Robust Two-Dose Vaccination Schemes and the Directed b-Matching Problem

Jenny Segschneider * Arie M.C.A. Koster †

February 15, 2023

Abstract

In light of the recent pandemic and the shortage of vaccinations during their roll-out, questions arose regarding the best strategy to achieve immunity throughout the population by adjusting the time gap between the two necessary vaccination doses. This strategy has already been studied from different angles by various researches. However, the deliveries of vaccination doses also proved to be highly uncertain, with manufacturers not being able to deliver the promised amount of vaccines on time.

In this paper, we study the robust version of this problem and its generalization to matchings on arbitrary graphs. By exploring the problem, we show that it is weakly NP-hard for a constant number of scenarios and strongly NP-hard else. Further, we propose a pseudo-polynomial algorithm for the weakly NP-hard subproblem with a constant number of scenarios and time between both doses. Finally, we perform computational experiments to better understand the behavior of the problem.

*RWTH Aachen University, Research Area Discrete Optimization
E-Mail: segschneider@math2.rwth-aachen.de

†RWTH Aachen University, Research Area Discrete Optimization
E-Mail: koster@math2.rwth-aachen.de

1 Introduction

In a global pandemic, production and distribution of an effective vaccine is an important task to solve. Many optimization problems arise in this context. An overview of these problems was given by Duijzer et al. in [4] and by Wouters et al. in [21]. The second paper is focused on the ongoing COVID-19 pandemic while the first paper concentrates on the outbreak of an arbitrary infectious disease. In this paper, we focus on a problem arising when a full vaccination requires two doses per patient given within a fixed time frame. This was the case for most available vaccines against the Corona Virus Disease COVID-19 in 2021. At the beginning of the vaccine roll-out, there was only a limited amount of vaccine available which had to be used as effectively as possible to slow down the spread of the disease and possibly prevent many deaths. However, it was unclear if this would be achieved by vaccinating as many patients as possible with only one dose to achieve partial immunity or if completing immunization with a second dose as early as possible for a smaller portion of society would be better. We call this problem the Two-Dose Scheduling Problem or, for short, the Two-Dose Problem. In addition to that, the exact number of vaccine doses delivered was subject to uncertainty. Sometimes, doses of vaccine could not be delivered as promised and delivery dates were delayed. We propose a variant of the problem including these delays using the concept of Robust Optimization as introduced by Ben-Tal et al. [2].

Vaccination Strategies for the deterministic Two-Dose Problem have been researched extensively in the last three years. This section does not claim completeness and is only meant as an overview for recent publications and approaches in this topic. Jurgens and Lackner [7] evaluated the effect of delaying the second dose to 6, 9 or 12 weeks after the first dose. Moghadas et al. [11] proposed an agent-based model using data from the United States. One often used approach is to use compartment models, which split the population in compartments depending on their vaccination and health status, i.e., the compartments Susceptible, Exposed, Infectious, Recovered, and Vaccinated (SEIRV). The model is then build based on differential equations describing the transitions between different compartments. Silva et al. [19] proposed SEIR models (compartmental models using the compartments Susceptible, Exposed, Infectious and Recovered) to minimize the number of expected deaths due to the pandemic and included different subpopulations. Parino et al. [13] extended an SIR model by adding different stages and applied it to the COVID-19 pandemic in Italy. Most recently, Omae et al. [12] proposed a SIRVVD model including compartments for both Vaccinations with one and two doses.

Another section of research centers on the question whether (and how many) doses of vaccine should be stored for use as a second dose. Tuite et al. [20] compared two

different approaches, the fixed strategy of always storing half of the delivered doses and the flexible strategy, showing that the flexible strategy prevents more COVID-19 cases. Mak et al. [9] compare three different strategies: holding back half of the doses, using all doses as they arrive, and expanding the time between first and second dose. This paper extends Segschneider and Koster [17]. They introduced a new deterministic model based on the theory of b-matchings for the Two-Dose problem and showed NP-hardness for the case where each patient requires three doses.

The Two-Dose Problem with uncertainties in the deliveries has not been the focus of attention. To the best of our knowledge, there have only been two further publications considering this problem. Shumsky et al. [18] use a stochastic dynamic program to suggest recommended strategies for how many doses should be stored, or in their words set-aside, and show their performance in numerical experiments. Recently, Calafiore et al. [3] propose an probabilistic optimization framework to find an optimal vaccination strategy in which each constraint is only violated with a fixed risk ε. They base this framework on a deterministic model that is similar to the one we use in this paper. The only two differences are the dropping of the integrality constraint and adding a parameter α which measures the portion of the population that may receive their second dose with a delay. However, our model for the Two-Dose Problem with uncertain deliveries is different. In contrast to them, we focus on a robust approach when including uncertainties and do not allow any constraints to be violated. This results in more conservative but robust solutions.

The contributions of this paper are threefold. First, on the basis of the deterministic model [17], we introduce two new formulations of the Robust Two-Dose Problem and the robust directed b-Matching Problem. Second, we examine their complexity and show strong and (at least) weak NP-hardness for subproblems. For the weakly NP-hard subproblem, we then construct a pseudo-polynomial algorithm. Finally, we conduct a computational study with the aim of understanding the problem and its characteristics further.

This paper is organized as follows. In Section 2, we introduce the deterministic problem and important notation and summarize key results on this version of the Two-Dose Problem. We introduce two versions of the robust Two-Dose Problem in Section 3: the Fixed Robust Two-Dose Problem and the version this work concentrates on, the Directed Robust Two-Dose Problem. The second version, we generalize to the Directed Robust b-Matching Problem and investigate the correlation between both problems in Section 4. Next, we study their complexity in Section 5 and give a pseudo-polynomial algorithm for a special case of the Directed Robust Two-Dose Problem in Section 6. Last, we conduct a computational study in Section 7.

2 Deterministic Problem Description and Formulation

In this section, we revisit the results from [17] that we will subsequently build upon in later sections. The Two-Dose Problem (without uncertainty) consists of finding a schedule for first and second doses, given a time frame for the second dose relative to the first dose and limited supply. We assume a discrete time horizon of n time steps. At time step $t \in \{1, \dots, n\} =: T$, b_t doses of vaccine are delivered, up to u_t doses can be stored for later usage and ν_t doses can be given to a patient at that time step. We denote by $B_t := \sum_{i=1}^{t} b_i$ the sum of all doses delivered up to time step t. Each patient has to receive their second dose of vaccine within $\varepsilon_{\min}$ to $\varepsilon_{\max}$ time steps after their first dose. We call this constraint the timing constraint. For a first dose at time step $i \in T$, the second dose has to be given in the time frame $\{i + \varepsilon_{\min}, \dots, i + \varepsilon_{\max}\} \cap T =: J(i)$. Similarly, we denote by $I(j) = \{j - \varepsilon_{\max}, \dots, j - \varepsilon_{\min}\} \cap T$ the feasible time frame for a first dose with a second dose at time step $j \in T$. We call a pair of two time steps (i, j) a feasible appointment if $j \in J(i)$. For each feasible appointment (i, j), $s_{i,j}$ is the number of patients to receive their first dose at time step i and their second dose at time step j. Additionally, we assume n to be chosen big enough such that $\sum_{t=1}^{n-\varepsilon_{\min}-1} b_t \geq \sum_{t=n-\varepsilon_{\min}}^{n} b_t$ and there are no doses left over that would require a second dose after time step n. We can assume this without loss of generality because an instance violating this constraint can be adapted to satisfy it by adding $\varepsilon_{\min}$ time steps at the end without delivery.

The Two-Dose Problem can then be formulated as an integer linear program (ILP) as follows:

$$\max \sum_{i=1}^{n} \sum_{j \in J(i)} c_{i,j} s_{i,j} \tag{1}$$

$$\text{s.t.} \sum_{i=1}^{t} \left(2 \cdot \sum_{j=i+\varepsilon_{\min}}^{t} s_{i,j} + \sum_{j=t+1}^{\min\{n, t+\varepsilon_{\max}\}} s_{i,j} \right) \leq B_t \qquad \forall t \in T \tag{2}$$

$$B_t - \sum_{i=1}^{t} \left(2 \cdot \sum_{j=i+\varepsilon_{\min}}^{t} s_{i,j} + \sum_{j=t}^{\min\{n, t+\varepsilon_{\max}\}} s_{i,j} \right) \leq u_t \qquad \forall t \in T \tag{3}$$

$$\sum_{j \in J(t)} s_{t,j} + \sum_{i=0}^{t-\varepsilon_{\min}} s_{i,t} \leq \nu_t \qquad \forall t \in T \tag{4}$$

$$s_{i,j} \in \mathbb{Z}_+ \qquad \forall i \in T,\ j \in J(i) \tag{5}$$

$$s_{i,j} = 0 \qquad \forall i \in T,\ j \notin J(i). \tag{6}$$

Constraints (2) ensure that in each time step, only doses previously delivered and unused are used for vaccinations. We call these constraints delivery constraints. Constraints (3) and (4) are the storage capacity constraint and vaccination speed constraint respectively, ensuring the upper bound on both of these values. Note that by setting $u_n = 0$, we can enforce that each delivered dose is used. We cannot use partial doses for a vaccination, so (5) ensures integrality of the solution. Lastly, (6) ensures the timing constraint by setting $s_{i,j}$ for all non-feasible appointments to 0.

Remark 1 We can assume without loss of generality that all values $c_{i,j}$ are non-negative. Since all delivered doses have to be used, there are $\frac{1}{2}\sum_{t=1}^{n} b_t$ appointments in each feasible schedule. Thus, setting the objective coefficient to $c_{i,j} - k$ for some $k \in \mathbb{R}$ only changes the objective by a constant value and does not influence maximality. For an instance with negative value $c_{i,j}$, we can set $k := \min_{i \in T, j \in J(i)} c_{i,j}$ and change the value to $c_{i,j} - k$ resulting in an equivalent instance with non-negative values.

2.1 The Matching Approach

The computational experiments in Section 7 show that the ILP introduced in the previous section can be solved relatively fast. However, as a basis for generalizing the Robust Two-Dose Problem, we now introduce an alternate view using the theory of perfect b-matching problems. The approach is based on the idea that each appointment is a pair of two "matched" time steps.

However, in a regular b-matching, there is no way of moving capacity from one vertex to another the way storing doses would require. Thus, we designed a gadget incorporating the concept of storage.

For each instance of the Two-Dose Problem, we define an instance of the perfect, weighted b-matching problem which models the instance of the Two-Dose Problem. To this end, we define a Graph $\mathcal{G}$ with a vertex $v_t \in V$ for each time step $t \in \{1, \dots, n\}$ and edges connecting those that form a feasible appointment. We will refer to these vertices as time step vertices and to the edges connecting them as appointment edges. The value $m_{\{v_i, v_j\}}$ of the perfect matching on appointment edge $\{v_i, v_j\}$ defines the schedule for the corresponding appointment: $s_{i,j} = m_{\{v_i, v_j\}}$. Additionally, we include gadget vertices v_t^1, v_t^2 and v_t^3 for each time step $t \in T$ connecting the time step vertices and functioning as a storage. The gadget and interpretation for the first two time steps is depicted in Figure 1.

Formally, the new graph is then defined as

$$V = \{v_i, v_i^1, v_i^2, v_i^3 \colon i = 1, \dots, n-1\} \cup \{v_n, v_n^1, v_n^2\}, \tag{7}$$

$$\begin{aligned} E = &\{\{v_i, v_j\} \colon i = 1, \dots, n - \varepsilon_{\min}\,; j \in J(i)\} \cup \\ &\{(v_i, v_i^1) \colon i = 1, \dots, n\} \cup \{(v_i^1, v_i^2) \colon i = 1, \dots, n\} \cup \\ &\{(v_i^2, v_i^3) \colon i = 1, \dots, n-1\} \cup \{(v_i^3, v_{i+1}^2) \colon i = 1, \dots, n-1\}. \end{aligned} \tag{8}$$

We set capacity β_v on the vertices $v \in V$, upper bounds q_e and value w_e on the edges $e \in E$ as follows:

$$\beta_{v_t} = \beta_{v_t^1} = \beta_{v_t^2} = \beta_{v_t^3} = B_t, \tag{9}$$

$$w_e = \begin{cases} c_{i,j} & \text{if } e = \{v_i, v_j\},\ j \in J(i) \\ 0 & \text{else.} \end{cases} \tag{10}$$

$$q_e = \begin{cases} \nu_t & \text{if } e = \{v_t^1, v_t^2\} \\ u_t & \text{if } e = \{v_t^2, v_t^3\} \\ \infty & \text{else} \end{cases} \tag{11}$$

For a given perfect β-matching m, the value m_{v_i,v_j} denotes $s_{i,j}$, that is the number of vaccination appointments with first dose given at time step i and second dose at time step j. Let $y_t = \sum_{j \in J(t)} m_{t,j} + \sum_{i \in I(t)} m_{i,t}$ be the number of doses used at time step

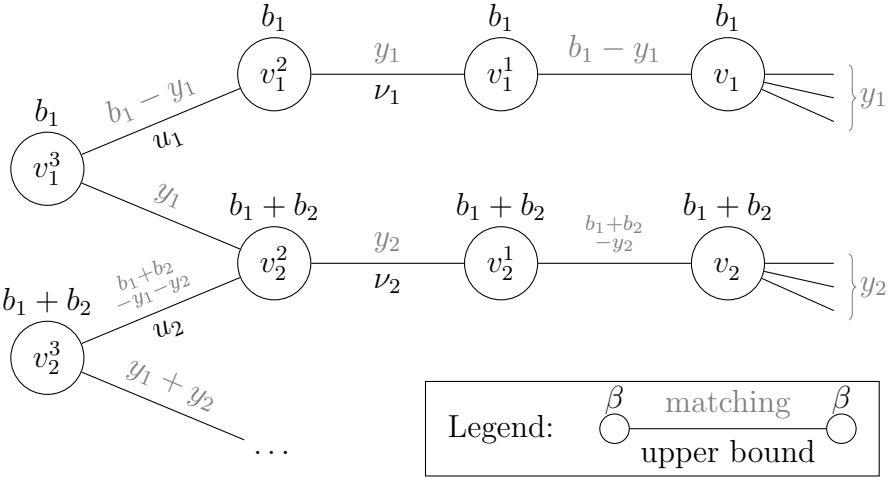


Figure 1: Storage Gadget and interpretation for the first two time steps

$t \in T$. Since m is a perfect matching and the time step vertices v_t have only one edge not counted in this sum. Thus, the value of the matching on this edge has to be $m_{\{v_t,v_t^1\}} = \beta_{v_t} - y_t = B_t - y_t$ in order to satisfy the perfect matching constraint

$\sum_{e\in\delta(v_t)} m_e = \beta_t$. We apply this argumentation for each vertex of the gadget to get the value of the perfect matching m on the gadget edges depending on y. In Figure 1, these values are presented by the gray numbers on top of the gadget edges. Notably, $m_{\{v_t^1,v_t^2\}} = y_t$ the number of doses used at time step t. Thus, the vaccination speed constraint is equivalent to the upper bound $q_{\{v_t^1,v_t^2\}}$ on that matching edge. The same holds for the storage capacity constraint and the upper bound on edge $\{v_t^2, v_t^3\}$. In each time step, the value of the matching on that edge is $m_{\{v_t^2,v_t^3\}} = \sum_{i=1}^{t} b_i - y_i$ equal to the number of doses that need to be stored in time step t. Thus, the upper bound of $q_{\{v_t^2,v_t^3\}} = u_t$ ensures the storage capacity constraint is kept.

Finally, for a perfect matching m on $\mathcal{G}$, we defined a schedule $s(m)$ for the corresponding instance of the Two-Dose Problem as $s(m)_{i,j} = m_{\{v_i,v_j\}}$. Since the value w is 0 for all gadget edges and equal to the value of the appointment for each appointment edge, it holds $w^T m = c^T s(m)$ and the maximum weight perfect β-Matching on $\mathcal{G}$ defines the maximum value schedule for the corresponding Two-Dose Problem instance. We concluded:

Theorem 1 ([17]) For each perfect β-matching $(m_e)_{e\in E}$ on $\mathcal{G}$, there is a corresponding feasible schedule s with value $c^T s = (w)^T m$ satisfying the storage capacity and vaccination speed constraints and vice versa.

Since weighted b-matching can be solved in polynomial time (c.f. [1], [15], [16]), the Two-Dose Scheduling Problem with storage capacity and vaccination speed constraints can also be solved in polynomial time. Note, that the Three-Dose Problem is NP-complete [17].

3 Robust Two-Dose Problem Definition

As deliveries are not certain, we now include uncertainty in the deliveries. This results in an ILP with uncertainty in the right hand side. Let $\mathcal{B} \subseteq \mathbb{Z}_+^n$ be a given set of scenarios. We assume that for each scenario, there exists a feasible solution and the total number of doses delivered is the same for all scenarios. Deliveries are only delayed and not canceled. Other than that, we pose no additional conditions on $\mathcal{B}$ in this theoretical section. For the computational study in Section 7, we use randomly generated uncertainty sets with a finite number of scenarios.

In this section, we introduce two models to include uncertainty. They are similar to the two decision models introduced in [10]. The first model, called Fixed Robust Two-Dose Problem is a one stage problem and consists of finding the best vaccination schedule feasible for each scenario. The second model is a two-stage problem. In the first stage, the number of first doses is set for each time step and in the second stage,

after realization of the scenario, the final schedule is set respecting the number of first doses set in the first stage. We call this model the Directed Robust Two-Dose Problem. The name Directed will become clear in Section 3.2.

3.1 Fixed Robust Two-Dose Problem

The Fixed Robust Two-Dose Problem consist of finding a schedule which is feasible in all scenarios and maximizes the worst-case objective value. With unlimited storage capacity and vaccination speed, the problem can be described by the following integer linear program.

$$\begin{aligned} &\max_{s} \; c^T s \\ &\text{s.t.} \sum_{i=1}^{t} \sum_{\substack{j \in J(i) \\ j \leq t}} 2s_{i,j} + \sum_{i=1}^{t} \sum_{\substack{j \in J(i) \\ j > t}} s_{i,j} \leq B_t && \forall b \in \mathcal{B},\ t \in T \\ &\quad s_{i,j} \in \mathbb{Z}_+ && \forall i \in T, j \in J(i) \end{aligned}$$

The assumption about the total number of doses is necessary for the existence of a feasible solution without leftover doses. A feasible schedule s uses exactly $2 \cdot \sum_{i=1}^{n} \sum_{j \in J(i)} s_{i,j}$ doses. If the number of doses delivered is smaller, the schedule is not feasible. If there are more doses delivered, these additional doses are leftover. If there are two scenarios with a different number of total doses, a feasible schedule can only use $\min_{b \in \mathcal{B}} \sum_{i=t}^{n} b_t$ doses and additional doses in other scenarios cannot be used. We now define a minimum scenario $b^{\min}$ which is not necessarily in $\mathcal{B}$. In each time step $t \in \{1, \ldots, n\}$, this scenario satisfies

$$\sum_{i=1}^{t} b_i^{\min} = \min_{b \in \mathcal{B}} \sum_{i=1}^{t} b_i. \tag{12}$$

Theorem 2 Let $\mathcal{B}$ be a set of scenarios and $b^{\min}$ as defined above. If a schedule is not feasible for a scenario $b \in \mathcal{B}$, it is not feasible for $b^{\min}$. Reversely, if a schedule is not feasible for $b^{\min}$, then it does not satisfy the delivery constraint in a time step $i \in T$ for a scenario $b \in \mathcal{B}$.

Proof
From the definition of $b^{\min}$ we know that there is a scenario $b^* \in \mathcal{B}$ with $\sum_{j=1}^{t} b_j^* = \sum_{j=1}^{t} b_j^{\min}$ for each time step $t \in T$. If there is a scenario $b \in \mathcal{B}$ violating the delivery constraint in time step t, then $b^{\min}$ also violates this constraint. Reversely, if $b^{\min}$ violates the delivery constraint in a time step $t \in T$, then there must be a scenario $b \in \mathcal{B}$ with $\sum_{i=1}^{t} b_i^{\min} = \sum_{i=1}^{t} b_i$. Thus, this scenario also violates the delivery constraint in time step t. □

Since we can assume that all deliveries over scenarios are natural numbers, this scenario is well-defined. Thus, we can infer the following corollary.

Corollary 1 A schedule is feasible for each scenario in $\mathcal{B}$ if and only if it is feasible for $b^{\min}$ and the problem reduces to the deterministic Two-Dose Problem.

This result will still hold true if we include the vaccination speed constraint, since this constraint is only dependent on the schedule and not on the deliveries.

The same approach can be used when the storage constraint is included. In this case, we also need to define the maximum scenario b^{max} analogous to $b^{\min}$ as

$$\sum_{i=1}^{t} b_i^{\max} = \max_{b \in \mathcal{B}} \sum_{i=1}^{t} b_i.$$

Theorem 3 A schedule is feasible for all scenarios in $\mathcal{B}$ if and only if it satisfies the delivery constraint in $b^{\min}$ and the storage constraint in $b^{\max}$.

Proof
From theorem (2), we know that the delivery constraint is satisfied for all scenarios if and only if it is satisfied for $b^{\min}$. Thus, for feasibility of all scenarios, we only need to show that the storage constraint is satisfied. Remember that the storage constraint is given by

$$B_t - \sum_{i=1}^{t} \sum_{\substack{j \in J(i) \\ j \leq t}} 2s_{i,j} - \sum_{i=1}^{t} \sum_{\substack{j \in J(i) \\ j > t}} s_{i,j} \leq u_t \qquad \forall b \in \mathcal{B},\ t \in T.$$

If the storage constraint is not satisfied for a scenario $b \in \mathcal{B}$ and time step $t \in T$, then it holds

$$\sum_{i=1}^{t} b_i^{\max} - \sum_{i=1}^{t} \sum_{\substack{j \in J(i) \\ j \leq t}} 2s_{i,j} - \sum_{i=1}^{t} \sum_{\substack{j \in J(i) \\ j > t}} s_{i,j} \geq B_t - \sum_{i=1}^{t} \sum_{\substack{j \in J(i) \\ j \leq t}} 2s_{i,j} - \sum_{i=1}^{t} \sum_{\substack{j \in J(i) \\ j > t}} s_{i,j} \geq u_t$$

and $b^{\max}$ also violates the constraint in time step $t \in T$. Reversely, if $b^{\max}$ violates the storgae constraint in time step $t \in T$, there is a scenario $b \in \mathcal{B}$ with $\sum_{i=1}^{t} b_i^{\max} = \sum_{i=1}^{t} b_i$ that also violates this constraint. □

Finding in optimal schedule that satisfies the delivery constraint in $b^{\min}$ and the storage constraint in $b^{\max}$ can also be done in polynomial time with the b-matching approach by using $b^{\min}$ on the inner nodes v_i and v_i^1 where the vaccination speed and dose constraints are enforced and $b^{\max}$ on the outer nodes v_i^2, v_i^3 where the storage capacity constraint is enforced.

In summary, the Fixed Robust Two-Dose Problem of finding the best schedule feasible for all scenarios reduces to the deterministic Two-Dose Problem.

3.2 Directed Robust Problems

In practice, we rarely need to fix the whole schedule up-front. It suffices to first give every patient a date for their first dose only. Then, when the deliveries are already known, we can give them the time for their second dose and thus the complete appointment. We call this problem the Directed Robust Two-Dose Problem (DRTD) to confirm with the naming of the extended, graph theoretical problem introduced later in this section. Formally, this results in a two-stage problem. In the first stage, only the number of first doses for each time step is fixed. We call this a preliminary schedule, or for short pre-schedule. Then, the scenario is realized and finally, in the second stage, the times for the second doses are set and thus the final schedule fixed. We denote the number of first doses, the pre-schedule, given in each time step t as $x_t \in \mathbb{Z}_+^n$. For each scenario $b \in \mathcal{B}$, we define a separate schedule $s^b \in \mathbb{Z}_+^{n \times n}$ feasible for that scenario. Finally, we denote by $z \in \mathbb{R}$ the value of the worst case scenario. The problem can then be expressed as the following ILP:

$$\max z \tag{13}$$

$$\text{s.t.} \sum_{i=1}^{n} \sum_{j \in J(i)} c_{i,j} s_{i,j}^b \geq z \qquad \forall\, b \in \mathcal{B} \tag{14}$$

$$\sum_{j \in J(t)} s_{t,j}^b = x_t \qquad \forall\ \begin{matrix} t \in \{1, \dots, n\}, \\ b \in \mathcal{B} \end{matrix} \tag{15}$$

$$\sum_{i=1}^{t} \left(2 \cdot \sum_{j=i+\varepsilon_{\min}}^{t} s_{i,j}^b + \sum_{j=t+1}^{\min\{n,\, t+\varepsilon_{\max}\}} s_{i,j}^b \right) \leq B_t \qquad \forall\ \begin{matrix} t \in \{1, \dots, n\}, \\ b \in \mathcal{B} \end{matrix} \tag{16}$$

$$B_t - \sum_{i=1}^{t} \left(2 \cdot \sum_{j=i+\varepsilon_{\min}}^{t} s_{i,j}^b + \sum_{j=t}^{\min\{n,\, t+\varepsilon_{\max}\}} s_{i,j}^b \right) \leq u_t \qquad \forall\ \begin{matrix} t \in \{1, \dots, n\}, \\ b \in \mathcal{B} \end{matrix} \tag{17}$$

$$\sum_{j \in J(t)} s_{t,j}^b + \sum_{i=0}^{t-\varepsilon_{\min}} s_{i,t}^b \leq \nu_t \qquad \forall\ \begin{matrix} t \in \{1, \dots, n\}, \\ b \in \mathcal{B} \end{matrix} \tag{18}$$

$$s_{i,j}^b \in \mathbb{Z}_+ \qquad \forall\, i \in \{1, \dots, n\},\ j \in J(i),\ b \in \mathcal{B} \tag{19}$$

$$s_{i,j}^b = 0 \qquad \forall\, i \in \{1, \dots, n\},\ j \notin J(i),\ b \in \mathcal{B}. \tag{20}$$

The objective is to maximize the value of the worst case scenario. To this end, we introduce a new variable z smaller than the objective value of each scenario as per (14) and maximize it in (13). Thus, in an optimal solution, z is equal to the value of the

worst case scenario. Feasibility of each schedule s^b is ensured by the constraints (16)-(20). Constraint (15) ensures that each scenario has the same amount of first doses in each time step and satisfies the pre-schedule. Note, that x is redundant and could easily be removed. However, we added it for the sake of readability.

We now propose a generalized version of the DRTD Problem: the Directed Robust Perfect b-Matching Problem (DRPbM). This problem is defined on a directed Graph $D = (V, A)$. First, for a vertex $v \in V$, let $\delta^+(v) = \{(v, w) \in A \colon w \in V\}$ the set of all outgoing arcs of v and $\delta^-(v) = \{(w, v) \in A \colon w \in V\}$ the set of all incoming arcs in v. Then, similar to the DRTD Problem, we first fix the sum of the matching over all outgoing arcs $h \in \mathbb{Z}_+^{|V|}$ for each vertex with incoming arcs ($\delta^-(v) \neq \emptyset$). We call this a preliminary matching or, for short, pre-matching. The complete matching is then fixed after the realization of the scenario. We exclude vertices $v \in V$ without outgoing arcs from the pre-matching because the complete matching has to be perfect. Thus, $\sum_{a \in \delta(v)} = h_v \equiv \beta_v$ and the pre-matching would need to be equal to the capacity in that vertex. Including v in the pre-matching would lead to infeasibility if there is uncertainty in β_v and no change if β_v is constant over all scenarios.

The ILP formulation of this problem is similar to that of the DRTD Problem (13)-(20).

$$\max z \tag{21}$$

$$\text{s.t.} \sum_{a \in A} c_a m_a^b \geq z \qquad \forall\, b \in \mathcal{B} \tag{22}$$

$$\sum_{a \in \delta^+(v)} m_a^b = h_v \qquad \forall\, v \in V \colon \delta^-(v) \neq \emptyset,\ b \in \mathcal{B} \tag{23}$$

$$\sum_{a \in \delta^+(v)} m_a^b + \sum_{a \in \delta^-(v)} m_a^b = b_v \qquad \forall v \in V,\ b \in \mathcal{B} \tag{24}$$

$$m_{i,j}^b \in \mathbb{Z}_+ \qquad \forall\, v \in V,\ b \in \mathcal{B} \tag{25}$$

Again, there is a feasible matching for each scenario which is ensured by (24). The constraint (23) ensures that for all matchings and each vertex, the sums over the outgoing arcs are equal to those fixed by the preliminary matching. Lastly, the objective is again to maximize the value of the worst-case scenario.

We can also formulate the Directed Robust b-Matching Problem in a similar way by relaxing the equality in the perfect matching constraint (24) to a regular matching constraint with inequality. Then, vertices without incoming arcs can also be included in the pre-schedule. However, we will see in the next section that we need the perfect matching constraint to connect this problem to the DRTD Problem. Thus, in this paper, we will only concentrate on perfect matchings.

4 Connecting the Directed Robust Problems

In the deterministic case, the Two-Dose Problem can be reduced to the b-Matching Problem. It is easy to see that the reverse holds true too. We will show something similar for the Directed Robust case. However, we will also examine the maximum time frame between two doses of vaccine $\varepsilon_{\max}$ in the Two-Dose Problem and how it translates to the b-matching Problem. To this end, we introduce the bandwidth of an undirected graph. All of the following results are shown in [6]. Let $G = (V, E)$, then G has bandwidth $k \in \mathbb{N}$ if there is a bijective ordering of the vertices $\pi \colon V \to \{1, \dots, |V|\}$ such that $|\pi(v) - \pi(w)| \leq k$ for each edge $\{v, w\} \in E$. Minimizing the bandwidth can be visualized as ordering the vertices to minimize the length of the edges and is NP-hard. We concentrate on a special case, the directed bandwidth, which is defined on a directed acyclic graph $D = (V, A)$. In addition to the regular bandwidth, each arc (v, w) also has to satisfy $\pi(v) < \pi(w)$. The decision problem of determining if a given graph has a certain bandwidth is also NP-hard.

In the following lemma, we reduce the DRTD Problem to the DRPbM Problem, putting emphasis on the connection between $\varepsilon_{\max}$ and directed bandwidth.

Lemma 1 Each instance of the Directed Robust Two-Dose Problem is equivalent to an instance of the Directed Robust Perfect b-Matching Problem with directed bandwidth at most $\max\{4 \cdot \varepsilon_{\max}, 5\}$.

Proof
Remember the construction of G in advance of Theorem 1. First, we show that this construction can be adapted to the Directed Robust Problem. In the construction of $\mathcal{G}$, we direct the edges as follows: each edge of the form $\{v_i, v_j\}$ is directed from the earlier time step to the later time step. I.e., if $i < j$, we get (v_i, v_j). Additionally, all edges adjacent to v_i^1 and v_i^3 are directed away from these vertices and all edges adjacent to v_i^2 are directed towards v_i^2 for each time step $i = 1, \dots, n$. With these definitions, $h_{v_i} = x_i$ is equal to the number of doses used for first shots in the schedule. In the gadget, it holds $h_{v_i^2} = 0$, $h_{v_i^1} = \beta_{v_i^1}$ and $h_{v_i^3} = \beta_{v_i^3}$ for each feasible solution. Thus, a preliminary matching on $\mathcal{G}$ is equivalent to a pre-schedule in the Two-Dose instance and the correctness of the reduction follows from Theorem 1.

Now, we need to show the upper bound on the bandwidth. To this end, we need to

find an ordering π of V with $|\pi(v) - \pi(w)| \leq 4 \cdot \varepsilon_{\max}$. For $k \in \{0, \ldots, \frac{|V|}{4} = n\}$, we set

$$\begin{aligned}\pi(v_k^1) &= 4 \cdot k + 1\\ \pi(v_k^3) &= 4 \cdot k + 2\\ \pi(v_k^2) &= 4 \cdot k + 3\\ \pi(v_k) &= 4 \cdot k + 4\end{aligned}$$

For the gadget edges, the edges (v_k^3, v_{k+1}^2) maximize the difference in π with $\pi(v_{k+1}^2) - \pi(v_k^3) = 5$. For an appointment arc $(v_i, v_j) \in A$, it holds $\pi(v_j) - \pi(v_i) = 4 \cdot j + 4 - (4 \cdot i + 4) = 4 \cdot (j - i) \leq 4 \cdot \varepsilon_{\max}$. Additionally, all arcs are directed forward with respect to π. Thus, the directed bandwidth of $\mathcal{G}$ can be at most $\max\{4 \cdot \varepsilon_{\max}, 5\}$, where 5 is only maximal if $\varepsilon_{\max} = 1$. □

Next, we show the reverse reduction, again emphasizing the connection between directed bandwidth and $\varepsilon_{\max}$.

Lemma 2 Each directed acyclic instance of the Directed Robust Perfect b-Matching Problem with directed bandwidth k is equivalent to an instance of the Robust Directed Two-Dose Problem with storage capacity $u = 0$ and $\varepsilon_{\max} = k$.

Proof
Let an instance of the DRPbM consist of a directed, acyclic graph $D = (V, A)$, values $c \in \mathbb{R}_+^{|A|}$, and scenarios for the b-vector $\mathcal{B} \subseteq \mathbb{Z}_+^{|V|}$.

Since the graph D has no directed cycles and directed bandwidth k, there is an ordering of its vertices $\pi \colon V \to \{1, \ldots, |V| = n\}$ such that there are no backward arcs (v_j, v_i) and $\pi(j) - \pi(i) \leq k$ for $i < j$. We construct an instance of the Robust Directed Two-Dose Problem with n time steps. Each time step t corresponds to the unique vertex $v \in V$ with $\pi(v) = t$ and the deliveries refer to the capacity accordingly. Next, we set $\varepsilon_{\min} = 1$ and $\varepsilon_{\max} = k$ such that all appointments corresponding to arcs in D are feasible due to the directed bandwidth. Lastly, we set the value of an appointment (i, j) to $c_{\pi^{-1}(i), \pi^{-1}(j)}$ if $(\pi^{-1}(i), \pi^{-1}(j)) \in A$ and to $-M$ otherwise for M sufficiently large. This way, appointments corresponding to pairs of vertices that are not adjacent in D will not be chosen by an optimal solution if there exists a feasible solution for the original instance. Thus, if there is an optimal solution with positive value for the Directed Robust Two-Dose Problem, there is a corresponding solution with the same value for the original instance of the DRPbM Problem. □

Combining both reductions leads to the following result.

Corollary 2 Each instance of the Directed Robust Two-Dose Problem with $\varepsilon_{\max} = k$ and arbitrary storage capacity u can be reduced to an instance of the Directed Robust Two-Dose Problem with $\varepsilon_{\max} \leq \max\{4 \cdot k, 5\}$ and storage capacity $\tilde{u} = 0$.

In an instance of the Directed Robust Two-Dose Problem with $u = 0$, no doses of vaccine can be stored. We say that in these instances, there is no storage. With the Corollary above, the subclass of the Directed Robust Two-Dose Problem without storage is at least as hard as the superclass with arbitrary storage capacity.

5 Complexity of Directed Robust b-Matching

In this section, we show NP-hardness of the DRPbM Problem. With the reduction from the previous section, these results also carry over to the DRTD Problem.

Theorem 4 The Directed Robust Perfect b-Matching Problem is strongly NP-hard for $|\mathcal{B}| \in \mathcal{O}(|V|)$ and directed bandwidth in $\mathcal{O}(|V|)$.

Proof
We reduce from the strongly NP-hard 3-SAT problem. Let $\phi = \bigwedge_{j=1,\dots,m} \psi_j$ be a logical formula with variables $x_1, x_2, \dots, x_p$ and clauses $\phi_1, \dots, \phi_m$. Let $L = \{x_1, \overline{x}_1, \dots, x_p, \overline{x}_p\}$ be the set of all literals.

For each variable x_i, let $D_i = (V_i, A_i)$ be a directed gadget graph with five vertices $d_i, v_{x_i}, v_{\overline{x_i}}, w_{x_i}$ and $w_{\overline{x_i}}$, for each variable x_i. We refer to the two vertices v_{x_i} and $v_{\overline{x_i}}$ as literal vertices. These five vertices are connected through the arcs (d_i, v_{x_i}), $(d_i, v_{\overline{x_i}})$, (v_{x_i}, w_{x_i}) and $(v_{\overline{x_i}}, w_{\overline{x_i}})$. The capacity of each of the three vertices d_i, v_{x_i} and $v_{\overline{x_i}}$ is $b_{d_i} = b_{v_{x_i}} = b_{v_{\overline{x_i}}} = 1$ in each scenario. The other two vertices w_{x_i} and $w_{\overline{x_i}}$ are used to represent the clauses ψ. Thus, we call them clause vertices. The capacity of these vertices is in $\{0, 1\}$ for each scenario. There are m scenarios $\mathcal{B} = \{b^1, \dots, b^m\}$ and the i-th scenario $b^i \in \mathcal{B}$ represents the clause ψ_i. In that scenario b^i, the capacity of the clause vertices corresponding to the literals included in ψ_i is 1 and the capacity of the other clause vertices is 0. For example, if $\psi_i = x_i \vee \overline{x_2} \vee x_p$, then in scenario b^i, the capacity of the clause vertices $w_{x_1}, w_{\overline{x_2}}$ and w_{x_p} is 1 and that of all other clause vertices is 0.

Finally, we add two vertices v_m and v_m^2. The first vertex v_m has incoming arcs from every literal vertex and capacity $b_{v_m} = p - 1$ in each scenario. The second vertex v_m^2 has outgoing arcs to every clause vertex and capacity $b_{v_m^2} = 2$.

The complete digraph $D = (V, A)$ is then given by

$$V = \left(\bigcup_{i=1}^{p} V_i \right) \cup \{v_m, v_m^2\},$$
$$A = \left(\bigcup_{i=1}^{p} A_i \right) \cup \{(v_l, v_m) \colon l \in L\} \cup \{(v_m, w_l) \colon l \in L\}.$$

The arcs $a = (v_l, w_l) \in A$ for $l \in L$ between the literal vertices and the clause vertices have value $c_a = 1$, the other arcs $a' \in A$ have value $c_{a'} = 0$. The complete digraph is depicted in Figure 2. The literal vertices are depicted in the second column and the

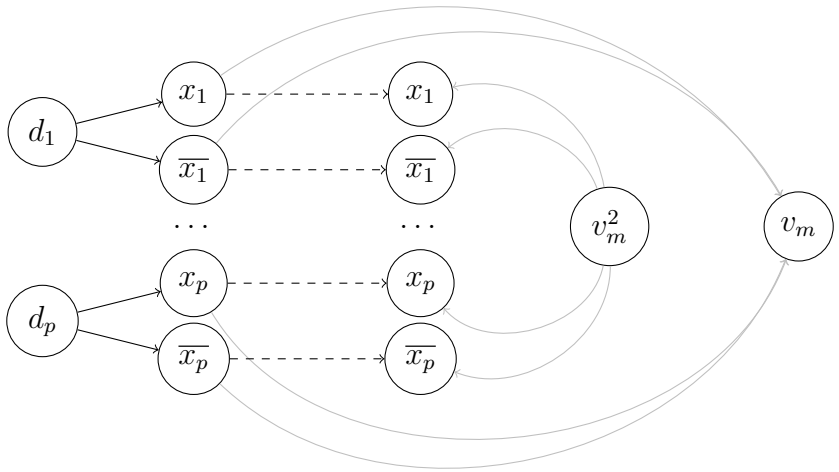


Figure 2: Reduction of SAT to DRPbM

clause vertices in the third column. The dashed arcs have value $c_a = 1$ and the black and gray arcs value $c_a = 0$.

There is a directed robust perfect b-matching with value > 0 for D if and only if ϕ is satisfiable.

We will show, that each preliminary matching induces an assignment which is feasible for a clause ψ_i if and only if there is a matching extending the preliminary matching in scenario b_i. In the first stage of the directed b-matching problem, we set the preliminary matching h, i.e., the sum over all outgoing arcs for each vertex. The vertices $d_i, w_{x_i}, w_{\overline{x_i}}, v_m$ and v_m^2 have either only incoming arcs or only outgoing arcs. For them, the preliminary matching has to be set to 0 if there are only incoming arcs and to their capacity if there are only outgoing arcs. For example, since the capacity of the vertices d_i

is 1 and these vertices have no incoming and two outgoing arcs, it is $h_{d_i} = 1$. This vertex has to be matched with one of the two connected literal vertices. For each variable x_i, exactly one of the two literal vertices has to satisfy the capacity of the vertex d_i and the other vertex has to be matched through an outgoing edge. The preliminary matching h sets this role and fixes, which literal vertex is matched to d_i and which literal vertex can be matched with a clause vertex or v_m. Thus, for each variable only one of two literal vertices is free to be matched with a clause vertex. These free vertices define the assignment: if v_{x_i} is the free vertex, we set x_i to true and if $v_{\overline{x_i}}$ is the free vertex, we set x_i to false. If this assignment satisfies ϕ, then for each clause there is a satisfying variable. Thus, for each clause vertex there is at least one literal vertex that could be matched to one of the clause vertices corresponding to the literals in the clause. The remaining two clause vertices with capacity 1 can then be matched to v_m^2 and the $p-1$ remaining literal vertices with preliminary matching h set to 1 are matched to v_m. Then, the resulting matching has value 1. If there is no such vertex for one clause, the assignment does not satisfy this clause and, by extension, ϕ. Thus, the construction is correct.

The constructed directed graph D is acyclic, has $|V| = 3p + m + 1 \leq 4p$ vertices and directed bandwidth at least $2p \in \mathcal{O}(|V|)$, since $2p$ vertices are adjacent to v_m and all directed towards v_m. Thus, the directed bandwidth is in $\mathcal{O}(|V|)$. □

With Theorem 2, the NP-hardness result also holds for the Two-Dose Problem with linear number of scenarios $|\mathcal{B}| \in \mathcal{O}(n)$ and $\varepsilon_{\max} \in \mathcal{O}(n)$. Also, when the formula ϕ is not satisfiable, there is no feasible directed matching because there is always at least one clause that cannot be matched to any of the p free literal vertices and v_m has capacity $p - 1$. Thus, one free literal vertex cannot be matched and the problem of finding a feasible robust directed b-matching is also strongly NP-hard. This specification can be avoided by adding arcs from all variable vertices to all constraint vertices and setting their value to 0. This way, there is always a feasible matching with value 0.

This result does not implicate any complexity result for the Directed Robust Perfect b-Matching Problem on graphs with constant directed bandwidth $\in \mathcal{O}(1)$ or instances of the Directed Robust Two-Dose Problem with $\varepsilon_{\max} \in \mathcal{O}(1)$. We will now show that strong NP-hardness also applies to the special cases with directed bandwidth at least 8. The reduction is similar to that in the previous proof, only that now we construct a graph with restricted bandwidth.

Theorem 5 The Directed Robust Perfect b-Matching Problem is strongly NP-hard for $|\mathcal{B}| \in \mathcal{O}(|V|)$ and directed bandwidth ≥ 8.

Proof

We again reduce from the strongly NP-hard 3-SAT problem. Let $\phi = \bigwedge_{j=1,\dots,m} \psi_j$ be a logical formula with variables $x_1, x_2, \dots, x_p$, clauses $\phi_1, \dots, \phi_m$ and $L = \{x_1, \overline{x}_1, \dots, x_p, \overline{x}_p\}$

the set of all literals. Let $D = (V, A), \mathcal{B}, c$ be the constructed instance from the previous reduction in the proof of Theorem 4. We will now construct a graph $\tilde{D} = (\tilde{V}, \tilde{A})$ with uncertainty set $\tilde{\mathcal{B}}$ and value $\tilde{c}$ where each directed, perfect $\tilde{b}$-matching is equivalent to a directed, perfect b-matching on D. For the construction, we fix an arbitrary scenario $b \in \mathcal{B}$ and define a new scenario $\tilde{b}$ on $\tilde{D}$. The uncertainty set $\tilde{\mathcal{B}}$ is then given by $\tilde{\mathcal{B}} = \{\tilde{b} \colon b \in \mathcal{B}\}$.

In order to reduce the directed bandwidth, we first take a closer look at why the bandwidth of D is large. We quickly see that ine reason are the vertices v_m and v_m^2. First, we examine v_m. The vertex has p incoming edges. Thus, in each ordering π of the vertices, v_m has an edge to one of these literal vertices v with $\pi(v_m) - \pi(v) \geq p$. Let $V_m = \{v_m, v_{x_i}, v_{\overline{x_i}} \colon i \in \{1, \dots, p\}\}$ the set of these vertices. Then, the subgraph on these vertices $D[V_m]$ is a directed star with center v_m. We construct an equivalent (w.r.t. the DRPbM Problem) directed graph $\tilde{D}[\tilde{V}_m] = (\tilde{V}_m, \tilde{A}_m)$ by adding a gadget as follows:

$$\begin{aligned}
\tilde{V}_m =& V_m \cup \{v_l^1, v_l^2 \colon l \in \{x_1, \overline{x_1}, \dots, x_{p-1}, \overline{x_{p-1}}, x_p\}\} \\
\tilde{A}_m =& \{(v_l, v_l^1), (v_l^2, v_l^1) \colon l \in \{x_1, \overline{x_1}, \dots, x_{p-1}, \overline{x_{p-1}}, x_p\}\} \\
& \cup \{(v_{x_i}^2, v_{\overline{x_i}}^1) \colon i \in \{1, \dots, p-1\}\} \\
& \cup \{(v_{\overline{x_i}}^2, v_{x_{i+1}}^1) \colon i \in \{1, \dots, p-1\}\} \\
& \cup \{(v_{\overline{x_p}, v_m}), (v_{x_p}^2, v_m)\}
\end{aligned}$$

This graph is depicted in Figure 3. The different colorings of the vertices highlights the three types of vertices v_l, v_l^1 and v_l^2 for each literal l. This is later used to visualize the ordering for computing the directed bandwidth. □

We can now define an ordering π_m on the vertices $\tilde{V}_m$ showing that this subgraph has directed bandwidth of 4 instead of p. For each literal vertex $v_{x_i}, v_{\overline{x_i}} \in V \setminus \{v_{x_1}\}$, let $\rho(v_{x_i}) = v_{\overline{x_{i-1}}}$ and $\rho(v_{\overline{x_i}}) = v_{x_i}$. In Figure 3, this represents the literal vertex in the row above. Additionally, let $\mathcal{P}(v_l) = \{\rho(v_l), \rho(\rho(v_l)), \dots, v_{x_1}\}$ the set of all literal vertices above v_l for each literal vertex $v_l \in V$. Then, we can define π_m as

$$\pi_m(v) = \begin{cases} 0 & \text{if } v = v_{x_1} \\ \pi_m(\rho(v)) + 3 & \text{if } v = v_l \\ \pi_m(v_l) + 1 & \text{if } v = v_l^2 \\ \pi_m(v_l) + 2 & \text{if } v = v_l^1 \\ 6p + 4 & \text{if } v = v_m \end{cases}$$

This ordering is visualized in Figure 4. The value for all arcs in the gadget is 0. Next, we set the capacity of the vertices. For all vertices in both subgraphs $v \in \tilde{V}_m$, we set

$$\tilde{b}_v^1 = \begin{cases} b_v & \text{if } v \in V \\ b_m = p - 1 & \text{else.} \end{cases}$$

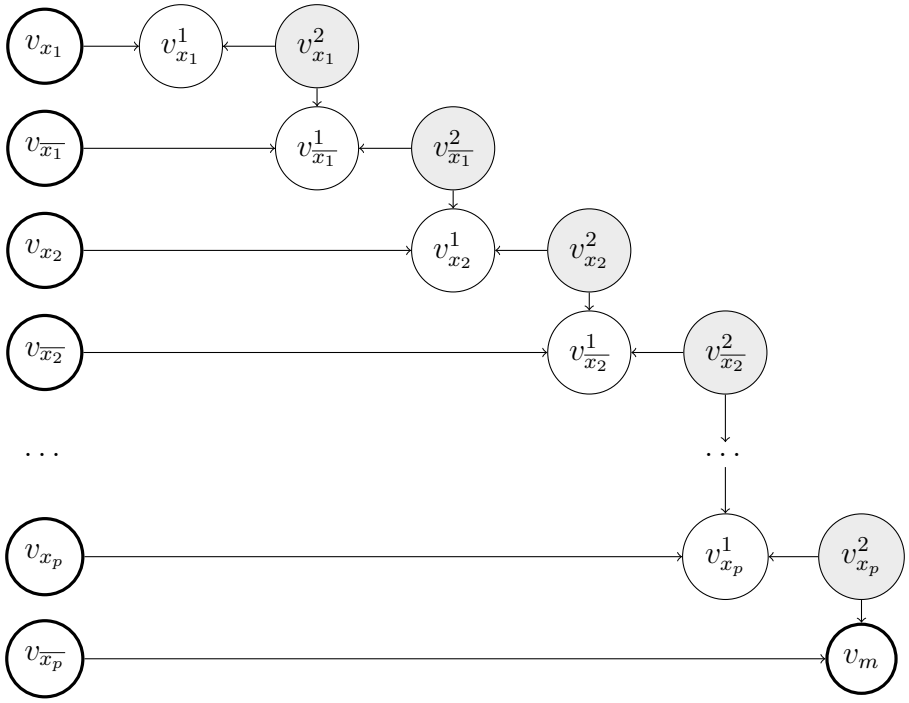


Figure 3: Gadget replacing $G[V_m]$ with reduced bandwidth

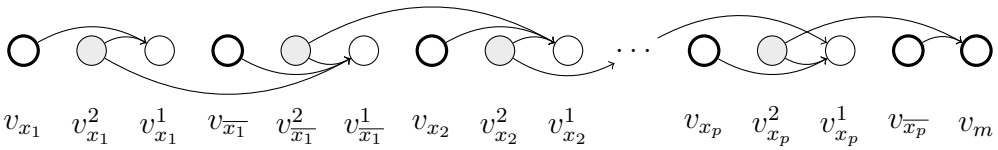


Figure 4: Ordering π_m of the Gadget with reduced bandwidth

Particularly, it is $\tilde{b}^1_{v_m} = b_m = p - 1$.

This gadget is similar to the one used as storage in Theorem 1.

Let $D + \tilde{D}[\tilde{V}_m] = (V \setminus V_m \cup \tilde{V}_m, A \setminus A_m \cup \tilde{A}_m)$ be the graph D where the subgraph $D[V_m]$ is replaced by $\tilde{D}[\tilde{V}_m]$ and $\tilde{b}^1$ the capacity on its vertices where $\tilde{b}^1_v = b_v$ for all $v \in V$. We will now show that there is an equivalent perfect $\tilde{b}^1$-matching on $D + \tilde{D}[\tilde{V}_m]$ and vice versa.

Let m be a perfect matching on D. Then, we define the perfect matching $\tilde{m}$ on $D + \tilde{D}[\tilde{V}_m]$ as follow:

$$\tilde{m}_a = \begin{cases} m_a & \text{if } a \in A \\ m_{(v_l, v_m)} & \text{if } a = (v_l, v^1_l) \\ \sum_{v \in \mathcal{P}(v_l)} m_{(v, v_m)} & \text{if } a = (v^2_{l'}, v^1_l),\ v_{l'} = \rho(v) \\ p - 1 - \sum_{v:\ \rho(v) \leq \rho(v_l)} m_{(v, v_m)} & \text{if } a = (v^2_l, v^1_l) \end{cases}$$

Simple arithmetic operations show that $\tilde{m}$ is a perfect $\tilde{b}^1$-matching on $D + \tilde{D}[\tilde{V}_m]$. Since all added edges have value 0 and the matchings are equivalent on the other edges, they have the same value.

Now, let $\tilde{m}$ be a perfect $\tilde{b}^1$ matching on $D + \tilde{D}[\tilde{V}_m]$. Then, we can define a perfect b-matching on D as

$$m_a = \begin{cases} \tilde{m}_a & \text{if } a \in A \\ \tilde{m}_{(v_l, v_l^1)} & \text{if } a = (v_l, v_m) \end{cases}$$

The perfect matching constraint for all vertices $v \in V \setminus \{v_m\}$ is implied by $\tilde{m}$ being a perfect $\tilde{b}^1$-matching. However,to show the perfect matching constraint for v_m, we first need to show that

$$\tilde{m}_{(v_{l'}^2, v_l^1)} = \sum_{v \in \mathcal{P}(v_l)} m_{(v, v_m)} \quad \forall v_l \in V \setminus \{v_{x_1}\},\ v_{l'} = \rho(v_l).$$

Looking at Figure 3, this means that the value of $\tilde{m}$ on the vertical arcs is equal to the sum over (the value of $\tilde{m}$ on) the outgoing arcs from literal vertices above that vertical arc. This also implies that $\tilde{m}_{(v_{\overline{x_p}}^2, v_m)} = \sum_{v \in \mathcal{P}(v_{\overline{x_p}})} m_{(v, v_m)}$. Thus, for all vertices incident to v_m in $D + \tilde{D}[\tilde{V}_m]$ denoted by $\tilde{\delta}^1(v_m)$ it holds

$$p + 1 = \sum_{v \in \tilde{\delta}^1(v_m)} \tilde{m}_{(v, v_m)} = \tilde{m}_{(v_{x_p}^2, v_m)} + \tilde{m}_{(v_{\overline{x_p}}, v_m)} = \sum_{v_l \in V} m_{(v_l, v_m)}.$$

With this, the perfect matching constraint for v_m would be satisfied.

We now show by induction, that $\tilde{m}_{(v_{x_p}^2, v_m)} = \sum_{v \in \mathcal{P}(v_{\overline{x_p}})} m_{(v, v_m)}$. Since the perfect matching constraint on $v_{x_1}^1$ and $v_{x_1}^2$ is satisfied, it holds $\tilde{m}_{(v_{x_1}^2, v_{x_1}^1)} = p - 1 - \tilde{m}_{(v_{x_1}, v_{x_1}^1)}$ and

$$\tilde{m}_{(v_{x_1}^2, v_{\overline{x_1}}^1)} = p - 1 - \left(p - 1 - \tilde{m}_{(v_{x_1}, v_{x_1}^1)}\right) = \tilde{m}_{(v_{x_1}, v_{x_1}^1)} = m_{(v_{x_1}, v_m)}.$$

We will now show the proposition for any literal Vertex $v_l \in V \setminus \{v_{x_1}\}$ under the assumption that it holds for $\rho(v_l) = v_{l'}$.

From the assumption, we know $\tilde{m}_{(v_{l''}^2, v_{l'}^1)} = \sum_{v \in \mathcal{P}(v_{l'})} m_{(v, v_m)}$ for $v_{l''} = \rho(v_{l'})$. Similar to the start of the induction, we use the perfect matching constraint on $v_{x_{l'}}^1$ and $v_{x_{l'}}^2$ to show the proposition. This way, we get

$$\begin{aligned} \tilde{m}_{(v_{x_{l'}}^2, v_{x_{l'}}^1)} &= p - 1 - \tilde{m}_{(v_{x_{l'}}, v_{x_{l'}}^1)} - \tilde{m}_{(v_{x_{l''}}^2, v_{x_{l'}}^1)} \\ &= p - 1 - m_{(v_{x_{l'}}, v_m)} - \sum_{v \in \mathcal{P}(v_{l'})} m_{(v, v_m)} = p - 1 - \sum_{v \in \mathcal{P}(v_l)} m_{(v, v_m)} \end{aligned}$$

and

$$\begin{aligned}\tilde{m}_{(v^2_{x_{l'}},v^1_{x_l})} &= p-1-\tilde{m}_{(v^2_{x_{l'}},v^1_{x_{l'}})} = p-1-\left(p-1-\sum_{v\in\mathcal{P}(v_l)} m_{(v,v_m)}\right)\\ &= \sum_{v\in\mathcal{P}(v_l)} m_{(v,v_m)}.\end{aligned}$$

This proves the proposition and thus the perfect matching constraint for $\tilde{m}$. Finally, we note that all added gadget vertices in $\tilde{V}_m$ have either only incoming or only outgoing arcs. Thus, each feasible pre-schedule on these vertices is either 0 or equal to the capacity. Using the proof of Theorem 4, it easily follows that there is a directed perfect $\tilde{b}^1$-matching on $D+\tilde{D}[\tilde{V}_m]$ with value >0 if and only if the 3SAT formula ϕ is satisfiable.

We can repeat this construction for the vertex v^2_m. Let $V^2_m = \{v^2_m, w_{x_i}, w_{\overline{x_i}} \colon i \in \{1,\dots,p\}\}$. Similar to V_m, this is also a directed star with center v^2_m. However, all arcs are directed away from v^2_m instead of toward it as in V_m. Thus, we construct a gadget $\tilde{D}[\tilde{V}^2_m] = (\tilde{V}^2_m, \tilde{A}^2_m)$ similar to $\tilde{D}[\tilde{V}_m]$ but reversing all arcs:

$$\begin{aligned}\tilde{V}^2_m =& V^2_m \cup \{w^1_l, w^2_l \colon l \in \{x_1, \overline{x_1}, \dots, x_{p-1}, \overline{x_{p-1}}, x_p\}\}\\ \tilde{A}^2_m =& \{(w^1_l, w_l), (w^1_l, w^2_l) \colon l \in \{x_1, \overline{x_1}, \dots, x_{p-1}, \overline{x_{p-1}}, x_p\}\}\\ &\cup \{(w^1_{\overline{x_i}}, w^2_{x_i}) \colon i \in \{1,\dots,p-1\}\}\\ &\cup \{(w^1_{x_{i+1}}, w^2_{\overline{x_i}}) \colon i \in \{1,\dots,p-1\}\}\\ &\cup \{(w_{v_m,\overline{x_p}}), (v_m, w^2_{x_p})\}\end{aligned}$$

$$\tilde{b}^2_v = \begin{cases} b_v & \text{if } v \in V \\ b^2_m = 2 & \text{else}\end{cases} \qquad \tilde{c}^2_a = \begin{cases} c_a & \text{if } a \in A \\ 0 & \text{else}\end{cases}$$

With the same reasoning as above, the directed bandwidth of this gadget is 4 and there is a directed perfect $\tilde{b}^2$-matching on $D+\tilde{D}[\tilde{V}^2_m]$ with value >0 if and only if the 3SAT formula ϕ is satisfiable. We can order the vertices of $\tilde{V}^2_m$ similar to those of $\tilde{V}_m$ but back-

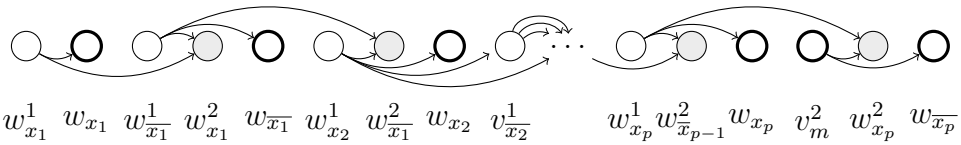


Figure 5: Ordering π^2_m of the second gadget with reduced bandwidth

wards. This results in the ordering π^2_m defined as

$$\pi_m^2(w) = \begin{cases} 0 & \text{if } w = w_{x_1}^1 \\ 1 & \text{if } w = w_{x_1} \\ \pi_m^2(\rho(v)) + 3 & \text{if } w = w_l \\ \pi_m^2(w_l) - 1 & \text{if } w = w_{l'}^2, \; v_{l'} = \rho(v_l) \\ \pi_m^2(w_l) - 2 & \text{if } w = w_l^1 \\ 6p + 2 & \text{if } v = v_m \end{cases}$$

This ordering is depicted in Figure 5. Since $\tilde{V}_m \cap \tilde{V}_m^2 = \emptyset$, we can now define the new reduction graph

$$\begin{aligned} \tilde{D} &= D + \tilde{D}[\tilde{V}_m] + \tilde{D}[\tilde{V}_m^2] \\ &= (V \setminus (V_m \cup V_m^2) \cup \tilde{V}_m \cup \tilde{V}_m^2, \\ &\quad A \setminus (A_m \cup A_m^2) \cup \tilde{A}_m \cup \tilde{A}_m^2) \\ &= (\tilde{V}, \tilde{A}) \end{aligned}$$

with capacity and value defined as

$$\tilde{b}_v = \begin{cases} \tilde{b}_v^1 & \text{if } v \in \tilde{V} \\ \tilde{b}_v^2 & \text{else} \end{cases}$$

$$\tilde{c}_a = \begin{cases} c_a & \text{if } a \in A \\ 0 & \text{else} \end{cases}$$

Using the proofs for both subgraphs $D + \tilde{D}[\tilde{V}_m]$ and $D + \tilde{D}[\tilde{V}_m^2]$, we can now follow that there is a directed perfect $\tilde{b}$-matching on $\tilde{D}$ with value > 0 if and only if the 3SAT formula ϕ is satisfiable.

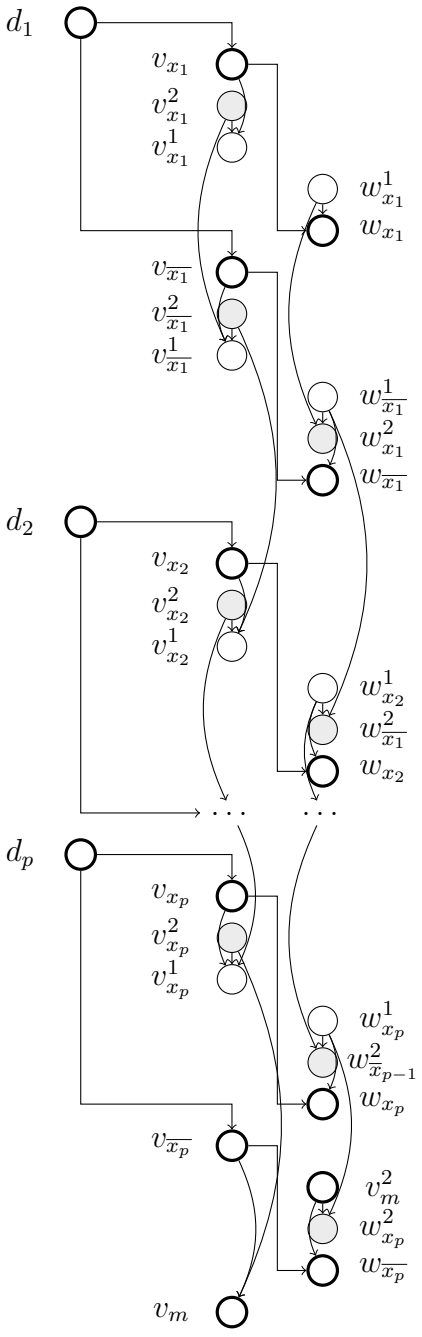


Figure 6: Ordering π of $\tilde{D}$

Finally, we only need to consider the bandwidth of the graph $\tilde{D}$. We define an ordering $\pi : \tilde{V} \to \{0, \ldots, |\tilde{V}|\}$ by combining the orderings π_m and π_m^2 and adding the vertices d_{x_i} in the following way

$$\pi(v) = \begin{cases} 0 & \text{if } v = d_1 \\ 13 \cdot i - 1 & \text{if } v = d_i, i \neq 1 \\ 6 & \text{if } v = v_{\overline{x}_1} \\ \pi(d_i) + 1 & \text{if } v = v_{x_i} \\ \pi(v_{x_i}) + 6 & \text{if } v = v_{\overline{x}_i}, i \neq 1 \\ \pi(v_l) + 1 & \text{if } v = v_l^2 \\ \pi(v_l) + 2 & \text{if } v = v_l^1 \\ 5 & \text{if } v = w_{x_i} \\ 10 & \text{if } v = w_{\overline{x}_1} \\ \pi(d_i) + 6 & \text{if } v = w_{x_i}, i \neq 1 \\ \pi(w_{x_i}) + 6 & \text{if } v = w_{\overline{x}_i}, i \neq 1 \\ \pi(w_l) - 1 & \text{if } v = w_{l'}^2, v_{l'} = \rho(v_l) \\ \pi(w_l) - 2 & \text{if } v = w_l^1 \end{cases}$$

This ordering is depicted in Figure 6. From the definition of π, we can see that the ordering is equivalent on each variable x_i with $i \in \{2, \dots, p-1\}$. Because of this, we can easily find the arcs $a = (v, w) \in \tilde{A}$ maximizing $\pi(w) - \pi(v)$. It holds

$$\max_{(v,w)\in\tilde{A}} \pi(w) - \pi(v) = \pi(v_{\overline{x}_i}^2) - \pi(v_{x_{i+1}^1}) = \pi(v_{\overline{x}_i}^1) - \pi(v_{\overline{x}_i}^2)$$

for each $i \in \{1, \dots, p-1\}$. Thus, the graph $\tilde{D}$ has bandwidth at most 8. This concludes the proof.

We have shown strong NP-hardness for a linear number of scenarios and directed bandwidth ≥ 8. Next, we will show weak NP-hardness for at least two scenarios and directed bandwidth ≥ 3.

Theorem 6 The robust directed b-matching Problem with $|\mathcal{B}| \geq 2$ and directed bandwidth $k \geq 3$ is at least weakly NP-hard.

Proof

We reduce from the weakly NP-hard Partition Problem. Given a finite set of numbers $M \subset \mathbb{N}$, the problem is to find a subset $P \subset M$ with $\sum_{x\in P} x = \sum_{x\notin P} x = \frac{1}{2}\sum_{x\in M} x$. For each element $x \in P$, we define a gadget with five vertices $V^x = \{d_x, v_x^{\text{in}}, v_x^{\text{out}}, w_x^P, w_x^{\overline{P}}\}$ and six arcs $A^x = \{(d_x, v_x^{\text{in}}), (d_x, v_x^{\text{out}}), (v_x^{\text{in}}), (v_x^{\text{in}}, v_c^P), (v_x^{\text{out}}, v_c^P), (v_x^{\text{in}}, v_c^{\overline{P}}), (v_x^{\text{out}}, v_c^{\overline{P}})\}$. The gadget is shown in Figure 7. The value of the arcs (v_x^{in}, w_x^P) and $(v_x^{\text{out}}, w_x^{\overline{P}})$ is x, that of all other arcs is 0. In the Figure, this is represented by the black numbers over each arc. There are 2 scenarios, b^P and $b^{\overline{P}}$. The vertices d_x, v_x^{in} and v_x^{out} have capacity in both

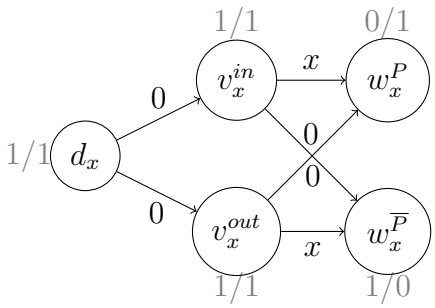


Figure 7: Gadget used for the reduction from Partition

scenarios. The vertex w_x^P has capacity 1 in scenario b^P and capacity 0 in scenario $b^{\overline{P}}$. Accordingly, $w_x^{\overline{P}}$ has capacity 1 in scenario $b^{\overline{P}}$ and capacity 0 in scenario b^P. The complete instance of the DRPbM Problem is given by the Graph $G = (\bigcup_{x\in M} V^x, \bigcup_{x\in M} A^x)$ with cost and both scenarios as defined above. Since the graph consists of n unconnected, isomorphic gadgets, its bandwidth is equal to the bandwidth of the gadgets. It is easy to see that the gadgets have bandwidth 3.

Similar to the last reduction, the preliminary matching fixes the matching partner of d_x. Thus, only one of the two vertices v_x^{in} and v_x^{out} remains free to be matched with either v_x^P or $v_x^{\overline{P}}$ depending on the scenario. If the preliminary matching fixes (d_x, v_x^{in}) as part of the matching, we choose $x \in P$ and vice versa. The first scenario b^P symbolizes the sum of all elements in P. In that case, all free nodes v have to be matched to v_x^P with value 0 if it is a node v_x^{out} and value x if it is a node v_x^{in}. Thus, the total value is the sum of all elements for which the preliminary matching set the node v_x^{in} as the free node. Conversely, the second scenario symbolizes choosing the sum of all elements not in P with corresponding total value of $\sum_{x\notin P} x$. The robust directed b-matching problem always considers the worst case, thus the result is the maximum value of the two scenarios $\sum_{x\in P} x$ and $\sum_{x\notin P} x$. This value is equal to $\frac{1}{2}\sum_{x\in M} x$ if and only if the two sums are equal and P is a feasible solution to the instance of Partition. □

The gadget from the proof is directed and acyclic and the ordering for the directed bandwidth can be easily computed. Thus, it can be formulated as a Robust Directed Two-Dose Problem with 2 scenarios and $\varepsilon_{\max} = 3$ according to Lemma 2. In summary, we could show strong NP-hardness for $|\mathcal{B}| \in \mathcal{O}(n)$ and $\varepsilon_{\max} \in \mathcal{O}(n)$ and at least weak NP-hardness for $|\mathcal{B}| \geq 2$ and $\varepsilon_{\max} \geq 3$.

6 Pseudo-Polynomial Algorithm

In this section, we concentrate on the at least weakly NP-hard case of the Directed Robust Two-Dose Problem. We define a pseudo-polynomial algorithm for the DRTD Problem with $\varepsilon_{\max} \in \mathcal{O}(1)$ and $|\mathcal{B}| \in \mathcal{O}(1)$ that, recursively, computes for each time step $t = n, \dots, 1$ the optimal solution of the subproblem restricted to the time steps $\{t, \dots, n\}$. With Lemma 2, this algorithm can also be used for the DRPbM Problem with directed bandwidth in $\mathcal{O}(1)$ and known ordering. We define the algorithm for instances without storage (storage capacity is $u_t = 0$ for each $t \in T$). With Corollary 2, the algorithm can be transferred to arbitrary storage capacities. Without storage, the vaccination speed constraint is either redundant or the instance has no feasible solution since all delivered doses have to be used in each time step. Thus, we exclude this constraint. Finally, we also assume the value of each appointment $c_{i,j}$ to be a non-negative integer with Remark 1.

We aim to break down the complete problem of finding an optimal pre-schedule on $\{1, \dots, n\}$ into manageable pieces and then solve the problem recursively. To this end, we concentrate on the subproblems of finding partial pre-schedules $x^t \in \mathbb{Z}^n$ on $\{t, \dots, n\}$. Each partial pre-schedules x^t is a vector of dimension n with $x^t_i = 0$ for each time step $i < t$ not in the time frame. The zero-entries at the beginning are included to ensure that for each time steps $i, t, t' \in T$, x^t_i and $x^{t'}_i$ describe the number of first doses administered in the same time step i. Further definitions later in this section will be made accordingly.

Due to the nature of this problem, it does not suffice to only find an optimal partial pre-schedule x^t for each time frame $\{t, \dots, n\}$. Since we want to maximize the value of the solution in the worst-case scenario and this scenario depends on the solution, an optimal partial pre-schedule might not be part of the overall optimal solution. Furthermore, for a pre-schedule with worst-case scenario b, restricting the pre-schedule to $\{t, \dots, n\}$ might lead to a different worst-case scenario b'. Then, it might be necessary to chose a partial pre-schedule on $\{t, \dots, n\}$ which is sub-optimal on scenario b' but optimal on b. However, we cannot easily compute which scenario $b \in \mathcal{B}$ will be the worst-case scenario for the optimal pre-schedule. Thus, we need to find partial solutions with different objective values for each scenario, called target values. Additionally, we cannot simply "cut off" a partial solution at time step t. We need to consider appointments with a first dose before t and a second dose after t. To this end, we need to reserve doses in time steps $\{t, \dots, \min\{n, t + \varepsilon_{\max} - 1\}\} =: S^t$. Second doses for further time steps have a first dose in $\{t, \dots, n\}$ and thus need no further reservation.

The remainder of this section is organized as follows. First, we formally define reserved doses and target values. Then, we will use these states to define the subset of states P^t that we need to keep track of for each time frame $\{t, \dots, n\}$ as well as their

compatible partial pre-schedule. To compute these, we will next introduce a recursive method of computing P^t using P^{t+1}. Finally, we summarize all results in an algorithm computing the optimal solution using pseudo-code and prove its correctness and pseudo-polynomial runtime.

6.1 Defining the States

When breaking down the problem into subproblems, we need to reserve doses in S that will serve as second dose for a first dose before time step t. The subproblem then consists of finding a feasible pre-schedule with corresponding schedules for each scenario consuming all delivered doses in $\{t, \dots n\}$, except for the reserved doses. In an (optimal) solution, the number of reserved doses cannot be identical for each scenario since the number of doses delivered is also not identical but the pre-schedule has to be. Thus, we need to set the number of reserved doses for each scenario $b \in \mathcal{B}$ separately. The matrix of the reserved doses for each scenario $b \in \mathcal{B}$ and time step $i \in S^t$ is denoted by $\left(h^t_{b,i}\right)_{b\in\mathcal{B}, i\in T}$ with $h^t_{b,i} = 0$ for all $i \notin S^t$. For a fixed scenario $b \in \mathcal{B}$, we denote the vector of reserved doses for all time steps S^t as $h^t_b \in \mathbb{Z}^n_+$ with zero-entries for each time step not in S^t.
We denote by H^t_b the set of all possible (feasible and infeasible) reserved doses h^t_b for time step t and scenario $b \in \mathcal{B}$. With $\{0\}^i = \underbrace{\{0\} \times \cdots \times \{0\}}_{i \text{ times}}$, it holds

$$H^t_b = \{0\}^{t-1} \times \{0, \dots, b_t\} \times \cdots \times \{0, \dots, b_{\min\{t+\varepsilon_{\max}-1, n\}}\} \times \{0\}^{n-(\min\{t+\varepsilon_{\max}-1, n\})}.$$

With this, we can define the set of all possible h^t as

$$H^t = \bigtimes_{b\in\mathcal{B}} H^t_b. \tag{26}$$

Two partial pre-schedules x^t and $\tilde{x}^t$ on the time frame $\{t, \dots, n\}$ with the same number of reserved doses in S^t can now be used interchangeably. This means that if x^t is part of a complete pre-schedule x on T (i.e., $x_i = x^t_i$ for each $i \geq t$) then the solution $\tilde{x}$ with x^t exchanged by $\tilde{x}^t$ (i.e., $\tilde{x}_i = x_i$ for $i < t$ and $\tilde{x} = \tilde{x}^t$ for $i \geq t$) is also a feasible solution. However, x and $\tilde{x}$ will not necessarily have the same objective value.

As the worst-case objective value among the scenarios does not suffice to determine the overall optimal solution, we also fix a target value for the schedule in each scenario. In other words, let $f^t \in \mathbb{Z}^{|\mathcal{B}|}_+$ be a vector of target values for each scenario. A partial pre-schedule on $\{t, \dots, n\}$ satisfies f^t if there is a schedule for each scenario $b \in \mathcal{B}$ with objective value f^t_b and fitting the partial pre-schedule. Now, two partial solutions with the same number of reserved doses h^t and the same target value f^t can be used interchangeably without altering the objective value of the solution.

Hence, we need to define the set of all possible target value vectors $f^t \in \mathbb{Z}^{|\mathcal{B}|}$. For a time step $t \in T$ and a scenario $b \in \mathcal{B}$, f_b^t is the target value in scenario b on the problem restricted to $\{t, \dots, n\}$. We denote by F^t the set of all such vectors. Then, it holds

$$F^t = \left\{ 0, \dots, \max_{b \in \mathcal{B}, k,l \in \{t,\dots,n\}} c_{k,l} \cdot \left(\sum_{j=t}^{n} b_j \right) \right\}^{|\mathcal{B}|}. \tag{27}$$

6.2 Fitting, Partial Pre-Schedules

For time step $t \in \{1, \dots, n\}$, reserved doses $h^t \in H^t$ and target objective $f^t \in F^t$, a fitting, partial pre-schedule $x^t(h^t, f^t)$ is a pre-schedule on $\{t, \dots, n\}$ for which schedules s^b exist for each scenario $b \in \mathcal{B}$ which respects the first doses $x^t(h^t, f^t)$, the reserved doses h^t, the delivery constraint and the target value f^t, i.e.

$$\sum_{j \in J(i)} s_{i,j}^b = \left(x^t(h^t, f^t)\right)_i \qquad \forall i \in \{t, \dots, n\},\ b \in \mathcal{B} \tag{28}$$

$$\left(x^t(h^t, f^t)\right)_i + \sum_{\substack{j \in I(i) \\ j \geq t}} s_{j,i}^b \leq b_i - h_{b,i}^t \qquad \forall i \in \{t, \dots, n\},\ b \in \mathcal{B} \tag{29}$$

$$\sum_{i=t}^{n} \sum_{j \in J(i)} c_{i,j} \cdot s_{i,j}^b = f_b \qquad \forall b \in \mathcal{B} \tag{30}$$

For easier notation, we define the set of all pairs of reserved doses and target value $(h^t, f^t) \in H \times F$ with a fitting, partial pre-schedule as P^t for each time step $t \in T$:

$$P^t = \{(h^t, f^t) \in H^t \times F^t \colon \exists x^t(h^t, f^t) \in \mathbb{Z}_+^n \text{ and } s^b \in \mathbb{Z}_+^{n \times n} \text{ for each } b \in \mathcal{B} \text{ such that (28)-(30) are satisfied}\} \tag{31}$$

6.3 Recursive Computation of the Fitting, Partial Pre-Schedules

We now recursively compute for each time step $t \in T$ the sets P^t and for each $(h^t, f^t) \in P^t$ a fitting, partial pre-schedule. Starting at the last time step n, in each time step $t \in T$ we use the (previously computed) set P^{t+1} and partial pre-schedules for time steps $\{t+1, \dots, n\}$ to compute P^t with new fitting, partial pre-schedules on $\{t, \dots, n\}$. For the beginning of the recursion, we examine the last time step $t = n$. In this case, all delivered doses have to be used as a second dose. Thus, the only pair of reserved doses and target value with a fitting, partial pre-schedule is h^n with $h_{b,n}^n = b_n$ and target value $f^n = (0, \dots, 0)$ with only zero-entries. Thus, $P^n = \{(h^n, f^n)\}$ with fitting, partial pre-schedule $x^n(h^n, f^n) = (0, \dots, 0)$.

Next, we examine the recursion steps. Once arrived at time step t, we assume that for time step $t+1$, both the set P^{t+1} and for each $(h^{t+1}, f^{t+1}) \in P^{t+1}$ a fitting, partial pre-schedule $x^{t+1}(h^{t+1}, f^{t+1}) \in \mathbb{Z}_+^n$ on the time steps $\{t+1, \dots, n\}$ is known. The procedure

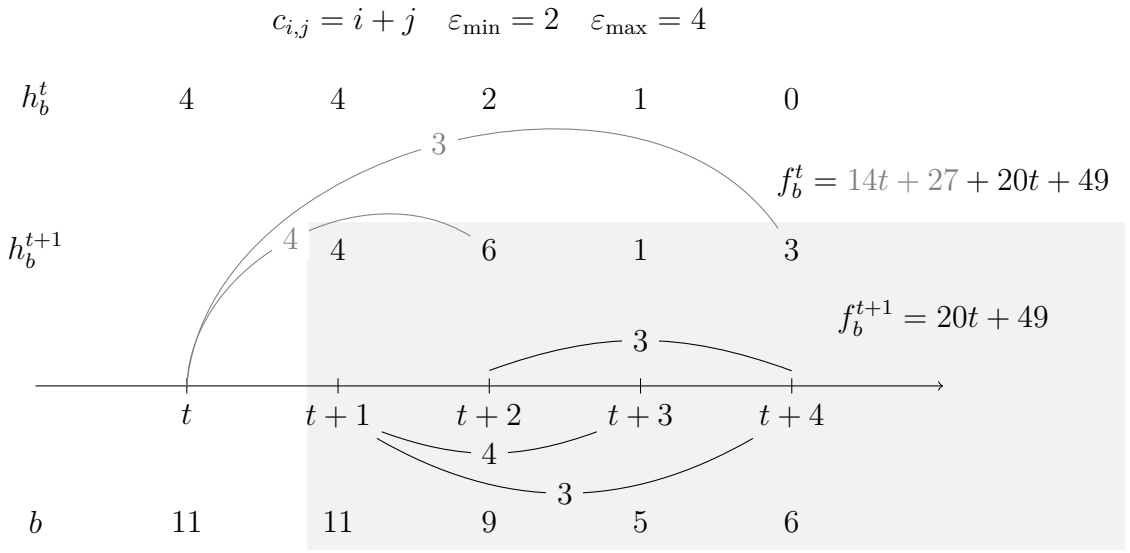


Figure 8: Predecessor illustration

is illustrated exemplarily in Figure 8 for a fixed scenario $b \in \mathcal{B}$. The figure illustrates the extension of a pair $(h_b^{t+1}, f_b^{t+1}) \in P^{t+1}$ to a successor $(h_b^t, f_b^t) \in P^t$ starting at time step t. A fitting schedule for the pair (h_b^{t+1}, f_b^{t+1}) is illustrated in the light gray box. The black edges connecting time steps on the timeline represent a schedule on $\{t+1, \dots, n\}$ for scenario b with reserved doses h_b^{t+1} and objective value f_b^{t+1}. The corresponding pre-schedule then consists of only the first doses in this schedule and has to be the same for all scenarios. A schedule fitting the successor (h_b^t, f_b^t) is then given by adding the gray edges to the already established schedule represented by the black edges. The right end of the gray edges is in the reserved doses h_b^{t+1} to represent that these doses are being used for the added appointments starting in time step t. Accordingly, the reserved doses of the successor $h_b^t := h_{b,i}^{t+1} - s_{t,i}^b$ are the reserved doses of the predecessor h_b^{t+1} minus those used by the gray appointments. The objective value of the schedule starting at time step t is the objective value of the predecessor schedule starting at $t+1$ plus the value of the additional appointments with a first dose at time step t. Note, that the additional appointments with a first dose in time step t have to consume all reserved doses from time step $t + \varepsilon_{\max} = t + 4$, since any leftover doses in this time step could not be used as a second dose for a first dose before t. Thus, reserving them would lead to leftover doses which is infeasible.

Note, that there can be more than one predecessor for a pair (h^t, f^t). Formally, we

define the predecessor-successor relation as follows:

Lemma 3 Let $t \in \{1, \dots, n-1\}$ and $(h^t, f^t) \in H^t \times F^t$. Then, $(h^t, f^t) \in P^t$ if and only if there is a $(h^{t+1}, f^{t+1}) \in P^{t+1}$ which satisfies

(i) $h_i^t = h_i^{t+1}\ \forall i \in S^{t+1} \setminus J(t),\ h_i^t \leq h_i^{t+1}\ \forall i \in J(t)$, and
$h_i^t = 0\ \forall i \in \{t + \varepsilon_{\max}, \dots, n\}$

(ii) $b' - h_{b',t}^t = b - h_{b,t}^t = \sum_{j \in J(t)} \left(h_{b,j}^{t+1} - h_{b,j}^t\right)\ \forall b, b' \in \mathcal{B}$

(iii) $f_b^t = f_b^{t+1} + \sum_{j \in J(t)} c_{t,j} \cdot \left(h_{b,j}^{t+1} - h_{b,j}^t\right)\ \forall b \in \mathcal{B}$

We call (h^{t+1}, f^{t+1}) a *predecessor* of (h^t, f^t). Conversely, we call (h^t, f^t) a *successor* of (h^{t+1}, f^{t+1}). Additionally, for a fitting partial pre-schedule $x^{t+1}(h^{t+1}, f^{t+1})$, the partial pre-schedule $\tilde{x}^t(h^t, f^t)$ with

$$\left(\tilde{x}^t(h^t, f^t)\right)_i = \begin{cases} b - h_{b,t}^t & \text{if } i = t \text{ for any } b \in \mathcal{B} \\ \left(x^{t+1}(h^{t+1}, f^{t+1})\right)_i & \text{else} \end{cases} \tag{32}$$

is fitting for (h^t, f^t).

Proof
First, we note that $(\tilde{x}^t(h^t, f^t))_t$ is well-defined by (32) since condition (ii) in the lemma implies that this value is the same for each scenario $b \in \mathcal{B}$. Let $t \in T$ and $\tilde{P}^t \subseteq H^t \times F^t$ be the set of all pairs (h^t, f^t) for which a predecessor $(h^{t+1}, f^{t+1}) \in P^{t+1}$ exists.

Assume, there is a pair $(h^t, f^t) \notin \tilde{P}^t$ for which a fitting, partial pre-schedule $x^t(h^t, f^t)$ and schedules s^b for each scenario $b \in \mathcal{B}$ exists (and thus $(h^t, f^t) \in P^t$). We define (h^{t+1}, f^{t+1}) as

$$h_{b,i}^{t+1} = \begin{cases} h_{b,i}^t & \text{if } i \in S \setminus J(t) \\ h_{b,i}^t + s_{t,i}^b & \text{if } i \in J(t) \end{cases} \text{ and } f_b^{t+1} = f_b^t - \sum_{j \in J(t)} c_{t,j} \cdot s_{t,j}^b. \tag{33}$$

By definition, x^{t+1} with $x_i^{t+1} = x_i^t$ for all $i \in \{t+1, \dots, n\}$ is a fitting, partial pre-schedule for (h^{t+1}, f^{t+1}). By induction, it holds $(h^{t+1}, f^{t+1}) \in P^{t+1}$. Further, condition (ii) holds by construction. However, the pair also satisfies conditions (i) and (iii) by definition and is a predecessor of (h^t, f^t). Thus, it has to be $(h^t, f^t) \in \tilde{P}^t$.

Now, let $(h^t, f^t) \in \tilde{P}^t$ and $\tilde{x}^t(h^t, f^t)$ as defined in the lemma. Additionally, let (h^{t+1}, f^{t+1}) be their predecessor with fitting, partial pre-schedule $x^{t+1}(h^{t+1}, f^{t+1})$ and schedules s^b for each scenario $b \in \mathcal{B}$. We now define schedules on $\{t, \dots, n\}$ as $\tilde{s}^b$ for each scenario $b \in \mathcal{B}$ with

$$\tilde{s}_{i,j}^b = \begin{cases} s_{i,j}^b & \text{if } i \neq t \\ h_j^{t+1} - h_j^t & \text{if } i = t \text{ and } j \in J(i) \end{cases}. \tag{34}$$

We now show that these schedules are feasible by showing that they satisfy conditions (28)-(30). Let $b \in \mathcal{B}$ and $i \in \{t+1, \dots, n\}$. The first dose constraints (28) ensures that the number of first doses used by the schedules is equal to the number of first doses fixed by the pre-schedule. Since the condition holds for the schedules s^b starting at $t+1$, this is satisfied because

$$\sum_{j \in J(t)} \tilde{s}^b_{t,j} \overset{(34)}{=} \sum_{j \in J(t)} \left(h^{t+1}_{b,j} - h^t_{b,j}\right) \overset{\text{(ii)}}{=} b_t - h^t_{b,t} \overset{(32)}{=} \left(\tilde{x}^t(h^t, f^t)\right)_t$$

$$\sum_{j \in J(i)} \tilde{s}^b_{i,j} \overset{(34)}{=} \sum_{j \in J(i)} s^b_{i,j} \overset{(28)}{=} \left(x^{t+1}(h^{t+1}, f^{t+1})\right)_i \overset{(32)}{=} \left(\tilde{x}^t(h^t, f^t)\right)_i \quad \forall i > t.$$

The delivery constraint (29) ensures that in each time steps, the number of doses used is equal to the number of doses delivered minus the number of doses reserved. Let $\mathbb{1}_{i \in J(t)} = 1$ if $i \in J(t)$ and $\mathbb{1}_{i \in J(t)} = 0$ otherwise. Then, it holds

$$\begin{aligned}\sum_{j \in J(t)} \tilde{s}^b_{t,j} + \sum_{\substack{j \in I(t) \\ j \geq t}} \tilde{s}^b_{j,t} &= \left(\tilde{x}^t(h^t, f^t)\right)_t + 0 \\ &\overset{(32)}{=} b_t - h^t_{b,t}\end{aligned}$$

and for $i > t$

$$\begin{aligned}\sum_{j \in J(i)} \tilde{s}^b_{i,j} + \sum_{\substack{j \in I(i) \\ j \geq t}} \tilde{s}^b_{j,i} &= \left(\tilde{x}^t(h^t, f^t)\right)_i + \sum_{\substack{j \in I(i) \\ j > t}} \tilde{s}^b_{j,i} + \tilde{s}^b_{t,i} \cdot \mathbb{1}_{i \in J(t)} \\ &\overset{(32)}{=} \left(x^{t+1}(h^{t+1}, f^{t+1})\right)_i + \sum_{\substack{j \in I(i) \\ j > t}} s^b_{j,i} + \tilde{s}^b_{t,i} \cdot \mathbb{1}_{i \in J(t)} \\ &\overset{(29)}{=} b_i - h^{t+1}_{b,i} + \tilde{s}^b_{t,i} \cdot \mathbb{1}_{i \in J(t)} \\ &\overset{(33)}{=} b_i - h^t_{b,i}.\end{aligned}$$

The target value constraints (30) ensures that the target value is equal to the objective value of each of these schedules. This is satisfied by the schedules because

$$\begin{aligned}\sum_{i=t}^{n} \sum_{j \in J(i)} c_{i,j} \cdot \tilde{s}^b_{i,j} &\overset{(34)}{=} \sum_{j \in J(t)} c_{t,j} \cdot (h^{t+1}_j - h^t_j) + \sum_{i=t+1}^{n} \sum_{j \in J(i)} c_{i,j} \cdot s^b_{i,j} \\ &\overset{(30)}{=} \sum_{j \in J(t)} c_{t,j} \cdot (h^{t+1}_j - h^t_j) + f^{t+1}_b \overset{\text{(iii)}}{=} f^t_b.\end{aligned}$$

This shows that the schedule $\tilde{s}^b$ are fitting for $\tilde{x}^t(h^t, f^t)$ and have value f^t_b in each scenario. Thus, $\tilde{x}^t(h^t, f^t)$ is a fitting, partial pre-schedule for (h^t, f^t). This shows $\tilde{P}^t = P^t$ and concludes the proof. □

6.4 The Algorithm

For each pair $(h^t, f^t) \in H^t \times F^t$, we can find a predecessor by checking for each pair $(h^{t+1}, f^{t+1}) \in P^{t+1}$ if the three conditions are met. These definitions naturally lead to a recursive algorithm, which is presented in Algorithm 1. The main result is stated in Theorem 7.

Theorem 7 Let $(\varepsilon_{\min}, \varepsilon_{\max}, n, c, u, \nu, \mathcal{B})$ with $\varepsilon_{\max} \in \mathcal{O}(1)$ and $|\mathcal{B}| \in \mathcal{O}(1)$ be an instance of the Directed Robust Two-Dose Problem. Then, the optimal pre-schedule is given by $x^1((0,\dots,0), \tilde{f}^1)$ with

$$\tilde{f}^1 = \underset{((0,\dots,0),f^1)\in P^1}{\arg\max} \min_{b\in\mathcal{B}} f_b^1 \tag{35}$$

It can be computed in pseudo-polynomial time.

Proof
First, we show correctness of the algorithm. By definition, the computation of P^n is correct. With Lemma 3, it follows that the algorithm computes each P^t correctly. Assuming P^1 is the set of all pairs (h^1, f^1) for which a fitting pre-schedule exists, $x^1((0,\dots,0), \overline{f}^1)$ is obviously an optimal solution. If there was a better solution, the corresponding $f^1 \in F^1$ would also satisfy $((0,\dots,0), f^1) \in P^1$ and thus (35) would be at least as good. The choice of $h^1 = (0,\dots,0)$ is necessary since the storage capacity is 0 and thus, all doses have to be used up when they are delivered. With this, the proof of correctness is finished.

To show a pseudo-polynomial runtime, we need to calculate the number of loop iterations in Algorithm 1 in the worst case as checking the conditions can be done in polynomial time. This is equal to

$$\sum_{i=1}^{n-1} |H^t \times F^t| \cdot |P^{t+1}| \leq \sum_{i=1}^{n-1} |H^t \times F^t| \cdot |H^{t+1} \times F^{t+1}| < (n-1) \cdot \max_{t\in T} |H^t \times F^t|^2.$$

Let

$$b_{\max} = \max_{b\in\mathcal{B}, i\in T} b_i, \; B^t_{\max} = \max_{b\in\mathcal{B}} \sum_{i=1}^{t} b_i \text{ and } c^t_{\max} = \max_{k,l\in\{t\dots,n\}} c_{k,l}.$$

Then, it is $|H^t| \in \mathcal{O}(b_{\max}^{\varepsilon_{\max}\cdot|\mathcal{B}|})$ and $|F^t| \in \mathcal{O}((c^t_{\max} \cdot B^t_{\max})^{|\mathcal{B}|})$ per definition (26) and (27). With $\varepsilon_{\max} \in \mathcal{O}(1)$ and $|\mathcal{B}| \in \mathcal{O}(1)$, the number of elements in $H^t \times F^t$ is pseudo-polynomial in the input for every $t \in T$. Thus, there is only a pseudo-polynomial number of loop iterations, which can be computed in polynomial time. Finally, computing the optimal solution (35) for a given set P^1 can be done by checking each element of P^1 which can be done in pseudo-polynomial time. Thus, the runtime of the algorithm is psedo-polynomial. □

Algorithm 1 Exact algorithm for the Directed Robust Two-Dose Problem

Require: time frame $n = \{1, \dots, n\}$, second dose time frame $(\varepsilon_{\min}, \varepsilon_{\max})$, objective Values $c \in \mathbb{Z}_+^{n \times n}$, uncertainty set for the deliveries $\mathcal{B} = \{b^1, \dots, b^k\} \subseteq \mathbb{Z}_+^n$

H^t according to (26) for each $t \in \{1, \dots, n-1\}$
F^t according to (27) for each $t \in \{1, \dots, n-1\}$
$P^t = \emptyset$ for each $t \in \{1, \dots, n-1\}$
$P^n = \{((h_b^n)_{b \in \mathcal{B}}, (0, \dots, 0))\}$, where $h_b^n = (0, \dots, 0, b_n) \in \mathbb{Z}_+^n$
$x^n(h^n, (0, \dots, 0)) = (0, \dots, 0) \in \mathbb{Z}_+^n$
for t=n-1,...,1 **do**
    for $(h^t, f^t) \in H^t \times F^t$ **do**
        for $(h^{t+1}, f^{t+1}) \in P^{t+1}$ **do**
            # test if (h^{t+1}, f^{t+1}) is predecessor of (h^t, f^t) with conditions (3)
            if not $h_i^t = h_i^{t+1}\ \forall i \in S^{t+1} \setminus J(t)$ **then**
                # $(h^{t+1}, f^{t+1}) \in P^{t+1}$ is not a predecessor, continue testing next pair
                continue;
            end if
            if not $h_i^t \geq h_i^{t+1}\ \forall i \in J(t)$ **then**
                continue;
            end if
            if not $b' - h_{b',t}^t = b - h_{b,t}^t = \sum_{j \in J(t)} h_{b,j}^t - h_{b,j}^{t+1}\ \forall b, b' \in \mathcal{B}$ **then**
                continue;
            end if
            if not $f_b^{t+1} = f_b^t - \sum_{j \in J(t)} c_{t,j} \cdot \left(h_{b,j}^{t+1} - h_{b,j}^t\right)\ \forall b \in \mathcal{B}$ **then**
                continue;
            end if
            # all conditions are met, predecessor found
            $P^t = P^t \cup (h^t, f^t)$
            $x^t(h^t, f^t) = (0, \dots, 0, b^1 - h_{b^1,t}^t, (x^{t+1}(h^{t+1}, f^{t+1}))_{t+1}, \dots, (x^{t+1}(h^{t+1}, f^{t+1}))_n)$
            break; # start testing next $(h^t, f^t) \in H^t \times F^t$
        end for
    end for
end for
opt_value $= \underset{((0,\dots,0), f^1) \in P^1}{\arg\max} \min_{b \in \mathcal{B}} f_b^1$
opt_pre-schedule $= x^1((0, \dots, 0), \text{opt\_value})$ **return** opt_pre-schedule

7 Computational Study

In this section, we introduce the results of a computational study with the goal of better understanding the properties of the Directed Robust Two-Dose Problem introduced in Section 3.2. The problem was stated as a Mixed Integer Problem following (13)-(20) and solved using Gurobi. The scenarios used were randomly generated with bounds as well as total number of doses delivered, following the scheme at the beginning of the COVID-19 pandemic in Germany. All numbers used in this section were taken from [5]. We set $n = 77$ which includes 53 weeks or one year with planned deliveries and an additional $2 \cdot 12$ weeks for delayed deliveries and to use up all remaining doses of vaccine. In each generated scenario, $172,219,154$ doses are delivered in total. This is equal to the number of vaccine doses delivered to Germany from the beginning of the COVID-19 Pandemic up to the end of 2021. We set the storage capacity to the total number of doses since storage capacity was never a problem in Germany. In the first 13 time steps, the upper bound on the vaccination speed is set to about 3.08 Million, which is the number of patients the opened vaccination centers could vaccinate in a week. After that, some regular doctors were also allowed to administer vaccinations to patients and we can only estimate the upper bound using actually given doses. Since the numbers

time frame	vaccination speed bound/day
1-13	439,000
14-16	800,000
17-18	1,200,000
19-22	1,500,000
23-49	1,510,000
50-n	1,600,000

Figure 9: Upper Bound on Vaccination Speed per Time Step

drawn from [5] count per day and we set each time step to represent a week, we multiply theses bounds by 7.

We compare two different objective functions c^1 and c^2. Both represent the expected gain of immunity throughout the society by a vaccination at a certain appointment. In the first objective function, the value of a feasible appointment (i, j) is equal to $j - i$ times the expected efficacy of one dose only plus $n - j$ times the expected efficacy of both doses of vaccine. The efficacies were taken from the survey [14] concerning the vaccine Corminaty. Since an earlier first dose creates more value than an earlier second dose, an optimal solution for a deterministic problem with this value will prioritize giving first doses over second doses. We reversed this for c^2 by adding a decay rate for the efficacy

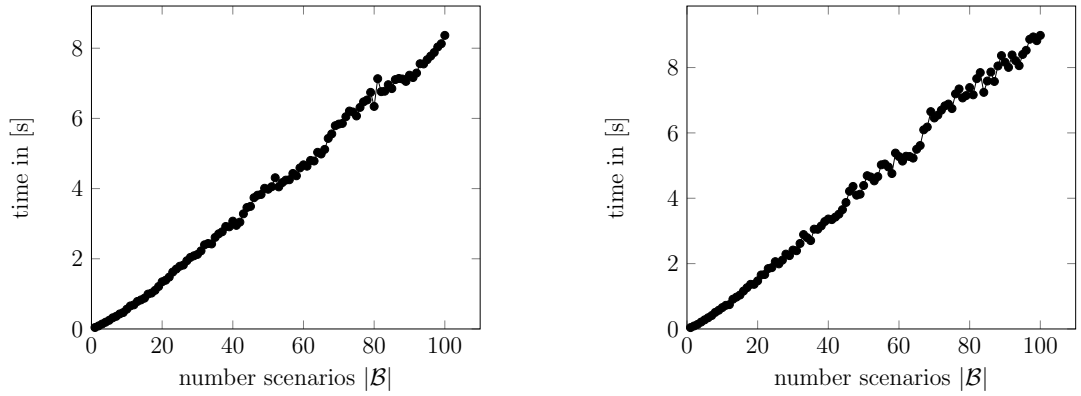
of one dose. With this change, an optimal solution for the deterministic problem with c^2 will prioritize finishing with a second dose over partial immunity by one dose only.

The scenarios are computed randomly. We start of with a scenario b^0 which has equal deliveries of $3,249,418$ doses per time step for each time step $\{1,\dots,53\}$. Then, we randomly delay the arrival of these doses. The last time step with delayed deliveries is $t = 65$.

We begin by examining changes caused by the number of scenarios $|\mathcal{B}|$. We will examine four characteristics. The first two are computation speed (Section 7.1) and objective value (Section 7.2). Then, for the optimal pre-schedule of each instance, we will compute the optimal objective value for a schedule realizing that pre-schedule in the scenario $b^{\min}$ minimizing the total number of doses previously delivered in each time step as defined by (12) (cf. Section 7.3). Lastly, in Section 7.4 we compare the percentage of instances whose pre-schedule is feasible for a newly generated scenario thus measuring the robustness against delays that were not included in $\mathcal{B}$. For each $|\mathcal{B}| \in \{1,\dots,100\}$, we generated 100 different sets of scenarios and computed the characteristics for both value functions c^1 and c^2.

7.1 Computation times

We start with the correlation between the number of scenarios $|\mathcal{B}|$ and the computation time which is shown in Figure 10. We present the results for both value functions c^1 and c^2 independently. We can see that the average computation time rises nearly linear with



(a) first value function c^1 (b) second value function c^2

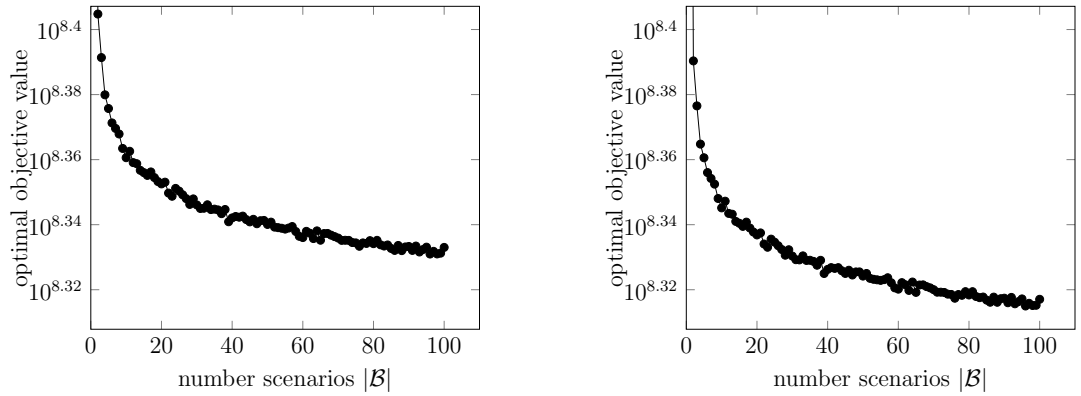
Figure 10: Speed profile for different number of scenarios

the number of scenarios for both value functions. Since both the number of variables and the number of constraints are linear in $|\mathcal{B}|$, this seems surprising. However, it becomes much clearer when looking at the influence of adding a scenario on the problem. Let $\mathcal{B}$ be

a set of scenarios and $b \notin \mathcal{B}$ a new, randomized scenario. If there is a scenario $\tilde{b} \in \mathcal{B}$ with $\sum_{i=t}^{n} b_i \geq \sum_{i=t}^{n} \tilde{b}_i$, then we could get b by delaying doses in scenario $\tilde{b}$. Moreover, each (pre-)schedule feasible for $\tilde{b}$ is also feasible for b. Thus, the scenario b is redundant and the optimal solution for $\mathcal{B} \cup \{b\}$ is the same as the optimal solution for $\mathcal{B}$. Generating a redundant scenario becomes more likely for higher $\mathcal{B}$. Thus, the computation time increases slower in the number of scenarios than expected. We can also see that even for $|\mathcal{B}| = 100$, the problem can be solved in under 9 seconds on average.

7.2 Objective Value

Next, we compare the optimal objective value depending on the number of scenarios and the value function. Note that the more scenarios are considered, the more conservative the objective function will be and thus the value decreases. This is shown in Figure 11. Note that in this figure, we used a logarithmic scale on the y-axis to better visualize the results. At first, we can see that the objective value for c^1 is always higher than that



(a) first value function

(b) second value function

Figure 11: Average Optimal Objective Value on a Logarithmic Scale

for c^2. This is due to the fact that the value of appointment in c^2 is lower because it includes decay of the gained immunity through the first dose.

However, they show the same behavior when considering the number of scenarios. For low number of scenarios, the objective value drops fast when adding scenarios. For larger numbers of scenarios, this process slows down considerably, even on the logarithmic scale. To understand this progress better, we need to take a closer look at the influence of adding a new scenario to $\mathcal{B}$. We say that a scenario $b^* \in \mathcal{B}$ dominates another scenario $b \in \mathcal{B}$ if for every time step $t \in T$, it is $\sum_{i=1}^{t} b_i^* \geq \sum_{i=1}^{t} b_i$. It is easy to see that each (pre-)schedule feasible in b^* is also feasible in b. This also implies that the optimal objective value of finding a schedule in scenario b is always worse than in b^*. Thus, if a scenario

$b \in \mathcal{B}$ is dominated by another scenario in $\mathcal{B}$, we call it redundant and can exclude this scenario and use $\mathcal{B} \setminus \{b\}$ without changing the result. For randomly generated sets $\mathcal{B}$, the probability of generating a redundant scenario is higher the more scenarios already generated. Thus, the optimal objective value falls slower with increasing number of scenarios. The decreasing values also emphasize the importance of reliable delivery plans. If only few scenarios are sufficient, magnitudes of objective value can be achieved.

7.3 Price of Robustness

Next, we examine the solution for the minimum scenario $b^{\min}$ in comparison to the worst case solution, which is shown in Figure 12. Through this, we examine the so-called price of robustness: the loss in objective value by including more scenarios. However, it also describes a gain of the Directed Robust Two-Dose Problem compared to the Fixed Robust Two-Dose Problem introduced in Section 3.1, where only the minimum scenario $b^{\min}$ was utilized.

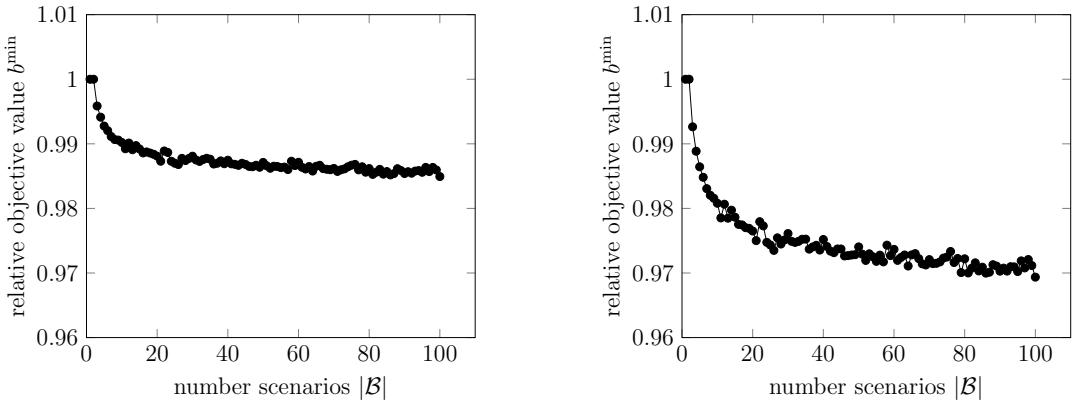
According to (12), the minimum scenario $b^{\min}$ satisfies

$$\sum_{i=1}^{t} b_i^{\min} = \min_{b \in \mathcal{B}} \sum_{i=1}^{t} b_i \qquad \forall t \in T.$$

In each time step, it represents the worst-case availability of vaccine up to that time step. For each randomized instance, we first computed an optimal solution for the generated uncertainty set $\mathcal{B}$, including the optimal pre-schedule x^*. Then, we calculate the value of an optimal schedule for the minimum scenario $b^{\min}$ satisfying the computed pre-schedule x^*. We call this the optimal objective value of $b^{\min}$ in $\mathcal{B}$. We now compare this value relative to the optimal objective value of that instance. Note, that even though $b^{\min}$ is not necessarily in $\mathcal{B}$, there is always a feasible schedule for $b^{\min}$ satisfying the pre-schedule x^*.

From the definition of $b^{\min}$ we know that each schedule feasible for $b^{\min}$ is also feasible for each scenario in $\mathcal{B}$. Thus, the optimal objective value of $b^{\min}$ is not larger than that of any other scenario $b \in \mathcal{B}$ and thus smaller or equal than the optimal objective value of the complete instance. Because of this, the relative objective value of $b^{\min}$ is at most 1. If $b^{\min} \in \mathcal{B}$, it automatically holds that the optimal objective value of $b^{\min}$ is greater or equal to the worst-case optimal value in $\mathcal{B}$. Thus, equality holds and the relative optimal objective value is equal to 1. This is of course always the case for $|\mathcal{B}| = 1$. However, even for smaller sets of up to 20 scenarios, this is rarely the case and never happens for bigger $|\mathcal{B}|$. This also causes the fast drop in the relative objective value for small values of $|\mathcal{B}|$ in both objective functions.

We again see a difference between both objective functions, the curve for the second objective function c^2 favoring second doses over first doses is lower than that of the first



(a) first value function

(b) second value function

Figure 12: Relative Value of Scenario $b^{\min}$ relative to optimal objective value

objective function. This time, this is not due to the fact that the objective values of c^2 are always lower than those of c^1 because all results are relative. The reason for this difference lies in the different value of first and second doses in both value functions. In c^1, the first dose is favored and an earlier first dose is worth more than an earlier second dose. This also implies that delaying a second dose for a fixed first dose does not change much in terms of value. Opposed to this, c^2 favors an earlier second dose, so delaying a second dose for a fixed first dose will lower the value of the appointment more strongly. In the DRTD Problem, the time steps for all first doses are set through the pre-schedule and thus the same for all scenarios. Only the times of the second doses vary between scenarios. Thus, for the first value function c^1, the difference of optimal objective value between scenarios is smaller than for the second value function c^2. Reversely, this also means that the objective value for each scenario in $\mathcal{B}$ is higher compared to that of $b^{\min}$ for the second objective function. This may be an advantage if $b^{\min} \notin \mathcal{B}$ or the scenarios in $\mathcal{B}$ are spread out and $b^{\min}$ has much later deliveries than most scenarios in $\mathcal{B}$.

7.4 Realized Robustness

Regardless how many scenarios are in $\mathcal{B}$, it is unlikely that the real delays are precisely described by one of these input scenarios. Therefore, we would like to investigate how "robust" the solutions are for scenarios not part of $\mathcal{B}$. In earlier studies, this evaluation is known as realized robustness [8]. In our case, we would like to know how robust the computed pre-schedule is for new scenarios. To measure the effect of the number of scenarios used in the robust optimization on the realized robustness, we generate one set $\mathcal{B}$ of 100 scenarios, order them arbitrarily, and consider the instances with the first k scenarios for $k \in \{1, \ldots, 100\}$. Furthermore, we generate a set $\mathcal{N}$ of 100 new scenarios

to evaluate the robust solutions. To this end, for each of the 100 instances we compute the optimal pre-schedule and then determine the number of scenarios in $\mathcal{N}$ for which a feasible schedule respecting the optimal pre-schedule exists. We repeated this 100 times. The scenarios in $\mathcal{N}$ were generated such that the overall delay on them is not bigger than the delay on $b^{\min}$ for the starting set $\mathcal{B}$. We define the delay of a scenario $\delta(b)$ as the sum of the number of time steps each dose is delayed compared to the starting scenario b^0:

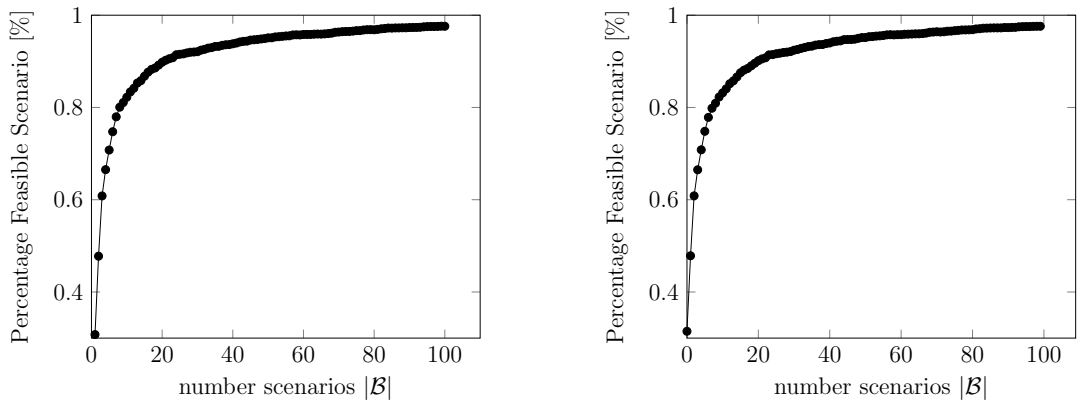
$$\delta(b) = \sum_{t=1}^{n} (n-i) \cdot (b_i^0 - b_i).$$

Let b^{last} be the scenario delivering all doses at the last possible time step $t = 65$. Then, $\delta^{\max} := \delta(b^{\text{last}}) = \sum_{t=1}^{n} (n-i) \cdot b_i^0$ is the maximum delay possible. The corresponding scenario would deliver all doses at the last possible time step. Then, the relative delay $\Gamma(b)$ is given by

$$\Gamma(b) = \frac{\delta(b)}{\delta^{\max}}.$$

The unique scenario with $\Gamma(b) = 1$ is b^{last} and conversely, b^0 is the only scenario with $\Gamma(b) = 0$. We generate the scenarios in $\mathcal{B}$ with random delays. However, scenarios with a higher delay than $\delta(b^{\min})$ have a high probability of being infeasible.

To avoid this behavior, we instead assume that the new scenarios in $\mathcal{N}$ do not have more delay than the scenarios in the starting set $\mathcal{B}$ with 100 scenarios. In reality, this would suggest that the expected scenarios in $\mathcal{B}$ were chosen with a realistic delay. Formally, each $b \in \mathcal{N}$ has at most the same delay as $b^{\min}$, thus $\delta(b) \leq \delta(b^{\min})$. The results are presented in Figure 13.



(a) first value function

(b) second value function

Figure 13: Percentage of Feasible Scenarios in $\mathcal{N}$

We can see for both value functions that the number of feasible, newly generated

scenarios rises with the number of scenarios. Overall, the curve is very step for small sets of $\mathcal{B}$. At about 20 scenarios, the curve flattens and rises much slower. Even in the worst case with 1 scenario, more than 50% of the instances were feasible with the both value functions.

When comparing the two value functions, we would have expected to see better results for the second value function. For c^2, in an optimal, deterministic solution the second dose will be given as early as possible since the efficacy of the first dose only diminishes over time if no second dose is given. Due to this, most used appointments will have a time difference of close to $\varepsilon_{\min}$ between first and second dose. In the directed robust Two-Dose Problem, the time of all first doses is fixed through the pre-schedule. Since minimizing the time between both doses is optimal here, the pre-schedule is chosen such that the second dose can be given as early as possible in each scenario. Now, if some doses are delayed and arrive later than expected, we can simply delay the second dose and still get a feasible solution, only sacrificing some value. In contrast, for the first value function c^1, in the deterministic case it is optimal to delay the second dose as much as possible. In this case, if some doses are delayed, we cannot simply give the second dose later since it is already scheduled for the latest possible time spot. However, the computations do not support this theory. Considering this property, both value function behaved the same on 99.83% of all computed instances. This indicates that the probability of generating a scenario feasible for only one of the two value functions is very slim. Thus, the choice of the value function has very little effect on the realized robustness of the solution.

To summarize, we can see that a higher number of scenarios hedges against uncertainties even for scenarios not originally considered in $\mathcal{B}$. However, the computation time rises linearly with the number of scenarios. Thus, compromises need to be made between robustness and computation time. If the delay in the considered scenarios is chosen well, a small number of scenarios suffices to reach good robustness against unforeseen scenarios. Considering the two value functions, we can say that c^1 leads to a higher worst-case value but also to a lower relative value compared to the minimum scenario $b^{\min}$ while c^2 has a higher relative value compared to $b^{\min}$ but has a lower absolute value. In terms of realized robustness of the solution, both value functions behave very similar.

8 Conclusion

In this paper, we concentrated on a version of the Robust Two-Dose Scheduling problem. The problem deals with the question of scheduling vaccinations when each patient requires two doses and the deliveries are uncertain and can be delayed. We showed that the problem is weakly NP-hard for a linear number of scenarios and short times

between two doses of vaccines and strongly NP-hard if the time between both doses or the number of scenarios can be linear in the number of time steps. We also generalized the problem to a Robust b-Matching Problem with comparable complexity. Next, we introduced a pseudo-polynomial algorithm for the weakly NP-hard case. Finally, in the computational study, we showed the influence of different value functions and number of expected scenarios on the robustness of the solution.

When examining the complexity of the Directed Robust Two-Dose Problem, we were not able to determine the complexity of the special case with a fixed number of scenarios $|\mathcal{B}| \in \mathcal{O}(1)$ and a time gap between two vaccine doses linear in the maximum time frame $\varepsilon_{\max} \in \mathcal{O}(n)$. For this case, we could only show at least weak NP-hardness, but not derive a pseudo-polynomial algorithm. Solving this would show whether the hardness of the problem is solely caused by the order of ε or also depends on the number of scenarios. Additionally, the Directed Robust Two-Dose Problem already implies an order in choices linked to time steps. It might be interesting to examine an online version of the Two-Dose Problem or transfer some of the results from the rich literature on Online Matching Problems.

References

[1] Richard P Anstee. A polynomial algorithm for b-matchings: an alternative approach. *Information Processing Letters*, 24(3):153–157, 1987.

[2] Aharon Ben-Tal, Alexander Goryashko, Elana Guslitzer, and Arkadi Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical programming*, 99(2):351–376, 2004.

[3] Giuseppe Carlo Calafiore, Francesco Parino, Lorenzo Zino, and Alessandro Rizzo. Dynamic planning of a two-dose vaccination campaign with uncertain supplies. *European Journal of Operational Research*, 304(3):1269–1278, 2023.

[4] Lotty Evertje Duijzer, Willem Van Jaarsveld, and Rommert Dekker. Literature review: The vaccine supply chain. *European Journal of Operational Research*, 268(1):174–192, 2018.

[5] Bundesministerium fr Gesundheit (BMG). Impfdashboard data, March 2021.

[6] Michael R Garey, Ronald L Graham, David S Johnson, and Donald Ervin Knuth. Complexity results for bandwidth minimization. *SIAM Journal on Applied Mathematics*, 34(3):477–495, 1978.

[7] Graham Jurgens and Kyle Lackner. Modelled optimization of sars-cov-2 vaccine distribution: an evaluation of second dose deferral spacing of 6, 12, and 24 weeks. *medRxiv*, 2021.

[8] Arie MCA Koster, Manuel Kutschka, and Christian Raack. Robust network design: Formulations, valid inequalities, and computations. *Networks*, 61(2):128–149, 2013.

[9] Ho-Yin Mak, Tinglong Dai, and Christopher S Tang. Managing two-dose covid-19 vaccine rollouts with limited supply: Operations strategies for distributing time-sensitive resources. *Production and operations management*, 2022.

[10] Michel Minoux. Robust lp with right-handside uncertainty, duality and applications. *Encyclopedia of Optimization,*, pages 3317–3327, 2007.

[11] Seyed M Moghadas, Thomas N Vilches, Kevin Zhang, Shokoofeh Nourbakhsh, Pratha Sah, Meagan C Fitzpatrick, and Alison P Galvani. Evaluation of covid-19 vaccination strategies with a delayed second dose. *PLoS biology*, 19(4):e3001211, 2021.

[12] Yuto Omae, Makoto Sasaki, Jun Toyotani, Kazuyuki Hara, and Hirotaka Takahashi. Theoretical analysis of the sirvvd model for insights into the target rate of covid-19/sars-cov-2 vaccination in japan. *IEEE Access*, 10:43044–43054, 2022.

[13] Francesco Parino, Lorenzo Zino, Giuseppe C Calafiore, and Alessandro Rizzo. A model predictive control approach to optimally devise a two-dose vaccination roll-out: A case study on covid-19 in italy. *International Journal of Robust and Nonlinear Control*, 2021.

[14] Fernando P Polack, Stephen J Thomas, Nicholas Kitchin, Judith Absalon, Alejandra Gurtman, Stephen Lockhart, John L Perez, Gonzalo Pérez Marc, Edson D Moreira, Cristiano Zerbini, et al. Safety and efficacy of the bnt162b2 mrna covid-19 vaccine. *New England Journal of Medicine*, 383(27):2603–2615, 2020. PMID: 33301246.

[15] W. R. Pulleyblank. Faces of matching polyhedra, 1973.

[16] Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

[17] Jenny Segschneider and Arie M. C. A. Koster. Optimal vaccination strategies for multiple dose vaccinations. In *International Symposium on Combinatorial Optimization*.

[18] Robert A Shumsky, James Smith, Anne Hoen, and Michael Gilbert. Allocating covid-19 vaccines: Save one for the second dose? *Tuck School of Business Working Paper*, (3816200), 2021.

[19] Paulo JS Silva, Claudia Sagastizábal, Luís Gustavo Nonato, Claudio José Struchiner, and Tiago Pereira. Optimized delay of the second covid-19 vaccine dose reduces icu admissions. *Proceedings of the National Academy of Sciences*, 118(35), 2021.

[20] Ashleigh R Tuite, Lin Zhu, David N Fisman, and Joshua A Salomon. Alternative dose allocation strategies to increase benefits from constrained covid-19 vaccine supply. *Annals of internal medicine*, 174(4):570–572, 2021.

[21] Olivier J Wouters, Kenneth C Shadlen, Maximilian Salcher-Konrad, Andrew J Pollard, Heidi J Larson, Yot Teerawattananon, and Mark Jit. Challenges in ensuring global access to covid-19 vaccines: production, affordability, allocation, and deployment. *The Lancet*, 397(10278):1023–1034, 2021.