# Prioritising and Scheduling Road Projects by Genetic Algorithm

Min Qiu, Department of Information Management and Marketing
The University of Western Australia, Nedlands, Western Australia 6009
e-mail: mqiu@ecel.uwa.edu.au

**Abstract** One problem facing a road authority is to prioritise and schedule road projects for future years, fitting the identified projects into a timetable. Since the combination space for a project timetable is extremely large, an efficient search procedure is needed. This problem has three parts: applying selection criteria, applying constraints (budget, linkages between projects, etc.), and using an approach to find the optimum solution. The benefit-cost ratio in benefit-cost analysis or the weighted score in multicriteria analysis has been used to rank road projects. However, this method cannot find an optimum solution, nor can it cope with constraints imposed by linkages between projects. Goal programming (GP) can represent both multiple criteria and constraints which link projects, but the GP model is based on the assumption that project effects are divisible, so that a partially finished project will contribute proportionally to the benefits. In practice, this assumption is valid for upgrading projects (Type 1 projects) but not for new road links (Type 2 projects). This paper discusses a genetic algorithm (GA) to deal effectively with Type 1 and Type 2 projects. A string of integers is used to represent project priorities in a GA individual. Budget constraints are imposed in the process to map the project priority matrix onto a road project timetable. Constraints imposed by linkages between projects are applied by using a penalty function. In the case of all Type 1 projects, the GA finds nearly the same project timetable as obtained by GP. When projects are differentiated into Type 1 and Type 2 projects, the GA finds a project timetable that is different from that in the case of all Type 1 projects.

## 1. INTRODUCTION

One problem facing a road authority is to prioritise a set of identified road projects, and then schedule these projects for a period of years into the future. In this process, the road authority normally tries to achieve maximum investment effectiveness, subject to its budget in that schedule period and other conditions. Practically, the way of dealing with this problem varies among road authorities. For example, some road authorities prioritise and schedule road projects at two sequential decision-making stages, but the others directly schedule road projects without prioritising them (TRB [1978]). Theoretically, the final result of this process is a constrained timetable for the identified road projects, no matter how a road authority does this job.

A project timetable is a combination of projects for a schedule period. The number potential combinations can be determined as follows. If p denotes the number of identified road projects, then for any one year in a schedule period, the number of possible project combinations is $2^p$. If p is equal to 35, the number of possible project combinations for one year is $3.435973837\times10^{10}$. A schedule period covers more than one year, so that the number of possible timetables is more than $3.435973837\times10^{10}$.

Because the number of possible timetables is extremely large, an efficient method for searching for the optimum timetable is needed.

This paper first gives an overview of the constituents of the problem, then discusses a genetic algorithm (GA) approach to prioritise and schedule road projects, and finally presents the results of applying the GA approach.

## 2. AN OVERVIEW

This problem has three parts: criteria for prioritising and scheduling road projects, constraints on prioritising and scheduling road projects, and an approach to searching for the optimum solution.

There can be one or more criteria used for prioritising and scheduling road projects. The selection of criteria depends on actual applications. Traditionally, road authorities have used the benefit-cost ratio as the only criterion, but recently road authorities have selected more and more criteria.

There are a variety of constraints imposed on prioritising and scheduling road projects. The budget is the most common constraint, restricting the maximum expenditure within a whole schedule period or by individual years in the schedule period (Humphrey [1981]). Operationally, there are constraints that specify linkages between some projects (for example, one project cannot be completed before another project.). These constraints are called operational constraints.

Two techniques have been used to model prioritising and scheduling road projects: ranking and optimisation (Humphrey [1981] and Taplin et al. [1995]).

A ranking approach ranks or prioritises identified road projects in descending order by a measure, such as the benefit-cost ratio in Cost-benefit Analysis (CBA) or the score in Multicriteria Analysis (MCA). Then, the ranking approach successively includes the road projects into a project timetable until the budget in the schedule period is exhausted. One drawback of this technique is that it cannot cope with operational constraints. Furthermore, the project timetable obtained by this technique is not necessarily the optimum project timetable.

Linear programming (LP) with one or more objectives has been used for scheduling road projects (Melinyshyn et al [1973] and Taplin et al. [1994]). The LP technique directly produces the optimum timetable for a set of identified road projects. In linear programming, it is easy to impose budget constraints, and some but not all operational constraints.

The LP technique is superior to ranking in that LP can come up with the optimum solution, and is more powerful in coping with some operational constraints.

The following is an example of applying linear programming with multiple objectives, that is, goal programming (GP), to schedule road projects. In this example, 35 road projects are to be scheduled for 10 years, 17 criteria are identified and their measures for each project are predetermined. Budgets for 10 years are projected and it is required that the 10th project be not completed before the 11th project. The aspiration value for each criterion (goal) is also predetermined.

Minimise:

$$\sum_{i=1}^{17} a_i y_i \qquad (1)$$

Subject to:

$$\sum_{k=1}^{10}\sum_{j=1}^{35} \frac{1}{(1+r)^{(k-1)}} b_{ij} x_{jk} + y_i = g_i \qquad (i=1...17) \qquad (2)$$

$$\sum_{j=1}^{35} c_j x_{jk} \le m_k \qquad (k=1...10) \qquad (3)$$

$$\sum_{k=1}^{n} x_{10k} \le \sum_{k=1}^{n} x_{11k} \qquad (n=1...10) \qquad (4)$$

$$\sum_{k=1}^{10} x_{jk} \le 1 \qquad (j=1...35) \qquad (5)$$

$$x_{jk} \ge 0 \qquad (j=1...35,\ k=1...10) \qquad (6)$$

Where:

$a_i$ = the weight applied to the ith goal
$y_i$ = the amount deviated from the ith goal
$g_i$ = the aspiration value of the ith goal
$x_{jk}$ = the proportion of project j constructed in year k
r = discount rate
$b_{ij}$ = the contribution to goal i made by project j
$c_j$ = the cost of project j
$m_k$ = the budget available for year k

The underlying construct is the time value of capital. This technique is based on the assumption that project effects are divisible, so that a partially finished project will contribute proportionally to the benefits, as indicated by (2).

Because the effects of some criteria used for scheduling road projects are traffic dependent, the above assumption is valid for upgrading road links (Type 1 projects) but not for new road links (Type 2 projects). An upgrading road link project is implemented on its existing alignment, and when it is being implemented, the link is partially open to traffic. Once a part of the project is finished, it can immediately be open to traffic and have a proportional effect. On the other hand, in most cases a new road link will not serve traffic until it is finished, that is, before its completion it makes no contribution to the network.

Taking into account the difference between Type 1 and Type 2 projects in the divisibility of project effects, it is more accurate to represent (2) as the following.

$$\sum_{j=1}^{35} G_{ij} + y_i = g_i \qquad (i=1...17) \qquad (7)$$

$$G_{ij} = \begin{cases} \sum_{k=1}^{10} \frac{1}{(1+r)^{(k-1)}} b_{ij} x_{jk} & \text{(for Type 1 projects)} \\ \frac{1}{(1+r)^{(f_j-1)}} b_{ij} & \text{(for Type 2 projects)} \end{cases} \qquad (8)$$

Where:

$f_j$ = the year in which project j is finished
$G_{ij}$ = the discounted contribution to ith goal made by project j

Imposing constraints is the only way in linear programming to apply different conditions and requirements. It is hard to express the conditions in (8) by constraints. Therefore, it is difficult for linear programming to cope with the situation in which road projects are differentiated into Type 1 and Type 2 projects, and another approach is needed.

The remaining parts of this paper discuss the application of a genetic algorithm (GA) to deal effectively with this situation, and present the results of some experiments using the GA.

## 3. APPLICATION OF A GENETIC ALGORITHM

Genetic algorithms (GAs) are a method developed recently. For searching in a linear system, GAs are not as efficient as LP, but are more flexible than LP in that GAs can cope with a variety of conditions and requirements, such as the conditions in (8) above. This flexibility of GAs brings the modelling of some complex systems closer to the actual situations.

### 3.1 Genetic Algorithm Steps

An individual in a GA represents one solution of a problem. An individual is a string of either binary digits (0 or 1), or real numbers, depending on actual applications. A group of individuals forms a generation, which is a subset of the solution space, and the number of individuals in each generation is the population size. A GA finds the optimal solution by simulating a process in which GA individuals evolve from generation to generation.

Step 1: The process starts with randomly initialising the individuals in the first generation, that is, assigning a value to each digit or number for every individual. The values assigned are in the domain range for each digit or number.

Step 2: This step decodes and evaluates the information or the solution contained in each individual. The evaluated value of an individual is called the fitness of the individual, and the bigger (for a maximisation problem) or the smaller (for a minimisation problem) the value, the fitter the individual.

Step 3: This step applies a reproduction scheme to individuals in the existing generation (parents) to reproduce the next generation (offspring). There are many reproduction schemes, such as roulette wheel selection, tournament selection, "steady state" selection and so on (Goldberg et. al. [1991]). Generally, the fitter an individual in a parent generation, the more likely that the individual gets one or more copies in the next offspring generation. The population size of an offspring generation equals the population size of the previous parent generation.

Step 4: After reproduction, various genetic operators are applied to the individuals in an offspring generation. Crossover and mutation operators are two the most common operators. The function of a crossover operator is to partly swap information between a pair of individuals to generate a new pair of individuals. The pair of individuals are chosen randomly, and the iterations of applying a crossover operator depend on a predetermined probability and the population size. Mutation operator mutates the value of a digit or a number in an individual, to produce a new individual. The mutated individual and the position of the digit or the number are chosen randomly. The iterations for mutation depend on a predetermined probability, the population size, and the number of digits or numbers in an individual.

Step 2 to Step 4 are repeated until a closure condition is satisfied. Normally the closure condition is a given number of generations.

There are several methods to impose constraints in GAs. The first one is to penalise the fitness of an infeasible individual that violates the constraints. The second one is to fix infeasible individuals according to the constraints. The third one is to throw away infeasible individuals and then generate feasible individuals to make up the population size. The last one is to always generate feasible individuals by a specially designed process or genetic operators, such as crossover and mutation (Michalewicz [1992]). The first three methods are applied between applying genetic operators and evaluating individuals. The last one can be in any step, depending on actual applications.

If a GA is properly tuned, the fittest individual will converge on the optimum or a near optimum solution. The power of a GA in search lies in its ability to transfer useful information from generation to generation.

### 3.2 Representing the Solution of the Problem

One issue of applying GAs to prioritising and scheduling road projects is to represent the problem by a GA individual. This issue can be addressed by analysing the decision-making process of a road authority.

A road authority first assigns priorities on identified road projects, and then fits these projects into a budget plan, to produce a road project timetable. This process can be expressed by (9).

$$\begin{pmatrix} P_1 & P_2 & \cdots & P_u & \cdots & P_p \end{pmatrix} \xrightarrow[\text{constraint}]{\text{Budget}} \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1w} & \cdots & x_{1y} \\ x_{21} & x_{22} & \cdots & x_{2w} & \cdots & x_{2y} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{q1} & x_{q2} & \cdots & x_{qw} & \cdots & x_{qy} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{p1} & x_{p2} & \cdots & x_{pw} & \cdots & x_{py} \end{pmatrix} \quad (9)$$

project priority project timetable

Where:

$P_u$ = the code number of a project of which the ranking number or priority number is u
$p$ = the number of projects
$x_{qw}$ = the proportion of project q (i.e. the code number of the project is q) built in year w
$y$ = the number of schedule years

Naturally, a GA individual can use a string of integers to represent the matrix of project priority in (9). For each individual, the value of an integer is a project code number, and the position of the integer in the individual is the priority number of the project. For example, a GA individual for 10 road projects:

(3 5 1 6 4 10 8 9 7 2).

In this GA individual, project 3 has the highest priority number, 1, and project 2 has the lowest priority number, 10.

### 3.3 Applying Constraints

In this application, the budget constraint is applied by a process that maps a matrix of project priorities or a GA individual onto a project timetable. Like the simple ranking method, this process sequentially selects projects into a timetable until all the budget is used up, and then drops the remaining projects (that is, in the matrix of project timetable in (9) all the elements for the dropped projects are zero). Although the dropped projects are not included into the timetable, they still remain in the GA individual, so that all the information on the GA individual can be transferred to the next generation of GA individuals. Besides, for a selected project, say project j, the year ($s_j$) in which the project is started, and the year in which the project is finished ($f_j$), are determined in the mapping process. Therefore, in this context, equation (8) can be modified as equation (10).

$$G_{ij} = \begin{cases} \sum_{k=s_j}^{f_j} \frac{1}{(1+r)^{(k-1)}} b_{ij} x_{jk} & \text{(for Type 1 projects)} \\ \frac{1}{(1+r)^{(f_j-1)}} b_{ij} & \text{(for Type 2 projects)} \end{cases} \tag{10}$$

The way to impose an operational constraint depends on the definition of the constraint. This paper just discusses one type of operational constraint: one project (say project j1) cannot be completed before another project (say project j2). In the GA, this constraint can be expressed by

$$f_{j2} - f_{j1} \geq 0. \tag{11}$$

Inequality (11) is different from inequality (4) in two aspects. One aspect is that to construct the constraint inequality (11) uses the years in which projects in question are finished, rather than the yearly construction proportions of these projects. Another aspect is that inequality (11) is just one inequality, while inequality (4) is the general form of a group of inequalities.

To impose such an operational constraint (say constraint c) a penalty function $\Phi_c$ can be used in a way:

$$\Phi_c = \begin{cases} (f_{j2} - f_{j1})^2 & \text{(when } f_{j2} < f_{j1}) \\ 0 & \text{(when } f_{j2} \geq f_{j1}) \end{cases} \tag{12}$$

Therefore, after operational constraints having been applied, the objective function (1) becomes

$$\sum_{i=1}^{17} a_i y_i + \gamma \cdot \sum_{c=1}^{m} \Phi_c \tag{13}$$

Where:

$\gamma$ = predetermined penalty coefficient
m = number of operational constraints

In general, the process to calculate the fitness function (that is, objective function) for a GA individual in this application can be summarised as the following:

- Applying the budget constraint to the GA individual, so as to get a project timetable, as well as the information about starting and completing years for each project selected;
- Calculating the discounted contribution to each goal made by every selected project in the timetable, as indicated by equation (10);
- Imposing operational constraints and calculating penalty functions, as in equation (12); and
- Finally calculating fitness function, as indicated by (13).

## 4. EXPERIMENTS

The date set for the example in section 2 was used for experiments. The basic data are described as the following.

- 17 criteria were identified for prioritising and scheduling road projects, and the value for each goal's aspiration was predetermined;
- There were 35 road projects to be prioritised and scheduled, and the contribution to each goal made by every project was predetermined;
- The scheduling period is 10 years, the budget available in each year being predetermined; and
- Other data and parameters, such as the discount rate.

The reproduction scheme and genetic operators used in the experiments are combined pairwise tournament selection and preserving elite scheme, partial mapped crossover, mutation, and reversion. The detail descriptions of these reproduction scheme and genetic operators are shown by Goldberg [1989].

### 4.1 Experiment 1

In this experiment using the GA, all road projects were assumed to be Type 1 projects. In this case, the problem can be solved by both GP and a GA.

If a problem can be solved by linear programming, there is no need to solve it by a GA. The point of applying the GA to this situation is to test the GA's ability to find the optimum solution. For this purpose, the optimum solution obtained by GP was used as a benchmark for the GA's solutions.

Figure 1 shows the convergence of objective functions for a series of GA runs with the same data set. In Figure 1, each thick line represents the objective function for the best GA individual with generation number for each run. The value of objective function for the optimal solution obtained by GP is shown in Figure 1 by a thin horizontal dash line. It can be seen that from about generation 50 the objective functions of the best GA individuals converge on a value, which is less than 0.2 per cent above the GP optimum.

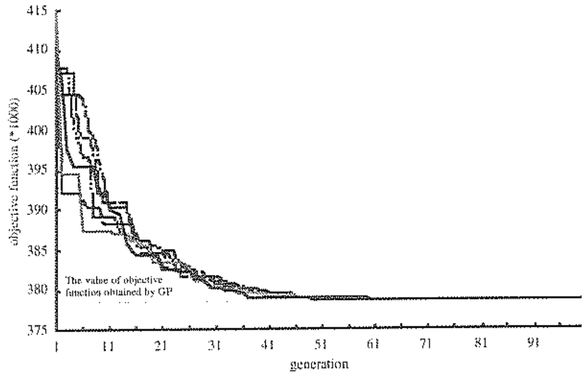


**Figure 1**: Variation of Objective Function - Experiment 1

### 4.2 Experiment 2

In experiment 2 using the GA, road projects were differentiated into Type 1 and Type 2 projects. Type 2 projects do not confer any benefits until completed. In this case, the problem can only be solved by a GA. The information on project type for some projects is in the second column of Table 1.

Figure 2 shows the convergences of objective functions for nine runs with the same data set. The objective functions in two runs converged on 378837, the lowest value in the nine runs.

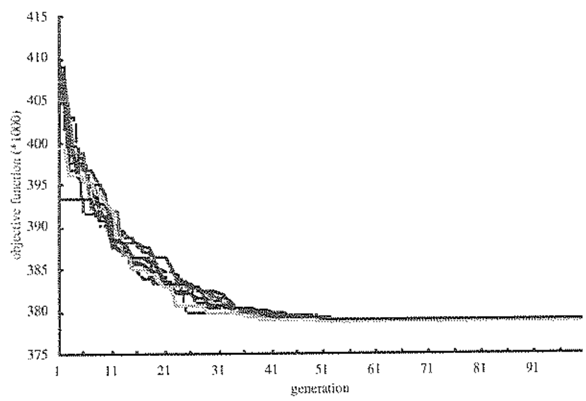


**Figure 2**: Variation of Objective Function - Experiment 2

### 4.3 Result Comparisons

Table 1 shows the investment allocations for the projects that were selected into at least one of the project timetables obtained by GP, and by GA experiment 1 and experiment 2. The investment allocation in a year for each project is equal to the product of the project's cost multiplied by the project's proportion built in that year.

Theoretically, there should not be any difference between the solutions by GP and experiment 1 using the GA. The differences (the first and second rows for projects 4, 29 and 32) are because the GA has not found the optimum solution so far, as it is normally claimed that a GA can find a near optimum solution for a problem. On the other hand, the differences are in only 3 of the 35 projects or 0.62 per cent of the total budget for the 10 years. This suggests that the GA can be applied to the problem with an acceptable accuracy.

As shown in Table 1, different assumptions on project types in experiment 1 and experiment 2 do make a difference (the second and third rows for project 2, 4, 5, 8, 9, 13, 21, and 23) in project timetables. The basic phenomenon in experiment 2 is that the implementation of a Type 2 project (except project 2 in year 10) is fitted into one year, whereas in experiment 1 some Type 2 projects (projects 5 and 13) are split into two continued years.

**Table 1**: Investment Allocations Obtained by GP, and by GA Experiment 1 (GA 1) and Experiment 2 (GA 2)[a]

| No. | Type[b] | Exp. | Investment Allocation in Schedule Years | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 1 | GP | | | | | | | | | | 4300 |
| | | GA 1 | | | | | | | | | | 4300 |
| | | GA 2 | | | | | | | | | | 4300 |
| 2 | 2 | GP | | | | | | | | | | |
| | | GA 1 | | | | | | | | | | |
| | | GA 2 | | | | | | | | | | 158 |
| 3 | 1 | GP | 400 | | | | | | | | | |
| | | GA 1 | 400 | | | | | | | | | |
| | | GA 2 | 400 | | | | | | | | | |
| 4 | 1 | GP | | | | | | | | | | |
| | | GA 1 | | | | | | | | | | 159 |
| | | GA 2 | | | | | | | | | | |
| 5 | 2 | GP | | 6147 | 1553 | | | | | | | |
| | | GA 1 | | 6147 | 1553 | | | | | | | |
| | | GA 2 | | | 7700 | | | | | | | |
| 6 | 1 | GP | | | | | | | | 1140 | 4260 | |
| | | GA 1 | | | | | | | | 1139 | 4261 | |
| | | GA 2 | | | | | | | | 1139 | 4261 | |
| 8 | 2 | GP | | 2311 | | | | | | | | |
| | | GA 1 | | 2311 | | | | | | | | |
| | | GA 2 | 2311 | | | | | | | | | |
| 9 | 1 | GP | | | 7830 | | | | | | | |
| | | GA 1 | | | 7830 | | | | | | | |
| | | GA 2 | | 5247 | 2583 | | | | | | | |
| 11 | 1 | GP | | | | | | | 6087 | 13 | | |
| | | GA 1 | | | | | | | 6087 | 13 | | |
| | | GA 2 | | | | | | | 6087 | 13 | | |
| 12 | 2 | GP | | 6600 | | | | | | | | |
| | | GA 1 | | 6600 | | | | | | | | |
| | | GA 2 | | 6600 | | | | | | | | |
| 13 | 2 | GP | 2904 | 2396 | | | | | | | | |
| | | GA 1 | 2904 | 2396 | | | | | | | | |
| | | GA 2 | | 5300 | | | | | | | | |
| 14 | 2 | GP | | | | | | | 3400 | | | |
| | | GA 1 | | | | | | | 3400 | | | |
| | | GA 2 | | | | | | | 3400 | | | |
| 15 | 1 | GP | 15650 | | | | | | | | | |
| | | GA 1 | 15650 | | | | | | | | | |
| | | GA 2 | 15650 | | | | | | | | | |
| 16 | 1 | GP | | | | | | | | 12000 | | |
| | | GA 1 | | | | | | | | 12000 | | |
| | | GA 2 | | | | | | | | 12000 | | |
| 17 | 1 | GP | | | | | | | | | 14694 | 11706 |
| | | GA 1 | | | | | | | | | 14694 | 11706 |
| | | GA 2 | | | | | | | | | 14694 | 11706 |
| 19 | 1 | GP | | | 8671 | 18954 | 18954 | 18954 | 9467 | | | |
| | | GA 1 | | | 8670 | 18953 | 18953 | 18953 | 9465 | | | |
| | | GA 2 | | | 8670 | 18953 | 18953 | 18953 | 9465 | | | |
| 21 | 2 | GP | | | 900 | | | | | | | |
| | | GA 1 | | | 900 | | | | | | | |
| | | GA 2 | | 900 | | | | | | | | |
| 23 | 1 | GP | | 1500 | | | | | | | | |
| | | GA 1 | | 1500 | | | | | | | | |
| | | GA 2 | 593 | 907 | | | | | | | | |
| 25 | 1 | GP | | | | | | | | 5801 | | |
| | | GA 1 | | | | | | | | 5801 | | |
| | | GA 2 | | | | | | | | 5801 | | |
| 29 | 2 | GP | | | | | | | | | | |
| | | GA 1 | | | | | | | | | | 1010 |
| | | GA 2 | | | | | | | | | | 1010 |
| 31 | 2 | GP | | | | | | | | | | 780 |
| | | GA 1 | | | | | | | | | | 780 |
| | | GA 2 | | | | | | | | | | 780 |
| 32 | 2 | GP | | | | | | | | | | 1169 |
| | | GA 1 | | | | | | | | | | |
| | | GA 2 | | | | | | | | | | |
| 35 | 1 | GP | | | | | | | | | | 999 |
| | | GA 1 | | | | | | | | | | 999 |
| | | GA 2 | | | | | | | | | | 999 |

[a] The projects, which were not selected into any one of the project timetables obtained by GP, GA 1 and GA 2, are not listed in the table

[b] In GP and GA 1, all projects were assumed to be Type 1, the project types listed below are only applied in GA 2.

## 5. CONCLUSIONS AND DISCUSSION

This paper demonstrated that the GA can be used for prioritising and scheduling road projects, especially in the case where some projects do not confer any benefits until completed.

The GA was used to show that a string of integers is appropriate for representing a project priority matrix in a GA individual, and that budget and operational constraints can be imposed in the GA by mapping a project priority matrix onto a project timetable, and using a penalty function, respectively.

Using the optimum solution obtained by GP as a benchmark, experiment 1 using the GA showed that the GA can find a near optimum solution at an acceptable computing cost. For the results reported in this paper, the cost is 6 runs of the GA, and each run requires about 1 hour. The higher accuracy of solutions obtained by the GA can be achieved by more runs of the GA.

Experiment 2 using the GA showed that when the difference between Type 1 and Type 2 projects in the project effects is considered, the Type 2 projects selected into a project timetable are fitted into one year rather than split into more than one year, because of their delayed benefits. The project timetable obtained in experiment 2 can be easily modelled by using a GA, but with difficulty by linear programming.

All experiments reported in this paper share a common assumption. The assumption is that the effects of each road project are only dependent on the project itself, ignoring whether the other projects will be implemented or not. This assumption is reflected in the fact that the contribution to each goal made by every project is predetermined and fixed. Therefore, a further improvement in prioritising and scheduling road projects can be made by considering the dependencies between identified road projects in terms of their effects.

Another improvement can be made by differentiating the effects of a project into two or more types. For example, the effects of a project can divided into: the effect that just exists during the period of implementing the project, such as disturbance to existing traffic by constructing the project; the effect that is dependent on traffic and can only be effective after the project is completed, such as savings in travel time and vehicle operating costs; and the effect that lasts for ever from starting implementation of the project, such as the damage to natural environment.

These possible improvements in prioritising and scheduling road projects can only be realised by using genetic algorithms together with road network modelling methods.

## ACKNOWLEDGMENTS

The author thanks Professor John Taplin for his valuable advice and supervision. The author also thanks Associate Professor Jim Everet for his encouragement in writing this paper, Ms Viviam Salim for her help in English writing, and Main Roads WA for supplying valuable data. Any error in the paper is the responsibility of the author.

The author holds a PhD scholarship from the University of Western Australia.

## REFERENCE

Goldberg, D. E., Genetic algorithms in searching, operation and machine learning, 412 pp, Addison-Wesley, 1989.

Goldberg, D. E., K. Deb, A comparative analysis of selection schemes used in genetic algorithms, *Foundations of Genetic Algorithms*, 69-93, G. J. E. Rawlins, ed., Morgan Kaufmann, 1991.

Humphrey, T. F., Evaluation criteria and priority setting for state highway programs, *National Cooperative Highway Research Program Synthesis of Highway Practice, No. 84*, 32 pp., Washington D.C., 1981.

Melinyshyn, W., R. Crowther and J. D. O'Doherty, Transportation planning improvement priorities: development of a methodology, *Highway Research Record, No. 458*, 1-12, Washington D. C., 1973.

Michalewicz, Z., Genetic algorithms + data structures = evolutionary programs, pp 252, Springer-Verlag, 1992.

Taplin, J. H. E., M. Qiu and Z. Zhang, Policy-sensitive selection and phasing of road investments with a goal program, unpublished paper, Dept. of Information Management & Marketing, The University of Western Australia, 1994.

Taplin, J. H. E., M. Qiu and Z. Zhang, Allocation of public funds by goal programming on multiple criteria, *Australian Journal of Public Administration, 54(1)*, 58-64, March 1995.

Transportation Research Broad (TRB), Priority programming and project selection, *National Cooperative Highway Research Program Synthesis of Highway Practice, No. 48*, 31 pp., Washington, D.C., 1978.