Improvements in Water Resource Planning

Oleg Barbin, Michael O'Sullivan, Cameron Walker
University of Auckland
Department of Engineering Science

Abstract

In the face of ever-changing water supply and demand, UK water utility companies are obligated to provide long-term plans every five years that outline additional developments and restrictions needed to ensure a safe supply of water. To assist with this, ICS Consulting (UK) is developing an optimisation framework for producing investment plans that accommodate uncertainty in supply and demand. As well as options for investing in infrastructure, water utilities can utilise temporary restrictions on water demand that carry a non-linear economic cost. In an integer programming context, this presents significant challenges, as the cost function now needs to be discretised and controlled through binary variables. We aimed to find novel ways to deal with the computational burden presented by the way restrictions are handled in the optimisation framework. Both existing methods and completely new options were closely investigated. At the completion of this project, we have contributed several new solution techniques and improvements. These additions were combined with all existing approaches and, through a testing framework, we were able to extract valuable information about: 1) probable causes of slow solution times; and 2) the best approach to apply for further development.

Key words: YPP, Water Resource Planning, Integer Programming

1 Introduction

The human population has been rapidly growing for the last 50 years. Utility companies around the world are under constant pressure to improve and expand their assets in order to keep up with this growing demand. In particular, many water utilities have adopted an approach based on Integrated Water Resource Management (IWRM) as proposed by the Global Water Partnership (GWP). IWRM is defined as *"a process which promotes the co-ordinated development and management of water, land and related resources, in order to maximise the resultant economic and social welfare in an equitable manner without compromising the sustainability of vital ecosystems"*. [GWP Technical Advisory Commitee, 2000]

Four main principles, known as the Dublin Principles, underpin IWRM:

- Fresh water is a finite and vulnerable resource, essential to sustain life, development and the environment.
- Water development and management should be based on a participatory approach, involving users, planners and policymakers at all levels.
- Women play a central part in the provision, management and safeguarding of water.
- Water has an economic value in all its competing uses and should be recognised as an economic good.

To develop long-term strategies that recognise impending challenges and adhere to the Dublin Principles, water companies around the world have been exploring various options with regards to the formulation of such plans. In particular, every 5 years, UK water utilities are obligated to provide a 25-year plan that aims to address the issue of growing demand [UK Environment Agency, 2017]. Thames Water in South-Eastern UK have undertaken the construction of a mathematical model that aims to find the optimum long-term investment strategy.

A stochastic sampling and optimisation framework for Water Resources Planning (WRP) has been developed by ICS Consulting (UK) and the Department of Engineering Science at the University of Auckland. However, the modelling approaches explored in this framework result in very slow solution times when solving for a large number of potential demand scenarios and corresponding restrictions (that are implemented if the plan does not provide enough supply for a particular scenario). This project aims to re-formulate particular parts of the existing optimisation model (a mixed integer programme) in order to reduce the time taken to solve the problem.

2 Background

2.1 Thames Water and ICS Consulting (UK)

Thames Water, a UK water supply company, have shared their Water Resource Management Plan (WRMP) data with ICS Consulting (UK) (ICS) to assist ICS in their development of a framework for generating good WRMPs that that cater to the peculiarities of their network. In turn, ICS have enlisted the Department of

Engineering Science (DES) at the University of Auckland (UoA) to assist with the modelling/optimisation part of the project.

To produce a WRMP, Thames Water utilises available data to forecast future supply and demand while incorporating uncertainty into these projections. A given system is then taken and tested against these estimates to highlight any supply deficits. Thames Water then attempts to bridge the gap between supply and demand by incorporating options or restrictions into its planning horizon. UoA DES has changed this approach to include uncertainty directly in the decision making for options and restrictions:

1. Use Monte Carlo sampling to generate scenarios of future supply, demand and the associated uncertainty;
2. Calculate supply deficit for each scenario;
3. If all deficits = 0 then STOP;
4. Find best options and restrictions for dealing with deficit across all scenarios.

While the overall process is very broad in scope (including statistical models of supply/demand and Monte Carlo sampling), this project focuses on step 4 – finding the best options and restrictions to deal with deficit across all scenarios.

2.2 Demand and Supply Options

At its core, the model revolves around balancing water supply and demand for a number of periods into the future. This can be done through the use of either supply and/or demand management options [Whitelock-Bell, 2016] such as:

- Leakage reduction;
- Water imports;
- Desalination plants.

In addition to the above, Thames Water could utilise emergency measures to reduce public water usage over a short period of time – these are known as restrictions. Unlike options, restrictions are temporary in their effect and thus require separate variables and constraints to model them properly. Additionally, restrictions are not linearly costed – restriction cost increases non-linearly as its duration and frequency of use increase.

Since non-linearity presents significant computational challenges, this property of restriction costs has been the focus of a large amount of research that has been undertaken at the University of Auckland. This previous work will be summarised next in Section 3.

3 Previous Model Developments

ICS Consulting (UK) (ICS) are not yet satisfied with their existing solutions for optimising water management for Thames Water. As such, several models have been developed at the University of Auckland in order to try and improve both solution quality and speed. An initial model has been developed, but struggled to solve the problem within an acceptable time frame. Subsequent work focused mainly on trying to achieve reasonable solution times that could be implemented in a commercially viable package to be used by ICS.

3.1 Base Model

This section is summary of work undertaken by Dr. Michael O'Sullivan and Assoc. Prof. Cameron Walker [O'Sullivan and Walker, 2016]. They developed a model that produces a least-cost WRMP that satisfies a variety of constraints including meeting level of service targets for restrictions with a given level of confidence. Please note that only the most relevant parts of the model have been presented in this paper.

3.2 Useful Abbreviations & Concepts

- Sample – a description of future demand and supply generated by Monte Carlo Sampling.
- Zone – A Water Resource Zone, a discrete planning unit with self-contained demand and supply.
- Restriction – a temporary measure to reduce water usage (see Section 2.2).
- LoS – Level of Service. For restrictions, this refers to how often it can be enforced in the planning period.

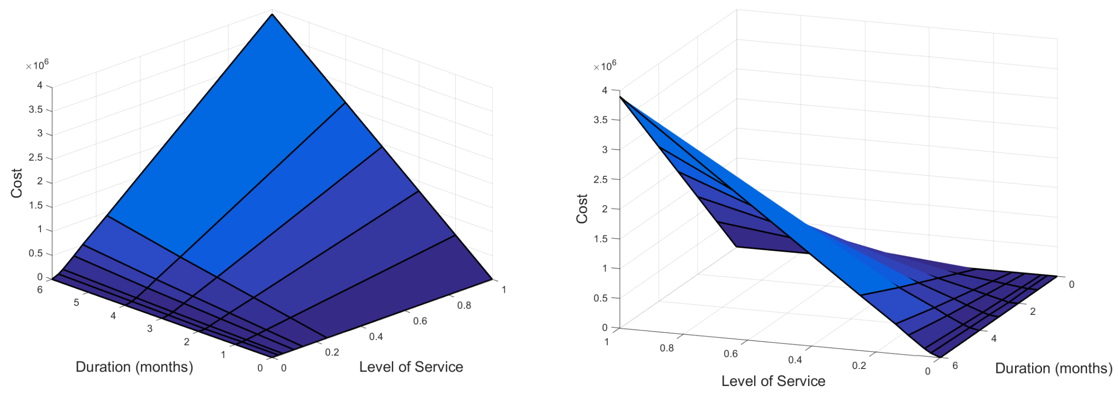
3.3 Definitions and Notations

Sets and Parameters		
$1, 2, \ldots, T$	=	the periods in the planning horizon
Z	=	the zones in the planning area
Ω	=	samples used in planning
J	=	the set of possible restrictions
Variables		
$e_{jz} \in \{0, 1\}$	=	restriction j is enforced in zone z;
$0 \leq p_{\omega jz\delta\lambda} \leq 1$	=	SOS convexity variable for restriction j in zone z under sample ω. ($\delta \in \Delta_j$ and $\lambda \in \Lambda_j$);
$q_{\omega jz\delta\lambda} \in \{0, 1\}$	=	SOS binary variable for restriction j in zone z under sample ω. ($\delta \in \Delta_j$ and $\lambda \in \Lambda_j$);

3.4 Non-Linearity

Through extensive economic research ICS consulting have determined that the costs resulting from restrictions are best modelled with a non-linear function that increases exponentially with both duration and level of service as shown in Figure 1. A conventional approach to tackling this issue is to discretise the non-linear function into a grid of points. From this grid we can construct piecewise approximations to the non-linear function by employing fractional weighting variables (denoted by $p_{jt\omega\delta\lambda}$) that make up a weighted sum of any square in the grid. Because these variables essentially create a convex combination of grid points, they are henceforth referred to as convexity variables.

We can observe that there is a certain concavity to the surface. If the model could choose any points across the grid, it could easily "cheat" and produce a point that lies below the actual surface.



(a) Restriction Cost: Front View

(b) Restriction Cost: Side View

Figure 1: Piecewise linear approximations of the aforementioned costing functions. Note the diagonal concavity when moving from LoS = 0, Duration = 6 to LoS = 1, Duration = 0.

3.5 SOS Formulation

One way of selecting a convex combination [Beale and Tomlin, 1970] is to simply select a square of 4 neighbouring points. A natural way to accomplish this in an Integer Programming context would be to create a grid of 0-1 variables that can be used to toggle the weighting variables on and off. This approach is called a Special Ordered Set of Order 2 (or SOS-2 for short) and was the initial approach used by O'Sullivan and Walker. These variables are denoted by $q_{\omega jz\delta\lambda}$.

To control these binary variables, we need both the standard master-slave constraints and a mechanism through which we force the variables to be used in a 2-by-2 square, in this case $\hat{n}$, the set of all "non-neighbours" of a point:

$$p_{j\omega\delta\lambda} \leq q_{jz\omega\delta\lambda}, \qquad j \in J, z \in Z, \omega \in \Omega, (\delta, \lambda), \lambda \in \Lambda_j, \delta \in \Delta_j \tag{1}$$

$$\sum_{\delta \in \Delta_j} \sum_{\lambda \in \Lambda_j} q_{jz\omega\delta\lambda} \leq 4e_{jz}, \qquad j \in J, z \in Z, \omega \in \Omega \tag{2}$$

$$\sum_{(\delta_{\hat{k}}, \lambda_{\hat{l}}) \in \hat{N}_{j\omega\delta_k\lambda_l}} q_{jz\omega\delta_{\hat{k}}\lambda_{\hat{l}}} \leq |\hat{N}_{j\omega\delta_k\lambda_l}|(1 - q_{jz\omega\delta_k,\lambda_l}), \qquad j \in J, z \in Z, \omega \in \Omega, \delta_k \in \Delta_j, \lambda_l \in \Lambda_j \tag{3}$$

3.6 Slice Formulation

In a previous part IV project in 2016 [Whitelock-Bell, 2016], Lucy Whitelock-Bell deduced that there are regions in each restriction's cost space in which the cost function is strictly convex. Thus, as long as we restrict out search to these regions, otherwise known as slices, any given convex combination of points will provide an accurate approximation of the actual cost.

When implemented, the slice formulation resulted in $n+m-2$ binary variables for a $n \times m$ restriction grid, being a substantial improvement over the SOS formulation in terms of the number of variables in the model. The core mathematical concept of the slice model is described as follows [Whitelock-Bell, 2016].

$\Theta_{\omega jz}$ = Set of slices spanning $c_R(j, \omega, z, \delta, \lambda)$;

$o_{\omega jz\theta}$ = Slice binary variable for slice $\theta \in \Theta_{\omega jz}$ for restriction j in zone z under sample ω

Intuitively, only one slice per restriction in each zone under each scenario may be turned on, if the restriction is being used in that zone:

$$\sum_{\theta \in \Theta_{\omega jz}} o_{\omega jz\theta} \leq e_{jz}, \quad j \in J,\ z \in Z,\ \omega \in \Omega \tag{4}$$

The SOS convexity variable at every point must be less that or equal to the sum of slices that overlap that point:

$$p_{\omega jz\delta\lambda} \leq \sum_{(\delta,\lambda) \in \theta} o_{\omega jz\theta}, \quad j \in J,\ z \in Z,\ \omega \in \Omega \tag{5}$$

3.7 Slice Echelon

Following Whitelock-Bell's 2016 part IV project, O'Sullivan and Walker have undertaken further work to speed up the model [O'Sullivan and Walker, 2016]. Among these contributions is "echelon form" – forcing the model to use binary variables in a certain pre-determined order, improving branching during the branch-and-bound tree. By introducing a specific order to the slices, we can reformulate the sets and constraints in the following way:

$o_{\omega jz\theta} = \theta^{th}$ slice binary variable for restriction j in zone z under sample ω ($\theta \in \Theta_{\omega jz}$)

To enforce the echelon form, we must first set each binary variable to be less than or equal to the one before it

$$o_{\omega jz\theta} \geq o_{\omega jz\theta+1}, \quad j \in J,\ z \in Z,\ \omega \in \Omega, \theta < |\Theta_{\omega jz}| \tag{6}$$

Then, we must adjust the master/slave constraints accordingly.

$$p_{\omega jz\delta\lambda} \leq \sum_{(\delta,\lambda) \in \theta} (o_{\omega jz\theta} - o_{\omega jz\theta+1}), \quad j \in J,\ t = 1, 2, ..., T,\ \omega \in \Omega, \theta < |\Theta_{\omega jz}| \tag{7}$$

$$p_{\omega jz\delta\lambda} \leq \sum_{(\delta,\lambda) \in \theta} o_{\omega jz\theta}, \quad j \in J,\ t = 1, 2, ..., T,\ \omega \in \Omega, \theta = |\Theta_{\omega jz}| \tag{8}$$

To most effectively control large subsets of slices at once, we can aim to always branch on the "most middle" variable, called bisection branching.

In addition to testing on the existing techniques presented in this section, our research provides some new methods aimed at improving the MIP performance. These methods will be presented next in Section 4 before all techniques/methods are compared in Section 5

4 New Methods

Although a large amount of work has aimed to improve the solution time of the model, none of the approaches have been able to best the SOS formulation [Whitelock-Bell, 2016]. As a result the optimisation model has been unable to achieve anything resembling reasonable solution times for the real-world case of 100 samples being examined. The main reason for the slow solution times is the interaction between the large number of scenarios and the complicated restriction costs, causing multiplicative growth as both increase in number.

When an initial examination of previous work was conducted and the results of the slice formulation analysed, it seemed that the difficulty of the restriction modelling lies not so much in the number of variables that arise, but rather in some underlying property of the problem that makes it more suited to specific solution methods. Thus the goal of this project became very broad – simply finding something that works better than the existing solution.

4.1 Battleship Formulation

The first approach I tried preserved the square-based selection process featured in the SOS formulation and improved the way in which the squares are selected. I realised that it is possible to introduce a 2D coordinate system for specifying which square should be selected - very similar to a game of battleships where each players needs to guess the coordinates of the correct points. We can select an allowable range for both level of service and the duration of a restriction - both constrained between two neighbouring values. While this reduces the number of variables, the downside of this 2D selection would potentially be apparent during the branch and bound process. If our 1-branch corresponds to a chosen 2×2 square, then two 1-branches are needed to narrow down a convex solution in the battleship formulation compared to one 1-branch required by the slice method.

The formulation was very similar to the slice formulation, except the "slice" variables and constraints needed to be implemented in two directions instead of one.

$\Theta^d_{\omega jz}$ = Set of slices spanning $c_R(j, \omega, z, \delta, \lambda)$ in the duration direction;
$o^d_{\omega jz\theta}$ = Slice duration binary variable for slice $\theta \in \Theta^d_{\omega jz}$ and restriction j in zone z under sample ω
$\Theta^l_{\omega jz}$ = Set of slices spanning $c_R(j, \omega, z, \delta, \lambda)$ in the LoS direction;
$o^l_{\omega jz\theta}$ = Slice LoS binary variable for slice $\theta \in \Theta^l_{\omega jz}$ and restriction j in zone z under sample ω

Again, only one restriction combination may be used:

$$\sum_{\theta \in \Theta^d_{\omega jz}} o^d_{\omega jz\theta} \leq e_{jz}, \quad j \in J,\ z \in Z,\ \omega \in \Omega \tag{9}$$

$$\sum_{\theta \in \Theta^l_{\omega jz}} o^l_{\omega jz\theta} \leq e_{jz}, \quad j \in J,\ z \in Z,\ \omega \in \Omega \tag{10}$$

The SOS convexity variable at every point must be less that or equal to the sum of slices overlapping that point for both duration and level of service slices.

$$p_{\omega jz\delta\lambda} \leq \sum_{(\delta,\lambda) \in \theta} o^d_{\omega jz\theta}, \quad j \in J, z \in Z,\ t = 1, 2, ..., T,\ \omega \in \Omega \tag{11}$$

$$p_{\omega jz\delta\lambda} \leq \sum_{(\delta,\lambda) \in \theta} o^l_{\omega jz\theta}, \quad j \in J, z \in Z,\ t = 1, 2, ..., T,\ \omega \in \Omega \tag{12}$$

4.2 Forcing Cut

While developing the new formulations, one concept that all methods share in common became apparent. If a solution is integer feasible, then a combination of weights producing a point close to the true cost is selected. However, the reverse does not hold true, i.e., if a correctly costed combination of weights is selected the solution may not be integer feasible, increasing the time taken to solve the problem. Thus I set out to come up with a solution to this phenomenon and try to force the model to become more naturally integer, i.e., remove fractionality from the solution space. I introduced a cut that "forces" the slice binary variables in any formulation to sum to 1 if the weights selected are on or above the surface, hence the term "Forcing Cut". To demonstrate this, let us return to the slice formulation, where this cut is most natural.

The sum of convexity variables in any given slice $\theta \in \Theta_{\omega jz}$ must be less than or equal to the sum of all the binary variables for slices that overlap with θ, including θ itself:

$$\sum_{(\delta,\lambda)\in\theta} p_{\omega jz\delta\lambda} \leq \sum_{\beta\in\Theta_{\omega jz}|\exists(\delta,\lambda)\in\beta\cap\theta} o_{j\omega\beta}, \quad j \in J,\ z \in Z,\ \omega \in \Omega,\ \theta \in \Theta_{\omega jz} \tag{13}$$

An equivalent constraint can be applied to battleship formulation, but needs to be modified slightly for echelon form.

4.3 Split Slices

Further pursuing the idea of closely tying cost-convex solutions to integrality, my next approach was to break up the convex variable tables into separate sets – one for each slice. This removes all overlap between slices by creating duplicate points for every convex variable that appears in more than one slice. While introducing more linear variables into the model, this approach means that the value of a slice binary variable can be bounded below by the sum of the slice's convexity variable, so convex solutions within a slice force integer slice binary variables. To implement this approach, new tables were created in the database and some constraints needed slight tweaking, namely (5) , (7) and (8).

Replacing (5) by (14) gives a constraint that acts as both a master/slave constraint and a cut that enforces a tight relationship between convexity and integrality.

$$\sum_{(\delta,\lambda)\in\theta} p_{\omega jz\delta\lambda} \leq o_{\omega jz\theta}, \quad j \in J,\ \omega \in \Omega \tag{14}$$

Similarly, echelon constraints (7) and (8) need to be modified in the following way:

$$\sum_{(\delta,\lambda)\in\theta} p_{\omega jz\delta\lambda} \leq o_{\omega jz\theta} - o_{\omega jz\theta+1}, \quad j \in J,\ \omega \in \Omega, \theta < |\Theta_{\omega jz}| \tag{15}$$

$$\sum_{(\delta,\lambda)\in\theta} p_{\omega jz\delta\lambda} \leq o_{\omega jz\theta}, \quad j \in J,\ \omega \in \Omega, \theta = |\Theta_{\omega jz}| \tag{16}$$

The next chapter aims to compare the different approaches outlined in this section and Chapter 3 of the report.

5 Comparing the Formulations

Throughout the development of this model, numerous ways of formulating the cost functions have been developed. However, the effectiveness of these methods relative to one another has not been fully explored in detail. Thus it has been difficult to make any sort of meaningful conclusions or to guide the development of the model in a dedicated way.

The formulations to compare are as follows, with a variety of potential improvements available for each formulation:

- Original SOS formulation;
- Slice formulation;
- Square formulation;
- Battleship formulation;
- Split formulation.

The standard way of measuring the performance of a mixed integer program is to look at the solution times. However, the problems presented here are too large to be solved using the computational resources available in a reasonable amount of time. As such, to measure the relative quality of each formulation, we run all the problems for four hours and measure the optimality gap.

All available solution methods were first tested on a reduced problem of 20 scenarios. This allowed us to isolate good formulations that can then be evaluated on a range problem sizes – 5, 10, 15, 20, 25. All runs use 15 options, 5 restrictions and 3 zones.

Through this testing framework we have selected the following formulation for extended testing:

- F1 – Slice echelon bisection, the baseline existing formulation;
- F2 – Slice echelon with bisection and forcing cuts;
- F3 – Slice formulation with the forcing cut;
- F4 – Battleship echelon formulation with bisection and forcing cut;
- F5 – Split formulation with slice echelon and bisection branching.

The results of evaluating these formulations are presented in Figure 2.

To investigate the importance of the source data, i.e., which samples are used, was for the 10 and 15 scenario performance, we can repeat these tests with problems constructed from alternative datasets. Initially, scenarios from the 20 sample problem were used to construct the data for the 15 scenario problem and, similarly, the 10 sample problem was created from the 25 sample problem. For the next set of results, the 20 scenario problem was used to create the 10 sample problem and the 25 scenario problem was truncated to solve the 15 scenario problem. The differences between the two sets of results are highlighted in Table 1. In this table we compare the performance of formulations relative to one another by ranking them for each set of samples. In addition, the difference in the optimality gap between each set of samples at the end of the solve is presented to highlight the impact on individual formulations.

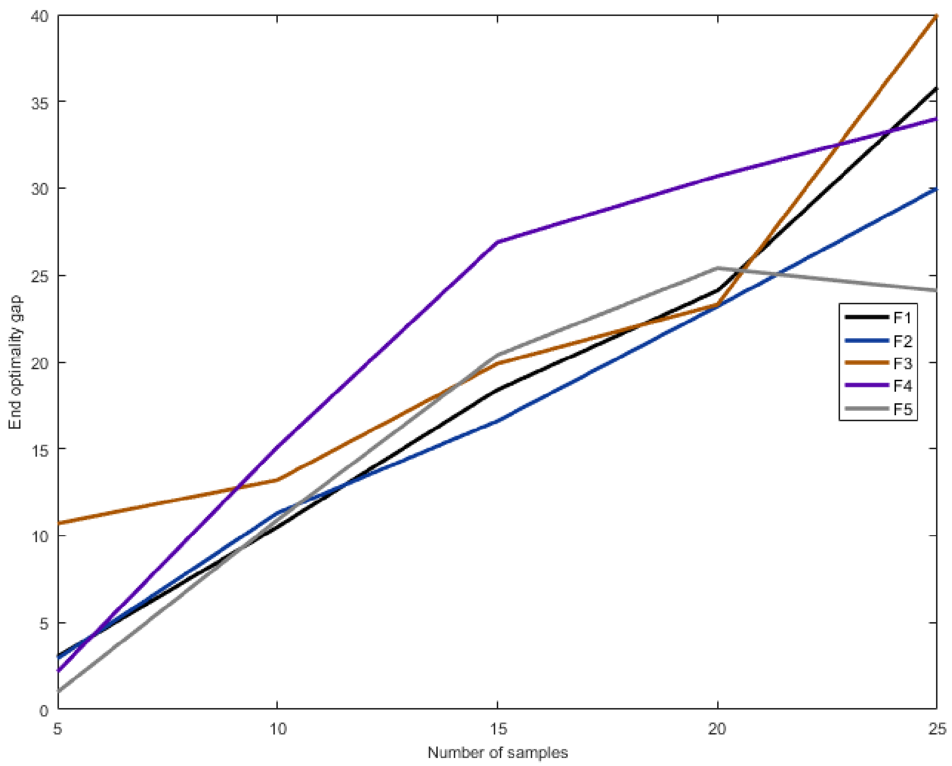


Figure 2: Initial testing results

Formulation	Rank				Gap change (%)	
	10 Scen.		15 Scen.		10 Scen.	15 Scen.
	Set 1	Set 2	Set 1	Set 2		
F1	1	3	2	2	+1.2	+0
F2	3	2	1	5	-0.6	+10.8
F3	4	4	3	3	+0.8	+2.9
F4	5	5	5	4	+1.9	-0.1
F5	2	1	4	1	-1.3	-2.7

Table 1: Ranked results comparing effects of individual scenarios

The next section aims to obtain meaningful conclusions from the data presented in this paper.

6 Discussions

6.1 Final Results

Before any comments are made regarding formulation quality, a point regarding the development environment needs to be made. The virtual machine used for this project had limited computational power. In addition to this, the cloud resources allocated to the machine are not constant – the VM may receive additional processing capability if the total demand on the server is fairly low. The impact of the virtual environment is that direct comparisons between two solves is not conclusive, as they may not be utilising the same resources.

The first aspect of the results in Figure 2 we can comment on is the consistency of each formulation. When considering the effect of new developments on the solution quality, it is clear that the forcing cut is almost always beneficial to the solution times, no matter the formulation. F5, the split formulation, also seems to perform relatively well under certain conditions, but more investigative work needs to be undertaken to understand the efficacy of this approach.

When designing the experiments to test the formulations, one broad assumption was made: that the individual scenarios have no effect on the solution quality; the performance of a formulation is dictated exclusively by the problem size. We can see from Table 1 that this is clearly not the case. When the specific scenarios were changed, some formulations improved while others deteriorated although the size of the problem remained constant. Because individual scenarios affect the performance, single runs of smaller problem sizes are not sufficient to fully compare different methods.

6.2 Future Work

Before final conclusions can be made about the formulations, more testing needs to be done. A standard computer with additional memory and much higher processing power needs to be used so that it may be possible to actually run the solves to completion. This will provide us with a much better metric that can be used to compare formulations. To fully narrow down the best way of solving the problem, different scenario samples of the same size would need to be run in order to get a clear of picture just for one problem size. In perfect conditions, a good test would be repeating this process across three to four problem sizes and removing the time limit to record the final solution time.

It would seem at this point that any further improvements in the restriction space of the model would need significant reformulation of the restriction tables. As such, it would be most likely better to shift development attention to other parts of the model. There is potential work that can be done across scenarios, improving the way restriction point variables are handled as the number of scenarios increases. Additionally, decomposition approaches across scenarios can also be utilised in an attempt to speed up solution times.

7 Conclusion

The system developed by ICS Consulting (UK) in conjunction with the University of Auckland was, and still is, suffering from slow solution times, impeding development. At the outset of this project, the goal was somewhat broad – finding methods to improve solution times by changing the way restriction cost modelling worked.

Some of the newly developed techniques provide a positive impact on the solution times. The forcing cut is a good addition to the repertoire of techniques available when solving this problem; it is applicable to a range of methods and usually improves the optimality gap. The split formulation also carries some promise, but warrants further testing under different conditions.

After extensive testing was conducted, it became clear that problem size was not the only factor affecting formulation performance. Individual scenarios played a big role in how fast a particular method was able to reduce the optimality gap, which is a crucial piece of knowledge for further advancement of the model. Using this, we are able to effectively guide future development by constructing a much more effective testing framework.

Acknowledgements

First and foremost, I would like to thank my supervisors Dr. Michael O'Sullivan and Dr. Cameron Walker for their unyielding support during this project. They have provided me with both necessary guidance and the freedom to take the project in my own direction, making this project a pleasure to work on throughout the year.

In addition, big thanks to ICS Consulting for initiating this project and granting me an opportunity to work on a very applied project that builds my practical skills. In particular, I would like to acknowledge Joshua Dawes for his great help in getting the database up and running and Dr. Tim Watson for his feedback at the end of the project.

References

Beale, E. M. L. and Tomlin, J. A. (1970). Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. *Proceedings of the Fifth International Conference on Operational Research*, pages 447–454.

GWP Technical Advisory Commitee (2000). Integrated resource water management. `http://www.gwp.org/globalassets/global/toolbox/publications/background-papers/04-integrated-water-resources-management-2000-english.pdf`. Accessed: 31/08/2017.

O'Sullivan, M. and Walker, C. (2016). Water resources planning project progress report 3.

UK Environment Agency (2017). Water resources planning: how water companies ensure a secure supply of water for homes and businesses. `https://www.gov.uk/government/publications/water-resources-planning-managing-supply-and-demand/water-resources-planning-how-water-companies-ensure-a-secure-supply-of-water-for-homes-and-businesses`. Accessed: 31/08/2017.

Whitelock-Bell, L. (2016). Improving solution speed for water resources planning models.