Minimizing Infection in a Topology using Mathematical Programming*

Wesly C. Ataide[1], **Pedro B. Castellucci**[1], **Álvaro J. P. Franco**[1], **Rafael de Santiago**[1]

[1]Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

(wesly.ataide), pedro.castellucci, alvaro.junio, r.santiago@(grad.)ufsc.br

Abstract. *Minimizing the spread of an infection is a key strategy for controlling an epidemic. In urban regions, the spread can be reduced by restricting movement between adjacent districts. We propose a non-linear integer programming model to minimize the number of infected people in a network. The network considers the relations between districts and the movement of people in a city. We develop methods to build and improve solutions. The model and the methods are evaluated on an instance based on real data. Finally, the results indicate how challenging the problem is from an optimization and numeric perspective.*

1. Introduction

After the first cases of COVID-19 were recorded in December 2019 in China, the infection rapidly spread, posing a great danger to healthcare systems and the world economy. The measures adopted by many countries in response consisted of limiting the operation of establishments and encouraging their citizens to stay at home, aiming to reduce contact between individuals and, thus, slow contagion. With these actions, however, comes the question of when is it safe to reopen, which places may be reopened, and what activities may be allowed [Chang et al. 2021]. To answer these questions, epidemiological models have become crucial. With a large volume of data, they can identify the most vulnerable socioeconomic groups, point classes of establishments most likely to spread the infection, and find patterns to maximize the effectiveness of restrictions [Chang et al. 2021] . Measures that reduce the infection are important instruments in the fight against COVID-19, as they both reduce the number of infected people and help guarantee treatment for people who eventually do become infected [Chang et al. 2021]. As a result, to predict numbers such as individuals infected by a disease has become fundamental to reduce the negative impacts of the pandemic [Liu et al. 2020]. In this work we present a non-linear integer model of an epidemiological problem where the goal is to minimize the number of infected people over time in a connected graph. Moreover, we use a heuristic to find a feasible solution and an optimization tool over a real instance to analyze the first results.

2. Model

Our model of the problem, based on [Franco 2021], consists of minimizing the infection in a network with neighborhoods interconnected by roads, represented by a graph in which the neighborhoods are the vertices (with an associated population) and each road between neighborhoods i and j is a pair of arcs (i, j) and (j, i). The epidemiological model chosen

*This work was supported by Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina.

was the SIR model, which divides the population into three groups: susceptible, which can become infected; infected, who can transmit; and recovered, which do not transmit and cannot be infected. Taking this into account, the population of each vertex i for each instant t consists of the number of susceptible S_i^t, infected I_i^t and recovered R_i^t. This choice was made before the confirmation that certain COVID variants are capable of reinfecting. Besides, any other epidemiological model that consider reinfection, such as SIS, could be assumed. Movement between vertices is also modeled, but it is restricted so that people from vertex i can only visit adjacent vertices of i. We also define for each arc (i, j) the values $\beta_{i \to j}$, that represent the percentage of people residing at vertex i and visiting vertex j, i.e., that "use the road (i, j)". Also, a binary variable $x_{i \to j}$ is associated to each arc (i, j). We constrained $x_{i \to j}$ to have the same value as $x_{j \to i}$. The value in the variable $x_{i \to j}$ indicates that the traffic is blocked if $x_{i \to j} = 0$ or free if $x_{i \to j} = 1$. It is relevant to note that this construction makes the graph more easily analyzed as a directed graph from the perspective of the β values and as an undirected graph from the perspective of the x variables. Thus, when we refer to a tree or connected components in this graph, we are analyzing it in an undirected way. Blocked and free streets create a variation in the number of people at each of the vertices, affecting contagion. In our model, people circulating at each vertex i are divided into two groups: those who came from a neighboring vertex i (from $\chi(i)$) and those who live and have stayed in i. For the first group, for each neighboring vertex $j \in \chi(i)$, the number of people coming to i obeys the following formulas: $S_{j \to i}^t = \beta_{j \to i} x_{j \to i}^t S_j^t$ [1]. The second group, on the other hand, consists of the people who did not leave the vertex i and, therefore, it is sufficient to subtract the percentage of people who left: $S_{i \to i}^t = (1 - \sum_{j \in \chi(i)} (\beta_{i \to j} x_{i \to j}^t)) S_i^t$. It should be noted that if a road is blocked, the percentage of people who intended to use it is added to the vertex where they live. Thus, the number of people of each category circulating in i is given by: $\dot{S}_i^t = S_{i \to i}^t + \sum_{j \in \chi(i)} S_{j \to i}^t$. The goal is to minimize the number of infected people over time. It is necessary to consider for each vertex and instant the number of people who recovered and the number of newly infected people, which is a ratio of the number of encounters between susceptible and infected people: $S_i^{t+1} = S_i^t - v \mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t)$; $I_i^{t+1} = I_i^t - \xi I_i^t + v \mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t)$; $R_i^{t+1} = R_i^t + \xi I_i^t$. The constants v and ξ are the virulence and the recovery ratio, respectively. As the percentage of infected people circulating in a vertex i is $\dfrac{\dot{I}_i^t}{\dot{S}_i^t + \dot{I}_i^t + \dot{R}_i^t}$, we take this value as the proportion of the susceptible population who have an encounter with an infected person at this vertex. The expected number of encounters for the population of vertex i (and where the non-linearity lies in our model) is therefore

$$\mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t) = \frac{S_{i \to i} \dot{I}_i^t}{\dot{S}_i^t + \dot{I}_i^t + \dot{R}_i^t} + \textstyle\sum_{j \in \chi(i)} \dfrac{S_{i \to j} \dot{I}_j^t}{\dot{S}_j^t + \dot{I}_j^t + \dot{R}_j^t}.$$

We have a limitation on how we choose to block the streets, however: we must ensure that it is possible to access all neighborhoods, that is, the subgraph induced by the free streets must be connected. We start to obtain this by adding the following constraints: $\sum_{(i,j) \in E} f_{i \to j} + 1 = \sum_{(j,i) \in E} f_{j \to i}$ for each vertex $i \neq s$; and $\sum_{(s,j) \in E} f_{s \to j} = n - 1$, where $f_{i \to j}$ represents an integer flow through the arc (i, j), and n is the number of vertices in

[1] To get the other two formulas for the first group, corresponding to $I_{j \to i}^t$ and $R_{j \to i}^t$, simply replace S by I or by R in the last formula. We will follow this pattern throughout this section.

the graph. These constraints ensure over f, an s-branching flow, with some source vertex s (see e.g. [Carvalho et al. 2020]). Note that each s-branching flow induces a spanning tree in the graph when we take as edges every pair of arcs (i, j) where $f_{i\to j} > 0$ or $f_{j\to i} > 0$. Since our arcs come in pairs, the graph structure is essentially maintained. The vertex s can be chosen arbitrarily, as a consequence of the following assertion: *Let G be a connected graph, and T a spanning tree of G. For any vertex v in T, there is an v-branching flow that induces T.* To close the connectivity constraints with the feasible solutions and to integrate it with contagion, we added restrictions so that a pair of arcs is not blocked when there is flow over it: for every arc (i, j), we have $nx_{i\to j} \geq f_{i\to j}$, ($n$ is the total number of vertices). Notice that this does not imply that the obtained (possibly optimal) graph has to be a spanning tree, only that it is connected. To summarize, the non-linear optimization program desires to minimize $\sum_{i\in V} I_i^{t+1}$, where V is the set of vertices of the input graph, subject to the circulation people constraints ($\dot{S}_i^t$, $\dot{I}_i^t$, $\dot{R}_i^t$ for each vertex i) and the connectivity graph constraints. The non-linearity is in the objective function and an important question is to know whether it is possible to turn this model linear. An answer for this question is not in the scope of this work.

3. A starting solution with iterated search

To give the solver an upper bound, we provide an initial spanning tree to the software and calculate the number of infected people considering that tree. In addition, we try to find spanning trees that decrease the number of infected people using an iterative search process that we will describe next. First, since T is a spanning tree of a graph G, let us define an adjacent spanning tree of T as a spanning tree that differs by only one edge. Then we proceed as follows. We start with any spanning tree T. We search among the set of all adjacent spanning trees of T and choose the tree T' that minimizes the number of infected people. If the infected value of T' is less than that of T, T' takes the place of T and we repeat the procedure. Otherwise, we stop. We can repeat this process for a given number of iterations, defining several local minima. Naturally, we only keep what leads to the least number of infected people. The specific process of searching through all of the adjacent trees deserves more detail. Given a spanning tree T, we choose an edge of the graph that is not in T and add it to T. This forms a cycle (T plus the inserted edge). For each edge in this cycle, we remove it from the tree and recalculate the number of infections, saving the tree that minimizes it. We repeat it, choosing another edge until all edges that were not in the original tree have been chosen.

4. Computational results

To evaluate the solution method and the model, we build an instance based on real data from the city of Florianópolis (SC - Brazil). The data was collected from *Covidômetro*[2] and the 2010 census, according to the IBGE database. *Covidômetro* contains data regarding confirmed, active, recovered and death cases related to COVID-19 of each city district. The IBGE database was used to define the population of each district. For the few inconsistent cases of district definition, we grouped districts based on proximity and estimated population based on current active cases. To solve the optimization problem we use the SCIP solver [Bestuzheva et al. 2021], through JuMP [Dunning et al. 2017]. SCIP is one

[2] https://covidometrofloripa.com.br/

of the best performing open source solvers and has an free academic license. It also allows the non-linear constraints present in our model. As an initial upper bound, we provide a spanning tree, as described in Section 3. The complete implementation can be found in `https://github.com/weslyca/TCC`. All the experiments were performed on an Aspire ES1-572 Acer Notebook, with an Intel(R) Core(TM) i3-6006u, 2.00GHz and 4GB of RAM. Regarding the software, we use Python 3.9.9 and Julia 1.7.0 with JuMP 0.22.1. Considering a 16 regions graph, the best value obtained was 622.238778 (14.24% of gap after of 4×10^6 branching nodes); however, when we calculate the number of infected people, we obtain 622.287136, which is different and slightly higher than the best tree returned by the heuristic. This difference indicates a potentially challenging model from a numerical perspective that we decide to report here. This effect could be reduced by tuning some parameters of the solver. However, there is a trade-off between numerical precision and solution time which should be further evaluated.

5. Conclusion

In this work, we expanded a model of contagion in a network of neighborhoods. We developed a formulation of the model as a non-linear programming problem in order to find an optimal choice of roads to block. The results illustrate the difficulty in dealing with the complexity of the problem and with numerical issues. The model could be modified to account for reinfections by adopting a corresponding epidemiological model. To finish, the circulation of people in the present model is restricted to the adjacent vertices. We invite the reader to think in a similar model that accepts walks of any size.

References

Bestuzheva, K., Besançon, M., Chen, W.-K., Chmiela, A., Donkiewicz, T., van Doornmalen, J., Eifler, L., Gaul, O., Gamrath, G., Gleixner, A., Gottwald, L., Graczyk, C., Halbig, K., Hoen, A., Hojny, C., van der Hulst, R., Koch, T., Lübbecke, M., Maher, S. J., Matter, F., Mühmer, E., Müller, B., Pfetsch, M. E., Rehfeldt, D., Schlein, S., Schlösser, F., Serrano, F., Shinano, Y., Sofranac, B., Turner, M., Vigerske, S., Wegscheider, F., Wellner, P., Weninger, D., and Witzig, J. (2021). The SCIP Optimization Suite 8.0. Technical report, Optimization Online.

Carvalho, C., Costa, J., Sales, C. L., Lopes, R., Maia de Oliveira, A. K., and Nisse, N. (2020). On the characterization of networks with multiple arc-disjoint branching flows. Research report, UFC ; INRIA ; CNRS ; Université Côte d'Azur ; I3S ; LIRMM ; Université de Montpellier.

Chang, S., Pierson, E., Koh, P. W., Gerardin, J., Redbird, B., and Grusky, D. Leskovec, J. (2021). Mobility network models of covid-19 explain inequities and inform reopening. *Nature*, 589:82–87.

Dunning, I., Huchette, J., and Lubin, M. (2017). Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320.

Franco, Á. J. P. (2021). Epidemic model with restricted circulation and social distancing on some network topologies. In *Cellular Automata*, pages 261–264.

Liu, C., Wu, X., Niu, R., Wu, X., and Fan, R. (2020). A new sair model on complex networks for analysing the 2019 novel coronavirus (covid-19). *Nonlinear Dynamics*, 101(3):1777–1787.