PSINET: Aiding HIV Prevention Amongst Homeless Youth by Planning Ahead

A. Yadav, L. S. Marcolino, E. Rice, R. Petering, H. Winetrobe, H. Rhoades, M. Tambe
{amulyaya, sorianom, ericr, petering, hwinetro, hrhoades, tambe}@usc.edu
University of Southern California, Los Angeles, CA, 90089

H. Carmichael
hcarmichael@myfriendsplace.org
LCSW, Executive Director, My Friend's Place, Los Angeles, CA 90028

Abstract

Homeless youth are prone to Human Immunodeficiency Virus (HIV) due to their engagement in high risk behavior such as unprotected sex, sex under influence of drugs, etc. Many non-profit agencies conduct interventions to educate and train a select group of homeless youth about HIV prevention and treatment practices and rely on word-of-mouth spread of information through their social network. Previous work in strategic selection of intervention participants does not handle uncertainties in the social network's structure and evolving network state, potentially causing significant shortcomings in spread of information. Thus, we developed PSINET, a decision support system to aid the agencies in this task. PSINET includes the following key novelties: (i) it handles uncertainties in network structure and evolving network state; (ii) it addresses these uncertainties by using POMDPs in influence maximization; and (iii) it provides algorithmic advances to allow high quality approximate solutions for such POMDPs. Simulations show that PSINET achieves ∼60% more information spread over the current state-of-the-art. PSINET was developed in collaboration with My Friend's Place (a drop-in agency serving homeless youth in Los Angeles) and is currently being reviewed by their officials.

1 Introduction

Homelessness affects ∼2 million youths in USA annually, 11% of whom are HIV positive, which is 10 times the rate of infection in the general population (Aidala and Sumartojo 2007). Peer-led HIV prevention programs such as Popular Opinion Leader (POL) (Kelly et al. 1997) try to spread HIV prevention information through network ties and recommend selecting intervention participants based on Degree Centrality (i.e., highest degree nodes first). Such peer-led programs are highly desirable to agencies working with homeless youth as these youth are often disengaged from traditional health care settings and are distrustful of adults (Rice and Rhoades 2013; Rice 2010).

Agencies working with homeless youth prefer a series of small size interventions deployed sequentially as they have limited manpower to direct towards these programs. This fact, along with emotional and behavioral problems of youth makes managing groups of more than 5-6 youth at a time very difficult (Rice et al. 2012b). Strategically choosing intervention participants is important so that information percolates through their social network in the most efficient way.

The purpose of this paper is to introduce PSINET (**P**OMDP based **S**ocial **I**nterventions in **N**etworks for **E**nhanced HIV **T**reatment), a novel Partially Observable Markov Decision Process (POMDP) based system which chooses the participants of successive interventions in a social network. The key novelty of our work is a unique combination of POMDPs and influence maximization to handle uncertainties about: (i) friendships between people in the social network; and (ii) evolution of the network state in between two successive interventions. Traditionally, influence maximization has not dealt with these uncertainties, which greatly complicates the process of choosing intervention participants. Moreover, this problem is a very good fit for POMDPs as: (i) we conduct several interventions sequentially, similar to sequential actions taken in a POMDP; and (ii) we must handle uncertainty over network structure and evolving state, similar to partial observability over states in a POMDP.

However, there are scalability issues that must be addressed. Unfortunately, our POMDP's state (2^{300} states) and action spaces ($\binom{150}{10}$ actions) are beyond the reach of current state-of-the-art POMDP solvers and algorithms. To address this scale-up challenge, PSINET provides a novel on-line approximation algorithm, that relies on the following key ideas: (i) compact representation of transition probabilities to manage the intractable state and action spaces; (ii) combination of Q_{MDP} heuristic (a well known offline approximate solver) with Monte-Carlo simulations to avoid exhaustive search of the entire belief space; and (iii) voting on multiple POMDP solutions, each of which efficiently searches a portion of the solution state space to improve accuracy. Each such POMDP solution (which votes for the final solution) is a decomposition of the original problem into a simpler problem. Thus, PSINET efficiently searches the combinatorial state and action spaces based on several heuristics in order to come up with good solutions.

Our work is done in collaboration with My Friend's Place

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

(MFP)[1], a non-profit agency assisting Los Angeles's homeless youth to build self-sufficient lives by providing education and support to reduce high-risk behavior. Our collaborators conducted extensive interviews with homeless youth at My Friend's Place to ascertain the structure of their friendship based social network. Figure 1 shows some of the homeless youth at My Friend's Place while Figure 2 shows the team of My Friend's Place. Figure 3 shows a social circus program called Cirque du Monde, one of the ways in which My Friend's Place intervenes with "at-risk" youth. Therefore, we evaluate PSINET on real social networks of youth attending this agency. This work is being reviewed by officials at My Friend's Place towards final deployment.

Figure 1: Homeless youth at My Friend's Place

Figure 2: My Friend's Place team along with a co-author

2 Related work

There are three primary areas of related work that we discuss in this section. First, we discuss work in the field of influence maximization, which was first explored by Kempe, Kleinberg, and Tardos (2003), who provided a constant-ratio approximation algorithm to find 'seed' sets of nodes

Figure 3: Cirque du Monde: A social circus program at My Friend's Place for intervening with "at-risk" youth

to optimally spread influence in a graph. This was followed by many speed up techniques (Leskovec et al. 2007; Kimura and Saito 2006; Chen, Wang, and Wang 2010). All these algorithms assume no uncertainty in the network structure and select a single seed set. In contrast, we select several seed sets sequentially in our work to select intervention participants. Also, our problem takes into account uncertainty about the network structure and evolving network state. Golovin and Krause (2011) introduced adaptive submodularity and discussed adaptive sequential selection (similar to our work) in viral marketing. However, unlike our work, they assume no uncertainty in network structure and state evolution.

Another field of related work involves two (or more) players trying to spread their own 'competing' influence in the network (broadly called influence blocking maximization, or IBM). Some research exists on IBM where all players try to maximize their own influence spread in the network, instead of limiting others' (Bharathi, Kempe, and Salek 2007; Kostka, Oswald, and Wattenhofer 2008; Borodin, Filmus, and Oren 2010). Tsai, Nguyen, and Tambe (2012) try to model IBM as a game theoretic problem and provide scale up techniques to solve large games. Just like our work, Tsai et al. (2013) consider uncertainty in network structure. However, Tsai et al. (2013) do not consider sequential planning (which is essential in our domain) and thus, their methods are not reusable in our domain.

The final field of related work is planning for reward/cost optimization. In POMDP literature, a lot of work has been done on offline planning; some notable offline planners include GAPMIN (Poupart, Kim, and Kim 2011) and Symbolic Perseus (Spaan and Vlassis 2005). However, since it has been suggested that online planners are able to scale up better (Paquet, Tobin, and Chaib-Draa 2005), we focus on online POMDP planners in this paper. For online planning, we mainly focus on the literature on Monte-Carlo (MC) sampling based online POMDP solvers since this approach allows significant scale-ups. Silver and Veness (2010) proposed the *Partially Observable Monte Carlo Planning* (POMCP) algorithm that uses Monte-Carlo tree search in online planning. Also, Somani et al. (2013) present the DESPOT algorithm, that improves the worst case perfor-

[1]See http://myfriendsplace.org/

mance of POMCP. Bai et al. (2014) used Thompson sampling to intelligently trade-off between exploration and exploitation in their $D^2NG - POMCP$ algorithm. These algorithms maintain a search tree for all sampled histories to find the best actions, which may lead to better solution qualities, but it makes these techniques less scalable (as we show in our experiments). Therefore, our algorithm does not maintain a search tree and uses the Q_{MDP} heuristic (Littman, Cassandra, and Kaelbling 1995) to find best actions.

Others have also looked at planning/scheduling problems for optimization. Just like our work, Burns et al. (2012) sample possible futures to find optimal plans. However, while they consider online continual planning problems (i.e., problems in which additional goals arrive during execution of previous goals), we have fixed goals and uncertain observations in our problem. Also, Siddiqui and Haslum (2013) and Asai and Fukunaga (2014) use ideas of decomposition of planning problems into simpler problems in order to improve efficiency (similar to our work, but they do not handle POMDPs).

3 Our Approach

Partially Observable Markov Decision Processes (POMDPs) are a well studied model for sequential decision making under uncertainty (Puterman 2009). Intuitively, POMDPs model situations wherein an agent tries to maximize its expected long term *rewards* by taking various *actions*, while operating in an environment (which could exist in one of several *states* at any given point in time) which reveals itself in the form of various *observations*. The key point is that the exact state of the world is not known to the agent and thus, these actions have to be chosen by reasoning about the agent's probabilistic beliefs (belief state). The agent, thus, takes an action (based on its current belief), and the environment transitions to a new world state. However, information about this new world state is only partially revealed to the agent through observations that it gets upon reaching the new world state. Hence, based on the agent's current belief state, the action that it took in that belief state, and the observation that it received, the agent updates its belief state. The entire process repeats several times until the environment reaches a terminal state (according to the agent's belief state). More formally, a POMDP is a tuple $\wp$ given by:

$$\wp = \langle \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{O}, \boldsymbol{T}, \boldsymbol{\Omega}, \boldsymbol{R} \rangle \tag{1}$$

where the various symbols are defined as follows:

- $\boldsymbol{S} :=$ Set of possible world states,
- $\boldsymbol{A} :=$ Set of possible actions,
- $\boldsymbol{O} :=$ Set of possible observations,
- $\boldsymbol{T}(s, a, s') :=$ Transition probability of reaching s' from s, upon taking action a,
- $\boldsymbol{\Omega}(o, a, s') :=$ Observation probability of observing o, upon taking action a and reaching state s'
- $\boldsymbol{R}(s, a) :=$ Reward of taking action a in state s

A POMDP policy Π maps every possible belief state β (which is a probability distribution over world states) to an action $a = \Pi(\beta)$. Our aim is to find an optimal policy Π^* which, given an initial belief β_0, maximizes the expected cumulative long term reward over H horizons (where the agent takes an action and gets a reward in each time step until the horizon H is reached). Computing optimal policies offline for finite horizon POMDPs is PSPACE-Complete. Thus, focus has recently turned towards online algorithms, which only find the best action for the current belief state (Paquet, Tobin, and Chaib-Draa 2005; Silver and Veness 2010). Thus, online planning interleaves planning and execution at every time step.

POMDP Model of our Domain

In describing our model, we first outline the homeless youth social network and then map it onto our POMDP. The social network of homeless youth is a directed graph $G = (\mathbf{V}, \mathbf{E})$ with $|\mathbf{V}| = n$. Every $v \in \mathbf{V}$ represents a homeless youth, and every $\{e = (B, C) | B, C \in \mathbf{V}\} \in \mathbf{E}$ represents that youth B has nominated (listed) youth C in their social circle.

Further, $\mathbf{E} = \mathbf{E_c} \cup \mathbf{E_u}$, where $\mathbf{E_c}(|\mathbf{E_c}| = l)$ is the set of certain edges, i.e., friendships which we are certain about. Conversely, $\mathbf{E_u}(|\mathbf{E_u}| = m)$ is the set of uncertain edges, i.e., friendships which we are uncertain about. For example, youth may describe their friends "vaguely", which is not enough for accurate identification (Rice et al. 2012b; 2012a). In this case, there would be uncertain edges from the youth to each of his "suspected" friends.

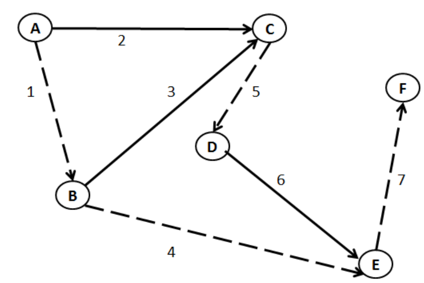


Figure 4: A sample 6 node uncertain graph

Each uncertain edge ($e \in \mathbf{E_u}$) exists with an *existence probability* $u(e)$, the exact value of which is determined from domain experts. For example, if it is uncertain whether node B is node A's friend, then $u(A, B) = 0.5$ implies that B is A's friend with a 0.5 chance. Accounting for these uncertain edges is important as our node selection might depend heavily on whether these edges exist with certainty or not. We call this graph G an "uncertain graph" henceforth. Figure 4 shows an uncertain graph on 6 nodes (A to F) and 7 edges. The dashed and solid edges represent uncertain (edge numbers 1, 4, 5 and 7) and certain (edge numbers 2, 3 and 6) edges, respectively.

In our work, we use the independent cascade model, a well studied influence propagation model (Kimura and Saito 2006). In this model, every node $v \in \mathbf{V}$ has an h-value, where $h : \mathbf{V} \rightarrow \{0, 1\}$. $h(v) = 1$ and 0 determines whether a node is influenced or not, respectively. Nodes only change their h-value (from 0 to 1) once, when they get influenced. Once a node gets influenced, it cannot go back to being uninfluenced. If node $v \in \mathbf{V}$ gets influenced at time step t, it influences each of its 1-hop uninfluenced neighbors with a *propagation probability* $p(e)\ \forall e \in \mathbf{E}$ for *all future time steps*. Moreover, every edge $e \in \mathbf{E_u}$ has an f-value (which represents a sampled instance of $u(e)$ and is unknown apriori), where $f : \mathbf{E_u} \rightarrow \{0, 1\}$. $f(e) = 1$ and 0 determines whether the uncertain edge exists with certainty in the real graph (i.e., the youth at the end of that uncertain edge are actually friends) or not (i.e., the youth at the end of that uncertain edge are not friends), respectively. For $e \in \mathbf{E_u}$, the influence probability (given by $p(e) \times u(e)$) is contingent on the edge's actual existence.

Note that eliminating all uncertain edges by replacing them with certain edges which propagate influence with probability $p(e) \times u(e)$ is not possible. This is because, in our model, when we pick nodes, we resolve uncertainty in their neighboring edges (explained later), so the probability would change from $p(e) \times u(e)$ to either $p(e)$ or 0 (depending on whether we found out that the uncertain edge exists or it does not). If the probability changes to $p(e)$, then influence will spread along this edge with probability $p(e)$ for all future time steps. Otherwise, if the probability changes to 0, no influence will spread in future time steps. Due to this changing probability value, we cannot apply the transformation of replacing uncertain edges.

Recall that we need a policy for selecting nodes for successive interventions in order to maximize the influence spread in the network. Nodes selected for interventions are assumed to be influenced ($h(v) = 1$) post-intervention with certainty. However, there is uncertainty in how the h-value of the unselected nodes changes in between successive interventions. For example, in Figure 4, if we choose nodes B and D for the 1^{st} intervention, we are uncertain whether nodes C and E (adjacent to nodes B and D) are influenced before nodes for the 2^{nd} intervention are chosen. We now provide a POMDP mapping onto our problem.

States A state consists of the state of the nodes (i.e., whether they are influenced or not), along with the state of the uncertain edges (i.e., whether they exist or not). The state of the nodes is given by their h-values and the state of the uncertain edges is given by their f-values. Our POMDP has 2^{n+m} states.

Actions Every subset of k nodes (k is the number of nodes selected per intervention) is a POMDP action. For example, in Figure 4, one possible action is $\{A, B\}$ (assuming $k = 2$). Our POMDP has $\binom{n}{k}$ actions.

Observations Previous studies such as Rice et al. (2012b) show that homeless youth are found to be more willing to discuss their social ties in presence of outreach workers in an intervention. Therefore, we assume that we can "*observe*" the f-values of uncertain edges outgoing from the nodes chosen in an action. This translates to asking intervention participants about their 1-hop social circles, which is within the agency's capacity. For example, by taking action $\{B, C\}$ in Figure 4, the f-values of edge 4 and 5 (i.e., uncertain edges in the 1-hop social circle of nodes B and C) would be observed. Consider $\Theta(\boldsymbol{\alpha}) = \{$e | e = (B,C) s.t. B $\in \boldsymbol{\alpha}\ \wedge$ e $\in \mathbf{E_u}\}\ \forall \boldsymbol{\alpha} \in \boldsymbol{A}$, which represents the ordered tuple of uncertain edges that are observed when the agency takes action $\boldsymbol{\alpha}$. Then, our POMDP observation upon taking action $\boldsymbol{\alpha}$ is defined as $o(\boldsymbol{\alpha}) = \langle f(e_1), f(e_2), ..., f(e_i)\rangle\ \forall e_i \in \Theta(\boldsymbol{\alpha})$, i.e., the f-values of the observed uncertain edges. The number of possible POMDP observations is exponential in Θ.

Transition Probabilities Consider states $s = \langle H, F\rangle$ and $s' = \langle H', F'\rangle$ and action $\boldsymbol{\alpha} \in \mathbf{A}$. In order for $\boldsymbol{T}(s, \boldsymbol{\alpha}, s')$ to be non-zero, we require the following three conditions to hold:

$$F'[i] = F[i]\ \forall\ i \text{ s.t. } e_i \notin \Theta(\boldsymbol{\alpha}) \tag{2}$$

$$H'[i] = H[i]\ \forall\ i \text{ s.t } H[i] = 1 \tag{3}$$

$$H'[i] = 1\ \forall\ i \text{ s.t. } v_i \in \boldsymbol{\alpha} \tag{4}$$

If any of the conditions (2), (3) or (4) is not true, then $\boldsymbol{T}(s, \boldsymbol{\alpha}, s') = 0$. Intuitively, equation 2 means that all uncertain edges which were not observed will not change their f-values. Equations 3 and 4 mean that all nodes which were already influenced in the previous state, along with all nodes that we influence as a result of action $\boldsymbol{\alpha}$ will remain influenced in the final state. If they don't remain influenced in the final state, then that final state cannot be reached (as influenced nodes cannot go back to being uninfluenced). For the cases where these conditions hold, we provide a heuristic method to calculate transition probabilities in the next section (as accurate calculation needs to consider all possible paths in a graph through which influence could spread, which is $O(n!)$ in the worst case).

Transition Probability Heuristic In this section, we explain our transition probability heuristic that we use for estimating our POMDP's transition probability matrix. Essentially, we need to come up with a way of finding out the final state of the network (probabilistically) prior to the beginning of the next intervention round. Prior to achieving the final state, the network evolves in a pre-decided number of time-steps. Each time step corresponds to a period in which friends can talk to their friends. Therefore, a time step value of 3 implies allowing for friends at 3 hops distance to be influenced.

However, we make an important assumption that we describe next. Consider 2 different chains of length four (nodes) as shown in Figure 5. In Chain 1, only the node at the head of the chain is influenced (shown in black) and the remaining three nodes are not influenced (shown in white). The probability of the tail node of this chain getting influenced is $(0.5)^3$ (assuming no edge is uncertain and probability of propagation is 0.5 on all edges). In Chain 2, all nodes except the tail node is already influenced. In this case, the tail node gets influenced with a probability $0.5 + (0.5)^2 + (0.5)^3$. Thus, it is highly unlikely that influence will spread to the end node of the first chain as opposed to the second chain. For this reason, we only keep chains of the form of Chain 2 and accordingly prune our graph (explained next).

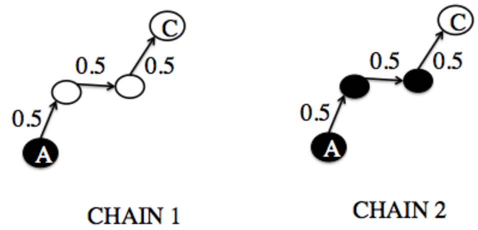


Figure 5: Chains in social networks

Consider a weighted adjacency matrix representation for pruned graph G_σ (created from graph G), s.t.

$$G_\sigma(i,j) = \begin{cases} 1 & \text{if } (i,j) \in \mathbf{E_c} \wedge (H[i] = 1 \vee \boldsymbol{\alpha}[i] = 1) \\ u(i,j) & \text{if } (i,j) \in \mathbf{E_u} \wedge (H[i] = 1 \vee \boldsymbol{\alpha}[i] = 1) \\ 0 & \text{if } otherwise. \end{cases} \tag{5}$$

G_σ is a *pruned* graph which contains only edges outgoing from influenced nodes. We prune the graph because influence can only spread through edges which are outgoing from influenced nodes. Note that G_σ only considers chains of type 2 and prunes away chains of type 1. Using these assumptions, we use G_σ to construct a diffusion vector $\overrightarrow{\boldsymbol{D}}$, the i^{th} element of which gives us a measure of the probability of the i^{th} node to get influenced. This diffusion vector $\overrightarrow{\boldsymbol{D}}$ is then used to estimate $\boldsymbol{T}(s, \boldsymbol{\alpha}, s')$.

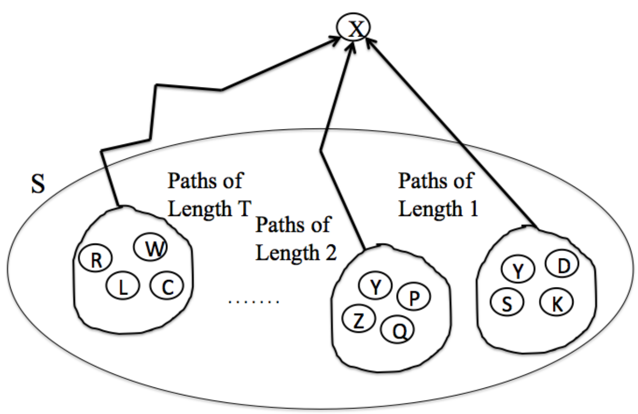


Figure 6: X is any uninfluenced node. S (the big oval) denotes the set of all influenced nodes. All these nodes have been categorized according to their path length from node X. For e.g., all nodes having a path of length 1 (i.e., Y, D, S, K) are distinguished from all nodes having path of length T (i.e., R, W, L, C). Note that node Y has paths of length 1 and 2 to node X.

Figure 6 illustrates the intuition behind our transition probability heuristic. More details on the heuristic can be found in our IAAI 2015 paper (Yadav et al. 2015). For each uninfluenced node X in the graph, we calculate the total number of paths (like Chain 2 in Figure 5) of different lengths $L = 1, 2, \ldots, \mathcal{T}$ from influenced nodes to node X. Since influence spreads on chains of different lengths according to different probabilities, the probabilities along all paths of different lengths are combined together to determine an approximate probability of node X to get influenced before the next intervention round. Since we consider all these paths independently (instead of calculating joint probabilities), our approach produces an approximation[2].

Observation Probabilities Calculating $\boldsymbol{\Omega}(o, \boldsymbol{\alpha}, s')$ is trivial as the final state s' already has f-values of uncertain edges, and you know which nodes you pick in action $\boldsymbol{\alpha}$. Thus, given s' and $\boldsymbol{\alpha}$, only one observation is possible: $o(\boldsymbol{\alpha}, s') = \{F'[i] \, \forall e_i \in \Theta(\boldsymbol{\alpha})\}$.

Rewards The reward of taking action $\boldsymbol{\alpha} \in \boldsymbol{A}$ in state $s = \langle H, F \rangle$ (denoted by $R(s, \boldsymbol{\alpha})$) is given as: $\boldsymbol{R}(s, \boldsymbol{\alpha}) = \sum_{s' \in \boldsymbol{S}} T(s, \boldsymbol{\alpha}, s')(\|s'\| - \|s\|)$, where $\|s'\|$ is the number of influenced nodes in s'. This gives the expected number of new influenced nodes.

PSINET

Initial experiments with the ZMDP solver (Smith 2013) showed that state-of-the-art offline POMDP planners run out of memory on 10 node graphs. Thus, we focused on online planning algorithms and tried using POMCP (Silver and Veness 2010), a state-of-the-art online POMDP solver which relies on Monte-Carlo (MC) tree search and rollout strategies to come up with solutions quickly. However, it keeps the entire search tree over sampled histories in memory, disabling scale-up to the problems of interest in this paper. Hence, we propose a MC based online planner that utilizes the Q_{MDP} heuristic and eliminates this search tree.

POMDP black box simulator MC sampling based planners approximate the value function for a belief by the average value of η (say) MC simulations starting from states sampled from the current belief state. Such approaches depend on a POMDP black box simulator $\Gamma(s_t, \boldsymbol{\alpha}_t) \sim (s_{t+1}, o_{t+1}, r_{t+1})$ which generates the state, observation and reward at time $t+1$, given the state and action at time t, in accordance with the POMDP dynamics. In Γ, o_{t+1}, s_{t+1} and r_{t+1} are generated as follows:

- $\boldsymbol{o_{t+1}}$: Every edge e in $\Theta(\boldsymbol{\alpha}_t)$ is sampled (either kept or removed) according to the existence probability on the edge in order to generate o_{t+1}.
- $\boldsymbol{s_{t+1}}$: In order to get s_{t+1} from s_t, we sample the h-values of nodes which are neither influenced in s_t or in the action $\boldsymbol{\alpha}_t$ by using the D_i values of these nodes. So, we flip a weighted coin with probability D_i to find whether node i is influenced in s_{t+1}. Note that s_{t+1} calculated this way represents a state sampled according to $\boldsymbol{T}(s_t, \boldsymbol{\alpha}_t, s_{t+1})$.
- $\boldsymbol{r_{t+1}}$: $\|s_{t+1}\| - \|s_t\|$, where $\|s_{t+1}\|$ is the number of influenced nodes in s_{t+1}.

[2] http://teamcore.usc.edu/people/amulya/appendix.pdf provides details/proofs.

Algorithm 1: PSINET

Input: Belief state β, Uncertain graph G

Output: Best Action κ

1 Sample graph to get Υ different instances;

2 **for** $\delta \in \boldsymbol{\Upsilon}$ **do**

3 $\quad FindBestAction(\delta, \boldsymbol{\alpha}_{\delta}, \beta)$;

4 $\kappa = VoteForBestAction(\boldsymbol{\Upsilon}, \boldsymbol{\alpha})$

5 $UpdateBeliefState(\kappa, \beta)$;

6 return κ;

Q_{MDP} It is a well known approximate offline planner, and it relies on $Q(s,a)$ values, which represents the value of taking action a in state s. It precomputes these $Q(s,a)$ values for every (s,a) pair by approximating them by the future expected reward obtainable if the environment is fully observable (Littman, Cassandra, and Kaelbling 1995). Finally, Q_{MDP}'s approximate policy Π is given by $\Pi(\beta) = \arg\max_a \sum_s Q(s,a)\beta(s)$ for belief β. Our intractable POMDP state and action spaces makes it infeasible to calculate $Q(s,a)\ \forall\ (s,a)$. Thus, we propose to use a MC sampling based online variant of Q_{MDP} in PSINET.

Algorithm Flow Algorithm 1 shows the flow of PSINET. In Step 1, we randomly sample all $e \in \mathbf{E_u}$ in G (according to $u(e)$) to get Υ different graph instances, forming a set $\boldsymbol{\Upsilon}$. Each of these instances is a different POMDP as even though we remove uncertainty about $f(e) \forall e \in \mathbf{E_u}$, the h-values of nodes are still partially observable (i.e., we know that nodes that we picked for the intervention get influenced with certainty, but we do not have any observation concerning the other nodes). Since each of these instances fixes $f(e)\ \forall e \in \mathbf{E_u}$, the belief β is represented as an unweighted particle filter where each particle is a tuple of h-values of all nodes. This belief is shared across all instantiated POMDPs. For every graph instance $\delta \in \boldsymbol{\Upsilon}$, we estimate the best action $\boldsymbol{\alpha}_{\delta}$ in graph δ, for the current belief β in step 3 (this process is done in parallel for each distinct graph instance, as shown in Figure 7). In step 4, we find our best estimation κ of the optimal action for belief β, over all $\delta \in \boldsymbol{\Upsilon}$ by voting amongst all the actions chosen by $\delta \in \boldsymbol{\Upsilon}$. Then, in step 5, we update the belief state based on the chosen action κ and the current belief β. PSINET can again be used to find the best action for this or any future updated belief states. We now detail the steps in Algorithm 1.

Sampling Graphs In Step 1, we randomly keep or remove uncertain edges to create one graph instance. As a single instance might not represent the real network well, we instantiate the graph Υ times and use each of these instances to vote for the best action to be taken.

Finding Best Action Step 3 uses Algorithm 2, which finds the best action for a single network instance, and works similarly for all instances. Figure 8 illustrates the flow of Algorithm 2. For each instance, we find the action which maximizes long term rewards averaged across η (we use $\eta = 2^8$) MC simulations starting from states (particles) sampled from the current belief β. Each MC simulation samples a particle from β and chooses an action to take (choice of action is explained later). Then, upon taking this action, we

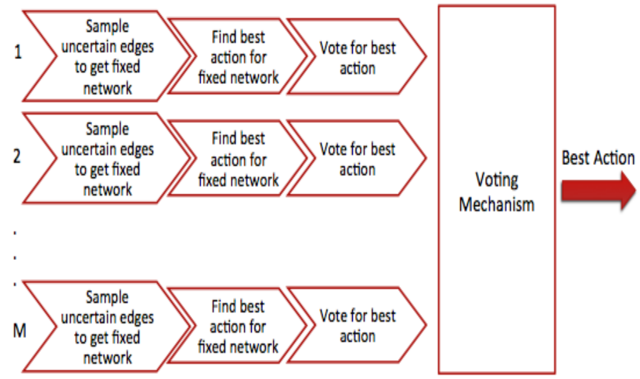


Figure 7: Parallelizing execution of several threads

follow a uniform random rollout policy (until either termination, i.e., all nodes get influenced, or the horizon is breached) to estimate the long term reward, which we get by taking the "selected" action. This reward from each MC simulation is analogous to a $Q(s,a)$ estimate. Finally, we pick the action with the maximum average reward.

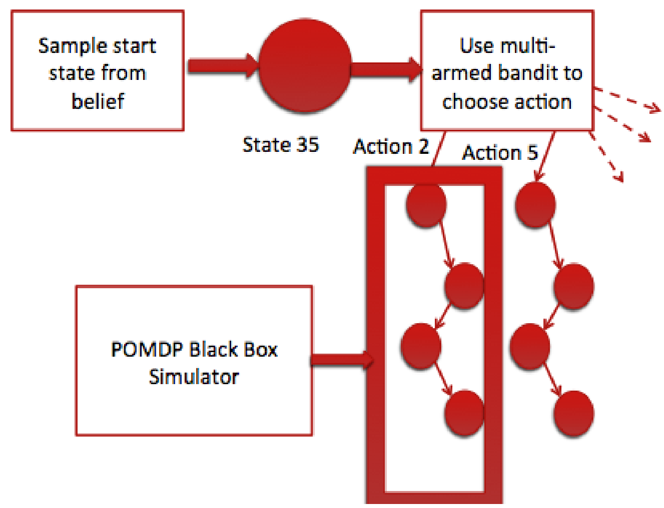


Figure 8: Flow inside Find Best Action

Multi-Armed Bandit We can only calculate $Q(s,a)$ for a select set of actions (due to our intractable action space). To choose these actions, we use an *Upper Confidence Bound* (UCB1) implementation of a multi-armed bandit to select actions, with each bandit arm being one possible action. Every time we sample a new state from the belief, we run UCB1, which returns the action which maximizes this quantity: $\sigma(s,a) = Q_{MC}(s,a) + c_0\sqrt{\frac{\log N(s)}{N(s,a)}}$. Here, $Q_{MC}(s,a)$ is the running average of Q(s,a) values across all MC simulations run so far. $N(s)$ is number of times state s has been sampled from the belief. $N(s,a)$ is number of times action a has been chosen in state s and c_0 is a constant which deter-

Algorithm 2: FindBestAction

Input: Graph instance δ, belief β, η simulations
Output: Best Action $\boldsymbol{\alpha}_\delta$

Initialize $counter = 0$;
while $counter++ < N$ **do**
  $s = SampleStartStateFromBelief(\beta)$;
  $a = UCB1\_MultiArmedBandit(s)$;
  $\{s', r\} = SimulateRolloutPolicy(s, a)$;
$\boldsymbol{\alpha}_\delta$ = action with max average reward;
return $\boldsymbol{\alpha}_\delta$;

mines the exploration-exploitation tradeoff for UCB1. High c_0 values make UCB1 choose rarely tried actions more frequently, and low c_0 values make UCB1 select actions having high $Q_{MC}(s, a)$ to get an even better $Q(s, a)$ estimate. Thus, in every MC simulation, UCB1 strategically chooses which action to take, after which we run the rollout policy to get the long term reward.

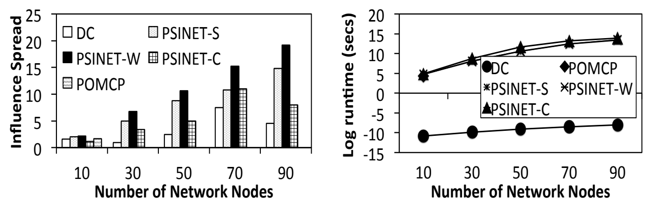
Voting Mechanisms In Step 4, each network instance votes for the best action (found using Step 3) for the uncertain graph and the approximate best action is chosen by aggregating these votes according to different voting schemes. We propose using the following three different voting schemes:

- **PSINET-S** Each instance's vote gets equal weight.
- **PSINET-W** Every instance's vote gets weighted differently. The instance which removes x uncertain edges has a vote weight of $W(x) = x \,\forall x \leq m/2$ and $W(x) = m - x \,\forall x > m/2$. This weighting scheme approximates the probabilities of occurrences of real world events by giving low weights to instances which removes either too few or too many uncertain edges, since those events are less likely to occur. Instances which remove $m/2$ uncertain edges get the highest weight, since that event is most likely.
- **PSINET-C** Given a ranking over actions from each instance, the Copeland rule makes pairwise comparisons among all actions, and picks the one preferred by a majority of instances over the highest number of other actions (Pomerol and Barba-Romero 2000). It is a popular voting rule because it is *Condorcet consistent* (i.e., if an action is preferred to every other action in a majority of the votes, it will be selected with certainty). Similar to Jiang et al. (2014), we generate a partial ranking for each instance by using multiple runs of Algorithm 2.

Belief State Update Recall that every MC simulation samples a particle from the belief, after which UCB1 chooses an action. Upon taking this action, some random state (particle) is reached using the transition probability heuristic. This particle is stored, indexed by the action taken to reach it. Finally, when all simulations are done, corresponding to every action $\boldsymbol{\alpha}$ that was tried during the simulations, there will be a set of particles that were encountered when we took action $\boldsymbol{\alpha}$ in that belief. The particle set corresponding to the action that we finally choose forms our next belief state.

4 Experimental Evaluation

We provide two sets of results. First, we show results on artificial networks to understand our algorithms' properties on abstract settings, and to gain insights on a range of networks. Next, we show results on the two real world homeless youth networks that we had access to. In all experiments, we select 2 nodes per round and average over 20 runs, unless otherwise stated. We set the value of $\mathcal{T} = 3$ (the number of hops considered for influence spread in Equation **??**) in all experiments. PSINET-(S and W) use 20 network instances and PSINET-C uses 5 network instances (each instance finds its best action 5 times) in all experiments, unless otherwise stated. The propagation and existence probability values were set to 0.5 in all experiments (based on findings by Kelly et al. (1997)), although we relax this assumption later. In this section, a $\langle X, Y, Z\rangle$ network refers to a network with X nodes, Y certain and Z uncertain edges. We use a metric of "indirect influence spread" (IIS) throughout this section, which is the number of nodes "indirectly" influenced by the intervention participants. For example, on a 30 node network, by selecting 2 nodes each for 10 interventions (horizon), 20 nodes (a lower bound for any strategy) are influenced with certainty. However, the total number of influenced nodes might be 26 (say) and thus, the IIS is 6. *All comparison results are statistically significant under bootstrap-t ($\alpha = 0.05$).*



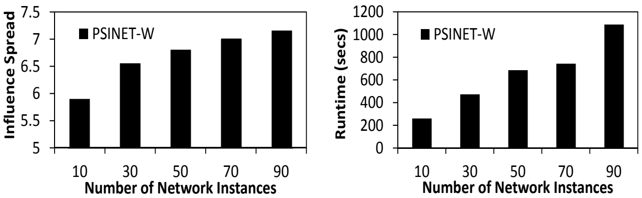
(a) Solution Quality (b) Runtime

Figure 9: Comparison on BTER graphs

Artificial networks First, we compare all algorithms on Block Two-Level Erdos-Renyi (BTER) networks (having degree distribution $X_d \propto d^{-1.2}$, where X_d is number of nodes of degree d) of several sizes, as they accurately capture observable properties of real-world social networks (Seshadhri, Kolda, and Pinar 2012).

In Figure 9(a), we compare solution qualities of Degree Centrality (DC), POMCP and PSINET-(S,W and C) on BTER networks of varying sizes. In DC, nodes are selected in subsequent rounds in decreasing order of out-degrees, where every uncertain edge $e \in \mathbf{E_u}$ adds $u(e)$ to the node degrees. We choose DC as our baseline as it is the current modus operandi of agencies working with homeless youth. The x-axis shows number of network nodes and the y-axis shows IIS across varying horizons (number of interventions). This figure shows that all POMDP based algorithms beat DC by ~60%, which shows the value of our POMDP model. Further, it shows that PSINET-W beats PSINET-(S and C). Also, *POMCP runs out of memory on 30 node graphs.*

In Figure 9(b), we show runtimes of DC, POMCP and PSINET-(S,W and C) on the same BTER networks. The x-axis shows number of network nodes and the y-axis shows log (base e) of runtime (in seconds). Figure 9(b) shows that DC runs quickest (as expected) and all PSINET variants run in almost the same time. Thus, Figures 9(a) and 9(b) tell us that while DC runs quickest, it provides the worst solutions. Amongst the POMDP based algorithms, PSINET-W is the best algorithm that can provide good solutions and can scale up as well. Surprisingly, PSINET-C performs worse than PSINET-(W and S) in terms of solution quality. Thus, we now focus on PSINET-W.

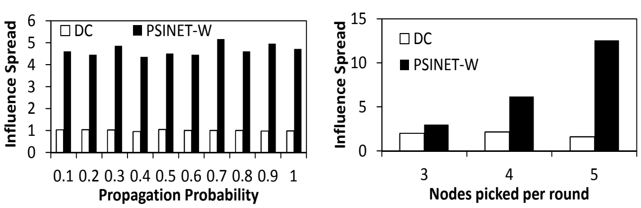


(a) Solution Quality (b) Runtime

Figure 10: Increasing number of graph instances

Having shown the impact of POMDPs, we analyze the impact of increasing network instances (which implies increasing number of votes in our algorithm) on PSINET-W. In Figure 10(a), we show solution quality of PSINET-W with increasing network instances, for a $\langle 40, 71, 41\rangle$ BTER network with a horizon of 10. The x-axis shows the number of network instances and the y-axis shows IIS. Unsurprisingly, this figure shows that increasing the number of network instances increases IIS as well.

In Figure 10(b), we show runtime of PSINET-W with increasing network instances, for a $\langle 40, 71, 41\rangle$ BTER network with a horizon of 10. The x-axis shows the number of network instances and the y-axis shows runtime (in seconds). This figure shows that increasing the number of network instances increases the runtime as well. Thus, a solution quality-runtime tradeoff exists, which depends on the number of network instances. Greater number of instances results in better solutions and slower runtimes and vice versa. However, for 30 vs 70 instances, the gain in solution quality is <5% whereas the runtime is ∼2X, which shows that increasing instances beyond 30 yields marginal returns.



(a) Varying $p(e)$ (b) Scaling nodes/round

Figure 11: Comparison of Degree Centrality with PSINET-W across varying parameters

Next, we relax our assumptions about propagation ($p(e)$) probabilities, which were set to 0.5 so far. Figures 11(a) shows the solution quality, when PSINET-W and DC are solved with different $p(e)$ values on the network edges (the p(e) values were changed for both the network that was input to the algorithm and the network on which the algorithm's policy was executed) for a $\langle 40, 71, 41\rangle$ BTER network with a horizon of 10. The x-axis shows $p(e)$ and the y-axis shows IIS. This figure shows that varying $p(e)$ minimally impacts PSINET-W's improvement over DC, which shows our algorithms' robustness to these probability values (we get similar results upon changing $u(e)$).

In Figure 11(b), we show solution qualities of PSINET-W and DC on a $\langle 30, 31, 27\rangle$ BTER network by varying the number of nodes selected per round (k). We use a horizon of 3 (in order to ensure the performance of our algorithm on varying horizon lengths). The x-axis shows increasing k, and the y-axis shows IIS. This figure shows that even for a small horizon of length 3, PSINET-W significantly beats DC. For increasing values of k, PSINET-W beats DC with increasing margins.

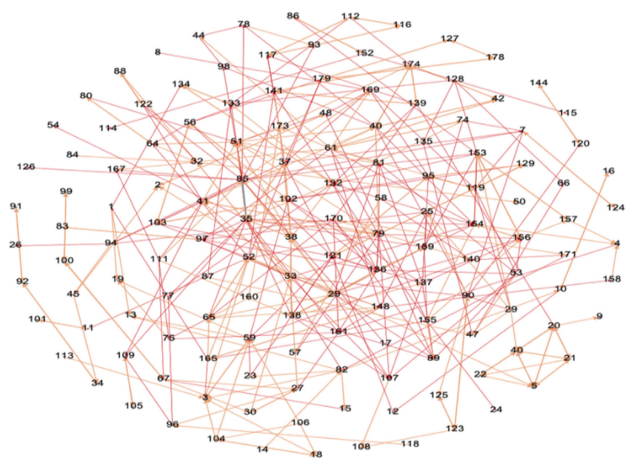


Figure 12: One of the friendship based social network of homeless people visiting My Friend's Place

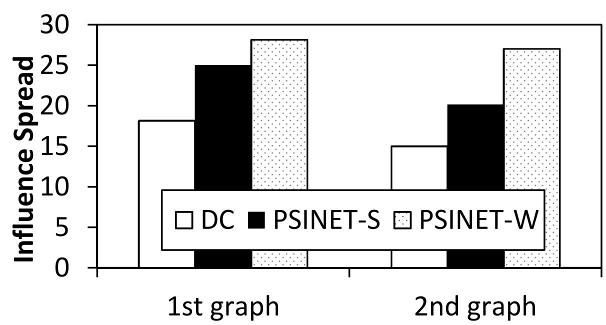


Figure 13: Solution Quality for Real World Network

Real world networks Figure 12 shows one of the two real-world friendship based social networks of homeless

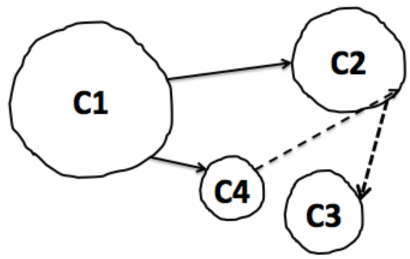


Figure 14: Sample BTER Graph

youth (created by our collaborators through surveys and interviews of homeless youth attending My Friend's Place), where each numbered node represents a homeless youth. Figure 13 compares PSINET variants and DC (horizon = 30) on these two real-world social networks (each of size $\sim \langle 155, 120, 190 \rangle$). The x-axis shows the two networks and the y-axis shows IIS. This figure clearly shows that all PSINET variants beat DC on both real world networks by $\sim$60%, which shows that PSINET works equally well on real-world networks. Also, PSINET-W beats PSINET-S, in accordance with previous results. Above all, this signifies that we could improve the quality and efficiency of HIV based interventions over the current modus operandi of agencies by $\sim$60%.

We now differentiate between the kinds of nodes selected by DC and PSINET-W for the sample BTER network in Figure 14, which contains nodes segregated into four clusters (C1 to C4), and node degrees in a cluster are almost equal. C1 is biggest, with slightly higher node degrees than other clusters, followed by C2, C3 and C4. DC would first select all nodes in cluster C1, then all nodes in C2 and so on. Selecting all nodes in a cluster is not "smart", since selecting just a few cluster nodes influences all other nodes. PSINET-W realizes this by looking ahead and spreads more influence by picking nodes in different clusters each time. For example, assuming $k = 2$, PSINET-W picks one node in both C1 and C2, then one node in both C1 and C4, etc.

5 Implementation Challenges

Looking towards the future of testing the deployment of this procedure in agencies, there are a few implementation challenges that will need to be faced. First, collecting accurate social network data on homeless youth is a technical and financial burden beyond the capacity of most agencies working with these youth. Members of this team had a large three year grant from the National Institute of Mental Health to conduct such work in only two agencies. Our solution, moving forward (with other agencies) would be to use staff at agencies to delineate a first approximation of their homeless youth social network, based on their ongoing relationships with the youth. The POMDP procedure would subsequently be able to correct the network graph iteratively (by resolving uncertain edges via POMDP observations in each step). This is feasible because, as mentioned, homeless youth are more willing to discuss their social ties in an intervention (Rice et al. 2012b). We see this as one of the major strengths of this approach.

Second, our prior research on homeless youth (Rice and Rhoades 2013) suggests that some structurally important youth may be highly anti-social and hence a poor choice for change agents in an intervention. We suggest that if such a youth is selected by the POMDP program, we then choose the next best action (subset of nodes) which does not include that "anti-social" youth. Thus, the solution may require some ongoing management as certain individuals either refuse to participate as peer leaders or based on their anti-social behaviors are determined by staff to be inappropriate.

Third, because of the history of neglect and abuse suffered by most of these youth, many are highly suspicious of adults. Including a computer-based selection procedure into the recruitment of peer leaders may raise suspicions about invasion of privacy for these youth. We suggest an ongoing public awareness campaign in the agencies working with this program to help overcome such fears and to encourage participation. Along with this issue, there is a secondary issue about protection of privacy for the individuals involved. Agencies collect information on their youth, but most of this information is not to be shared with researchers. We suggest working with agencies to create procedures which allow them to implement the POMDP program without having to provide identifying information to our team.

6 Conclusion

This paper presents PSINET, a POMDP based decision support system to select homeless youth for HIV based interventions. Previous work in strategic selection of intervention participants does not handle uncertainties in the social network's structure and evolving network state, potentially causing significant shortcomings in spread of information. PSINET has the following key novelties: (i) it handles uncertainties in network structure and evolving network state; (ii) it addresses these uncertainties by using POMDPs in influence maximization; and (iii) it provides algorithmic advances to allow high quality approximate solutions for such POMDPs. Simulations show that PSINET achieves $\sim$60% improvement over the current state-of-the-art. PSINET was developed in collaboration with My Friend's Place and is currently being reviewed by their officials.

7 Acknowledgements

This research was supported by MURI Grant W911NF-11-1-0332. Rice, Rhoades, Winetrobe and Petering were supported by NIMH Grant number R01-MH093336.

References

Aidala, A. A., and Sumartojo, E. 2007. Why housing? *AIDS and Behavior* 11(2):1–6.

Asai, M., and Fukunaga, A. 2014. Applying Problem Decomposition to Extremely Large Domains. *ICAPS Workshop on Knowledge Engineering for Planning and Scheduling*.

Bai, A.; Wu, F.; Zhang, Z.; and Chen, X. 2014. Thompson sampling based Monte-Carlo planning in POMDPs.

Bharathi, S.; Kempe, D.; and Salek, M. 2007. Competitive influence maximization in social networks. In *Internet and Network Economics*. Springer. 306–311.

Borodin, A.; Filmus, Y.; and Oren, J. 2010. Threshold models for competitive influence in social networks. In *Internet and network economics*. Springer. 539–550.

Burns, E.; Benton, J.; Ruml, W.; Yoon, S. W.; and Do, M. B. 2012. Anticipatory On-Line Planning. In *International Conference on Automated Planning and Scheduling*.

Chen, W.; Wang, C.; and Wang, Y. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1029–1038. ACM.

Golovin, D., and Krause, A. 2011. Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization. *Journal of Artificial Intelligence Research* 42:427–486.

Jiang, A. X.; Marcolino, L. S.; Procaccia, A. D.; Sandholm, T.; Shah, N.; and Tambe, M. 2014. Diverse Randomized Agents Vote to Win. In *Proceedings of the 28th Neural Information Processing Systems Conference*, NIPS 2014.

Kelly, J. A.; Murphy, D. A.; Sikkema, K. J.; McAuliffe, T. L.; Roffman, R. A.; Solomon, L. J.; Winett, R. A.; and Kalichman, S. C. 1997. Randomised, controlled, community-level HIV-prevention intervention for sexual-risk behaviour among homosexual men in US cities. *The Lancet* 350(9090):1500.

Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 137–146. ACM.

Kimura, M., and Saito, K. 2006. Tractable models for information diffusion in social networks. In *Knowledge Discovery in Databases: PKDD 2006*. Springer. 259–271.

Kostka, J.; Oswald, Y. A.; and Wattenhofer, R. 2008. Word of mouth: Rumor dissemination in social networks. In *Structural Information and Communication Complexity*. Springer. 185–196.

Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.; and Glance, N. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 420–429. ACM.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning Policies for Partially Observable Environments: Scaling Up. In *International Conference on Machine Learning (ICML)*. Morgan Kaufmann.

Paquet, S.; Tobin, L.; and Chaib-Draa, B. 2005. An online POMDP algorithm for complex multiagent environments. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 970–977. ACM.

Pomerol, J.-C., and Barba-Romero, S. 2000. *Multicrite-*

Pomerol, J.-C., and Barba-Romero, S. 2000. *Multicriterion decision in management: principles and practice*, volume 25. Springer Science & Business Media.

Poupart, P.; Kim, K.-E.; and Kim, D. 2011. Closing the Gap: Improved Bounds on Optimal POMDP Solutions. In *International Conference on Automated Planning and Scheduling*.

Puterman, M. L. 2009. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rice, E., and Rhoades, H. 2013. How should network-based prevention for homeless youth be implemented? *Addiction* 108(9):1625.

Rice, E.; Barman-Adhikari, A.; Milburn, N. G.; and Monro, W. 2012a. Position-specific HIV risk in a large network of homeless youths. *American journal of public health* 102.

Rice, E.; Tulbert, E.; Cederbaum, J.; Adhikari, A. B.; and Milburn, N. G. 2012b. Mobilizing homeless youth for HIV prevention: a social network analysis of the acceptability of a face-to-face and online social networking intervention. *Health education research* 27(2):226.

Rice, E. 2010. The positive role of social networks and social networking technology in the condom-using behaviors of homeless young people. *Public health reports* 125(4):588.

Seshadhri, C.; Kolda, T. G.; and Pinar, A. 2012. Community structure and scale-free collections of Erdős-Rényi graphs. *Physical Review E* 85(5):056109.

Siddiqui, F. H., and Haslum, P. 2013. Plan quality optimisation via block decomposition. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, 2387–2393. AAAI Press.

Silver, D., and Veness, J. 2010. Monte-Carlo planning in large POMDPs. In *Advances in Neural Information Processing Systems*, 2164–2172.

Smith, T. 2013. ZMDP Software for POMDP/MDP Planning.

Somani, A.; Ye, N.; Hsu, D.; and Lee, W. S. 2013. DESPOT: Online POMDP Planning with Regularization. In *Advances In Neural Information Processing Systems*, 1772–1780.

Spaan, M. T., and Vlassis, N. 2005. Perseus: Randomized point-based value iteration for POMDPs. *Journal of artificial intelligence research* 195–220.

Tsai, J.; Qian, Y.; Vorobeychik, Y.; Kiekintveld, C.; and Tambe, M. 2013. Bayesian security games for controlling contagion. In *Social Computing (SocialCom), 2013 International Conference on*, 33–38. IEEE.

Tsai, J.; Nguyen, T. H.; and Tambe, M. 2012. Security Games for Controlling Contagion. In *Conference on Artificial Intelligence (AAAI)*.

Yadav, A.; Marcolino, L.; Rice, E.; Petering, R.; Winetrobe, H.; Rhoades, H.; Tambe, M.; and Carmichael, H. 2015. Preventing hiv spread in homeless populations using psinet. In *In Proceedings of the Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence (IAAI-15)*.