Multi-objective Optimal Control of a Pandemic Model for Covid-19 Management

Master-Thesis

submitted by
Jens Gebert

at

Universität Konstanz

Department of Mathematics and Statistics

Supervisor and Reviewer: Prof. Dr. Stefan Volkwein
Konstanz, 07.03.2022

Table of Contents

Introduction

Decision makers face the need of making difficult decisions with no clear, favorable path on a regular basis. This is illustrated especially impressive since the start of the Covid-19 pandemic in the beginning of 2020. These decisions consist often of different, contradictory objectives. Therefore there is a need for easy and fast applicable tools to find optimal solutions.
A possible way to decide multi-objective problems efficiently and scientifically is presented by this work, exemplary on the use of different countermeasures in the pandemic management in Berlin, Germany, in November 2020. First a model of ordinary differential equations (ODE) is introduced to simulate the pandemic proceeding. This model is then run for different countermeasures and a regression with respect to these is made. Consequently it is possible to forecast the pandemic behaviour in dependency of the use of countermeasures. In a second step a variance of the subdivision algorithm is introduced, combining the approaches from [Del04] and [Ser06] as appropriate tool to find optimal solutions for contradictory objectives with respect to the use of countermeasures. This tool is then applied to an exemplary objective function in order to illustrate the use of the subdivision algorithm and how decision making can be based upon its findings.
This work is highly influenced by the work of Wulkow et al. [Wul21], as it recreates their work mostly. However, it delivers more mathematical background for the underlying mechanism as well as a more detailed view of how to apply the different tools correctly and can therefore be understood as an extended instruction to apply multi-objective optimization to practical decision making.

1 The SIR model based on [Wul21]

Wulkow et al. [Wul21] studied, how to find an optimal set of countermeasures against the COVID-19 pandemic, despite lack of information about the future proceeding. Therefore they run an agent-based model (ABM), allowing accurate predictions via a realistic combination of person-centric data-driven human mobility and behaviour, stochastic infection models and disease progression models including micro-level inclusion of governmental intervention strategies [Wul21]. The output data is then applied to a macro model, which makes it possible to calculate a proceeding of the pandemic for a long period and a great amount of people. This macro model consists of a system of ODEs which is fitted to the ABM data and solved by a simple explicit one-step method. This work makes use of their ODE system and the found parameters.

1.1 The general SIR model

A common way to simulate the pandemic at a macro level is based on SIR models. This macro level can for example be the pandemic behaviour for the entire population of a country. For this purpose, the population is divided into different compartments and linked by a system of ODEs. The simplest of these models consists of the groups susceptible to the virus (S), infected (I), and recovered (R), thus the name SIR model. For any time and the total population N, $N = S(t) + I(t) + R(t)$ is valid. For the sake of simplicity, from now on the time dependency will not be written out explicitly.
Let $k_I > 0$ be the infection rate, i.e., the probability of moving from compartment S to compartment I. This rate is not an inherent quantity of the virus, but is an estimated parameter and covers also the average amount of people met during a day. In addition, the probability that a person is infected at time t is $\frac{I(t)}{N}$. Hence, the probability of infection for a susceptible person at time t is $k_I \frac{I(t)}{N}$. Furthermore, infected individuals

recover at a rate of $k_R > 0$. Overall, the system results in

$$\begin{aligned} \dot{S}(t) &= -k_I \frac{I(t)}{N} \\ \dot{I}(t) &= k_I \frac{I(t)}{N} \\ \dot{R}(t) &= k_R I(t). \end{aligned} \tag{1.1.1}$$

Obviously, this is a very crude model. There is a wide variety of models that include compartments such as fatalities, people in quarantine, and many others. For example in [Kim22] and [Gri21] the models are a lot more detailed, containing also compartments for different ages and differente grades of severeness of the infection. Theoretically, the model could be designed to be extremely accurate and multifaceted, but would then lose the practicality and necessary simplicity of a macro model.

1.2 The ODE system of [Wul21]

[Wul21] a more detailed model is used. In Figure 1.2.1 the corresponding model is visualized.

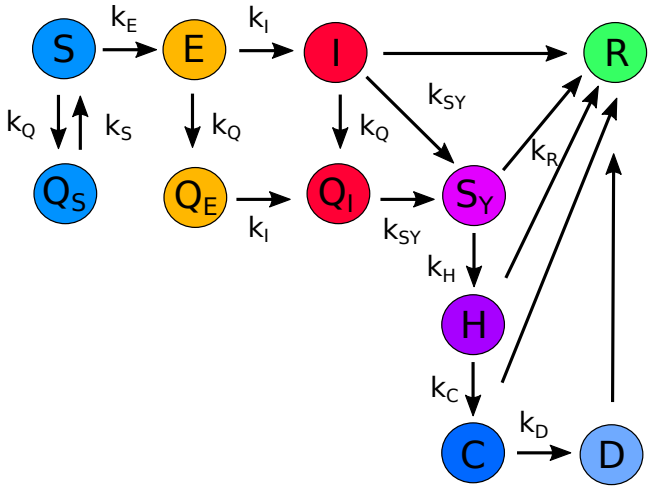

Figure 1.2.1: Compartments and switch rates according to the system of ODEs used in [Wul21].

To resolve the infection event, the following groups are introduced:

- (E) Those exposed to the virus, who are infected but not yet infectious themselves.
- (S_Y) The symptomatic, who are diseased and develop symptoms. People in I are therefore infectious but have not yet developed symptoms and hence are more often undetected.
- (H) The hospitalized, who are admitted to a hospital due to an infection.
- (C) The critically ill, who are in intensive care due to an infection.
- (D) The deceased, who have died due to an infection.

Quarantine interventions are implemented in many countries to keep the number of infected people low. People in quarantine may develop symptoms if infected, but may not infect anyone. Accordingly, the compartments Q_S, Q_E, and Q_I are added in the model. Healthy, susceptible people may go into self-isolation due to a moment of suspicion and switch to the compartment Q_S. After some time, it becomes clear that they are not ill and they return to the susceptible group. People who are exposed to the virus and go into quarantine (Q_E) become contagious there and then switch to Q_I. People who are already contagious and go into quarantine switch directly to Q_I. From both I and Q_I, people switch to either the recovered or the symptomatically infected. From here on, at each step, people can either recover or be hospitalized, enter the ICU, and ultimately die. The corresponding ODE system then follows accordingly.

$$
\begin{aligned}
\dot{S}(t) &= -k_E \frac{S(t) \cdot I(t)}{N} + k_S Q_S(t) - k_Q S(t) \\
\dot{E}(t) &= k_E \frac{S(t) \cdot I(t)}{N} - k_I E(t) - k_Q E(t) \\
\dot{I}(t) &= k_I E(t) - k_R I(t) - k_{SY} I(t) - k_Q I(t) \\
\dot{S_Y}(t) &= k_{SY} I(t) + k_{SY} Q_I(t) - k_R S_Y(t) - k_H S_Y(t) \\
\dot{H}(t) &= k_H S_Y(t) - k_R H(t) - k_C H(t) \\
\dot{C}(t) &= k_C H(t) - k_R C(t) - k_D C(t) \\
\dot{D}(t) &= k_D C(t) \\
\dot{R}(t) &= k_R I(t) + k_R S_Y(t) + k_R H(t) + k_R C(t) + k_R Q_I(t) \\
\dot{Q_S}(t) &= -k_S Q_S(t) + k_Q S(t) \\
\dot{Q_E}(t) &= -k_I Q_E(t) + k_Q E(t) \\
\dot{Q_I}(t) &= k_I Q_E(t) + k_Q I(t) - k_{SY} Q_I(t) - k_R Q_I(t)
\end{aligned} \tag{1.2.1}
$$

Whether individuals quarantine or self-isolate depends largely on the effectiveness of contact tracing. There is no compartment of detected cases in the model. However, it is

assumable, that all people with symptoms are detected, while people without symptoms in general are not. Therefore S_Y is used to indicate the detected cases. In case of a successful contact tracing, let nCT be the number of people who isolate per symptomatically infected (and therefore detected) person, CT the number of those who were tested positively and whose contacts are traced as well, and $k_q > 0$ be the rate of people isolating themselves after being informed because of successful contact tracing. Then the rate $k_Q(t) = k_q nCT\frac{CT(t)}{N}$, is obtained, showing the number of people moving from suspicion into self-isolation. It is assumed that a person's contacts are traced when that person is symptomatic and not already in quarantine, and that contact tracing ends when the person quarantines. This yields an additional differential equation.

$$\dot{CT}(t) = k_{SY} I(t) - k_q \frac{S(t) + E(t) + I(t)}{N} CT(t) \tag{1.2.2}$$

It has been shown in several studies ([Lin20], [Mül20], [Gri21]) that the effectiveness of contact tracing decreases dramatically when too many cases are recorded. To take account ofs this effect, an upper bound $CTMax$ is introduced for the change of CT. Thus, the input rate $k_{SY} I$ can never exceed $CTMax$. At last the model itself is written as

$$\dot{x}(t) = F(t, x(t)) \qquad x(0) = x_0 \tag{1.2.3}$$

with the state vector

$$x(t) = (S(t), E(t), I(t), S_Y(t), H(t), C(t), D(t), R(t), Q_S(t), Q_E(t), Q_I(t), CT(t))$$

at time t, such that $F(t, x(t))$ corresponds to the right side of the ODEs.

1.2.1 Existence and uniqueness

In order to find a solution for the system of ODEs 1.2.3, first existence and uniqueness of the solution has to be proved. Since the system of ODE consists only of linear combinations and multiplications of input variables, the right side of the system, F, is continuous derivable. The next step is to prove Lipschitz continuity.

Mean value theorem [Ran18]: For every continuous derivable function F with image in $\mathbb{R}^m$ and a, b with $\{a + t \cdot (b - a) | t \text{ with } t \in [0, 1]\} \subseteq \text{Def}(F)$ exists a set

$$(c_i)_{i \in \mathbb{N} \leq m} \subseteq \{a + t \cdot (b - a) | t \text{ with } t \in [0, 1]\}$$

such that

$$F(b) - F(a) = \nabla F_i(c_i) \cdot (b - a). \tag{1.2.4}$$

No compartment can exceed the total population N. Therefore the state vector x lies always in $[0, N]^{12}$ and is therefore bound. Hence it is possible to define $L = \max(\nabla F_i(c_i))$ as Lipschitz constant. Now it is possible to apply the theorem of Picard-Lindelöf.
Picard-Lindelöf theorem [Sch17]: Consider the initial value problem

$$\dot{x}(t) = F(t, x(t)), \quad x(t_0) = x_0$$

If F is (globally) Lipschitz continuous in x and continuous in t, then there exists a unique global solution $x(t)$.

From this theorem existence and uniqueness follow directly for the problem 1.2.3.

1.3 Parameter adjustment

In [Wul21] an ABM is used to determine the parameters for the ODE model. The ABM simulates the daily routines of individual persons (agents) and thus simulates pandemic behaviour. However, due to the large computational cost of the ABM, it is not practical to use it to make predictions over longer time periods. Instead, the ABM is used to simulate a small time period and then fit the ODE model to the numbers obtained. The ABM is simulated for various combinations of countermeasures. The different countermeasures are school closures, mask wearing and contact tracing. Since it is unrealistic to have 100% contact tracing, 100% is in reality representing 60% of successful contact tracing. All parameters are then fixed for the case where no measures are taken at all. For all other combinations of countermeasures, only k_E is then changed. This results in a model that forecasts the pandemic as a function of the infection rate k_E. The parameters for the case when no measures are taken at all are shown in Table 1.3.1 along with the initial values obtained from the ABM for the different compartments.

It should be noted that the ABM data does not match the real-life data. This is because the ABM simulation was created for a future scenario which is, due to political decisions, not applying to the real course anymore. However, this is not of crucial relevance, since this work is focusing on technical issues, rather than real-life observing and forecasting.

Parameters		Starting Values	
k_E	1.24	N	3574000
k_I	0.095	S	3481400
k_{SY}	0.611	E	10549
k_H	0.017	I	6540
k_C	0.176	S_Y	12845
k_R	0.045	H	944
k_D	0.258	C	484
k_S	0.193	D	179
k_q	0.193	Q_S	0
nCT	60	Q_E	0
$CTMax$	200	Q_I	0
		CT	0

Table 1.3.1: Estimated parameters of the ODE model to fit the ABM data for the case, when no action is taken, the population size N and the initial values obtained from the ABM. N is added to the starting values in order to give a compact overview over all parameters and values necessary to run the model.

The corresponding development to the start values of Table 1.3.1 is shown in Figure 1.3.1. The infection rate k_E depends on the different countermeasures. To establish a quantitative relationship, a regression model of the following form is used.

$$k_E(c) = \delta + \sum_{i=1}^{3} \alpha_i c_i + \sum_{i=1}^{3} \sum_{\substack{i=1 \\ j \geq 1}}^{3} \beta_{ij} c_i c_j \tag{1.3.1}$$

where $c = (c_1, c_2, c_3)$ are the countermeasures and α, β and δ are the adjustment factors. In [Wul21] it is shown, that a polynomial regression of order two provides satisfying results. The infection rate k_E values obtained by the ABM for the different combinations of countermeasures are shown in Table 3.0.1 in the appendix. Those values can then be used for the regression. The calculated k_E for the same countermeasures over k_E calculated by the ABM are shown in Figure 1.3.2. With the help of the ODE model, forecasts can now be made for the course of the pandemic depending on the countermeasures. In mathematical terms the state vector x has now a second dependency such that $x(t, c(t))$ describes the state of the different compartments in dependency of the time t and the current countermeasures c. In Figure 1.3.3 the strictest possible countermeasures ($c = (1, 1, 1)$) and no countermeasures at all ($c = (0, 0, 0)$) are plotted. This model is able to dissolve the impact of mask wearing, contact tracing and school closure. In [Gri21] it is outlined that the countermeasures gain effectiveness, if they are tailored to

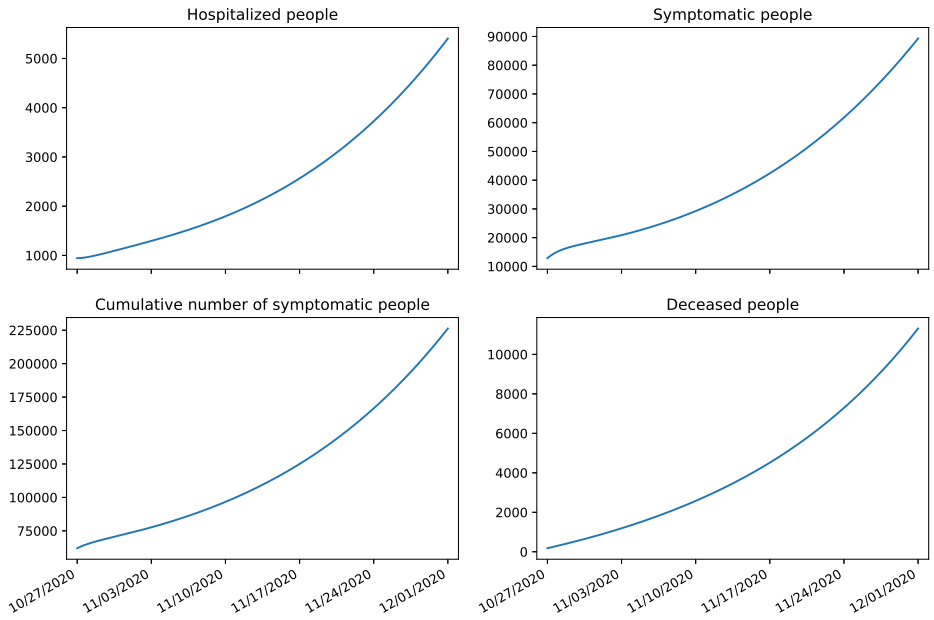

Figure 1.3.1: Development of the compartments 'hospitalized', 'symptomatic', 'deceased' and the cumulative number of the symptomatic infected people beginning at the 27th of October 2020 for the following five weeks. The start values are obtained from Table 1.3.1 which refers to the case of no countermeasures at all.

specific groups. The simpler approach introduced here works well for demonstrating the use of the multi-objective optimization strategy further down. However, to find the best possible pandemic management in real life, the use of tailored countermeasures would be necessary.

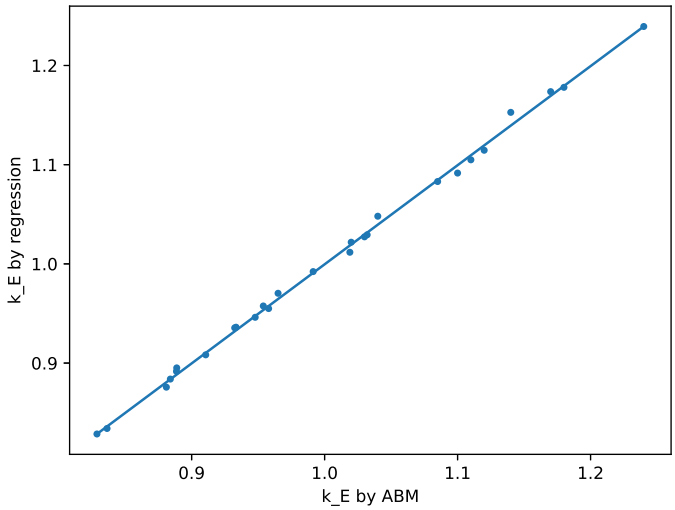

Figure 1.3.2: The k_E calculated via quadratic regression over the k_E calculated by the ABM, both with respect to the same countermeasures. A perfect fit with no inaccuracy would be characterized by all points forming a straight line as visualized.

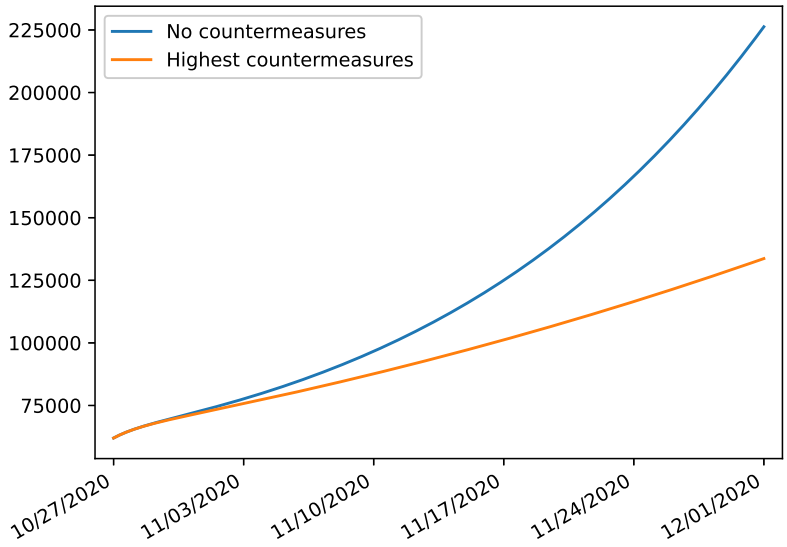

Figure 1.3.3: Cumulative number of symptomatic infectious people for the highest possible countermeasures and for no countermeasures at all.

1.4 One step methods

A simple approach to solve a system of ODEs can be made with explicit one step methods. They are all based on the same basic procedure. From a known point an estimation with respect to the ODEs and a certain step size is made and this new, estimated point is taken as starting point for the next step. A more extensive introduction to one step methods is made in [Bä20]. There one can also look up the proofs for convergence and convergence speed, which are here considered to be presumed.
The most simple method is the explicit Euler method (EM). Starting with a start point x_0, the next point is calculated via the first Taylor term. Let the distance between the new point x_{k+1} and x_k be h and F the right side of the ODEs as referred to in Section 1.2. Then the next iterate x_{k+1} is then calculated via

$$x_{k+1} = x_k + F(t_k, x_k)(x_{k+1} - x_k) = x_k + h \cdot F(t_k, x_k). \tag{1.4.1}$$

A more complex approach can be realized by the Runge-Kutta methods. In fact the EM is a Runge-Kutta method of first order. The methods of second and fourth order (RK2, RK4) are realized with additional calculations and therefore are more accurate, but also more expensive in terms of processing time.
The RK2 or also called Heun's method looks as follows:

$$\begin{aligned} K1 &= F(t_k, x_k) \\ K2 &= F(t_k + h, x_k + h \cdot K1) \\ x_{k+1} &= x_k + \frac{h}{2}(K1 + K2). \end{aligned} \tag{1.4.2}$$

The RK4 has two more terms:

$$\begin{aligned} K1 &= F(t_k, x_k) \\ K2 &= F(t_k + \frac{h}{2}, x_k + \frac{h}{2} \cdot K1) \\ K3 &= F(t_k + \frac{h}{2}, x_k + \frac{h}{2} \cdot K2) \\ K4 &= F(t_k + h, x_k + h \cdot K3) \\ x_{k+1} &= x_k + \frac{h}{6} \cdot (K1 + K4) + \frac{h}{3} \cdot (K2 + K3). \end{aligned} \tag{1.4.3}$$

The order of the methods is determined by their order of convergence. Let $y(t_i)$ be the exact solution of the ODEs at the time t_i and y_i an estimated solution. The estimated solution has order p of convergence when there is a scalar C such that

$$\max_{i=0,\dots,n} ||y_i - y_{t_i}|| \leq C \cdot h^p.$$

The EM has order of convergence one, RK2 order of two and RK4 order of four. This means: If the step size is cut in half, it would be expected that the maximal error of the EM is cut in half as well, while the maximal error of the RK4 is only a sixteenth part of what it used to be. In general the EM is used for solving a system of ODEs over a long time with many iterations and which are not too unstable with respect to errors. The RK2 and RK4 are used when small errors can lead to the loss of convergence or when higher precision is of great importance.

1.5 Comparison of different methods

As most SIR models are computed with the EM, the question arises what error is to be expected. Obviously the true future course of the pandemic is not known. For this reason a 'true' course is computed with the most accurate method and with a small step size. In this case this is the RK4 with 10000 iterations per 35 days. For every method and different step sizes the relative error in every step in respect to the 'true' course is shown in Figure 1.5.1.

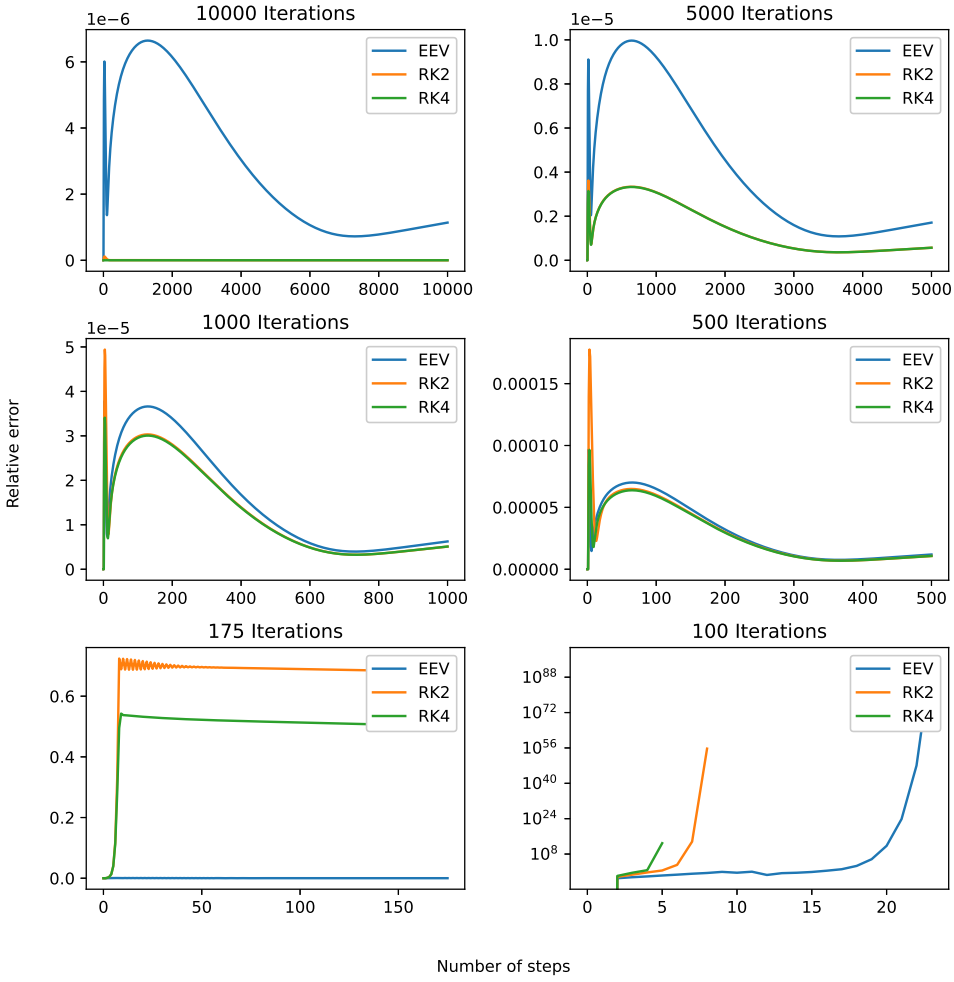

Figure 1.5.1: The relative error in every step between the proceedings and the 'true' case (RK4 with 10000 iterations), dissolved for every step and different step sizes.

For 10000 iterations the behaviour is in line with the expectations. The EM has the biggest relative error, while the error of RK2 is quite low and the error of the RK4 with respect to itself is of course zero. While the error for the EM seems to double itself for the doubled step size, the relative error of the other methods increases a lot faster. This is in line with the order of convergence of the methods, being of order 1 (EM), order 2 (RK2) and order 4 (RK4). While the methods are 'just' inaccurate for the big step size of 175 iterations per 35 days, the convergence is lost at 100 iterations.

Error \ Iterations	10000	5000	1000	500
EM	$2.1 \cdot 10^{-5}$	$4.3 \cdot 10^{-5}$	0.00015	0.00031
RK2	$5.1 \cdot 10^{-8}$	$1.1 \cdot 10^{-5}$	0.00010	0.00027
RK4	0	$1.0 \cdot 10^{-5}$	$9.8 \cdot 10^{-5}$	0.00025

Duration \ Iterations	10000	5000	1000	500
EM	1.021	0.49	0.12	0.050
RK2	1.93	0.98	0.20	0.11
RK4	3.81	1.92	0.39	0.20

Table 1.5.1: Above is shown the total relative error between the methods and the 'true' case (RK4 with 10000 iterations), calculated with the trapezoidal rule (1.5.1) for different step sizes. Below is shown the duration in seconds for the calculation of the different methods and step sizes.

It is possible to calculate the total relative error between the course of the different methods and the 'true' course after the time $T = t_n$. To calculate the occurring integral, the trapezoidal rule is used. Let again $y(t_i)$ be the exact solution and y_i be an estimation to it. Then the relative error δ_{rel} follows

$$\begin{aligned}\delta_{rel} &= \sqrt{\int_0^T \frac{||y_t - y(t)||_2^2}{||y(t)||_2^2}} \\ &\approx \sqrt{h}\sqrt{\frac{||y_0 - y(t_0)||_2^2}{2||y(t_0)||_2^2} + \sum_{i=1}^{n-1} \frac{||y_i - y(t_i)||_2^2}{||y(t_i)||_2^2} + \frac{||y_n - y(t_n)||_2^2}{2||y(t_n)||_2^2}}. \end{aligned} \tag{1.5.1}$$

The calculated errors for the more reasonable step sizes are shown in Table 1.5.1. As well as before it is possible to observe, that for a great amount of iterations the error of the RK4 is growing a lot faster than the error of the other two methods. However, while the error of the EM roughly doubles its size for the doubled step size, the relative error of the other two methods is almost the same for a lower amount of iterations, which is not as expected since the order of convergence is different. The reason therefore might be, that the order of convergence only delivers a maximal border, but obviously the error can be less. The order of convergence is also defined for the total error, while here the relative error is used.

Furthermore, the processing times for the different methods are shown in table 1.5.1. It is clearly visible that the RK2 and RK4 need more time for the same amount of itera-

tions, but calculate more accurate. The different methods provide different advantages and disadvantages and need to be picked on the basis of the task.
In this work the EM with 500 iterations is chosen. The error is still very small, the precision of the findings are not from utmost importance, since only the overall development is observed and the processing time is superior.
While these findings provide clarity about the expected error lying in the methods themselves, it is not clear, how uncertainties within the data affect the precision of the methods. This topic is investigated in [Kim22] via a sensitivity analysis. The findings there show that the robustness between the parameters varies, but is in general quite robust with respect to errors in the input data.

2 Optimal Countermeasures

It is clear that the fewest people would be infected under all encompassing measures. However, this would result in extreme high costs. Moreover, in terms of pandemic response, it might be prudent to keep the number of infected people at a certain level and drive contagion within a controlled level. With such a wide-ranging issue, there are often different interests that conflict with each other. Finding the optimal compromises for this is the subject of multi-objective optimization.

2.1 Pareto front

The following introduction to multi-objective optimality with respect to this multi-objective problem is already described in [Wul21]. However, for the sake of completeness it will be introduced here as well and enriched with some theoretical background. For a more extensive insight see [Ehr05].
Let $\mathcal{J}(x(t,c(t)),c(t)) = (J_1(x(t,c(t)),c(t)),...,J_n(x(t,c(t)),c(t)))^T$ be the desired objectives depending on the current numbers for the different compartments $x(t,c(t)) = (S(t,c(t)),E(t,c(t)),I(t,c(t)),...)$ and the countermeasures $c(t)$ at the time t. Let further be $\mathcal{C}$ the countermeasure space, consisting of all possible countermeasures, t_1 and t_2 be starting and ending time of the observation period and $c : [t_1,t_2] \to \mathcal{C}$ a countermeasure function, which describes the counter-measures $c(t) \in \mathcal{C}$ applied at the time $t \in [t_1,t_2]$. With introducing the terminal cost function $h_k(t)$ and the running cost function $g_k(t)$ with respect to J_k, it is possible to formulate the objective function

$$J_k(x(t,c(t)),c(t)) = \int_{t_1}^{t_2} g_k(x(t,c(t)),c(t))\mathrm{d}t + h_k(x(t_2,c(t_2)),c(t_2)),$$
$$k = 1,...,n. \tag{2.1.1}$$

To handle the dependencies easier a few arrangements are made. Since in this work only constant countermeasure functions are observed and the state vector $x(t)$ is only dissolved at the end time t_n of the observation period, only $\mathcal{J}(x(t_n,c(t_n)),c(t_n))$ is

observed. Therefore the time dependency can be suppressed, turning the $c(t_n)$ to c and therefore the objective function into $\mathcal{J}(x(c), c)$. Hence, $\mathcal{J}$ is only dependent from c and the function can be further reduced to $\mathcal{J}(c)$.
The optimal set of countermeasures is then defined as

$$\min_{c \in \mathcal{C}} \mathcal{J}(c). \tag{2.1.2}$$

Most often there exists not 'the' optimal set to minimize $\mathcal{J}(c)$, but rather a infinite amount of optimal sets. However, it is possible to find regions enclosing those optimal sets.
Since $\mathcal{J}$ is a n-dimensional vector, a new order relation has to be defined. Thus, the relation $\mathcal{J}(c) \leq_p \mathcal{J}(\tilde{c})$ is introduced, meaning $J_k(c) \leq J_k(\tilde{c})$ for every $k \in [1, ..., n]$. c is called Pareto optimal if there is no other scheme $\tilde{c} \in \mathcal{C}$ such that holds

$$\mathcal{J}(c) \neq \mathcal{J}(\tilde{c}) \quad \text{and} \quad \mathcal{J}(c) \leq_p \mathcal{J}(\tilde{c}). \tag{2.1.3}$$

The set of all Pareto optimal points is referred to as Pareto front.
In [Kuh51] a necessary condition for Pareto optimal points is given for solutions of restrained optimization problems. The restrains are defined by $\mathcal{C} = [0, 1]^3$. $\mathcal{C}$ can be described by

$$\begin{aligned} &\mathcal{C} = \{x \in \mathbb{R}^3 : e(x) = 0, \quad g(x) \leq 0\} \\ &g : \mathbb{R}^3 \to \mathbb{R}^6, \quad x \rightarrowtail \begin{pmatrix} x - 1 \\ -x \end{pmatrix} \\ &e(x) = 0. \end{aligned} \tag{2.1.4}$$

g and e are continuously derivable. Let furthermore $\mathcal{J}$ be continuous. It is necessary to prove this later on for the concrete objective functions. Then all the requirements are met to apply the KKT conditions.
KKT conditions [Kuh51]: Let $\overline{c} \in \mathcal{C}$ be a Pareto optimal point. Then there exist vectors

$$\begin{aligned} \overline{\alpha} \in \mathbb{R}^n_{\leq 0} \text{ mit } \sum_{i=1}^{n} \overline{\alpha}_i = 1, \\ \overline{\lambda} \in \mathbb{R}, \quad \overline{\mu} \in \mathbb{R}^6, \end{aligned}$$

such that,

$$\sum_{i=1}^{n} \overline{\alpha}_i \nabla J_i(\overline{c}) + \sum_{i=1}^{1} \overline{\lambda}_i \nabla e_i(\overline{c}) + \sum_{i=1}^{6} \overline{\mu}_i \nabla g_i(\overline{c}) = 0, \tag{2.1.5}$$

$$e(\overline{c}) = 0, \tag{2.1.6}$$

$$\overline{\mu} \geq 0, \quad g(\overline{c}) \leq 0, \quad \overline{\mu}^T g(\overline{c}) = 0. \tag{2.1.7}$$

To meet $\overline{\mu}^T g(\overline{c}) = 0$, $\overline{\mu} = 0$ has to be and $\nabla e = 0$. Hence, Equation 2.1.5 simplifies to

$$\sum_{i=1}^{n} \overline{\alpha}_i \nabla J_i(\overline{c}) = 0. \tag{2.1.8}$$

Event though this is not a sufficient condition for Pareto optimality, all Pareto optimal points have to satisfy these conditions. In fact in practice it shows, that points satisfying these conditions are pretty good conjectures for Pareto optimal points [Del04].
Let $\hat{\alpha}$ be the solution of the quadratic optimization problem

$$\min_{\alpha \in \mathbb{R}^k} \left\{ \left\| \sum_{i=1}^{k} \alpha_i \nabla J_i(c) \right\|_2^2 : \alpha_i \geq 0, i = 1, ..., k, \sum_{i=1}^{k} \alpha_i = 1 \right\}. \tag{2.1.9}$$

We consider the weighted objective function

$$O_{\hat{\alpha}}(c) = \sum_{i=1}^{n} \hat{\alpha}_i J_i(c). \tag{2.1.10}$$

The local minima of this function satisfy

$$\sum_{i=1}^{n} \hat{c}_i \nabla J_i(c) = 0, \tag{2.1.11}$$

which suits the form of Equation 2.1.8 and therefore holds the KKT conditions.
Seen that way, the set of all minima of $O_{\hat{\alpha}}$ contains the Pareto front. This motivates the search for the minima of $O_{\hat{\alpha}}$ in order to find the set of optimal countermeasures.

2.2 Dynamical system

To find the minima of $O_{\hat{\alpha}}$ a special algorithm is used. This algorithm needs a discrete dynamical system to operate on. Such a discrete dynamical system consists of a semigroup

with a zero, G, operating on a manifold M. This means there is a function

$$T : G \times M \to M$$
$$(g, x) \rightarrowtail T_g(x),$$

such that $T_g \circ T_h = T_{g \circ h}$, $T_e = id_M$.
Let $f : \mathcal{C} \to \mathcal{C}$ be for now a random, but continuous, self-imaging function. Then $(\mathbb{N}_0, \mathcal{C}, \Phi)$ with

$$\Phi(k, x_0) = f^k(x_0) = f(x_k) = x_{k+1} \quad , k = 0, 1, 2, ... \tag{2.2.1}$$

is a dynamical system, since $f^{m+n} = f^m \circ f^n$ and of course $\Phi(0, x) = x$. This was a short introduction to dynamical systems. Since these only deliver a mathematical structure and the properties of these is not of greater importance, this introduction is rather short. However, interested readers can learn more about dynamical systems in [Den05].
Let

$$q(c) := \nabla O_{\hat{\alpha}}(c) = \sum_{i=1}^{k} \hat{\alpha}_i \nabla J_i(c). \tag{2.2.2}$$

In order to find the minima of $O_{\hat{\alpha}}$ a discrete dynamical system of the form

$$c_{k+1} = c_k - h_k q_k \tag{2.2.3}$$

is used, where $h_k > 0$ is a step length with respect to the current step k. Then

$$\dot{c} = -q(c) \tag{2.2.4}$$

is a gradient system [Wil10] and since

$$\langle \nabla O_{\hat{\alpha}}(c), -q(c) \rangle = -|\nabla O_{\hat{\alpha}}(c)|^2 \leq 0, \tag{2.2.5}$$

$O_{\hat{\alpha}}(c)$ is also a Liapunov function [Wil10]. For every other case then $\mathcal{J} = 0$ (which is a proper assumption for this work) it is even a strict Liapunov function and therefore $O_{\hat{\alpha}}(c_{k+1}) < O_{\hat{\alpha}}(c_k)$. Hence, $-q(c)$ is a decent direction with respect to $O_{\hat{\alpha}}$.

2.3 Subdivision algorithm

As outlined before, there may be more than one local minimum of function $O_{\hat{\alpha}}(c)$ (2.1.10). A 'normal' approach with, for example the Newton method, would only deliver one minimum, depending on the starting point. To find a Pareto front, consisting of infinitely many points would be impossible. A way to find a region including the Pareto front is by using a modified version of the subdivision algorithm, presented in [Ser06]. The modification only affects the selection step of the subdivision algorithm. While the original algorithm presented in [Del04] and [Del02] is using the box selection $\mathcal{B}_k$ in the k-th iteration and a function f as follows

$$\mathcal{B}_k = \{B \in \hat{\mathcal{B}}_k : \text{ theres exists } \hat{B} \in \hat{\mathcal{B}}_k \text{ such that } f^{-1}(B) \cap B \neq \emptyset\}, \tag{2.3.1}$$

while here

$$\mathcal{B}_k = \{B \in \hat{\mathcal{B}}_k : f(B) \cap B \neq \emptyset\}, \tag{2.3.2}$$

is used. The latter one keeps the algorithm simpler, which is beneficial in terms that it is easier to implement, but also a lot faster. Since the processing time is an important issue, this is the main reason for the change.

2.3.1 Subdivision algorithm

The idea behind the subdivision algorithm is as follows: Close to a minimum the gradient should be very small and following a descent direction should soon lead to a minimum in this direction, so that the direction has to be changed. Hence, if there would be a box around this minimum, choosing a random point of that box and following a descent direction should end up in a point in the same box. On the other side, if the box is far away from a minimum, it is to be expected, that choosing a random point and following a descent direction should end up in a point outside the box. The latter one gets then dropped, the first one is kept. If the boxes are covering the whole domain of a function in the beginning and if they are getting smaller and dropped as outlined over several steps, then a set of points close to the Pareto front should be left over.
In mathematical terms let Q be a compact domain and $\mathcal{B}_0$ a collection of finitely many subsets such that $\bigcup_{B \in \mathcal{B}_0} B = Q$. Let furthermore be

$$x_{k+1} = f(x_k) \quad , k = 0, 1, 2, ... \tag{2.3.3}$$

a discrete dynamical system. Then the next box collection $\mathcal{B}_k$ is obtained as following:

Subdivision

A new box covering $\hat{\mathcal{B}}_k$ is obtained by shrinking the boxes and adding new boxes such that

$$\bigcup_{B\in\hat{\mathcal{B}}_k} B = \bigcup_{B\in\mathcal{B}_{k-1}} B \tag{2.3.4}$$

and

$$\operatorname{diam}(\hat{\mathcal{B}}_k) < \operatorname{diam}(\hat{\mathcal{B}}_{k-1}). \tag{2.3.5}$$

Selection

The new box selection follows

$$\mathcal{B}_k = \{B \in \hat{\mathcal{B}}_k : f(B) \cap B \neq \emptyset\}. \tag{2.3.6}$$

The convergence of this algorithm towards the fixed points of the underlying dynamical system is shown in [Ser06], but would now lead too far.
Even though the subdivision algorithm used in this work follows [Ser06], the implementation of the underlying dynamical system into the algorithm follows [Del04]:
The dynamical system f is identified as described in Section 2.2. c_0 is one test point in $\mathcal{B}_k$. To cover $f(B)$ and still have a applicable algorithm the approximation $f(B) \approx f(c_0^1, c_0^2, ...)$ for finite many test points c_0^j is made.

Step length

The step length is calculated according to the following rules:

- Find the largest $n \in \mathbb{N}$ such that, for every $i = 1, ..., n$

$$\mathcal{J}(c_k + nh_0q_k) < \mathcal{J}(c_k + (n-1)h_0q_k) \tag{2.3.7}$$

$$\langle \nabla J_i(c_k + (n-1)h_0q_k), q_k\rangle < 0. \tag{2.3.8}$$

h_0 is a scan length, that should be in a reasonable scale with respect to the radius of the boxes.

- Determine for every $i \in \{1, ..., n\}$ with

$$J_i(c_k + nh_0p_k) > J_i(c_k + (n-1)h_0p_j) \tag{2.3.9}$$

a new

$$c_k^i = c_k + ((n-1)j_0 + \Delta_i)p_j, \quad \Delta_i \in (0,1) \tag{2.3.10}$$

via quadratic backtracking. The c_k^i with minimal Δ_i is chosen as testing object and named $\hat{c}_k$. If

$$\mathcal{J}(\hat{c}_k) <_p \mathcal{J}(c_k + (n-1)h_0p_k) \tag{2.3.11}$$

is valid, then $\hat{c}_k$ is accepted as new iterate $c_{k+1} = \hat{c}_k$. If not, this step is repeated to find a new iterate between $c_k + (n-1)h_0p_k$ and $\hat{c}_k$.

2.3.2 Sampling algorithm

A different, derivable free, version of the subdivision algorithm is the sampling algorithm. While it is less robust with respect to errors [Del04] and the use of too few test points, it is superior in terms of distinguishing global and local optimality. There is also no need to determine the gradient of the objective function. On the other side there may be a need for more test points, which exterminates the benefit in terms of processing time. The sampling algorithm follows this pattern:

Subdivision

This step is the same as for the subdivision algorithm.

Selection

Choose a set of test points $\mathcal{X}_B \subset B \in \hat{\mathcal{B}}_k$.
Let N be the set of Pareto optimal points of $\bigcup_{B \in \hat{\mathcal{B}}_k} \mathcal{X}_B$ and choose the new box covering as $\mathcal{B}_k := \{B \in \hat{\mathcal{B}}_k : \exists y \in \mathcal{X}_B \cap N\}$.

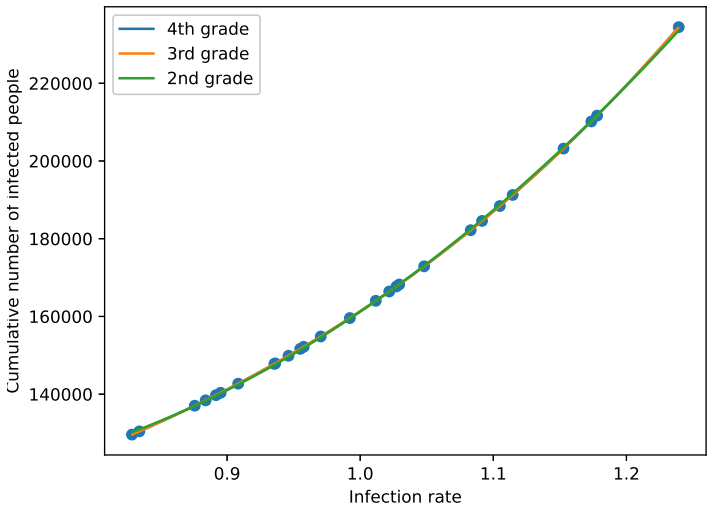

Figure 2.4.1: Regressions of $S_Y(k_E)$ with polynomials of grade two, three and four. While there is a little difference between grade two and three, a higher grade does not contribute to any visible improvement.

2.4 Cost functions

To put the algorithms to work, two objective functions are created.

$$J_1(c) = S_Y(c), \tag{2.4.1}$$

$$J_2(c) = 3c_1^2 + 0.4c_2 + 0.6c_3. \tag{2.4.2}$$

J_1 refers to the symptomatically infected and J_2 is a randomly distributed cost function of the countermeasures to demonstrate the work of the algorithms. J_2 is therefore not backed up by any scientific research, but assumes the costs of the different countermeasures in a heuristic way. With this cost function school closures are very expensive, while mask wearing is the cheapest countermeasure. First, to use J_1, one has to find the dependency $S_Y(k_E)$. This is possible via a polynomial regression of grade three. In Figure 2.4.1 the regressions with polynomials of grade two, three and four are shown. While there is a little difference between grade two and three, a higher grade does not contribute to any visible improvement. Therefore a polynom of grade three is chosen for the regression. Since the connection $k_E(c)$ is already known (c. Section 1.3), it is possible to find $S_Y(c)$ directly.
The continuity of $\mathcal{J}$ is delivered by the following theorem.

Continuity of ODE solutions [Sch17]: Is $x(t)$ a solution of the ODE $\dot{x}(t) = F(t, x(t))$ and F r times continuously derivable, then $x(t)$ is $r+1$ times continuously derivable. If $r = \infty$, then $x(t)$ is r times continuously derivable.

Hence, $J_1(c) = S_Y(c)$ is continuous and of course J_2 as well.

2.5 Applying the algorithms

Both algorithms are working properly for themselves and it is possible with both to find the Pareto front. However, it might take great effort in terms of processing time. For example the subdivision algorithm can easily be caught in many local Pareto optimal points or rather in their neighbourhood, resulting in a great amount of processing time. Therefore a combination of these two algorithms is chosen, in order to eliminate possible weaknesses of the algorithms. The main idea is to let the subdivision algorithm find areas with Pareto optimal points and cover it densely with small boxes and therefore many test points and apply then the sampling algorithm from time to time to sort out only local Pareto optimal points in order to decrease the amount of boxes and thereby the processing time.
At the beginning a uniform grid $[0, 1]^3$ is created and all Pareto optimal points with respect to this grid determined. This start box covering is shown in Figure 2.5.1. Then the resulting grid points are used as box center in order to apply this start box covering to the algorithms. The best results in a reasonable processing time (less than 10 hours) are obtained for an alternately use of the two algorithms. First the subdivision algorithm is run two times with 27 test points in each box. Hence, the covering with box centers is dense enough, that the sampling algorithm can be run with only one test point (the box center) to thin out unnecessary boxes again. After only two loops of those two algorithms the result in Figure 2.5.2 is obtained. The Pareto front seems to build a straight line from point (0.1,0,0.8) to point (0.1,1,0.6) with some accumulations at the end points of the line. While the line gets straighter and sharper the more often the algorithms are repeated, the accumulations at the end points vanish more and more. Therefore it is probable, that the true Pareto front has no accumulations at the end points of the line. However, the processing time takes too long (more than ten hours) to verify this.
This means, that there is no space for change in terms of school closure and also the influence of contact tracing is small. On the other side there seems a big room for changes in terms of mask wearing. This behaves like anticipated since the school closures are very

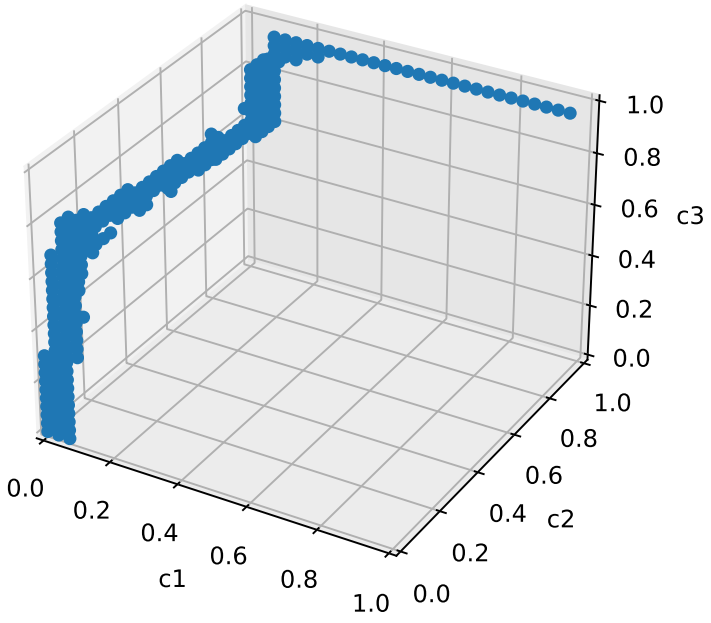

Figure 2.5.1: Start box covering obtained by examining the Pareto optimal points of a grid. The dots are the box centers.

expensive, while mask wearing is the cheapest countermeasure. Mathematically spoken, all the points within the Pareto front are equivalent. That means that no single point is superior to the others. Therefore there is still a political decision to make. However, with the subdivision algorithm it is possible to sort out an assessable set of different options, depending on what are the underlying objectives.

While the algorithms work and the Pareto front is recognizable, a higher resolution would be desirable. This is prevented by the long processing time. Possible solutions would be either to find more efficient algorithms or to improve the program code, so the program itself gets more efficient. It also has to be noticed, that this approach is useful when it comes to find a suiting tableau of countermeasures and therefore only a single tool in the pandemic management. In [Koe20] it is outlined, that a successful optimization based policy has to be a process of continually monitoring, feedbacking and adapting instead of running one model at the beginning and stick to its findings. Also the impact of

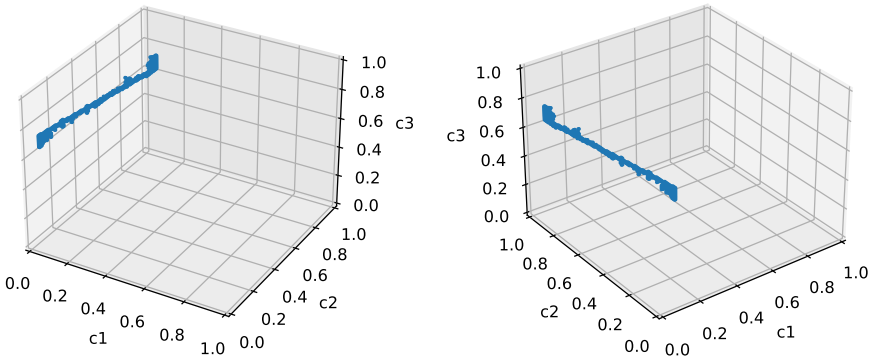

Figure 2.5.2: Approximated Pareto front obtained by two loops with two iterations and 27 test points of the subdivision algorithm and one iteration of the sampling algorithm with one test point. The countermeasures are c_1 school closure, c_2 mask wearing, c_3 contact tracing.

inaccurate data is not discussed here. [Pei18] for example approaches this topic through implementing the perturbation ϵ into the definition of Pareto optimal points and adjust the subdivision algorithm accordingly.

3 Summary

This work delivers a simple tool to make multi-objective decisions effectively and scientifically, exemplary by the use of countermeasures to control the Covid-19 pandemic in Berlin, Germany. First a model is introduced to forecast the behavior of the pandemic. It is a system of ODEs, that simulates the whole population in different compartments as susceptible, infected, etc.. This forecast is made for different possible countermeasures and a regression in dependency of those is made in order to be able to forecast the pandemic in dependency of countermeasures. The system of ODEs is solved by different one step methods as the Euler method and the Runge-Kutta methods of order 2 and 4. Then these methods are compared with respect to occurring errors and process time. It shows, that the Euler method is superior for big step sizes, while the other methods overtake quickly for smaller step sizes.
In a second step two variances of the subdivision algorithm are introduced, combining the approaches from [Del04] and [Ser06]. The whole process of integrating the algorithms is described as well as a concrete use on an exemplary objective function. The best results within less then ten hours of processing time is obtained by a combination of the algorithms to create on one side enough a close enough covering and on the side sort out unnecessary boxes to decrease the processing time. Concretely the combination consists of two loops with two iterations and 27 test points of the subdivision algorithm and one iteration of the sampling algorithm with one test point. The algorithm is especially useful for finding a good set of different countermeasures and gives insight over all possible, but also preferred combinations. In this case it showed for example that there is a wide freedom for restrictions concerning mask wearing, while there is a very narrow scope in terms of school closures. However, the impact of perturbations in the data is not discussed here, even though it is a very important facet of multi-objective management, since real life data, especially in a pandemic, will always be perturbed.
All in all this works illustrates the process of deciding multi-objective decisions as well as it motivates through the concreteness and simplicity the use of mathematical and scientific tools in decision making. However, for applying these techniques in real life, the model would need to be more detailed and the process of multi-objective decision

making would need to be applied and feedbacked permanently.

Appendix

School closure (%)	Mask wearing (%)	Contact tracing (%)	k_E
0	0	0	1,24
0	0	50	1,085
0	0	100	0,9914
0	50	0	1,17
0	50	50	1,02
0	50	100	0,9324
0	100	0	1,11
0	100	50	0,9539
0	100	100	0,881
50	0	0	1,18
50	0	50	1,032
50	0	100	0,9478
50	50	0	1,12
50	50	50	0,965
50	50	100	0,8886
50	100	0	1,04
50	100	50	0,9106
50	100	100	0,8364
100	0	0	1,14
100	0	50	1,019
100	0	100	0,9334
100	50	0	1,1
100	50	50	0,9579
100	50	100	0,884
100	100	0	1,03
100	100	50	0,8888
100	100	100	0,8288

Table 3.0.1: The infection rate k_E determined from the ABM for different combinations of countermeasures.

Selbstständigkeitserklärung

Ich versichere hiermit, dass ich die vorliegende Masterarbeit mit dem Thema:
Multi-objective Optimal Control of a Pandemic Model for Covid-19 Management
selbstständig verfasst und keine anderen Hilfsmittel als die angegebenen benutzt habe.
Die Stellen, die anderen Werken dem Wortlaut oder dem Sinne nach entnommen sind,
habe ich in jedem einzelnen Falle durch Angaben der Quelle, auch der benutzten Sekund¨arliteratur,
als Entlehnung kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Konstanz, 07.03.2022
Ort, Datum

Jens Gebert

List of references

[Bä20] Günter Bärwolff, *Numerische Lösung gewöhnlicher Differentialgleichungen*, Numerik für Ingenieure, Physiker und Informatiker, pp. 225–276, Springer Berlin Heidelberg, Berlin, Heidelberg, 2020.

[Del02] Michael Dellnitz, Oliver Schütze und Stefan Sertl, *Finding zeros by multilevel subdivision techniques*, IMA Journal of Numerical Analysis **22**, 167-185 (2002).

[Del04] Michael Dellnitz, Oliver Schütze und Thorsten Hestermeyer, *Covering Pareto sets by multilevel subdivision techniques*, Journal of Optimization, Theory and Applications (2004).

[Den05] Manfred Denker, *Einführung in die Analysis dynamischer Systeme*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[Ehr05] Matthias Ehrgott, *Multicriteria Optimization (2. ed.)*., Multicriteria Optimization: Second Edition (2005).

[Gri21] Veronika Grimm, Friederike Mengel und Martin Schmidt, *Extensions of the SEIR model for the analysis of tailored social distancing and tracing approaches to cope with COVID-19*, Scientific Reports **11** (2021).

[Kim22] Nena Kimpfler, *Modellierung der Covid-19 Infektionen anhand des Sei^{3}rd Modells*, 2022.

[Koe20] Johannes Koehler, Lukas Schwenkel, Anne Koch, Julian Berberich, Patricia Pauli und Frank Allgöwer, *Robust and optimal predictive control of the COVID-19 outbreak*, Annual Reviews in Control **51** (2020).

[Kuh51] H. W. Kuhn und A. W. Tucker, *Nonlinear programming*, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950 (Berkeley and Los Angeles), University of California Press, 1951, pp. 481–492. MR 0047303 (13,855f)

[Lin20] Matthias Linden, Sebastian B. Mohr, Jonas Dehning, Jan Mohring, Michael Meyer-Hermann, Iris Pigeot, Anita Schöbel und Viola Priesemann, *Überschreitung der Kontaktnachverfolgungskapazität gefährdet die Eindämmung von COVID-19*, Dtsch Arztebl International **117**, 790-791 (2020).

[Mül20] Sebastian A Müller, Michael Balmer, Billy Charlton, Ricardo Ewert, Andreas Neumann, Christian Rakow, Tilmann Schlenther und Kai Nagel, *Using mobile phone data for epidemiological simulations of lockdowns: government interventions, behavioral changes, and resulting changes of reinfections*, medRxiv (2020).

[Pei18] Sebastian Peitz und Michael Dellnitz, *A Survey of Recent Trends in Multiobjective Optimal Control—Surrogate Models, Feedback Control and Objective Reduction*, Mathematical and Computational Applications **23** (2018).

[Ran18] Rolf Rannacher, *Analysis 2: Differential- und Integralrechnung für Funktionen mehrerer reeller Veränderlichen*, 1. auflage Auflage, Heidelberg University Publishing, Heidelberg, 2018.

[Sch17] Jürgen Scheurle, *Grundlegende Theorie*, pp. 43–67, Springer International Publishing, Cham, 2017.

[Ser06] Stefan Sertl und Michael Dellnitz, *Global Optimization using a Dynamical Systems Approach*, Journal of Global Optimization **34**, 569-587 (2006).

[Wil10] Mathias Wilke und Jan Pruess, *Gewöhnliche Differentialgleichungen und dynamische Systeme*, vol. -1, 01 2010.

[Wul21] Hanna Wulkow, Tim O. F. Conrad, Nataša Djurdjevac Conrad, Sebastian A. Müller, Kai Nagel und Christof Schütte, *Prediction of Covid-19 spreading and optimal coordination of counter-measures: From microscopic to macroscopic models to Pareto fronts*, PLOS ONE **16**, 1-29 (2021).