Planning Sequential Interventions to tackle Depression in Large Uncertain Social Networks using Deep Reinforcement Learning

Aye Phyu Phyu Aung[1,2], Senthilnath Jayavelu[2], Xiaoli Li[2], Bo An[1]

[1]*School of Computer Science and Engineering, Nanyang Technological University, Singapore*
[2]*Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore*

Abstract

Studies, with the increasing concern for mental health, have shown that interventions along with social support can reduce stress and depression. However, counselling centers do not have enough resource to provide counselling and social support to all the participants in their interest. This paper helps social support organizations (e.g., university counselling centers) sequentially select the participants for interventions. Meanwhile, Deep Reinforcement Learning (DRL) has shown significant success in learning an efficient policy for sequential decision-making problems in both fully observable environments and partially observable environments with small action space. In this paper, we consider emotion propagation from other neighbours of the influencees, initial uncertainties of mental states and influence in the student network. We propose a new architecture called DRLPSO (Deep Reinforcement Learning with Particle Swarm Optimization) to enhance learning performance in a partially observable environment with large state and action space. DRLPSO consists of two stages: the Discrete Particle Swarm Optimization (DPSO) and Deep Q-learning integrated with Long Short-Term Memory (DQ-LSTM). In the first stage, we apply DPSO by initializing n particles that converge to multiple optimal actions for each belief state. In the second stage, the action with the best Q-value from the DPSO action set is executed to obtain belief and observation (history of action). We evaluated the proposed method empirically with the simulated student networks with mental state propagation compared to the state-of-the-art algorithms. The experimental results demonstrate that DRLPSO outperforms the state-of-the-art DRL methods by an average of 32%.

Keywords: Mental state propagation, Dynamic influence maximization, Deep

Reinforcement Learning, Discrete Particle Swarm Optimization, Deep Q-Learning

1. Introduction

According to the World Health Organization (WHO), mental health has become a significant concern with the estimates of more than 350 million people worldwide are affected by depression. Studies have shown that interventions and social support can reduce stress and depression. In this study, we aim to address a practical and critical decision-making problem in our society - students, in particular, undergraduate or graduate students, face more depression and stress in the current very competitive environment. However, we have limited resources (such as counselling services), to provide intervention for alleviating this serious problem. In our study, we formulate this problem as *a dynamic influence maximization problem* that has uncertain initial mental states and relationships/connections between students that are only updated with the observations obtained along with the interventions. We focus on handling such a dynamic influence maximization problem using Deep reinforcement learning (DRL) which adapts the students' changing mental states and connections in a dynamic environment.

DRL has shown great success in solving problems with uncertainty and high dimensional state space such as continuous control [1, 2] and Atari Learning Environment (ALE) [3]. In the literature, the mentioned problem is formulated as Partially Observable Markov Decision Process (POMDP) where the state of the environment is uncertain, partially observable or even cannot be observed, and arbitrarily long histories of observations are needed to extract sufficient features for optimal action selection [4]. The authors solved it by designing a POMDP solver with abstraction and graph partitioning techniques to break up the social network so that the problem is solvable by the proposed POMDP solver. However, this technique breaks up the student network causing information loss. To overcome this, we propose DRL with Particle Swarm Optimization (DRLPSO) to plan the optimal sequential actions in the uncertain dynamic environment with large action space.

Our key contributions can be summarized as follows:

- We propose Deep Q-learning integrated with Long Short-Term Memory (LSTM) to handle dynamic influence maximization problem with large action space.

- To handle a very large action space of POMDP formulation in DRL, one straightforward way to solve this problem is by randomly sampling a smaller action set, but this may not lead to the optimal solution as the particle is unable to explore the entire action space. To overcome this problem Discrete Particle Swarm Optimization (DPSO) is applied by initializing n particles that search through the entire action space for the optimal action based on a belief state. The Q-network subsequently chooses the action with maximum reward as the predicted action for each round.

- The main advantage of DPSO is to optimize the search of the maximum Q-value action by generating n particles which effectively capture multiple optima in the multi-modal action space.

- We demonstrate the efficiency and effectiveness of DRLPSO in comparison with the existing five DRL algorithms using Monte Carlo simulation. We also compare our DRLPSO solution with the previous POMDP solver solution. The results show that the proposed DRLPSO outperforms the POMDP solver solution in terms of effectiveness.

2. Related Works

Deep Recurrent Q-learning Architectures: DRL is widely used to solve large sequential decision-making problems. In particular, for uncertain environments formulated as POMDPs, Deep Recurrent Q-learning (DRQN) has shown significant performance over the existing approaches. Hence, in this paper, we focus on DRQN methods. To represent the uncertain environment, the problem is formulated as a POMDP and the solution is obtained using DRL. For instance, Egorov (2015) solved the POMDP problem with a deep neural network and represented the Q-function of POMDP with belief and action instead of state and action. Later works such as [5] proposed a recurrent model with convolutional layers and LSTM layer to solve the POMDP environments of ALE. This model represents the Q-function with observation as a parameter instead of using the state. More recently, DDRQN [6] and ADRQN [7] improved the work by adding a history of actions to the Q-function's parameters. However, these works have either considered belief or observation (history of action) individually but not both together. They also do not consider the large action space of dynamic environments.

To overcome this issue, we propose a two-stage DRLPSO that contains DPSO which initializes n particles by acting as a discrete optimizer and Deep Q-learning

integrated with LSTM model (DQ-LSTM) to handle large sequential decision-making problems in a dynamic environment. However, DQ-LSTM alone cannot handle the problem as LSTM cannot have many outputs when the action space is combinatorial explosion $\binom{N}{K}$ where N is the number of students and K is the number of chosen student at each round. The state space also grows exponentially with the number of students in the network i.e., $(\mu + 1)^N$ states. The use of DPSO is that its particles converge to the multiple optima as proven by [8, 9, 10]. The optimal action set from DPSO is then used as the action set to Q-network with the deep neural network architecture that trains for the optimal policy in a dynamic environment.

Students' Stress and Risk of Depression. Many different factors can lead to depression such as genetics, medication, physical or substance abuse and stress [11]. Among them, stress (feeling of frustration, anger and nervousness) is a significant factor for a high risk of depression and anxiety, esp. for university students [12, 13, 14]. Hence, we aim to reduce stress levels to reduce the risk of depression.

There has been evidence shown by the studies that emotions (happiness or stress) can spread from person to person via emotion propagation [15, 16]. Therefore, we construct the emotion propagation model where a person's stress (mental state) is reduced after the intervention, after which he/she spreads his/her happiness through emotion propagation in the network and reduces the stress levels of his/her neighbours. This propagation is one-degree from the seed node since the influence propagation does not normally go beyond that in real-world networks [17]. Since the neighbours' mental states affect a person's mental state both positively and negatively [18], we considered the happy/stressed emotions of each neighbour in the propagation model. We assume that the mental states of intervened students are reduced with certainty considering that the unforeseen external factors would not arise while being monitored during intervention [19].

This student counselling problem of a dynamic environment with large action space is formulated as a POMDP problem and solved with Multi-level Partitioning, Abstraction and Reasoning on top of a POMDP solver in [4]. In this paper, we solve the problem using the proposed DRLPSO method.

3. Problem Description

Nowadays, depression affects many people, reducing their ability to work and socialize. WHO estimated the number of people affected by depression as high as 350 million people worldwide [20]. Moreover, depression can challenge the mortality of the population with the increase of suicide rates and other causes [16].

Since counselling and social support can help mitigate this problem by reducing the stress levels of people [18], counselling services are emerging to help them. In this work, we tackle the problem where a university sets up a counselling center and provides interventions through its counsellors conducting dialogue sessions with several participants to find out about their mental states and provide therapy accordingly.

We consider $\mathbb{T}$ rounds of counsellor's interventions of a group of students and at each intervention, the counsellor selects K students. Specifically, for each intervention, the counsellor obtains the observations about the mental states of the selected students as well as the influence between them and their neighbours, as they are interacting with each other frequently and their stress levels will thus be mutually influenced by one another. The estimates for the mental states of the students which we consider as a belief for the next intervention is updated according to the newly obtained observations. An illustrative example is shown in Appendix (A.1). The objective of the counsellor is to decrease the stress level of all students in the network, reducing the overall risk of depression.

3.1. Interventions in Social Network

The connection network of N students is represented by a directed graph $G = \langle V, E\rangle$ where $V(|V| = N)$ represents the nodes and E represents the edges. Every $i \in V$ represents a student in the network and the connection $e = \{(i, j)|i, j \in V\} \in E$ represents that student i is a friend of student j and w_{ij} represents how closely student i is associated to j. We refer the term w_{ij} as the ***influence*** that i induces to student j. Since the friendship between a pair of students is mutual [21], we represent the network as the bidirectional graph where $(i, j) \in E$ and $(j, i) \in E$. However, w_{ij} may not be equal to w_{ji}, this depends on how they influence on each other and we set $w_{ii} = 0$. Let $\mathcal{N}^{in}(i)$ and $\mathcal{N}^{out}(i)$ be the incoming and outgoing associated neighbours, i.e., for incoming neighbours, $(j, i) \in E$ with $0 < w_{ji} \leq 1$ for $j \in \mathcal{N}^{in}(i)$, and for outgoing neighbours, $(i, j) \in E, 0 < w_{ij} \leq 1$ for $j \in \mathcal{N}^{out}(i)$. The mental state of a student is one of the values in the discrete set $\mathcal{M} = \{0, 1, 2, \cdots, \mu\}$[1] in which 0 represents the stress-free state and μ represents the highest stress level. Therefore, the students' mental states are represented by $\mathbf{v} = \langle v_1, ..., v_N\rangle$ where $v_i \in \mathcal{M}, i \in V$ is the mental state of student i.

Here, we assume that in each intervention, counsellor decreases the selected

[1] In the current literature, the mental states can only be roughly evaluated by some in-explicit words, such as *mild*, *moderate* and *severe* depressive episodes [20].

student's stress level by a positive integer value δ[2]. Due to the change in the student's mental state, his/her emotion will propagate to the associated friends $j \in \mathcal{N}^{out}(i)$ by 1-hop propagation where the extent of influence varies by w_{ij}.

The extent of influence of i on j is represented by $\Delta_{i \to j}$ which is defined as:

$$\Delta_{i \to j} = \lfloor \frac{w_{ij}(\mu - v_i)}{w_{ij}(\mu - v_i) + \sum_{k \in \mathcal{N}^{in}(j) \setminus \{i\}} v_k \cdot w_{kj}} \cdot \delta \rfloor \tag{1}$$

where $k \in \mathcal{N}^{in}(j) \setminus \{i\}$ which is the set of students who are associated to j by excluding i. The equation implies that when the influencer i is less stressed, v_i is smaller and $\Delta_{i \to j}$ is larger. When $\sum_{k \in \mathcal{N}^{in}(j) \setminus \{i\}} v_k \cdot w_{kj}$ is larger, i.e., other associated neighbours are more stressed and $\Delta_{i \to j}$ is smaller. Hence, the total reduction of j's mental state value is given by,

$$\Delta j = a_j \cdot \delta + \sum\nolimits_{a_i = 1, i \in V \setminus \{j\}} \Delta_{i \to j} \tag{2}$$

where $\mathbf{a} = \langle a_i \rangle, \forall i \in V$ such that $a_i = 1$ if student i is selected, otherwise $a_i = 0$. In Eq. (2), the first term $a_j \cdot \delta$ is the influence induced by the counsellor and the second term is the influence induced by the propagation from the intervened neighbours of j such that $\Delta_{i \to j}$ are aggregated for all incoming neighbours i of j that are intervened. δ is the discrete value of intervention effectiveness on students' mental states set by the intervention professionals.

3.2. Uncertainties and Partial Observability

The counselling centers do not have any prior information of the students' mental states and influencing neighbours. Considering this, we model the uncertainty of students' mental states at the t^{th} intervention to be $\hat{P}_{t-1}$ which is of size $N \times (\mu + 1)$ where each row $\hat{\mathbf{p}}_i^{t-1} = \langle \hat{p}_i^{t-1}(m) \rangle$ is the probability distribution over the discrete set $\mathcal{M}$ of student i. $\hat{p}_i^{t-1}(m)$ is the probability of student i being evaluated as mental state $m \in \mathcal{M}$ at t. For the uncertain influence, we also define $\hat{W}_0$ which is of size $N \times N$ containing the estimates of influence between each pair of students.

Initially, we set the values of $\hat{P}_0$ and $\hat{W}_0$ as the available information on students. In each intervention, the mental states of the selected students and the influence between them and their neighbours are observed. Hence, in t^{th} intervention,

[2]If $v_i < \delta$, we assign $v_i = 0$ after decrease. This also applies to Δ_j in Eq (2).

counsellor derives $\hat{P}_t$ from the belief which is updated during the intervention. The rule for the belief update is described in the next section. $\hat{W}_t$ is also updated by assigning $\hat{w}_{ij} = w_{ij}, \forall j \in \mathcal{N}^{out}(i)$ and $\hat{w}_{ji} = w_{ji}, \forall j \in \mathcal{N}^{in}(i)$ for each intervened student i.

3.3. POMDP Formulation

POMDPs are sequential decision-making models under uncertainty [22]. Formally, a POMDP is defined as $\mathcal{P} = \langle S, A, O, T, \Omega, R, b^0 \rangle$ where S is the state set, A is the action set, O is the set of observations, T and Ω are the transition and observation probabilities respectively, R is the reward and b^0 is the initial belief over the states.

States and Initial Belief (S and b^0) : A state in S is defined as $\mathbf{s} = \langle \mathbf{v}, \hat{W} \rangle$ where $\mathbf{v}$ denotes the students' mental states and $\hat{W}$ is defined as $\hat{w}_{ij} = w_{ij}$ if the influence of student i on j is known by counsellor, otherwise $\hat{w}_{ij} = \hat{w}^0_{ij}$, where $\hat{w}^0_{ij}$ is the initial estimation of w_{ij} by counsellor. The counsellor has an initial belief b^0 which is a probability distribution over S and $b^0_{\mathbf{s}}$ is the probability that the POMDP is at $\mathbf{s}$ in the beginning of the interventions.

Actions, Observations and Observation Probability (A, O, Ω): The counsellor's selection of K students at each intervention is defined as action $\mathbf{a}$: if $a_i = 1$, the student i is selected; $a_i = 0$ otherwise, given the constraint $\sum_{i \in N} a_i = K$. All actions belong to the action set A. The counsellor's observation by taking the action $\mathbf{a} \in A$ at state $\mathbf{s}$ is defined as $o(\mathbf{s}, \mathbf{a}) = \{v_i, w_{ij}, w_{ji} | \forall a_i = 1, j \in V, \mathbf{v} \in \mathbf{s}\}$, i.e., the mental states and the associated influence of the intervened students. All observations belong to the set O. Ω is the observation function of the POMDP which is uniquely defined by the action $\mathbf{a}$ and the state $\mathbf{s}$. The observation function is defined as normal distribution $f_i(v_i, 1)$ to allow the uncertainties in the evaluation for the student's mental states and the influence between the associated neighbours' during intervention which is given by,

$$\Omega(o, \mathbf{s}, \mathbf{a}) = \prod_{v_i \in o} f_i(v_i | \nu_i), \nu_i \in o(\mathbf{s}, \mathbf{a}) \tag{3}$$

$$f_i(v | \nu_i) = \frac{1}{2\pi} e^{-\frac{v - \nu_i}{2}} \tag{4}$$

where mean ν_i is the mental state value of a student from the observation obtained by taking action $\mathbf{a}$ and variance is 1.

Transition Probabilities Heuristic (T) : In this phase, the counsellor takes action $\mathbf{a}$ and the change of students' mental states ($\mathbf{s} \rightarrow \mathbf{s}'$) is calculated using Eq. (2).

This change of states is denoted by $T(\mathbf{s}, \mathbf{a}, \mathbf{s}')$ [3],

$$T(\mathbf{s}, \mathbf{a}, \mathbf{s}') = \begin{cases} P_{eff}, & \text{if } \mathbf{s}' = \langle \mathbf{v}', \hat{W}' \rangle; \\ (1 - P_{eff}), & \text{if } \mathbf{s}' = \langle \mathbf{v}, \hat{W}' \rangle; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where $\mathbf{v}'$ and $\hat{W}'$ are students' mental states and influence of the new state $\mathbf{s}'$ that are updated by,

$$v'_j = v_j - \Delta j \tag{6}$$

$$\hat{w}'_{ij} = \begin{cases} w_{ij}, & \text{if } a_i = 1 \text{ or } a_j = 1; \\ \hat{w}_{ij}, & \text{otherwise.} \end{cases} \tag{7}$$

where $v'_j \in \mathbf{v}', \hat{w}'_{ij} \in \hat{W}', \forall i, j \in V$.

Reward and Policy (R and π) : The reward $R(\mathbf{s}, \mathbf{a})$ of taking action $\mathbf{a} \in A$ in state $\mathbf{s} = \langle \mathbf{v}, \hat{W} \rangle$ is defined by,

$$R(\mathbf{s}, \mathbf{a}) = \sum\nolimits_{\mathbf{s}' \in S} T(\mathbf{s}, \mathbf{a}, \mathbf{s}') \left(\sum\nolimits_{i \in V} (v_i - v'_i) \right) \tag{8}$$

We define the history at intervention t as a sequence of past actions and observations $H_t = \{\langle a_1, o_1 \rangle, \ldots, \langle a_t, o_t \rangle\}$. We denote $\mathcal{H}_t$ as the set of all possible histories at t. The policy is defined as $\pi : \mathcal{H}_t \rightarrow A$ which takes in history H_t as input and outputs the action $\mathbf{a}$. The expected reward for π starting from b^0 is defined as $V^\pi(b^0) = \sum_{t=1}^{\mathbb{T}} \mathbb{E}[R(\mathbf{s}_t, \mathbf{a}_t) | b^0, \pi]$ where $\mathbb{E}[\cdot]$ outputs the expected value of the input. The optimal policy π^* maximizes $V^\pi(b^0)$ given by,

$$\pi^* = \arg\max_{\pi} V^\pi(b^0) \tag{9}$$

Belief Update: In each state $\mathbf{s}$, we have the deterministic value of $\hat{W}$ where each element is either $\hat{w}_{ij}$ or w_{ij}. Hence, the initial belief b^0 can be defined by $\hat{P}_0$ and $\hat{W}_0$ such that for $\mathbf{s} = \langle \mathbf{v}, \hat{W} \rangle$,

$$b^0_{\mathbf{s}} = \begin{cases} \prod_{v_i \in \mathbf{v}} \hat{p}^0_i(v_i), & \text{if } \hat{W} = \hat{W}_0 \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

[3] P_{eff} is the probability value defined according to statistics for effectiveness of therapy.

At intervention t, each state $\mathbf{s} = (\mathbf{v}, \hat{W})$ with belief $b_{\mathbf{s}}^{t-1}$ transits to $s' = (\mathbf{v}', \hat{W}')$ upon taking action $\mathbf{a}$ and the counsellor obtains $o \in O$ with the probability of $\Omega(o, \mathbf{s}', \mathbf{a})$. Further, belief is updated using,

$$b_{\mathbf{s}'}^{t} = \gamma \cdot \Omega(o, \mathbf{s}', \mathbf{a}) \cdot \sum\nolimits_{\mathbf{s} \in S} T(\mathbf{s}, \mathbf{a}, \mathbf{s}') \cdot b_{\mathbf{s}}^{t-1} \tag{11}$$

where γ is the normalizing constant given by $\gamma = 1/(\sum_{\mathbf{s}' \in S} \Omega(o, \mathbf{s}', \mathbf{a}) \cdot \sum_{\mathbf{s} \in S} T(\mathbf{s}, \mathbf{a}, \mathbf{s}') \cdot b_{\mathbf{s}}^{t-1})$. $\hat{P}_t$ is updated based on the belief update using $\hat{p}_j^t(m) = \sum_{\mathbf{s}' \in S, v_j' = m} b_{\mathbf{s}'}^t$.

4. Deep Reinforcement Learning with Particle Swarm Optimization

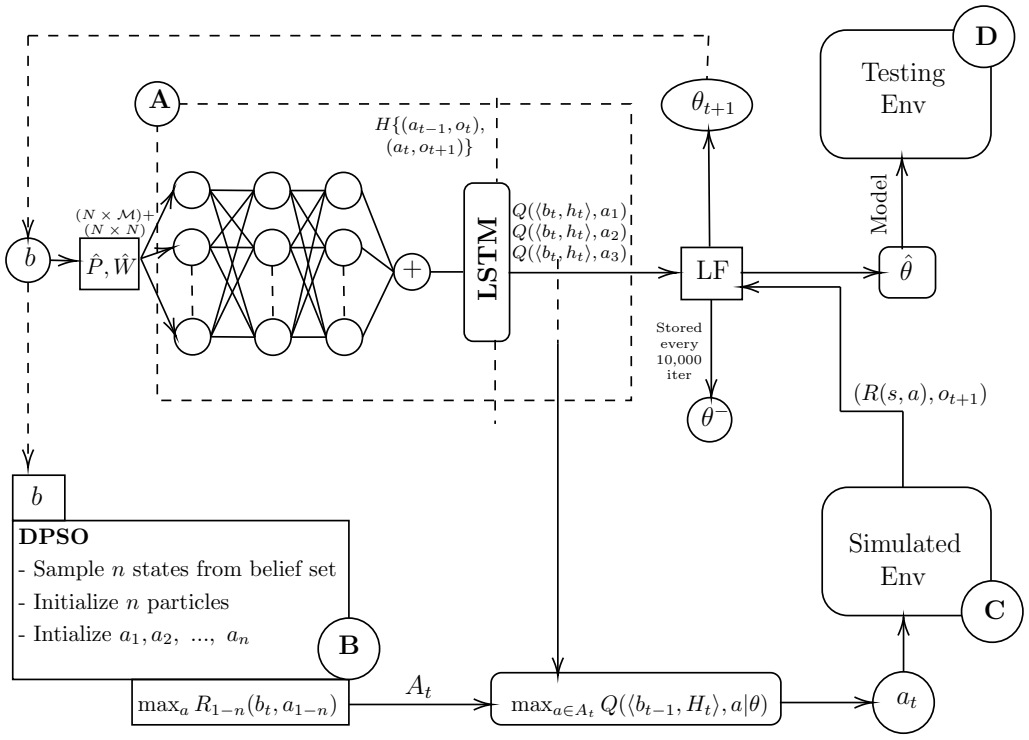

Figure 1: DRLPSO Architecture

As discussed in section 2 (Related Works), dynamic influence maximization is NP-Hard which is a challenging task to obtain the optimal solution. Thus, solving by the brute force method is infeasible. Moreover, being formulated as POMDP, the traditional RL method and DQN are infeasible due to Q-function being represented with state values instead of partial observation and no historical infor-

mation is preserved. Thus, we use deep Q-learning integrated with LSTM layer (DQ-LSTM). However, the state and action space of this problem grow exponentially with the number of students in the network i.e., $(\mu + 1)^N$ states and the number of chosen student at each round i.e., $\binom{N}{K}$ actions. For this problem, the output layer of LSTM could be complicated as it cannot handle the large action space as it results in a large number of outputs. This could reduce the performance of deep recurrent Q-network. Hence, in this study, we embed DPSO to pick the optimal action. By doing so, the training phase of DQ-LSTM can converge to an optimal policy.

Instead of randomly selecting an action, DPSO initializes n particles to optimize the reward function and later converge to a set of optimal actions for a given belief. DQ-LSTM gets an action set optimized by DPSO for each step. This also makes the deep Q-learning converge to the optimal solution. Therefore, we propose DRLPSO (Deep Reinforcement Learning with Particle Swarm Optimization) where DPSO finds the optimal action for each step and Deep Q-learning integrated with LSTM ensures the optimal policy for the dynamic environment.

4.1. DRLPSO Architecture

The overview of DRLPSO is shown in Figure 1. We initialize the problem as a discrete environment that takes action as input and outputs observation and reward. The system consists of 4 major components (Blocks A, B, C and D of Figure 1).

- Deep Q-learning with LSTM as the top layer (Block A)
- Discrete Particle Swarm Optimization Model to predict the best action (Block B)
- Simulated Environment to train the network (Block C)
- Testing Environment (Block D)

Block A is used for training the parameters to predict the Q-values. The deep Q-learning consists of ReLU as activation function, 3 hidden layers of neurons and learning are performed by backpropagation. LSTM is integrated to retain historical information. The training procedure consists of 1) Choose the action for best Q-value using DPSO (Block B); 2) Execute the action in the simulated environment (Block C), obtain observation and reward; 3) Optimize parameter θ by backpropagation until the loss converges. After the training, we test the model in

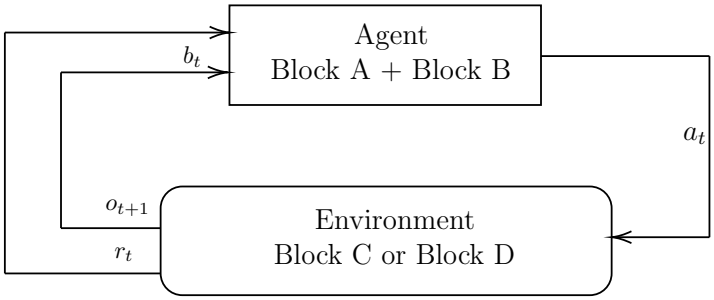

Figure 2: DRLPSO components represented as Markov Model commonly used in RL

the testing environment (Block D). The components are represented as a Markov Model in Figure 2.

Discrete Particle Swarm Optimization: In training the deep recurrent Q-network, existing methods randomly sample an action, without considering the observation received from the environment. Instead, DRLPSO uses DPSO (Block B) which takes belief state updated with the observation. We initialize n particles to explore and exploit by parallelly converging to multiple optima. For DQ-LSTM, we adapt the action set from the convergence of the DPSO particles to the output layer of LSTM.

We modified the traditional continuous PSO [23] to the Discrete PSO. We initialize the position boundary B to be $nCr(N, K)$ i.e., the number of different, unordered combinations of K students from the network with N students. We randomize the swarm positions from 0 to $B - 1$ as the initial discrete actions. While optimizing the objective function defined in Eq. (8), we update the velocity $\vec{V}_i$ and position X_i values of each particle by,

$$\vec{V}_i(t+1) \;=\; \vec{V}_i(t) + (c_1 r_1)(pbest_i - X_i(t)) + (c_2 r_2)(gbest_i - X_i(t)) \quad (12)$$

$$X_i(t+1) = \begin{cases} 0, & \text{if } \lceil X_i(t) + \vec{V}_i(t+1) \rceil < 0 \\ B, & \text{if } \lceil X_i(t) + \vec{V}_i(t+1) \rceil > B \\ \lceil X_i(t) & +\vec{V}_i(t+1) \rceil, \text{ otherwise} \end{cases} \quad (13)$$

where c_1, c_2 are self confidence values, r_1, r_2 are randomized values and $pbest_i$ is the maximum reward position (action) the particle has visited, and the best among all the subswarm particles is stored in $gbest_i$. Generally, $gbest_i$ will converge to a single solution if there exists only one optimum. In this problem, the search space

consists of multiple optimal solutions. Hence, the particles will form a subswarm and converge to multiple peaks to map with the output layer of DQ-LSTM.

Deep Recurrent Q-learning: After we obtain the predicted action, we execute the action to the simulated environment (Block C) to obtain the reward $R(s, a)$ and observation o_{t+1}. We then update the belief b_t and history h_{t+1} to calculate the predicted loss $\hat{y}$ and the actual loss y. DQN uses experience replay and two networks to train parameter θ: the main network to optimize θ value and targeted network to retain θ^- which is updated every 10,000 iterations [24]. Similarly, we adopt experience replay and the two networks and use the stale updated θ^- given by the target network to get the actual Q-values. This technique has been empirically shown to be tractable and stable.

Generally, the state transition problems with MDP considers the Q-update using state and action given by,

$$Q(\mathbf{s}_t, \mathbf{a}_t) \;=\; Q(\mathbf{s}_t, \mathbf{a}_t) \,+\, \alpha(R(\mathbf{s}_t, \mathbf{a}_t) \,+\, \gamma \max_{\mathbf{a}} Q(\mathbf{s}_{t+1}, \mathbf{a}) \,-\, Q(\mathbf{s}_t, \mathbf{a}_t)) \quad (14)$$

where α is the learning rate and γ is the discount factor. This Q-function is well-suited for MDP where the states are fully observed. But for the problems involving POMDP, Q-function is parametrized by belief (b), action (a) and observation (o) since $s \neq o$ due to the partial observability and we cannot observe the current state $\mathbf{s}_t$ in POMDP. We have to use a deep network to better estimate Q-values from belief and observation. We denote the weights and biases of the Q-networks as θ and the function can be denoted by $Q(\langle b_{t-1}, H_t\rangle, a_t|\theta)$. We first calculate the Q-values for each belief, history of action and observation and subsequently update the parameters of the deep network by minimizing the loss function at each iteration i of the training phase.

Actual Q-value is calculated by,

$$y = R(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}} Q(\langle b_t, H_{t+1}\rangle, \mathbf{a}|\theta_i^-) \quad (15)$$

Predicted Q-value is calculated by,

$$\hat{y} = Q(\langle b_{t-1}, H_t\rangle, a_t|\theta_i) \quad (16)$$

The l_2 loss function (LF) is given by,

$$L(\langle b_t, H_t\rangle, a_t|\theta_i) = (y - \hat{y})^2 \quad (17)$$

Finally, we perform backpropagation to train θ which is updated using,

$$\theta_{i+1} = \theta_i + \alpha \nabla_\theta L(\theta_i) \tag{18}$$

When the training converges, we save the model $\hat{\theta}$ and execute the testing environment (Block D) for the $\mathbb{T}$ rounds to get the optimal policy.

4.2. DRLPSO Algorithm

The proposed DRLPSO is described in Algorithm 1. In this algorithm, we first initialize the experience replay similar to the existing DRL methods, the number of iterations to train, the initial belief according to the environment and the parameters for Q-network and target network (Lines 1-2). For each iteration, we simulate the environment for training, initialize the action, history and assign the initial belief across the state set. Next, we perform $\mathbb{T}$ rounds of intervention. During each round of intervention, we predict the optimal action set with DPSO by randomly initializing n particles with different positions in the discrete action space until they converge to multiple optimal positions (actions) (line 7). DPSO finds the actions with the best reward based on the belief b_{t-1}. From the optimal action set of DPSO, we choose the action where the particle gives the maximum predicted Q-value (line 8). This action is executed in the simulated environment to obtain reward R and observation o_{t+1} and update b_t (lines 9-10). We then store the transition sequence in experience replay (line 12). After that, we randomly sample the transition sequences as a mini-batch and update parameter θ by Eqs. (15-18) (lines 13-16). Finally, we obtain the converged parameter $\hat{\theta}$ (line 18).

5. Experiment Results

In this section, we evaluate the proposed DRLPSO and compare it with the five existing methods for large sequential decision-making problems. We run the simulated student networks with mental state propagation to evaluate the performance of DRLPSO. We synthesize the problem instances since there is no publicly available data that studies the stress level of the people in a network. However, evaluations of algorithms on simulated networks are widely applied [25, 26, 27] as they serve as an important reference towards the real-world applications of the architecture. Further, we plan to do the experiments and carry out interventions for the university students to know about their stress levels, give them counselling and improve their performance.

Algorithm 1: DRLPSO

Result: Parameter θ

Initialize the experience replay D, # of iterations M;

Initialize Q-network and Target-Network with θ and θ^{-} respectively;

/ train the network shown in Block A in Figure 1 */*

for *iteration= 1 to* M **do**

  Simulate environment and b^0 ; *// Block C in Figure 1*

  Initialize the first action $a_0 = noaction$, $h_1 = \emptyset, o_1 = \emptyset$;

  for $t = 1...\mathbb{T}$ *rounds* **do**

    A_t = best action set from DPSO module where velocity and position updates using Eqs. (12) and (13) ; *// Block B in Figure 1*

    $a_t = \arg\max_{a \in A_t} Q(\langle b_{t-1}, H_t \rangle, a|\theta)$;

    Execute action a_t to obtain reward $R_t(.)$ and observation o_t;

    Update b_t according to o_t using Eq. (11);

    Store transition $\{\langle b_{t-1}, h_t \rangle, a_t, r_t, b_t\}$ in D ;

    Randomly sample a mini batch of transition sequences from D and index as j;

    $$y_j = \begin{cases} r_j, \text{ if } t = \mathbb{T} \\ \text{Eq.}(15), \text{ otherwise} \end{cases}$$

    Compute gradient using the loss function (Eq. (17));

    Update θ according to Eq. (18);

  end

end

Save model as $\hat{\theta}$;

5.1. Experiment Setup

We simulate a student network G by leveraging Erdos-Ŕenyi random network generation [28], where exactly $|E|$ edges are randomly constructed between each pair of nodes (students). Assuming that the students usually study/stay in groups of 3 or 4 which is also the size of a team for a regular group project, we set $|E| = 3N$ to let each node have at least 3 connections on average. Then, we assign $\hat{w}_{ij} \in W_0$ as randomized values between $[0, 1]$. After that, we set $\mu = 9$ and randomly assign the nodes with mental state values between $[0 - 9]$.

We compare the performance of proposed DRLPSO against the five baseline methods such as Degree Centrality (DC) [29]; HEAL [26]; DBQN [30]; DRQN [6] and ADRQN [7]. Since DRQN and ADRQN use the convolutional layers as input, we convert the observed states as 84x84 grey-scale images through Python

Pillow Module and train both networks accordingly. To mitigate the problem of handling large action space, we randomly sample the action space of the problem with 30 actions to train using DBQN, DRQN and ADRQN. To make a par comparison, we assign the LSTM output layer of DRLPSO to have 30 outputs. We assign the randomized initial estimate mental state and influence values for all the methods.

Our experimental settings are as follows: we set $(\mathbb{T}, K, \delta, \mu)$ as $(5, 1, 2, 9)$ for networks of 5 nodes, $(5, 5, 2, 9)$ for networks of 30 nodes and $(10, 5, 2, 9)$ for even larger networks. We implement all the aforementioned programs in Python and run all the experiments on a system of 3.2GHz 4-core Intel CPU and NVIDIA DGX-1 with 32 GB of RAM.

5.2. Evaluation

Parameter setting of DPSO

We set the parameter values of c_1 and c_2 with SwarmOps [31]. Since we define the LSTM output as 30 Q-values, we aim to map the size of the converged optimal action set to 30. Figure 3 illustrates the number of unique peaks captured by varying the number of particles in the DPSO to search for the best action set for a dynamic environment having a network of 30 students. Here, the minimum and average peak captures are shown after 10 runs. From the results, we can infer that DPSO with at least 90 particles has to be initialized.

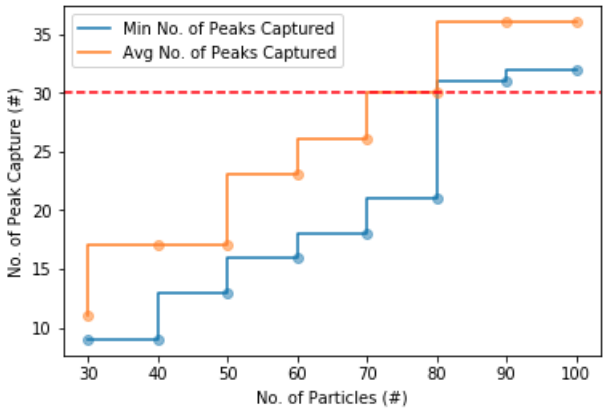

Figure 3: Peak convergence by varying the number of particles

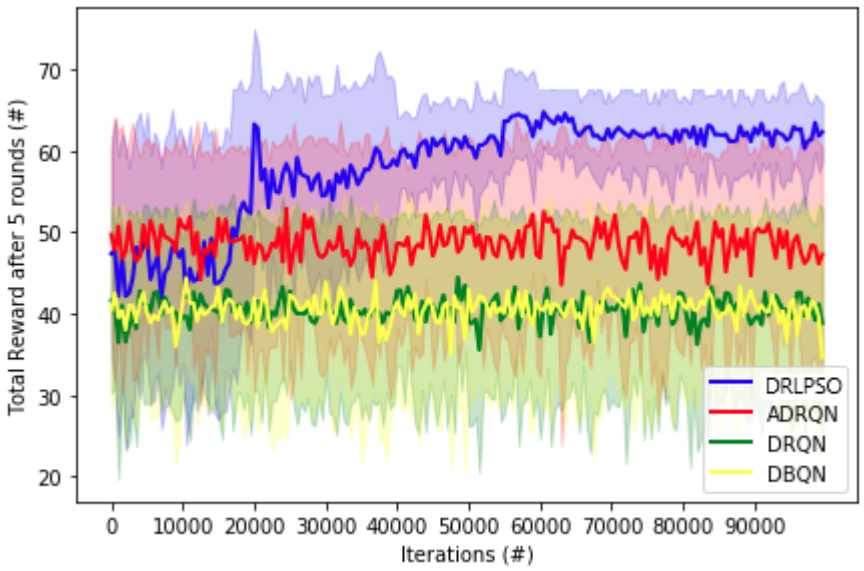

Figure 4: Performance of learning algorithms in 100k iterations

Training Rewards

Figure 4 shows the training curves of the deep Q-learning algorithms with 100,000 iterations. After every 500 iterations, we predict the actions for 5 rounds on 30 problem instances of the testing environment and record the total reward. From the figure, we observe that DRLPSO converges within the limited number of iterations and achieves much higher reward values in comparison with the other learning methods.

Performance Analysis

We train each algorithm to analyze the total reward after $\mathbb{T}$ rounds of intervention with 30 problem instances of the testing environment for each network size. Table 1 shows the reward comparison of different sizes of networks with the aforementioned experimental settings. We use $\sum_{t=1}^{\mathbb{T}} R_t(\mathrm{s}, a)$ as evaluation metric which is the total overall stress level reduced in the process of intervention. From the table, we observe that rewards of the previous dynamic influence maximization methods such as DC and HEAL are lower reward value than the proposed DRLPSO. This is because DC does not consider uncertainty and partial observation, while HEAL does not consider the uncertainty of initial mental state values. However, HEAL cannot scale up for networks larger than 30 nodes.

Similarly, when the comparison is made among 5 learning algorithms. Table 1 shows DQ-LSTM is only feasible for 5 students but it stops running for 30

Table 1: Reward Comparison

Nodes	DC ($\mu \pm \sigma$)	HEAL ($\mu \pm \sigma$)	DBQN ($\mu \pm \sigma$)
5	9.9 ± 1.668	13.43 ± 1.96	10.56 ± 1.977
30	45.03 ± 2.809	47.2 ± 3.59	31.76 ± 15.36
100	102.6 ± 9	-	155.4 ± 25.53
200	119.53 ± 18.584	-	128.9 ± 16.572
500	118.53 ± 20.91	-	133.93 ± 17.3

Nodes	DRQN ($\mu \pm \sigma$)	ADRQN ($\mu \pm \sigma$)	DQ-LSTM	DRLPSO ($\mu + \sigma$)
5	11.56 ± 1.716	11.53 ± 1.81	13.48 ± 1.743	$\mathbf{13.8} \pm 1.69$
30	38.33 ± 13.514	48.8 ± 11.74	-	$\mathbf{62.3} \pm 3.9$
100	162 ± 21.94	168.3 ± 37.47	-	$\mathbf{225} \pm 12.62$
200	138.33 ± 17.516	171.5 ± 24.81	-	$\mathbf{232.1} \pm 16.04$
500	143.13 ± 19.72	175.17 ± 18.27	-	$\mathbf{233.06} \pm 12.31$

students and above as it fails to handle larger state and action space i.e., state space of 10^{30} and action space of $\binom{30}{5} = 142506$. Our proposed DRLPSO outperforms DBQN, DRQN and ADRQN in the total reward with a high mean reward value and lower standard deviation over $\mathbb{T}$ rounds with 30 runs. This is because, in DBQN, DRQN and ADRQN, the action sets are randomly sampled to a set of 30 actions instead of a large action space unlike DRLPSO uses DPSO to optimize the action set before mapping to the Q-values from LSTM.

We also propose the Average Percentage of Reward Increase (APRI), to compare our proposed DRLPSO with baseline methods defined by,

$$APRI = \frac{\mu(R_{DRLPSO}) - \mu(R_{Algo})}{\mu(R_{DRLPSO)}} \times 100\% \tag{19}$$

where $\mu(.)$ is the mean reward value from the Monte Carlo simulation of the respective algorithms. We calculate APRI for the network size of 30 nodes where we observe that our proposed DRLPSO significantly outperforms DC, HEAL, DBQN, DRQN, ADRQN by 27.7%, 24.2%, 49%, 38.4% and 21.6% respectively. Thus, the overall stress level in the network will be best reduced after $\mathbb{T}$ rounds of intervention with the counsellors by selecting the students by DRLPSO method at each round.

Comparison with MLPRAP method

Here, we will compare with our previous method, MLPRAP: Multi-Level Partition algorithm with Reasoning and Abstracted Planning which uses POMDP

solver and multi-level partitioning of the original graph [4]. We only compare with the better technique proposed in the paper MLPRAP-C which hierarchically partitions the original student network in clusters until each partition has under l students which is the maximum limit of the students that can be solved by the aforementioned hardware configuration.

We have also compared using the realistic networks against MLPRAP method and observe better results. The first network is Zachary Karate Club dataset (Karate) with 34 nodes and 78 edges [32] which is the friendship data of the members of a university karate club. This will closely reflect the relationship between students in the network and the effectiveness of interventions. We assign $\hat{w}_{ij} \in W_0$ as randomized values from $[0, 1]$. The second dataset is Mobile-1 dataset (Mobile) which has 107 nodes and 513 edges [33]. It consists of the logs of calls and cell tower IDsx of users for ten months. We assign communication count between users i and j as $\hat{w}_{ij}$.

Table 2: Reward Comparison

Nodes	MLPRAP-B$(\mu + \sigma)$	MLPRAP-C $(\mu + \sigma)$	DRLPSO $(\mu + \sigma)$
100	133 ± 4.74	154.17 ± 6.06	$\mathbf{225} \pm 12.62$
200	-	150.23 ± 5.14	$\mathbf{232.1} \pm 16.04$
500	-	161.5 ± 5.17	$\mathbf{233.06} \pm 12.31$
1000	-	160.8 ± 6.88	$\mathbf{210.27} \pm 13.20$
1500	-	161.3 ± 5.43	$\mathbf{217.60} \pm 12.42$
Karate	80 ± 0.7	84.6 ± 1.22	$\mathbf{89.2} \pm 1.51$
Mobile	62.1 ± 1.21	62.7 ± 1.88	$\mathbf{66.4} \pm 1.83$

The comparison is as shown in Table 2. Although it takes longer to train to be able to start using to select the students for intention, DRLPSO has better results than MLPRAP-C and MLPRAP-B since there is no loss of information due to the partitioning with the DRL methods.

6. Discussion

The overview of evaluation with the actual study is as shown in Figure 5.

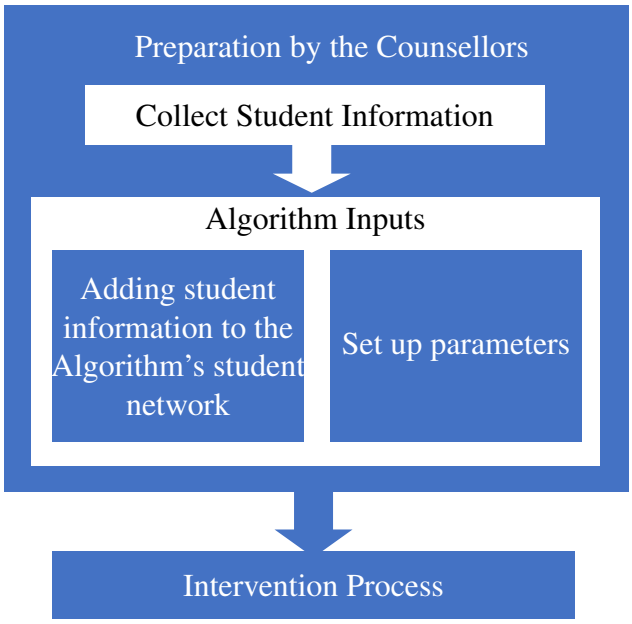

Figure 5: Overview of the implementation plan

We will collaborate with University Counselling Center (UCC) to conduct a real scenario with graduate students. Student data and mental state assessments are masked with a pseudonym and direct ID links will be destroyed after the counselling process. The participants are given a consent form and free to withdraw anytime. The detailed plan is as follows.

- We prepare the well-documented implementation plan of the intervention process and design the stress assessment questionnaire according to the questionnaires from the established literature such as (Perceived Stress Scale, 1994).
- We then submit the documents to the review process from Institutional Review Board for Research Involving Human Subjects to ensure that the research activity follows applicable legislation of Singapore.
- After the approval from the board, we perform the necessary data collection to estimate the students' mental state and influence values. We first get the students' consents with a consent form where the consequence of the study/ interventions is explained in detail, understandable language usage. Afterwards, we collect the following information from the students:

1. College; 2. Supervisor, Co-supervisor; 3. Course Enrollment; 4. Physical Location (Lab/Office location, Home/ Dormitory Address); 5. Participation in student clubs; 6. Publication List; 7. Academic Performance: Grades; 8. Performance: Evaluation by Supervisor.

- Next, we set up the environment according to collected student information.

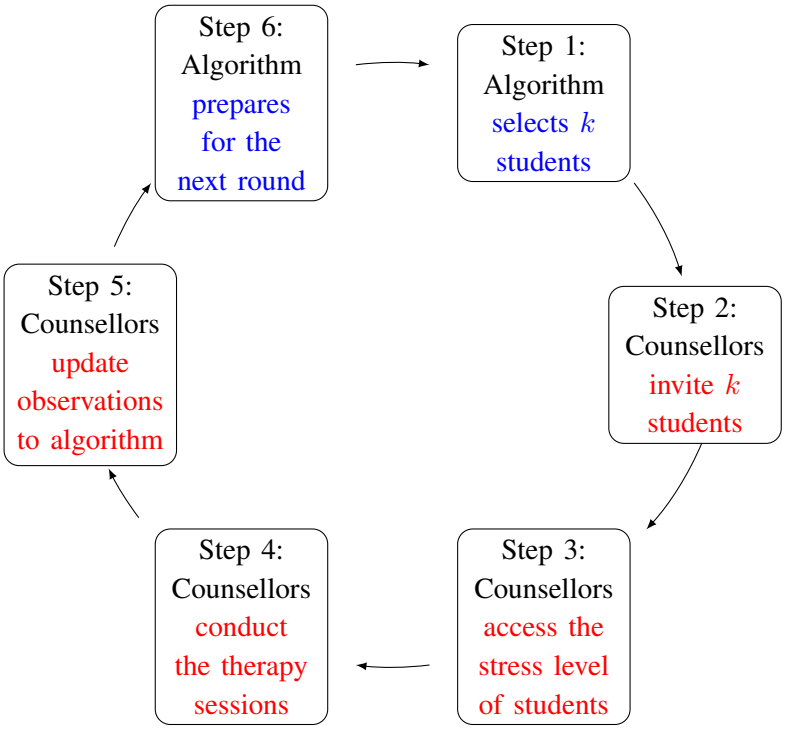

Figure 6: Intervention process

After the preparation step is done, we begin the intervention process. UCC initiates the intervention process in coordination with the algorithm. The intervention process is illustrated in Figure 6. At each round, UCC invites the candidate students who are selected by the algorithm. During the intervention round, the counsellors conduct a therapy session with the selected students, evaluate/assess their mental states based on a set of designed questionnaires and learn their social-circles. When the intervention is over the counsellors update the newly collected information to the algorithm. The algorithm then updates the network, estimates the stress levels of other students by the reasoning module and selects students for the subsequent rounds.

7. Conclusion

In this paper, we propose a novel architecture DRLPSO to handle the partially observable dynamic environments with large action space. The DRLPSO algorithm uses the Deep Q-learning integrated with LSTM (DQ-LSTM) and DPSO to optimally select the predicted action, obtain Q-values and train the network. We use belief and history of action and observation as input to DQ-LSTM. Results have shown that in a dynamic environment with a large action space, the proposed DRLPSO achieves a higher total reward compared to the DRL approaches by an average of 32% compared to the baseline. Moreover, comparing with POMDP solution, while DRLPSO takes time for training due to it being DRL solution, it outperforms in terms of effectiveness. In the future, we will evaluate the algorithm with the real-world setting and further explore on online reinforcement learning methods so that the training time for DRLPSO can be reduced.

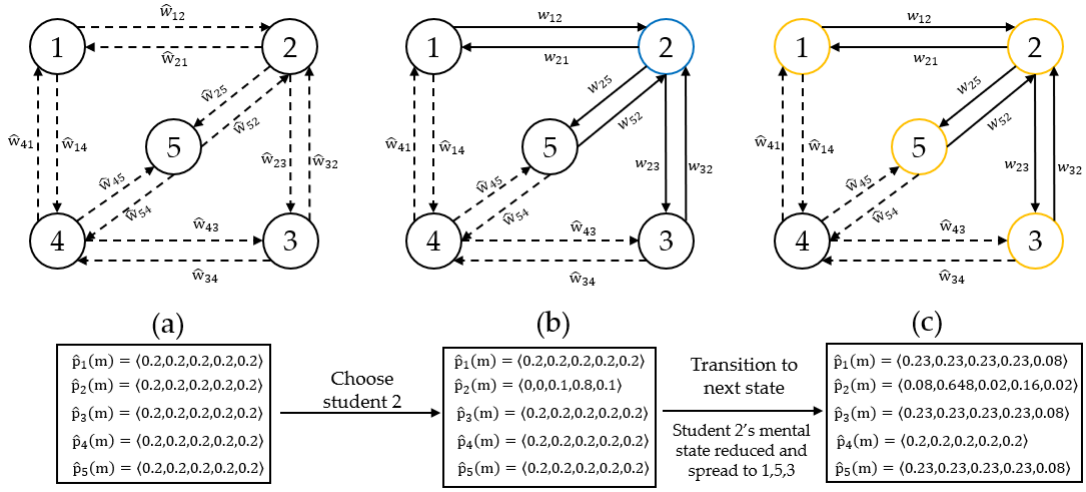

Figure 1: Illustrative example of intervention round 1

Appendix

A.1 Illustrative Example of Interventions

Figure 1 shows the changes in the uncertain dynamic environment with a network of 5 students during the intervention process with (a) the initial belief state, (b) observation after executing the first action and (c) the transition to the next state. We use $\mu = 4$ in this example.

Initially, the belief state is assigned with the estimated mental state and influence values with $\hat{P}$ and $\hat{W}$. Hence, the estimated mental state of each student is assigned with equal probability values $\hat{p}_i(m)$ and estimated influence values are represented by the dotted lines. After executing the first action by selecting student 2, the counsellors evaluate student 2's mental state as 3, and the influence between student 2 and his/her neighbours are observed which are represented by the solid lines. Considering the observation probability, the updated mental state estimates in the belief are as shown in Figure 1(b). Student 2's mental state is reduced due to counselling and the effect is spread to the associated neighbours (students 1, 5 and 3). Considering the transition probability and the emotion propagation model, the belief state values for the next state is as shown in Figure 1(c). The same procedure repeats in further rounds.

A.2 Numerical Example

We present a numerical example of the dynamic environment with 3 students having $\mu = 3$. The actual mental state and influence values are,

$$\text{actual\_mental\_state} = [3, 3, 1]$$

$$\text{actual\_influence} = \begin{pmatrix} 0 & 0.5 & 0.7 \\ 0.3 & 0 & 0.5 \\ 0.7 & 0.7 & 0 \end{pmatrix}$$

and we set $K = 1$ and hence the actions $a_0 = \langle 1, 0, 0 \rangle$ is choosing student 1, $a_1 = \langle 0, 1, 0 \rangle$ is choosing student 2 and $a_2 = \langle 0, 0, 1 \rangle$ is choosing student 3 respectively.

Initially, we assign equal probability for all initial mental state values to be 0.25 and the influence to be 0.5. Thus, the initial belief state is:

$$\hat{P}_0 = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$
$$\hat{W}_0 = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{pmatrix}$$

Hence, b_0 is assigned as:

$$b_0 = [0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, \\ 0.25, 0.25, 0.25, 0.25, 0, 0.5, 0.5, 0.5, 0, 0.5, 0.5, 0.5, 0]$$

and initial history is $\emptyset$.
The deep Q-learning architecture with 21 inputs and 3 output is initialized as 21×21 matrix for the first layer, 21×21 matrix for the second layer and 21×3 matrix for the third layer and we use ReLU as activation function between each layer. Using these values, the first forward pass gives the predicted Q-values as:

$$Q(\langle b_{t-1}, H_t \rangle, a_0) = 0.13611966$$
$$Q(\langle b_{t-1}, H_t \rangle, a_1) = 0$$
$$Q(\langle b_{t-1}, H_t \rangle, a_2) = 2.30237779$$

At first state, the probabilities for mental states are equal and thus, DPSO

particles will converge to all 3 actions based on b_0 and a_2 chosen (student 3) as the first action since it is with the highest Q-value.

After choosing student 3, we obtain reward as 1 and the new belief updated from observation as:

$$\begin{aligned} b_1 = [&0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, \\ &1, 0, 0, 0, 0, 0.5, 0.7, 0.5, 0, 0.5, 0.7, 0.7, 0] \\ h_2 = \{&\langle [0, 0, 1], 1, 0.7, 0.7, 0.7, 0.5 \rangle\} \end{aligned}$$

Hence, the actual Q-value according to Eq. (15) using $\gamma = 0.5$ is,

$$R(s, a) + \gamma \cdot \max_a Q(\langle b_t, H_{t+1} \rangle, a) = 2.39935325$$

Using the values, we can calculate the loss value,

$$L = 0.054324307568820125$$

Then, through backpropagation, we use the gradient of the loss function and use gradient descent optimizer to update θ to minimize the loss.

In iteration 2, according to belief b_1, the particles of DPSO converge at a_0 as the student 3 mental state value is 0 and the estimated influence of student 1 and neighbours are greater than student 2's values. After action a_0, the new belief and history are:

$$\begin{aligned} b_2 = [&0, 1, 0, 0, 0.25, 0.25, 0.25, 0.25, \\ &1, 0, 0, 0, 0, 0.5, 0.7, 0.3, 0, 0.5, 0.7, 0.7, 0] \\ h_3 = \{&\langle [0, 0, 1], 1, 0.7, 0.7, 0.7, 0.5 \rangle, \langle \\ &[1, 0, 0], [3, 0.5, 0.5, 0.7, 0.3] \rangle\} \end{aligned}$$

Hence, the actual Q-value according to Eq. (15) is,

$$R(s, a) + \gamma \cdot \max_a Q(\langle b_t, H_{t+1} \rangle, a) = 3.797936785$$

We use gradient descent optimizer update θ to minimize the loss until it converges to $L < 10^{-3}$.

References

[1] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv preprint arXiv:1509.02971 (2015).

[2] P. Xu, I. Karamouzas, Particle-based adaptive discretization for continuous control using deep reinforcement learning, arXiv preprint arXiv:2003.06959 (2020).

[3] M. G. Bellemare, Y. Naddaf, J. Veness, M. Bowling, The arcade learning environment: An evaluation platform for general agents, Journal of Artificial Intelligence Research 47 (2013) 253–279.

[4] A. P. P. Aung, X. Wang, B. An, X. Li, We mind your well-being: Preventing depression in uncertain social networks by sequential interventions, in: Proceedings of the 30th International Conference on Automated Planning and Scheduling.

[5] M. Hausknecht, P. Stone, Deep recurrent Q-learning for partially observable MDPs, in: 2015 AAAI Fall Symposium Series.

[6] J. N. Foerster, Y. M. Assael, N. de Freitas, S. Whiteson, Learning to communicate to solve riddles with deep distributed recurrent Q-networks, arXiv preprint arXiv:1602.02672 (2016).

[7] P. Zhu, X. Li, P. Poupart, G. Miao, On improving deep reinforcement learning for POMDPs, arXiv preprint arXiv:1704.07978 (2017).

[8] F. Van den Bergh, A. P. Engelbrecht, A convergence proof for the particle swarm optimiser, Fundamenta Informaticae 105 (2010) 341–374.

[9] M. R. Bonyadi, Z. Michalewicz, A locally convergent rotationally invariant particle swarm optimization algorithm, Swarm Intelligence 8 (2014) 159–198.

[10] J. Senthilnath, S. Omkar, V. Mani, T. Karthikeyan, Multiobjective discrete particle swarm optimization for multisensor image alignment, IEEE Geoscience and Remote Sensing Letters 10 (2013) 1095–1099.

[11] K. F. Helmers, D. Danoff, Y. Steinert, M. Leyton, S. N. Young, Stress and depressed mood in medical students, law students, and graduate students at mcgill university, Academic Medicine 72 (1997) 708–714.

[12] E. R. Blackmore, S. A. Stansfeld, I. Weller, S. Munce, B. M. Zagorski, D. E. Stewart, Major depressive episodes and work stress: results from a national population survey, American Journal of Public Health 97 (2007) 2088–2093.

[13] P. J. Lucassen, V. M. Heine, M. B. Muller, E. M. van der Beek, V. M. Wiegant, E. Ron De Kloet, M. Joels, E. Fuchs, D. F. Swaab, B. Czeh, Stress, depression and hippocampal apoptosis, CNS & Neurological Disorders-Drug Targets (Formerly Current Drug Targets-CNS & Neurological Disorders) 5 (2006) 531–546.

[14] S. Khan, R. A. Khan, Chronic stress leads to anxiety and depression, Ann Psychiatry Ment Health 5 (2017) 1091.

[15] J. H. Fowler, N. A. Christakis, Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study, Bmj 337 (2008) a2338.

[16] R. W. Eyre, T. House, E. M. Hill, F. E. Griffiths, Spreading of components of mood in adolescent social networks, Royal Society Open Science 4 (2017) 170336.

[17] S. Goel, D. J. Watts, D. G. Goldstein, The Structure of Online Diffusion Networks, in: Proceedings of the 13th ACM Conference on Electronic Commerce, pp. 623–638.

[18] A. E. Rafferty, M. A. Griffin, Perceptions of organizational change: A stress and coping perspective., Journal of Applied Psychology 91 (2006) 1154.

[19] UCLA, The STAND program, `https://depression.semel.ucla.edu/stand_home`, 2018. Accessed: 2020-03-30.

[20] WHO, The ICD-10 classification of mental and behavioural disorders: diagnostic criteria for research.

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

[22] M. L. Puterman, Markov Decision Processes: Discrete stochastic dynamic programming.

[23] J. Kennedy, Particle swarm optimization, Encyclopedia of machine learning (2010) 760–766.

[24] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing Atari with deep reinforcement learning, arXiv preprint arXiv:1312.5602 (2013).

[25] A. Yadav, L. S. Marcolino, E. Rice, R. Petering, H. Winetrobe, H. Rhoades, M. Tambe, H. Carmichael, Preventing HIV spread in homeless populations using PSINET., in: AAAI, pp. 4006–4011.

[26] A. Yadav, H. Chan, A. Xin Jiang, H. Xu, E. Rice, M. Tambe, Using social networks to aid homeless shelters: Dynamic influence maximization under uncertainty, in: AAMAS, pp. 740–748.

[27] B. Wilder, A. Yadav, N. Immorlica, E. Rice, M. Tambe, Uncharted but not uninfluenced: influence maximization with an uncertain network, in: AAMAS, pp. 1305–1313.

[28] P. Erdos, On random graphs, Publicationes Mathematicae 6 (1959) 290–297.

[29] U. Kang, S. Papadimitriou, J. Sun, H. Tong, Centralities in large networks: Algorithms and observations, in: Proceedings of the 2011 SIAM International Conference on Data Mining, pp. 119–130.

[30] M. Egorov, Deep Reinforcement Learning with POMDPs, Technical Report, Technical Report, Stanford University, 2015.

[31] M. E. H. Pedersen, Good parameters for particle swarm optimization, Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001 (2010) 1551–3203.

[32] W. W. Zachary, An Information Flow Model for Conflict and Fission in Small Groups, Journal of Anthropological Research 33 (1977) 452–473.

[33] J. Tang, T. Lou, J. Kleinberg, Inferring Social Ties Across Heterogenous Networks, in: Proceedings of the 5th ACM International Conference on Web Search and Data Mining, pp. 743–752.