Answers by the authors on the referee report of reviewer #2

Date: 08/05/2017

Because the supplement of reviewer #2 contains multiple pages, this answer will be fairly long. Furthermore, we have been in contact with reviewer #2 in order to constructively solve the major points of criticism of reviewer #2. In the following, we attempted to summarize these points and the related discussion between us (the authors) and reviewer #2 from our point of view:

- The major point of criticism is (in the words of reviewer #2) “copying” and “lack of references.” While we disagree with the gravity of these qualifications, we do acknowledge that in hindsight, some references could have been better and clearer. We took this criticism seriously and worked with the reviewer to (primarily) improve the introduction of the paper. However, the main content is, in our opinion, still original work.
- Furthermore, we think that other points of criticism which relate to earlier published work and the novelty of ours can be solved almost completely by more precise and clearer formulation of the aim and goals of the paper.

The following pages are based on the supplement of the review of reviewer #2. We took the original text of the supplement and divided it up into smaller pieces. This original text by reviewer #2 is presented in blue, italic text, while our answers are numbered and in black. The headings (“General comment” and “Specific comments”) are retained from the original review text.

General comment:

Before reading this referee comment, the reader must be aware of the fact that the authors of this paper actively asked me to referee their paper. I thank them for this opportunity and making me aware of this and previous papers. I have had a meeting with two authors of this paper to discuss my first impressions. This referee comment benefitted from the insight the authors provided me in this meeting.

In this general comment I will state that this paper copies earlier work. No proper references are made to this work. Hence, this paper doesn't meet minimal scientific standards. I will provide references to plenty published reports and articles to support my claim. In the Netherlands, many involved experts, including many full professors at various Dutch universities (which wrote and published referee reports on this earlier work or supported the development of earlier work) can be asked to confirm my claim.

The main claim by the authors that a shortest path/dynamic programming approach (previously presented and discusses by other authors) to solve economic optimal dike heightening is 'advantageous' needs to be elaborated a lot more. Many previously stated and published arguments against dynamic programming are not mentioned. Moreover, the scientific ambition of this paper is not clear. Furthermore, no calculations are presented by Dupuits et al. (2017) to support their claim. I will provide arguments for this claim.

We thank the reviewer for taking time to sit down with us.

We do not comment on these first three paragraphs, as it seems that all the mentioned issues return in a more detailed form in the remainder of the review document.

This paper copies the approach by Zwaneveld & Verweij (2014a) for finding an optimal configuration for interdependent lines of flood defences. In Zwaneveld (2012; section 1.1; in 2014 provided to the authors by email) several approaches are discusses to solve this model. This modelling approach by Zwaneveld & Verweij (2014a, 2014b) including graph-based (shortest path or minimum cost flow problems) solution approaches and the preferred ILP approach was earlier copied and described by Yuceoglu (2015, Chapter 5: Safe Dike heights in the Netherlands). This PhD-thesis builds on the work by Zwaneveld and Verweij (2014a , 2014b) and discusses graph –based algorithms to solve the so-called Diqe-Opt model (see later for a discussion of this model). Zwaneveld & Verweij (2014a) identify several algorithms to solve the problem both to proven optimality and to solve the problem heuristically (with the advantage that computing times remain limited). Zwaneveld & verweij92014a0 aplly their model to ral world problem instances to support crucial Cabinet decisions for the Netherlands.

1) We share a similar approach as Zwaneveld & Verweij (2014a), by using graphs to model the problem. However, contrary to the approach of Zwaneveld & Verweij (2014a), we use a greedy algorithm to solve the shortest path problem instead of modelling it as an I(L)P model. The reason for using a greedy algorithm is, among other reasons, an attempt to reduce the number of risk calculations (see also answer 3 for a more detailed answer). Therefore, we disagree with the use of the word 'copy'. Nevertheless, we did miss that in the appendix of Zwaneveld & Verweij (2014a) the problem was already identified as a graph problem. We will therefore correct that specific aspect in the paper.

I apply and explain an algorithm to solve the problem to proven optimality. Their algorithms requires hardyly any programmng efforts and little solution time ('less than one minute or so'). The ideas to solve the problem heuristically were not implemented in practice due to the fact that the algorithm to solve the problem to proven optimality was superior according to Zwaneveld and Verweij (2014a).

2) Based on our own experience, finding the shortest path in a graph with a greedy algorithm can be explained intuitively for most engineers working in the field of

flood risk. This explanation is one of the motivations of writing Section 2. It is the opinion of the authors that an IP model (specifically the model as proposed by Zwaneveld and Verweij (2014a)) requires at least a basic knowledge of integer programming models. Even if the model code of the IP model is available to a user, the model code will need to be expanded if that user wants to add an additional line of defence. Granted, the extension is relatively straightforward (provided the user understands the linear programming model), but it is not a "blind copy-paste action". Contrary to this, our approach builds (and solves) the graph automatically for an arbitrary number of lines of defence. This will be further explained in the paper.

3) Furthermore, we clearly state in our aim that we want to achieve computational efficiency by means of reducing the number of risk calculations. In the IP approach by Zwaneveld and Verweij (2014a), the flood risk calculations are not considered as a part of the solving time. Instead, the risk calculations are assumed to have be carried out beforehand in their approach. If all flood risk calculations (and other calculations) are done beforehand, and the only issue at hand is the solving time of the algorithm, the IP approach will be more efficient than the greedy algorithm. We don't contest this in the paper. However, we don't consider the solving time of the graph algorithm as dominant, we think there are plenty of scenarios where the flood risk calculation time is dominant (and even limiting). In that case, we have shown that a greedy algorithm (i.e. one that does not necessarily visits all vertices) coupled with an efficient evaluation of the risk estimates (i.e. only calculate the risk if a vertex is actually visited, as opposed to calculating the risk for all existing vertices), is expected to result in fewer risk calculations for most situations. How many calculations will be saved depends wholly on the case study characteristics and discretization. This will be further explained in the paper.

In line with Brekelmans et al (2012), Eijgenraam et al, (2010; in revised version published as: Eijgenraam et al. 2016) a dynamic programming (read: shortest path approach) is identified in Zwaneveld (2012) as one of the options to solve the model. Zwaneveld and Verweij (2014a, Annex A, Figure A; 2014b) and Zwaneveld (2012) contribute to these earlier papers by identifying that the dike optimization problem can be seen as a graph based problem. For example, Zwaneveld and Verweij (2014b, p.29) state that the the dike optimization model 'satisfies the most fundamental of all network flow problems (Ahuja et al., 1993), namely the minimum cost flow model'. This point was missed by earlier published work and also by Dupuits et al. 2017. Dupuits et al. (2017) present a graph based representation which is identical to the graph representation of Zwaneveld and Verweij (2014a) and Zwaneveld and Verweij (2014b). For example, compare Figure A in Zwaneveld and Verweij (2014b, p. 29) with the figures of Dupuits et al. 2017. They are clearly (almost) identical.

4) Regarding the figures containing graphs being almost identical: we think that graphs with this kind of structure always look almost identical. Nevertheless, as already mentioned in 1), we will add references to the appendix of Zwaneveld & Verweij regarding seeing the flood defences optimisation problem as a graph.
Regarding the dynamic programming approach in the listed papers (Eijgenraam et al, (2010), Eijgenraam et al, (2016)): dynamic programming is mentioned there without specifying which dynamic programming algorithm is actually used. Dynamic programming can actually entail a number of algorithms (Cormen (2009)). Furthermore, in Cormen (2009), a clear distinction is made between dynamic

programming and greedy algorithms. We explicitly mention that we use a greedy algorithm. Therefore, (part) of our contribution is to use a greedy algorithm instead of a dynamic programming approach. This will be further elaborated upon in the paper.

Unfortunately, this paper by Dupuits et al. (2017) does not clearly refer to these previous papers and reports which they copy and build upon. In my opinion, this paper needs a thorough revision to correctly and clearly refer to the work of previous mentioned authors to meet minimal scientific standards.

5) see 1), where we address these citation issues.

This paper states in the introduction that "However, existing cost-benefit analyses tend to focus on flood defences with independent lines of (Kind 2014), or are not readily generically applicable (e.g. Zwaneveld and Verweij (2014a). Therefore, the aim of this paper is to find general, computationally efficient approach… with arbitrary number of lines"
The authors do not mention the fact that Zwaneveld and Verweij (2014a, including background papers), Bos and Zwaneveld (2012) and Zwaneveld and Verweij (2016, UK CPB discussion paper on previous Dutch reports) do present for the first time a generic , computationally approach to assess dependent flood defense systems whit arbitrary number of flood defense lines.

6) We choose the words "readily generically applicable" with the argumentation of 2) in mind. In 2), we acknowledge that the IP model is extensible, but extending it requires at least some editing. Also in 2), we indicated how we will make this clear in a revised version.

Moreover, these authors do apply their approach in a real world environment and under time pressure to obtain economic optimal flood protection policy measure for the Lake IJssel region (including many dependent dike rings and barrier dams). Dupuits et al. (2017) are aware of this approach and solution method since they apply it in section 4.3. Zwaneveld and Verweij kindly provided Dupuits et al. (2017) with their programming code and data to allow scientific reuse of their earlier work.
Although this approach is not yet published in in a UK written scientific journal (the authors are working on it, see Zwaneveld and Verweij , 2016)), the scientific quality had been assessed by two different committees with professors and other experts (see Donders et al., 2013; Van Ierland et al., 2014). This was due to the fact that very important hydrological and economic policy decisions are based upon the application of the Diqe-Opt model (in Bos and Zwaneveld 2012; Zwaneveld and Verweij 2014a). The Ministry of Infrastructure and Environment had to be sure about the quality of the Diqe-Opt model and the two reports. Documents are published on the UK and Dutch based CPB-website. A few documents are also presented to the Dutch Parliament: they can also be found at the website of the Dutch Parliament.
The latter 2014-committee of four professors at Dutch universities conclude in Dutch: "Het CPB heeft met deze studie een belangrijke stap vooruit gezet in het onderzoek naar waterveiligheid. Het is een indrukwekkende studie waarin zeer veel hydrologische en economische kennis op een prachtige manier wordt samengebracht. Met name het meenemen van afhankelijkheden in de overstromingskansen van dijken is een belangrijke innovatie. Het ontwikkelde model Diqe-Opt is een vernieuwend en zeer nuttig instrument" [UK translation: "CPB has made with this study an important step forward in the search for water safety. It is an impressive study in which very many hydrological and economic knowledge is combined in a wonderful way. In particular, the inclusion of dependencies in the flood dikes of opportunities is an important innovation. The developed model Diqe-Opt is an innovative and very useful tool"". Note that Zwaneveld and Verweij (2014) name their 3 generally applicable method: Diqe-Opt. Due to the generally applicable of the Diqe-Opt approach the model is by request from the Ministry of Infrastructure and the Environment being transferred to hydrological consultancy company Deltares. Deltares can use the model as long as proper references are made to earlier CPB-work by Zwaneveld and Verweij. The Dutch institutions setting of the CPB (employer of Zwaneveld and Verweij) prohibit these activities by CPB. This innovative Diqe-Opt approach was also recognized by a recent Dutch handbook on water safety (ENW, 2016, Literature list to Chapter 4).

Hence, Dupuits et al. (2017) should state that they copy the Diqe-Opt model by Zwaneveld and Verweij (2014a , 2014b) instead that the "aim to find an … approach". Proper and clear references are missing towards this earlier work in the starting sections of this paper.

7) We are thankful for sharing the model with us. We already referred to using it in Section 4.3, and we will extend our gratitude to the acknowledgements. Regarding the suggested change for the aim, missing references and the usage of 'copy': see 1) where we address the same comment.

Dupuits et al (2017) present a shortest path algorithm to the problem definition as presented by Zwaneveld and Verweij (2014a).

8) We interpreted this as a similar comment (regarding the problem definition/approach) as one that is already answered by our answer in 1). Therefore, see 1) for our answer.

For the cases presented in section 4.1 and 4,2, almost identical and probably more efficient dynamic programming approaches (shortest path approaches) are presented in Eijgenraam et al. (2010) and Brekelmans et al. (2012). Proper references should be made to this earlier work.

9) According Cormen et al (2009), greedy algorithms typically need less computational time than dynamic programming approaches. From that perspective, we do not see how dynamic programming can be more efficient than greedy algorithms. We will clarify this in the paper.
   Furthermore, we do not think we need to show an extensive comparison with existing solutions, as we do not present our approach as a competitor to these existing methods. See also our answer in 16) where we answer the same comment.

I do not see the value added by the shortest path approach of section 2 and section3 in addition to these two papers.

10) Because we believe we use a different algorithm to solve the shortest path approach, see also 4) where we mention the difference between dynamic programming and greedy algorithms.
11) The purpose of section 2 is to show the basic functioning of the greedy algorithm with respect to a (very) simplified problem, in combination with a graph. We believe this helps to create a fundamental understanding for engineers who are not familiar with the concept; as the intended target audience might not be as familiar with graph algorithms as an operations research audience.
12) Section 2.4 contains references which are relevant for the used greedy algorithm, and provides context to whether or not the shortest path (optimal solution) of a graph with non-negative edge weights is found.

Zwaneveld and Verweij (2014b , Annex A) present an alternative approach for these two cases in section 4.1 and 4,2 based upon an ILP-model. Applying this approach requires no programming effort whatsoever since user friendly software can be used to model the problem.

13) We addressed the implementation of I(L)P models in 2).

No efforts are required to solve the stated model since standard LP/IP solvers can be used. Note that an LP-solver is at present a plug in tool in Microsoft Excel and modelling languages as CPLEX, GAMS and AIMMS are easily available. Free and easy to use solvers are easily available.

14) CPLEX, GAMS and AIMMS are all tools that are geared towards applications with, for example, IP models such as used in Zwaneveld and Verweij (2014). This requires at least basic knowledge of IP problems, and knowledge of the specific tools. See also 2). This will be more clearly written and emphasized in the paper.

15) GAMS and AIMMS are algebraic modelling systems with proprietary licenses. CPLEX is a solver with a proprietary license. Free solvers, at least at the time of writing, don't scale well with thousands of decision variables and will have a hard time to solve the problems as discussed in Zwaneveld and Verweij (2014). On the contrary, our approach can be used in any general-purpose programming language, such as for example the freely available open-source languages Python and Julia. Inherently, this means that threshold for application and use is lower than with proprietary licenses. This will be more clearly written and emphasized in the paper.

The authors should mention in section 4.1 and 4.2 the use of these competitive and in some cases almost identical approach and should compare it with their approach. This comparison was already presented in Zwaneveld and Verweij (2014b, Annex A). They provide arguments and conclude that an ILP-approach is by far more preferable than a dynamic programming approach. Dupuits et al. (2017) should – as a minimum- discuss this work by Zwaneveld and Verweij (2014b).

16) The purpose of the (simplified) examples in section 4 is to show that the greedy algorithm works; not to show a comparison between various shortest path algorithms or IP models. See also 3) for the context of our method. However, given your comment, we will emphasize the reason why we are not showing a comparison. We will improve this description in the paper.

Furthermore, it is unclear why Dupuits et al 2017 conclude that an dynamical programming approach is 'relatively easy'. From a discussion with the authors, I learned that their intention and scientific ambition is to present a heuristic approach to solve the model by Zwaneveld & Verweij (2014a). Although heuristic approaches are presented in Zwaneveld & Verweij (2014a, 2016), these were not yet implemented. The advantages of this heuristic approach is to reduce calculation time with the disadvantage that a non-optimal solution is found. Furthermore, to help the authors, some persons may prefer a dynamic programming approach over an ILP approach. I also learned that a dynamic programming approach is easier to understand for many people than an ILP-approach. Therefore, Zwaneveld & Verweij(2014a, 2014b) always present their model as a graph problem and then introduce that they prefer to solve this graph problem to optimality by using an ILP-approach.

17) Our aim is to use a greedy algorithm (see also 4) for the difference between dynamic programming and greedy algorithms). We think that the UCS algorithm, as discussed in for example Fellner (2011), is easy to implement because the fundamental concepts related to UCS area taught in basic algorithm classes (Fellner, 2011). Furthermore, we consider the actual implementation in code as easy because of the brevity of the algorithm (see Fellner (2011), about 10 lines of pseudo code), coupled with the fact that the target audience (civil engineers) is more likely to be familiar with general purpose programming languages (see also 15)) than IP models (see also 14)).

However, the ambition by the authors, as I learned from personal communication with them, doesn't meet their statement on page 5 of Dupuits et al. (2017) : "For our applications, we did not come up with a heuristic function, which reduces the choice of a graph alfgorithm to either Dijkstra or UCS". Hence, this requires more explanation by the authors. I cannot see why both claims are valid.

We interpreted this comment as that the reviewer sees a greedy algorithm as a heuristic. This interpretation is answered in 18). However, we meant to discuss greedy algorithms with an optional heuristic function, which is clarified in 19).

18) An attribute of greedy algorithms is that, while efficient, they sometimes don't find the shortest path in a problem. We assume this is what the reviewer means with a heuristic. However, in Section 2.4 (see also 12)) we think we provide some references containing strong arguments for the greedy algorithm that it does find the optimal path for the discussed application area. Therefore, we refrained from calling the greedy algorithm a heuristic.
19) A heuristic function, as mentioned in the paper, relates to a subset of greedy algorithms. Heuristic functions can be used with a greedy algorithm in order to give additional information in an attempt to speed up the finding of the shortest path. A graph algorithm which can use such a heuristic is the A* algorithm. If the heuristic function always returns zero (i.e. no heuristic function), the A* algorithm reduces to the Dijkstra algorithm or the closely related UCS algorithm. This is what we meant with a heuristic function. We will improve this description in the paper.

For the case presented in section 4.3 proper references should be made that this is a simplified version of Zwaneveld and Verweij (2014a).

20) We do not see how a case as simple and general as shown in section 4.3, which in this general form can be found in many coastal areas around the world, should be referred to as a simplified version of the case presented in Zwaneveld and Verweij (2014a). We think that the fact that we use (and refer to) the method of Zwaneveld and Verweij (2014a) as a benchmark for the correct answer implies that the method of Zwaneveld and Verweij (2014a) can be applied to the case study as well.

Again, an explicit discussion of the pro's and cons of solving the model by Zwaneveld and Verweij (2014b) by using an ILP-approach and their approach should be presented. Zwaneveld and Verweij (2014a) and Zwaneveld (2012) do present such a comparison and they conclude that the ILP-solution approach is superior to dynamic programming (or: shortest path) for real-life applications. This earlier assessment should be presented. Why do Dupuits et al. (2017) conclude the opposite?

21) After careful re-reading of our own words, we cannot find any evidence that suggests we even compare the performance of ILP to our greedy algorithm, let alone conclude that the greedy algorithm is superior. The only qualification we make is that the outcomes are equal, with the Zwaneveld and Verweij (2014a) method being the benchmark.

Why do the authors present only 'toy problems instances'which can easily be solved using existing approaches?

22) Because a more complex case study would shift focus from the topic, which is to explain the application of graphs in combination with a greedy algorithm in the context of computationally expensive risk calculations (A more complex case study would require a significant amount of introduction regarding the assumed hydrodynamic interaction between multiple lines of defence, for example). The main point of the simplified examples is to show, in conjunction with earlier mentioned points in 12) and 18) and section 2.4, that the optimal path (or a path close to the optimal path) is found. A more complex case study will be an integral part of a follow-up research. We will improve this description in the paper.

Zwaneveld and Verweij (2014a) and Bos and Zwaneveld (2012) were capable of solving very large real-time problem instances given very short research leadtime and research capacity.

23) Our approach has different goals than the mentioned approaches. This was already touched upon in 2), 3), 14) and 15). These arguments in summary: the computational cost of risk calculations is an issue, knowledge of IP models (and specifically the model by Zwaneveld and Verweij) is not commonly present, and various valid licenses for relevant software packages are absent. For our intended use (see also our aim of the paper) and the intended target audience, we do consider the replies of 2), 3), 14) and 15) highly relevant.

Moreover, setting up a dynamic programming algorithm is requires very substantial programming efforts as is clear from section 2 and 3 from this paper. The approach by Zwaneveld and Verweij (2014a) requires only the code "SOLVE DIQE-OPT MODEL USING CPLEX" to obtain the proven optimal solution. Hence, the claim that a dynamic programming is 'more easy' than a ILP-approach by Zwaneveld and Verweij (2014a) is not valid or – at best - not properly motivated in my opinion.

24) We do not agree with this statement. See also 23) for a summary about the relevance of our approach. In our experience, re-creating the model of Zwaneveld and Verweij (2014a) took a roughly equal amount of programming (We had to recreate the model due to an absence of the necessary proprietary licenses for GAMS and CPLEX). This excludes the amount of time it took to familiarize ourselves with the IP model by Zwaneveld and Verweij and its inner workings. Furthermore, even though our approach did require a programming effort, we can share this code with anyone who has access to a computer, because it was coded using a freely available open-source language. Therefore, in principle, no additional coding effort is required by third parties regarding implementation of graphs and the greedy algorithm. We will improve this description in the paper.

Specific comments

Section 2.1:

The representation of the problems copies the approach by Zwaneveld and Verweij (2014a) and Zwaneveld and Verweij (2014b). Especially the graphs in this section are strikingly identical to Figure A from Zwaneveld and Verweij (2014a) See also almost identical figures in Yuceoglu (2015). Also references should also be made to Dynamic programming approach by Eijgenraam et al. (2010) and Brekelman et al. (2012) which seems to be mathematically identical. Proper references are missing to this earlier work. Dupuits et al. (2017) should clearly state that they copy previous work.

The cases presented in paragraph 4.1 (single flood defense) and 4.2 (independent lines of defences) can be solved by the dynamic programming (or shortest path approach) which is extensively discussed in Eijgenraam *et al (2010) (a revised version of this paper was published as Eijgenraam et al 2016) and briefly discussed in Brekelmans et al. (2012). Zwaneveld & Verweij (2014b, 'paper under revise and resubmit') make the point in Annex A that these shortest path problems can be much easier solved to proven optimality using LP-relaxation or IP-model formulation. All this should be mentioned.*

25) This remark seems to repeat a number of earlier made remarks, which we answered in our earlier replies. Addendum: Our focus is not to find the most efficient graph algorithm, our focus is to find a graph algorithm which handles risk calculations (most) efficiently. In our opinion, this makes the requested addition of earlier made comparisons between dynamic programming and IP models non-relevant (at least not relevant for this paper).

Section 2 and 3:
The presented approach is basically the well known shortest path algorithm. The discussion should can be deleted or removed to an electronic companion . I do not see any scientific added value in comparison with earlier work by Brekelmans et al (2012), Eijgenraam et al (2010) and the large literature of shortest path problems and dynamic programming. I personally prefer to refer to the well-written UK –based Wikipedia discussion of the subject(see Zwaneveld, 2012).

26) As mentioned in previous points:
- in our opinion we don't use dynamic programming (we use a greedy algorithm).
- Furthermore, we think section 2 serves as an essential introduction for non-experts in the domain of graph optimization, which we think is relevant given the journal's audience. We will make the introduction of section 2 clearer to better reflect this goal.

The claim that repetitiveness of vertices is in most cases incorrect. Note that vertex 12 represent a later year than vertex 7 (see Figure 9) . Due to yearly increases of economic growth and flood probabilities al risk calculation has to be calculated again. Hence, vertex 7 and 12 are not identical and no calculation time is saved.

27) We disagree with this comment. Risk calculations are not mentioned in section 3.1. We use the repetitive characteristics to reduce the size of the data structures belonging to the vertices and edges.

Note that Brekelmans et al. (2012, p.1343) state that a simple 'homogenous case can be conveniently solved using dynamic programming. Unfortunately, this is not possible for the nonhomogenous cases, because the state space explodes…. We show how the nonhomogeneous diek height problem can be solved as a MINLP-problem.' A more or less similar statement by Brekelmans et al. (2012, p. 1345): "Unfortunately, the state space grows too large ….which implies that the dynamic programming approach is not applicable". This is the – very good- reason why Brekelmans et al. (2012) prefer their MINLP approach.

Dupuits et al (2017) do not properly discuss this exploding problem, i.e. exploding state spaces and exponential calculation time of all sorts in the problem size. Nor do they refer to these previously mentioned authors which did identify this problem before.

28) We do not use dynamic programming, we use a greedy algorithm. Furthermore, we believe the state space explosion can be partly negated with the help of 3.2 and

particularly with section 3.4 (if 3.4 is applicable), because these techniques can reduce the amount of edges and vertices (and therefore the computational time).

From the paper I have got the impression that they apply a shortest path algorithm to solve the problem to proven optimality given – theoretically- computing time which are exponential in the problem size. From personal communication with the authors, I did get a different impression, namely that they aim to present a non-optimal solution approach given limited computing time. The authors should clarify that ambition.

A more or less similar remark holds for the algorithm. From the paper I get the impression that they implemented the algorithm themselves to find a proven optimal solution. From personal communication, I did get the impression that they use standard plug-in heuristic procedures to solve the graph. Hence, no programming effort whatsoever is required. The latter would make their approach of course more easy to use but also make their algorithm less innovative. The authors should clarify their ambition.

29) Unfortunately, it seems like we didn't make this sufficiently clear during our personal communication. We use an existing greedy shortest path algorithm, which we tried to implement in an efficient manner for the particular problem of economic optimization of multiple lines of defence. Efficiency was sought in the associated graph data structures and in efficient evaluation of risk calculations (i.e. Section 3). See many of our earlier replies, for example 12), 18), 19) and 22), for further clarifications and replies.

Section 5:
The authors state that Kind (2014) proposes an linear programming approach. This is incorrect. Kind (2014) doesn't propose any method. He uses the approach by Brekelmans et al. (2012), which is an MINLP-approach. The IP-approach was proposed by Zwaneveld and Verweij (2014b, 'paper in revise and resubmit' to an academic journal). An IP-approach is not identical to a linear programming approach.

30) Thank you for these suggestions. We will improve these references.

The claim that the application area is roughly similar to Zwaneveld and Verweij (2014a) is incorrect. The application area is completely identical and copied from Zwaneveld and Verweij (2014a). Furthermore, reference should be made that dynamic programming/shortest approaches of cases in section 4.1 and 4.2 to Eijgenraam et al (2010) and Brekelmans et al. (2012). And to heuristic ideas (and some attempts) to solve the dike height problem in previous work by Eijgenraam, Brekelmans and Den Hertog and Zwaneveld & Verweij (2014a, 2014b, 2016)

31) We refer to our answer of 1) regarding the suggested use of the word 'copy'. See also our answer in 3) for our aim, which (although similar) emphasizes different aspects than Zwaneveld and Verweij. Therefore, we refrained from using words such as 'identical' and 'copy'. We believe our proposed approach is complementary to the approach by Zwaneveld and Verweij (and other earlier proposed methods); it is not meant as a replacement.

Lines 17-22 Page 17: The authors should mentioned that fact that the approach by Zwaneveld and Verweij (2014b) was especially develop to include other flood defence systems than height-dependent dikes.

32) Thank you for this suggestion. We will add this to the relevant parts of the paper.

The fact that Dupuits et al (2017) can also include these approaches is a direct consequence of the fact that they copy the approach by Zwaneveld and Verweij (2014a, 2014b) and, therefore, both have identical application areas. The Diqe-Opt model was already used to asses many of these alternative flood defence systems in Bos and Zwaneveld (2012) and Zwaneveld and Verweij (2014a). See also Donders et al. (2013) and van Ierland et al. (2014)

33) Regarding the use of qualifications like 'copy' and 'identical application areas': See 1) and 3). See also 32) for properly referring to using alternative flood defence systems with Zwaneveld and Verweij (2014).

Section 6

The authors claim that it is an advantage that 'their approach do not need pre-calculate risk which linear programming approaches do'. However, the IP-approach by Zwaneveld and Verweij(2014a) – again this is NOT a linear programming approach – indeed does require risk estimates in a pre-processing step. In addition, stating that risk calculation can be performed 'on the fly' is complete impractical in a real-world setting of Zwaneveld & Verweij (2014a), Bos and Zwaneveld (2012), Brekelmans et al (2012) and Eijgenraam et al. (2016), since it requires in general running hydrological models. Hence, the approach by Dupuits et al. (2017) requires in each iteration to consult a hydrological experts to run their model and to report the result back. Doing these calculations in a pre-processing step as advocated by Zwaneveld and Verweij (2014a and 2014b) and Bos and Zwaneveld (2012) has very significant practical advantages. For real-world instances, risk calculation were no problem whatsoever in the approach by Zwaneveld & Verweij (2014a, 2014b), Brekelmans et al. (2012) and Eijgenraam et al. (2016).This argumentation is missing in this section.

34) We disagree that each iteration requires consulting a hydrological expert. In follow-up research, we are calculating risks 'on the fly' in a case study. We do not know the details of the referred to risk calculations, but we can predict that in the setting of this paper (multiple lines of defence of which the risk of a downstream defence is assumed to be dependent on all upstream defences), risk calculations will not be computationally cheap. This will be further emphasized in the paper.

Finally, the claim by Depuits et al. (2017) that their approach requires less risk calculation than the graph-based ILP—approach by Zwaneveld and Verweij(2014a) is not supported by calculations.

35) See 3) where we argue why we cannot predict the amount of savings (depends on the case). We did mention the savings of the Section 2 example, specifically in figure 12 of section 3.3. Furthermore, we will add the number of risk calculations to the examples of Section 4.