On the complexity of solving a decision problem with flow-depending costs: the case of the IJsselmeer dikes

Aida Abiad∗ Sander Gribling† Domenico Lahaye‡ Matthias Mnich§
Guus Regts¶ Lluis Vena∥ Gerard Verweij∗∗ Peter Zwaneveld††

Abstract

We consider a fundamental integer programming (IP) model for cost-benefit analysis flood protection through dike building in the Netherlands, due to Verweij and Zwaneveld. Experimental analysis with data for the IJsselmeer lead to integral optimal solution of the linear programming relaxation of the IP model. This naturally led to the question of integrality of the polytope associated with the IP model.

In this paper we first give a negative answer to this question by establishing non-integrality of the polytope. Second, we establish natural conditions that guarantee the linear programming relaxation of the IP model to be integral. We then test the most recent data on flood probabilities, damage and investment costs of the IJsselmeer for these conditions. Third, we show that the IP model can be solved in polynomial time when the number of dike segments, or the number of feasible barrier heights, are constant.

Keywords. Cost-benefit analysis; dynamic programming; integer programming

∗Corresponding author. Department of Quantitative Economics, Maastricht University, Maastricht, The Netherlands. A.AbiadMonge@maastrichtuniversity.nl.
†CWI, Amsterdam, The Netherlands. gribling@cwi.nl.
‡Delft Institute for Applied Mathematics, The Netherlands. d.j.p.lahaye@tudelft.nl.
§Department of Quantitative Economics, Maastricht University, Maastricht, The Netherlands. m.mnich@maastrichtuniversity.nl.
¶Korteweg de Vries Institute for Mathematics, University of Amsterdam, The Netherlands. guusregts@gmail.com.
∥Korteweg de Vries Institute for Mathematics, University of Amsterdam, The Netherlands. lluis.vena@gmail.com.
∗∗CPB Netherlands Bureau for Economic Policy Analysis, The Hague, The Netherlands. G.Verweij@cpb.nl.
††CPB Netherlands Bureau for Economic Policy Analysis, The Hague, The Netherlands. P.J.Zwaneveld@cpb.nl.
1 Introduction

Protection against increasing sea levels is an important issue around the world, including the Netherlands. Optimal dike heights are of crucial importance to the Netherlands as almost 60% of its surface is under threat of flooding from sea, lakes, or rivers. This area is protected by more than 3500 kilometers of dunes and dikes, which require substantial yearly investments of more than one billion Euro [8].

Recently, Zwaneveld and Verweij [8] presented an integer programming (IP) model for a cost-benefit analysis to determine optimal dike heights that allows highly flexible input parameters for flood probabilities, damage costs and investment costs for dike heightening. Their model improves upon an earlier model by Brekelmans et al. [2], who presented a dedicated approach without optimality guarantee, and which was in turn an improvement of the original model by van Dantzig [3] from 1956. The latter was introduced after a devastating flood in the Netherlands in 1953, with the goal of designing a long-lasting cost-efficient layout for a dike ring.

Our work is based on the integer programming model of Bos and Zwaneveld [1], Zwaneveld and Verweij [6] and a recent manuscript by Zwaneveld and Verweij [7], where the authors study the problem of economically optimal flood prevention in a situation in which multiple barrier dams and dikes protect the hinterland to both sea level rise as well as peak river discharges. Current optimal flood prevention methods (Kind [5], Brekelmans et al. [2], Zwaneveld and Verweij [8]) only consider single dike ring areas with no interdependency between dikes. Zwaneveld and Verweij [6, 7] present a graph-based model for a cost-benefit analysis to determine optimal dike heights with multiple interdependencies between dikes and barrier dams. Zwaneveld and Verweij [6] identify several solution approaches (e.g. a dynamic programming heuristic and branch-and-cut), and they also show that it can be solved quickly to proven optimality using a branch-and-cut approach for real world problem instances.

The natural question arising from the work of Zwaneveld and Verweij [7] is whether the linear programming relaxation of their IP model always admits an integral optimum.

1.1 Our contribution

Our first contribution is a negative answer to the question above. In particular, we show that the polytope associated to the IP model of Zwaneveld and Verweij [6, 7] is not necessarily integral.

Second, we derive sufficient conditions that ensure the LP relaxation to be integral. We
then experimentally verify whether these conditions are met by the most recent data on flood probabilities, damage and investment costs, which are presently used by the Dutch government. Finally, we show that the optimal dike heightening problem can be solved in polynomial time if either the number of barrier heights or the number of dike segments is constant.

This paper is organized as follows. In Section 2 we recap the IP model of Zwaneveld and Verweij [7] that forms the subject of our investigations. In Section 3 we discuss integrality of the polytope. In Section 4 we propose an alternative approach to solve the problem by means of dynamic programming. Finally, in Section 5 we present a natural abstract version of the dike height problem, which allows for several variations and open problems.

2 Integer programming model

In this section we present the IP model formulated by Zwaneveld and Verweij [7]. Before going into the details of the IP model, let us introduce some important terminology and the geographical configuration of the dikes in the Netherlands. A dike segment is a part of a dike that is protecting a region. It is possible that several segments protect the same area and in that case they are called a dike ring. In the Netherlands, dike ring areas and smaller dikes lie beneath the Afsluitdijk (or barrier dam) which is the outermost dike located in the north. The Afsluitdijk separates the North Sea and the IJsselmeer, an artificial lake; see Fig. 1 for an illustration.

![Map of the Netherlands with dike segments and the Afsluitdijk](image)

Figure 1: The relative locations of the North Sea, the IJsselmeer, and the Afsluitdijk and the dike ring enclosing it.
The IP model uses the following data:

- $D$ is the set of dike segments.
- $H_D$ is the set of possible heights for a dike segment. For ease of notation, we do not let $H_D$ depend on the dike segment, i.e., all dike segments have the same set of possible heights. We denote the height of a previous year by $h_1$, and that of the current year by $h_2$. Likewise, $H_B$ is the set of possible heights for the barrier dam and we denote the height of the barrier in the previous year by $h_B^1$, and that of the current year by $h_B^2$.
- $T$ is the set of time periods at which changes to a dike segment can be made (e.g., one can assume that changes are scheduled per year), for simplicity we assume (with abuse of notation) $T = \{0, 1, \ldots, T\}$.

The decision variables are:

- $CY(t, d, h_1, h_2)$: this binary variable takes value 1 if dike ring $d$ is updated in time period $t$ from height $h_1$ up to height $h_2$. If $h_1 = h_2$ then this dike ring segment is not strengthened in period $t$ and remains at its previous height. This decision variable is used for tracking investment (and maintenance) costs.
- $DY(t, d, h_2, h_B^R)$: this binary variable takes value 1 if at the end of period $t$ the barrier dam has height $h_B^2$, and dike segment $d$ is of height $h_2$. This variable is used to connect investments in dike segments (and the barrier dam) to expected damages. Another way to view it is that this variable linearizes the 0-1 variable $(\sum_{h_1} CY(t, d, h_1, h_2)) \cdot (\sum_{h_B^R} B(t, h_1^B, h_B^2))$.
- $B(t, h_1^B, h_B^R)$: this binary variable take value 1 if the barrier dam is updated in time period $t$ from height $h_1^B$ up to $h_B^2$. If $h_1^B = h_B^2$ then the barrier dam is not strengthened in period $t$ and remains at its previous height. This decision variable is used for bookkeeping investment (and maintenance) costs, flood probabilities and related expected damage costs of the barrier dam.

The input parameters are:

- $D_{\text{cost}}(t, d, h_1, h_2)$, the cost for investment and maintenance, if dike ring $d$ is strengthened in time period $t$ from $h_1$ to $h_2$. If $h_1 = h_2$, the dike ring segment is not strengthened and these costs only represent maintenance costs.
• $D_{\text{expdam}}(t, d, h_2, h_{B2}^B)$, the expected damage, i.e.,

$$D_{\text{expdam}}(t, d, h_2, h_{B2}^B) = \text{prob}(t, d, h_2, h_{B2}^B) \times \text{damage}(t, d, h_2, h_{B2}^B),$$

where $\text{prob}(t, d, h_2, h_{B2}^B)$ and $\text{damage}(t, d, h_2, h_{B2}^B)$ are respectively the probability of failure and the expected damage cost (the latter given that there is a flooding) in period $t$ given the height of the segment $h_2$ and the height of the barrier $h_{B2}^B$. Note that it is assumed that both the probability of failure and the expected damage upon failure of dike segment $d$ only depend on the height of segment $d$ and that of the barrier dam.

• $B_{\text{cost}}(t, d, h_1^B, h_{B2}^B)$, the cost for investment and maintenance, if the barrier dam is strengthened in time period $t$ from $h_1^B$ to $h_{B2}^B$. If $h_1^B = h_{B2}^B$, the barrier dam is not strengthened and these costs only represent maintenance costs.

• $B_{\text{expdam}}(t, h_{B2}^B)$, the expected damage of a flooding of the barrier dam, i.e. $\text{prob}(t, h_{B2}^B) \times \text{damage}(t, h_{B2}^B)$, here $\text{prob}(t, h_{B2}^B)$ and $\text{damage}(t, h_{B2}^B)$ are respectively the probability of failure and the expected damage cost (the latter given that there is a flooding), in period $t$ given the height of the barrier $h_{B2}^B$.

All input parameters are calculated in net present value of a certain year (i.e. 2020, which is the starting year for our calculations) and represent price levels in a certain year.

All in all, the IP model then reads as follows:

$$\min \sum_{t \in T} \sum_{d \in D} \sum_{h_1 \in H_D} \sum_{h_2 \geq h_1} D_{\text{cost}}(t, d, h_1, h_2) \cdot CY(t, d, h_1, h_2)$$

$$+ \sum_{t \in T} \sum_{d \in D} \sum_{h_2 \in H_D} \sum_{h_2^B} D_{\text{expdam}}(t, d, h_2, h_2^B) \cdot DY(t, d, h_2, h_2^B)$$

$$+ \sum_{t \in T} \sum_{h_1^B \in H_B} \sum_{h_2^B \geq h_1^B} \left(B_{\text{cost}}(t, h_1^B, h_2^B) + B_{\text{expdam}}(t, h_2^B)\right) \cdot B(t, h_1^B, h_2^B)$$
subject to

\[ CY(0, d, 0, 0) = 1, CY(0, d, h_1, h_2) = 0 \quad \forall d \in D, h_1, h_2 \in H_D, h_2 \geq h_1 \land h_2 > 0 \quad (4) \]

\[ \sum_{h_1 \leq h_2} CY(t - 1, d, h_1, h_2) = \sum_{h_3 \geq h_2} CY(t, d, h_2, h_3) \quad \forall t \in T_{>0}, d \in D, h_2 \in H_D \quad (5) \]

\[ \sum_{h_1 \leq h_2} CY(t, d, h_1, h_2) = \sum_{h_2^B} DY(t, d, h_2, h_2^B) \quad \forall t \in T, d \in D, h_2 \in H_D \quad (6) \]

\[ B(0, 0, 0) = 1, B(0, h_1^B, h_2^B) = 0 \quad \forall h_1^B, h_2^B \in H_B, h_2^B \geq h_1^B \land h_2^B > 0 \quad (7) \]

\[ \sum_{h_1^B \leq h_2^B} B(t - 1, h_1^B, h_2^B) = \sum_{h_2^B \geq h_2^B} B(t, h_2^B, h_3^B) \quad \forall t \in T \setminus \{0\}, d \in D, h_2^B \in H_B \quad (8) \]

\[ \sum_{h_1^B \leq h_2^B} B(t, h_1^B, h_2^B) = \sum_{h_2^B} DY(t, d, h_2, h_2^B) \quad \forall t \in T, d \in D, h_2^B \in H_B \quad (9) \]

\[ CY(t, d, h_1, h_2) \in \{0, 1\} \quad \forall t \in T, d \in D, h_1 \in H_D, h_2 \geq h_1 \in H_D \quad (10) \]

\[ DY(t, d, h_2, h_2^B) \in \{0, 1\} \quad \forall t \in T, d \in D, h_2 \in H_D, h_2^B \in H_B \quad (11) \]

\[ B(t, h_1^B, h_2^B) \in \{0, 1\} \quad \forall t \in T, d \in D, h_2^B \geq h_1^B \in H_B \quad (12) \]

Equations (6) and (9) are the linking constraints between the barrier and the dike segments using the variables \(DY\). Equations (5) and (8) are flow conditions. Equations (4) and (7) are the initial conditions. Equations (10), (11) and (12) are integrality constraints.

### 3 On the integrality of the polytope

The linear programming relaxation of the IP model from the previous section allows the decision variables to take values from the interval \([0, 1]\) instead of the integral \(\{0, 1\}\). We now give an example showing that the polytope defined by this relaxation can have vertices with non-integral coordinates.

The example involves the following sets indexing the variables:

- \(T = \{0, 1, 2\}\)
- one segment. Hence, we remove the dike index from all related variables.
- \(H = \{0, 1\}, H_B = \{0, 1\}\)
The point $P$, candidate to be a vertex of the polytope of the linear relaxation, has the following non-zero values:

| $(t, h_1, h_2)$ | $(0, 0, 0)$ | $(1, 0, 1)$ | $(1, 0, 0)$ | $(2, 1, 1)$ | $(2, 0, 0)$ |
|-----------------|------------|------------|------------|------------|------------|
| $CY(t, h_1, h_2)$ | 1          | 1/2        | 1/2        | 1/2        | 1/2        |
| $B(t, h_1, h_2)$ | 1          | 1/2        | 1/2        | 1/2        | 1/2        |
| $DY(t, h_2, h_2^b)$ | 1          | 1/2        | 1/2        | 1/2        | 1/2        |

The example is summarized in Fig. 3 where each arrow corresponds to one of the decision variables.

One can check that the example is a feasible solution (a point in the polytope). Indeed, the flow conditions are satisfied, as well as the equations linking the dummy variables $DY$ and the $CY$’s and $B$’s (Equations (6) and (9)).

To argue that the point $P$ is indeed a vertex of the polytope, we show that for every line with non-zero direction vector $v = (x_0, \ldots, x_{14})$ and for every $\varepsilon > 0$, either $P + \varepsilon v$ or $P - \varepsilon v$ is outside the polytope. Every coordinate $x_i$ of $v$ corresponds, uniquely, to a variable $B(\cdot)$, $CY(\cdot)$, or $DY(\cdot)$.

First observe that if $x_i$ is the coordinate related to a variable that is either 0 or 1 in $P$, then
\[ x_i = 0, \text{ as otherwise, for any } \varepsilon > 0, \text{ either } P + \varepsilon v \text{ or } P - \varepsilon v \text{ would be outside of the polytope.} \]

Hence, the only \( x_i \) that may be non-zero, are those for which the coordinate \( i \) in \( P \) is in the open interval \((0,1)\).

In our example, every equation involves at most 2 variables on each side of the equality, one of them being either 0 or 1. Hence the implications written below are forced by the previous observation. Assume, for instance, that the coefficient \( x_i \) corresponding to \( B(2,1,1) \) in \( v \) is negative.

- Then, by the flow constraints (Equation (8)), the coefficient of \( B(1,0,1) \) is negative.
- Then, by the flow constraints, the coefficient of \( B(1,0,0) \) is positive.
- Then, by the flow constraints, the coefficient of \( B(2,0,0) \) is positive.

Now, using the equations that link the variables \( B \) and \( DY \), we obtain that the the coefficient of \( DY(2,1,1) \) is positive, which implies that

- the coefficient of \( CY(2,1,1) \) in \( v \) is positive;
- then, by the flow constraints, the coefficient of \( CY(1,0,1) \) is positive;
- then, by the flow constraints, the coefficient of \( CY(1,0,0) \) is negative;
- then, by the flow constraints, the coefficient of \( CY(2,0,0) \) is negative.

Observe now that this implies that the coefficient of \( DY(2,0,0) \) has to be negative. However, let us now look at the coefficients of \( DY(1,0,1) \) and the one corresponding to \( DY(1,1,0) \).

If we use the links between the variables \( DY \) and \( B \), the coefficients corresponding to the variables \( DY(1,0,1) \) and \( DY(1,1,0) \) in \( v \) have to be negative and positive respectively. However, if we look at the equations linking the variables \( DY \) and \( CY \), the coefficients should have the opposite sign. Thus, these coefficients should be zero, implying that all the other coefficients have to be 0, which shows that no non-zero vector \( v \) exists.

The first coefficient involved in the argument was the one involving the variable \( B(2,1,1) \). Since the implications described here involve all the non-zero variables of the point, and the implications are reversible, the result now follows.

### 3.1 Sufficient conditions for integrality

In light of the above result, we now present some sufficient conditions on the objective function (1)–(3), that guarantee that either the linear relaxation of the integer program finds an
integral point as a solution, or that there is an integral point in the optimal face and a procedure
to find it.

**Proposition 1.** Consider the IP model from Sect. 2. Assume that,
(i) for every \( t, d, h_2, h'_2, h^B_2, h'_2^B \) such that \( h_2 \leq h'_2 \) and \( h^B_2 \leq h'_2^B \),

\[
D_{\text{expdam}}(t, d, h_2^B, h'_2) + D_{\text{expdam}}(t, d, h_2, h'_2^B) \geq D_{\text{expdam}}(t, d, h_2, h^B_2) + D_{\text{expdam}}(t, d, h'_2, h'_2^B)
\]

(13)

(ii) for every \( t, h^B_1, h'_1^B, h^B_2, h'_2^B \) such that \( h^B_1 \leq h'_1^B \) and \( h^B_2 \leq h'_2^B \),

\[
B_{\text{cost}}(t, h^B_1, h'_1^B) + B_{\text{cost}}(t, h'_1^B, h^B_2) \geq B_{\text{cost}}(t, h^B_1, h^B_2) + B_{\text{cost}}(t, h'_1^B, h'_2^B)
\]

(14)

(iii) for every \( t, d, h_1, h'_1, h_2, h'_2 \) such that \( h_1 \leq h'_1 \) and \( h_2 \leq h'_2 \),

\[
D_{\text{cost}}(t, d, h_1, h'_2) + D_{\text{cost}}(t, d, h'_1, h_2) \geq D_{\text{cost}}(t, d, h_1, h_2) + D_{\text{cost}}(t, d, h'_1, h'_2)
\]

(15)

Then, there is an optimal solution of the linear relaxation of the IP model in Sect. 2 with integer
coordinates.

Note that the term \( (B_{\text{cost}}(t, h^B_1, h'_1^B) + B_{\text{expdam}}(t, h^B_2)) \) from Equation (3) does not appear
in condition (ii) as it appears in both sides of the inequality.

**Proof of Proposition 1.** The problem from Sect. 2 can be thought of as several intertwined min-
cost flow problems (see Sect. 5), one for each dike, and one for the barrier. We say that a path
in a graph \( v_1e_1v_2e_2\ldots v_n \) with vertices \( v_1v_2\ldots v_n \) and edges \( e_1e_2\ldots e_{n-1} \), is a flow path when
the flow through each edge is the same. In our case, the vertices of the graph represent heights.

Let \( x_0 \) be a solution point given by the linear relaxation, and assume it is non-integral.
Using the monotone relations (14) and (15), the paths of the non-zero flows that \( x_0 \) defines
for each of the dikes and the barrier can be assumed to be completely ordered (as otherwise,
the flow values on the edges might be modified while maintaining the value of the in-flow and
out-flow at each vertex while not increasing the objective function). That is to say, we obtain
a layered flow: a flow path \( v_1e_1v_2e_2\ldots v_n \) with height profile \( v_1v_2\ldots v_n \) is above a flow path
\( w_1e'_1w_2e'_2\ldots w_n \) with height profile \( w_1w_2\ldots w_n \) when \( v_i \geq w_i \) for all \( i \) (i.e., no two flow-paths
strictly cross between two layers of vertices corresponding to two different consecutive times).
In particular, for each of the dikes \( d \), we can talk about a top path \( U_d \) (the height profile being
always larger or equal than all the other height profiles), and a bottom path \( L_d \), whose heights
are smaller or equal than all the other height profiles. There is also a top \( U_B \) and bottom \( L_B \) paths for the flow of the barrier.

Observe that, as \( x_0 \) is non-integral, at least one of the variables \( DY \) is non-integral (either not equal to zero or not equal to one).

Let \( DY_{\min} \) be the minimal distance of the non-integral variables to either 0 or 1. Using (13) as a guideline repeatedly, we modify the variables \( DY \) from \( x_0 \) to create a new feasible solution \( x_1 \) in which the variables \( DY(t, i, h_2, t^B) \) are “untangled”. That is: given \( t, i, h_2, h^B_2, h^B_1 \) such that

\[
\begin{align*}
& h_2 \leq h'_2, h^B_2 \leq h'^B_2 \\
& 1 - DY_{\min} \geq DY(t, i, h_2, h^B_2) \geq DY_{\min} \\
& 1 - DY_{\min} \geq DY(t, i, h'_2, h'^B_2) \geq DY_{\min} \\
& 1 - DY_{\min} \geq DY(t, i, h'_2, h'^B_2) \geq DY_{\min}
\end{align*}
\]

then, by modifying

\[
\begin{align*}
& \begin{cases}
  DY(t, i, h'_2, h'^B_2) \rightarrow DY(t, i, h'_2, h'^B_2) - DY_{\min} \\
  DY(t, i, h'_2, h'^B_2) \rightarrow DY(t, i, h'_2, h'^B_2) - DY_{\min} \\
  DY(t, i, h'_2, h'^B_2) \rightarrow DY(t, i, h'_2, h'^B_2) + DY_{\min} \\
  DY(t, i, h'_2, h'^B_2) \rightarrow DY(t, i, h'_2, h'^B_2) + DY_{\min}
\end{cases} \quad (16)
\end{align*}
\]

and keeping the other values of solution \( x_0 \), we obtain a new feasible solution \( x_1 \) as good as \( x_0 \). In particular, by repeated application of the argument leading to (16), we can assume that

\[
DY_{x_1}(t, i, h_2(U_i), h^B_2(U_B)) = \min \left\{ \sum_{h_2} DY_{x_0}(t, i, h_2, h^B_2(U_B)), \sum_{h_2'} DY_{x_0}(t, i, h_2(U_i), h^B_2) \right\}
\]

and that

\[
DY_{x_1}(t, i, h_2(L_i), h^B_2(L_B)) = \min \left\{ \sum_{h_2} DY_{x_0}(t, i, h_2, h^B_2(L_B)), \sum_{h_2'} DY_{x_0}(t, i, h_2(L_i), h^B_2) \right\},
\]

while the remaining variables of \( x_0 \) are kept equal in \( x_1 \). As the reassignment preserves the flow constraints, \( x_1 \) remains feasible. By (13), \( x_1 \) has the same objective value as \( x_0 \), since \( x_0 \) is optimal.

Let \( F_{\min} \) be the minimal difference to 0 or 1 of the flow through each \( L_d, U_d \) for every dike \( d \) and \( L_B \) or \( U_B \), which can be assumed to be the minimal value of

\[
\min_{t, i} \left\{ DY_{x_1}(t, i, h_2(U_i), h^B_2(U_B)), DY_{x_1}(t, i, h_2(L_i), h^B_2(L_B)) \right\}.
\]
As we shall see, $F_{\text{min}}$ is the minimal amount of flow which is reassigned between the upper and lower paths.

We note that $x_1$ is not a vertex of the polytope. Indeed, for any dike $d$, we can pair up $L_d \leftrightarrow L_B$ and $U_d \leftrightarrow U_B$. Using (14) and (15), this pairing is well defined and consistent. In particular, we can redirect an $\varepsilon$ amount of flow—where $0 < \varepsilon \leq F_{\text{min}}$—from each of the $L_d$ to $U_d$ and from $L_B$ to $U_B$, or vice versa (the redirection of the flow should be done on each of the paths simultaneously, either from upper to lower paths, or from lower to upper ones). Since there exists a $d$ (or $B$) for which the paths $L_d$ and $U_d$ differ, this flow-redirection by $\varepsilon$ gives a different point on the polytope of feasible points and shows that $x_1$ is not a vertex of the polytope.

Furthermore, for every $\varepsilon > 0$, the mentioned flow redirection should give the same value of the objective function (since otherwise $x_0$ would not have been an optimal solution). Hence we can choose to redirect the flow at our convenience; we redirect it so that the edge whose flow value is $F_{\text{min}}$ becomes either 0 or 1 (depending on whether its value is closer to 0 or to 1, if $F_{\text{min}} = 1/2$, we arbitrarily redirect the flow either way). In particular, we obtain a new solution $x_2$ where the number of edges with non-integral flow has been reduced by at least one. This procedure can be iterated until no non-integral flows are found. Therefore, an integral vertex of the polytope in the optimal face of the linear relaxation of the integer program is found.

**Corollary 1.** The conclusion of Proposition 1 also holds if we assume conditions (ii) and (iii), and condition (i) on the objective function is replaced by

(i') For each dike $d \in D$, either

$$D_{\text{expdam}}(t, d, h_2^B) + D_{\text{expdam}}(t, h_2^B, h_2^B) \leq D_{\text{expdam}}(t, d, h_2^B) + D_{\text{expdam}}(t, h_2^B, h_2^B)$$

(17)

for every $t, h_2^B, h_2^B, h_2^B$ such that $h_2 \leq h_2'$ and $h_2^B \leq h_2^B$, or

$$D_{\text{expdam}}(t, d, h_2^B, h_2^B) + D_{\text{expdam}}(t, d, h_2^B, h_2^B) \geq D_{\text{expdam}}(t, d, h_2^B) + D_{\text{expdam}}(t, d, h_2^B, h_2^B)$$

(18)

for every $t, h_2, h_2', h_2^B, h_2^B$ such that $h_2 \leq h_2'$ and $h_2^B \leq h_2^B$.

**Proof.** The argument of the proof of Proposition 1 should be modified as follows. Observe that the layering of the flow-paths can be maintained due to conditions (ii) and (iii). Then, the
flow path pairing that allows for the flow reassignment of the second part of the proof can be modified as follows. The dike $d$ uses the pairing

$$L_d \leftrightarrow L_B \text{ and } U_d \leftrightarrow U_B$$

if part (18) of condition (i) is satisfied, and it is exchanged by the new pairing

$$L_d \leftrightarrow U_B \text{ and } U_d \leftrightarrow L_B.$$ 

if part (17) is satisfied. In each of the cases we use either the modification on the $DY$ due to (16) or due to (16) where the $-DY_{min}$ and $+DY_{min}$ are exchanged.

If the solution is non-integral, there exists a flow reassignment between the paths paired with $U_B$, and the paths paired with $L_B$. Mutatis mutandis, the reassigning flow argument carries over to this new case.

To provide some intuition, observe that all inequalities appearing in conditions (i)–(iii) are of a similar form: for $a \leq a'$ and $b \leq b'$ we have that some function $c(\cdot, \cdot)$ satisfies $c(a, b) + c(a', b') \leq c(a, b') + c(a', b)$. Such an inequality for $c$ is naturally satisfied (in fact with equality) if $c$ is of the form $c(x, y) = f(y) - f(x) + c_0$ for some function $f$ and constant $c_0$. In the context of conditions (ii) and (iii) from Proposition 1, such a form is somewhat reasonable to expect: the cost of rising a dike from level $x$ to $y$ compares to the cost of rising the dike from level 0 to $y$, minus the effort already made to rise it from 0 to $x$, plus perhaps some inefficiency overhead $c_0$.

### 3.2 Computational results

Conditions (i’), (ii) and (iii) from Corollary 1 and Proposition 1 have been implemented and tested for the most recent data on flood probabilities, damage and investment costs, and the results confirm that they are often met.

In the first column of the following tables we specify the years that we used in our study: 5 year periods until 2100 and 10 year periods after 2100. In the first row we specify the specific dike rings. The description of the dike rings around Lake IJssel and the IJsseldelta is as follows (the numbers are also used in Fig. 3):

- zwf = Zuid-West Friesland = 6.4
- nfl = Noord-Oost Flevoland = 8.1
- wie = Wieringen = 12.1
- mas = Mastenbroek = 10.1
- sal = Salland = 53.1
- nop = Noord-Oost Polder = 7.1
- wfn = West-Friesland Noord = 13.2
- ijd = IJsseldelta = 11.1
- vol = Vollenhove = 9.1
- sal = Salland = 53.1
- ovl = Oost-Veluwe = 52.1
As safety levels we included 14 levels for the dike rings and 14 levels for the Afsluitdijk. As a result, for each dike ring \( \frac{1}{2} \times 14 \times 15 = 105 \) combinations of safety levels for both the dike rings and the Afsluitdijk could be evaluated. Hence, in total we tested \( 105 \times 105 = 11025 \) instances (numbers in the second column). The rest of numbers in the following tables correspond to the number of instances that fulfill the corresponding condition and are calculated for all dike rings (first numerical row) and year (first column) combinations. Note that the year 2020 is the initial year considered, thus the conditions of Corollary 1 and Proposition 1 are trivially satisfied for each dike ring (zwf, nop, nfl, wfn, wie, ijd, mas, vol, sal, ovl) and barrier (Afsluitdijk); the “2020” row is hence added for comparison purposes.

We ran the simulation with pumps on the Afsluitdijk and without additional strengthening measures of the Afsluitdijk (STA-program).
### Condition (i') from Corollary 1

| year | zwf  | nlp  | afl  | wfn  | wie  | jpl  | mas  | vol  | sal  | owf  |
|------|------|------|------|------|------|------|------|------|------|------|
| 2020 | 11025 | 11025 | 11025 | 11025 | 11025 | 11025 | 11025 | 11025 | 11025 | 11025 |
| 2021 | 10930 | 11025 | 9484 | 10935 | 11025 | 11035 | 11025 | 11025 | 11025 | 11025 |
| 2022 | 10815 | 11022 | 9494 | 10999 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2023 | 10750 | 11023 | 9510 | 10874 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2024 | 10936 | 11025 | 9530 | 10857 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2025 | 10970 | 11025 | 9535 | 10832 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2026 | 10992 | 11025 | 9588 | 10821 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2027 | 11000 | 11025 | 9618 | 10811 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2028 | 11010 | 11025 | 9666 | 10807 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2029 | 11017 | 11025 | 9674 | 10807 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2030 | 11021 | 11025 | 9677 | 10807 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2031 | 11024 | 11025 | 9698 | 10803 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2032 | 11025 | 11025 | 9715 | 10799 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2033 | 11025 | 11025 | 9762 | 10803 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2034 | 11025 | 11025 | 9777 | 10803 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2035 | 11025 | 11025 | 9812 | 10803 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2036 | 11025 | 11025 | 9847 | 10807 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2037 | 11025 | 11025 | 9895 | 10807 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2038 | 11025 | 11025 | 9927 | 10809 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2039 | 11025 | 11025 | 9993 | 10894 | 11025 | 11025 | 11025 | 11017 | 11025 | 11025 |
| 2040 | 11025 | 11025 | 10020 | 10954 | 10934 | 10906 | 10927 | 10498 | 10771 | 7937 |
| 2041 | 10975 | 11025 | 10098 | 10988 | 10752 | 10612 | 10304 | 10472 | 10776 | 7920 |
| 2042 | 10854 | 11025 | 10197 | 10987 | 10492 | 10683 | 10311 | 10470 | 10799 | 7876 |
| 2043 | 10636 | 11025 | 10236 | 10053 | 10206 | 10633 | 10316 | 10439 | 10825 | 7858 |
| 2044 | 10207 | 11025 | 10289 | 10092 | 9751 | 10654 | 10331 | 10388 | 10831 | 7842 |
| 2045 | 9608 | 11025 | 10365 | 10783 | 9205 | 10696 | 10369 | 10273 | 10847 | 7796 |
| 2046 | 9257 | 11025 | 10447 | 10436 | 9114 | 10747 | 10423 | 10120 | 10857 | 7755 |
| 2047 | 9153 | 11025 | 10456 | 9881 | 9114 | 10779 | 10492 | 9919 | 10874 | 7727 |
| 2048 | 8533 | 11025 | 10473 | 9299 | 8477 | 10715 | 10603 | 9745 | 10876 | 7607 |
| 2049 | 8359 | 11025 | 10520 | 9650 | 8477 | 10853 | 10681 | 9519 | 10874 | 7419 |
| 2050 | 7936 | 11025 | 10607 | 8802 | 8477 | 10868 | 10730 | 9261 | 10823 | 7384 |
| 2051 | 7517 | 11025 | 10640 | 8459 | 7840 | 10901 | 10773 | 7748 | 6573 | 7352 |
| 2052 | 7067 | 11025 | 10708 | 8251 | 7749 | 10921 | 10815 | 4884 | 5581 | 7339 |
| 2053 | 6579 | 11025 | 10745 | 7849 | 7293 | 10933 | 10846 | 3719 | 5222 | 7270 |
| 2054 | 5994 | 11025 | 10864 | 7501 | 6930 | 10954 | 10883 | 3403 | 5052 | 7276 |
| 2055 | 5475 | 11025 | 10935 | 7096 | 6830 | 10954 | 10923 | 3245 | 5021 | 7276 |
| 2056 | 4786 | 11025 | 10975 | 6653 | 6111 | 10973 | 10925 | 3212 | 5026 | 7253 |
| year | afsluitdijk |
|------|-------------|
| 2020 | 2380        |
| 2021 | 2016        |
| 2026 | 2016        |
| 2031 | 2016        |
| 2036 | 2016        |
| 2041 | 2016        |
| 2046 | 2016        |
| 2051 | 2016        |
| 2056 | 2016        |
| 2061 | 2016        |
| 2066 | 2016        |
| 2071 | 2016        |
| 2076 | 2016        |
| 2081 | 2016        |
| 2086 | 2016        |
| 2091 | 2016        |
| 2096 | 2016        |
| 2101 | 2016        |
| 2111 | 2016        |
| 2121 | 2016        |
| 2131 | 2016        |
| 2141 | 2016        |
| 2151 | 2016        |
| 2161 | 2016        |
| 2171 | 2016        |
| 2181 | 2016        |
| 2191 | 2016        |
| 2201 | 2016        |
| 2211 | 2016        |
| 2221 | 2016        |
| 2231 | 2016        |
| 2241 | 2016        |
| 2251 | 2016        |
| 2261 | 2016        |
| 2271 | 2016        |
| 2281 | 2016        |
| 2291 | 2016        |
| year | swf | np | nff | wfs | wie | jid | mas | vol | sal | oval |
|------|-----|----|-----|-----|-----|-----|-----|-----|-----|------|
| 2020 | 2380 | 2380 | 2380 | 2380 | 2380 | 2380 | 2380 | 2380 | 2380 | 2380 |
| 2021 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2026 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2031 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2036 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2041 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2046 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2051 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2056 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2061 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2066 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2071 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2076 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2081 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2086 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2091 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2096 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2101 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2111 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2121 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2131 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2141 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2151 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2161 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2171 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2181 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2191 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2201 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2211 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2221 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2231 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2241 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2251 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2261 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2271 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2281 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
| 2291 | 2162 | 2188 | 2089 | 2104 | 2018 | 2217 | 2180 | 2210 | 2085 |
4 Alternative approaches

A feasible solution to the IP presented in Sect. 2 can be interpreted as a choice of height $h^d(t)$ for each dike segment at each time period $t$, and a height $h^b(t)$ of the barrier dam. Abstractly, the cost of these height series can be written as a sum of cost terms which depend only on the ‘upgrade’ done in period $t$ to segment $d$ (i.e., a heightening of the dike, or merely the maintenance cost); we denote this by $\text{cost}^d(h^d(t-1), h^d(t), t)$ for segment $d$, and by $\text{cost}^b(h^b(t-1), h^b(t), t)$ for the barrier. Finally, there is also an expected damage cost for upgrading the dike and barrier to heights $h^d(t)$ and $h^b(t)$ in period $t$, denoted by $\text{dam}^{d,b}(h^b(t), h^d(t), t)$. The problem modeled in Sect. 2 can thus be written in the following way:

$$
\min \left\{ \sum_{t \in \mathcal{T}} \text{cost}^b(h^b(t-1), h^b(t), t) + \sum_{d \in D} \text{cost}^d(h^d(t-1), h^d(t), t) + \text{dam}^{d,b}(h^b(t), h^d(t), t) \right\}
$$

s.t.  

$$
\begin{align*}
& h^d(t) \in H^D, h^b(t) \in H^B \text{ for } d \in D, t \in \mathcal{T} \quad (20) \\
& h^d(t) \geq h^d(t-1) \text{ for } d \in D, t \in \mathcal{T} \quad (21) \\
& h^b(t) \geq h^b(t-1) \text{ for } t \in \mathcal{T} \quad (22)
\end{align*}
$$

The linear relaxation of the IP model presented in Sect. 2 can be solved in time polynomial in $|D|, |\mathcal{T}|, |H^D|$, and $|H^B|$. However, in general there is no guarantee that the returned solution is integral, see Sect. 3. In the next two sections we describe two different approaches to solving this problem. Both approaches have the benefit of solving the integer problem exactly. However, this comes at a cost: both approaches give a polynomial time algorithm only if one of the parameters is regarded as a constant. The first approach is to solve the integer program by ways of a dynamic program. The second approach comes down to enumerating all possible height profiles of the barrier dam, and for each profile solving shortest path problems on small graphs.

4.1 Dynamic programming

There are two key observations to be made. First, the second part of the objective function decomposes naturally into a sum of $|D|$ terms, each of which depends only on the barrier height and one segment. Secondly, for each time period the cost only depends on the dike/barrier heights at times $t-1$ and $t$. Together this allows us to solve the problem using a dynamic program. The recursion will be on the time period. We maintain a table which stores values $\text{opt}(h^b, h^d, t)$ for all $t \in \mathcal{T}, h^b \in H^B, h^d \in (H^D)^D$. Their interpretation is as that $\text{opt}(h^b, h^d, t)$ is equal to the minimum cost made, up to time $t$, when the barrier and segments are of height
$h^b$ and $h^d$ at time period $t$ respectively. We can compute the entries of this table by means of the following recursion:

$$\text{opt}(h^b, h^d, t) = \min \left\{ \text{opt}(h^b - i^b, h^d - i^d, t - 1) + \text{cost}^b(h^b - i^b, h^b, t) + \text{cost}(h^d - i^d, h^d, t) + \text{dam}(h^b, h^d, t) : h^b - i^b \in H_B, h^d - i^d \in (H_D)^{|D|} \right\}$$

It follows that each entry of the table can be computed in time $O(|H_B||H_D||D|)$. Hence, all entries of the table can be filled in time $O((|H_B||H_D||D|^2 \cdot |T|))$. Using the interpretation of $\text{opt}(h^b, h^d, t)$, it follows that the optimum of (19) is equal to

$$\min_{h^b \in H_B, h^d \in (H_D)^{|D|}} \text{opt}(h^b, h^d, T) .$$

This shows the following result:

**Theorem 1.** One can determine the optimal value of (19) in time $O((|H_B||H_D||D|^2 \cdot |T|))$.

### 4.2 Shortest paths

In the previous section we have seen an algorithm for computing the optimal dike/barrier height profiles which has polynomial runtime for a fixed number of dike segments, in this section we present a different algorithm, based on shortest paths, that runs in polynomial time when the number of possible barrier heights is fixed. We present an algorithm that computes the optimal value of (19) in time

$$O \left( \frac{\# \text{ segments}}{|D|} \cdot \frac{T \cdot |H_D|^2}{|T|^{|H_B|}} \right) .$$

To illustrate the basic idea we first discuss the algorithm for the setting of one dike segment and no barrier, we then add a barrier dam and from that the generalization to multiple dike segments and barriers easily follows.

#### 4.2.1 One dike segment, no barrier

First consider the situation with only one dike segment and no barrier. In this case the problem of minimizing the cost at time period $T$ becomes equivalent to finding a shortest $p$-$q$ path in the following graph. The source $p = (0, 0)$ is the initial height of the dike at time 0. Then, for each time $t \in \{1, \ldots, T\}$ and each possible height of the dike $h$, we define a node $(t, h)$. Finally we
define a sink node \( q \). The edges are defined as follows. We first add an edge between \((0,0)\) and \((1,h)\) for each \( h \in H_D \), with weight \( \text{cost}(0,h,1) \), similarly for each \( t \in \{1, \ldots, T\} \) and height pair \( h_1 \leq h_2 \) there is an edge from \((t-1,h_1)\) to \((t,h_2)\) with weight \( \text{cost}(h_1,h_2,t) \) equal to the financial cost associated to the decision of raising the dike segment from height \( h_1 \) to \( h_2 \) in time period \( t \). Notice that since there is no barrier, we can assume that the expected damage cost \( \text{dam}(t,h) \) are incorporated in \( \text{cost}(h_1,h_2,t) \). Finally, the nodes \((T,h)\) are all connected to the sink \( q \). In the figure below the incoming and outgoing arcs of a node \((t,h_2)\) are sketched for some \( 0 < t < T \) and \( h_2 \in H_D \). One observes that, indeed, the shortest \( p-q \) path corresponds to the best strategy of heightening this dike segment.

Recall, the shortest \( p-q \) path in a graph \( G = (V,E) \) with non-negative edge weights can be found in time \( O(|V|^2) \) using Dijkstra’s algorithm.

![Graph](image-url)

**4.2.2 One dike segment, a barrier**

We now consider the case of a single dike segment and a barrier. The observation we need to make is that the total financial cost incurred by upgrading the dike segment from height \( h_1 \) to height \( h_2 \) in time period \( t \) no longer only depend on the dike segment, they also depend on the height of the barrier at time point \( t \). This means that we cannot solve a shortest path problem for the barrier and dike segment separately: the costs on the dike segment graph depend on the path chosen in the barrier graph.

The key idea is that if we fix the height of the barrier at each time \( t \), then we reduce to the previous setting where all the costs are known. Hence, the optimal value of (19) can be found by minimizing over the possible height profiles \( h^b(t) \) of the barrier over time, the minimum cost of a \( p-q \) path in the network defined in the previous section (using the costs associated to \( h^b(t) \)) plus the cost of implementing height profile \( h^b(t) \). The outer minimization over the possible
height profiles \( h^b(t) \) is performed by enumeration, which takes time \( O(T|H_B|) \). This means that the optimal investment strategy for both the dike segment and barrier can be found in time

\[
O \left( (T \cdot |H_D|)^2 \cdot \left( \frac{T}{|H_B|} \right) \right) = O \left( (T \cdot |H_D|)^2 \cdot T^{|H_B|} \right).
\]

### 4.2.3 Multiple dike segments and a barrier

The approach of the previous section easily generalizes to the setting of multiple dike segments and a barrier. Once a height profile \( h^b(t) \) of the barrier dike is fixed, the optimal height profiles of each of the different dike segments can be computed independently. Hence the problem of finding the optimal investment strategy for multiple dike segments and a barrier can be solved in time

\[
O \left( |D| \cdot (T \cdot |H_D|)^2 \cdot T^{|H_B|} \right).
\]

This approach generalizes to the setting of multiple barriers and dike segments (where the costs of a dike segment at time \( t \) may depend on the height of several barriers). The complexity will be of the form

\[
O \left( |D| \cdot (T \cdot |H_D|)^2 \cdot T^{|H_B|} \cdot |B| \right),
\]

where \( |B| \) is the number of barriers. One should note that the above approach assumes the same discretization in time of the barrier and dike segments. It seems reasonable to assume a coarser discretization for the barrier of say \( T_B \) steps, this would reduce the above-mentioned formula to

\[
O \left( |D| \cdot (T \cdot |H_D|)^2 \cdot (T_B)^{|H_B|} \cdot |B| \right).
\]

### 5 An abstraction of the dike heightening problem

In this section we present a natural abstract version of the dike heightening problem, which allows for several variations and questions, which we believe have not been considered in the literature before. We believe that studying these variations may shed more light on the complexity of the dike height problem.

In the dike height problem we essentially have two directed graphs where each path in one of the two graphs (the one modeling the height of the barrier dam) influences the cost of arcs in the other graph. It is not difficult to show that if we were to allow any kind of influence of the path in the one graph on the cost of arcs in the other graph, the problem would become
NP-hard. Indeed, one can easily show that in this case the problem contains the problem of finding two vertex disjoint paths in a directed graph, which is NP-complete [4].

For this reason, we consider the following restricted problem.

**Definition 1.** For \( k \in \mathbb{N} \), a \textit{k-layered} graph is a directed graph \( D = (V, A) \) such that \( V \) is partitioned into \textit{layers} \( V = V_0 \cup V_1 \cup \ldots \cup V_k \cup V_{k+1} \) such that each \( a \in A \) is from \( V_i \) to \( V_{i+1} \) for some \( i = 0, \ldots, k \), where \( V_0 \) and \( V_{k+1} \) both consist of a single vertex and where \( |V_i| = |V_{i+1}| = \cdots = |V_k| \). We denote the arcs between \( V_i \) and \( V_{i+1} \) by \( A|V_i, V_{i+1}| \) and we refer to \( |V_i| \) as the \textit{partition size}.

With this in mind, we define the **Minimum Intertwined Cost Path** problem as follows. The problem takes as input \( d + 1 \) k-layered graphs \( G^1 = (V^1, A^1), G^2 = (V^2, A^2), \ldots, G^{d+1} = (V^{d+1}, A^{d+1}) \) with partitions \( V^j = V^{(j)}_1 \cup \ldots \cup V^{(j)}_{k+1} \) and cost functions \( c^j : A^j \to \mathbb{R}_{\geq 0} \) for \( j = 1, \ldots, d+1 \), and for each \( i = 1, \ldots, k \) and \( t = 2, \ldots, d+1 \) maps \( m^t_i : V^i \times A^1[V^i_{t-1}, V^i_t] \to \mathbb{R}_{\geq 0} \).

Given \( d + 1 \) paths \( P^1, P^2, \ldots, P^{d+1} \) with \( P^j = (a^j_1, v^j_1, a^j_2, v^j_2, \ldots, a^j_k, v^j_k, a^j_{k+1}) \) from \( V^{(j)}_0 \) to \( V^{(j)}_{k+1} \) with \( a^j_i = (v^j_i, v^j_{i+1}) \) for \( j = 1, \ldots, d + 1 \), we define the cost of the \((d+1)\)-tuple \((P^1; P^2, \ldots, P^{d+1})\) as

\[
\text{cost}(P^1; P^2, \ldots, P^{d+1}) = \sum_{i=1}^{k+1} \sum_{t=1}^{d+1} c^i(a^i_t) + \sum_{i=1}^{k+1} \sum_{t=2}^{d+1} m^t_i(v^i_t, a^i_t) .
\]

The objective is to compute the \((d+1)\)-tuple of paths \((P^{1*}; P^{2*}, \ldots, P^{d+1*})\) with minimum cost over all such \((d+1)\)-tuples.

In the **Minimum Intertwined Cost Path** problem, the dependence of \( \text{cost}(P^1; P^2, \ldots, P^{d+1}) \) on the paths \( P^2, \ldots, P^{d+1} \) is linear in the edges of \( P^2, \ldots, P^{d+1} \). Note that the IP problem from Sect. 2 is a specific case of the **Minimum Intertwined Cost Path** problem where the barrier acts as \( P^1 \), each of the dikes is represented one path \( P^j \), \( j = 2, \ldots, d + 1 \), and the cost functions \( m^t_i \) only depends on the vertices \( m^t_i(v^i_t, a^i_t) = m^t_i(v^i_t, v^i_{t+1}) \), in addition to the edges between \( V_i \) and \( V_{i+1} \) being restricted (only connecting vertices of non-decreasing heights).

This particular fact allowed us in Sect. 4.2.2 to give an algorithm for the problem, which runs in polynomial time if we consider the size of the sets in the partition of the vertices of the second graph as a constant. Clearly if the bipartite graphs between \( V^{(2)}_i \) and \( V^{(2)}_{i+1} \) are complete, then this dynamic programming approach will not work. It would be interesting to find out if some other approach may yield an efficient algorithm.

We end this section with some concrete questions.
**Question 1.** Is the Minimum Intertwined Cost Path problem NP-hard for unbounded number of possible heights?

We do not have an answer for Question 1, but we remark the following: with an appropriate cost function on the updating of the heights of one dike, instances of the Knapsack problem can be seen as optimizing the height of one dike. Indeed, the decision of updating the height of a dike at time $t \in \mathbb{N}$ corresponds to the decision of adding an certain number of copies of an item to the knapsack; the total height of the dike at time $t$ corresponds to the accumulated weight of the chosen items (counting multiplicities) to be carried among the first $t$ items. The cost function of upgrading the height at time $t$ by $kw_t$ units corresponds to the profit of adding $k$ copies of the item $t$, whose weight is $w_t$. The cost function of the upgrading the dikes is such that once the capacity of the knapsack is exceeded by a set of items, then the cost of keeping or upgrading the dike height is unreasonable high. With this correspondence, we observe that the optimal solution of the Knapsack problem corresponds to the optimal solution of the dike height. Computing an optimal solution to the Knapsack problem is well-known to be NP-hard. One of the inputs of the Knapsack problem is the logarithm of the total weight of the knapsack bag. Thus the dynamic program proposed in Sec. 4.1 is an exponential time algorithm.

If Question 1 has a positive answer, then it makes sense to consider the following questions.

**Question 2.** Under which conditions on the bipartite graphs $G^j[V_i, V_{i+1}]$, ($j = 1, 2$, $i = 1, \ldots, k$) is there a polynomial time algorithm for the Minimum Intertwined Cost Path problem?

**Question 3.** Suppose the partition size of $G_2$ is constant. Under which conditions on the bipartite graphs $G_j[V_i, V_{i+1}]$ ($j = 1, 2$, $i = 0, \ldots, k$) is there a polynomial time algorithm for the Minimum Intertwined Cost Path problem?

**Acknowledgements.** We thank Kees Roos for helpful discussions and for presenting the ideas of Brekelmans et al. [2]. We moreover thank André Woning from Rijkswaterstaat for useful background information, and Stan van Hoesel and Peter van de Ven for their careful reading of the manuscript. A large part of the results in this paper were obtained in the context of the 2017 Study Group Mathematics with Industry week, for which we want to thank the organizers.
References

[1] F. Bos and P.J. Zwaneveld, Een snelle kosten-effectiviteitsanalyse voor het Deltaprogramma IJsselmeergebied, *CPB Communication* (including a CPB Background Document), 27 September 2012, The Hague, The Netherlands.

[2] R. Brekelmans, D. den Hertog, K. Roos and C. Eijgenraam, Safe dike heights at minimal costs: the nonhomogeneous case, *Operations Research* **60**(6) (2012), 1342–1355.

[3] D. van Dantzig, Economic decision problems for flood prevention, *Econometrica* **24**(3) (1956), 276–287.

[4] S. Fortune, J. Hopcroft and J. Wyllie, The directed subgraph homeomorphism problem, *Theoretical Computer Science* **10** (1980), 111–121.

[5] J. Kind, Maatschappelijke kosten-batenanalyse waterveiligheid 21e eeuw (MKBA WV21), Deltares Report, Delft, The Netherlands, 2011.

[6] P.J. Zwaneveld and G. Verweij, Economisch optimale waterveiligheid in het IJsselmeergebied. MKBA Waterveiligheid: Afsluitdijk, Houtribdijk, IJsselmeer, IJssel- en Vechtdelta en markermeer, *CPB Communication* (2014), The Hague, The Netherlands.

[7] P.J. Zwaneveld and G. Verweij, Economic decision problems in multi-level flood prevention: a new graph-based approach used for real world applications, *Concept CPB Discussion paper* (2018), The Hague, the Netherlands.

[8] P.J. Zwaneveld, G. Verweij and S. van Hoesel, Safe dike heights at minimal costs: an integer programming approach, *European Journal of Operational Research*, to appear. For a previous version, see *CPB Discussion Paper* **277** (2014).