Robot dance: a region-wise automatic control of Covid-19 mitigation levels

Luís Gustavo Nonato, Tiago Pereira, Claudia Sagastizábal,
Paulo J. S. Silva (speaker)

June, 2020

IMECC

UNICAMP

CeMEAI
CEPID - Centro de Ciências Matemáticas Aplicadas à Indústria

Outline

Outline

Mitigation

Initial Covid-19 mitigation in the state o São Paulo are based on social distancing protocols:

- Simultaneous implementation: all state has started the social distancing protocol at the same moment ignoring the specific stage of each city.
- Long implementation: strict social distancing protocols must implemented over many months.
- Homogeneity: the protocol does not take into account economic roles and health infrastructure of each city.

New protocol

From June 1st on a new protocol is being adopted:

- Use criteria based on: Health care/ICU capacity, evolution of the number of infected.
- Each region of the state can be in a different situation.
- Uses different protocols that allow different sectors of the economy to open (partially) depending on the stage.
- The state should improve testing and data collection.
- Essentially reactive.

What we want to achieve

We developed a optimal control framework that:

- Takes into account intercity commute and health structure.
- Orchestrate the control among the cities to avoid all of them to stop at the same time.
- Flexible to allow for multiple scenarios: city versus state wide poll of intensive care units, short mitigation protocols, alternate between stricter and looser periods, etc.
- **Depends on good data!**

Outline

1. Mitigation strategy
2. SEIR model with commuting
3. Optimization
4. Case studies

SEIR with commute among the cities

We consider a set of K cities with fixed population

$$S_i + E_i + I_i + R_i = 1, \quad i = 1, \dots, K.$$

We divide the susceptible population of city i into classes as

S_{ij} = Fraction of the pop. of city i that leave it to work in city j.

Similarly for the other states E, I, and R.

For simplicity, we consider that infected people are sick and do not travel.

During the day

During the day the dynamics of susceptible from city i is affected by the other cities:

$$\frac{dS_i}{dt} = -\frac{1}{T_{\mathrm{inf}}} \sum_{j=1}^{K} \frac{r_j(t)}{P_j^{\mathrm{eff}}} S_{ij} I_{jj}, \tag{1}$$

where P_i^{eff} represents the actual fraction of the population at city i that changes due to inflow and out flow.

During the night and population splitting

During the night workers go back to their city and the usual dynamics is used.

The fraction of the population S_{ij} can be estimated using a matrix that describes the mobility between the cities that has at each entry

$p_{ij}(t) = \#$ daily accumulated percentage of inhabitants of i that travels

Therefore,

$$S_{ij} = S_i p_{ij}.$$

Final model

$$
\begin{aligned}
\frac{dS_i}{dt} &= -\alpha(t)\left(\frac{S_i}{T_{\text{inf}}}\sum_{j=1}^{N}\frac{r_j(t)}{P_j^{\text{eff}}}p_{ij}I_{jj}\right) - (1-\alpha(t))\left(\frac{r_i(t)}{T_{\text{inf}}}S_iI_i\right)\\
\frac{dE_i}{dt} &= \alpha(t)\left(\frac{S_i}{T_{\text{inf}}}\sum_{j=1}^{N}\frac{r_i(t)}{P_j^{\text{eff}}}p_{ij}I_{jj}\right) + (1-\alpha(t))\left(\frac{r_i(t)}{T_{\text{inf}}}S_iI_i\right) - \frac{1}{T_{\text{inc}}}E_n\\
\frac{dI_i}{dt} &= \frac{1}{T_{\text{inc}}}E_n - \frac{1}{T_{\text{inf}}}I_n\\
\frac{dR_i}{dt} &= \frac{1}{T_{\text{inf}}}I_n.
\end{aligned}
$$

Here α is a square pulse. It is 1 for $t \in [8am, 6pm]$ and 0 otherwise.

Outline

Control and optimization

We formulate a control problem that tries to drive the SEIR dynamics using $r_i(t)$ as controls. The problem is discretized arriving at

$$\begin{aligned} \min \quad & f(S, E, I, R, r) \\ \text{s.t.} \quad & (S, E, I, R) \in DM(r) \\ & (S, E, I, R, r) \in OC \\ & S_{i,t} + E_{i,t} + I_{i,r} + R_{i,t} = 1, \;\; i = 1, \ldots, K, \;\; t = 1, \ldots, T \\ & (S, E, I, R) \in [0, 1]^{4K}, 0 \leq r \leq r_0, \end{aligned}$$

$DM(r)$ represents the discretized version of the SEIR dynamics. OC are operational constraints.

Constraints

The SEIR dynamics was discretized using a 2nd order explicit Runge-Kutta method (modified Euler) with daily steps.

The operational constraints can be:

Initial conditions : fix the initial conditions for the SEIR model.

Constant controls in a time window: it is not possible to change the mitigation strategy daily.

Health care constraints: take into account the number of intensive care units by city or in pools.

Design mitigation profiles

We use three conflicting objective terms (minimization):

"Normal life":

$$\sum_{\substack{c=1,\ldots,n \\ t=1,\ldots,T}} w_t(r_0 - r_t).$$

Alternate in time : Change controls among time windows.

Alternate in space : Keep part of the state open for business:

$$-\sum_{\substack{t=1,\ldots,T \\ i=1,\ldots K \\ j>i}} D(i,j,t)(r_{i,t} - r_{j,t})^2.$$

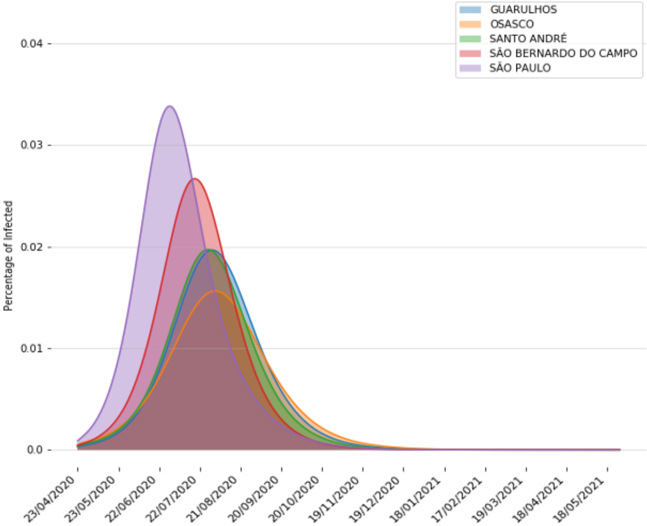
D takes into account the economic relations among regions.

Outline

Case studies

- Initial example with 11 of the largest cities in the state.
- Current r_0 is around 1.5, with social distancing measures in place till May 31st.
- Some of the cities have a very strong relation (metropolitan area of São Paulo), with movements of high proportions of the population among them.
- *Very optimistic* estimate considers that the health care system in São Paulo can sustain 1.5% of the population sick, while the number is 0.7% in the interior of the state.
- Constant controls in 14 days window.
- Implementation in Julia / JuMP to allow for rapid prototyping. Ipopt is the solver.

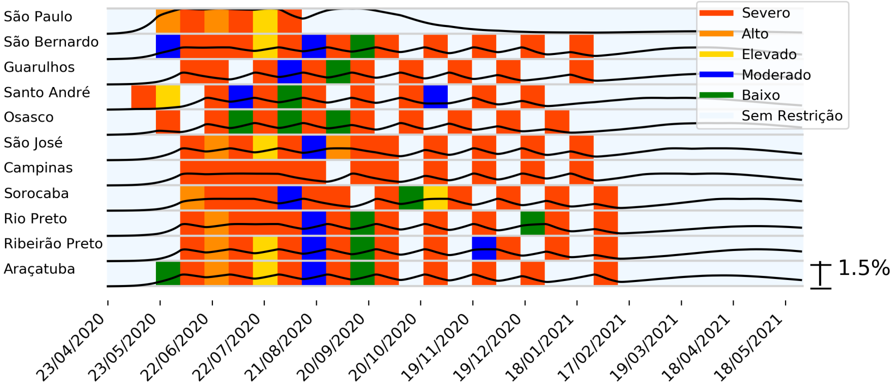
Current situation - offical data



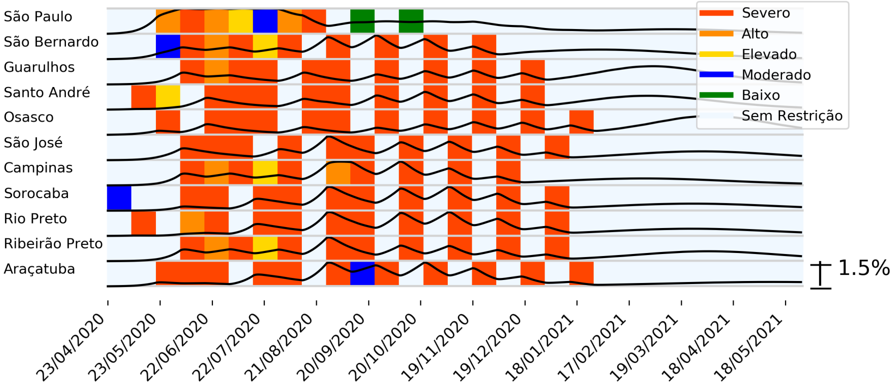
r_t to levels

r_t	Level
1.8	Low
1.6	Guarded
1.4	Elevated
1.2	High
0.8	Severe

Current vs control - São Paulo



São Paulo helps other cities



Conclusions

- We adapted the SEIR model to take into account daily commuting.
- We implemented a framework for rapid prototyping of mitigation controls. It is predictive not reactive.
- The mitigation will last for long, we have to learn to live with it.
- But we can alternate between more severe / less severe and among cities to keep part of the state open.
- The framework is flexible and allow us to “quickly” test other operational constraints and/or objectives.

To do

- Current code can cope with tens of cities/regions (lots of memory).
- Get rid of state variables that are not necessary.
- Use a sliding horizon or some other technique to allow for shorter simulations.
- Add stochasticity to model error on the data.

Report

Code