{A}, a < a' \rightarrow P^a_{0,1} < P^{a'}_{0,1}; P^a_{1,1} < P^{a'}_{1,1}$. Constraint set (i) implies that each arm is more likely to stay in the desirable state (i.e., $s = 1$) than transition there from the undesirable state (i.e., $s = 0$). Constraint set (ii) implies that messages and pulls are beneficial when received and that a strict preference relation over actions can be defined for each arm, such that `no-act` $\prec$ `message` $\prec$ `pull`.

Cost function: We map our action space to the cost vector $\vec{c} = [0, \psi, 1]$, where $0 \leq \psi < 1$. Intuitively, this mapping preserves standard notion that no cost is incurred when an arm does not receive any form of intervention. It also encodes the idea that the more beneficial an action is, the more expensive it is to provide, which motivates us to exploit positive externalities. Additionally, when there are no edges, i.e., $E = \emptyset$, and no messages can be sent, the unit cost of a pull lets us recover the original restless bandit setting, where we must choose which $k \ll n$ arms to pull at each timestep.

Objective and constraints: It is possible, though not tractable at scale, to take a constrained optimization-based approach to solving for the optimal policy, π^*. We build on Killian, Perrault, and Tambe (2021)'s approach below to show how our constrained setting can be modeled. To begin, let $\mathbf{s}$ represent a vector containing the state of each arm, i.e. $[s^i \in \mathcal{S} | i \in [n]]$, and let $\mathbf{X}$ represent a matrix containing binary decision variables, one for each of n arms and $|\mathcal{A}|$ actions. We require our local reward function, $r : \mathcal{S} \rightarrow \mathbb{R}$ to be non-decreasing in s, which is consistent with our goal of maximizing the expected time that each arm spends in the "desirable" state. Equation 1 formalizes our task:

$$
\begin{aligned}
J(\mathbf{s}) = \max_{\mathbf{X}} & \left\{ \sum_{i=0}^{n-1} r^i(s^i) + \beta \mathbb{E}[J(\mathbf{s}'), \mathbf{X}] \right\} \\
\text{subject to} \quad & \sum_{i=0}^{n-1} \sum_{j=0}^{|\mathcal{A}|-1} x_{i,j} \cdot c_j \leq B \\
& x_{i,1} \leq \bigvee_{i' \in \mathcal{N}_{\text{in}}(i)} x_{i',2} && \forall i \in [n] \\
& \sum_{j=0}^{|\mathcal{A}|-1} x_{i,j} = 1 && \forall i \in [n] \\
& \mathbf{X} \in \{0,1\}^{n \times |\mathcal{A}|}
\end{aligned}
\tag{1}
$$

Our goal is to find assignments of the decision variables contained in $\mathbf{X}$ such that expected discounted reward is maximized, subject to a series of feasibility constraints: (i) across all actions and arms, do not expend more than B budget; (ii) ensure that if `message` is chosen for an arm i, then that arm has at least one indegree neighbor i' such that `pull` was chosen; and, (iii) ensure that each arm receives exactly one action at each timestep. However, two challenges arise: (1) a direct solution via value iteration is exponential in n, and (2) Lagrangian relaxation-based approaches rely on the decoupling of arms, which jeopardizes the satisfaction of our neighborhood constraint on actions. This motivates us to propose a graph-aware, Whittle-based heuristic algorithm.

3 Algorithmic Approach

Here, we introduce GRETA, a graph-aware, Whittle-index-based heuristic algorithm that can be used to efficiently construct a constrained reward-maximizing policy. A key insight that GRETA exploits is that while we cannot decouple arms in the networked setting, since we must know whether any of an arm's indegree neighbors will receive a pull at time t to know if the arm is eligible to receive a message, we *can* compute two sets of Whittle indices for each arm, by considering each active action as a separate instance of a two-action problem. Note that the structural constraints ensure that for a given state, an arm will require a higher subsidy to forgo a pull as opposed to a message. We can then construct an augmented graph that allows us to compare the cumulative subsidy required for the arms represented by directed edge (u, v) to forgo a *pull* and *message*, respectively to those required by other directed edges $\in G$ (including, importantly, the inverse action-pair implied by edge (v, u)).

3.1 GRETA: A Graph-aware Heuristic

Set-up: We begin by building an augmented graph, G'. This graph contains every vertex and edge in G, along with a dummy vertex, -1, and directed edge $(u, -1)\ \forall u \in V$. This lets us map each directed edge (u, v) in G to the action pair (`pull`, `message`), and $(u, -1)$ to (`pull`, `no-act`). We also construct an augmented arm set, $[n] \cup \{-1\}$, and extend our bijective mapping from arms to vertices such that $\varphi : -1 \mapsto -1$. Appendix A.1 provides pseudocode.

Next, we pre-compute the Whittle index for each vertex-active action combination $(v, \alpha) \in V' \times \mathcal{A} \setminus \{0\}$. When we compute the Whittle index for a given (v, α) pair, we seek the infimum subsidy, m, required to make arm i (i.e., $\varphi^{-1}(v)$) indifferent between passivity (i.e., `no-act`) and receipt of action α at time t (Whittle 1988). We cannot compute the Whittle index for our placeholder -1 vertex because it is not attached to an MDP, so we map it to 0.

Algorithm 1: Compute Whittle indices for $V' \times \mathcal{A} \setminus \{0\}$

procedure WHITTLE($V', \alpha \in \{1, 2\}, \varphi$)

$$\lambda := i \mapsto \begin{cases} 0, & \text{if } i = -1 \\ \inf_m \{m \mid V_m(s^i_t, a^i_t = 0) \geq V_m(s^i_t, a^i_t = \alpha)\}, & \text{otherwise} \end{cases}$$

return $W_\alpha \leftarrow \{\lambda \circ \varphi^{-1}(v) \mid v \in V'\}$

$$V_m(s^i_t) = \max \begin{cases} m + r(s^i_t) + \beta V_m\left(s^i_{t+1}\right) & \textit{no-act} \\ r(s^i_t) + \beta[s^i_t V_m\left(P^\alpha_{1,1}\right) + (1 - s^i_t) V_m\left(P^\alpha_{0,1}\right)] & \alpha \end{cases} \tag{2}$$

The value function represents the maximum expected discounted reward that arm $i \in [n]$ with state s^i_t can receive at

time t given a subsidy m, discount rate β, and active action $\alpha \in \{1, 2\}$.

GRETA: With our augmented graph and Whittle index values in hand, we now present our algorithm. We provide pseudocode in Algorithm 2, and structure our exposition sequentially. At each timestep $t \in T$, GRETA takes as *input*: (1) an augmented set of restless arms, $[n] \cup \{-1\}$ embedded in an augmented directed graph, $G' = (V', E')$; (2) a budget, $B \in \mathbb{R}$; (3) a cost function, $C : \mathcal{A} \to \mathbb{R}$; (4) a message cost, $\psi \in [0, 1)$; and (5) a set of Whittle index values per active action $\alpha \in \{1, 2\}$, denoted by W_1 and W_2, respectively. Given these inputs, GRETA *returns* a reward-maximizing, constraint-satisfying action vector, $\vec{a}_t$.

Algorithm 2: GRETA: graph-aware, Whittle-based heuristic
Note: all sorts are descending; arrays are zero-indexed.

1: **procedure** GRETA($G', V', E', B, C, \psi, W_1, W_2$)
2: $\quad \vec{a}_t \leftarrow 0^{|V|}$
3: $\quad B' \leftarrow B$
4: $\quad$ **while** $\vee_{e \in E'}$ GETCOST$(u, v, \vec{a}_t, C) \leq B' \wedge E' \neq \emptyset$ **do**
5: $\quad\quad \mathtt{b} \leftarrow \min(B', 2)$
$\quad\quad$ ▷ Consider only pulls
6: $\quad\quad \hat{a}_2, \nu_2 \leftarrow$ PULLONLY$(E', \lfloor \mathtt{b} \rfloor, W_2)$
$\quad\quad$ ▷ Consider pulls *and* messages
7: $\quad\quad \hat{a}_{(1,2)}, \nu_{(1,2)}, E'_\oslash \leftarrow$ MP$(G', \mathtt{b}, C, \psi, \vec{a}_t, W_1, W_2)$
$\quad\quad$ ▷ Select max-val candidate actions; update $\vec{a}_t, B', G'$
8: $\quad\quad$ **if** $\nu_2 \geq \nu_{(1,2)}$ **then**
9: $\quad\quad\quad \vec{a}_t, B' \leftarrow$ MODACTSB$(G', C, \hat{a}_2, \vec{a}_t, B')$
10: $\quad\quad\quad E', G' \leftarrow$ UPDATEG$(V', E', \hat{a}_2, \emptyset)$
11: $\quad\quad$ **else**
12: $\quad\quad\quad \vec{a}_t, B', \leftarrow$ MODACTSB$(G', C, \hat{a}_{(1,2)}, \vec{a}_t, B')$
13: $\quad\quad\quad E', G' \leftarrow$ UPDATEG$(V', E', \hat{a}_{(1,2)}, E'_\oslash)$
14: $\quad$ **return** $\vec{a}_t$

In lines 2-3 of Algorithm 2, we initialize $\vec{a}_t$ such that each vertex is mapped to 0 (`no-act`), and set our *remaining budget* variable, B', equal to the per-timestep budget, B.

In lines 4-13, we iteratively update our action vector $\vec{a}_t$ until we have insufficient remaining budget to afford any available edge-action pair, or our augmented edge set, $E' = \emptyset$. The termination check in line 4 requires us to: (1) check if we've already incurred the cost of a *pull* or *message* (*message*) for vertex u (v); and (2) offset accordingly when we compute the cost of $(a_t^u = 2, a_t^v = 1)$, per Alg. 3.

Algorithm 3: Compute cost to pull u and message v

1: **procedure** GETCOST$(u, v, \vec{a}_t, C)$
2: $\quad c_u \leftarrow C(2)(1 - \mathbb{1}(a_t^u > 0)) + \mathbb{1}(a_t^u = 1)(C(2) - C(1))$
3: $\quad c_v \leftarrow C(1)(1 - \mathbb{1}(a_t^v = 1 \vee v = -1))$
4: **return** $c_u + c_v$

The subroutines called in lines 6-7 of GRETA serve to ensure that we will only deviate from the pull-assignment choices of graph-agnostic THRESHOLD WHITTLE—i.e., by considering a combination of pulls *and* messages—when it is strictly beneficial to do so.

Since pulls have unit cost, and $\psi \in [0, 1)$, we consider our per-timestep budget in sequential chunks of 2. We have two options for allocating each chunk over actions: (1) considering *only* pulls, and selecting the two arms with highest W_2 index values; or (2) considering messages *and* pulls, and selecting the set of directed (u, v) edges with highest edge-level subsidies such that each u receives a pull, and each v (excluding -1) receives a message. In lines 8-13, we select the candidate action set with the highest cumulative subsidy, and update $\vec{a}_t, B',$ and G' accordingly.

Pulls only: Allocation option (1) maps arms who have yet to receive a pull at time t to candidate actions $\in \{0, 2\}$ by sorting their W_2 index values in descending order and selecting the top-2 arms to receive pulls. App. A.1 gives pseudocode (Alg 7).

Messages *and* pulls: Allocation option (2) maps arms to candidate actions by computing an edge index value for each directed edge $\in E'$. Algorithm 4 provides pseudocode.

Algorithm 4: Cumulative subsidy of max pull-message set
Note: all sorts are descending; arrays are zero-indexed.

1: **procedure** MP$(G', \mathtt{b} \in \mathbb{R}, C, \psi, \vec{a}_t, W_1, W_2)$
2: $\quad G'' = (V'', E'') \leftarrow G'$
3: $\quad \hat{a}_{(1,2)} : v \in V'' \mapsto \vec{a}_t^v$
4: $\quad f : (u, v) \in E'' \mapsto \mathbb{R}$
5: $\quad E'_\oslash \leftarrow \emptyset$
6: $\quad \nu_{(1,2)} \leftarrow 0$
7: $\quad$ **while** $\vee_{e \in E''}$ GETCOST$(u, v, \hat{a}, C) \leq \mathtt{b} \wedge E'' \neq \emptyset$ **do**
8: $\quad\quad$ **for** $u \in V'' \setminus \{-1\}$ **do**
9: $\quad\quad\quad \mathcal{N}'_{\text{out}}(u) \leftarrow \{v | (u, v) \in E'' \wedge \hat{a}_{(1,2)}^v = 0\}$
10: $\quad\quad\quad$ EDGEINDICES$(f'', u, \mathcal{N}'_{\text{out}}(u), \mathtt{b}, \psi, W_1, W_2)$
11: $\quad\quad$ `values` $\leftarrow$ SORT$(\{f((u, v)) | (u, v) \in E''\})$
12: $\quad\quad$ **if** $|\mathtt{values}| = 0$ **then**
13: $\quad\quad\quad$ **break**
14: $\quad\quad$ **for** $f((u, v)) \in \mathtt{values}$ **do**
15: $\quad\quad\quad \mathtt{cost}_{u,v} \leftarrow$ COMPUTECOST$(u, v, \hat{a}_{(1,2)}, C)$
16: $\quad\quad\quad$ **if** $\mathtt{cost}_{u,v} \leq \mathtt{b}$ **then**
17: $\quad\quad\quad\quad h : u \mapsto 2; v \mapsto 1$
18: $\quad\quad\quad\quad \hat{a}_{(1,2)}, \mathtt{b} \leftarrow$ MODACTSB$(G'', C, h, \hat{a}_{(1,2)}, \mathtt{b})$
19: $\quad\quad\quad\quad E'', G'' \leftarrow$ UPDATEG$(V'', E'', \hat{a}_{(1,2)}, \emptyset)$
20: $\quad\quad\quad\quad \nu_{(1,2)} \mathrel{+}= f((u, v))$
21: $\quad\quad\quad\quad E'_\oslash \leftarrow E'_\oslash \cup \{(u, v)\}$
22: $\quad\quad\quad\quad$ **break**
$\quad\quad$ ▷ Return best arm-actions, cumulative subsidy, $E'_\oslash$
23: $\quad$ **return** $\hat{a}_{(1,2)}, \nu_{(1,2)}, E'_\oslash$

In line 2 of Algorithm 4, we start by defining G'' to be a local copy of our augmented graph, G'. We then create a function, $\vec{a}_{(1,2)}$ to map each vertex $v \in V''$ to its candidate action, which we initialize to be $\vec{a}_t^v$ (line 3). We do this because we require the current G' to determine which $(\mathtt{pull}_u, \mathtt{message}_v)$ edge-action combinations are possible, and for $\vec{a}_t$ to correctly compute the cost of these hypothetical actions, but we don't want to modify $\vec{a}_t$ or G' in-place. Next, in lines 4-5, we define a function, f that maps each edge $(u, v) \in E'$ to a real-valued edge index value, and a set, $E'_\oslash$, to hold the edges we will need to remove from G' if we select the candidate actions returned by Algorithm 4. In line 6, we initialize $\nu_{(1,2)} = 0$ to represent the cumulative subsidy of our candidate action set.

In lines 7-22 of Algorithm 4, we iteratively update our

candidate action function, $\hat{a}_{(1,2)}$, until we run out of (small-$\texttt{b}$) budget, or $E'' = \emptyset$. Inside each iteration of the WHILE-loop, we begin by computing an edge index value for each directed edge $(u, v) \in E'$ (lines 8-10). To do this, we loop over vertices in $V' \setminus \{-1\}$ (line 8), and for each vertex u, let $\mathcal{N}'_{\text{out}}(u) \subseteq \mathcal{N}_{\text{out}}(u)$ represent the subset of u's one-hop out-degree neighbors currently slated to receive a `no-act` at time t.

For each edge $(u, v) \in \mathcal{N}'_{\text{out}}(u)$, our edge index value represents the cumulative subsidy required to forgo a *pull* for arm u (i.e., W_2^u) and a *message* for arm v (i.e., W_1^v). Note: if we pull u, message v, and have budget left over, we can message up to $|\mathcal{M}_t^u|$ vertices $v' \in \mathcal{N}'_{\text{out}}(u)$ at time t *without* incurring additional pull costs, where $|\mathcal{M}_t^u| = |\mathcal{N}'_{\text{out}}(u)|$ if $\psi = 0$ and $\min(\lfloor \texttt{b}/\psi \rfloor, |\mathcal{N}'_{\text{out}}(u)|)$ for $\psi \in (0, 1)$.

To exploit this diminishing marginal cost, we sort u's, neighbors by their index-values and let the max-valued edge represent the cumulative, cost-feasible value of $\mathcal{N}'_{\text{out}}(u)$, rather than just (u, v). Algorithm 5 provides pseudocode.

Algorithm 5: Compute edge index values
Note: all sorts are descending; arrays are zero-indexed.

```
1:  procedure EDGEINDICES(f, u, N'_out(u), b, ψ, W_2, W_1)
2:      n_msgs ← |N'_out(u)| if ψ = 0 else min(⌊b/ψ⌋, |N'_out(u)|)
3:      msg_values ← SORT(g : v ∈ N'_out(u) ↦ W_1^v)
4:      max_edge ← (u, arg max_v msg_values)
5:      for v ∈ N'_out(u) do
6:          if (u, v) = max_edge then
7:              f((u, v)) ← W_2^u + Σ_{i=0}^{n_msgs−1} msg_values_i
8:          else
9:              f((u, v)) ← W_2^u + W_1^v
10:     return                                  ▷ f is updated in-place
```

Then, in lines 11-13 of Algorithm 4, we sort the edge-index values in descending order. Note that we break if there are no values to be sorted; this corresponds to the scenario in which no additional pulls are available/cost-feasible, and every arm not receiving a pull is already receiving a message, but we still have budget left—i.e., when $\psi = 0$. In lines 14-22, we choose the top cost-feasible edge-action pair from our sorted list, and update our candidate action function, $\vec{a}_{(1,2)}$ and local budget, $\texttt{b}$ accordingly. Note that if $\psi = 0$ and arm u receives a pull, we message every $v \in \mathcal{N}'_{\text{out}}(u)$. App. A.1 provides pseudocode for the MODACTSB subroutine (see Algorithm 8 in Appendix A.1).

Finally, we update our local copy of the augmented graph by removing (u, v), as well as any directed edge terminating in u, and the placeholder edge, $(u, -1)$. This is because: (a) we do not want to reconsider the edge-action pair we've selected; and (b) by virtue of how we select (u, v), $f((u, v)) \geq f((\cdot, u))$ *or* any such u-terminating edge is cost-prohibitive. App. A.1 provides pseudocode for the UPDATEG subroutine (see Algorithm 9). We conclude the MP subroutine (Algorithm 4) by returning our candidate action function, $\hat{a}_{(1,2)}$, the associated cumulative subsidy value, $\nu_{(1,2)}$, and the set of candidate edges to be removed from G', $E'_{\oslash}$.

Putting the pieces together: With the exposition of each of GRETA's subroutines complete, we now return to lines 8-13 of Algorithm 2. We compare the cumulative subsidy values returned by the PULLONLY and MP subroutines, and use the candidate action function associated with the maximum cumulative subsidy to update our action vector, $\vec{a}_t$, remaining budget, B', and augmented graph, G'. When the WHILE-loop terminates, we return $\vec{a}_t$. By virtue of how this action vector is constructed, it is reward-maximizing and guaranteed to satisfy the budget constraint.

3.2 Theoretical Analysis

Bounding expected reward: Per Theorem 3.1, the expected cumulative reward of GRETA with message cost, $\psi > 0$, will be lower-bounded by that of graph-agnostic THRESHOLD WHITTLE, and upper-bounded by GRETA with $\psi = 0$. See Appendix A.2 for a complete proof.

Theorem 3.1. *For a given set of $[n]$ restless or collapsing arms with transition matrices satisfying the structural constraints outlined in Section 2.3, corresponding directed graph, $G = (V, E)$, budget $B \in \mathbb{R}_{\geq 0}$, non-decreasing local reward function, $r : \mathcal{S} \to \mathbb{R}$, cumulative reward function, R, and cost vector $\vec{c} = [0, \psi, 1]$ such that $\psi \in [0, 1)$, we have:* $\mathbb{E}_{\text{TW}}[R] \leq \mathbb{E}_{\text{GH},\psi>0}[R] \leq \mathbb{E}_{\text{GH},\psi=0}[R]$

Proof Sketch. The *first inequality* follows from how GRETA constructs each $\vec{a}_t$. The *second inequality* follows from the fact that: (a) per our structural constraints and choice of r, $E[r_t^i | s_t^i, a_t^i]$ is strictly increasing with a_t^i $\forall i, t$; and (b) for $\psi = 0$, we can message at least as many arms as when $\psi > 0$. □

Computational complexity: Per Theorem 3.2, GRETA is efficiently computable in time polynomial in its inputs; see Appendix A.2 for a complete proof.

Theorem 3.2. *For convenience, let:* $\xi = \mathbb{1}(\psi > 0) \times \min(|E'|^2, \lfloor \frac{B}{\psi} \rfloor |E'|) + \mathbb{1}(\psi = 0) \times |V'||E'|$. *Then, for $\psi \in [0, 1)$ and time horizon, T, the time complexity of* GRETA *is:*

$$\begin{cases} O\left(\max\left(\xi^2 |V'|^2 \log |V'|,\ \xi^2 |V'||E'|^2\right) T\right), & \text{if } \psi > 0 \\ O\left(\max\left(\xi^2 |V'|^2 \log |V'|,\ \xi^2 |V'||E'|^2,\ \xi^2 |V'|^2 |E'|\right) T\right), & \text{otherwise} \end{cases}$$

These bounds indicate that GRETA is well-suited for sparse graphs and values of $\psi = 0$ or $\psi \to 0.5$ ($\psi > 0.5$ will also improve runtime, but may reduce opportunities to exploit externalities). Conversely, pathological cases will include large-scale dense graphs and values of the message cost, ψ, which approach but do not equal 0. We consider improving scale to be a valuable direction for future work. The combinatorial nature of the problem we consider suggests that sampling and/or distributed methods will be critical in this regard (Zhou et al. 2020; Almasan et al. 2022).

4 Experimental Evaluation

In this section, we demonstrate that GRETA consistently outperforms a set of robust graph-agnostic and graph-aware comparison policies. We begin by identifying the set of policies we compare against, as well as our evaluation metrics, graph generation, and mapping of arms to vertices. We proceed to present results from three experiments: (1) GRETA

versus the optimal policy (for small n); (2) GRETA versus comparison policies for a fixed cohort and graph; and (3) GRETA evaluated on a series of different budgets, message costs, and graph topologies.

4.1 Experimental Setup

Policies: In our experiments, we compare the policy produced by GRETA against a subset of the following graph-{agnostic$^{\oslash}$ and aware†} policies:

THRESHOLD WHITTLE (TW)$^{\oslash}$	Compute Whittle index values using pull as (only) active action. Pull $\lfloor B \rfloor$ arms with highest Whittle index values; all others get no-act (Whittle 1988; Mate et al. 2020).
RANDOM†	Construct G'; select budget-feasible edge-action pairs uniformly at random until budget exhausted.
CENTRALITY-WEIGHTED RANDOM†	Construct G'; select budget-feasible edge-action pairs weighted by out-degree centrality of `src` vertex until budget exhausted.
MYOPIC†	Construct G'; sort edge-action pairs by expected reward at $t+1$. Select cost-feasible pairs until budget exhausted.
VALUE ITERATION (VI) †	Find the optimal policy via value iteration for *system-level* MDP (intractable at scale, but computable for small $\lvert V \rvert$ and $\lvert E \rvert$).

Table 1: Comparison policies

We note that in the restless (but graph-agnostic) setting: (1) RANDOM and MYOPIC are common baselines. Here, we have extended them to the networked setting. (2) THRESHOLD WHITTLE represents a state-of-the-art approach. To the best of our knowledge, no additional (efficiently computable) graph-aware policies exist for the novel networked restless bandit setting we propose.

Objective: Our optimization task is consistent with assigning equal value to each timestep that any arm spends in the "desirable" state. This motivates our choice of a local reward function $r_t(s_t^t) := s_t^i \in \{0,1\}$ and undiscounted cumulative reward function $R(r(s)) := \sum_{i\in[N]} \sum_{t\in[T]} r(s_t^i)$.

Intervention benefit (IB): For each policy, we compute total expected reward, $\mathbb{E}_\pi[R(\cdot)]$, by taking the average over 50 simulation seeds. We then compute the intervention benefit as defined in Equation 3, where NOACT represents a policy in which no pulls or messages are executed, and GH represents the policy produced by GRETA.

$$\text{IB}_{\text{NoAct,GH}}(\pi) := \frac{\mathbb{E}_\pi[R_\pi(\cdot)] - \mathbb{E}_{\text{NoAct}}[R(\cdot)]}{\mathbb{E}_{\text{GH}}[R(\cdot)] - \mathbb{E}_{\text{NoAct}}[R(\cdot)]} \tag{3}$$

Graph generation: For each cohort of n restless arms that we consider in our experiments, we use a stochastic block model (SBM) to generate a graph with $|V| = n$ vertices (Holland, Laskey, and Leinhardt 1983). This generator partitions the vertices into blocks and stochastically inserts directed edges, with hyperparameter $p_{in}(p_{out}) \in [0,1]$ controlling the probability that a directed edge will exist between two vertices in the same (different) block(s).

We consider two options for $\varphi : [n] \to V$: (1) *random*; and (2) *by cluster*. For mapping (1), we generate $\lceil \frac{n}{10} \rceil$ blocks of uniform size, and map arms to vertices—and, by extension, to blocks—uniformly at random. This mapping represents allocation settings with a peer support component where participants are randomly assigned to groups, without regard for their behavioral similarity.

For mapping (2), we use an off-the-shelf K-MEANS algorithm to cluster the arms in flattened transition-matrix vector space (Pedregosa et al. 2011). We use the cardinality of the resulting clusters to determine the size of each block and map arms to vertices based on cluster membership. This mapping represents intervention allocation settings with a peer support component where participants with similar transition dynamics are grouped together.

4.2 GRETA vs. the Optimal Policy

In this experiment, we compare GRETA to π^*_{VI}, where π^*_{VI} denotes the optimal policy obtained via value iteration for the *system-level* MDP (Sutton and Barto 2018). This system-level MDP has state space $\mathcal{S}' := \{\mathcal{S}\}^n$, action space $\mathcal{A}' := \{\mathcal{A}\}^n$, a transition function, $P : \mathcal{S}' \times \mathcal{A}' \to \mathcal{S}'$, and reward function, $R' = \sum_{i\in[n]} s^i$. To ensure budget and neighborhood constraint satisfaction, only cost- and topologically feasible actions, $\mathcal{A}'' \subseteq \mathcal{A}'$ are considered. Figure 1 reports results for a synthetic cohort of 8 arms embedded in a fully connected graph (i.e., $p_{\text{in}} = p_{\text{out}} = 1.0$). We let $T = 120$, $\psi = 0.5$, and report unnormalized $\mathbb{E}_\pi[R]$, along with margins of error for 95% confidence intervals computed over 50 simulation seeds for values of $B \in \{1, 1.5, 2, 2.5, 3\}$. GRETA outperforms TW for each value of B (with predictably larger gaps for values with remainders $= \psi$ that graph-agnostic TW cannot exploit), and is competitive with respect to π^*_{VI}.

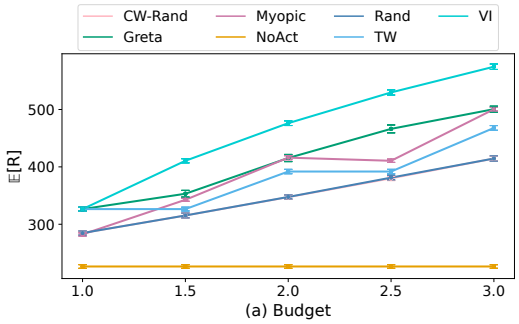

Figure 1: $\mathbb{E}[R]$ by policy and budget

4.3 GRETA vs. Alternative Policies

Here we compare GRETA to the graph-agnostic and graph-aware comparison policies outlined in Section 4.1. We consider a synthetic cohort of $n = 100$ restless arms whose transition matrices are randomly generated in such a way so as to satisfy the structural constraints introduced in Section 2. We use a stochastic block model (SBM) generator with $p_{in} = 0.2$ and $p_{out} = 0.05$, and consider both the *random* and *by cluster* options for φ. We let $T = 120$, $B = 10$, and $\psi = 0.5$.

In Table 2, we report results for each mapping-policy combination, along with margins of error for 95% confidence intervals computed over 50 simulation seeds.

$\varphi(i)$	Policy	$\mathbb{E}$[IB] (%) ($\pm$)
randomly	RANDOM	75.82 $\pm$ 0.890
	CWRANDOM	74.79 $\pm$ 1.068
	MYOPIC	87.83 $\pm$ 1.115
	TW	83.57 $\pm$ 0.779
	GRETA	**100.00** $\pm$ **0.000**
by cluster	RANDOM	64.19 $\pm$ 0.786
	CWRANDOM	63.59 $\pm$ 0.804
	MYOPIC	76.24 $\pm$ 0.921
	TW	72.65 $\pm$ 0.684
	GRETA	**100.00** $\pm$ **0.000**

Table 2: $\mathbb{E}$[IB] by choice of φ and policy

Key findings from this experiment include:

- The policy produced by GRETA achieves significantly higher $\mathbb{E}_\pi$[IB] than each of the comparisons.
- The gap in $\mathbb{E}_\pi$[IB] between GRETA and MYOPIC, which is the best-performing alternative, is larger for the *by cluster* mapping than the *random* mapping. This suggests that in assortative networks, relatively homogeneous transition dynamics within blocks facilitate the exploitation of the diminishing marginal costs associated with the pull-message dynamic.

4.4 Sensitivity Analysis

We conduct sensitivity analysis with respect to: (1) the budget, B; (2) the message cost, ψ; and (3) the underlying graph topology, via the p_{in} and p_{out} hyperparameters of our stochastic block model graph generator. As we vary each of the aforementioned hyperparameters, we consider a fixed cohort size of $n = 100$ randomly-generated, structural constraint-satisfying arms, a time horizon, $T = 120$, and a mapping $\varphi : i \in [n] \mapsto v \in V$ from arms to vertices that is determined by cluster. We report unnormalized $\mathbb{E}_\pi[R]$, along with margins of error for 95% confidence intervals computed over 50 simulation seeds, for GRETA, THRESHOLD WHITTLE, NOACT, and MYOPIC, which is the best-performing non-TW alternative. We describe each task below, and present results in Figure 2.

Budget: We hold message cost fixed at $\psi = 0.5$, let $p_{\text{in}} = 0.25$, $p_{\text{out}} = 0.05$, and consider values of $B \in \{5\%, 10\%, 15\%\}$ of n. As Figure 2(a) illustrates, $\mathbb{E}_\pi[R]$ intuitively rises with B for each policy considered. For each value of B, we find that GRETA achieves higher $\mathbb{E}_\pi[R]$ than the comparison policies and that the gap between GRETA and the best-performing alternative also increases with B.

Message cost: Here, we hold the budget fixed at 6, let $p_{\text{in}} = 0.25$, $p_{\text{out}} = 0.05$, and consider values of $\psi \in \{0.0, 0.25, 0.5, 0.75, 0.9\}$. As Figure 2(b) illustrates, $\mathbb{E}_\pi[R]$ decreases as the message cost, ψ, increases for GRETA and MYOPIC, while it remains constant for active-action agnostic NOACT and message-agnostic TW. For each value of ψ that we consider, GRETA achieves higher $\mathbb{E}_\pi[R]$ than each of the comparison policies. This gap is intuitively largest when $\psi = 0$, and decreases until GRETA converges with TW—notably, without suffering loss in total expected reward due to divisibility issues with respect to B, when $\psi = 0.75$.

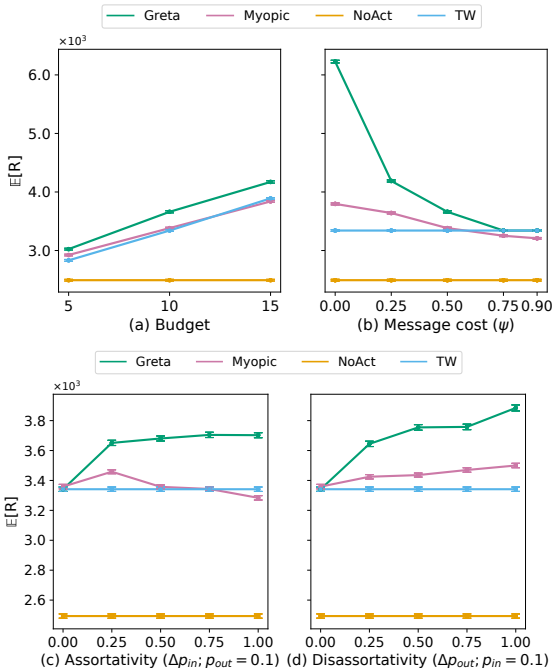

Figure 2: Sensitivity results, by varied hyperparameter

Graph topology: We hold the budget fixed at $B = 10$, let message cost, $\psi = 0.5$, and consider two sets of increasingly *assortative* (*disassortative*) $(p_{\text{in}}, p_{\text{out}})$ ordered pairs. In each case, we start with $E = \emptyset$—i.e., $(0.0, 0.0)$, and then hold p_{out} (p_{in}) fixed at 0.1 and steadily increase p_{in} (p_{out}). Figure 2(c) and (d) present results. For GRETA, while $\mathbb{E}_\pi[R]$ is generally increasing in the number of edges, the rate of growth levels off as assortativity rises but remains robust as disassortativity rises. This suggests that homophilic clustering of arms with respect to transition dynamics may undermine total welfare by inducing competition within neighborhoods, while heterophilic clustering can help to smooth out subgroups' relative demand for constrained resources over time.

5 Conclusion & Future Work

In this paper, we introduce networked restless bandits, a novel multi-armed bandit setting in which arms are restless and embedded in a directed graph. We show that this framework can be used to model constrained resource allocation in community settings, where receipt of the resource by an individual can result in spillover effects that benefit their neighbor(s). We also present GRETA, a graph-aware, Whittle-based heuristic algorithm which is constrained reward-maximizing and budget-constraint satisfying in our networked restless bandit setting. Our empirical results demonstrate that the policy produced by GRETA outperforms a set of graph-agnostic and graph-aware comparison policies for a range of different budgets, message costs, and graph topologies. Future directions include: (1) relaxing the assumption of perfect observability of transition matrices and/or graph topology; (2) considering individual and/or group fairness; and (3) incorporating sampling and/or distributed methods to improve scalability.

Acknowledgments

We were supported by NSF CAREER Award IIS-1846237, NSF D-ISN Award #2039862, NSF Award CCF-1852352, NIH R01 Award NLM013039-01, NIST MSE Award #20126334, DARPA GARD #HR00112020007, DoD WHS Award #HQ003420F0035, ARPA-E Award #4334192, ARL Award W911NF2120076 and a Google Faculty Research Award. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of U.S. government or funding agencies. We thank Samuel Dooley, Pranav Goel, Aviva Prins, Dr. Philip Resnik, and our anonymous reviewers for their helpful input and feedback.

References

Almasan, P.; Suárez-Varela, J.; Rusek, K.; Barlet-Ros, P.; and Cabellos-Aparicio, A. 2022. Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case. *Computer Communications*, 196: 184–194.

Fisher, E. B.; Boothroyd, R. I.; Elstad, E. A.; Hays, L.; Henes, A.; Maslow, G. R.; and Velicer, C. 2017. Peer support of complex health behaviors in prevention and disease management with special reference to diabetes: systematic reviews. *Clinical Diabetes and Endocrinology*, 3(1): 4.

Glazebrook, K. D.; Hodge, D. J.; and Kirkbride, C. 2011. General notions of indexability for queueing control and asset management. *The Annals of Applied Probability*, 21(3): 876 – 907.

Holland, P. W.; Laskey, K. B.; and Leinhardt, S. 1983. Stochastic blockmodels: First steps. *Social Networks*, 5(2): 109–137.

Jung, Y. H.; Abeille, M.; and Tewari, A. 2019. Thompson Sampling in Non-Episodic Restless Bandits. *CoRR*, abs/1910.05654.

Jung, Y. H.; and Tewari, A. 2019. Regret Bounds for Thompson Sampling in Episodic Restless Bandit Problems. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Killian, J. A.; Perrault, A.; and Tambe, M. 2021. Beyond "To Act or Not to Act": Fast Lagrangian Approaches to General Multi-Action Restless Bandits. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, 710–718. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073.

Liu, K.; and Zhao, Q. 2010. Indexability of Restless Bandit Problems and Optimality of Whittle Index for Dynamic Multichannel Access. *IEEE Transactions on Information Theory*, 56(11): 5547–5567.

Lu, S.; Hu, Y.; and Zhang, L. 2021. Stochastic Bandits with Graph Feedback in Non-Stationary Environments. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10): 8758–8766.

Mate, A.; Killian, J.; Xu, H.; Perrault, A.; and Tambe, M. 2020. Collapsing Bandits and Their Application to Public Health Intervention. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 15639–15650. Curran Associates, Inc.

Ortner, R.; Ryabko, D.; Auer, P.; and Munos, R. 2012. Regret Bounds for Restless Markov Bandits. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, ALT'12, 214–228. Berlin, Heidelberg: Springer-Verlag. ISBN 9783642341052.

Ou, H.-C.; Siebenbrunner, C.; Killian, J.; Brooks, M. B.; Kempe, D.; Vorobeychik, Y.; and Tambe, M. 2022. Networked Restless Multi-Armed Bandits for Mobile Interventions. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, 1001–1009. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

Papadimitriou, C. H.; and Tsitsiklis, J. N. 1994. The Complexity of Optimal Queueing Network Control. In *Proceedings of IEEE 9th Annual Conference on Structure in Complexity Theory*, 318–322. IEEE.

Pasanisi, A.; Fu, S.; and Bousquet, N. 2012. Estimating Discrete Markov Models from Various Incomplete Data Schemes. *Computational Statistics & Data Analysis*, 56(9): 2609–2625.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Steimle, L. N.; and Denton, B. T. 2017. *Markov Decision Processes for Screening and Treatment of Chronic Diseases*, 189–222. Cham: Springer International Publishing. ISBN 978-3-319-47766-4.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book. ISBN 0262039249.

Thompson, W. R. 1933. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3/4): 285–294.

Valko, M. 2016. *Bandits on graphs and structures*. Habilitation à diriger des recherches, École normale supérieure de Cachan - ENS Cachan.

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

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

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81.