# Optimizing network structure for preventative health

Bryan Wilder, Han Ching Ou, Kayla de la Haye, Milind Tambe
Center for Artificial Intelligence in Society
University of Southern California
{bwilder,hanchino,delahaye,tambe}@usc.edu

## ABSTRACT

Diseases such as heart disease, stroke, or diabetes affect hundreds of millions of people. Such conditions are strongly impacted by obesity, and establishing healthy lifestyle behaviors is a critical public health challenge with many applications. Changing health behaviors is inherently a multiagent problem since people's behavior is strongly influenced by those around them. Hence, practitioners often attempt to modify the social network of a community by adding or removing edges in ways that will lead to desirable behavior change. To our knowledge, no previous work considers the algorithmic problem of finding the optimal set of edges to add and remove. We propose the RECONNECT algorithm, which efficiently finds high-quality solutions for a range of different network intervention problems. We evaluate RECONNECT in a highly realistic simulated environment based on the Antelope Valley region in California which draws on demographic, social, and health-related data. We find the RECONNECT outperforms an array of baseline policies, in some cases yielding a 150% improvement over the best alternative.

## KEYWORDS

Preventative health; social networks; network design

## 1 INTRODUCTION

Hundreds of millions of people suffer from lifestyle-related conditions such as heart disease, stroke, or diabetes [1]. Obesity is an important factor in many of these conditions [20], which often starts in childhood [37] and is difficult to reverse once established [39]. Indeed, the World Health Organization has recently stressed the importance of establishing healthy lifestyle behaviors in the first two years of life [33] to prevent the onset of obesity. Eating and physical activity, health behaviors linked to obesity risk, are rooted in social networks [9, 10, 40, 45]: people infer what is normative from those around them [9, 18], they learn behaviors from others [2], and they unconsciously respond to social cues [36]. This is a multiagent phenomena since interactions and learning between members of the population (agents) drive the collective dynamics.

Hence, new interventions to combat obesity attempt to change the social environment around mothers and their young children by providing them with healthy role models or peer groups [37] (Figure 1). These interventions change the structure of the social network by adding new connections or removing existing ones. Deciding how to best modify a network to promote healthier behaviors among network members is a complex optimization problem. However, there is currently no algorithmic guidance available to practitioners.

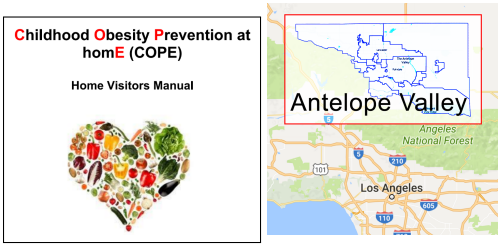


**Figure 1: Left: example of a social network anti-obesity intervention [37]. Right: the Antelope Valley region. [42].**

To fill this gap, we provide a comprehensive algorithmic approach to targeting social network interventions for preventative health. We model a service provider who uses one of two intervention designs to alter the network [44]. First, a peer mentoring program which creates edges between specific pairs of agents (a *dyad* intervention). This models pairing unhealthy agents with a healthy buddy. Second, a peer support group, where several agents are placed in a group and form edges to the other group members. The number of interventions carried out is always limited in practice since altering edges requires personal interaction (e.g., home visits) with a health worker to establish and maintain ties [37]. The question is which specific edges should be created to maximize the health of a set of targeted agents after some period of time.

This domain presents a number of challenges. *First*, normal properties in network intervention (e.g., submodularity in influence maximization [21]) do not apply: our objective function does not exhibit diminishing returns and is highly nonlinear. *Second*, interventions are not guaranteed to have beneficial effects at the community level. Indeed, a network modification which inadvertently increased the prominence of an unhealthy agent could negatively impact the health of others. *Third*, interventions can spark secondary changes in the network. For instance, people are typically only able to maintain a limited number of social ties [35]. So, if an intervention leads an agent to add a new relationship, he or she may drop an existing one. *Fourth*, we require an extremely efficient algorithm since even evaluating the full objective function for each possible edge to add is computationally costly. For instance, we show that the greedy algorithm, which sets the standard for computational efficiency in other network problems [16, 41], does not scale beyond 100-node networks in ours. Realistic interventions will need to accommodate networks with thousands of nodes.

We design the RECONNECT algorithm (short for Reconnecting social networks for preventative health) to find high-quality interventions strategies while accounting for the above challenges. Our

*Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. Andre, S. Koenig (eds.), July 2018, Stockholm, Sweden*

https://doi.org/doi

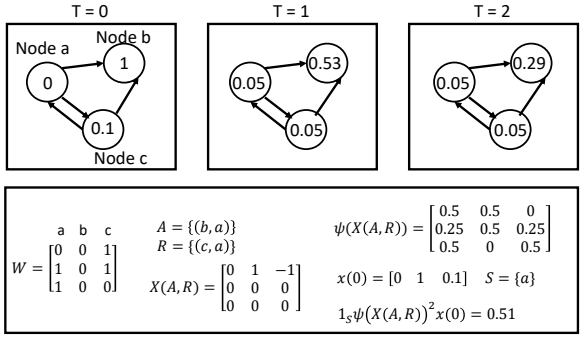


**Figure 2: Top: example of multiagent dynamics with $\lambda = 0.5$. Bottom: numerical example of adding and removing edges with corresponding perturbation matrix and objective value.**

primary motivation is interventions for obesity. However, similar challenges appear in many domains where practitioners attempt to harness or alter social influence processes as part of behavior change interventions, for example preventing smoking [46] or HIV [30]. While a great deal of work in public health considers such issues, to our knowledge no previous work addresses the *algorithmic* problem of optimally modifying network structure for preventative health. Existing algorithms (e.g., for search engine optimization [32] or wildlife corridor design [27, 48]) fail to address one or more of our domain's challenges. We survey previous work in Section 4.

In short, we make four main contributions. *First*, we introduce a new multiagent systems problem, preventative health network intervention (PHNI). PHNI encompasses a range of realistic network intervention designs. *Second*, we present the RECONNECT algorithm, which efficiently computes near-optimal targeting strategies for all of these settings in a unified manner. *Third*, we provide theoretical analysis of each intervention design which exploits the combinatorial structure in our domain to enable scale-up. *Fourth*, we compare our algorithm to an array of baselines in a highly realistic testbed which draws on data from the Antelope Valley community in California (Figure 1). We create a social simulation informed by demographic data (e.g., age, ethnicity, gender), spatial information on population density, and status-quo obesity rates. We find that RECONNECT outperforms all baselines, at times exceeding the best alternative by over 150%.

## 2 PROBLEM DESCRIPTION

**Model and objective:** The input to the PHNI problem is a directed graph $G = (V, E)$ with $|V| = n$. $G$ represents the interactions between agents in the community; a directed edge $(u, v)$ indicates that $u$ influences $v$. Each node (agent) $v \in V$ has a continuous state $x_v \in [0, 1]$, where higher states are better. $x_v$ represents the level of healthiness of agent $v$. For instance, 1 might correspond to being nonoverweight, while 0 is a weight categorized as overweight or obese.

Since obesity-related behaviors are subject to social influence [9, 10], we model a social influence processes where agents update their behavior (and thus weight status) in response to the behavior of those around them. The process takes place over discrete time steps $t = 1...T$ where $T$ is the time horizon. At each step, each node updates its state to be a convex combination of its previous state and the average of its neighbors. Formally, we have a parameter $\lambda \in [0, 1]$. Let $\delta^{in}(v)$ denote node $v$'s in-neighbors; likewise $\delta^{out}(v)$ denotes $v$'s out-neighbors. $d(v) = |\delta^{in}(v)|$ is the in-degree of $v$. Each node updates as

$$x_v(t+1) = (1-\lambda)x_v(t) + \lambda \frac{1}{d(v)} \sum_{u \in \delta^{in}(v)} x_u(t)$$

This averaging behavior can be seen as an instance of DeGroot learning, an extensively used model of social learning [11, 19, 28] which has been validated in large-scale experiments [7]. We include the term $(1-\lambda)x(t)$ to represent that healthiness (e.g., weight) does not change overnight; instead agents take a step towards the behavior that is considered normative based on their neighbors. Figure 2 provides an illustration. We assume that there is a set of target nodes $S \subseteq V$. The objective is to maximize $\sum_{v \in S} x_v(T)$, or the combined state of the target nodes after $T$ steps.

**Actions:** A number of different network intervention strategies have been proposed in the preventative health literature [43, 44]. Accordingly, we define a flexible modeling framework which allows us to handle these different settings in a unified algorithm. Note that any given intervention only falls into one setting; it is only the underlying algorithmic approach which is shared. We consider two basic interventions, of which different variants are possible. First, in an *dyad intervention*, we add a set of specific edges to the network. Any set of edges (which are not already present) are admissible. This models a peer mentor program, where nodes are connected with "healthy buddies", or healthier agents who they can learn from. Second, in a *group intervention* action, we place a set of $M$ nodes in a peer group. Each node forms links to each other node in the group. This models peer support groups, in which small groups of agents meet periodically.

When edges are added to the network, a corresponding number of edges must be removed. Edge removals are necessary because agents can only maintain a limited number of social ties [35]. Accordingly, whenever we add an edge $(u, v)$ (denoting that $v$ is influenced by $u$), $v$ must remove an edge $(w, v)$ for some other $w$. $u$ does not have to remove an edge because ties are directed, so the relationship is not necessarily reciprocated (though if edge $(v, u)$ were added, then $u$ would also have to remove an edge). There are two cases for edge removals. First, they may be controlled by the algorithm (e.g., a service provider recommends that an agent decrease contact with an unhealthy acquaintance). We call this the *controlled removal* case. Second, they may occur at random beyond the algorithm's control. I.e., if we add an edge to node $v$, then $v$ will drop an edge from some neighbor $u$ chosen at random. We call this the *random removal* case.

In total, our framework encompasses a range of possible intervention designs by taking an addition operation from $\{dyad, group\}$ and a removal operation from $\{controlled, random\}$. This has two benefits. First, we obtain unified algorithm which can handle many different preventative health settings. Second, this flexibility enables practitioners to compare the impact of different intervention designs and select the one which is most effective for their community. We again note that these intervention settings are separate (e.g., removals are either controlled or random but not both), but we provide a common algorithmic approach encompassing all of them.

## 3 CONTROLLED REMOVALS

We first present our formulation and algorithms for *controlled removals*. Then, Section 6 extends our approach to handle random removals.

**Constraints:** In the dyad setting, we consider two principle constraints on the actions available to the algorithm. First, there is a global budget constraint that at most $K_D$ edges can be added. The budget constraints models the fact that service providers have only limited resources (time, staff) to perform interventions. Second, there is a limit on the number of edges which may be added to each node. This models the fact that we cannot entirely change an agent's social circle: at most a limited number of new relationships can be added. Specifically, for each node, at most $\kappa_{in}$ new in-edges and $\kappa_{out}$ new out-edges can be added. In the group setting, the algorithm is allowed to form at most $K_G$ groups, each of which consists of $M$ nodes. The limit $K_G$ again reflects the service provider's resource limitations. We do not consider separate constraints on $\kappa_{in}$ and $\kappa_{out}$ because this would be redundant: a practitioner would choose the group size so that forming $M-1$ new edges (to the other group members) and removing corresponding edges from outside the group is realistic.

Let $\chi_v^{in}(A)$ denote the number of in-edges added to node $v$ in the set $A$. Similarly, $\chi_v^{out}(A)$ denotes the number of out-edges added to $v$ (because an in-edge was added to another agent). In either the dyad or group setting, we denote the feasible set of edges to add as $C_A$. E.g., in the dyad setting $C_A = \{A \subseteq \overline{E} : |A| \leq K_D, \chi_v^{in}(A) \leq \kappa_{in}$ and $\chi_v^{out}(A) \leq \kappa_{out}\ \forall v \in V\}$. Here, $A$ is a set of edges to add. In the group setting, $C_A$ is all sets of edges which are induced by a choice of $K_G$ groups of $M$ nodes each. Given a set of edges $A$, let $C_R(A)$ be the feasible set of removals, where each element is a valid set of edges to remove. Let $rem_v(R)$ be the number of $v$'s in-edges removed by $R$. We have $C_R(A) = \{R \subseteq E : rem_v(R) = \chi_v^{in}(A)\ \forall v \in V\}$, which requires that removals and additions balance at each node. The algorithm may select any $A \in C_A$ and $R \in C_R(A)$.

**Optimization problem:** Let $f(A,R) = \sum_{v \in S} x_v(T)$ denote the total state of target agents after $T$ time steps when the edges in $A$ are added to the network and the edges in $R$ are removed from the network. We aim to solve the following problem:

$$\max_{A \in C_A, R \in C_R(A)} f(A,R) \tag{1}$$

Note that the dyad/group distinction is captured by the feasible set $C_A$. We also give an alternate formulation of the function $f(A,R)$ which is helpful to formulate a continuous relaxation of our problem. Our objective is to formulate an optimization problem over a continuous set, where integral points in the feasible set correspond to the valid solutions of our problem. We now introduce the mapping from the discrete to continuous domains; see Figure 2 for an example.

Let $W$ be the transposed adjacency matrix of $G$ and $W_{norm}$ be the corresponding row-stochastic matrix ($W$ normalized by the sum of entries along each row so that every row sums to 1). Note that without any perturbation, the final state vector after $T$ timesteps is

$$x(T) = (\lambda W_{norm} + (1-\lambda)I)^T x(0)$$

where $I$ is the identity matrix. We can now identify each $(A,R)$ with a perturbation matrix $X(A,R)$ which is added to the original adjacency matrix. For example, suppose that we add a set of edges $A$ and remove a set of edges $R$. The corresponding perturbation matrix $X(A,R)$ has $X_{v,u} = 1\ \forall (u,v) \in A$ and $X_{v,u} = -1\ \forall (u,v) \in R$. Note that we set $X_{v,u}$ instead of $X_{u,v}$ because we are working with the transposed matrix $W$. The transposed adjacency matrix after the modifications is $W + X$. Let $rowsum(X)$ be a vector where entry $i$ is the sum of the $i$th row of $X$. Let $\psi$ be a function mapping a perturbation matrix $X$ to the final row-stochastic transition matrix:

$$\psi(X) = \lambda diag(1/rowsum(X+W))(X+W) + (1-\lambda)I$$

where $diag(v)$ is the matrix with the vector $v$ on the diagonal. That is, $\psi$ first row-normalizes $X + W$ so that every row sums to 1 and then takes the convex combination with the identity matrix $I$ (representing the weight that agents place on their own previous behavior). Let $1_S$ be the indicator vector which has a 1 in the entries corresponding to elements of $S$ and 0 elsewhere. We can now write $f(A,R) = 1_S^\top \psi(X(A,R))^T x(0)$ (which gives the total state of target agents after $T$ steps) and maximize over the perturbation matrix $X$.

In this formulation, we obtain a natural relaxation of the problem by allowing $X$ to take continuous values. Specifically, let $\mathcal{P} = conv(\{X(A,R) : A \in C_A, R \in C_R(A)\})$ where $conv$ denotes the convex hull. $\mathcal{P}$ is the convex hull of feasible solutions. For any $X \in \mathcal{P}$, we define the continuous objective $g(X) = 1_S^\top \psi(X)^T x(0)$. The continuous extension of our problem is $\max_{X \in \mathcal{P}} g(X)$.

## 4 PREVIOUS WORK

Previous work has addressed a range of network design problems targeting different application areas. One body of work concerns conservation planning, where the objective is to purchase nodes or edges in a graph so that animals are able to move between habitat locations [27, 38, 48]. Another area is preventing disease spread or other contagion by removing edges from a graph [23, 25]. None of this work considers the challenge of simultaneously adding and removing edges, which is a crucial feature of preventative health domains that RECONNECT accounts for. Further, the algorithmic techniques used in this previous work do not translate to our domain. E.g., conservation planning work typically relies on mixed integer linear programming. Their objective is linearizable because the ultimate objective is to encode whether a node in a graph is reachable. By contrast, our objective is highly nonlinear because all of the decision variables are raised to the power $T$, so mixed integer linear programming does not apply. Greedy algorithms (as in [23]) also do not apply due to scalability issues (as we detail later).

Khalil et al. [22] study edge addition and removal in the linear threshold model. However, they study each task in isolation, where we simultaneously add and remove edges in the same intervention. Further, their objective is to maximize (or minimize) the spread of a contagion starting from a random source node. By contrast, we wish to shape agents' behavior by amplifying the influence of particular nodes in the network, not increase the network's overall susceptibility to influence. Klein et al. [24] consider edge changes but do not optimize the global impact on the network. A parallel line of work concerns influence maximization [8, 21, 47, 49], where the task is to select a set of seed nodes who will effectively diffuse information throughout the network. This work is quite different from ours, both because we modify the *structure* of the network, and because the techniques used for influence maximization (based

**Algorithm 1** RECONNECT

1: //Run $L$ iterations of Frank-Wolfe on continuous extension
2: $X^0 \leftarrow 0$
3: **for** $\ell = 1...L$ **do**
4: $\quad \nabla^\ell \leftarrow \text{GRADIENTORACLE}(X^{\ell-1})$
5: $\quad Y^\ell \leftarrow \text{LINEARORACLE}(\nabla^\ell)$
6: $\quad X^\ell \leftarrow (1-\gamma_r)X^{\ell-1} + \gamma_\ell Y^\ell$
7: //Select best integral point
8: $X^* \leftarrow \arg\max_{\ell=1\ldots L} g(X^\ell)$
9: //Return corresponding additions and removals
10: $A \leftarrow \{(u,v) : X^*_{v,u} = 1\}$
11: In controlled setting:
12: $\quad R \leftarrow \{(u,v) : X^*_{v,u} = -1\}$
13: $\quad$ **return** $(A, R)$
14: In random setting:
15: $\quad$ **return** $A$

on submodularity) do not apply to our domain. Lastly, work in multiagent systems has more broadly considered the modeling of social influence (in health and other domains) [5, 15, 17, 50].

# 5 ALGORITHM

Our domain is algorithmically challenging because typical paradigms such as submodular optimization do not apply:

PROPOSITION 5.1. *Problem 1 is not submodular in A or R.*

A proof can be found in the supplement[1]. Our algorithmic approach is to optimize the continuous relaxation $g(X)$ and then round the solution to an integral point. This route is still algorithmically difficult ($g$ is easily shown to be nonconcave). However, there are two motivations for using the continuous approach in our domain. First, gradient-based techniques are guaranteed to converge to a *local* optimum of the continuous problem (even if finding a global optimum is computationally intractable). Second, our chosen gradient-based method can be made highly efficient by exploiting the structure present in our domain. This allows our algorithm to easily optimize over realistic networks with thousands of nodes, where purely combinatorial algorithms such as greedy optimization fail to scale up. While this procedure does not come with approximation guarantees (since the problem does not possess theoretically convenient properties like submodularity or convexity), our results show it to be empirically successful at rapidly finding high-quality solutions.

## 5.1 Approach

We now describe our approach to solving the continuous optimization problem. We use a Frank-Wolfe style algorithm [14]. Frank-Wolfe is a gradient-based optimization procedure originally developed for concave optimization. However, it is also known to converge to a local optimum for nonconcave problems [26]. We will see that the Frank-Wolfe approach exploits the combinatorial structure of our domain and delivers good practical results.

Algorithm 1 presents pseudocode for the Frank-Wolfe algorithm. The algorithm runs for some number $L$ steps, generating a series of feasible points $X^0 \ldots X^L \in \mathcal{P}$ (where $\mathcal{P}$ is the feasible set for any of our settings). $X^0$ can be an arbitrary feasible point (e.g., the all zeros matrix representing no perturbations). Each iteration $\ell$ alternates between two steps. First, we compute the gradient $\nabla^\ell$ of the objective at the current point $X^\ell$ (Line 4). Second, we find the point which optimizes the gradient over the feasible set by solving the linear optimization problem $Y^\ell = \arg\max_{Y \in \mathcal{P}} \langle Y, \nabla^\ell \rangle$ where $\langle ., . \rangle$ denotes the inner product (Line 5). Then, we set $X^{\ell+1} = (1-\gamma_\ell)X^\ell + \gamma_\ell Y^\ell$, where $\gamma_\ell$ is a step size denoting how far the algorithm moves towards $Y^\ell$. Since $\mathcal{P}$ is convex and $X^\ell$ is a convex combination of $X^{\ell-1}$ and $Y^\ell$, we are guaranteed that $X^\ell$ remains in $\mathcal{P}$

Frank-Wolfe is a general optimization methodology, not an out of the box approach: two oracles are required to instantiate the algorithm in our domain. First, a gradient oracle which supplies $\nabla g(X)$ for any given $X$. This is relatively straightforward for controlled removals (though additional difficulties arise with random removals in Section 6). Second, a linear optimization oracle which solves the problem $\arg\max_{Y \in \mathcal{P}} \langle Y, \nabla g(x) \rangle$. Developing fast algorithms for linear optimization is the crucial step enabling scale-up.

Once we have obtained the final point $X^L$, we need a way of rounding the fractional perturbation matrix to a discrete set of edges to modify, that is, some $A \in C_A$ (and $R \in C_R(A)$ in the controlled removal setting). Our approach to the continuous problem is well-suited to such rounding because at each step, its current solution is explicitly represented as a convex combination of vertices of $\mathcal{P}$. Each vertex represents some feasible integral strategy $(A, R)$. Hence, we simply examine each of the $(A, R)$ which form the continuous solution and take the one with highest objective value (lines 8-11). RECONNECT takes time $O(L(T_1 + T_2))$ where $T_1$ is the time to compute the gradient and $T_2$ is the time to perform linear optimization. Since these depend on the setting, we give values for them below.

## 5.2 Gradient evaluation

We now show how to efficiently calculate gradients of the objective (GRADIENTORACLE in Algorithm 1). Note that $g$ depends on $X$ through the function $\psi$, which row-normalizes its input to produce a valid transition matrix. Formally, for any $i \neq j$, $\psi(X)_{ij} = \lambda \frac{W_{ij} + X_{ij}}{d(i)}$ (the term for $(1-\lambda)I$ is constant and does not contribute to the gradient). We ignore the case $i = j$ because it is impossible to add self-loops. Since $d(i)$ is constant in all of the interventions that we consider, we have $\frac{d\psi(X)_{ij}}{dX_{ij}} = \frac{\lambda}{d(i)}$. Via the chain rule, this yields

$$\frac{dg}{dX_{ij}} = \frac{dg}{d\psi(X)_{ij}} \frac{d\psi(X)_{ij}}{dX_{ij}} = \frac{\lambda}{d(i)} \frac{dg}{d\psi(X)_{ij}}$$

and now we can calculate the gradient by computing $\frac{dg}{d\psi(X)}$. Note that this gradient is a matrix, where the $ij$ entry is $\frac{dg}{d\psi(X)_{ij}}$. Using matrix calculus [34], we can compactly express the derivative as

$$\frac{dg}{d\psi(X)} = \sum_{r=0}^{T-1} (\psi(X)^r)^\top 1_S^\top x(0) (\psi(X)^{T-1-r})^\top$$

By appropriately ordering computations, this can be computed in $T$ matrix multiplications for runtime $T_1 = O(Tn^\omega)$ where $\omega$ is the matrix multiplication constant. .

[1] https://www.dropbox.com/s/ethu39cucqakiyb/supplement_netopt.pdf?dl=0

**Algorithm 2** GreedyDyad

$A, R \leftarrow \emptyset$
**while** $|A| < K_D$ **do**
  $C \leftarrow \{((u,v),(w,v)) : (u,v) \in \overline{E} \setminus A, (w,v) \in E, \chi_v^{in}(A) < \kappa_{in}$ and $\chi_u^{out}(A) < \kappa_{out}\}$
  $((u,v),(w,v)) \leftarrow \arg\max_C \nabla_{v,u} - \nabla_{v,w}$
  $A \leftarrow A \cup \{(u,v)\}$
  $R \leftarrow R \cup \{(w,v)\}$
**return** $X(A,R)$

## 5.3 Linear optimization

We now introduce efficient algorithms for linear optimization over the feasible set $\mathcal{P}$. In general, this can be done by linear programming. However, our problem has $\Theta(n^2)$ variables (entries of $X$). Further, the number of linear inequalities to define $\mathcal{P}$ (which is the convex hull of exponentially many integral points) could be intractably large (particularly in the group setting). The key to computational efficiency is a set of highly efficient greedy algorithms which exploit the combinatorial structure of our domain. Importantly, greedy algorithms for *linear* optimization are very efficient even though greedy optimization of the full objective is computationally expensive.

Note first that any linear objective is optimized at a vertex of $\mathcal{P}$, where the perturbation matrix $X$ has integral values. Hence, in order to solve the continuous linear optimization problem, we can equivalently solve the combinatorial problem of selecting the set of edge additions and removals which best align with the gradient. To translate between the continuous and combinatorial problems, note the following decomposition of the linear objective:

$$\langle \nabla g(X^\ell), X(A,R) \rangle = \sum_{(u,v)\in A} (\nabla g(X^\ell))_{v,u} - \sum_{(u,v)\in R} (\nabla g(X^\ell))_{v,u}$$

which expresses the inner product with the gradient as the sum over which entries are selected by the non-zero elements of $X(A,R)$ (recall that the inner product $\langle \cdot, \cdot \rangle$ outputs a scalar which is the sum of the elementwise products of the two matrices). The first summation corresponds to the entries where $X(A,R)_{v,u} = 1$ and the second summation corresponds to entries where $X(A,R)_{v,u} = -1$. This lets us formulate the linear optimization oracle as the following combinatorial optimization problem:

$$\max_{(A,R)\in C} \sum_{(u,v)\in A} \nabla g(X^\ell)_{v,u} - \sum_{(u,v)\in R} \nabla g(X^\ell)_{v,u} \qquad (2)$$

This combinatorial problem is also linear; hence, its difficulty depends on the underlying constraint structure $C$. We analyze this structure for the different intervention settings that we consider and prove either approximation or hardness results.

**Dyad intervention:** When we choose individual edges to add and remove, a natural strategy is to greedily pick one *pair* of edges at a time, where each pair is of the form $((v,u),(w,u))$. Here $(v,u)$ is an edge to add and $(w,u)$ is a corresponding edge to remove. This ensures that additions and removals are balanced at each step. Algorithm 2 presents the corresponding greedy algorithm, which iteratively selects the pair with highest gradient contribution. Recall that the gradient is a matrix, and by adding edge $(u,v)$ we gain weight $\nabla g(X^\ell)_{v,u}$ towards the scalar objective of the linear optimization problem. We prove the following approximation guarantee:

PROPOSITION 5.2. *Algorithm 2 obtains a $\frac{1}{3}$-approximation to linear optimization in dyad interventions with controlled removals.*

PROOF. We use a hybrid argument. The greedy algorithm runs for $K_D$ iterations; let $S_i$ denote its selections up to iteration $i$. Let $H_0$ denote the set of edges added and removed by $OPT$. $H_i$ for $i = 1...K_D$ changes $H_{i-1}$ by adding the $i$th element that greedy selects and removing some of $OPT$'s elements to ensure feasibility. Hence, $H_{K_D}$ is greedy's selection. Let $w(S)$ be the total weight of $S$ (i.e., the objective value for Problem 2). We show that there is a valid sequence $H_1...H_{K_D}$ such that at each iteration $i$, $w(S_i) - w(S_{i-1}) \geq \frac{1}{2}(w(H_{i-1}) - w(H_i))$. To see this, suppose that greedy adds edge $(u,v)$ and removes edge $(w,v)$, adding the pair $((u,v),(w,v))$ to its solution. To maintain feasibility for $H_i$, we claim that at most two elements from $H_{i-1}$ need be discarded. Discarding any single item suffices to satisfy the global budget constraint $K_D$. Besides this, adding $((u,v),(w,v))$ to $H_i$ can violate at most two constraints: the constraint $\kappa_{in}$ the number of in-edges added to $v$ and the constraint $\kappa_{out}$ on the number of out-edges added to $u$. Both can be repaired by removing one pair which contains an in-edge for $v$ and one pair that contains an out-edge for $u$. Both pairs were candidates for greedy to choose in iteration $i$, so $((u,v),(u,w))$ must have at least as high weight as both individually. From this, the claim that $w(S_i) - w(S_{i-1}) \geq \frac{1}{2}(w(H_{i-1}) - w(H_i))$ follows. Hence, we lose a value at most twice what we gain. Summing over all iterations gives

$$\sum_{i=1}^{K_D} w(S_i) - w(S_{i-1}) \geq \frac{1}{2} \sum_{i=1}^{K_D} w(H_{i-1}) - w(H_i)$$

which implies that

$$w(S_{K_D}) - w(S_0) \geq \frac{1}{2}(w(H_0) - w(H_{K_D}))$$

and since $w(S_0) - 0$ and $w(H_K) = w(S_{K_D})$, we conclude that $w(S_{K_D}) \geq \frac{1}{3} w(H_0) = \frac{1}{3} OPT$.

□

We also note that Algorithm 2 takes time $T_2 = O(K_D n^2)$.

**Group intervention:** We now turn to the setting where the intervention establishes a set of peer groups. Each node in a given group establishes edges with the other nodes in that group. We can easily reduce from the densest $k$ subgraph (DKS) problem to show:

PROPOSITION 5.3. *Optimizing linear functions over the feasible set in the group setting is NP-hard*

Unfortunately, there are not even constant factor approximation algorithms known for DKS. We propose an efficient greedy heuristic which iteratively grows each group. A naive greedy approach would be to greedily form the first group $G_1$, then the second group $G_2$, and so on. However, this approach fails because it over-allocates target nodes to the first groups instead of spreading them equally over all of the groups. Too many target nodes in a single group results in "deviancy training" [12] where the target nodes exert a negative influence on each other, exceeding the positive influence of the healthy role models in the group (see Section 7). Accordingly, we take an

**Algorithm 3** GreedyGroup

1: $A, R \leftarrow \emptyset$
2: $G \leftarrow \{G_k = \emptyset | k = 1...K_G\}$
3: //Select best seed
4: **for** $k = 1...K_G$ **do**
5: //TopSum$(x, S)$ returns sum of largest $x$ elements of set $S$
6: *AddVal(v)* $\leftarrow$ TopSum$(M-1, \{\nabla_{v,u}, u \notin \delta^{in}(v)\})$
7: *RemoveVal(v)* $\leftarrow$ TopSum$(M-1, \{-\nabla_{v,w} w \in \delta^{out}(v)\})$
8: $v \leftarrow \arg\max_{v \in V \setminus G_{k-1} \dots G_1} AddVal(v) + RemoveVal(v)$
9: $G_k \leftarrow G_k \cup v$
10: //Select group members with highest weight gained
11: **for** $M$ iterations **do**
12: **for** $k = 1...K_G$ **do**
13: $v \leftarrow \arg\max_{v \in V \setminus G_{k-1} \dots G_1} GroupVal(G_k \cup \{v\}, \nabla) - GroupVal(G_k, \nabla)$
14: $G_k \leftarrow G_k \cup v$
15: //Add edges between group members who are not already linked
16: $A \leftarrow \{(u, v) : u, v \in G_i, i \in 1...K_G, v \notin \delta^{in}(u)\}$
17: //Remove edges with most negative gradient weight
18: $R \leftarrow \bigcup_{G_i} \bigcup_{v \in G_i} \{$the $|G_i \setminus \delta^{in}(v)| - 1$ edges $(u, v)$ of $\delta^{in}(v) \setminus G_i$ minimizing $\nabla_{v,u}\}$
19: **return** $X(A, R)$
20:
21: //Total gradient weight from adding group $G_i$
22: **function** GROUPVAL$(G_i)$
23: $A \leftarrow \{(u, v) : u, v \in G_i, i \in 1...K_G, v \notin \delta^{in}(u)\}$
24: $R \leftarrow \bigcup_{G_i} \bigcup_{v \in G_i} \{|G_i \setminus \delta^{in}(v)| - 1$ edges $(u, v)$ of $\delta^{in}(v) \setminus G_i$ minimizing $\nabla_{v,u}\}$
25: **return** $\sum_{u,v \in A} \nabla_{v,u} - \sum_{u,v \in R} \nabla_{v,u}$

alternate approach which grows the $K_G$ groups simultaneously, one node at a time. Algorithm 3 gives pseudocode. Algorithm 3 starts by finding a good "seed node" to start each group with (lines 4-9). Specifically, it chooses the $K_G$ nodes which have the most extreme $M-1$ entries in the corresponding row of the gradient. These are nodes which have many high-weighted edges to potentially add or remove and so represent good candidates for starting a group. Then, it adds one node at a time to each group until the maximum group size $M$ is reached (lines 11-14). Each new addition is chosen to maximize the total gradient value captured by the edges added and removed in the group. The total runtime is $T_2 = O(K_G M n^2)$.

## 6 RANDOM REMOVALS

We now extend RECONNECT to handle the case where edges are removed at random, not chosen by the algorithm. Here, the algorithm selects $A \in C_A$ and $R$ is drawn at random from some distribution $p_{remove}(A)$ over $C_R(A)$ (induced by the random choice of each agent on which edges to remove). We aim to solve the problem $\max_{A \in C_A} \mathbb{E}_{R \sim p_{remove}(A)}[f(A, R)]$.

In the random removal case, the full perturbation matrix is not chosen by the algorithm. The decision variable $X$ is a partial perturbation matrix which only has ones in the entries where edges are added. Then, the full perturbation matrix is a random variable which depends on which entries are removed. Let $\mathcal{P}_R = conv(\{X(A, \emptyset) : A \in C_A\})$. $\mathcal{P}_R$ is the convex hull of feasible partial perturbation matrices. Consider a point $X \in \mathcal{P}_R$, represented as a convex combination of integral points $X = \sum_i \theta_i X(A_i, \emptyset)$ where $\theta$ is the weights of the convex combination. We define the continuous problem as

$$g_R(X) = \sum_i \theta_i \mathop{\mathbb{E}}_{R \sim p_{remove}(A_i)} \left[ 1_S^\top \psi \left( X(A_i, R) \right)^T x(0) \right]$$

We now give the two main ingredients needed to apply RECONNECT in this setting.

**Gradient oracle:** Differentiating the expression for $g_R$ yields

$$\nabla g_R(X) = \sum_i \theta_i \mathop{\mathbb{E}}_{R \sim p_{remove}(A_i)} \left[ \nabla_X \left[ 1_S^\top X(A_i, R)^T x(0) \right] \right].$$

The main difficulty that we cannot compute the expectation (which ranges over exponentially many scenarios) in closed form. We resolve this dilemma by drawing samples $R_1^i \dots R_m^i$ from $p_{remove}$ for each $A_i$ and averaging over the gradient values at the samples, giving an unbiased estimator of the true gradient. Empirically, we find that relatively few samples suffice for good accuracy.

**Linear optimization oracles:** Our algorithms for linear optimization in the random removals case are analogous to those in the controlled removals case (Algorithms 2 and 3). They simply average over random samples of which edges are removed instead of allowing the algorithm to select the removed edges. In the dyad setting, we prove an approximation guarantee corresponding to that in the controlled case (Proposition 5.2):

PROPOSITION 6.1. *Greedily maximizing the expected change in gradient weight produces a $\frac{1}{3}$-approximation to the linear optimization problem with dyad interventions and random removals.*

See the supplement for a proof. The main insight is that, unlike in the controlled setting, the feasible set $C_A$ can be written as the intersection of two matroid constraints. The proposition then follows from known results for matroids [6]. In the group setting, our hardness result (Proposition 5.3) still applies. However, a greedy strategy modeled on Algorithm 3 (which randomly simulates the removed edges on lines 7 and 24) performs well in practice.

## 7 EXPERIMENTS

We build a highly realistic simulation environment to compare our algorithm to baseline approaches. Our simulation is based on the Antelope Valley community in California. Antelope Valley (AV) covers 2,934 square miles and is larger than the States of Delaware and Rhode Island. The population of the AV region is estimated to be about 447,472. Over 34% of the area's population is under the age of 18. Overall, 33% of the population is Latino and 15.6% is African American. Approximately 15% of the total population lives at or below the U.S. Federal Poverty Level. *Only 10% of children ages 0-17 in this region are perceived by their parents to be healthy*. Rates of obesity in this region are among some of the highest in Los Angeles County: 20% of children are obese (BMI above the 95th percentile), while 36% of adults are overweight and 35% are obese. Hence, AV is a highly salient target for anti-obesity interventions.

**Simulation features:** We use a range of geographic, demographic, and health-related data compiled from the U.S. Census, the Los Angeles County Department of Public Health, and Los Angeles Times Mapping L.A. project. Our simulation consists of a population of $n$

agents who comprise the nodes of a social network. Each agent is endowed with a geographic location, age, ethnicity, gender, and weight status. These features allow us to capture the interdependency between demographics, obesity, and network connections. Each agent's geographic location is one of the 13 major regions which comprise AV (see Figure 1). For each region, we use region-specific distributions for the remaining characteristics provided by the LA Times [42]. Each agent has an ethnicity from {White, Latino, Black, Asian, Other}, a gender from {male, female}, and an age range from {0-10, 11-18, 19-34, 35-49, 50-64, 65+}. They also have a weight status from {nonoverweight, overweight, obese}. "Nonoverweight" denotes that the agent's BMI is at most the 24.9, "overweight" denotes a BMI in range 25-29.9, and "obese" denotes a BMI of at least 30. In terms of the initial state $x$, we assign agents who are obese an initial state of 0, agents who are overweight an initial state of 0.1, and agents who are nonoverweight an initial state of 1. Overweight agents are assigned a state closer to obese than to nonoverweight because they are less likely to proactively model healthy behaviors.

**Generating the agents:** In each simulation, we generate a population of $n$ agents who comprise the subset of the community targeted by a particular intervention. For each agent, we first sample a geographic location with probability proportional to the population inhabiting each region. Then, we sample the agent's remaining demographic characteristics from their region-specific distribution. We sample the agent's weight status using data from the Los Angeles County Department of Public Health on the fraction of people who are nonoverweight, overweight, and obese among each demographic group in the AV region [31]. The target set $S$ was randomly chosen from the obese agents, with $|S| = 0.15 \cdot n$.

**Generating the network:** Once we have sampled a population of agents and their features, we simulate the social network. Our network simulation uses the features to produce networks which reflect widely observed characteristics of real-world networks. Specifically, we use the spatial preferential attachment model [4, 13]. Preferential attachment is a generative process which replicates the heavy-tailed degree distributions observed in many real networks [3]. In classical preferential attachment, the agents arrive one at a time. Each agent who arrives forms $m$ links to agents who arrived previously, selected with probability proportional to the previous agents' degrees. The spatial preferential attachment model modifies preferential attachment to account for homophily, another widely observed network characteristic where agents tend to form links to others with similar characteristics. Let $y_i$ be the feature vector of agent $i$ and $y_j$ be the feature vector of agent $j$. In the spatial preferential attachment model, the probability of $j$ linking to $i$ is proportional to $e^{-\rho||y_i - y_j||_2} d(i)$, where $\rho$ is a parameter. That is, the probability is larger when $i$ has a large degree, but decays with the distance between $i$ and $j$ in feature space. We take $\rho = 0.1$ throughout.

**Baselines:** No previous work directly addresses our setting, so we construct competitive baselines using alternate algorithmic approaches to the problem. First, *greedy*. In the dyad case, greedy successively adds the edge to add (and remove, if applicable) which results in the largest increase in the objective value. In the group case, greedy successively adds the agent to the group which results in greatest value (while also greedily choosing edges to remove in the controlled removal case). We also tried growing the groups simultaneously (as in Algorithm 3) and observed very similar results. Second, *matching*, which randomly matches unhealthy target agents with healthy ones. This represents a natural intervention strategy which connects agents to healthy influences without considering the overall network topology. In the dyad case, matching evenly divides the edge addition budget $K_D$ among the target agents, connecting each one to $\frac{K_D}{|S|}$ healthy agents chosen at random (recall that $S$ is the set of target agents). In the group case, it randomly places $\frac{|S|}{K_G}$ target agents into each group, and fills the rest of the group with random healthy agents. When removals are controlled, matching randomly removes edges from target agents to unhealthy neighbors. Third, *simulated annealing (SA)*, which is a common black-box optimization technique. SA is guaranteed to find the global optimum if run long enough, but may require exponential time to do so. Lastly, we test a variation of RECONNECT, *REC-myopic*, which runs RECONNECT with time horizon $T = 1$. Since the $T = 1$ objective is linear, our greedy algorithms can efficiently find high-quality solutions (with constant-factor approximation guarantee for the dyad case). However, REC-myopic does not consider long-term dynamics.

**Results:** We now analyze the solution quality and runtime produced by each algorithm in each of our four settings. Figure 3 shows solution quality as we vary the network size $n$. For the dyad case, we fix $K_D = 0.2 \cdot n$, $\kappa_{in} = \kappa_{out} = 3$, $\lambda = 0.5$. For the group case, we use $K_G = 0.05 \cdot n$ and $M = 5$. Both settings use $T = 10$. We ran RECONNECT for $L = 10$ iterations with step size $\gamma_\ell = \frac{1}{0.25\ell+1}$. The $x$ axis on each plot shows the number of agents in the network and the $y$ axis shows the improvement produced by each algorithm averaged over 24 networks. Improvement is the difference between the objective value produced by the algorithm's intervention and the value of not intervening (allowing the network to evolve for $T$ steps unmodified). For instance, the point in Figure 3(a), $n = 3000$, for RECONNECT indicates that RECONNECT produced a total improvement of 230 in the state of target nodes compared to not intervening. Intuitively, RECONNECT averted 230 cases of obesity (though the impact may be split fractionally across multiple agents). Error bars show one standard deviation, but are sometimes hidden underneath markers. RECONNECT outperforms all of the baselines in all settings. The difference is particularly dramatic in the controlled dyad case, where RECONNECT outperforms the next baseline by more than 150% for $n = 3000$. For the group case, REC-myopic is closer (though always outperformed by the full RECONNECT).

Greedy and SA do not scale past 250-node networks and were cut off after 24 hours of runtime (indicated by the dashed lines near the $x$ axis in Figure 3). Greedy's solution quality is comparable with RECONNECT for networks with under 250 nodes. However, we stress that realistic interventions will involve networks with thousands of nodes, and greedy's runtime explodes as $n$ grows past 100 nodes. Hence, RECONNECT's advantage is providing performance which is at least as good in a highly scalable manner. SA is always substantially outperformed by RECONNECT.

Figure 4 examines the runtime of each algorithm. RECONNECT successfully scales to networks with thousands of nodes within 100-1000 seconds. REC-myopic and matching also run quickly. However, greedy and SA are extremely computationally expensive. Greedy was terminated past 100-node networks after 24 hours of runtime, while SA was terminated for networks larger than 250 nodes. Greedy

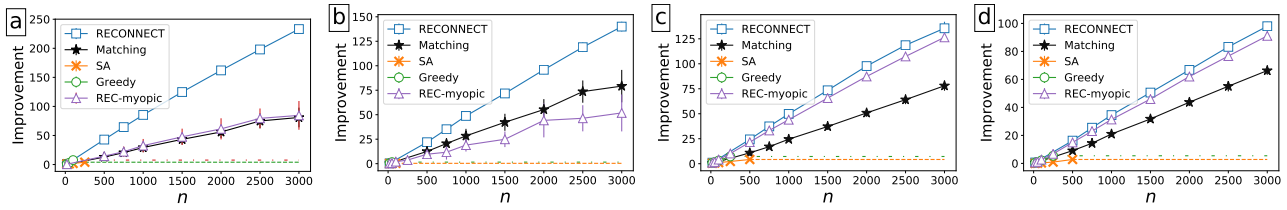


**Figure 3: Improvement produced by each algorithm as $n$ grows. (a) Dyad intervention, controlled removals (b) Dyad intervention, random removals (c) Group intervention, controlled removals (d) Group intervention, random removals.**

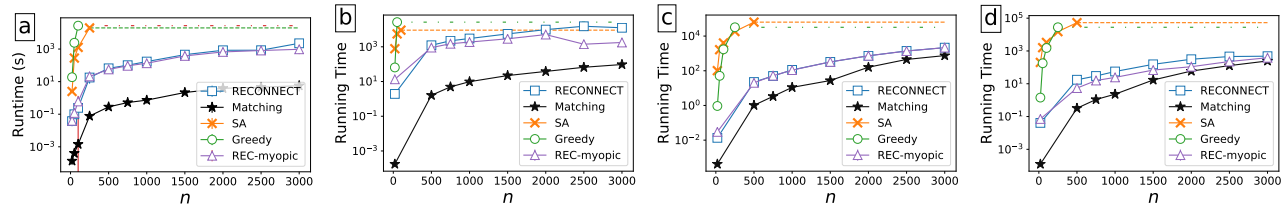


**Figure 4: Runtime for each algorithm as $n$ grows. (a) Dyad intervention, controlled removals (b) Dyad intervention, random removals (c) Group intervention, controlled removals (d) Group intervention, random removals.**

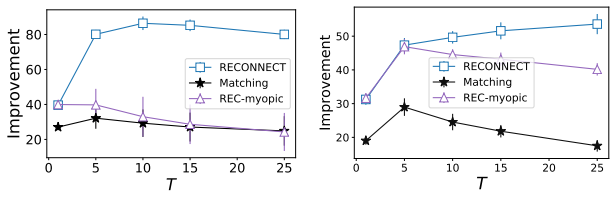


**Figure 5: Performance varying $T$ with controlled removals. Left: dyad, right: group. $n = 1000$.**

**Table 1: Average improvement on Facebook. "D" and "G" denote dyad/group; "C" and "R" denote controlled/random.**

| | D/C | D/R | G/C | G/R |
|---|---|---|---|---|
| RECONNECT | 263.6 | 80.9 | 120.2 | 108.6 |
| Matching | 20.36 | 28.1 | 7.59 | 10.39 |
| REC-myopic | 56.18 | 23.5 | 103.7 | 98.27 |

is slow in our domain because it needs $\Theta(n^2)$ objective function evaluations per iteration in the dyad setting (evaluating each edge to add or remove) and $\Theta(nK_G)$ in the group setting.

Figure 5 shows solution quality as we vary the time horizon $T$ for $n = 1000$. We show results for controlled removals; other settings, as well as results varying the budgets $K_D$ and $K_G$, can be found in the supplement. RECONNECT's advantage over REC-myopic grows as $T$ increases, particularly in the group setting; e.g., RECONNECT outperforms REC-myopic by 32.5% for $T = 25$. In the dyad setting, the performance of all algorithms decreases slightly for large $T$ since the long-term planning problem is more difficult. However, RECONNECT still substantially outperforms the baselines.

Lastly, we test robustness to alternate network topologies by experimenting on the *facebook* dataset with 2888 nodes [29]. We exclude greedy and SA due to the network size and use the same parameter settings as in Figure 1. Agent states are assigned uniformly at random since we have no demographic information. Table 1 shows that RECONNECT outperforms all of the baselines in each setting.

## 8 CONCLUSION

Social network based interventions are a critical multiagent tool for creating behavior change. We introduced the algorithmic problem of preventative health network interventions and proposed the RECONNECT algorithm to find scalable, high-quality solutions across a range of network intervention designs. RECONNECT exploits the problem's combinatorial structure to enable scale-up, and we give theoretical analysis for the subproblem that it solves in each domain.

To test our algorithm, we created a highly realistic simulation environment of the Antelope Valley region of California. AV is a highly salient target for preventative health interventions, and our simulation draws on a range of demographic and health-related data. RECONNECT outperforms an array of competitive baselines in each setting, sometimes improving on the best alternative by over 150%. Our work demonstrates that algorithmic approaches have substantial promise in aiding preventative health interventions.

**Acknowledgments:** This research was supported in part by NSF Grant CCF-1522054 and in part by a pilot grant awarded to K. de la Haye: Grant Number U54HD070725 from the Eunice Kennedy Shriver National Institute of Child Health & Human Development (NICHD). The project is co-funded by the NICHD and the Office of Behavioral and Social Sciences Research (OBSSR). The content is solely the responsibility of the authors and does not necessarily represent the official views of the NICHD or OBSSR. Wilder was supported by a NSF Graduate Research Fellowship

# REFERENCES


[1] Fatma Al-Maskari. 2010. Lifestyle diseases: An economic burden on the health services. *UN Chronicle. The Magazine of the United Nations* (2010).

[2] Albert Bandura. 1977. Social learning theory Englewood Cliffs. (1977).

[3] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.

[4] Marc Barthélemy. 2011. Spatial networks. *Physics Reports* 499, 1 (2011), 1–101.

[5] Rahmatollah Beheshti and Gita Sukthankar. 2014. A normative agent-based model for predicting smoking cessation trends. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 557–564.

[6] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.* 40, 6 (2011), 1740–1766.

[7] Arun G Chandrasekhar, Horacio Larreguy, and Juan Pablo Xandri. 2015. *Testing models of social learning on networks: Evidence from a lab experiment in the field*. Technical Report. National Bureau of Economic Research.

[8] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 199–208.

[9] Nicholas A Christakis and James H Fowler. 2007. The spread of obesity in a large social network over 32 years. *N Engl J Med* 2007, 357 (2007), 370–379.

[10] Kayla De La Haye, Garry Robins, Philip Mohr, and Carlene Wilson. 2011. Homophily and contagion as explanations for weight similarities among adolescent friends. *Journal of Adolescent Health* 49, 4 (2011), 421–427.

[11] Morris H DeGroot. 1974. Reaching a consensus. *J. Amer. Statist. Assoc.* 69, 345 (1974), 118–121.

[12] Thomas J Dishion, Joan McCord, and Francois Poulin. 1999. When interventions harm: Peer groups and problem behavior. *American psychologist* 54, 9 (1999), 755.

[13] Luca Ferretti and Michele Cortelezzi. 2011. Preferential attachment in growing spatial networks. *Physical Review E* 84, 1 (2011), 016103.

[14] Marguerite Frank and Philip Wolfe. 1956. An algorithm for quadratic programming. *Naval Research Logistics (NRL)* 3, 1-2 (1956), 95–110.

[15] Amer G Ghanem, Srinivasa Vedanarayanan, and Ali A Minai. 2012. Agents of influence in social networks. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 551–558.

[16] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. 2011. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*. ACM, 47–48.

[17] Barbara J Grosz, Sarit Kraus, Shavit Talman, Boaz Stossel, and Moti Havlin. 2004. The influence of social dependencies on decision-making: Initial investigations with a new game. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*. IEEE Computer Society, 782–789.

[18] Ross A Hammond. 2010. Social influence and obesity. *Current Opinion in Endocrinology, Diabetes and Obesity* 17, 5 (2010), 467–471.

[19] Matthew O Jackson. 2010. *Social and economic networks*. Princeton university press.

[20] Markus Juonala, Costan G Magnussen, Gerald S Berenson, Alison Venn, Trudy L Burns, Matthew A Sabin, Sathanur R Srinivasan, Stephen R Daniels, Patricia H Davis, Wei Chen, et al. 2011. Childhood adiposity, adult adiposity, and cardiovascular risk factors. *New England Journal of Medicine* 365, 20 (2011), 1876–1885.

[21] David Kempe, Jon Kleinberg, and Éva Tardos. 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*. ACM, 137–146.

[22] Elias Boutros Khalil, Bistra Dilkina, and Le Song. 2014. Scalable diffusion-aware optimization of network topology. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1226–1235.

[23] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. 2009. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 3, 2 (2009), 9.

[24] Michel Klein, Adnan Manzoor, Julia Mollee, and Jan Treur. 2014. Effect of changes in the structure of a social network on emotion contagion. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03*. IEEE Computer Society, 270–277.

[25] Chris J Kuhlman, Gaurav Tuli, Samarth Swarup, Madhav V Marathe, and SS Ravi. 2013. Blocking simple and complex contagion by edge removal. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 399–408.

[26] Simon Lacoste-Julien. 2016. Convergence rate of Frank-Wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345* (2016).

[27] Ronan Le Bras, Bistra Dilkina, Yexiang Xue, Carla P Gomes, Kevin S McKelvey, Michael K Schwartz, and Claire A Montgomery. 2013. Robust network design for multispecies conservation. (2013).

[28] Peter V Marsden and Noah E Friedkin. 1993. Network studies of social influence. *Sociological Methods & Research* 22, 1 (1993), 127–151.

[29] KONECT network dataset collection. 2017. http://konect.uni-koblenz.de/networks/ego-facebook. (2017).

[30] Kathleen Norr, Sheila Tlou, and Matshidiso Moeti. 2004. Impact of peer group education on HIV prevention among women in Botswana. *Health Care for Women International* 25, 3 (2004), 210–226.

[31] Los Angeles County Department of Public Health. 2015. Los Angeles County Health Survey. http://www.publichealth.lacounty.gov/ha/hasurveyintro.htm. (2015).

[32] Martin Olsen and Anastasios Viglas. 2014. On the approximability of the link building problem. *Theoretical Computer Science* 518 (2014), 96–116.

[33] World Health Organization. 2016. Report of the Commission on ending childhood obesity. 2016. (2016).

[34] Kaare Brandt Petersen and Michael Syskind Pedersen. 2008. The matrix cookbook. *Technical University of Denmark* 7 (2008), 15.

[35] Tuan Q Phan and Edoardo M Airoldi. 2015. A natural experiment of social network formation and dynamics. *Proceedings of the National Academy of Sciences* 112, 21 (2015), 6595–6600.

[36] Sarah-Jeanne Salvy, Kayla De La Haye, Julie C Bowker, and Roel CJ Hermans. 2012. Influence of peers and friends on children's and adolescents' eating and activity behaviors. *Physiology & behavior* 106, 3 (2012), 369–378.

[37] S-J Salvy, K Haye, Titus Galama, and Michael I Goran. 2017. Home visitation programs: an untapped opportunity for the delivery of early childhood obesity prevention. *Obesity Reviews* 18, 2 (2017), 149–163.

[38] Daniel Sheldon, Bistra Dilkina, Adam N Elmachtoub, Ryan Finseth, Ashish Sabharwal, Jon Conrad, Carla Gomes, David Shmoys, William Allen, Ole Amundsen, et al. 2010. Maximizing the spread of cascades using network design. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 517–526.

[39] M Simmonds, A Llewellyn, CG Owen, and N Woolacott. 2016. Predicting adult obesity from childhood obesity: a systematic review and meta-analysis. *Obesity reviews* 17, 2 (2016), 95–107.

[40] Sandra D Simpkins, David R Schaefer, Chara D Price, and Andrea E Vest. 2013. Adolescent friendships, BMI, and physical activity: untangling selection and influence through longitudinal social network analysis. *Journal of Research on Adolescence* 23, 3 (2013), 537–549.

[41] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 75–86.

[42] Los Angeles Times. 2017. Mapping L.A. http://maps.latimes.com/neighborhoods/region/antelope-valley/. (2017).

[43] Thomas W Valente. 2010. *Social networks and health: Models, methods, and applications*. Oxford University Press.

[44] Thomas W Valente. 2012. Network interventions. *Science* 337, 6090 (2012), 49–53.

[45] Thomas W Valente, Kayo Fujimoto, Chih-Ping Chou, and Donna Spruijt-Metz. 2009. Adolescent affiliations and adiposity: a social network analysis of friendships and obesity. *Journal of Adolescent Health* 45, 2 (2009), 202–204.

[46] Thomas W Valente, Beth R Hoffman, Annamara Ritt-Olson, Kara Lichtman, and C Anderson Johnson. 2003. Effects of a social-network method for group assignment strategies on peer-led tobacco prevention programs in schools. *American journal of public health* 93, 11 (2003), 1837–1843.

[47] Bryan Wilder, Amulya Yadav, Nicole Immorlica, Eric Rice, and Milind Tambe. 2017. Uncharted but not Uninfluenced: Influence Maximization with an uncertain network. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1305–1313.

[48] Yexiang Xue, Xiaojian Wu, Dana Morin, Bistra Dilkina, Angela Fuller, J Andrew Royle, and Carla P Gomes. 2017. Dynamic Optimization of Landscape Connectivity Embedding Spatial-Capture-Recapture Information.. In *AAAI*. 4552–4558.

[49] Amulya Yadav, Hau Chan, Albert Xin Jiang, Haifeng Xu, Eric Rice, and Milind Tambe. 2016. Using social networks to aid homeless shelters: Dynamic influence maximization under uncertainty. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 740–748.

[50] Chao Yu, Minjie Zhang, Fenghui Ren, and Xudong Luo. 2013. Emergence of social norms through collective learning in networked agent societies. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 475–482.