PSINET - An Online POMDP Solver for HIV Prevention in Homeless Populations

A. Yadav, L. 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% (10 times the infection rate in the general population) of whom are HIV positive (NAHC 2011). Peer-led HIV prevention programs such as 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 and 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 algorithm, that relies on the following key ideas: (a) compact representation of transition probabilities to manage the intractable state and action spaces; (b) combination of the QMDP heuristic with Monte-Carlo simulations to avoid exhaustive search of the entire belief space; and (c) 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[1], a non-profit agency assisting Los Angeles's homeless youth to build self-sufficient lives by providing educa-

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

tion and support to reduce high-risk behavior. Thus, we evaluate PSINET on real social networks of youth frequenting this agency. This work is being reviewed by officials at My Friend's Place towards final deployment.

2 Related work

There are two 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 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.

The second field of related work is planning for reward/cost optimization. In POMDP literature, a lot of work has been done on offline planning but we focus on online planning, since offline planning approaches are unable to scale up to problems of interest in our work (Smith 2013). We focus on the literature on Monte-Carlo (MC) sampling based online POMDP solvers since that sub-field is most related to our work. Silver and Veness (2010) proposed POMCP algorithm that uses Monte-Carlo tree search in online planning. Also, Somani et al. (2013) improved the worst case performance of POMCP in DESPOT algorithm. These two algorithms maintain a search tree for all sampled histories to find the best actions, which may lead to better solution qualities, but it makes the algorithm less scalable (as we show in our experiments). Therefore, our algorithm does not maintain a search tree and uses the QMDP 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. Just like our work, Siddiqui and Haslum (2013) and Asai and Fukunaga (2014) use ideas of decomposition of planning problems into simpler problems in order to improve efficiency. Finally, Keller and Helmert (2013) introduce Trial-Based Heuristic Tree Search for solving finite-horizon MDPs, which is a generalization of Monte-Carlo tree search techniques.

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 about the world state(belief state). The agent, thus, takes an action, based on its current belief, and 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 a new world state. 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 and it repeats the entire process until it either reaches a terminal state or the number of steps(actions) taken exceed the horizon length. More formally, a POMDP is a tuple $\wp$ given by:

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

where the various symbols are defined as follows:

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

A POMDP policy Π maps every possible belief state b to an action $a = \Pi(b)$. Our aim is to find an optimal policy $\Pi^* = \arg\max_\pi P^\pi$ (given an initial belief b_0), which maximizes the expected long term reward $P^\Pi = \sum_{t=1}^{H} E[R(s^t, a^t)]$ where H is the horizon. 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. Upon reaching a new belief state, online planning again plans for this new belief. 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 digraph $G = (V, E)$ with $|V| = n$. Every $v \in V$ represents a homeless youth, and every $\{e = (a, b)|a, b \in V\} \in E$ represents that youth a has nominated (listed) youth b in their social circle. Further, $E = E_c \cup E_u$, where $E_c(|E_c| = l)$ is the set of certain edges, i.e., friendships which we are certain about. Conversely, $E_u(|E_u| = m)$ 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.

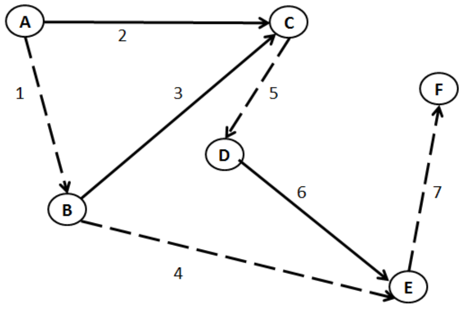


Figure 1: A 6 node uncertain graph

Each uncertain edge ($e \in E_u$) 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 $u(A, B) = 0.5$ (say) 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 1 shows an uncertain graph on 6 nodes (A to F) and 7 edges. The dashed and solid edges represent uncertain and certain 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 $v \in V$ has an h-value, where $h : V \rightarrow \{0, 1\}$. $h(v) = 1$ and 0 determines whether a node is influenced or not respectively. Nodes only change their h-value (from 0 to 1) when they first get influenced. If node $v \in V$ gets influenced at time step t, it influences each of its 1-hop un-influenced neighbors with a *propagation probability* $p(e)\ \forall e \in E$ for all future time steps. Moreover, every edge $e \in E_u$ has an f-value (which represents a sampled instance of $u(e)$ and is unknown apriori), where $f : E_u \rightarrow \{0, 1\}$. $f(e) = 1$ and 0 determines whether the uncertain edge exists with certainty in the real graph or not respectively. For $e \in E_u$, the influence probability (given by $p(e) * u(e)$) is contingent on the edge's actual existence.

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 1, 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 Consider strict total orders $<_v$ and $<_u$ on the sets V and E_u respectively. A state $S = \langle H, F \rangle$ is a 2-tuple. $H = \langle h(v_1), h(v_2), ..., h(v_i), ... \rangle\ \forall i \in 1..n$ is a binary tuple representing the h-values of all nodes (ordered by $<_v$). Also, $F = \langle f(e_1), f(e_2), ..., f(e_i), ... \rangle\ \forall i \in 1..m$ is a binary tuple representing the f-values of all uncertain edges (ordered by $<_u$) in the graph. Our POMDP has 2^{n+m} states.

Actions Every $\alpha \subset V$ s.t. $|\alpha| = k$ (k is the number of nodes selected per intervention) is a POMDP action. For example, in Figure 1, one possible action is $\{A, B\}$ (assuming $k = 2$). Our POMDP has $\binom{n}{k}$ actions.

Observations 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 (Rice et al. 2012b). For example, by taking action $\{B, C\}$ in Figure 1, 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 $\Theta(\alpha) = \{$e | e = (a,b) s.t. a $\in \alpha \wedge$ e $\in E_u\}\ \forall \alpha \in A$, which represents the ordered tuple of uncertain edges that are observed when the agency takes action α. Then, our POMDP observation upon taking action α is defined as $o(\alpha) = \langle f(e_1), f(e_2), ..., f(e_i) \rangle\ \forall e_i \in \Theta(\alpha)$ i.e., the f-values of the observed uncertain edges. In our POMDP, the number of observations is exponential in the size of Θ.

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

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

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

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

If any of the conditions (2), (3) or (4) is not true, then $T(s, \alpha, s') = 0$. 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 α will remain influenced in the final state. Otherwise, if 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 Consider a weighted adjacency matrix representation for graph G_σ (created from graph G) s.t.

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

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 does not consider influence spreading along a path consisting of more than one uninfluenced node, as this event is highly unlikely in the limited time in between successive interventions. However, nodes connected to a chain (of arbitrary length) of influenced nodes get influenced more easily due to reinforced efforts of all influenced nodes in the chain. We use G_σ to construct a diffusion vector $\mathbf{D}$, the i^{th} element of which gives us a measure of the probability of the i^{th} node to get influenced. This diffusion vector $\mathbf{D}$ is then used to estimate $T(s, \alpha, s')$.

A known result states that if G is a graph's adjacency matrix, then $G^r(i, j)$ (G^r = G multiplied r times) gives the number of paths of length r between nodes i and j (Diestel 2005). Additionally, note that if all edges e_i in a path of length r have different propagation probabilities $p(e_i)\ \forall\ i \in [1, r]$, the probability of influence spreading between two nodes connected through this path of length r is $\Pi_{i=1}^{r} p(e_i)$. For simplicity, we assume the same $p(e)\ \forall e \in E$; hence, the probability of influence spreading becomes p^r. Using these results, we construct diffusion vector $\mathbf{D}$:

$$\mathbf{D}(\mathbf{p}, \mathbf{T})_{\mathbf{nx1}} = \sum\nolimits_{\mathbf{t}\in[\mathbf{1},\mathbf{T}]} \Big(\big(\mathbf{p}\overline{\mathbf{G}}_\sigma\big)^{\mathbf{t}} * \mathbf{1}_{\mathbf{nx1}} \Big) \qquad (6)$$

Here, $\mathbf{D}(\mathbf{p}, \mathbf{T})$ is a column vector of size nx1, $\mathbf{p}$ is the constant propagation probability on the edges, $\mathbf{T}$ is a variable parameter that measures number of hops considered for influence spread (higher values of $\mathbf{T}$ yields more accurate $\mathbf{D}(\mathbf{p}, \mathbf{T})$ but increases the runtime[2]), $\mathbf{1}_{\mathbf{nx1}}$ is a nx1 column vector of 1's and $\overline{G}_\sigma$ is the transpose of G_σ. This formulation is similar to diffusion centrality (Banerjee et al. 2013) where they calculate influencing power of nodes. However, we calculate power of nodes to get influenced (by using $\overline{G}_\sigma$).

Proposition 1. $\mathbf{D_i}$*, the* i^{th} *element of* $\mathbf{D}(\mathbf{p}, \mathbf{T})_{\mathbf{nx1}}$*, upon normalization, gives an approximate probability of the* i^{th} *graph node to get influenced in the next round.*[2]*.*

Consider the set $\triangle = \{i \mid H'[i] = 1 \wedge H[i] = 0 \wedge \alpha[i] = 0\}$, which represents nodes which were uninfluenced in the initial state s ($H[i] = 0$) and which were not selected in the action ($\alpha[i] = 0$), but got influenced by other nodes in the final state s' ($H'[i] = 1$). Similarly, consider the set $\Phi = \{j \mid H'[j] = 0 \wedge H[j] = 0 \wedge \alpha[j] = 0\}$, which represents nodes which were not influenced even in the final state s' ($H'[j] = 0$). Using $\mathbf{D_i}$ values, we can now calculate $T(s, \alpha, s') = \Pi_{i\in\triangle}\mathbf{D_i}\Pi_{j\in\Phi}(1 - \mathbf{D_j})$, i.e., we multiply influence probabilities $\mathbf{D_i}$ for nodes which are influenced in state s', along with probabilities of not getting influenced $(1 - \mathbf{D_j})$ for nodes which are not influenced in state s'.

Observation Probabilities Given action $\alpha \in A$ and final state $s' = \langle H', F'\rangle$, there exists an observation $o(\alpha, s')$, which is uniquely determined by both α and s'. More formally, $o(\alpha, s')$ is given as follows: $o(\alpha, s') = \{F'[i]\ \forall e_i \in \Theta(\alpha)\}$. Therefore, we have the following result:

$$\Omega(o, \alpha, s') = \begin{cases} 1 & \text{if } o = o(\alpha, s') \\ 0 & \text{if } otherwise \end{cases} \qquad (7)$$

Rewards The reward of taking action $\alpha \in A$ in state $s = \langle H, F\rangle$ (denoted by $R(s, \alpha)$) is given as:

$$R(s, \alpha) = \sum\nolimits_{s'\in S} T(s, \alpha, s')(\|s'\| - \|s\|) \qquad (8)$$

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 ran 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 QMDP 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 n (say) MC simulations starting from states sampled from the current belief state. Such approaches depend on a POMDP black box simulator $\Gamma(s_t, \alpha_t) \sim (s_{t+1}, o_{t+1}, r_{t+1})$ 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:

- $\mathbf{o_{t+1}}$:Every edge e in $\Theta(\alpha_t)$ is sampled (either kept or removed) according to the existence probability on the edge in order to generate o_{t+1}.
- $\mathbf{s_{t+1}}$:Let $s_{t+1} = \langle H', F'\rangle$ and $s_t = \langle H, F\rangle$. To get s_{t+1} from s_t and α_t, we normalize $\mathbf{D}(\mathbf{p}, \mathbf{T})_{\mathbf{nx1}}$ to get probabilities of nodes getting influenced. Let $K = \{H[i] = 1 \vee \alpha_t[i] = 1\}$ represent the set of nodes which are certainly influenced. Then, $H'[i] = 1\,\forall i \in K$ and for all other i, $H'[i]$ is sampled according to $\mathbf{D}(\mathbf{p}, \mathbf{T})_{\mathbf{nx1}}[\mathbf{i}]$. Also, $F'[i] = F[i]\,\forall i \notin \Theta(\alpha_t)$ and $F'[i] = o_{t+1}[i]\,\forall i \in \Theta(\alpha_t)$. Note that s_{t+1} calculated this way represents a state sampled according to $T(s_t, \alpha_t, s_{t+1})$. Thus, using $\mathbf{D}(\mathbf{p}, \mathbf{T})_{\mathbf{nx1}}$, we compactly represent $T(s_t, \alpha_t, s_{t+1})\ \forall\{s_t, \alpha_t, s_{t+1}\}$.
- $\mathbf{r_{t+1}}$: $\|\mathbf{s_{t+1}}\| - \|\mathbf{s_t}\|$, where $\|s_{t+1}\|$ is the number of influenced nodes in s_{t+1}.

QMDP 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, QMDP's approximate policy Π is given

[2] https://www.dropbox.com/s/sh8pkiavlyk3zha/appendix.pdf provides details/proofs.

Algorithm 1: PSINET

```
Input: Belief state $\beta$, Uncertain graph $G$
Output: Best Action $\kappa$
1 Sample graph to get $\Delta$ different instances;
2 for $\delta \in \Delta$ do
3     $FindBestAction(\delta, \alpha_\delta, \beta)$;
4 $\kappa = VoteForBestAction(\Delta, \alpha)$
5 $UpdateBeliefState(\kappa, \beta)$;
6 return $\kappa$;
```

by $\Pi(b) = \arg\max_a \sum_s Q(s,a)b(s)$ for belief b. 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 QMDP in PSINET.

Algorithm Flow Algorithm 1 shows the flow of PSINET. In Step 1, we randomly sample all $e \in E_u$ in G (according to $u(e)$) to get Δ different graph instances. Each of these instances is a different POMDP as the h-values of nodes are still partially observable. Since each of these instances fixes $f(e) \ \forall e \in E_u$, the belief β is represented as an un-weighted 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 $\delta \in \Delta$, we find the best action α_δ in graph δ, for the current belief β in step 3. In step 4, we find the best action κ for belief β, over all $\delta \in \Delta$ by voting amongst all the actions chosen by $\delta \in \Delta$. 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 Δ times and use each of these instances to vote for the best action to be taken.

FindBestAction Step 3 uses Algorithm 2, which finds the best action for a single network instance, and works similarly for all instances. For each instance, we find the action which maximizes long term rewards averaged across n (we use $n = 2^8$) 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 follow a uniform random rollout policy (until either termination, i.e., all nodes get influenced, or the horizon is breached) to find 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.

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 a UCT 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 UCT, which returns the action which maximizes this quantity: $\Upsilon(s,a) = Q_{MC}(s,a) + c_0\sqrt{\frac{\log N(s)}{N(s,a)}}$. 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 determines the exploration-exploitation tradeoff for UCT. High c_0 values make UCT choose rarely tried actions more frequently, and low c_0 values make UCT select actions having high $Q_{MC}(s,a)$ to get an even better $Q(s,a)$ estimate. Thus, in every MC simulation, UCT 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 action with the highest votes is chosen. We propose 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 D runs of Algorithm 2.

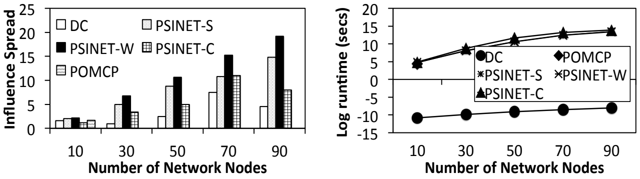
Belief State Update Recall that every MC simulation samples a particle from the belief, after which UCT chooses an action. Upon taking this action, some random state (particle) is reached using the transition probability heuristic. This particle is stored, indexed by the action taken to reach it. Finally, when all simulations are done, corresponding to every action α that was tried during the simulations, there will be a set of particles that were encountered when we took action α 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. PSINET-(S and W) use 20 network

```
Algorithm 2: FindBestAction
  Input: Graph instance $\delta$, belief $\beta$, $\mathbf{N}$ simulations
  Output: Best Action $\alpha_\delta$
  Initialize $counter = 0$;
  while $counter++ < \mathbf{N}$ do
      $s = SampleStartStateFromBelief(\beta)$;
      $a = UCT\_MultiArmedBandit(s)$;
      $\{s', r\} = SimulateRolloutPolicy(s, a)$;
  $\alpha_\delta$ = action with max average reward;
  return $\alpha_\delta$;
```

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 the section. 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 number of nodes "indirectly" influenced by 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

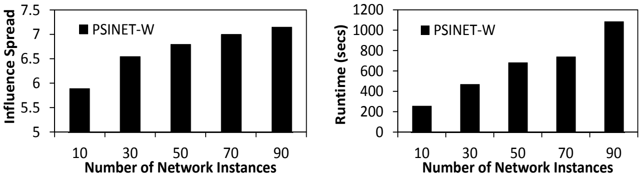
(b) Runtime

Figure 2: 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 2(a), we compare solution qualities of Degree Centrality (DC) (which selects nodes based on their outdegrees, and $e \in E_u$ add $u(e)$ to node degrees), POMCP and PSINET-(S,W and C) on BTER networks of varying sizes. 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 $\sim$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 2(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 2(b) shows that DC runs quickest (as expected) and all PSINET variants run in almost the same time. Thus, Figures 2(a) and 2(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.



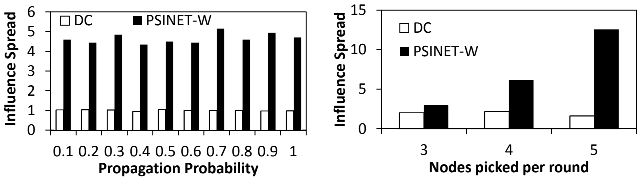
(a) Solution Quality

(b) Runtime

Figure 3: 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 3(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 3(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 $\sim$2X, which shows that increasing instances beyond 30 yields marginal returns.



(a) Varying $p(e)$

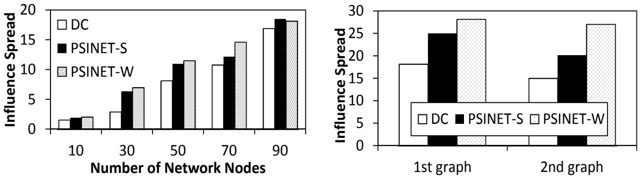
(b) Scaling nodes/round

Figure 4: 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 4(a) shows the solution quality, when PSINET-W and DC are solved with different $p(e)$ values respectively, 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 4(b), we show solution qualities of PSINET-W and DC on a $\langle 30, 31, 27\rangle$ BTER network (horizon=3) and vary number of nodes selected per round (K). The x-axis shows increasing K, and the y-axis shows IIS. This figure shows that even for a small horizon of length 3, which does not give many chances for influence to spread, PSINET-W significantly beats DC. For increasing values of K, PSINET-W beats DC with increasing margins.



(a) ER Networks

(b) Solution Quality

Figure 5: ER networks and Real world networks

Next, Figure 5(a) shows solution quality of DC and PSINET-(S and W) on simple Erdos-Renyi (ER) networks ($p = 0.5$). Even though ER networks do not capture most properties of real-world networks, we run on ER networks to see *how our algorithms perform on a different kind of network*. The x-axis is number of network nodes and the y-axis shows IIS. Figure 5(a) shows that PSINET-(S and W) beat DC on ER networks as well, with PSINET-W beating all other algorithms. However, the difference between DC and PSINET on ER networks is not as much as on BTER networks. However, since real-world networks are rarely similar to ER networks, this result does not invalidate PSINET-W's real-world applicability.

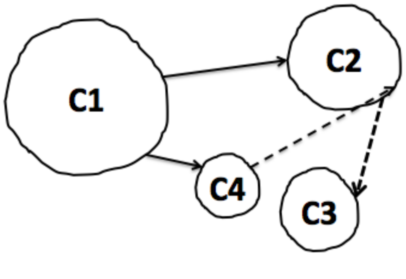


Figure 6: Sample BTER Graph

Real world networks Figure 5(b) compares PSINET variants and DC (horizon = 30) on two real-world social networks (created by our collaborators through surveys and interviews of homeless youth frequenting My Friend's Place's) of homeless youth (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 ∼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 ∼60%.

We now differentiate between the kinds of nodes selected by DC and PSINET-W for the sample BTER network in Figure 6, 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 toward 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 National Institute of Mental Health to conduct such work in only two agencies. Our solution, moving forward would be to use staff at agencies to delineate a first approximation of the 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 in each step). We see this as one of the major strengths of this approach.

Second, our prior research on homeless youth (Rice and Rhoades 2014) suggests that some structurally important youth may be highly anti-social and hence a poor choice for change agents in an intervention such as this. 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, is a secondary issue about protection of privacy for the individuals involved. Agencies collect information on their clients, but most of this information is not to be shared with third parties, such as 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 ∼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

Asai, M., and Fukunaga, A. 2014. Applying problem decomposition to extremely large domains. *ICAPS Workshop on Knowledge Engineering for Planning and Scheduling*.

Banerjee, A.; Chandrasekhar, A. G.; Duflo, E.; and Jackson, M. O. 2013. The diffusion of microfinance. *Science* 341(6144).

Burns, E.; Benton, J.; Ruml, W.; Yoon, S. W.; and Do, M. B. 2012. Anticipatory on-line planning. In *ICAPS*.

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.

Diestel, R. 2005. Graph theory. *Grad. Texts in Math, Springer-Verlag*.

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'14.

Keller, T., and Helmert, M. 2013. Trial-based heuristic tree search for finite horizon mdps. In *ICAPS*.

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.

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.

NAHC. 2011. The link between Homelessness and HIV.

Pomerol, J.-C., and Barba-Romero, S. 2000. *Multicriterion decision in management: principles and practice*. Springer.

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.

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.