Multi-Objective Optimisation in Decision Making for Infrastructure Asset Management

Lin Chen
Department of Civil and Environmental Engineering
University of Auckland
New Zealand
lche971@aucklanduni.ac.nz

Andrea Raith
Department of Engineering Science
University of Auckland
New Zealand
a.raith@auckland.ac.nz

Theunis F.P. Henning, Asaad Shamseldin
Department of Civil and Environmental Engineering
University of Auckland
New Zealand
t.henning@auckland.ac.nz, a.shamseldin@auckland.ac.nz

Abstract

Infrastructure asset management (IAM) is important for modern society. Decision making, as a critical part of IAM, helps in achieving the goals of IAM by determining management strategies. Decision making faces many challenges; therefore, multi-objective optimisation methods are applied to identify efficient solutions in order to assist decision making. Efficient solutions contains supported and non-supported solutions, which can be used as the basis of decision making. The identification of supported solutions is easier; while the identification of non-supported solutions is more challenging.

This paper analyses an IAM decision making problem and identifies efficient solutions to assist decision making using a two-phase optimisation approach. More specifically, this paper (1) introduces IAM decision making; (2) formulates practical decision making problems; (3) uses a two-phase optimisation approach that identifies supported solutions in a first phase and non-supported solutions in a second phase by different methods; and (4) discusses the result.

The two-phase optimisation approach is tested with a bi-objective IAM decision making problem. According to the test, practical decision making problems can be solved; and both supported and non-supported solutions are identified. When identifying non-supported solutions, methods perform differently and their application depends on the addressed problem.

Key words: Multi-Objective Optimisation, Decision Making, Infrastructure Asset Management.

1 Introduction

Infrastructure asset management (IAM) attempts to generate, identify and implement appropriate maintenance management strategies for infrastructure (such as road, bridge, etc.) in order to achieve specific goals (NAMS 2011). Decision making is a critical part of IAM, which determines the management strategies for infrastructure and helps in pursuing the goals of IAM.

In this paper, the decision making of IAM aims at identifying the best combinations of management strategies for a road network as an example.

Management strategies describe the maintenance and operation treatments that will be undertaken to a segment of infrastructure during a certain period. For example, a five-year strategy for a road segment contains the designed treatments for the next five years, such as crack seal at the first year and overlay at the fourth year. The treatments have different influence on infrastructure; hence the strategies lead to different results of infrastructure condition, expense, level of service, etc. These results are estimated and used as the basis of strategy selection. For example, there are several strategies associated with specific cost and condition as shown in Figure 1. Comparing strategy A (only overlay at the fourth year) and strategy B (crack seal at the first year and overlay at the fourth year), strategy A costs less money, but strategy B leads to better road condition. If the objective of IAM is the least cost, the strategy A is preferred; while if the objective of IAM is the best road condition under specific budget, strategy B is preferred.

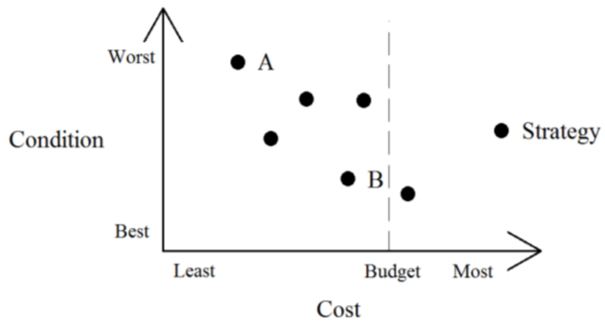

Figure 1. Typical decision making of IAM (Henning, et al. 2013).

Traditionally, decision making is based on the financial benefit or the benefit-cost ratio. Strategies with greater financial benefit or bigger benefit-cost ratio are preferred. Some researchers consider more than one objective in IAM; where objectives are aggregated into one overall objective using a weighted sum approach (Chassiakos, et al. 2005; Fwa and Chan 1993).

However, these decision making methods may not be sufficient for modern decision making. Firstly, more objectives and constraints should be considered, such as financial aspect, infrastructure condition, level of service, etc., which may be conflicting and cannot always be expressed in monetary equivalent, so that they cannot be aggregated into a single objective function in a meaningful way. Secondly, network decision making often generates more benefits, but may bring a vast amount of alternative strategies to be considered. Thirdly, decision making leads to better decision when decision makers are able to fully explore the available trade-offs. Therefore, a method that can clarify the decision making problem and indicate the trade-offs is necessary.

Therefore, multi-objective optimisation (MOO) is applied to assist decision making of IAM.

2 Multi-Objective Optimisation (MOO) in Decision Making

MOO solves optimisation problems with more than one objective and constraints (Hillier and Lieberman 2005). MOO can describe multi-objective decision making more practically and rational especially when objectives cannot be reasonably aggregated into a single objective. Some MOO algorithms can handle many decision variables (strategies) so that a great number of strategies can be analysed in acceptable time.

MOO methods identify efficient solutions. Efficient solutions are those solutions that cannot be improved in one objective without worsening another (Hillier and Lieberman 2005). The solution pool contains all efficient solutions; hence it always contains the best combination of strategies no matter what the decision maker's preference is. A decision maker only needs to select one solution from the solution pool without considering other non-efficient possibilities. This largely eases decision making. Then conventional decision aiding methods can be applied at this point to find one most preferred solution (Belton and Stewart 2002). In addition, efficient solutions show the objective values that can be achieved and the relationship of objectives, which helps in understanding and clarifying the decision making problem.

Efficient solutions can be divided into two groups: supported and non-supported solutions (Ehrgott 2005). Supported solutions are those efficient solutions of multi-objective problems, which are the optimal solutions of a weighted sum single-objective problem. They are located on the boundary of the convex hull of feasible objective vectors. Non-supported solutions are located in the interior of this convex hull as shown in Figure 2.

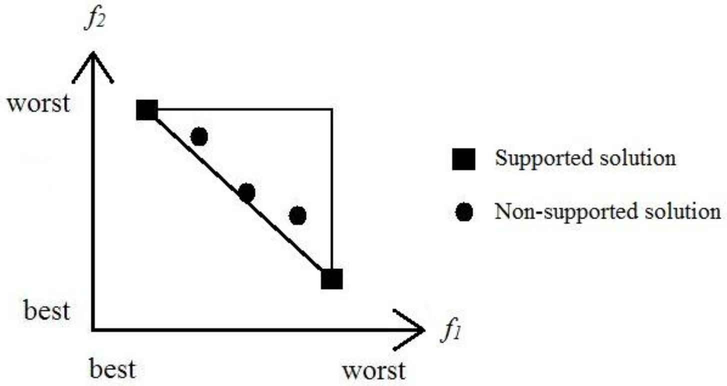

Figure 2. Supported and non-supported solutions.

For decision making, both supported and non-supported solutions are needed. Non-supported solutions can fill the gap between supported solutions so that considering them can enhance the quality and distribution of identified efficient solutions. Moreover, both supported and non-supported solutions are efficient; and any one of them can be the best choice under certain circumstance. Therefore, both supported and non-supported solutions are needed to provide an unbiased and comprehensive solution pool to decision making.

It is relatively easy to obtain supported solutions, while finding non-supported solutions can be more challenging. In this paper, a two-phase optimisation approach is used to identify both supported and non-supported solutions.

3 Formulation

To apply MOO, a decision making problem should be mathematically formulated. For example, a small road network of a Canadian city is divided into m segments. For segment j, all feasible strategies $\mathcal{S}_j$ are generated. The total number of strategies for the entire network is n. The decision making problem is transferred into finding m strategies from n alternatives, one for each segment. Therefore, integer programming is selected. Decision variables $\boldsymbol{x} = (x_1, x_2 \cdots, x_n)$ are binary, representing the selection of strategies. When strategies i is selected, x_i equals 1; otherwise, x_i equals 0. For each strategy, the corresponding criteria are estimated, such as if strategy i is applied, the cost and the condition of this road segment are estimated. Objectives and constraints are selected from the available criteria and formulated as linear functions, such as objectives of maximum present value (PV) benefits (Equation (1)) and minimum PV cost (Equation (2)), and the constraints of annual acceptable network condition (Equation (3)). An extra constraint (Equation (4)) is added to ensure only one strategy is selected for each segment.

$$\max\ f_1(\boldsymbol{x}) = \sum_{i=1}^{n} B_i * x_i \tag{1}$$

$$\min\ f_2(\boldsymbol{x}) = \sum_{i=1}^{n} C_i * x_i \tag{2}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} cond_{iy} * x_i \leq Condition \quad \text{for } y = 1,2,\cdots Y \tag{3}$$

$$\sum_{i \in \mathcal{S}_j} x_i = 1 \qquad \text{for } j = 1,2,\dots J \tag{4}$$

where B_i = PV benefit after strategy i is applied

C_i = PV cost after strategy i is applied

$cond_{iy}$ = condition in year y if strategy i is applied

$Condition$ = acceptable network condition

Y = the number of analysis years

The general formulation of this problem is shown in Equations (5)-(7), where the objectives can be formulated to be maximised or minimised.

$$\max/\min \quad f_k(\boldsymbol{x}) \qquad \text{for } k = 1,2,\cdots,K \tag{5}$$

$$\text{s.t.} \quad g_l(\boldsymbol{x}) \leq \text{Limit}_l \quad \text{for } l = 1,2,\cdots,L \tag{6}$$

$$\sum_{i \in \mathcal{S}_j} x_i = 1 \quad \text{for } j = 1,2,\dots m \tag{7}$$

where $f_k(\boldsymbol{x})$ = objective function k

$g_l(\boldsymbol{x})$ = constraint function l

Limit_l = acceptable value of constraint l

K = the number of objectives

L = the number of constraints

4 Two-Phase Optimisation Method

Most of the previous applications of MOO in decision making of IAM are still based on heuristic algorithms (Flintsch and Chen 2004; Fwa, et al. 2000; Hilber, et al. 2007). Heuristic algorithms try to identify solutions as close as possible to efficient solutions, but the obtained solutions may not be real efficient solutions, especially when many strategies are involved. Therefore, a MOO method that can solve decision making problems and identify efficient solutions is necessary.

This paper uses a two-phase optimisation method that identifies both supported solutions and non-supported solutions for bi-objective decision making of IAM.

4.1 First phase

The first phase attempts to identify all supported solutions. Many algorithms are available for this phase. In this paper, a dichotomic approach is applied. The dichotomic approach is an exact optimisation method for bi-objective optimisation problems, which transfers MOO into several single-objective optimisation problems using aggregate weights. This algorithm guarantees to identify all supported solutions in objective space.

The algorithm of the dichotomic approach is shown in Algorithm 1 (Przybylski, et al. 2010). At the beginning, the extreme solutions are identified by optimising each objective individually. Then every two solutions ($\boldsymbol{x}^s$ and $\boldsymbol{x}^{s+1}$) that are consecutive in objective space are used to establish a single-objective optimisation problem (Equation (10) and (11)) with aggregate weights w_1 and w_2 defined as Equation (8) and (9).

$$\mathrm{w}_1 = f_2(\boldsymbol{x}^s) - f_2(\boldsymbol{x}^{s+1}) \tag{8}$$

$$w_2 = f_1(\boldsymbol{x}^{s+1}) - f_1(\boldsymbol{x}^s) \tag{9}$$

$$\max w_1 f_1(\boldsymbol{x}) + w_2 f_2\,(\boldsymbol{x}) \tag{10}$$

$$\text{s.t.} \quad \boldsymbol{x} \in \boldsymbol{\mathcal{X}} \tag{11}$$

where $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ = two objective function with decision variable *x*

$\boldsymbol{\mathcal{X}}$ = the set of feasible solutions given by Equation (6) and (7)

s =solution index

Algorithm 1: Dichotomic Approach

Calculate solution pool $\boldsymbol{\mathcal{S}}$

$\boldsymbol{\mathcal{S}} = \emptyset$	# $\boldsymbol{\mathcal{S}}$ solution pool
while $k \le K$	# K number of objectives ($K = 2$)
$\boldsymbol{x} = \text{Solve}(f_k, \boldsymbol{g})$	# solve the lexicographic optimisation problem with
$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{S}} \cup \boldsymbol{x}$	main objective f_k and constraints $\boldsymbol{g}$

end while
while any new solution is identified
    for each consecutive solutions $\boldsymbol{x}^s, \boldsymbol{x}^{s+1} \in \boldsymbol{\mathcal{S}}$
        calculate $\boldsymbol{w}$ using Equation (8) and (9)
        solve Equation (10) and (11), obtain $\boldsymbol{x}$
        if $\boldsymbol{x}$ is feasible:
            $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{S}} \cup \boldsymbol{x}$
        end if
    end for
end while

Then this single-objective optimisation problem (an integer programming) is solved by a standard single-objective optimisation method. If a new feasible solution is identified, this solution is added into the supported solution pool and will be used to find new solutions in later iterations. The algorithm stops when no more new solutions are identified.

4.2 Second phase

The second phase attempts to identify non-supported solutions. This paper separately applies a Genetic Algorithm (GA) and a dichotomic based method to identify non-supported solutions.

4.2.1 Genetic Algorithm (GA)

GA is a widely applied heuristic algorithm, which is based on the principle of natural selection and the recombination of genes (Konak, et al. 2006; Man, et al. 1996; Morcous and Lounis 2005).

GA "exhibit[s] impressive efficiency in many experimental studies" (Grefenstette 1986). It is inherently parallel so that a group of independent solutions are simultaneously improved. According to the authors' previous experience, comparing with other heuristic algorithms, GA identifies good solutions for IAM decision making. Moreover, the implementation of GA is easy and the number of parameters is small.

In this paper, a modified Nondominated Sorting Genetic Algorithm II (NSGA II) for multi-objective optimisation is applied. NSGA II tries to identify efficient solutions by evaluating and sorting the solutions (Kalyanmoy Deb et al. 2002). After the first phase, as discussed in Section 4.1, all supported solutions are identified and input into NSGA II as initial parents (solutions). A group of new solutions (children) are generated by crossover and mutation of every pair of parents, and added into the solution pool. The non-dominated solutions in the solution pool are marked as non-dominated level 1 and removed from the pool; and then the non-dominated solutions of the remaining solutions in the solution pool are marked as non-dominated level 2 and removed from the pool; and so on until the solution pool is empty. The solutions with higher non-dominated level (smaller) are selected as parents to generate new solutions. The process repeats until stopping criterion is satisfied.

4.2.2 Dichotomic based method

The dichotomic based method is established based on dichotomic approach. Firstly all supported solutions are identified. For every two supported solutions $\boldsymbol{x}^s$ and $\boldsymbol{x}^{s+1}$ that are consecutive in objective space, the dichotomic approach is applied again with extra constraints, (Equation (13) and (14)). These constraints force the algorithm to focus on new solutions between the two identified ones. The formulation of the dichotomic based method is shown in Equation (12)-(15). The algorithm stops when all supported solutions are used to identify non-supported ones. All the identified new solutions are non-supported solutions.

$$\max w_1 f_1(\boldsymbol{x}) + w_2 f_2(\boldsymbol{x}) \tag{12}$$

$$\text{s.t.} \quad f_1(\boldsymbol{x}) \le \max\left(f_1(\boldsymbol{x}^s), f_1(\boldsymbol{x}^{s+1})\right) - \varepsilon \tag{13}$$

$$f_2(\boldsymbol{x}) \le \max\left(f_2(\boldsymbol{x}^s), f_2(\boldsymbol{x}^{s+1})\right) - \varepsilon \tag{14}$$

$$\boldsymbol{x} \in \boldsymbol{\mathcal{X}} \tag{15}$$

where ε = a very small constant

5 Case study

A small road network of a Canadian city is analysed, as mentioned in Section 3. Four cases are established based on this road network, as shown in Table 1. All feasible ten-year management strategies for every segment with corresponding criteria are generated by the software tool dTIMS CT 8 (Deighton Associates Limited 2008). The total number of strategies is shown in Table 1. The formulation is shown in Equation (1)-(4).

Table 1. Addressed decision making problem

Case	Number of Segments	Number of strategies
1	5	132
2	10	237
3	20	639
4	50	1823

The two-phase optimisation approach, including all the algorithms, is implemented on a computer using Python. Single-objective optimisation problems are solved by Gurobi, which solves integer problems based on branch and bound.

Figure 3 is an example of solutions for Case 1. The supported solutions are too few to show the entire efficient solutions range. GA and dichotomic based method can identify non-supported solutions that fill the gap between supported solutions, so that the efficient solutions are more consecutive.

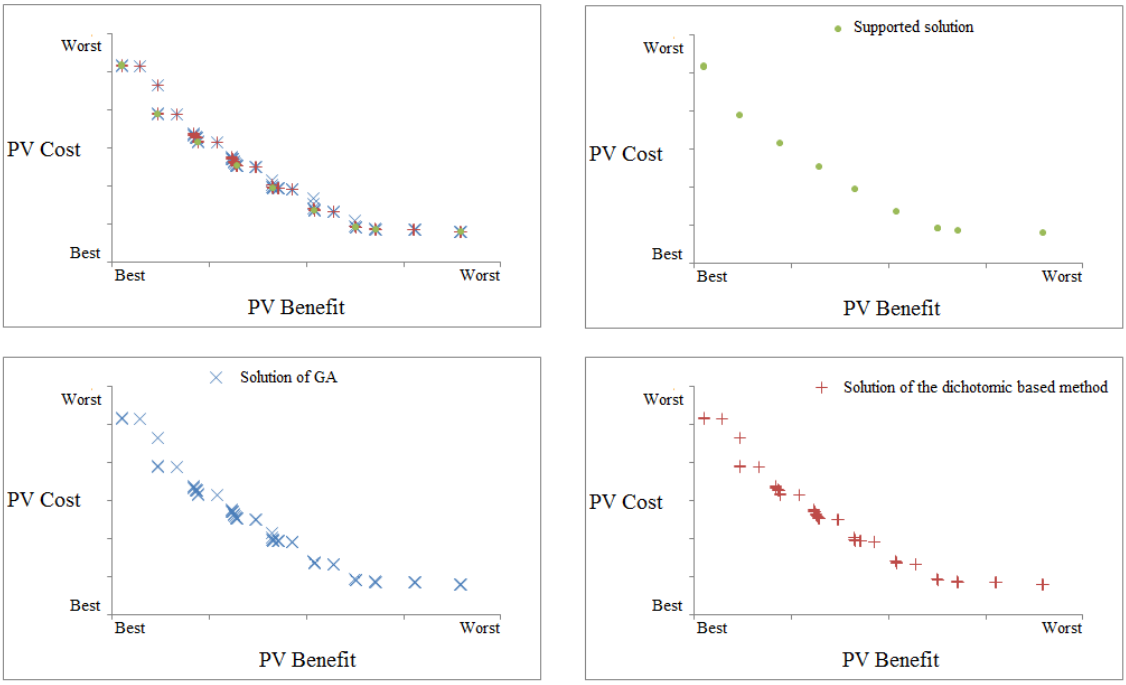

Figure 3. Identified solutions for Case 1.

The number of identified efficient solutions is shown in Table 2. The first phase identified all supported solutions and their number is shown in Column (2). Column (3) and (6) show the number of efficient solutions identified by GA and dichotomic based method separately after the second phase. GA identifies more non-supported solutions when the number of strategies is small; while dichotomic based method identifies more when the number of strategies grows bigger.

Table 2. Number of identified solutions

Case	Phase 1	Phase 2			
		GA[1,2]			Dichotomic based method
		Solutions	Objective Values	Mutation rate	
(1)	(2)	(3)	(4)	(5)	(6)
1	11	66	63	0.20	63
2	22	346	268	0.05	272
3	50	989	701	0.01	1128
4	126	1519	948	0.01	8175

1 All the programmes run three times and the numbers shown in the table is the average value

2 Different mutation rates are applied and the number shown in the table is the one with maximum number of efficient solutions.

For the addressed decision making problem, some different solutions have the same objective values, therefore, they are the same point in objective space. GA can identify these same-objectives solutions. Excluding the same-objectives solutions, the number of the solutions with different objective values identified by GA is shown in Column (4), which is smaller or equals the one of the dichotomic based method. The difference in obtained solutions between GA and the dichotomic based method increases with the growth of the number of strategies (decision variables). However, the dichotomic based method finds new solutions based on objective values; therefore, it cannot identify different solutions with same objective values.

GA has two important parameters: mutation rate and stopping criteria. Mutation rate determines the randomness of new solutions. In this paper, eight mutation rates are tested ranging from 0.01 to 0.3. The mutation rate generating the most efficient solutions is shown in Column (5). When the number of decision variables is small, the mutation rate should be bigger; while when the number of decision variables is big, the mutation rate should be reduced. Stopping criteria are the termination trigger, which not only determines the computation time as stated later, but also the solution quality. Generally speaking, more iterations result in better solutions. In this paper, stopping criteria are that the identified solutions remain the same for 20 iterations and the maximum number of iterations is 300. If more iterations are calculated, the identified solutions are probably better, especially for big problems.

The computation time is shown in Table 3. This table demonstrates that finding supported solutions is relatively fast, whereas most computation time is spent on finding non-supported solutions. In the first phase, the computation time per efficient solution increases with the number of strategies, so that the total computation time is also increasing. In the second phase, the computation time depends on the applied method. The total computation time of GA depends on the stopping criteria; therefore, the computation time per efficient solution varies. The computation time per solution of the dichotomic based method increases with the growth of the number of strategies. Additionally, the total computation time is also related with the number of non-supported solutions. Comparing the computation time of Phase 2, the dichotomic based method is faster than GA.

Table 3. Computation time[1]

Section	Phase 1		Phase 2			
			GA[2]		Dichotomic based method	
	Total (s)	per solution (s)	Total (s)	per solution (s)	Total (s)	per solution (s)
1	0.11	0.010	119.37	1.809	0.74	0.012
2	0.43	0.012	2814.10	8.133	7.73	0.028
3	1.95	0.039	7498.28	7.581	103.02	0.091
4	11.09	0.088	13412.22	8.830	4447.60	0.544

1 The computation time is only for optimisation, excluding input and output data.
2 The time is the average computation time to identify the non-supported solutions in Table 2

6 Conclusion

This paper introduces decision making; and then uses a two-phase optimisation method to solve multi-objective decision making problems that arise in IAM and identify efficient solutions.

As shown in the case study, both supported and non-supported solutions are identified. The dichotomic approach can identify all supported solutions for bi-objective decision making problems in the first phase. Then a GA and a dichotomic based method are applied to identify non-supported solutions in the second phase. However, their performance is different.

GA can identify different non-supported solutions with identical objective values. Its algorithm is flexible and can be controlled by defining stopping criteria. However, the solutions identified by GA are not consistently good. The solution quality reduces when the number of strategies is growing. The programme should be run several times to get a more reliable result.

The dichotomic based method can identify all non-supported solutions with different objective values. For the addressed problem, the dichotomic based method is faster than GA. However, when a large number of non-supported solutions exist, its computation time is very long.

The selection of a proper algorithm for the second phase depends on the addressed problems. If the problem is very small, GA may identify the solutions with same and different objective values; therefore, its solutions are more comprehensive. Even though the computation time of GA may be longer, it is still acceptable. If the addressed problem is big, a dichotomic based method identifies better solutions. However, at least one termination trigger is recommended in case too many non-supported solutions exist. If more solutions are needed for a big decision making problem, GA can be applied to identify same-objective solutions after the dichotomic based method.

However, the applied two-phase optimisation method is only for bi-objective decision making. When more objectives are analysed, another exact optimisation method that can handle three or more objectives is necessary.

Acknowledgement

This research is supported by Deighton Associate Ltd., including the software dTIMS CT 8, database, financial support, and so forth. The authors would also like to thank China Scholarship Council for the support by a doctoral scholarship.

7 References

Belton, V. and T.J. Stewart. 2002. *Multiple Criteria Decision Analysis: An Integrated Approach*: Springer.

Chassiakos, A.P., P. Vagiotas, and D.D. Theodorakopoulos. 2005. "A knowledge-based system for maintenance planning of highway concrete bridges." *Advances in Engineering Software*, 36(11–12):740-749.

Deighton Associates Limited. 2008. "dTIMS CT reference guide." Deighton Associates Limited. Canada.

Ehrgott, M. 2005. *Multicriteria Optimization*, 2^{nd} edition, Springer.

Flintsch, G.W., and C. Chen. 2004. "Soft computing applications in infrastructure management." *Journal of Infrastructure Systems*, 10(4):157-166.

Fwa, T.F., and W.T. Chan. 1993. "Priority rating of highway maintenance needs by neural networks." *Journal of Transportation Engineering*, 119(3):419-432.

Fwa, T.F., W.T. Chan, and K.Z. Hoque. 2000. "Multiobjective optimization for pavement maintenance programming." *Journal of Transportation Engineering*, 126(5):367-374.

Henning, T.F.P., D.C. Roux, and G. Hart. 2013. "Will we get what we wanted? – project level decision making versus strategic investment analysis." In *NZTA/NZIHT 14th Annual Conference*.

Hilber, P., V. Miranda, M.A. Matos, and L. Bertling. 2007. "Multiobjective optimization applied to maintenance policy for electrical networks." *IEEE Transactions on Power Systems*, 22(4):1675-1682.

Hillier, F.S. , and G.J. Lieberman. 2005. *Introduction to Operations Research*. 8^{th} edition, Elizabeth A. Jones, New York.

Kalyanmoy D., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation*, 6(2):182-197.

Konak, A., D.W. Coit, and A.E. Smith. 2006. "Multi-objective optimization using genetic algorithms: A tutorial." *Reliability Engineering & System Safety*, 91(9):992-1007.

Man, K.F., K.S. Tang, and S. Kwong. 1996. "Genetic algorithms: concepts and applications." *IEEE Transactions on Industrial Electronics*, 43(5):519-534.

Morcous, G., and Z. Lounis. 2005. "Maintenance optimization of infrastructure networks using genetic algorithms." *Automation in Construction*, 14(1):129-142.

NAMS. 2011. *International infrastructure management manual*. National Asset Management Support Group (NAMS Limited). Wellington.

Przybylski, A., X. Gandibleux, and M. Ehrgott. 2010. "A Recursive Algorithm for Finding All Nondominated Extreme Points in the Outcome Set of a Multiobjective Integer Programme." *Journal on Computing*, 22(3):371-386.