ELSEVIER
CrossMark
Available online at www.sciencedirect.com
ScienceDirect
Procedia Engineering 70 (2014) 1505 – 1512
Procedia Engineering
www.elsevier.com/locate/procedia

12th International Conference on Computing and Control for the Water Industry, CCWI2013

Artificial intelligence techniques for flood risk management in urban environments

W. Sayers[a,b], D. Savić[a], Z. Kapelan[a]*, R. Kellagher[b]

[a] *University of Exeter, Exeter, EX4 4SB, UK*
[b] *HR Wallingford, Wallingford, Oxfordshire, OX10 8BA, UK*

Abstract

Urban flooding is estimated to cause £270 million pounds worth of damage each year in England and Wales alone. There has, therefore, been a clear need to develop improved methods of identifying intervention strategies to reduce flood risk in urban environments. This paper describes ground-work performed towards evaluating the relative suitability of several algorithms applied to multi-objective optimisation of flood risk intervention strategies in an urban drainage network. An effective methodology is described for reducing an array of return period/duration rainfall files to a minimum, and it is described how this methodology makes possible comparisons of optimisation algorithms. This work has been undertaken as part of a STREAM-IDC EngD project which is a collaborative effort between the University of Exeter, and HR Wallingford.

Keywords: Innovation; Technology; Multi-objective; Optimisation; Optimization; Drainage; Floods; Flooding

1. Introduction

Urban flooding (usually caused by rainfall overwhelming drainage systems) is a serious problem which is estimated to cost £270 million per year in England and Wales with 80,000 homes at risk according to a report by the Parliamentary Office of Science and Technology (2007). In addition to water damage to properties and businesses there is a significant humanitarian cost in terms of both physical and mental health issues as a result of

* Corresponding author. Tel.: +44 (0)1392 724054; fax: +44 (0)1392 217965.
*E-mail address:*z.kapelan@exeter.ac.uk

doi:10.1016/j.proeng.2014.02.165

disease, forced relocation (temporary or otherwise) and property damage. Looking at these facts and estimates it is clear that there has been a need to develop improved methods for identifying the most cost-effective intervention strategies that best mitigate damage from flooding events. HR Wallingford, in partnership with the department of trade and industry (now the department for business, innovation and skills) , has recently developed a project titled DTI-SAM (Department of Trade and Industry, System-based Analysis and Management of urban flood risks) (see the reports by HR Wallingford (2009) and Kellagher et al. (2009)). In the DTI-SAM project, a risk-based methodology was produced along with an accompanying toolset for assessing the flood damage likely within a given area utilizing a given drainage network, an Infoworks CS model and a rapid flood spreading model developed in-house, described in Lhomme et al. (2008). The flood damage is assessed in terms of expected annual damage (EAD) which is a cost-based measurement representing the expected average infrastructure costs as a result of flooding each year. This toolset has been modified in this work to allow it to be easily run in an automated fashion, and re-branded as ADAPT (A Drainage Analysis and Planning Tool). As part of this modification, the simulation algorithm and the user interface were separated, and a new user interface developed that combines control of the simulation with control of the optimisation algorithms. In addition to this, a simplified costing model has been developed to estimate the cost (in terms of contractor charges and material expenses) of making a given set of alterations to a drainage system. The EAD and the Cost model are then to be used in a multi-objective algorithm, which aims to balance the two factors and present a pareto front, as originally described in Pareto (1896), of potential solutions. This Pareto front can then be utilised by a trained Engineer to guide decision making.

The selected optimisation methodology will be based on a multi-objective genetic algorithm, the non-dominated sorting genetic algorithm, version two (NSGA-2) as described by Deb et al. (2002). The new methodology incorporates the option of using artificial neural network meta-models in a similar fashion to Behzadian et al. (2009), or the learnable evolution model for multi-objective optimisation (LEMMO) studied in Jourdan et al. (2005).

Estimating the cost of changes to the drainage network is a straightforward process, completed practically instantaneously on any modern computer. The process of estimating EAD, however, is extremely computationally intensive when using all possible rainfall data, i.e., return periods and durations (taking around 5 hours for a full run). A full evaluation would lead to an unfeasibly lengthy run-time if a full optimisation utilizing all this data were to be attempted. Therefore in order to make the use of this algorithm feasible, the possibility of reducing the amount of rainfall data utilised have been investigated. This was done with the aim of having the least possible impact on the accuracy of the EAD figure generated (in particular, the relative differences between EAD figures must be maintained, in order for the multi-objective algorithm to function correctly). Once this process is complete, the user will be able to gain an estimation of the Pareto front with reasonable accuracy. This, in turn, will allow comparing optimisation algorithm options in terms of how quickly they achieve a reasonable approximation of the Pareto front.

Nomenclature

ADAPT	A Drainage Analysis and Planning Tool
DTI-SAM	Department of Trade and Industry, System-based Analysis and Management of flood risks
EAD	Expected Annual Damage
LEMMO	Learnable Evolution Model for Multi-objective Optimisation
NSGA-2	Non-dominated Sorting Genetic Algorithm version 2
RFSM	Rapid Flood Spreading Model

2. Methodology

This section describes the methodology of the work being performed, starting with a definition of the problem that is being optimised, followed by the calculation of EAD, a description of the cost calculation, and finally descriptions of the optimisation algorithms that have been implemented and will be tested using the methodology outlined in this paper.

2.1. Problem definition

The problem, to which a solution is ultimately being sought by work performed within this paper, is that of drainage networks being overwhelmed by extreme rainfall events in urban environments. Urban areas have little natural protection in the form of open ground permeable to rainfall as well as ground surfaces that are effectively a series of complex channels, and the impact on drainage networks and the effects of subsequent flooding is magnified by this. The companies responsible for maintaining urban drainage networks, when making changes in efforts to reduce these flooding events, naturally look for the most effective and solution which fits their available budget for flood prevention. They will also be interested in any means by which they can save money, if they can still gain most of the benefits of a more expensive solution.

Historically, this problem has been solved by largely non-automated efforts, with engineers utilising their expertise to suggest flood risk intervention strategies, then analysing those strategies in terms of cost and risk reduction. As the DTI-SAM project has allowed the automated computational evaluation of risk, given a drainage model and the accompanying above-ground data, it is proposed to run an automated multi-objective optimisation algorithm, analysing risk vs. cost. This should result in a range of possible solutions (from least-cost and least-effect, to most-cost and most-effective); an engineer can then employ expertise to identify the most promising solutions within this range. The benefits of this approach are that the engineer in question should find this method less intensive of time, and there's always a possibility of an optimisation algorithm producing solutions that an Engineer may not have considered for various reasons, which could then be utilised as is or tweaked for best effect.

The main issue with this approach is the computing time required to perform the multiple risk and cost analyses to conduct an optimisation. Whilst the costing is a relatively straight-forward algorithm and is not overly heavy in terms of processing power, the EAD (risk) assessment is considerably more intensive. In order to utilise this EAD measure, therefore, the use of heuristics within the multi-objective optimisation is suggested so that run-times will remain feasible. The heuristics that we are proposing are the use of meta-models within the multi-objective algorithm to estimate the objective function (cost and EAD) values, and/or the inclusion of a machine-learning based algorithm as part of the multi-objective algorithm. In order to evaluate these heuristics in terms of how effectively and how quickly they converge to a good estimation of the pareto-front we are searching for, we first need an estimate of that Pareto front. This estimate of the Pareto front is the output of the work performed in this paper.

2.2. Calculation of EAD

Expected Annual Damage (EAD) is an estimation of the amount of damage (in pounds sterling in the models utilised here) that will occur on average over a given year. A number of design storms are needed for this calculation, ranging from a return period of two to one thousand years, and durations of thirty to six hundred minutes (Table 1). Altogether the set used in this work contains 700 design storms (see Table 1) each of which requires one simulation run of Infoworks CS in order to gain EAD from the full set.

Each of these 700 design storms is run through an Infoworks CS model, which identifies excess water volume at each manhole node on the network. This excess water volume is then utilized by the rapid flood-spreading model (RFSM) which is an in-house developed application at HR Wallingford. RFSM identifies flood depth across a surface map of the terrain that the drainage system covers. It then compares this data with data on infrastructure in the area, and calculates an estimated damage for each design storm. These rainfall events are evaluated shortest return period first, then in ascending order until convergence of EAD figures is achieved, or there are no rainfall files left.

Table 1. Design-storm rainfall files.

Parameter	Data							
Return period (yrs)	2	5	10	… (in steps of 10)	300	500	750	1000
Duration (min)	30			So on, steps of 30				600

2.3. Calculation of cost

The secondary objective of the multi-objective algorithm is “cost”. In this case it means the cost of the proposed changes to the network. The constants in the cost calculation are all customizable (they are constant in the sense that they should not change during the algorithm run), and sensible defaults are provided to give a “rough idea”. These defaults have been used for testing purposes for all experiments described in this paper, and they are listed at the end of this section in Table 2.

All cost calculations initially determine whether a network has been modified at all, or whether it is identical to the original network. All modified networks cost calculations include an initial sum to reflect the “mobilization” costs – i.e. the costs of hiring contractors, getting them and their equipment on site, and other associated costs with initially beginning a task of this nature. The pipe alteration costs are then estimated by multiplying the product of a constant “Intervention Cost” value (I) to represent costs of piping purchase, excavation, etc. and the length of pipe (Eq. l) in question by the cross-section area (c) of the pipe. Storage alteration costs are estimated by adding the product of an “Intervention Cost” constant (S) signifying cost of materials, etc., and the area of the storage node in meters squared (a), to a second “Base Cost” constant (b) that represents the costs associated with excavation, removal of existing storage node if necessary, etc. All non-modified pipes or storage nodes result in zero cost.

Therefore the total cost of a network is the mobilization cost (M), plus the cost of each modified pipe in the network, calculated as described above, plus the cost of each modified storage node in the network (see Eq. 1).

Table 2. Default constant values for cost calculations.

Mobilization Cost (M)	Pipe Intervention Cost (I)	Storage Intervention Cost (S)	Storage Base Cost (b)
£50,000	£1,000	£500	£10,000

$$C = M + \left(\sum_{i=0}^{n} I \times l_i \times c_i \right) + \left(\sum_{k=0}^{n} S \times a_k + b \right) \tag{1}$$

2.4. Multi-objective optimisation

In these studies the NSGA-2 multi-objective genetic algorithm is utilized with additional improvements based on the study performed by Behzadian et al. (2009). The aim is to re-purpose this algorithm for the design of flood intervention strategies for urban drainage systems, as well as improve upon its performance for this task. The implementation of NSGA-2 has been written specifically for this project, in order to gain optimum integration with the balance of the ADAPT toolset.

In addition to the straight NSGA-2 algorithm, a meta-modelling process utilizing artificial neural networks and mirroring the approach used by Behzadian et al. (2009) is included, whereby the NSGA-2 algorithm is first run with the normal objectives (e.g. the cost model and the full EAD modelling and calculation steps) for a specified number of steps, and the EAD values generated during this time, as well as the networks that generated them, are internally stored. This data is then utilized to train an artificial neural network to approximate EAD values from a set of inputs representing a drainage network. From this point on in the algorithm, the ANN is utilized to estimate EAD values initially. All solutions that the combination neural network and cost models place in the first ‘n’ ranks (with ‘n’ being user-defined) are then re-evaluated utilizing the standard EAD estimation algorithm. All of these re-evaluations are stored (both EAD data and network data, as before) and periodically throughout the algorithms process (every ‘i’ iterations, with ‘i’ being user-defined) this data is utilized to further train the neural network, introducing further accuracy in the neural network for distinguishing between increasingly promising solutions, as the overall algorithm progresses and these promising solutions should become more common.

Finally, the “learnable evolution model for multi-objective optimization” or LEMMO algorithm which is a multi-objective implementation of the “learnable evolution model” or LEM algorithm developed by Michalski et al. (2000) is also implemented. This algorithm can be utilized as an addition to an evolutionary algorithm by various methods described in Jourdan et al. (2005). The LEMMO-fix4 solution trialled in that paper was found to be the most effective by several measures, so our implementation is as close to that as possible. The implementation that has been developed utilises an artificial neural network, rather than the AQ algorithm described in Michalski et al. (1980) and Michalski et al. (1983) that is more commonly utilized in LEM algorithm implementations, or the C4.5 decision tree induction system utilised in the LEMMO implementation by Jourdan et al. (2005). This modification allows advantage to be taken of the already-developed ANN solution for the ADAPT meta-modelling process and should function equally well at identifying which networks are superior and which inferior.

3. Simplified EAD assessment

Due to the large number of EAD calculations necessary during a multi-objective optimisation scenario, it is important that every EAD calculation takes place using the minimal computing resources possible. The standard set of design-storms previously utilised (during DTI-SAM) provides a good approximation of EAD, but was not designed to be part of an optimisation algorithm. Within the internals of an optimisation algorithm it is possible to a certain extent afford to lose accuracy, provided that the change in EAD remains relative across intervention strategies, allowing differentiation between higher and lower quality solutions. Therefore a possible approach to improving the computational efficiency of producing EAD estimations is to reduce the number of return periods/durations of rainfall that are being evaluated for each EAD estimation.

Previous studies such as Woodward (2012) and Ward et al. (2011) have shown that reducing resolution of the problem space being evaluated can result in significantly lowered model run time, but does significantly affect accuracy of results. In Woodward (2012) it was found that significant performance improvements could be obtained by reducing numbers of return periods, whilst retaining sufficient accuracy for the test results to be useful as part of a multi-objective optimisation algorithm.

3.1. Testing plan

Initially it is important that testing is performed on more than one network, therefore a minor but important first step is the generation of twenty testing networks from our initial network. Once this has been accomplished, a full EAD estimate will be generated for each network, taking roughly five hours per network. Tests will be performed with different durations initially (in an attempt to identify a smaller combination of durations that will give results as close as possible to the original EAD estimation), followed by tests utilising these durations but analysing fewer return periods. This should hopefully give us a combination of a few return periods and durations which when run, give a good approximation of the EAD for a given network that maintains the relative variance in EAD between different drainage networks.

3.2. Generation of test solutions

It is critical that any solution proposed is effective across more than one drainage system – as the NSGA-2 algorithm will be generating many variants of the original drainage system in order to identify the optimal flood risk intervention. Therefore a number of drainage systems are needed to perform testing on. Therefore twenty different networks have been generated which will be used as test networks. These networks were generated by selecting three rainfall files to represent one fairly small & fairly common storm (return period thirty years & duration thirty minutes, run A), one reasonably sized not-so-common storm (return period one hundred and seventy years & duration three hundred minutes, run B) and one highly unusual & very extreme storm (return period one thousand years & duration six hundred minutes, run C).

The NSGA-2 algorithm was then run three times, with each run only evaluating EAD via one of these rainfall files. This does not give a good approximate EAD, but gives the NSGA-2 algorithm some value to work towards improving (essentially one optimisation run is biased towards reducing damage from fairly regular storms, one towards rare storms, and one towards very extreme storms). Each run was for fifty iterations with a population size of twenty. This gave a grand total of sixty potential testing networks to choose from once the algorithm had run its course, so from each run a spread of networks (seven from run 'A', six from run 'B', and seven from run 'C', for a total of twenty) was taken which had well distributed EAD scores to be our test networks.

3.3. Full EAD evaluation

For each of the twenty selected networks, a full EAD evaluation was then run so that a base figure to compare our runs with fewer return periods and durations could be identified. Each run produces a curve as the return periods are iterated through, as shown below in Figure 1, which is the curve for "Net 5". The final results of these runs are shown in Table 3, Table 4 and Table 5.

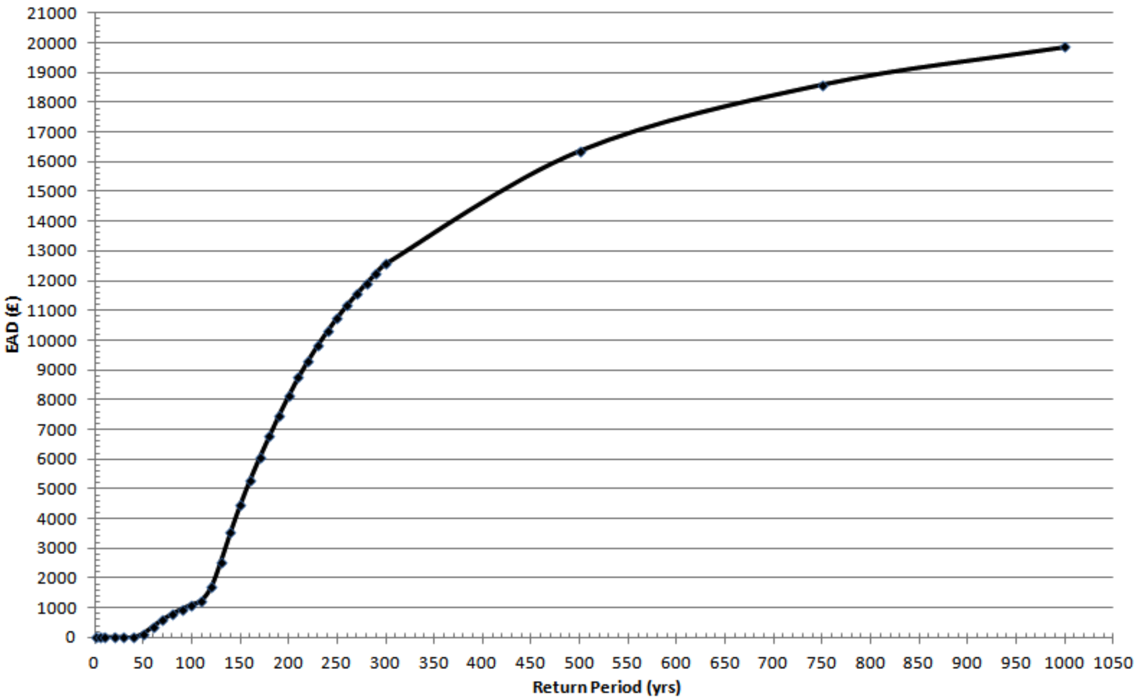

Figure 1. Example of EAD curve (Net 5)

Table 3. Full EAD figures networks 1-7

Net 1 (£)	Net 2 (£)	Net 3 (£)	Net 4 (£)	Net 5 (£)	Net 6 (£)	Net 7 (£)
2968.5	4948.6	11765.3	12180.9	19860.6	1540.1	1938.8

Table 4. Full EAD figures networks 8-13

Net 8 (£)	Net 9 (£)	Net 10 (£)	Net 11 (£)	Net 12 (£)	Net 13 (£)
11045.4	12407.4	12429	12440	15362.9	15706.1

Table 5. Full EAD figures networks 14-20

Net 14 (£)	Net 15 (£)	Net 16 (£)	Net 17 (£)	Net 18 (£)	Net 19 (£)	Net 20 (£)
15303.9	16384.3	17618.7	17773.3	17977.0	17985.3	18355.8

3.4. Duration testing

Once these initial tests were completed and base figures for EAD gained, tests were initially run with fewer durations. On all 20 networks a single duration of 600 minutes (i.e. the most extreme duration) produced the same or very close to the same EAD score on all networks (see Table 6, Table 7 and Table 8). The mean error of runs using only 600 minute duration rainfall versus using all rainfall was 14.51, which given the estimated nature of EAD and the size of the sums in question, is extremely minor. The decision was therefore taken to proceed with utilising only 600 minute duration rainfall files, and proceed with attempting to reduce the number of return periods being analysed.

Table 6. EAD values from runs using 600 durations only, networks 1-7

Net 1 (£)	Net 2 (£)	Net 3 (£)	Net 4 (£)	Net 5 (£)	Net 6 (£)	Net 7 (£)
2691.5	4948.6	11764.9	12180.9	19857.5	1540.0	1938.8

Table 7. EAD values from runs using 600 durations only, networks 8-13

Net 8 (£)	Net 9 (£)	Net 10 (£)	Net 11 (£)	Net 12 (£)	Net 13 (£)
11045.4	12407.4	12429	12440	15362.9	15706.1

Table 8. EAD values from runs using 600 durations only, networks 14-20

Net 14 (£)	Net 15 (£)	Net 16 (£)	Net 17 (£)	Net 18 (£)	Net 19 (£)	Net 20 (£)
15303.3	16383.4	17617.4	17770.6	17976.2	17983.2	18354.6

3.5. Return period testing

Initially, a good spread of return periods were selected, with the goal of narrowing down which area of the spectrum requires the most focus. Previous work where this kind of reduction in return periods has been accomplished, such as Woodward (2012), with minimal impact on accuracy suggests the use of seven return periods in their case. Therefore the initial number of return periods chosen was similar, with eight return periods (2, 10, 100, 200, 300, 500, 750 and 1000 years) chosen. Analysis of this initial test (mean absolute error 4846) suggested that the resolution of the data was too low at the bottom of the range, where too much inaccuracy in identifying the return period at which flooding begins has a large effect on the rest of the EAD curve. With this in mind, a second test was undertaken with only seven return periods but with much more focus at the lower end of the range (return periods 2, 20, 40, 80, 160, 750, 1000). The results from this second test (mean absolute error 2220) matched the full EAD curve far better than test 1 at the lower end of the range, but generally went wide (either high or low) towards the middle of the curve. A third test, utilising return periods 2, 20,40, 80, 160, 300, 500, 750 and 1000 years, was then run in an attempt to compensate for this by providing better resolution of data at critical points along the EAD curve. This third test performed better (mean absolute error 1570), but with still a lack of resolution at the low end for some networks, although generally better across the middle range now. A fourth test was performed with fewer of the low-end return periods, but focusing on the low-mid range slightly more (return periods 2, 80, 160, 300, 500, 750 and 1000 years), in an attempt to cut down on the total number of return periods whilst maintaining accuracy. However this test resulted in further inaccuracies (a mean absolute error of over 13,000 in fact, the highest of all tests).

4. Conclusion

The key finding of this work is the methodology of generating a number of test problems by utilising the optimisation algorithm with a simple objective function that approximates in some broad fashion a fully representative function (in this case one RP and duration, with the RP and duration varied over the 20 runs to avoid introducing too much bias), generating a full EAD for each of the test problems generated using this method and then attempting to identify the key duration/s by comparing different durations vs. the full EAD, and key return periods by examining the EAD curves for key points where the EAD curve experiences changes. This can then be further narrowed down by testing and adjusting your selection of return periods and/or durations, until a set is found that gives good enough results from an optimisation viewpoint and takes considerably less time to complete than the full set. The initial full EAD runs took roughly five and a half hours each, in comparison, solutions analysed using the rainfall set in test 3 took roughly ten to fifteen minutes, which is a major time saving.

4.1. Future work

Moving on from this work, a methodology has been developed which can be used to identify the return period and duration combinations that will yield a reasonable approximation of the EAD value for a particular network, whilst maintaining the differences between networks to allow discrimination of poorer or better performing solutions. This knowledge can be utilised to compare standard NSGA-2 with other algorithms in terms of how quickly they achieve a reasonable approximation of the Pareto front that is being searched for and therefore identify the most effective algorithm for this particular problem.

References

Behzadian, K., Z. Kapelan, D. A. Savic, and A. Ardeshir. 2009. "Stochastic Sampling Design using Multiobjective Genetic Algorithm and Adaptive Neural Networks." *Environmental Modelling & Software* 24 (4) pp. 530-541.

Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6 (2) pp. 182-197.

HR Wallingford. 2009. SAM - System-based analysis and management of urban flood risks. Oxford.

Jourdan, L., D. Corne, D. Savic, and G. Walters. 2005. Preliminary Investigation of the 'Learnable Evolution Model' for Faster/Better Multiobjective Water Systems Design. In *The Third International Conference on Evolutionary Multi-Criterion Optimization*, edited by Carlos Coello Coello, Arturo Hernández Aguirre and Eckart Zitzler. Guanajuato, Mexico: Springer.

Kellagher, R. B. B., Y. Cesses, M. Di Mauro, and B. Gouldby. 2009. An Urban Drainage Flood Risk Procedure, A Comprehensive Approach. In *The WaPUG Annual Conference*. Hilton Hotel, Blackpool: HR Wallingford.

Lhomme, Julien, Paul Sayers, Ben Gouldby, Paul Samuels, Martin Wills, and Jonatan Mulet-Marti. 2008. Recent Development and Application of a Rapid Flood Spreading Method. In *FLOODrisk*. Keble College, Oxford, UK: HR Wallingford.

Michalski, R. S., and R. L. Chilausky. 1980. "Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis." *International Journal of Policy Analysis and Information Systems* 4 (2) pp. 125-160.

Michalski, R. S., and J. Larson. 1983. Incremental Generation of VL1 Hypotheses: The Underlying Methodology and the Description of Program AQ11. Urbana: Department of Computer Science, University of Illinois.

Michalski, Ryszard S., Floriana Esposito, and Lorenza Saitta. 2000. *Learning Evolution model: Evolutionary Processes Guided by Machine Learning*. Vol. 38, *Machine Learning*.

Pareto, Vilfredo. 1896. *Cours D'Economie Politique Vol. I & II*. Lausanne: F. Rouge.

Parliamentary Office of Science and Technology. 2007. "Urban Flooding." *postnote* 289.

Ward, P. J., H. de Moel, and J. C. J. H. Aerts. 2011. "How are flood risk estimates affected by the choice of return-periods?" *Natural Hazards and Earth System Sciences* 11 pp. 3181-3195.

Woodward, Michelle. 2012. *The use of real options and multi-objective optimisation in flood risk management*, CEMPS, University of Exeter, Exeter, UK.