21st Australasian Transport Research Forum
Adelaide, September 1997

Optimising the Sequence of Urban Road Improvement Projects by Genetic Algorithm

Vivian Salim
Department of Information Management and Marketing
University of Western Australia

Abstract:

This paper demonstrates the application of a Genetic Algorithm, which is a randomised search method based on the concept of evolution, to be problem of searching for an optimal timetable of road project construction. The technique is particularly useful in long term planning of an urban transportation network. Our objective is to provide a schedule that allocates yearly road construction budgets in a way that maximises economic benefits over the planning period (for example, over fifteen years). Since the late 1950's, many studies in the optimisation arena have been conducted on variations of this problem. A brief survey of these studies and a comparison to the Genetic Algorithm will be presented. A Genetic Algorithm is applied to the problem of road project scheduling using the output from traffic assignments on a subset of the Perth, Western Australia road network. The results of the optimisation process and its implications and value to policy-making is discussed

Contact Author:

Vivian Salim
Department of Information Management and Marketing
University of Western Australia
Nedlands WA 6009

telephone: (09) 380 3195 fax: (09) 380 1004
email: vsalim@ecel.uwa.edu.au

Introduction

The aim of this paper is to present a new approach to solving the practical problem of optimising expenditure on projects in an urban road system. Given an origin-destination flow pattern, the capacity of existing roads is adjusted, or new links are added in order to maximize net benefits. There are many variations to this problem. For instance, the objective may be to minimize travel time given a fixed budget for road investments, or conversely, to minimize the capital invested for a given level of service. The optimisation problem at hand has also been referred to as the network design problem, or NDP (Magnanti and Wong, 1984).

In this paper, the set of potential projects to improve the transport network are given. This differs from the more conventional NDP problem, which includes all the links in the network as potential candidates for improvement. The aim is to schedule this set of projects over a working period of y years, then evaluate the impacts of these projects over its lifetime of a further Y years. In the schedule or timetable of projects, given a budget constraint, some projects will not be scheduled for construction at all. However, it will be illustrated later that it is possible to extend the potential set of links to be improved to include the entire network.

In the literature, it is evident that an overiding difficulty in analysing this problem is that projects are highly interdependent in an urban road network. The sequence of implementing projects is important. It has also been shown that the problem being analysed has non-polynomial time complexity (NP-complete) so that currently existing solution methods will take an exponentially increasing amount of time to solve the problem as the problem size increases.

The rest of the paper is structured as follows. First a brief survey of previous work is presented. Then it is suggested that a genetic algorithm is suitable for solving this problem and a particular GA is proposed. This GA is applied to a portion of the Perth road system. Finally, some conclusions are made and further research directions are discussed.

A Brief Survey of the Literature

Among the numerous studies on the NDP, a few examine the effects of investment alternatives over time. Long range planning in road projects is essential as roads are not only expensive, but have a long life span. One method used is to combine linear and dynamic programming (Bergendahl, 1969). In that paper, Bergendahl formulates the optimal operation of the road network as a multicommodity flow linear programming problem. The overall optimal solution, including the size and time of investment is then obtained through dynamic programming (Bellman, 1957). In this method, however, the demand between every pair of nodes is assumed to be known, which is not the case in a metropolitan area. In addition, the set of all possible routes at time t is evaluated using linear programming so that, for every possible road network (or *state*), the minimum cost is obtained. Bergendahl evaluates every state in order to capture the interdependence between different road projects, especially in urban areas. That is, if the capacity of a road is improved, the traffic on this road will increase as drivers adjust their travel habits. These changes then cause the traffic patterns on surrounding roads to change, thereby influencing the decision on which investment should be next selected so that the network is optimised. However, in urban areas, the number of OD pairs, possible routes and projects are typically large. So, although this method attempts to capture interdependence of alternative investments in urban areas, because every possible road state is to be enumerated, it would be inefficient to use Bergendahl's method for a realistically sized network.

Frey and Nemhauser (1972) modelled a different sort of feedback between service characteristics and flow over time. The optimal timing of network expansion is cast as a convex programming problem, where flows are functions of travel time, which in turn is a function of flows. It is assumed that the uni-modal flows will decrease as travel time increases, and this loss of trips may be interpreted as a loss of revenue to another mode, or as a social cost. An origin-destination (OD) pair of nodes is defined to be the potential demand on a particular path. That is, it is the path demand when unlimited capacity is assumed on all the links in the network. With this definition then, every OD pair is constrained to only one path. The authors point out that this makes the model suitable for application to a rural setting, or a railway system.

Rothengatter (1979) defines interdependencies between projects as "vertical" or "horizontal". Interactions within the same period are referred to as horizontal, while interactions across periods are vertical. In the evaluation of benefits, instead of discounting the costs, Rothengatter ranks the contribution of each time period in accordance to its importance with respect to the staging decision. Interestingly, the last period in the planning horizon is considered the most important, as the total project bundle is given at that stage. Then, next in importance is the first period under consideration. A "correct" ranking must therefore be decided upon prior to starting the analysis. The author rightly points out that this is normally implicit in the discount rate. However, the model imposes the constraint that there are no vertical interdependencies. This means that a project in time t may not borrow funds from time $t+1$ as this will impose an additional constraint in time $t+1$. In addition, traffic flows in time t does not influence traffic flows or distribution in time $t+1$. The solution approach uses a modified version of the branch-and-bound integer programming procedure by Boyce, Fahri and Weischedel (1973). Since the last period is considered the most important, the backwards dynamic programming method used first computes which projects will ultimately be implemented in the network at the end of the planning horizon, given the aggregate budget for the entire period. This set of projects constitutes the reduced set of candidate links to be considered in the next most important period by branch and bound. The process starts from time period T and repeats for every period upto period T-1.

The advantage of using genetic algorithms

Some of the modelling assumptions mentioned above can be discarded by using genetic algorithms (GAs) because of the flexibility GAs have in coding problems. In the model, vertical interdependence in the budget constraint may be accommodated. Because the GA process essentially carries out simulations of what happens when a set of projects is implemented in a given year, the concern that improvements on specific roads will change the traffic pattern will also be modelled.

All the exact methods will be prohibitive to use as the transportation network increases in size, because the discrete transportation NDP is NP-complete[1] (Xiong and Schneider, 1995). A classic example of an NP-complete network optimisation problem is the travelling salesman problem (or TSP, where a salesman has to visit *n* cities exactly once and return to his starting point). According to Magnanti and Wong, the NDP is in fact a generalised network optimisation problem, which includes the TSP as a specific problem type. The TSP has a complete OD structure, where the demand between every pair of nodes in a network is known, while the NDP has an arbitrary OD structure. Furthermore, the objective function in the TSP is linear with respect to flow variables, and has no complicated side constraints. The NDP requires the constraint that the minimum cost route choice is to be used for each commodity in the network. Hence the NDP is more difficult then the already notorious TSP.

However, the very fact that the NDP is NP-complete makes the problem suitable for solution by genetic algorithm. Not only can genetic algorithms handle computationally intensive problems, they can also handle non-convexity in the objective function (Goldberg, 1989). Heuristics decision rules may be used which do not consider the interdependence of flows and improvements. However, because of the change in flow when a project is added to a network, an addition of a link in the network may increase total travel time (Steenbrink, 1974). Genetic algorithm includes problem specific detail, but is potentially a global optimiser and so may avoid the pitfalls of approximate methods.

Introduction to genetic algorithms

GAs are inspired by the process in nature where desirable genetic material from parents combine and are encoded in chromosomes to be handed down to offspring. Offspring with these features are then fitter and will survive to reproduce in the next generation. These features then get propagated throughout the population.

---

[1] An NP-complete problem is one with non-polynomial time complexity. As the problem size increases, the time needed to compute the solution exactly increases exponentially. So the solution cannot be found in a reasonable time span (see Garey and Johnson, 1979).

In GAs, a problem is encoded into a string of numbers, called a chromosome or individual. Ideally, this string has the ability to mimic genes, in that specific values of genes at certain positions in the string will make an individual fit to survive. Every chromosome has an associated fitness value which quantifies this. This value is evaluated through a fitness function, which defines how well the string is performing as a solution to the problem. Fitness values partly determine the probability of a given string mating and reproducing in a particular generation. The use of a population of possible solutions to the problem offers a parallelism in the search space known as implicit parallelism.

In the area of the transportation NDP, GAs have been used to process constraints between projects (Xiong and Schneider, 1995). For example, some projects may not be selected simultaneously, while some projects must be selected simultaneously. Outside of transportation, there are prolific applications of GA, from simulating economic systems (Dawid, 1996) to path planning in a mobile robot environment (Michalewicz, 1996).

Solution of the problem by genetic algorithm

A natural representation for this problem is to simply encode a string as a sequence of projects. The projects at the front of the sequence get implemented first and projects at the end of the string may not get implemented at all, depending on the budget constraint. All the potential projects are represented in the string. So that if 100 projects are under consideration, then the length of the string is fixed at 100.

The budget for the planning period of y years will have to be estimated. Given a sequence, the budget allocated every year will dictate which projects get implemented in year 1, year 2 and so on. Each year, the budget is exhausted and if a project is only fractionally completed, it may borrow funds from the following year. This then translates into a timetable of projects. The step by step process is illustrated in Appendix A, Algorithm A1.

Next, the fitness value of all chromosomes must be evaluated. A fitness value is dependent on the output of traffic assignment, which is executed for each time period analysed. Specifically, it is the system cost of the base case, or "do nothing case" minus the cost when the projects are implemented in the order determined by the timetable. The objective is to maximise the difference of the discounted costs. A summary of the procedure for accomplishing this is given in Appendix A, Algorithm A2.

Genetic Operators

In a general GA, there are three fundamental genetic operators: selection, crossover and mutation. Selection attempts to simulate natural selection, in which the fitter individuals have a higher probability of survival. Crossover is the process whereby offspring inherit a combination of desirable genes from parents. Finally, mutation is a method of introducing randomness and increased diversity in the population.

Tournament selection is used because it has been proved to yield superior results. In this method, two or more individuals are randomly selected from the population and the fitness values compared. The fitter individuals must fill an intermediate population pool before crossover is executed.

Crossover is a crucial step in the GA, as it is the operator where "good genes" from parents are inherited to produce even better individuals. The simplest form of crossover is the interchange of two segments of a string at a random cutpoint. For example, if the parent strings are:

parent1	A	A	A	B	B	B	B
parent2	C	C	C	D	D	D	D

then with crossover after the cutpoint, which is between positions 3 and 4, the new offspring are:

child1	A	A	A	D	D	D	D
child2	C	C	C	B	B	B	B

In the chromosomal representation of the problem under consideration, there is an added restriction that no project may be implemented twice, so that the string 1234345 is infeasible. It is clear that simple crossover may result in illegal strings. To counter this problem, repair algorithms could be written to transform illegal individuals, or crossover could be modified so that the resulting new strings are always feasible. The crossover routine used is partially mapped crossover (PMX) proposed by Goldberg and Lingle (1985) for the travelling salesman problem. When crossover occurs, there is effectively a mapping between the values of the strings. If the following strings are considered for crossover:

parent1	1	5	3	4	2	6	7
parent2	5	6	7	4	1	2	3

then interchanging the elements between two cutpoints at positions 2/3, and 5/6, would result in the following illegal strings:

cross1	1	5	7	4	1	6	7
cross2	5	6	3	4	2	2	3

The mapping for this exchange is 3↔7, 4↔4 and 2↔1. The duplications outside the interchanged segments are repaired by using this mapping again. The below shows the resulting individuals, with the numbers in italics representing the elements which were repaired.

child1	*2*	5	7	4	1	6	*3*
child2	5	6	3	4	2	*1*	*7*

For problems such as project scheduling or the travelling salesman problem, the order in which a number appears in a string is important. Hence, to supplement crossover, there is the mutation operator and the inversion operator, which reorders the elements within a string.

In mutation, two random positions in a string are chosen and the values within exchanged. For example, AAABDEC becomes AAACDEB for mutation of elements B and C.

Inversion, as its name suggests, is a process which inverts the ordering of a segment of the chosen string. That is, AABCDA becomes AADCBA if segment BCD is inverted.

These operators usually occur with a small probability, as the main recombination of fit individuals occur with selection and crossover. Too high a probability may result in slow convergence of the algorithm.

For details on the exact procedure for running the GA, the reader is referred to Appendix A, algorithm A3.

Experimental Results

The GA as described above was implemented on the Northwest corridor of the Perth, Western Australia metropolitan region. This region consists of a total of 2335 links and 782 nodes. The morning peak, with 11928 OD pairs, is selected as the period of analysis. As a test, the 26 components of three broad projects (Table 1) were chosen to be considered over a planning period of five years, with impacts also being analysed over the next ten years in five year intervals. The costs are discounted at a rate of 4%.

Table 1 Description of Broad Road Projects

No.	Project Description
1	Mitchell Freeway widening
2	Wanneroo Road widening. This road starts and ends further north than 1 above.
3	Marmion Road widening. This road is to the west of the Wanneroo Road project, but does not stretch as far north as 2.

The above projects are encoded into road link form and the resulting number of links to be selected and scheduled totals 26. Hence in the GA, a chromosome of length 26 is required. For this length of string, the solution space is still large, consisting of 26! = 4.0329E+26 possibilities. The cumulative budget over the 5 years is set so that it covers approximately 75% of the total cost of all 26 projects. This means that some projects will not get implemented. However, the projects that are not considered still remain in the lower end of the string so that useful information is retained.

For a population size of 200, 50 iterations were performed, with the time taken per iteration being approximately 2 hours. The burden of computations lie in the traffic assignments, as each iteration consist of (5+2)*200 traffic assignments - one for each year and each individual. However care has been taken to optimise the Frank-Wolfe traffic assignment algorithm used, so that each assignment takes approximately 5 seconds on a Sun Ultra 1 workstation.

The chart below shows the progression of the best individual across generations. It can be seen that the best individual does not change appreciably from generation 34 onwards. This flattening indicates that the GA has reached a steady state and improvements hereafter are negligible.

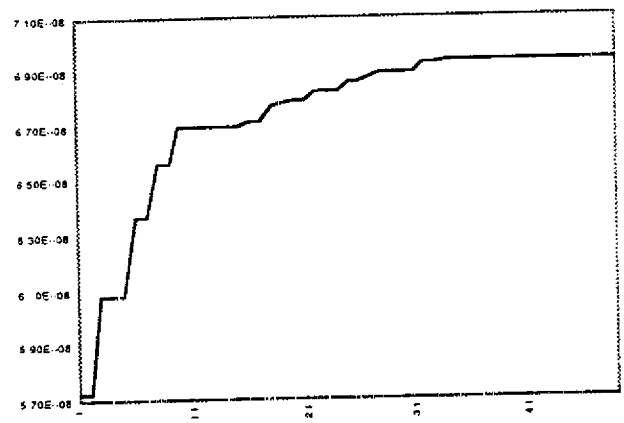

Figure 1 Plot of fitness of the best individual in a generation

The timetable for the fittest individual at the last generation is given in Appendix B, timetable 1. It can be seen that the north-bound widening projects on Wanneroo Road are very important, as are the north-bound widening projects on Mitchell Freeway. Projects 1-4 are selected for scheduling even though the ramps are constructed without the freeway links. Timetable 2 in Appendix B does not select projects 1-4 but has almost exactly the same payoff. The effect of not implementing projects 2, 3 and 4 in timetable 2 is that about half of project 8 gets implemented, and this makes no difference in the benefits.

Timetable 3 in Appendix B shows an alternative timetable which is different from timetable 1 but has a very similar payoff (the fitness value is less than 1% lower). There is similarity in the order of scheduling projects 5-26, with a tendency for the cheaper projects in the southern direction to be favoured. The big difference is that the new sections of Mitchell Freeway is constructed by year 4. This alternative may be more attractive from a policy perspective, as a new section of freeway is a highly visible form of investment.

Discussion and extensions

A practical feature is that at the end of the iterations, we have not just one solution but a set of fairly good solutions to pick from. Hence, should political priorities change, it is possible to select an individual in the population which is not optimal, but close to optimal, and has projects with the desired policy effects. Furthermore, the model is flexible and incorporates many of the vertical and horizontal interdependencies which is a feature of urban traffic network project analysis.

A possible criticism of this model is that only travel time is used as a criterion for optimisation. There are many other criteria which are not considered, for example, land use or environmental impacts. However, these can readily be taken into account by GA (Qiu, 1995). Vehicle operating cost calculations are based on time savings, and are included in the model. The model is also capable of analysing different time periods within the day, for example, the AM peak, the PM peak and the off-peak. However, there is the computational burden of a traffic assignment for every distinct period.

Directions for further research arise from the difficulty that in this representation the full set of potential projects may not be known beforehand. Ideally, the entire set of links in the network and the potential projects on them would be under consideration. This would result in an extremely long string, consuming vast amounts of memory. However, the budget constraint imposes a natural limit on the length of a string. For example, an individual will consist of a set of randomly chosen projects that exhaust the combined budget over the project planning period. If the same genetic operators were applied to this representation, once the initial random population has been set, we are essentially constraining ourselves to a subset of the network, as it is not possible that every single link has been selected into the population. Therefore, the genetic operators such as mutation and crossover would have to be modified to encompass the search space. Further research will consider this issue and whether the model may be extended to include other modes.

Acknowledgement

The author would like to thank John Taplin for many helpful comments and suggestions.

Appendix A Algorithms

Algorithm A1 Translating an individual into a construction schedule or timetable.

1. Let the year be $y = 1$. Let the project p_i be the ith element in the individual. Set $i = 1$. Let the budget be $b(y)$, the cost of a project, $c(p_i)$.
2. If $b(y) \geq c(p_i)$ then complete project p_i and set $b(y) = b(y) - c(p_i)$. Go to 3. Else go to 4.
3. Let $i = i + 1$ and return to 2. Stop if i reaches the number of projects.
4. Set $y = y + 1$ and return to 2. Stop if y reaches the end of the planning period.

Algorithm A2 Evaluating the fitness of a chromosome

1. Find the construction schedule of the chromosome Let the total cost be denoted by $t_c = 0$ Set $y = 1$.
2. Implement the projects for year y in the network. Run the traffic assignment for year y in the improved network. Using the flows, calculate the discounted costs, c_y. Set $t_c = t_c + c_y$
3. Let $y = y + 1$. If y reaches the end of the planning period then go to 4, else go to 2.

   Run the traffic assignment for year y. Using the flows, calculate the discounted costs, c_y. Set $t_c = t_c + c_y$
4. Let $y = y + 1$. If y reaches the end of the evaluation period then go to 6, else go to 4.
5. Set the fitness value to the total base cost minus the total cost.

Algorithm A3 A GA for the dynamic project scheduling problem for optimising a transport network

1. Set iterations $i = 1$, year $y = 1$.
2. Initialize a random population of strings.
3. For every member of the population, evaluate its fitness function as described.
4. Perform selection, crossover, mutation and inversion as described.
5. Set $i = i + 1$. If i reaches the maximum number of iterations, then stop. Else, go to 3.

Appendix B Timetables

Timetable 1 **Best project timetable determined by GA at iteration 50**

	Prj No.	Prj Type	Dirn	cost in millions	year 0	1	2	3	4
MITCHELL	1	nl	N	86.93					
	2	r	N	0.38				1	
	3	nl	S	86.93					0.25
	4	r	S	0.38		1			
	5	w	N	0.33			0.64	0.36	
	6	w	N	2.11		1			
	7	w	N	0.05	1				
	8	w	S	0.08					
	9	w	S	2.20	1				
	10	w	S	0.31				0.11	0.89
WANNEROO ROAD	11	w	N	5.69	1				
	12	w	N	0.11	1				
	13	w	N	1.47	1				
	14	w	N	1.34	0.75	0.25			
	15	w	N	0.25	1				
	16	w	N	1.79	1				
	17	w	N	0.50		1			
	18	w	N	0.29		0.36	0.64		
	19	w	S	5.69					
	20	w	S	0.11					
	21	w	S	1.47	1				
	22	w	S	1.34					
	23	w	S	0.25		1			
	24	w	S	1.79			1		
	25	w	S	0.50					
	26	w	S	0.29				1	

Key for project types

nl new link
r ramp
w widening

Timetable 2 **Alternative project timetable with almost exactly the same fitness value as Timetable 1**

	Prj No.	Prj Type	Dirn	cost in millions	year 0	1	2	3	4
MITCHELL	1	nl	N	86.93					
	2	r	N	0.38					
	3	nl	S	86.93					
	4	r	S	0.38					
	5	w	N	0.33			0.7	0.3	
	6	w	N	2.11		1			
	7	w	N	0.05	1				
	8	w	S	0.08					0.56
	9	w	S	2.20	1				
	10	w	S	0.31				0.22	0.78
WANNEROO ROAD	11	w	N	5.69	1				
	12	w	N	0.11	1				
	13	w	N	1.47	1				
	14	w	N	1.34	0.75	0.25			
	15	w	N	0.25	1				
	16	w	N	1.79	1				
	17	w	N	0.50		1			
	18	w	N	0.29		0.48	0.52		
	19	w	S	5.69					
	20	w	S	0.11					
	21	w	S	1.47	1				
	22	w	S	1.34					
	23	w	S	0.25		1			
	24	w	S	1.79			1		
	25	w	S	0.50					
	26	w	S	0.29				1	

Key for project types

nl new link
r ramp
w widening

Timetable 3 **Second best project timetable determined by the GA, with objective function less than 1% below the best**

	sub no.	Prj Type	Dirn	cost in millions	year 0	1	2	3	4
	3	nl	N	86.93					1
M	18	r	N	0.38		1			
I	4	nl	S	86.93					1
T	5	r	S	0.38				1	
C	30	w	N	0.33			0.38	0.62	
H	31	w	N	2.11			1		
E	32	w	N	0.05	1				
L	27	w	S	0.08					1
L	28	w	S	2.20	1				
	29	w	S	0.31					
	38	w	N	5.69	0.87	0.13			
W	40	w	N	0.11	1				
A	42	w	N	1.47	1				
N	44	w	N	1.34	1				
N	46	w	N	0.25	1				
E	48	w	N	1.79	1				
R	50	w	N	0.50		1			
O	52	w	N	0.29		0.29	0.71		
O	37	w	S	5.69			1		
	39	w	S	0.11					
R	41	w	S	1.47		1			
O	43	w	S	1.34					0.21
A	45	w	S	0.25		1			
D	47	w	S	1.79		1			
	49	w	S	0.50					
	51	w	S	0.29				0.68	0.32

Key for project types

nl new link
r ramp
w widening

References

Bellman R (1957) *Dynamic Programming* Princeton: Princeton University Press

Bergendahl, G (1969) A combined linear and dynamic programming model for interdependent road investment planning *Transportation Research* 3, 211-28

Boyce, D E, Farhi A, and Weischedel, R (1973) Optimal network problem: a branch and bound algorithm *Environment and Planning,* 5, 519-533

Dawid, H (1996) *Adaptive Learning by Genetic Algorithms Analytical Results and Applications to Economic Models*, Berlin: Springer-Verlag

Garey M R and Johnson D S (1979) *Computers and intractability: a guide to the theory of NP-completeness* San Francisco: W H Freeman and Co

Goldberg D E and Lingle R (1985) Alleles, loci and the TSP, in *Proceedings of the First International Conference on Genetic Algorithms* (edited by Greffenstette, J J) Hillsdale: Lawrence Erlbaum Associates

Frey, S C and Nemhauser, G I (1972) Temporal expansion of a transportation network -I *Transportation Science* 6, 306-23; II - 6, 395-406

Goldberg D E (1989) *Genetic Algorithms in Search, Optimization and Machine Learning* Reading: Addison Wesley

Magnanti, T L and Wong, R T (1984) Network design and transportation planning: models and algorithms *Transportation Science* 18(1), 1-54

Michalewicz, Z (1996) *Genetic Algorithms+Data Structures=Evolution Programs (Third edition)* New York: Springer Verlag

Qiu M (1995) "Prioritising and Scheduling Road Projects by Genetic Algorithm" *Proceedings of International Congress on Modelling and Simulation*, 1, 290-295, Newcastle, Australia.

Steenbrink P A (1974) *Optimization of Transport Networks* London: Wiley-Interscience

Rothengatter W (1978) Application of optimal subset selection to problems of design and scheduling in urban transportation networks *Transportation Research B* 13B, 49-63

Xiong Y and Schneider J B Processing of constraints in transportation network design problem *Journal of Computing in Civil Engineering* 9 (1), 21-28