Optimization Letters (2023) 17:1853–1872
https://doi.org/10.1007/s11590-022-01963-w

ORIGINAL PAPER

On the firefighter problem with spreading vaccination for maximizing the number of saved nodes: the IP model and LP rounding algorithms

Yongge Yang[1] · Po-An Chen[2] · Yu-Ching Lee[1] · Yung-Yan Fanchiang[2]

Received: 29 December 2021 / Accepted: 20 December 2022 / Published online: 30 December 2022

Abstract

When an infectious disease spreads, how to quickly vaccinate with a limited budget per time step to reduce the impact of the virus is very important. Specifically, vaccination will be carried out in every time step, and vaccinated nodes will no longer be infected. Meanwhile, the protection from vaccination can spread to the neighbors of a vaccinated node. Our goal is to efficiently find optimal and approximation solutions to our problem with various algorithms. In this paper, we first design an integer linear program to solve this problem. We then propose approximation algorithms of (1) Linear programming (LP) deterministic threshold rounding, (2) LP dependent randomized rounding, and (3) LP independent randomized rounding. We prove that the LP independent randomized rounding algorithm has a high probability of finding a feasible solution that gives an approximation ratio of $(1-\delta)$, where a small constant δ between 0 and 1 reduces the lower bound on the *feasibility probability*. We also provide experimental results for three different rounding algorithms to show that they perform numerically well in terms of approximation ratios. These analytical and numerical studies allow each individual to adopt the most appropriate approximation algorithm to efficiently resolve the vaccination problem when her reliance on commercial optimization solvers is costly.

Keywords Firefighter problem with vaccination spreading · Vaccination strategy · Integer program · Randomized rounding

✉ Po-An Chen
poanchen@nycu.edu.tw

✉ Yu-Ching Lee
yclee@ie.nthu.edu.tw

Extended author information available on the last page of the article

1 Introduction

Humans have been battling infectious diseases, such as SARS, COVID-19, etc. There will be serious consequences if we cannot efficiently prevent the spread of viruses. Thus, when an epidemic outbreak occurs, it is important to find ways to save more people with limited budgets or resources in the form of vaccination. Similarly, when there is an outbreak of a computer virus or fire, the key question is to efficiently find the best strategy for vaccinating the people or controlling the fire so that the spread can be contained. Note that the classic firefighter problem is when a set of fire nodes breaks out at time 0, and the fire will spread to its neighbor nodes if the nodes are not saved by the firefighter, how can we send a limited number of firefighters to stop the fire at each time step? If there are no more nodes to spread to, the process will terminate.

A vaccination problem, an extension of the classic firefighter problem, was proposed in [2]. In this problem, all the nodes in graph $G = (V, E)$ are in one of the three different states: "infected", "stateless" and "vaccinated". Moreover, no two or more states at one node exist at the same time, and the nodes might change their state at each time step. In the beginning, all the nodes are labeled "stateless", and then there is a given node or set of nodes infected at time 0 in G, i.e., the outbreak. We are allowed to vaccinate B nodes if time $t > 0$, but we can vaccinate only the "stateless" nodes. After the vaccination, the "infected" nodes will still infect all the "stateless" nodes adjacent to the "infected" nodes; if there are "vaccinated" nodes, the nodes will not be infected in the infecting process. When there are no more "stateless" nodes to be infected, the process will stop. Note that thus far the vaccination problem is similar to the classic firefighter problem. Here, the neighbors of a node are nodes that are connected to it and reachable via one single edge.

Furthermore, Anshelevic et al. [2] considered two models. The first is the "non-spreading vaccination model" where vaccinated nodes cannot spread vaccination to their neighbors, and is similar to the classic firefighter problem. The second is the "spreading vaccination model", where vaccinated nodes can also vaccinate their neighbors in the next time step. In this paper, we focus on the *spreading vaccination* model. Note that if stateless nodes are going to be infected and vaccinated at the same time, they will be vaccinated. Our goal is to create a *vaccination strategy* with the budget constraint in each time step, i.e., what subset of nodes to be vaccinated in each time step, that can maximize the number of uninfected nodes after the process ends. This problem is known as the firefighter problem with the "MaxSave" objective. (More details about the MaxSave objective are given in Sect. 2.)

An example of the vaccination strategy in the spreading vaccination model is given in Fig. 1. We set the initial infection s at node 0, and the budget for each time step is 1. At time 0, node 0 is infected. At time 1, we vaccinate node 2, and because in this time step none of the nodes were vaccinated at time 0, there are no vaccinated nodes that can start to spread vaccination at time 1. Meanwhile, the infected node infects nodes 4, 5, and 6. At time 2, we vaccinate node 3, and node

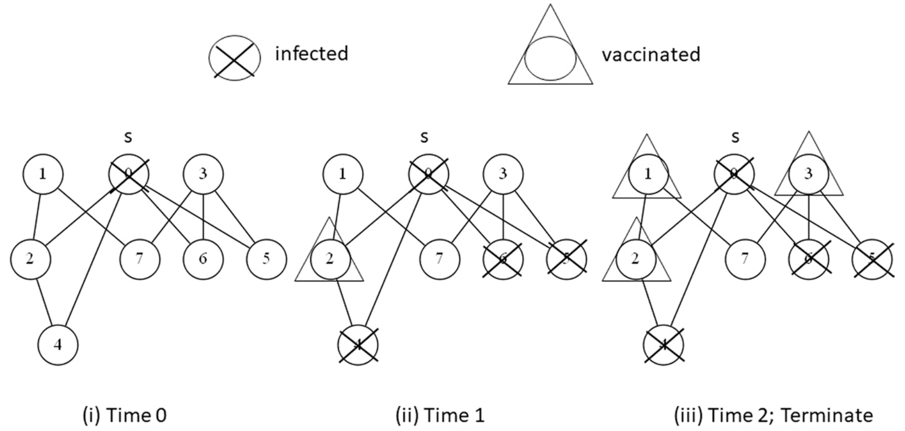

Fig. 1 An example of the process

1 will also be vaccinated by node 2 because of the spreading vaccination. When there are no nodes that can be infected, the process stops. Our vaccination strategy is to vaccinate node 2 at time 1 and vaccinate node 3 at time 2. We will save a total of four nodes (nodes 1, 2, 3, and 7) with this vaccination strategy.

1.1 Previous works

There are several different kinds of firefighter problems whose algorithms and complexity results are examined by [9], with several different objectives, such as maximizing the number of saved nodes and minimizing the number of vaccinated nodes for saving a certain subset of targeted nodes. There are also related works on a vaccination problem with a completely different nature and an outbreak detection problem. More realistically, there are models of disease infection spreading processes with consideration of vaccination.

- *Greedy algorithms for submodular set functions with partition matroids* Our firefighter problem is a special case of the problem of maximizing a monotone submodular set function over a partition matroid constraint [2]. In particular, our problem uses time steps to partition the ground set, and the constraint in our problem is the budget for the number of nodes that can be vaccinated per time step. See [2] for more details. By maximizing a monotone submodular set function over a partition matroid constraint, there is a simple greedy algorithm that gives a 1/2 approximation [18] and a more sophisticated greedy algorithm that gives a $(1 - 1/e)$ approximation [5] for the firefighter problem with the spreading vaccination model.
- *Firefighter problem on trees* L. Cai et al. [4] obtained a $(1 - 1/e)$-approximation for the firefighter problem on trees by the LP relaxation and randomized rounding algorithm. Hartke [10] used a relaxation of the integer program of the firefighter problem proposed by [15]. For all nodes in the graph, they found the

strategy set that can maximize the defended nodes under two set of constraints that need to be satisfied. In the original integer program, the first set of constraint is that at each time step, at most one node can be defended, and the second set of constraints is that at most one ancestor of each node is defended (including itself). Under LP relaxation, the first set of constraints means that the sum of fractional solutions at each time step is at most 1, and the second set of constraints means that the sum of fractional solutions assigned to the ancestors of each node is at most 1. Because of the difficulty of solving the problem while adding the nonlinear constraint, the authors used another method to narrow the integrality gap.

- *Vaccination problem and outbreak detection* In addition to the firefighter problem, there are some researches about vaccination strategies. In [3], the authors considered which nodes should be vaccinated before a virus starts to spread from a random node. When a node is vaccinated, it will not be infected when the virus spreads. In addition, there are two different types of costs for a node: the vaccination cost and the infected cost. This problem was shown to be reduced to the “sum-of-squares partition” problem, and the approximation guarantee proved to be $O(\log^{1.5} n)$. In [6], the authors showed that using the technique of region-growing rounding a natural linear program can improve the approximation ratio from $O(\log^{1.5} n)$ to $O(\log z)$, where z is the support size of the outbreak distribution.

  In [13], they showed that instead of saving the maximum number of nodes in the given graph during the disease outbreak, they considered detecting an outbreak earlier in a network by selecting a small set of people to be sensors, which will alert when they detect the virus. The goal of this problem is to minimize the infection while the outbreak is detected by the sensors so that people can take appropriate measures faster and save more lives. In addition because of the submodularity of this problem, the authors found an efficient algorithm (cost-effective lazy forward selections). There are also many similar problems that can be solved by this framework, such as detecting contaminants in the water distribution network and selecting specific blogs to read so that we do not miss important information.
- *Disease infection spreading processes with consideration of vaccination* There is a paper on epidemiological modeling (the so-called “compartmental” models, including the susceptible–infectious–recovered (SIR) model) and obtained results for epidemiological thresholds [12]. That is, the density of susceptible people must exceed a critical value for an epidemic outbreak to occur. In [8], the authors compared population-based prediction models (the compartmental models) and spatially explicit individual-based prediction models for animal disease transmission and found that spatial individual-based models typically eradicate disease with approximately 10 times lower immunization coverage than population prediction models.

  More recently, there is a special report about the COVID-19 [1], and the authors pointed out that the models will involve more variables as researchers discover more about the virus. Many studies about infectious disease considers the compartmental models. (For a survey, see [11].) However, these models do not explicitly consider the role of social network structure on disease transmission. For this issue, in [7] the authors considered the targeted immunization problem in the epidemic outbreaks and solved this by the influence maximization problem. Their results suggested that identifying optimal immunization populations is particularly important for containing infectious disease outbreaks in small networks. Also in [20], the authors studied the transmission process of infectious diseases through the influence maximization problem to investigate the discrete transmission properties and modeled behavioral changes associated with preventive measures, e.g., wearing or not wearing masks, during epidemic transmission through the network.

1.2 Our results

According to [2], the firefighter problem with the MaxSave objective in the spreading vaccination model can be formulated as maximizing a submodular set function with matroid constraints, and because of the submodularity property, the approximation ratio of this problem is 1/2 with a deterministic greedy algorithm [18] and is $(1 - 1/e)$ as a result of the work [5], where e is the Euler's number.

Inspired by the linear programming (LP) rounding approach (for example, see [17]), we construct an integer linear programming (IP) formulation for this vaccination-spreading firefighter problem with the MaxSave objective, which can be solved by a generic IP solver. Then, because our problem is NP-complete [2], we propose three approximation algorithms of LP rounding. Note that we focus on the vaccination strategies and simply adopt a deterministic infection spreading model,[1] instead of explicitly modeling the disease infection spreading process like in several forementioned previous works, for simplicity of integer programming and thereby enabling approximability via LP relaxation. Specifically, we give one deterministic threshold rounding algorithm and two randomized rounding algorithms, one of which is analyzed for obtaining a feasible solution that guarantees an approximation ratio with a high probability. We evaluate all of them numerically with experiments as well.

1.3 Organization of this paper

We formally introduce the problem in Sect. 2 and propose the IP model and the LP rounding algorithms with some analyses for the approximation ratios in Sect. 3. We give the simulations and numerical results in Sect. 4 and conclude with future work in Sect. 5.

[1] Nonetheless, we can extend the model to consider a stochastic infection spreading process via stochastic (integer) programming like in [19] for influence maximization (in expectation).

2 Preliminaries: vaccination-spreading firefighter problem with the MaxSave objective

According to [2], given that a definition of the vaccination strategy is $X \subseteq V \times D$, (where V is the set of the vertices in graph G, $D = \{1, 2, \dots, T\}$, T is the longest path of the shortest paths from the original infected nodes in graph G,) node u is vaccinated at time $t \in D$ with the vaccination strategy X if $(u, t) \in X$. If the nodes are vaccinated, they will not become infected or stateless, the vaccinated nodes will always be vaccinated. The vaccination budget constraint states that at each time step, there are at most B nodes vaccinated.

As we stated in the Introduction, we adapt the spreading vaccination model; that is, when node u is vaccinated at time $t \geq 1$, stateless nodes v which are adjacent to u such that $(u, v) \in E$, will also be vaccinated at time $t + 1$. If there is a situation in which the nodes are going to be infected and vaccinated at the same time t, the nodes will be vaccinated instead of being infected. The process stops when the infection cannot spread further; that is, there are no stateless nodes adjacent to an infected node. In below we formalize the vaccination strategy problem, which is named also as the "firefighter" problem with spreading vaccination [2].
FIREFIGHTER PROBLEM WITH SPREADING VACCINATION
INSTANCE: A rooted graph $(G(V, E), s)$, and an integer $B \geq 1$
OBJECTIVE: There is an initial node s that has been infected at time 0. We aim to find the vaccination strategy X with the budget constraint in each time step. When the process is stopped, the number of nodes that are not infected is maximized.

A set $S(v) \subseteq V \times D$ for every node $v \in V$ is defined to characterize whether node $v \in V$ is saved by the vaccination strategy X or not, which is

$$S(v) := \{(u, t) | u \in V \text{ and } 0 < t \leq d(s, v) - d(u, v)\}. \tag{1}$$

The tuple (u, t) is vaccinated node u at time t, where $d(u, v)$ is the length of the shortest path from node u to node v in graph G. It means the node v will be saved if our vaccination strategy satisfies the condition $t \leq d(s, v) - d(u, v)$. We can also use this condition to check all the nodes in G and find out how many nodes were saved by the vaccination strategy X, that is, $X \cap S(v) \neq \emptyset$.

Anshelevic et al. [2] showed that the firefighter problem in the spreading vaccination model is a problem of maximizing a monotone submodular set function over a partition matroid constraint. Using this property, they conducted some analysis for their greedy algorithms. They did not employ any IP/LP for their problem, which is, on the contrary, the approach that we take to design LP rounding algorithms in this paper.

3 IP model and LP rounding algorithms

In this section, we first design an integer linear program to solve the firefighter problem with spreading vaccination in Sect. 3.1. In Sect. 3.2, we further propose approximation algorithms of (1) LP deterministic threshold rounding, (2) LP dependent randomized rounding, and (3) LP independent randomized rounding. The major result presented is twofold. First, we showed that Algorithm 1 is an approximation algorithm; that is, there exists a valid bound on the objective values obtained (Theorem 1, Theorem 2). Second, we prove that Algorithm 3 has a high probability of finding a feasible solution that gives an approximation ratio of $(1 - \delta)$, where a small constant δ between 0 and 1 reduces the lower bound on the feasibility probability (Theorem 3).

3.1 Integer linear program

We aim to maximize the number of saved nodes within the limited time steps T. Recall that s is the initial infected nodes, and B is the budget per time step. Let u denote the node we want to vaccinate and v denote the node we want to save. For each $v \in V, u \in V$ and $1 \leq t \leq T$, the variables y_v and $x_{u,t}$ are defined as follows:

$$y_v = \begin{cases} 1 & \text{if } v \text{ is saved by the vaccination strategy;} \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{u,t} = \begin{cases} 1 & \text{if node } u \text{ is vaccinated at time t;} \\ 0 & \text{otherwise.} \end{cases}$$

The 0-1 integer linear program of the vaccination-spreading firefighter problem for maximizing the saved nodes is as follows:

$$\max \sum_{v \in V} y_v \tag{2}$$

$$\text{s.t.} \quad \sum_{u \in V} x_{u,t} \leq B \quad \forall t \in D, \tag{3}$$

$$y_v \leq \sum_{u \in V} \sum_{t=1}^{d(s,v)-d(u,v)} x_{u,t} \quad \forall v \in V, \tag{4}$$

$$x_{u,t} \in \{0, 1\} \ \forall u \in V, \ t \in D, \tag{5}$$

$$y_v \in \{0, 1\} \ \forall v \in V. \tag{6}$$

For a vaccination strategy, our objective is to maximize the number of saved nodes. The first set of constraints means that it can only vaccinate at most B nodes in each time step. The second set of constraints means that whether node v can be saved or not is determined by whether node u can be vaccinated before infected node s spreads to node v. That is, if node u has been vaccinated at any time from $t = 1$ to $d(s, v) - d(u, v)$, then node v could be saved because $d(s, v) > d(u, v)$. Thus, node v will not be saved if none of node u are vaccinated before infected node s spreads to node v: that is, $\sum_u \sum_{t=1}^{d(s,v)-d(u,v)} x_{u,t} = 0$, and $y_v = 0$. Otherwise, the node v will be saved if any one of node u is vaccinated before infected node s spreads to node v: that is, $\sum_u \sum_{t=1}^{d(s,v)-d(u,v)} x_{u,t} \geq 1$ and $y_v = 1$. For convenience, let the objective value of the optimal (integral) solution be OPT.

3.2 LP rounding as approximation algorithms

As the decision variables $x_{u,t}$ and y_v are binary variables, the problem is difficult to solve. Thus, we transform the original integer linear program into a linear program, which is

$$\max \sum_{v \in V} y_v \tag{7}$$

$$\text{s.t.} \quad \sum_{uinV} x_{u,t} \leq B \quad \forall t \tag{8}$$

$$y_v \leq \sum_{u \in V} \sum_{t=1}^{d(s,v)-d(u,v)} x_{u,t} \quad \forall v \tag{9}$$

$$x_{u,t} \in [0, 1] \; \forall u \in V, \; t \in D, \tag{10}$$

$$y_v \in [0, 1] \; \forall v \in V \tag{11}$$

Let the objective value of the optimal fractional solution $\{\bar{x}_{u,t}\}_{u,t}$ and $\{\bar{y}_v\}_v$ be OPT_f, and OPT_f must be greater than or equal to OPT.

3.2.1 LP deterministic threshold rounding

Because any optimal fraction solution $\{\bar{x}_{u,t}\}_{u,t}$ may not be an integer value, we need to transform $\{\bar{x}_{u,t}\}_{u,t}$ into integers. In this subsection, we simply introduce a threshold ts to determine whether a node should be vaccinated or not. It follows that

$$x_{u,t} = \begin{cases} 1 & \text{if } \bar{x}_{u,t} \geq ts; \\ 0 & \text{otherwise .} \end{cases}$$

Obviously, a higher threshold TS makes a higher $\bar{x}_{u,t}$ rounded to 1. The update mechanism of the threshold is that if the budget constraints are violated for any one time step, then we set the threshold higher to let fewer nodes be vaccinated; once the budget constraints are not violated for each time step, we immediately stop updating the threshold. As the threshold value *TS* gradually increases from the initial value, the number of nodes that will be vaccinated gradually decreases. If we still update the threshold, the consequent results cannot output better results. Let $\tilde{D}$ denote the total number of threshold values, the details are presented below.

Algorithm 1 LP Threshold Rounding Algorithm

```
Solve the LP above to get the optimal fractional solution $\{\bar{x}_{u,t}\}_{u,t}$, $\{\bar{y}_v\}_v$.
Set $TS = 1/\tilde{D}$ and $d = 1$.
while $1 \leq d \leq \tilde{D}$ do
    for $1 \leq t \leq T$ and $u \in V$ do
        if $\bar{x}_{u,t} \geq TS$ then
            Set $x_{u,t} = 1$.
        else
            Set $x_{u,t} = 0$.
        end if
    end for
    if The number of $x_{u,t}$ s.t. $x_{u,t} = 1$ is not greater than $B$ for each $t$ then
        Exit.
    else
        Set $TS = TS + 1/\tilde{D}$ and $d = d + 1$.
    end if
end while
Output the number of saved nodes, i.e., the corresponding $\sum_v y_v$.
```

Remark 1 A version with a dynamic binary search for a threshold *TS*, such that $|TS - T^*| \leq 1/\tilde{D}$ (instead of the current version with a linear search for such *TS*), would shorten the running time logarithmically.

Let $x(TS)$ be the number of vaccinated nodes with threshold *TS* subject to using only a single threshold. Because the MaxSave objective is submodular in the set of vaccinated nodes, it is *concave in the number of vaccinated nodes*. If the MaxSave function (of the number of vaccinated nodes) is L-Lipschitz for a constant L, and the number of vaccinated nodes is a ℓ-Lipschitz function of a threshold value for another constant ℓ, we can have an additive approximation as follows, which is proved in Appendix A.

Theorem 1 *Suppose that threshold T^* between 0 and 1 induces a number of vaccinated nodes feasibly over time that maximize the objective function f, which is L-Lipschitz in the number of vaccinated nodes x for a constant L, and x is a ℓ-Lipschitz function of a threshold value for another constant ℓ. The output feasible solution of Algorithm 1 is a $\tilde{D}/(L \cdot \ell)$-additive approximation to $f(x(T^*))$ for a constant $\tilde{D}$.*

Furthermore, if the MaxSave function is assumed to be *concave in the threshold value*, we can have an even better approximation ratio as follows, which is proved in Appendix B.

Theorem 2 *Suppose that threshold T^* between 0 and 1 induces a number of vaccinated nodes feasibly over time that maximize the objective function f, which is* *concave in the threshold value. The output feasible solution of Algorithm* 1 *is a* $(1 - 1/e)$*-multiplicative approximation to* $f(x(T^*))$.

3.2.2 LP randomized rounding algorithm 1 (repetition without substitution)

Now, we introduce one of the randomized rounding algorithms: the randomized rounding algorithm using repetition without substitution. The optimal fractional solution $\{\bar{x}_{u,t}\}_{u,t}$ cannot determine whether node u is vaccinated or not. Thus, we need to transform the optimal fractional solution $\{\bar{x}_{u,t}\}_{u,t}$ into 0 or 1 to form a vaccination strategy. In a randomized rounding algorithm, the fractional solution $\{\bar{x}_{u,t}\}_{u,t}$ will round to 0 or 1, and this gives the objective values whose optimal one is denoted as OPT_r, less than or equal to OPT_f. The following algorithm gives the details.

Algorithm 2 LP Randomized Rounding Algorithm 1 (Repetition Without Substitution)

```
Solve the LP above to get the optimal fractional solution $\{\bar{x}_{u,t}\}_{u,t}$, $\{\bar{y}_v\}_v$.
for $1 \le t \le T$ do
    $PA \leftarrow$ collect all the $\bar{x}_{u,t}$ with same $t$.
    repeat
        Select a node $u$ to vaccinate, i.e., set $x_{u,t} = 1$, with distribution $\{\frac{\bar{x}_{u,t}}{\sum_u \bar{x}_{u,t}}\}_u$ in $PA$.
        Remove the selected node $u$ from $PA$.
    until The process above is repeated for more than $B$ times
end for
Output the number of saved nodes, i.e., the corresponding $\sum_v y_v$.
```

Given the fractional optimal solution $\{\bar{x}_{u,t}\}_{u,t}$ and $\{\bar{y}_v\}_v$, at each time step t, we first collect all the nodes that are not infected or vaccinated and their corresponding fractional solution $\bar{x}_{u,t}$. Then we select one node u to vaccinate from a distribution. Finally we remove this node u because its state has been determined. In each time step, we select nodes B times to ensure that the budget is spent. Although we do not provide an approximation analysis for this intuitive method,[2] we present the numerical experiments in Sect. 4 to show its performance in terms of approximation ratios.

3.2.3 LP randomized rounding algorithm 2 (independence)

In the two previous proposed algorithms, if node u is chosen to be vaccinated, it will be removed from the candidate node set. However, in this algorithm, the nodes that have been vaccinated or infected will not be removed. We only determine whether node u needs to be vaccinated at time t based on its corresponding fractional solution. This ensures that the probability of each $x_{u,t}$ rounding to 1 is independent. After T time steps, the randomized rounding process produces a vaccination strategy. We evaluate whether this strategy satisfies Inequality (8) and Inequality (9). By repeating the process for $c \log nT$ rounds, we can select a

[2] It is still possible to conduct a careful analysis in a similar way to the approximation ratio analysis for the next algorithm, using conditional probabilities due to the dependency of variable rounding in this algorithm.

vaccination strategy that maximizes the number of saved nodes. The next theorem tells us that the LP randomized rounding algorithm 2 (Algorithm 3) could have a feasible solution that approximates the objective within a ratio $(1-\delta)$ with a high probability that decreases with δ.

Algorithm 3 LP Randomized Rounding Algorithm 2 (Independence)

```
Solve the LP above to get the optimal fractional solution $\{\bar{x}_{u,t}\}_{u,t}$, $\{\bar{y}_v\}_v$, and the optimal objective value $OPT_f$.
repeat
  for $1 \le t \le T$ do
    for $u \in V$ do
      Round $x_{u,t}$ to 1 with probability of $\bar{x}_{u,t}$.
      Round $y_v$ to 1 with probability of $\bar{y}_v$.
    end for
  end for
until The process above is repeated for more than $c \log nT$ rounds
Set $\{x_{u,t}\}_{u,t}$ and $\{y_v\}_v$ to the values when Inequality (8) and Inequality (9) are satisfied, choose a randomized
  rounding solution that maximizes $\sum_v y_v$, and output it.
```

Theorem 3 *With high probability, we can find a feasible solution such that the number of nodes saved by Algorithm* 3 *is at least* $(1-\delta)$ *times the optimal* (*integral*) *solution for some constant* $0 < \delta \le 1$.

Proof We would like that (i) $\sum_u x_{u,t} \le B$ for all t, and (ii) $1 \le \sum_u \sum_{t=1}^{d(s,v)-d(u,v)} x_{u,t}$ for all v.

For each t, in one of the $c \cdot \log nT$ rounds by Markov's inequality

$$\begin{aligned}\mathbf{Pr}\left[\sum_{u\in V} x_{u,t} \ge B+1\right] & \\ \le \frac{\mathbf{E}\left[\sum_{u\in V} x_{u,t}\right]}{B+1} = \frac{\sum_{u\in V}\mathbf{E}[x_{u,t}]}{B+1} = \frac{\sum_{u\in V}\bar{x}_{u,t}}{B+1} & \\ \le \frac{1}{1+1/B}. &\end{aligned} \tag{12}$$

For the $c \cdot \log nT$ times of the process, the probability that $\sum_u x_{u,t} \ge B+1$ every round is at most for some proper constant $c' > 0$

$$\left(\frac{1}{1+1/B}\right)^{c\cdot\log nT} \le \frac{1}{c'nT}. \tag{13}$$

By the union bound, we obtain the probability that event $\sum_u x_{u,t} \ge B+1$ happens for at least one time step among the total T time steps is at most $\frac{1}{c'n}$.

Let $D_v = \sum_{u\in V} \max\{d(s,v)-d(u,v), 0\}$ for all v. If $D_v = 0$, it means that the node v has been infected (since the summation of the right-hand side of (ii) above is 0). For node v, in one of the $c \cdot \log nT$ stages, the probability that node v will be saved is

$$\begin{aligned}&\mathbf{Pr}\left[1 \le \sum_{u\in V}\sum_{t=1}^{d(s,v)-d(u,v)} x_{u,t}\right]\\&\ge 1-\Pi_{u\in V,t:d(s,v)-d(u,v)>0}(1-\bar{x}_{u,t}) \ge 1-\left(1-\frac{1}{D_v}\right)^{D_v} \ge 1-\frac{1}{e},\end{aligned} \tag{14}$$

where the first inequality comes from the fact that the right side is the minimum of the left side. The probability that node v will be saved is equal to *the probability that at least one node u will be vaccinated before s spread to v*. The complement is that none of these nodes u would be vaccinated. Recall that $\bar{x}_{u,t}$ means the probability that node u will be vaccinated at t, and $1-\bar{x}_{u,t}$ means the probability that node u will not be vaccinated at t. Therefore, we obtain the first inequality. The maximum probability of the complement is when the probability is equal to $(1-\frac{1}{D_v})^{D_v}$; thus we obtain the second inequality. The last inequality is because $\left(1-\frac{1}{D_v}\right)^{D_v} \le \frac{1}{e}$.

For the $c \cdot \log nT$ times of the process, the probability that $1 > \sum_u \sum_{t=1}^{d(s,v)-d(u,v)} x_{u,t}$ every time is at most for some proper constant $c'' > 0$

$$\left(\frac{1}{e}\right)^{c\cdot\log nT} = (e^{-\log nT})^c = \left(\frac{1}{nT}\right)^c \le \frac{1}{c''nT}. \tag{15}$$

By the union bound, we obtain that the probability that at least one node will not be saved among the total n nodes is at most $\frac{1}{c''T}$.

Analysis of approximation ratio. For one pair of randomized rounding solutions that satisfies Inequality (8) and Inequality (9), we obtain the objective value $APPROX$, and it is easy to obtain κ such that $APPROX = (1-\kappa)OPT_f$. However, each pair of randomized rounding solutions that satisfies Inequality (8) and Inequality (9) has a different value of κ. Recall that we have $OPT_f \ge OPT \ge APPROX \ge 1$, where $OPT_f = \mathbf{E}[\sum_{vinV} y_v] = \sum_{vinV} \bar{y}_v$. For a given constant $0 < \delta \le 1$, we obtain the following inequality by the Chernoff bound

$$\mathbf{Pr}[APPROX \le (1-\delta)OPT_f] \le e^{-\frac{OPT_f\delta^2}{2}} \le \frac{1}{e^{\delta^2/2}}. \tag{16}$$

The inequality above implies that the probability that a vaccination strategy generated by a feasible randomized rounding solution can save fewer than $(1-\delta)OPT_f$ nodes is at most $\frac{1}{e^{\delta^2/2}}$. Because of $OPT_f \ge OPT$, we obtain the following two inequalities

$$\mathbf{Pr}[APPROX \le (1-\delta)OPT] \le \mathbf{Pr}[APPROX \le (1-\delta)OPT_f] \le \frac{1}{e^{\delta^2/2}},$$

and

$$\mathbf{Pr}[APPROX \ge (1-\delta)OPT] \ge \mathbf{Pr}[APPROX \ge (1-\delta)OPT_f] \ge 1-\frac{1}{e^{\delta^2/2}}.$$

These two inequalities show the probability that each vaccination strategy generated by a feasible randomized rounding solution satisfies the approximation ratio $1 - \delta$.

The probability that none of the vaccination strategies generated by a feasible randomized rounding solution can achieve a $(1 - \delta)$ approximation ratio is, at most, for some proper constant c''' that *depends on the choice of* δ,

$$\left(\frac{1}{e^{\delta^2/2}}\right)^{c \cdot \log nT} \leq \frac{1}{c'''nT}.$$

Therefore, the probability that at least one feasible strategy can achieve a $(1 - \delta)$ approximation ratio is at least

$$1 - \frac{1}{c'''nT}.$$

After these bad events, we have that the number of nodes saved by the algorithm's feasible solution is at least $(1 - \delta)OPT$ with probability of at least

$$\left(1 - \frac{1}{c'''nT}\right)\left(1 - \frac{1}{c''T}\right)\left(1 - \frac{1}{c'n}\right),$$

which can be lower bounded by

$$1 - \frac{1}{c'''nT} - \frac{1}{c''T} - \frac{1}{c'n}$$

using the union bound because the probability bound of the all-considered bad event takes the union of each, which can be upper bounded by the sum of them. □

4 Simulations and numerical results

In this section, we test the performance of the three proposed algorithms. The tested graph $G = (V, E)$ is created by the Stanford Network Analysis Project (SNAP) [14].[3] First, we use the mathematical programming solver Gurobi 9.5 to solve our original integer programming model and the corresponding relaxed linear programming model. Then we obtain the integer solution and the fractional solution with respect to the integer programming model and the linear programming model. Using the fractional solution, we test Algorithm 1, Algorithm 2, and Algorithm 3. All algorithms were implemented in Python 3.7.6, and the experiments were executed on a laptop with an Intel Core i5-8265U CPU1.60GHz 1.80 GHz and an x64 processor.

Note that there are some default mechanisms in Gurobi that will speed up the solving process of the IP solver, but we want to use only the branch-and-bound strategy to solve the IP in order to compare it with our rounding algorithms more fairly.

[3] The random graphs are generated by the random graph generator API provided by SNAP. They are GNP random graphs (Erdös-Renyi graphs).

Table 1 The performance comparison for different algorithm ($B = 2$)

Problem	Algorithm	Number of saved nodes	Running time (sec)
$(\|V\|, \|E\|) = (1000, 4500)$	IP	581	5.83
	Algorithm 1		
	Threshold = 0.4	436	17.69
	Algorithm 2	Avg = 540.8, max = 574	4.17
	Algorithm 3	565	10.29
$(\|V\|, \|E\|) = (1250, 4650)$	IP	899	44.39
	Algorithm 1		
	Threshold = 0.34	640	42.51
	Algorithm 2	Avg = 813, max = 864	7.25
	Algorithm 3	846	14.33
$(\|V\|, \|E\|) = (2200, 6000)$	IP	1571	142.56
	Algorithm 1		
	Threshold = 0.37	1317	219.55
	Algorithm 2	Avg = 1451.1, max = 1537	31.60
	Algorithm 3	1454	84

The Gurobi IP solver "presolve" the problem before solving the original problem, and the presolve mechanism will remove some constraints and variable bounds. Heuristic algorithms provided by Gurobi affect the results, too. Moreover, the Gurobi MIP solver runs in parallel with multiple threads and the "planes-cutting" strategy affect the result of MIP solver, too. Thus, we modify some parameters and make sure that we can solve the problem with 1 thread with no presolve and without any heuristics and planes-cutting strategy.

4.1 Random graphs

When we set the vaccination budget $B = 2$ at each time step, Table 1 presents the results of the original integer program, Algorithm 1, Algorithm 2, and Algorithm 3 for different graph scales. An observation is that, although these three algorithms cannot produce a strategy that is very similar to the vaccination strategy of IP, they can also perform well. Algorithm 2 performs relatively well, and its number of saved nodes is quite close to that of IP, and the running time may be also faster than that of IP. Algorithm 1 has a relatively poor performance likely because the threshold has not been set well enough to approximate the optimal one. Finding an approximately optimal threshold requires considerable effort, but the closer an updated threshold is to the optimal one, the better the performance of the algorithm is. That is, the larger value of $\tilde{D}$ is (for instance, $\tilde{D} = 0.01$ in the simulations), the smaller the approximation error to the (single) optimal threshold value is so there is a tradeoff between the performance in terms of approximation and the running time linearly or logarithmically depending on $\tilde{D}$.

We also supplement the results with some simulations that only allow the usage of limited memory in Gurobi's computation (parameterized by its memory limit).

Table 2 The performance comparison for different algorithm ($B = 3$)

Problem	Algorithm	Number of saved nodes	Running time (sec)
$(\|V\|, \|E\|) = (1000, 4500)$	IP	682	3.14
	Algorithm 1		
	Threshold = 0.34	675	42
	Algorithm 2	Avg = 642.9, max = 669	7.85
	Algorithm 3	646	19
$(\|V\|, \|E\|) = (1250, 4650)$	IP	1033	55.4
	Algorithm 1		
	Threshold = 0.36	673	40.64
	Algorithm 2	Avg = 940.4, max = 983	13.63
	Algorithm 3	977	19.73
$(\|V\|, \|E\|) = (2200, 6000)$	IP	1878	19.72
	Algorithm 1		
	Threshold = 0.43	1855	98.63
	Algorithm 2	Avg = 1751.8, max = 1843	23.79
	Algorithm 3	1782	29

For a graph $(|V|, |E|) = (2200, 6000)$ with $B = 3$, when the usage of limited memory is 0.3, the IP solver cannot obtain the solution due to being out of memory while the running time of our LP rounding algorithm is 38.29 sec; when the usage of limited memory is 0.5, the running time of the IP solver and the LP rounding algorithm are 52.69 and 36.92 sec, respectively. For a larger graph $(|V|, |E|) = (3250, 8000)$ with $B = 3$, with the memory limit of 0.7, the running time of our LP rounding algorithm is 28.15 sec, but the IP solve cannot obtain the solution due to being out of memory. In summary, the advantage of our LP rounding algorithm compared with the IP solver is more obvious when the problem size is larger, but even when the problem size is relatively smaller, the computation of the IP solver can run out of memory with stricter memory limitation.

In addition, more results for random graphs with larger budgets, i.e., $B = 3, 4$, are provided in Tables 2 and 3 for completeness before presenting the results for empirical networks. The tables of results for $B = 2, 3, 4$, each with 3 graphs in different scales, along with the results with memory limit control altogether show competitiveness of our rounding algorithms in terms of guaranteeing decent approximation ratios and computational time compared with the so-called "vanilla" IP solver. We observe that Algorithm 1 can still have a decent performance for some graph scales so we can say that a suitable threshold can improve the performance of Algorithm 1 significantly. The running time Algorithm 1 shows that it is not as efficient as the other two algorithms. The performance of Algorithm 3 becomes better when the size and budget of the graph become larger because the probability of a node being repeatedly vaccinated becomes smaller, making the probability of a node being vaccinated change only slightly compared to the optimal fractional solution. Although

Table 3 The performance comparison for different algorithm ($B = 4$)

Problem	Algorithm	Number of saved nodes	Running time (sec)
$(\|V\|, \|E\|) = (1000, 4500)$	IP	764	4.47
	Algorithm 1		
	Threshold = 0.5	344	16
	Algorithm 2	Avg = 690.5, max = 694	5.4
	Algorithm 3	701	10.54
$(\|V\|, \|E\|) = (1250, 4650)$	IP	1141	6.42
	Algorithm 1		
	Threshold = 0.41	1118	43.15
	Algorithm 2	avg = 1051.3, max = 1112	12
	Algorithm 3	1115	21.45
$(\|V\|, \|E\|) = (2200, 6000)$	IP	2098	5.74
	Algorithm 1		
	Threshold = 0.01	2098	291.56
	Algorithm 2	Avg = 1426.6, max = 1470	24.57
	Algorithm 3	2098	295.54

Table 4 Comparison of the number of saved nodes with IP and Algorithm 2 (B = 1)

	IP	Algorithm 2	Ratio
Facebook social network	973	973	1
Router network	1957	1957	1
Blue verified network	3885	3885	1

Algorithm 2 has relatively short running time among these three approximation algorithms, it does not performs well enough when the size and budget of the graph become large due to the fact that after each time step, it removes the selected nodes from the candidate set and recalculates the weight of each node in the candidate set. It uses this weight as the probability of vaccination and thus makes the probability of a node being vaccinated change a lot compared to the optimal fractional solution.

4.2 Empirical networks

We provide three experiments with the real networks retrieved from [16]. The first is the social network extracted from Facebook: the nodes are the people, and the edges are the connections with people. The second is the router network: the nodes are the routers, and the edges are the connection with routers. The third is Facebook pages with a blue verified network: the nodes are the pages, and the edges are the connections about how they like each other.

About the first LP Randomized Rounding Algorithm. We test the performance of Algorithm 2 with these three real networks when the vaccination budget is 1 and 2. The comparison of the saved nodes between IP and Algorithm 2 according to the

Table 5 Comparison of the number of saved nodes with IP and Algorithm 2 (B = 2)

	IP	Algorithm 2	Ratio
Facebook social network	1142	1107.4	0.97
Router network	2112	2112	1
Blue verified network	3889	3436.9	0.88

experiment is given in Tables 4 and 5. The number of saved nodes solved by Algorithm 2 is the average of ten rounding cycles. The ratio is the similarity between the number of saved nodes solved by IP and Algorithm 2.

5 Conclusions and future work

In this work, we used the exact algorithm and approximation algorithm to solve the problem proposed by [2]. We proposed a linear integer program to obtain the optimal solution. In addition, because of computing efficiency, we proposed one deterministic threshold rounding algorithm and two different LP randomized rounding algorithms. In terms of the objective value, our algorithms are approximations with polynomial running time while the optimal exact solution by the IP demands much longer running time when the problem size is larger due to the inscability nature of solving IPs. The analytical and numerical studies allow each individual to adopt the most appropriate approximation algorithm to efficiently resolve the vaccination problem when her reliance on commercial optimization solvers is costly.

For the first LP randomized rounding algorithm, the experimental results are given. For the deterministic threshold rounding algorithm, we gave a simple analysis. For the second LP randomized rounding algorithm, we gave an analysis that the algorithm will with a high probability find a feasible solution, and the number of nodes saved by it is at least $(1-\delta)$ times the optimal objective value with some constant $0<\delta\le 1$.

Although we simply focus on the vaccination strategies for a deterministic infection spreading model, we can nonetheless extend the model to consider a stochastic infection spreading process via stochastic (integer) programming like in [19] for influence maximization (in expectation). We may design better (randomized) rounding or other LP relaxation-based algorithms. We stated that our problem [2] has a submodularity property and can be solved by maximizing a monotone submodular function over a partition matroid constraint. In [19], the authors proposed a two-stage stochastic IP model, using the submodularity of the objective for the influence maximization problem and delayed constraint generation, to obtain the optimal solution faster. This framework may be used to tackle our problem as well.

A proof of theorem 1

From the definition of L-Lipschitzness, $f(x(T^*)) - f(x(T^* + 1/\tilde{D})) \leq L \cdot |x(T^*) - x(T^* + 1/\tilde{D})|$. We know that for $|x(T^*) - x(T^* + 1/\tilde{D})| > 0$

$$f(x(T^* + 1/\tilde{D})) \geq f(x(T^*)) - L \cdot |x(T^*) - x(T^* + 1/\tilde{D})|.$$

From the definition of ℓ-Lipschitzness, $x(T^*) - x(T^* + 1/\tilde{D}) \leq \ell \cdot |T^* - (T^* + 1/\tilde{D})|$. We have that $x(T^*) - x(T^* + 1/\tilde{D}) \leq \ell \cdot (1/\tilde{D})$. Thus,

$$f(x(T^* + 1/\tilde{D})) \geq f(x(T^*)) - L \cdot \ell \cdot (1/\tilde{D}).$$

B proof of theorem 2

Let TS_d be the threshold value at iteration d. Since $TS_{d+1} - TS_d = TS_d - TS_{d-1} = 1/\tilde{D}$ by the concavity of the MaxSave function f, we have that

$$f(x(TS_d)) - f(x(TS_{d-1})) \geq f(x(TS_{d+1})) - f(x(TS_d)) \geq 0.$$

Let $n = T^*\tilde{D}$ be the largest number of iterations executed before exiting the while loop. Then,

$$\begin{aligned} f(T^*) = f(TS_n) &= f(TS_{n-1}) + f(TS_n) - f(TS_{n-1}) \\ &\leq f(TS_{n-1}) + f(TS_{n-1}) - f(TS_{n-2}) \\ &= f(TS_{n-2}) + 2\left[f(TS_{n-1}) - f(TS_{n-2})\right] \\ &\leq f(TS_{n-3}) + 3\left[f(TS_{n-2}) - f(TS_{n-3})\right] \\ &\leq \cdots \\ &\leq f(TS_1) + (n-1)\left[f(TS_2) - f(TS_1)\right]. \end{aligned}$$

Replacing the constants $1, 2, 3, \cdots, n-1$ in these inequalities with n, we can see that

$$f(TS_n) \leq f(TS_{n-d}) + n\left[f(TS_{d+1}) - f(TS_d)\right].$$

It can be derived that

$$f(TS_{d+1}) \geq \frac{1}{n} f(TS_n) + \left(1 - \frac{1}{n}\right) f(TS_d).$$

It follows that

$$
\begin{aligned}
f(TS_{d+1}) &\geq \frac{1}{n}f(TS_n) + \left(1 - \frac{1}{n}\right)f(TS_d)\\
&\geq \frac{1}{n}f(TS_n) + \left(1 - \frac{1}{n}\right)\left[\frac{1}{n}f(TS_n) + \left(1 - \frac{1}{n}\right)f(TS_{d-1})\right]\\
&\geq \cdots\\
&\geq \frac{f(TS_n)}{n}\left(1 + \left(1 - \frac{1}{n}\right) + \left(1 - \frac{1}{n}\right)^2 + \cdots + \left(1 - \frac{1}{n}\right)^d\right) + f(TS_1)\\
&\geq \frac{\left(1 - \frac{1}{n}\right)^d}{n(1 - (1 - \frac{1}{n}))}f(TS_n)\\
&\geq \left(1 - \frac{1}{n}\right)^d f(TS_n)\\
&\geq \left(1 - \frac{1}{n}\right)^n f(TS_n)\\
&\geq \left(1 - \frac{1}{e}\right)f(TS_n).
\end{aligned}
$$

Note that $f(TS_1) \geq 0$ and $d \leq n$, and using $1 - x \leq e^{-x}$ for any $x \in \mathbb{R}$, we obtain the above inequality.

Acknowledgements This is supported in part by MST 108-2923-E-009-008-MY2.

Data availability The datasets analysed during the current study are available in the SNAP repository, http://snap.stanford.edu.

References

1. Adam, D.: Special report: the simulations driving the world's response to covid-19. Nature **580**(7802), 316–319 (2020)
2. Anshelevich, E., Chakrabarty, D., Hate, A., Swamy, C.: Approximability of the firefighter problem: Computing cuts over time. In: Proceedings of the 20th International Symposium on Algorithms and Computation, pp. 974–983 (2009)
3. Aspnes, J., Chang, K., Yampolskiy, A.: Inoculation strategies for victims of viruses and the sum-of-squares partition problem. J. Comput. Syst. Sci. **72**(6), 1077–1093 (2006)
4. Cai, L., Verbin, E., Yang, L.: Firefighting on trees: $(1 - 1/e)$-approximation, fixed parameter tractability and a subexponential algorithm. In: International Symposium on Algorithms and Computation, pp. 258–269. Springer (2008)
5. Calinescu, G., Chekuri, C., Pal, M., Vondrák, J.: Maximizing a monotone submodular function subject to a matroid constraint. SIAM J. Comput. **40**(6), 1740–1766 (2011)
6. Chen, P.-A., David, M., Kempe, D.: Better vaccination strategies for better people. In: Proceedings of the 11th ACM conference on Electronic commerce, pp. 179–188 (2010)
7. Cheng, C.-H., Kuo, Y.-H., Zhou, Z.: Outbreak minimization vs influence maximization: an optimization framework. BMC Med. Inf. Decis. Mak. **20**(1), 1–13 (2020)
8. Eisinger, D., Thulke, H.-H.: Spatial pattern formation facilitates eradication of infectious diseases. J. Appl. Ecol. **45**(2), 415 (2008)
9. Finbow, S., MacGillivray, G.: The firefighter problem: a survey of results, directions and questions. Australas. J Comb. **43**, 57–78 (2009)
10. Hartke, S.G.: Attempting to narrow the integrality gap for the firefighter problem on trees. In: Discrete Methods in Epidemiology, pp. 225–231 (2004)

11. Hethcote, H.W.: The mathematics of infectious diseases. SIAM Rev. **42**(4), 599–653 (2000)
12. Kermack, W..O.., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. Proc. Royal Soc. London Ser. A Contain. Papers Math. Phys. Character **115**(772), 700–721 (1927)
13. Leskovec, J., Andreas, K., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 420–429 (2007)
14. Leskovec, J., Sosič, R.: Snap: a general-purpose network analysis and graph-mining library. ACM Trans. Intell. Syst. Technol. **8**(1), 1 (2016)
15. MacGillivray, G., Wang, P.: On the firefighter problem. J. Combin. Math. Combin. Comput. **47**, 83–96 (2003)
16. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: AAAI (2015)
17. Vazirani, V.V.: Approximation algorithms. Springer, New York (2010)
18. Wolsey, L.A., Nemhauser, G.L.: Integer and combinatorial optimization. Wiley, New York (1988)
19. Wu, H.-H., Küçükyavuz, S.: A two-stage stochastic programming approach for influence maximization in social networks. Comput. Optim. Appl. **69**(3), 563–595 (2018)
20. Yao, S., Fan, N., Hu, J.: Modeling the spread of infectious diseases through influence maximization. Optim. Lett. **16**(5), 1563–1586 (2022)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Yongge Yang[1] · Po-An Chen[2] · Yu-Ching Lee[1] · Yung-Yan Fanchiang[2]

Yongge Yang
s106034466@m106.nthu.edu.tw

Yung-Yan Fanchiang
dennisyes25@gmail.com

[1] Department of Industrial Engineering and Engineering Management, National Tsing Hua University, 101, Section 2, Kuang-Fu Road, Hsinchu City 300044, Taiwan

[2] Institute of Information Management, National Yang Ming Chiao Tung University, 1001 University Road, Hsinchu City 300010, Taiwan