PSINET: Assisting HIV Prevention Among Homeless Youth by Planning Ahead

Amulya Yadav, Leandro Soriano Marcolino, Eric Rice, Robin Petering, Hailey Winetrobe, Harmony Rhoades, Milind Tambe, Heather Carmichael

■ *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, and so on. Many nonprofit 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 one single social network Previous work in strategic selection of intervention participants does not handle uncertainties in the social networks' 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: (1) it handles uncertainties in network structure and evolving network state; (2) it addresses these uncertainties by using POMDPs in influence maximization; and (3) it provides algorithmic advances to allow high-quality approximate solutions for such POMDPs. Simulations show that PSINET achieves around 60 percent 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 its officials.*

Homelessness affects around 2 million youths in the United States annually, 11 percent of whom are infected with the human immunodeficiency virus (HIV), 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 (that is, nodes with the most number of friendships picked 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 personnel to direct toward these programs. This fact, along with the emotional and behavioral problems of youth, makes managing groups of more than five or six young people 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 article is to introduce partially observable Markov decision process (POMDP)–based social interventions in networks for enhanced HIV treatment (PSINET), a novel system that 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 friendships between people in the social network; and the 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 we conduct several interventions sequentially, similar to sequential actions taken in a POMDP; and 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 ($\binom{150}{10}$ actions) spaces are beyond the reach of current state-of-the-art POMDP solvers and algorithms. To address this scaleup challenge, PSINET provides a novel online approximation algorithm that relies on the following key ideas: (1) compact representation of transition probabilities (explained later) to manage the intractable state and action spaces; (2) combination of the Q_{MDP} heuristic (a well known offline approximate solver) with Monte Carlo simulations to avoid exhaustive search of the entire belief space; and (3) 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 (MFP),[1] a nonprofit 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 networks. 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 toward final deployment.

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 that can optimally spread influence in a graph. This was followed by many speedup 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 Inf-BM). Some research exists on Inf-BM 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 Inf-BM as a game-theoretic problem and provide scaleup 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 article. For online planning, we mainly focus on the literature on Monte Carlo (MC) sampling–based online POMDP solvers since this approach allows significant scaleups. Silver and Veness (2010) proposed the partially observable Monte Carlo planning (POMCP) algorithm, which uses Monte Carlo tree search in online planning. Also, Somani et al. (2013) present the DESPOT algorithm, which improves the worst-case performance of POMCP. Bai et al. (2014) used Thompson sampling to trade off intelligently 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 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.

Our Approach

Partially observable Markov decision processes 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 that could exist in one of several states at any given point in time and that 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).

More formally, a full description of the POMDP includes the sets of possible environment states, the set of actions that the agent can take, and the set of possible observations that the agent can observe. In addition, the full POMDP description includes a transition matrix for storing transition probabilities, which specify the probability with which the environment transitions from one state to another, conditioned on the immediate action taken. Another component of the POMDP description is the observation matrix for storing observation probabilities, which specify the probability of getting different observations in different states, conditioned on the action taken to reach that state. Finally, the POMDP description includes a reward matrix, which specifies the agent's reward of taking actions in different states.

A POMDP policy Π provides a mapping from every possible belief state (which is a probability distribution over world states) to an action. 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 toward 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.

The 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 specified by its nodes ($\boldsymbol{V}$) and edges ($\boldsymbol{E}$). Let the number of nodes in G be n. Every node in $\boldsymbol{V}$ represents a homeless youth, and every edge of the form $\{e = (B, C)\}$ (where B and C are graph nodes) represents that youth B has nominated (listed) youth C in his or her social circle. Further, the set of edges $\boldsymbol{E}$ consists of a set of certain edges ($\boldsymbol{E}_c$), which represent friendships that we are certain about. Moreover, $\boldsymbol{E}$ consists of a set of uncertain edges ($\boldsymbol{E}_\mathbf{u}$), which represent friendships that 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 or her suspected friends. No edge can be both certain and uncertain.

Each uncertain edge e 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 1 shows an uncertain graph on six nodes (A to F) and seven 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 has an h-value, which is 1 or 0, depending on 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 gets influenced at time step t, it influences each of its one-hop (that is, separated by a single edge) uninfluenced neighbors with a propagation probability $p(e)$ for all future time steps. Moreover, every uncertain edge e has an f-value (which represents a sampled instance of $u(e)$ and is unknown a priori). The f-value is 1 or 0 depending on whether the uncertain edge exists with certainty in the real graph (that is, the youth at the end of that uncertain edge are actually friends) or not (that is, the youth at the end of that uncertain edge are not friends), respectively. For uncertain edges e, 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 that 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

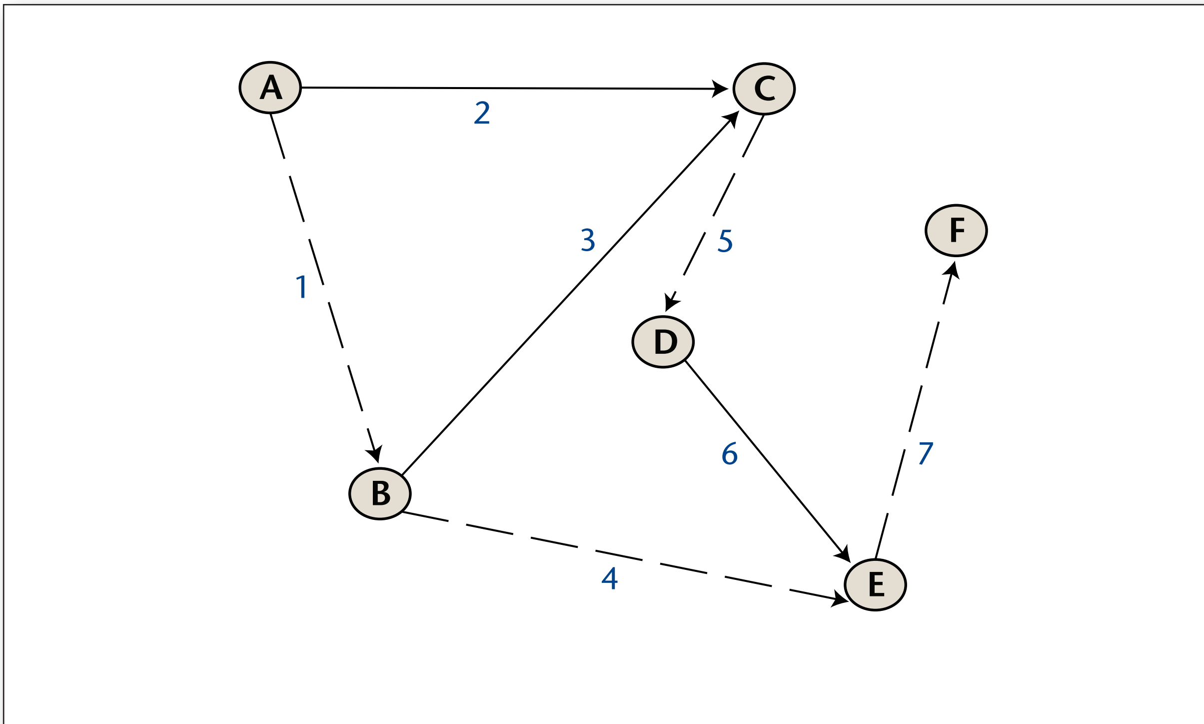

Figure 1. A Sample Six-Node Uncertain Graph.

(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 (or equivalently, their h-value becomes 1) after the 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 first intervention, we are uncertain whether nodes *C* and *E* (adjacent to nodes *B* and *D*) are influenced before nodes for the second intervention are chosen. We now provide a POMDP mapping onto our problem.

States

A state consists of the state of the nodes (that is, whether they are influenced or not), along with the state of the uncertain edges (that is, 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 and is given as input) 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

Previous studies such as Rice et al. (2012b) show that homeless youth are found to be more willing to discuss their social ties in the 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 one-hop social circles, and this is within the agency's capacity. For example, by taking action $\{B, C\}$ in figure 1, the f-values of edges 4 and 5 (that is, uncertain edges in the one-hop social circle of nodes *B* and *C*) would be observed.

Transition Probabilities

Here, we give a high-level intuition for how transition probabilities are computed. Basically, in order to find out the complete transition matrix, we need to calculate the probability of reaching final state s', given that we took action α in initial state s. If we can calculate this probability for fixed values of s, α, and s', we can then find out the entire transition matrix by calculating transition probabilities for all possible combinations of s, α, and s'. Thus, we now discuss how to calculate the transition probability for a given s, α, and s', which we denote by $\boldsymbol{T}(s, \alpha, s')$.

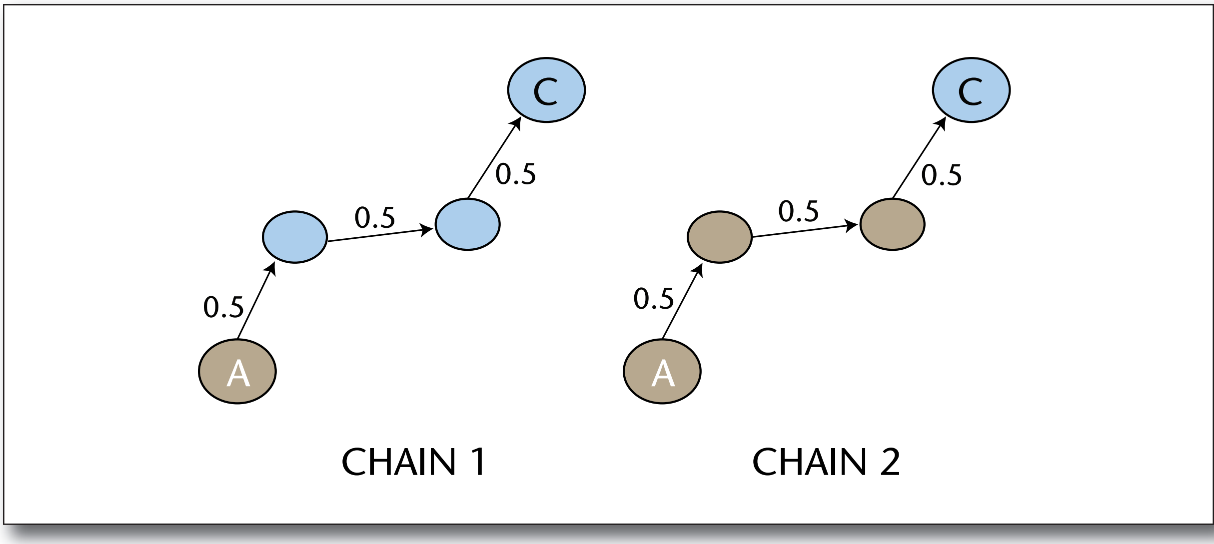

Figure 2. Chains in Social Networks.

For $\boldsymbol{T}(s, \alpha, s')$ to be nonzero, we require that the f-values of all uncertain edges in state s, which were not observed as a result of taking action α, will not change in the final state s'. This is because the f-values of an uncertain edges can only change as a result of observing that uncertain edge. Also, we require that all nodes that were already influenced in the previous state s (that is, those h-values that were 1 in s), remain influenced in the final state s' (that is, these h-values remain 1 in s'), irrespective of the taken action α. Finally, we also require that all nodes that we influence as a result of action α will remain influenced in the final state (that is, the h-values of nodes picked in action α will become 1 in the final state s').

If any of these three conditions is not satisfied, then $\boldsymbol{T}(s, \alpha, s') = 0$. 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 predecided 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 three hops distance to be influenced.

However, we make an important assumption that we describe next. Consider two different chains of length four (nodes) as shown in figure 2. 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 are 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).

We construct a pruned graph G_σ (created from graph G) that only contains edges outgoing from influenced nodes. We prune the graph because influence can only spread through edges that are outgoing from influenced nodes. More details on G_σ can be found in our IAAI paper (Yadav et al. 2015). 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 $\mathbf{D}$, 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 the transition probabilities.

Figure 3 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 2) 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

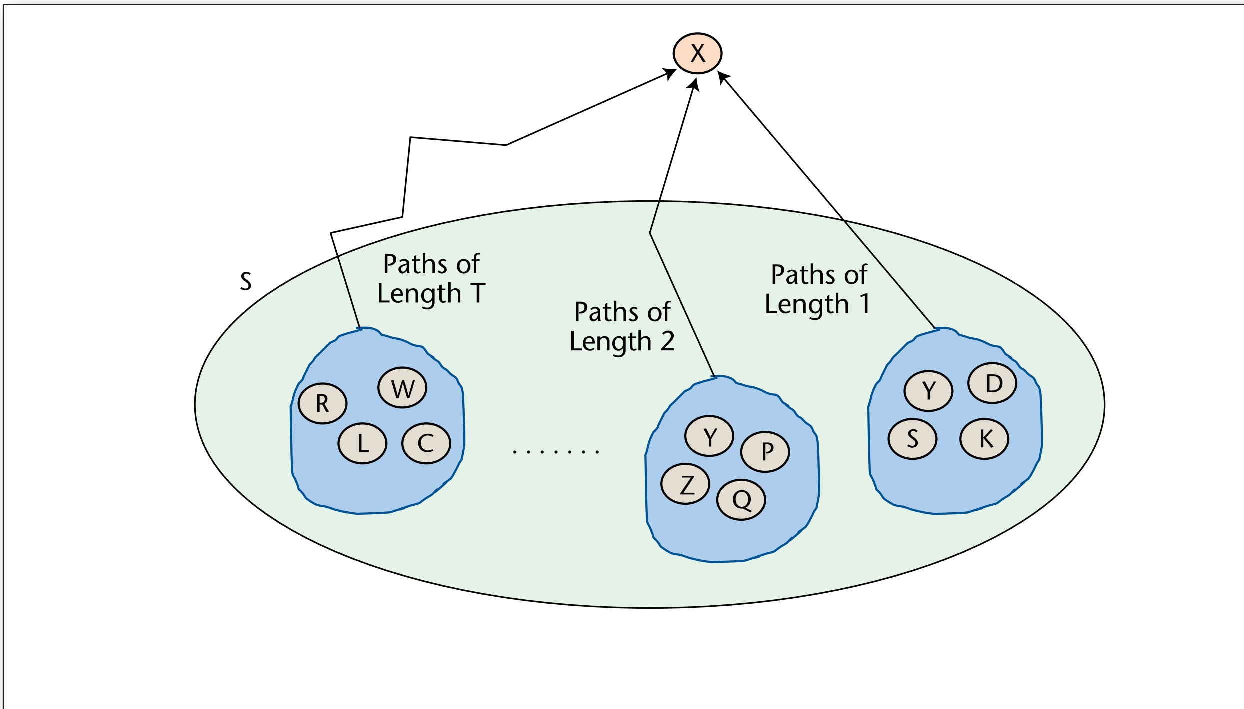

Figure 3. The Intuition Behind our Transition Probability Heuristic.

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 example, all nodes having a path of length 1 (that is, *Y, D, S, K*) are distinguished from all nodes having path of length *T* (that is, *R, W, L, C*). Note that node *Y* has paths of length 1 and 2 to node *X*.

(instead of calculating joint probabilities), our approach produces an approximation.[2]

Observation Probabilities

Here, we give a high-level intuition for how observation probabilities are computed. Basically, in order to find out the complete observation matrix, we need to calculate the probability of observing observation o, given that we took action α to reach final state s'. If we can calculate this probability for fixed values of o, α, and s', we can then find out the entire observation matrix by calculating observation probabilities for all possible combinations of o, α, and s'. Thus, we now discuss how to calculate the observation probability for a given o, α, and s', which we denote by Ω (o, α, s').

Calculating Ω(o, α, s') is trivial as the final state s' already has f-values of all uncertain edges, and you know which uncertain edges you will observe, as a result of knowing the nodes that you pick in action α. Thus, given s' and α, only one observation is possible which is composed of the f-values (from s') of the observed uncertain edges. More details can be found in our IAAI paper (Yadav et al. 2015).

Rewards

The reward of taking action α in state s (denoted by $\mathbf{R}(s, \alpha)$) is the expected number of new influenced nodes (where the expectation is taken with respect to all possible final states s' that could be reached by taking action α in state s).

PSINET

Initial experiments with the ZMDP solver (a software package that contains many offline planning algorithms)[3] 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 that 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 scaleup to the problems of interest in this article. Hence, we propose an 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 Γ, which generates the state, observation, and reward at time $t + 1$, given the state and action at time t, in accordance with the

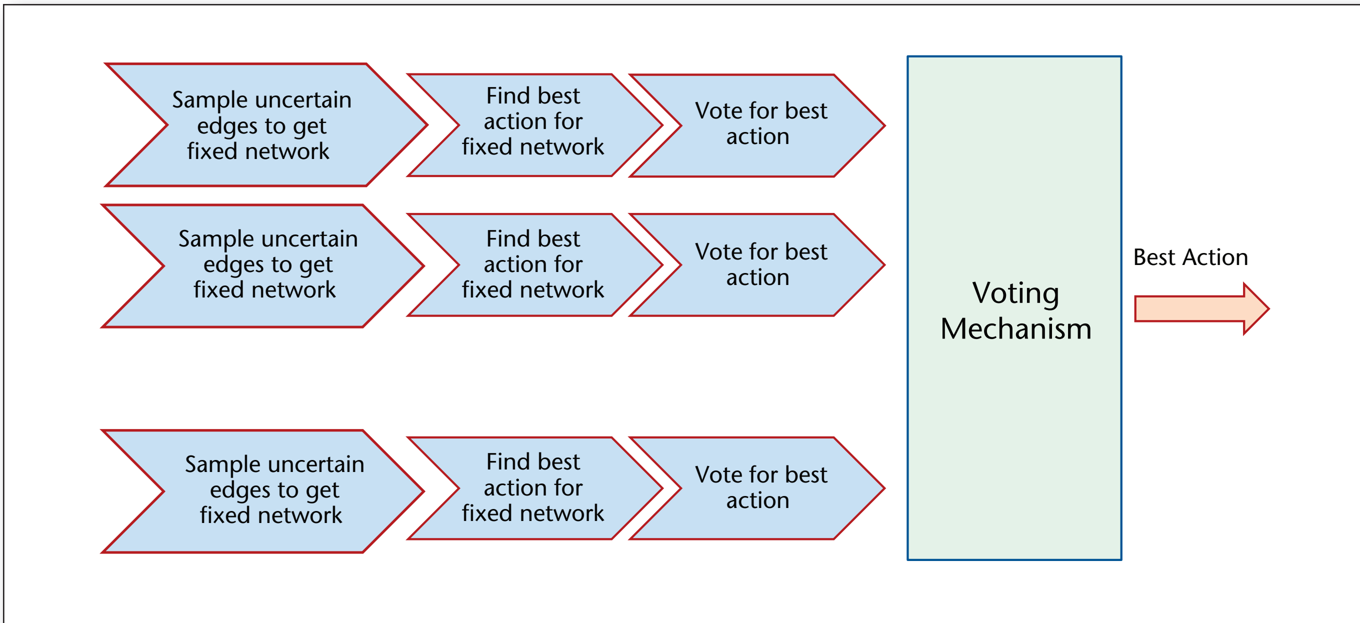

Figure 5. Parallelizing Execution of Several Threads.

POMDP dynamics. More details on the construction of Γ can be found in Yadav et al. (2015).

Q_{MDP}

Q_{MDP} is a well known approximate offline planner, and it relies on $Q(s, \alpha)$ values, which represent the value of taking action α in state s. It precomputes these $Q(s, \alpha)$ values for every (s, α) pair by approximating them by the future expected reward obtainable if the environment is fully observable. On a high level, Q_{MDP}'s approximate policy for a given belief state selects the action that maximizes the expected future value of that belief state. More details on Q_{MDP} can be found in the paper by Littman, Cassandra, and Kaelbling (1995).

Unfortunately, our intractable POMDP state and action spaces makes it infeasible to calculate $Q(s, \alpha)$ for all pairs of (s, α). Thus, we propose to use an MC-sampling-based online variant of Q_{MDP} in PSINET.

ALGORITHM 1 : PSINET

INPUT : Belief State β, Uncertain Graph G
OUTPUT : Best Action κ

1. Sample graph G to get Δ different instances;
2. **For** $\delta \in \Delta$
3. FindBestAction ($\delta, \alpha_\delta, \beta$);
4. κ = VoteForBestAction (Δ, α);
5. UpdateBeliefState (κ , β);
6. Return κ ;

Figure 4. Algorithm 1 Pseudocode.

Algorithm Flow

Algorithm 1 (figure 4) shows the flow of PSINET. In step 1, we randomly sample all uncertain edges *e* in G (according to *u*(*e*)) to get different graph instances (when we repeat this sampling process multiple times), which form a set Δ. Each of these instances is a different POMDP as even though we remove uncertainty about the f-values of all the uncertain edges, the h-values of nodes are still partially observable (that is, 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 the f-value of all uncertain edges, 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 $\delta \in \Delta$, we estimate the best action α_δ 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 5). In step 4, we find our best estimation κ of the optimal action for belief β, by voting among 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.

ALGORITHM 2 : FindBestAction

INPUT : Graph instance δ , belief β , N simulations
OUTPUT : Best Action α_δ

1. Initialize *counter* = 0;
2. **While** *counter*++ < N
3. $\quad$ s = SampleStartStateFromBelief (β);
4. $\quad$ a = UCT_MultiArmedBandit (s);
5. $\quad$ {s', r} = SimulateRolloutPolicy (s, a);
6. α_δ = action with maximum average reward;
7. Return α_δ;

Figure 6. Algorithm 2 Pseudocode.

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 several times to create a set Δ of instances. Then, we use each of these instances to vote for the best action to be taken.

Finding Best Action

Step 3 uses algorithm 2 (figure 6), which finds the best action for a single network instance, and works similarly for all instances. Figure 7 illustrates the details of algorithm 2 (pseudocode in figure 6). For each instance, we find the action that 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 follow a uniform random rollout policy (until either termination, that is, 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 $Q(s, \alpha)$ estimate. Finally, we pick the action with the maximum average reward.

Multiarmed Bandit

We can only calculate $Q(s, \alpha)$ 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 multiarmed 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 trades off exploitation with exploration to come up with an action choice. More details about the implementation can be found in the IAAI paper (Yadav et al. 2015). In essence, 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 three different voting schemes: PSINET-S, PSINET-W, and PSINET-C.

PSINET-S: Each instance's vote gets equal weight.

PSINET-W: Every instance's vote gets weighted differently. This weighting scheme approximates the probabilities of occurrences of real-world events by giving low weights to instances that remove either too few or too many uncertain edges, since those events are less likely to occur. Instances that remove exactly half the number of uncertain edges get the highest weight, since that event is most likely (Yadav et al. 2015).

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 (that is, 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 (figure 6).

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 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.

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 two 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 all experiments. PSINET-S and PSINET-W use 20 network instances and PSINET-C uses 5 network instances (each instance finds its best action five 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.

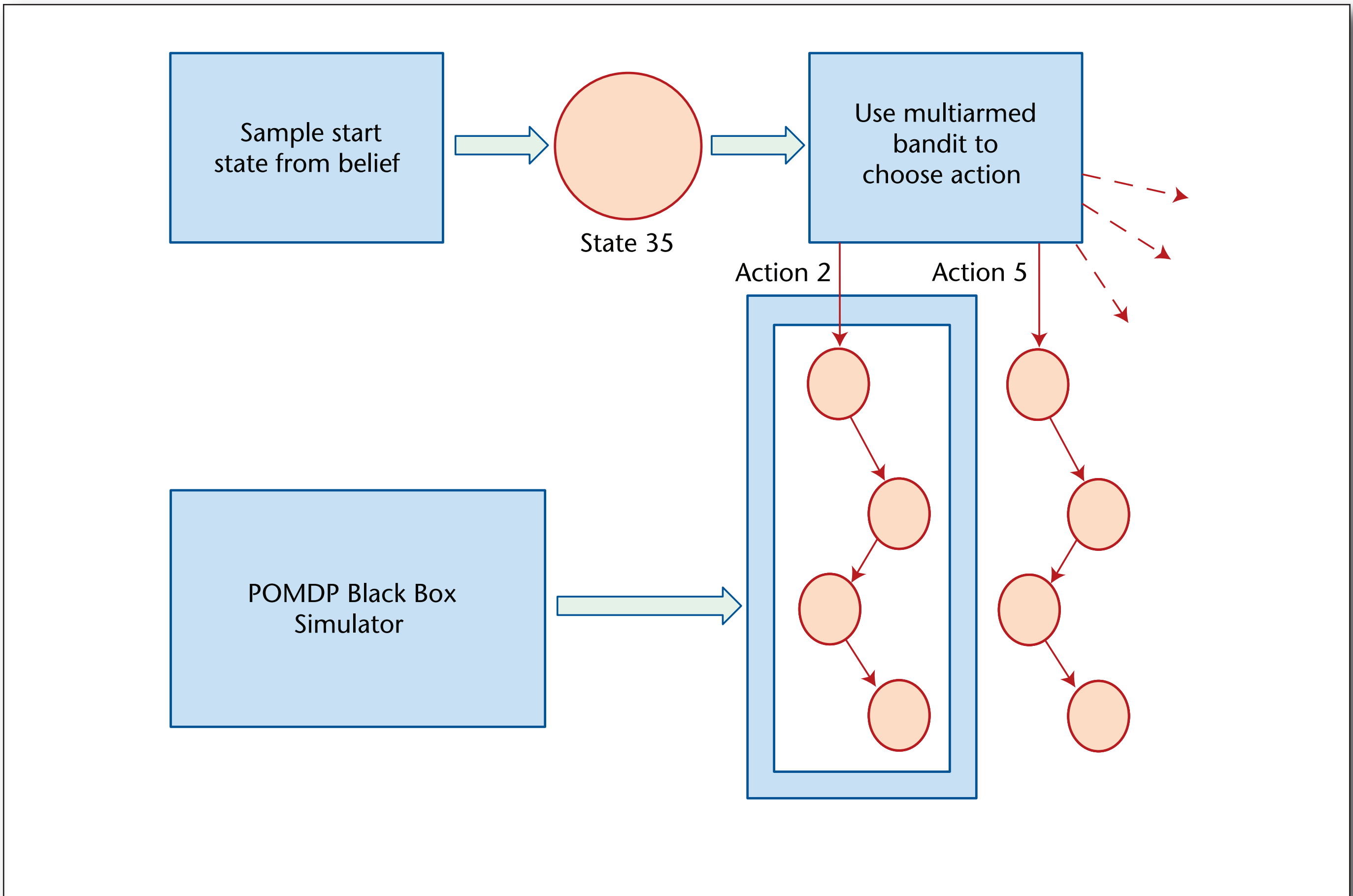

Figure 7. Flow Inside Find Best Action.

[1997]), although we relaxed this assumption later. In this section, a ⟨X, Y, Z⟩ 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* ($\alpha = 0.05$).

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 the 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 8a, we compare solution qualities of degree centrality (DC), POMCP and PSINET-S, PSINET-W, and PSINET-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 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 around 60 percent, which shows the value of our POMDP model. Further, it shows that PSINET-W beats PSINET-S and PSINET-C. Also, POMCP runs out of memory on 30-node graphs.

In figure 8b, we show run times of DC, POMCP, and PSINET-S, PSINET-W, and PSINET-C on the same BTER networks. The x-axis shows the number of network nodes and the y-axis shows log (base e) of run time (in seconds). Figure 8b shows that DC runs quickest (as expected) and all PSINET variants run in almost the same time. Thus, figures 8a and 8b 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

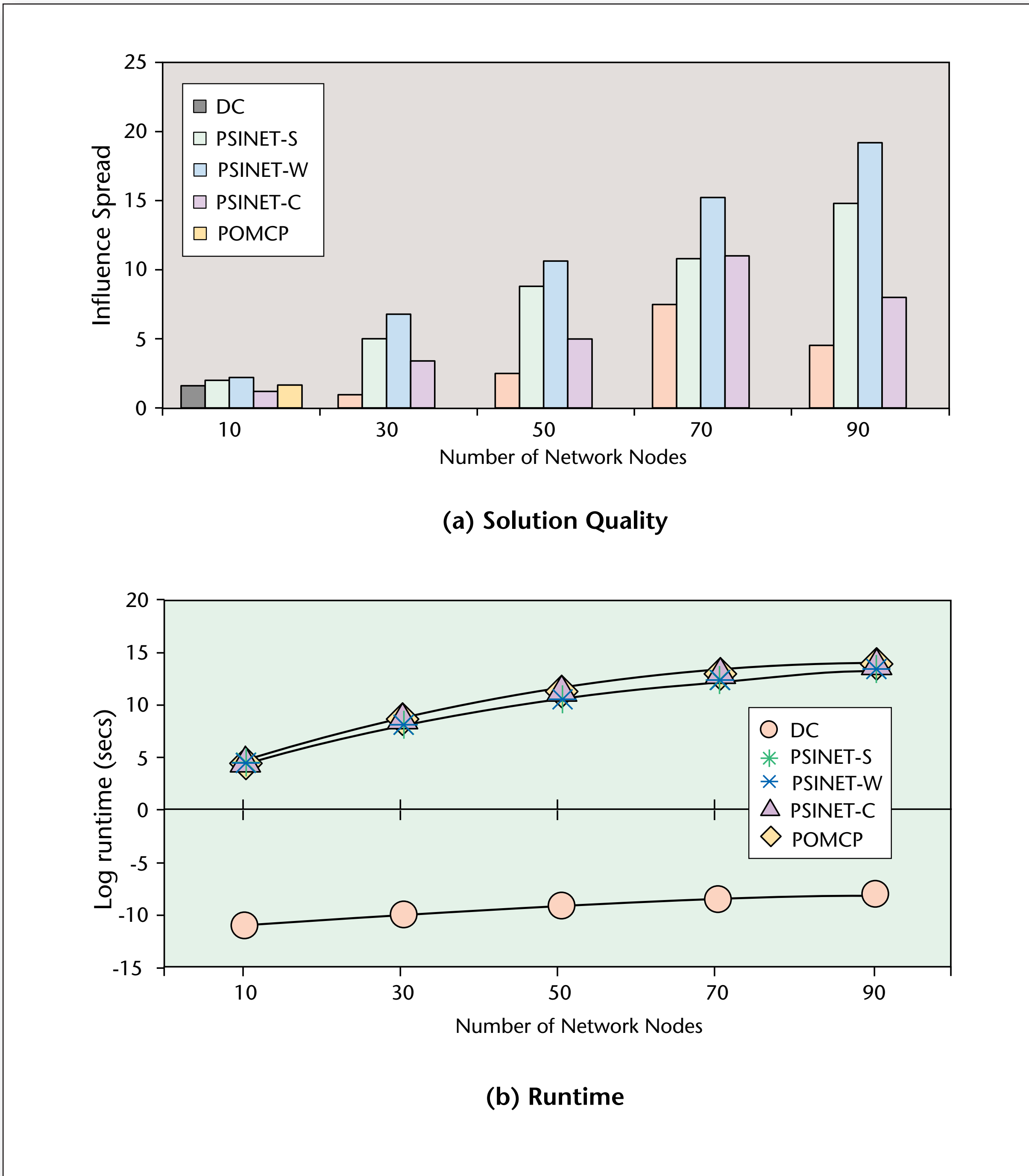

Figure 8. Comparison on BTER Graphs.

PSINET-S in terms of solution quality. Thus, we now focus on PSINET-W.

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 9a, we show solution quality of PSINET-W with increasing network instances, for a ⟨40, 71, 41⟩ BTER network with a horizon of 10. The *x*-axis shows the number of network instances and the *y*-axis shows ΠS. Unsurprisingly, this figure shows that increasing the number of network instances increases ΠS as well.

In figure 9b, we show run time of PSINET-W with increasing network instances, for a ⟨40, 71, 41⟩ BTER network with a horizon of 10. The *x*-axis shows the number of network instances and the *y*-axis shows

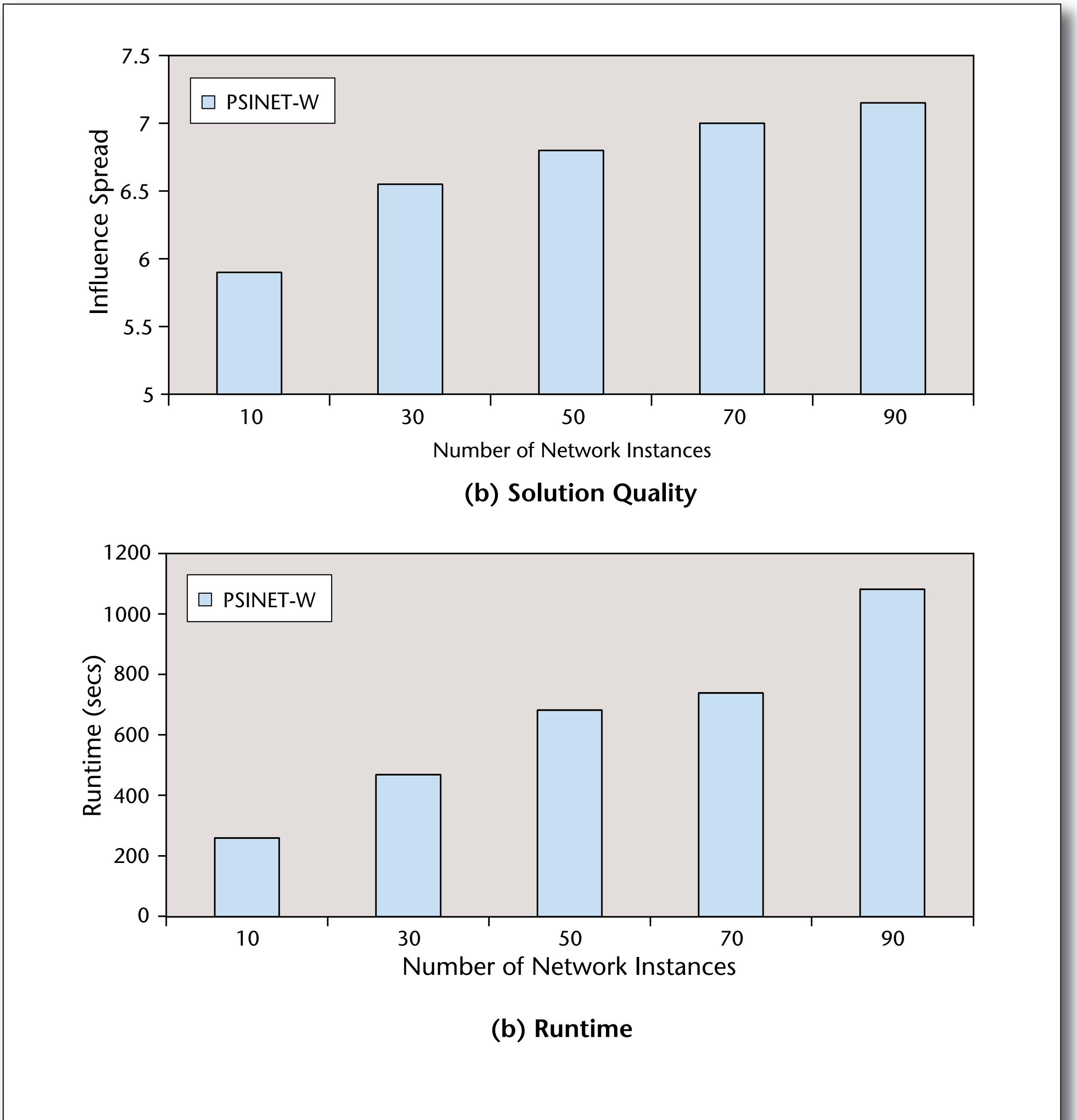

Figure 9. Increasing Number of Graph Instances.

run time (in seconds). This figure shows that increasing the number of network instances increases the run time as well. Thus, a solution quality/run-time trade-off exists, which depends on the number of network instances. A greater number of instances results in better solutions and slower run times and vice versa. However, for 30 versus 70 instances, the gain in solution quality is less than 5 percent whereas the run time is around twice as long, which shows that increasing instances beyond 30 yields marginal returns.

Next, we relax our assumptions about propagation $p(e)$ probabilities, which were set to 1.5 so far. Figure 10a 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 ⟨40, 71, 41⟩ 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 affects PSINET-W's improvement over DC, which shows our

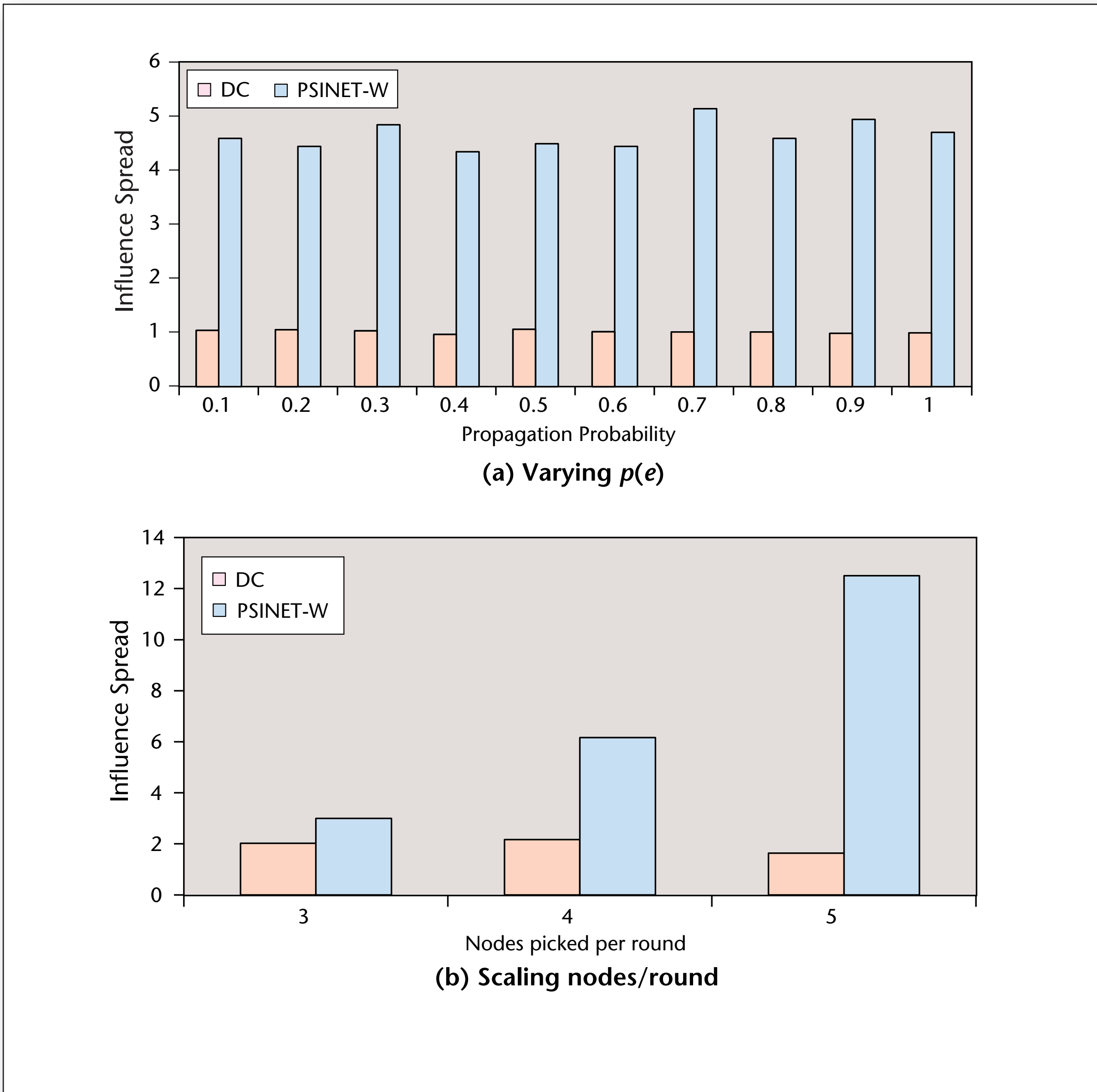

Figure 10. Comparison of Degree Centrality with PSINET-W Across Varying Parameters.

algorithms' robustness to these probability values (we get similar results upon changing $u(e)$).

In figure 10b, we show solution qualities of PSINET-W and DC on a ⟨30, 31, 27⟩ 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.

Real-World Networks

Figure 11 shows one of the two real-world friendship-based social networks of homeless 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 12 compares PSINET variants and DC (horizon = 30) on these two real-world social networks (each of size around ⟨155, 120, 190⟩). 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 around 60 percent, 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 around 60 percent.

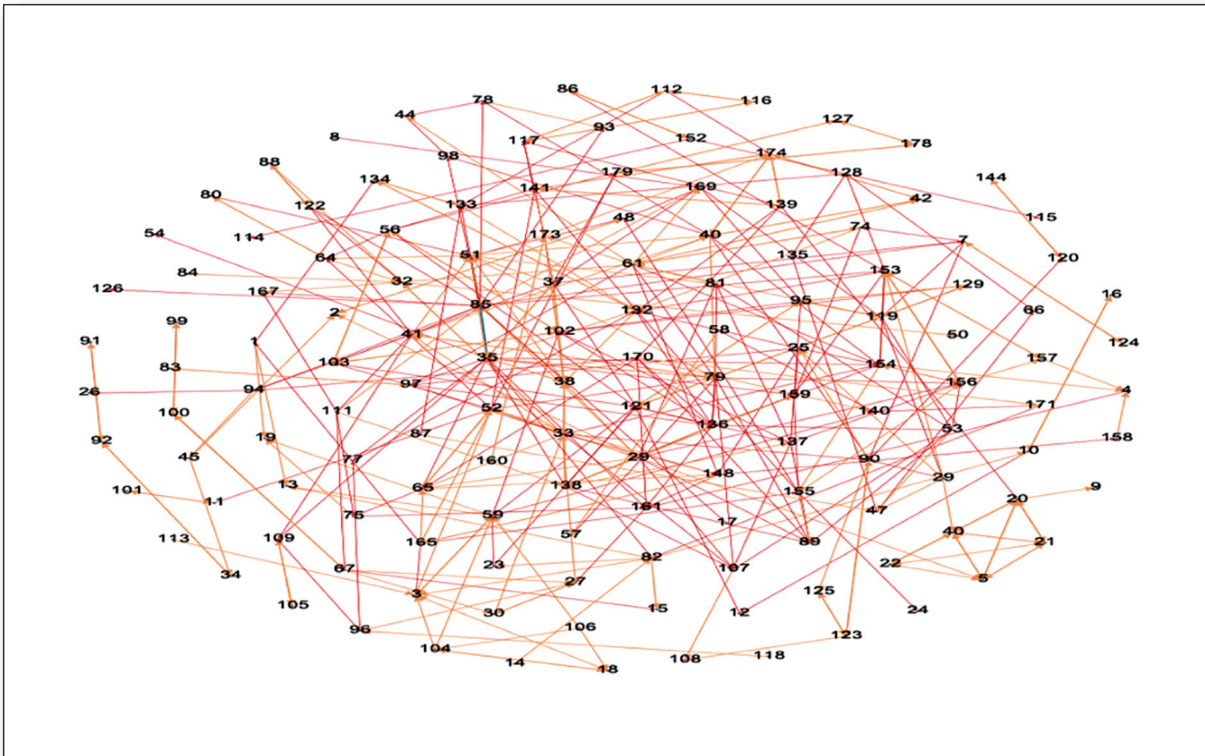

Figure 11. One of the Friendship-Based Social Networks of Homeless People Visiting My Friend's Place.

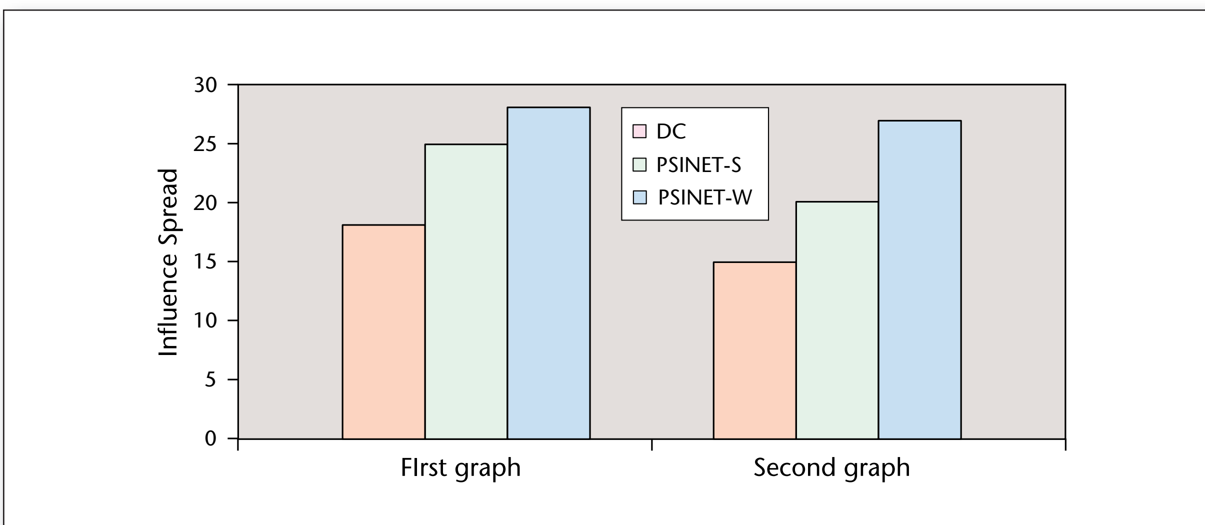

Figure 12. Solution Quality for Real-World Network.

We now differentiate between the kinds of nodes selected by DC and PSINET-W for the sample BTER network in figure 13, which contains nodes segregated into four clusters (C1 to C4), and in which 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 influence 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, and so on.

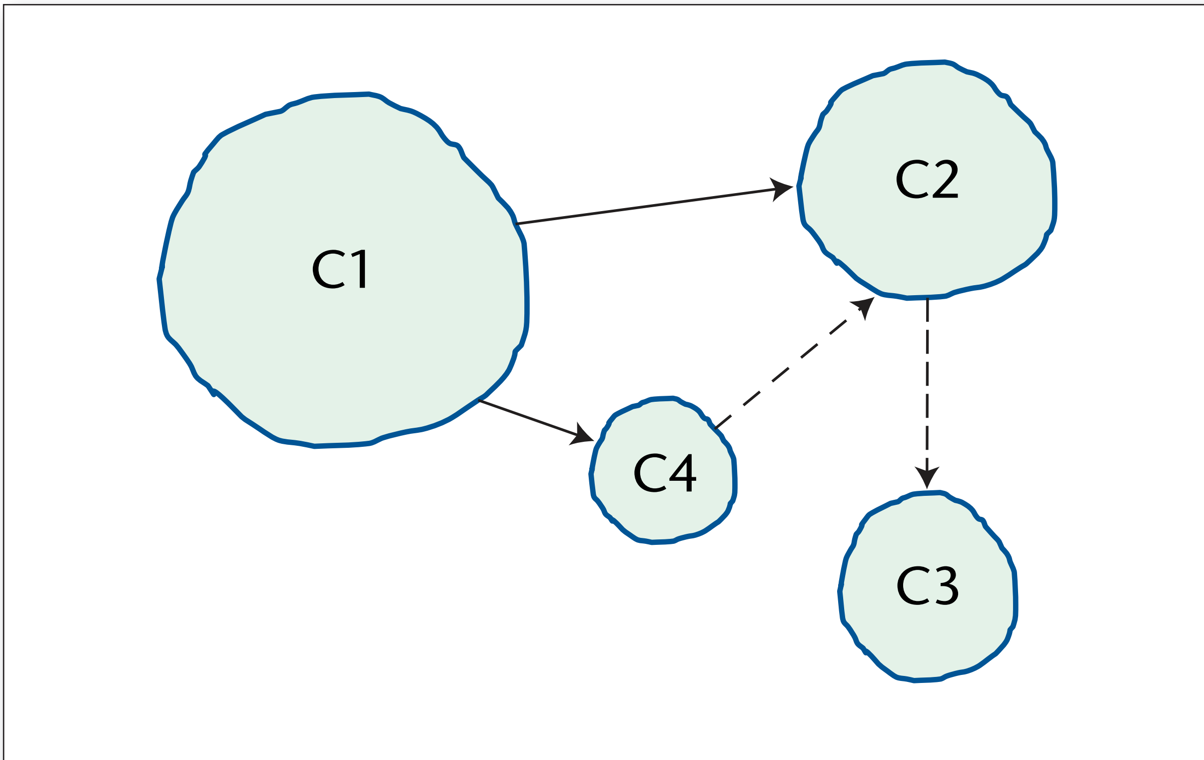

Figure 13. Sample BTER Graph.

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 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 through 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 antisocial and hence a poor choice for change agents in an intervention. We suggest that if such youth are selected by the POMDP program, we then choose the next best action (subset of nodes) that does not include those "antisocial" youth. Thus, the solution may require some ongoing management as certain individuals either refuse to participate as peer leaders or, based on their antisocial 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 that allow them to implement the POMDP program without having to provide identifying information to our team.

Conclusion

This article has presented 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: (1) it handles uncertainties in network structure and evolving network state; (2) it addresses these uncertainties by using POMDPs in influence maximization; and (3) it provides algorithmic advances to allow high-quality approximate solutions for such POMDPs. Simulations show that PSINET achieves around 60 percent improvement over the current state of the art. PSINET was developed in collaboration with My Friend's Place and is currently being reviewed by its officials.

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.

Notes

1. See myfriendsplace.org.

2. See teamcore.usc.edu/people/amulya/appendix.pdf for more details and proofs.

3. See T. Smith (2013), ZMDP Software for POMDP/MDP Planning. www.longhorizon.org/trey/zmdp

References

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

Bai, A.; Wu, F.; Zhang, Z.; and Chen, X. 2014. Thompson Sampling Based Monte-Carlo Planning in POMDPs. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling* (ICAPS 2014). Palo Alto, CA: AAAI Press.

Bharathi, S.; Kempe, D.; and Salek, M. 2007. Competitive Influence Maximization in Social Networks. In *Internet and Network Economics,* Lecture Notes in Computer Science, Volume 4858, 306–311. Berlin: Springer.

Borodin, A.; Filmus, Y.; and Oren, J. 2010. Threshold Models for Competitive Influence in Social Networks. In *Internet and Network Economics,* Lecture Notes in Computer Science Volume 6484, 539–550. Springer.

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. New York: Association for Computing Machinery.

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*. La Jolla, CA: Neural Information Systems Processing Foundation, Inc.

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. New York: Association for Computing Machinery.

Kimura, M., and Saito, K. 2006. Tractable Models for Information Diffusion in Social Networks. In *Knowledge Discovery in Databases: 10th European Conference on Principles and Practice of Knowledge Discovery in Databases* PKDD 2006, Lecture Notes in Computer Science Volume 4213, 259–271. Berlin: Springer.

Kostka, J.; Oswald, Y. A.; and Wattenhofer, R. 2008. Word of Mouth: Rumor Dissemination in Social Networks. In *Structural Information and Communication Complexity: 15th International Colloquium*, Lecture Notes in Computer Science Volume 5058, 185–196. Berlin: Springer..

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. New York: Association for Computing Machinery.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning Policies for Partially Observable Environments: Scaling Up. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML)*. San Francisco: 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. New York: Association for Computing Machinery.

Pomerol, J.-C., and Barba-Romero, S. 2000. *Multicriterion Decision in Management: Principles and Practice,* volume 25, Springer Science & Business Media. Berlin: Springer.

Poupart, P.; Kim, K.-E.; and Kim, D. 2011. Closing the Gap: Improved Bounds on Optimal POMDP Solutions. In *Proceedings of the 21st International Conference on Automated Planning and Scheduling.* Palo Alto, CA: AAAI Press.

Puterman, M. L. 2009. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: 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(1): 141–147.

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 Erdos-Rényi Graphs. *Physical Review E* 85(5): 056109.

Silver, D., and Veness, J. 2010. Monte-Carlo Planning in Large POMDPs. In *Advances in Neural Information Processing Systems 23: Proceedings of the 24th Annual Conference on Neural Information Processing Systems,* 2164–2172. La Jolla, CA: Neural Information Systems Processing Foundation, Inc.

Somani, A.; Ye, N.; Hsu, D.; and Lee, W. S. 2013. DESPOT: Online POMDP Planning with Regularization. In *Advances in Neural Information Processing Systems, 26: Proceedings of the 27th Annual Conference on Neural Information Processing Systems 2013,* 1772–1780. La Jolla, CA: Neural Information Systems Processing Foundation, Inc.

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 *Proceedings of the IEEE International Conference on Social Computing* (SocialCom), 33–38. Piscataway, NJ: Institute for Electrical and Electronics Engineers.

Tsai, J.; Nguyen, T. H.; and Tambe, M. 2012. Security Games for Controlling Contagion. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence* (AAAI). Palo Alto, CA: AAAI Press.

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 *Proceedings of the Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence* (IAAI-15). Palo Alto, CA: AAAI Press.

Amulya Yadav is a Ph.D. candidate in the University of Southern California's Viterbi School of Engineering.

Leandro Soriano Marcolinois is a Ph.D. candidate in the University of Southern California's Viterbi School of Engineering.

Eric Rice is an associate professor in the University of Southern California's School of Social Work.

Robin Petering is a Ph.D. candidate in the University of Southern California's School of Social Work.

Hailey Winetrobe is an MPH project specialist and faculty member in the University of Southern California's School of Social Work.

Harmony Rhoades is a research assistant professor in the University of Southern California's School of Social Work.

Milind Tambe is the Helen N. and Emmett H. Jones Professor in Engineering and a professor in the Computer Science and Industrial and Systems Engineering Departments at the University of Southern California's Viterbi School of Engineering.

Heather Carmichael is executive director of My Friend's Place.