22nd Australasian Transport Research Forum
Sydney, September 1998

A comparison of different methods for selecting and scheduling transport investments

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

Abstract:

This paper examines results from computational comparisons between different methods for choosing and scheduling transport investments. Using a road system which is widely used as a test network in the literature, two genetic algorithm specifications are compared and contrasted with existing procedures. The genetic algorithm approach presented in this paper differs from other methods, such as simulated annealing, in two aspects. First, the process of selecting and scheduling projects is done simultaneously, rather than in two stages, and second, some assumptions in modelling may be relaxed. The former is important as network design should consider at the outset when selected projects are to be implemented. In fact, it has been stated that it may be more important to find a good time schedule of investments than to determine the optimal investment program for a future time period. A dynamic planning concept is therefore required as a decision support tool that stresses the time scale of project realization in the economic evaluation of projects.

Contact author:

Vivian Salim
Department of Information Management and Marketing
The University of Western Australia
NEDLANDS WA 6907

Telephone: (08) 9380 3195 Fax: (08) 9380 1004
Email: vsalim@ecel.uwd.edu.au

Introduction

A problem faced by road authorities is that of choosing a subset of projects to be implemented from a list of proposed projects. Although this problem is easy to state, it is deceptively hard. The crux of the difficulty lies in the conflicting objectives between road planner and road user. The planner's design objective is to minimize total system time, while the road user seeks to minimize their own personal travel time. Hence there are two levels of optimisation and the shape of the search surface cannot be explicitly evaluated. In an urban situation, the system performance of a design must be evaluated by executing a user equilibrium traffic assignment. Hence the problem is quite computationally intensive. In addition, the NDP has been classified together with the hardest problems in optimisation (Magnanti and Wong, 1984).

This difficult problem and its variations is referred to as the network design problem (NDP) and it has been well studied in the literature since Leblanc (1975). Leblanc tested his branch and bound method for solving the discrete NDP on a road network with 76 links and 24 nodes in Sioux Falls, South Dakota (see Figure 2 in the appendix). Since then, this network has become a standard network used in the empirical testing of methods in several papers. The aim of this paper is to compare the findings in these papers with two different genetic algorithms (GA).

The chosen method, GA, is a powerful randomised optimisation procedure inspired by the theory of evolution and genetics. It has had much success in solving computationally difficult problems. In GA, the problem is encoded into a population of strings of numbers known as "chromosomes". Each string represents a solution to the problem and has an associated "fitness" value dictating how "fit to survive" or optimal the solution is. Evolution is then simulated on the population normally using the 3 standard procedures *selection, crossover* and *mutation* for a certain number of generations or until a convergence criterion is satisfied. An example of a convergence criterion may be variety of individuals in the population. A smaller variation indicates that the best individual is propagating strongly through the population and taking over. Selection is the process of picking out the winners in the population which have a higher probability of mating and surviving into the next generation. Crossover is the process where the information in strings are interchanged so that better individuals may be found. Mutation is a randomising procedure to reduce the tendency of a homogenous population. For a more detailed exposition on GA, the reader is referred to Goldberg (1989) and Salim (1997).

There are two distinct categories of NDPs evident in the literature, the discrete case and the continuous case. In both cases, links to be added or modified in the network are pre-specified. For the discrete case, the level of capacity improvement is determined by the analyst or planner and the decision variable is binary, that is, whether a project should be chosen or not. However, due to the computational difficulty experienced with a large number of integer variables, the focus in recent years has been on the continuous case, where the capacity is not pre-determined and is obtained by the solution method. Solution methods tested using the road network of Sioux Falls are summarised in Tables 1 and 2.

Table 1 Previous work on the discrete version of the Network Design Problem using the Sioux Falls network as an example.

	Authors	*Solution method*
1	Leblanc (1975)	Branch and bound
2	Poorzahedy et al (1982)	Solves an approximate version of the NDP with branch and backtrack
3	Chang and Chang (1993)	Solves the modified NDP as given in Poorzahedy et al. (1982). Improves on the algorithm given therein.

Table 2 Previous work on the continuous version of the Network Design Problem using the Sioux Falls network as an example.

	Authors	*Solution method*
1	Suwansrikul et al. (1987)	Uses a heuristic algorithm called equilibrium decomposed optimisation (EDO).
2	Friesz et al. (1992)	Uses simulated annealing (SA), a randomised search procedure

GAs have been used to solve both the discrete and continuous NDP before. Xiong (1993) solved the discrete NDP in his PhD thesis and conducted experiments on the Sioux Falls network. However, the reported results had more to do with the performance of the GA and actual solutions were not given. Cree et al. (1996) investigated the feasibility of applying GA to the continuous network design problem. The Sioux Falls network was not trialed in that paper and the biggest network that particular GA specification was tested on was 16 links.

The contribution this paper makes is that the timing of projects is accounted for. This is an additional complication that is not often considered. Rothengatter (1979) terms interactions in a road network as horizontal or vertical. Horizontal interactions occur within the same period, while vertical interactions occur across periods. The pertinent question is, in what order should the projects be carried out given the budget constraint? Rothengatter (1986) addresses this using heuristic selection rules and a step by step procedure as given in Figure 3 (see appendix). Computations were carried out on Leblanc's Sioux Falls network and in addition, two modes, road and rail, were incorporated into the model. Unfortunately, because different projects are specified in this paper, the results from Rothengatter (1986) is not directly comparable with the other papers.

The method presented in this paper only considers one mode for simplicity. The consideration of two modes makes Rothengatter's model appear more complete. However, as may be seen from Figure 3, the method uses the cost/benefit ratio of each project to obtain a ranking of projects, and then assigns them to time periods using the budget constraint. Hence, traffic interactions are not considered until the next step. The GA on the other hand, selects and schedules projects simultaneously, taking into account traffic interactions while scheduling is being done. This is important as network design should consider at the outset when selected projects are to be implemented, especially when it may be more important to find a good time schedule of investments than to determine the optimal investment program for a future time period (Rothengatter, 1986) A dynamic planning concept is therefore required as a decision support tool that also produces the priorities of selected projects.

Model 1: The discrete case

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 gets implemented first and the 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. This then translates into a timetable of projects.

Next, the fitness value of all chromosomes must be evaluated. A standard procedure in transportation is to compare the system cost of the base case (or "do nothing case") and the cost when the projects are implemented in the order determined by the timetable. The objective, or fitness value, is therefore to maximise the difference of the discounted costs. The analysis is carried out over the planning period, when scheduling is required, and a further number of years to evaluate the effectiveness of the timetable, known as the evaluation period.

Genetic Operators

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. Because the chromosome is sequence dependent, the crossover routine used is partially mapped crossover (PMX) originally proposed for the travelling salesman problem (Michalewicz, 1996). PMX ensures that strings remain feasible after crossover, with every city being visited once or every project implemented once. This is accomplished by establishing a mapping between the segments of strings that are interchanged, then replacing any repeated numbers using this mapping. For instance, in the strings 52314 and 23541, say the numbers to be interchanged are in the 4th and 5th position Then the mapping is 3↔5 and 1↔4. The resulting strings after crossing is 52544 and 23311. After applying the partial mapping, the strings are 32541 and 25314. 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.

Model 2: The continuous case

While parallels may be drawn between the discrete case to the travelling salesman problem, in the continuous case, the equivalent is the classic timetabling problem, which is a notoriously difficult problem. The timetabling problem assigns a teacher to a class with the variable being the number of hours. In our case, we have to assign projects to priorities, with the variable being capacity. The constraints are: (1) Each project requires $\$x$ to complete. (2) Each year has a budget of $\$y$. There have been attempts to solve the timetabling problem using GAs. Michalewicz (1996) suggests that the most natural GA representation is to have a binary matrix as a chromosome. In the equivalent project timetabling case, the matrix would look like Figure 4. It makes sense to divide the capacities into discrete lots, as in practice, it is not possible, for example, to add half a lane to a road. Each project sub-matrix can have at most one non-zero element as each project can be implemented only once. The sum of the costs for projects implemented in each row must be less than or equal to the budget of that year. The problem is that this representation is memory intensive as the matrix is very sparse.

With the property that each sub-matrix can have at most one "1", this is equivalent to the following 2Xn array, where n is the number of projects:

5	1	3	2	7	9	10	8	6	4
3.2	4.0	0.5	1.2	2.8	5.5	4.8	5.6	2.2	3.4

The first row represents the project number and is gives the timing in which projects should be implemented, and the second row represents how much additional capacity is required on the project stated in the first row. In keeping with the Sioux Falls literature, the extra capacity is given in units of thousands of vehicles. The two rows are essentially two "strings" that are interconnected to form one "chromosome". Each column position represents the importance of a project so that projects at the start of the array are more important than projects at the end.

However, there is an exception to this rule. This is when it so happens that the second row shows a capacity of zero, then the gene expression is from the second row. That is, the project is of low priority and is not scheduled into the transport plan at all. Hence, it is important to include zero in the list of possible capacities. It is possible to implement the same budget constraining process as in the discrete case, where projects are implemented along the string until the budget is exhausted.

The same genetic operators as described in the discrete specification may be implemented on the array with the exception of crossover. In 2-point crossover, segments of chromosomes are interchanged between 2 randomly chosen cut-points. This is implemented on the same cut-points for both rows of the array. Hence both rows have exactly the same segments being interchanged. However, the partial mapping only needs to be implemented for the first row as it is assumed that all capacity increments are allowable for all projects so no repair mechanism is required. This may not be the case in reality as each road has unique physical conditions such as the number of lanes, the shoulder width, and whether a median strip exists. Perhaps only a couple of possible capacities are allowed per project. This is a refinement which is outside the scope of this paper, as the intention is to compare from the GA with existing methods in the literature such as simulated annealing

Although the discrete case is computationally harder using traditional optimisation methods, GAs can actually handle the discrete case more naturally. That is, in the discrete case, a single integer valued string is sufficient to represent the problem in genetic form. On the other hand, the large number of variables required using programming methods makes the problem difficult to handle

Results

Discrete Case

In a comparative study, benchmarks are required to ascertain the performance of an algorithm. As can be expected, there is a trade-off between speed and accuracy. The benchmarks found for the Sioux Falls case are summarised in Table 3. Speed is measured in iterations of the Frank-Wolfe (FW) algorithm for finding user-equilibrium (UE) flows.

Table 3 Benchmarks for the discrete specification of the Sioux Falls network[a].

Classification	*Authors*	*Speed*	*Objective (vehicle hours)*
Fastest, discrete.	Chang et al. (1993)	20 iterations of the FW algorithm	83, 700
Best, discrete.	Poorzahedy et al. (1982)	51 iterations of the FW algorithm	47, 000

[a] The same OD table and network specification was used as given in Leblanc (1975)

In order to test that the same solution is obtained by GA, only one project year is initially considered with the same budget of $3,000,000 for the first year. These projects are then evaluated over ten years. The power of GA was evident on this simple problem. Using only a population of size 10, and convergence after 5 iterations, the same solution as in Poorzahedy et al. (1982) was found (see Table 4). That is, (1,0,0,1,1) with a fitness value of 7.44e+10. This fitness value is different from the objective function quoted in Table 3 which is the vehicle hours for a single year. GAs maximise fitnesses while the other methods minimised costs. Therefore, the fitness is defined in dollar terms as the difference in discounted travel time costs over ten years between the "do nothing" base case and the case where projects are implemented in the planning time horizon, which is one year. Although the solution in Chang et al. (1993) is obtained very quickly, when evaluated over ten years, the net benefit is in fact negative.

Table 4 Best solutions for five projects in the discrete case.

	Chang et al. (1993)	*Poorzahedy et al. (1982)*	*Discrete GA*
Links 16 and 19	1	1	1
Links 17 and 20	0	0	0
Links 20 and 25	1	0	0
Links 29 and 48	1	1	1
Links 39 and 74	0	1	1
Fitness value ($)	-2.87e+09	7.44e+10	7.44e+10

Next, using the same parameters, the Sioux Falls projects are scheduled over two consecutive years. The resulting timetable is shown in Table 5. It is surprising that although more projects are implemented, the fitness value has actually dropped from the single year case. Hence projects 2 and 3 are detrimental to the system. This highlights Braess's Paradox, where the addition of a link may actually increase overall travel time. A shortcoming is realised in this specification which needs to be corrected. As long as there are sufficient funds, the translation procedure of a chromosome to a timetable forces any remaining projects in the list to be completed.

Table 5 GA solution for a two-year planning period.

	Project links	*year 1*	*year 2*
1	Links 16 and 19	1	0
2	Links 17 and 20	0	1
3	Links 25 and 26	1	0
4	Links 29 and 48	1	0
5	Links 39 and 74	0	1
	Fitness	5.36e+10	

With only 5 projects, the possible solutions are 5!=120 which may even be completely enumerated and so it may be concluded that this small problem poses no great difficulty for the GA. The number of user equilibrium traffic assignments required is 10*5=50. In fact, the same GA method was used to generate road investment timetables for years 0 to 10 for the Northwest corridor of Perth, which is a non-trivial network with 56 projects, 782 nodes, and 2335 links. The analysis was carried out over the morning peak (with 11,928 origin-destination or OD pairs), the off-peak (with 9,853 OD pairs) and the afternoon peak (with 11,635 OD pairs) (see Salim 1998).

Continuous Case

Once again, the same Sioux Falls OD table, project and network specifications as given in Suwansrikul et al. were coded. In the continuous case, there are ten projects which results in 10!=3,628,800 combinations. Speed of computation is measured in the number of user-equilibrium (UE) assignments required.

Table 6 Benchmarks for the continuous specification of the Sioux Falls network.

Classification	*Authors*	*Method*	*Speed*	*Objective*
Fastest, continuous.	Suwansrikul et al (1987)	EDO[a]	12 UE assignments	83.08
Best, continuous.	Friesz et al. (1993)	Simulated Annealing	3900 UE assignments	80.78

[a] Equilibrium Decomposed Optimisation

A decent sized population of 100 is used. 30 capacity intervals spaced 0.2 units apart (0.0, 0.2, 0.4 ,....., 5.8) constitute the range of the search space. These capacities are spaced quite close to each other. In reality, only a few feasible capacity alternatives would be available for each road project and some of these values may overlap. Hence, 30 possible values may be a sensible estimate for a problem of this size with 10 possible project alternatives. Using this GA, it is practicable to have irregularly spaced intervals.

Convergence was rapid, and by the 11th generation, the population is homogeneous, with the entire population occupied by the best individual. The solution, together with the solutions obtained in the benchmark cases, is displayed in Table 7. As is evident, a superior solution to those in Suwansrikul et al. (1987) and Friesz et al. (1992) was found. The number of UE traffic assignments required is 11X100=1100. This is more efficient than the simulated annealing method of Friesz et al., which required 3900 traffic assignments. In Suwansrikul et al. (1987) and Friesz et al. (1992), the budget was added on to the objective function as a penalty for high spending. The budget was constrained at a very large number in the GA experiments, with the effect that the problem is not constrained at all, hence the higher budget required in the GA result. Another run was implemented, this time with the budget constrained at $4,200,000 (Table 7), which is close to the value of the simulated annealing case. At convergence, the fitness value obtained was 1.588e+06, with an investment value of $4,177,960. This fitness value is lower than the one obtained by simulated annealing, however, the bundle of projects costs slightly ($32,480) less. The difference in fitness is tracked in Figure 1, which shows the progress of annual travel time costs for the two algorithms. The difference between the two sum to $16,220 over ten years.

Next, the two year case is considered. It is assumed that the budget per year is $2,000,000. At convergence, the timetable shown in Table 8 was obtained. Because fewer projects are implemented per year due to the lower budget allowed, the value of the fitness function is lower. The project capacities chosen are directionally consistent. That is, links 16 and 19 have a capacity increase of around 5 units when rounded to a whole number, links 17 and 20 has 3 units, links 25 and 26 has 2, links 25 and 48 has 3, and links 39 and 74 has 4.

Evaluation and Future Directions

Two GA specifications were compared with other methods in the literature. Both GAs are found to have some merit over previous attempts. Both specifications evaluate projects dynamically over time. The selection of a project is considered simultaneously with the priority and timing of the project, which has never been attempted before in the literature.

In the discrete case, the GA did just as well as the benchmark found in the problem set in Sioux Falls. However, using integer programming methods in the discrete case means that only small problems may be solved, whereas the GA is capable of solving realistically sized networks, such as the Northwest corridor of metropolitan Perth (see Salim, 1997).

In the single year continuous case, comparable results with the simulated annealing method by Friesz et al. (1992) were obtained at a more efficient rate. Consistent results were obtained in the 2-year case. A criticism of the continuous specification has been that it is difficult to translate the improvement in capacities into physical reality as there are many variables and constraints in the condition of the real road in question. However, in the GA method, the road engineer may pick a number of feasible capacities which will constitute the possible values allowable in the second row of the GA

Future research will closely examine the pros and cons of the continuous and discrete GA specifications as a decision making tool. The shortcoming of the discrete GA will be analysed to see whether it is difficult to specify an improved version of the algorithm, which allows projects to be discarded even though there is enough budget to pay for it. Because projects are defined at the link level, it was found that in the 2-year plan for the continuous case, some roads, for example links 17 and 20, have either direction implemented in different years. This is due to the higher congestion on link 17 compared to link 20. However, in practice, there may be a penalty in cost for constructing projects on the same road over separate years. GA is flexible enough to handle this through the use of a penalty function.

In conclusion, there is still scope for improvement before the GA may be used as decision support software for road authorities, but the results thus far has been promising.

Table 7 Best solutions obtained in the continuous case. Ten projects were considered.[a]

	Suwansrikul et al. (1987)	*Friesz et al. (1992)*	*Continuous GA (unconstrained)*	*Continuous GA (budget constrained)*
Link 16	4.6	5.4	4.0	4.6
Link 17	1.5	2.3	3.6	2.8
Link 19	5.5	5.5	5.4	4.8
Link 20	2.3	2.0	4.8	3
Link 25	1.3	2.6	2.2	1.6
Link 26	2.3	2.5	5.6	2.4
Link 29	0.4	4.5	4.8	3
Link 39	4.6	4.5	5.4	3.8
Link 48	2.7	4.2	3.2	3
Link 74	2.7	4.7	5.4	3.8
Budget	$ 2,647,580	$ 4,210,440	$ 5,432,960	$ 4,177,960
Fitness	1.64e+06	1.69e+06	1.71e+06	1.588e+06

[a] Capacity improvements are given in units of thousands of vehicles per hour.

Table 8 Best GA solution in the continuous case for ten projects and two year planning period: Sioux Falls network.

	year 1	*year 2*
Link 16	4.6	0
Link 17	2.8	0
Link 19	4.8	0
Link 20	0	3.0
Link 25	1.6	0
Link 26	0	2.4
Link 29	3.0	0
Link 39	0	3.8
Link 48	0	3
Link 74	0	3.8
Budget used	$ 3,876,720	
Fitness value	1.566e+06	

Acknowledgments

This paper forms part of the author's PhD dissertation in the Department of Economics and the Department of Information Management and Marketing. The author would like to thank her supervisors, John Taplin and Michael McAleer.

APPENDIX

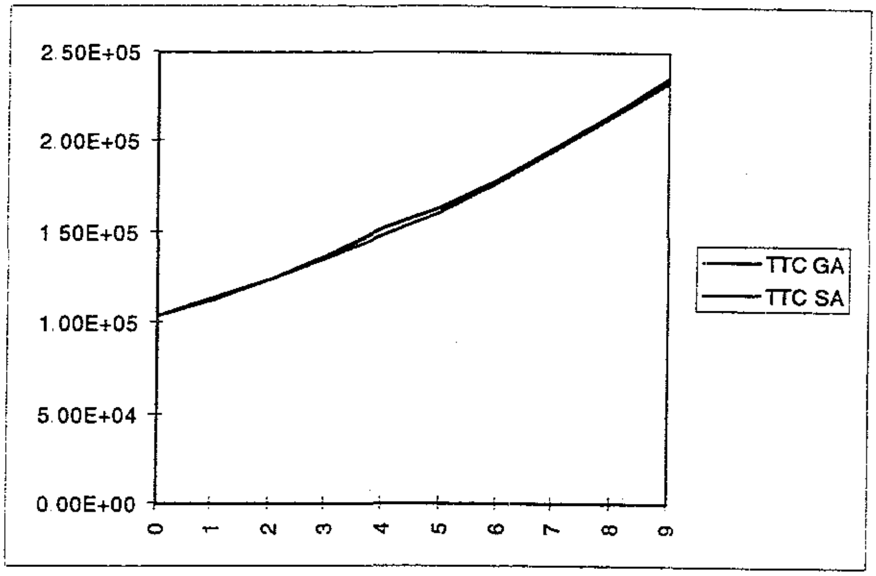


Figure 1 Plot of Annual Travel Time Costs (TTC) in dollars against Years for the Genetic Algorithm (GA) and the Simulated Annealing Algorithm (SA).

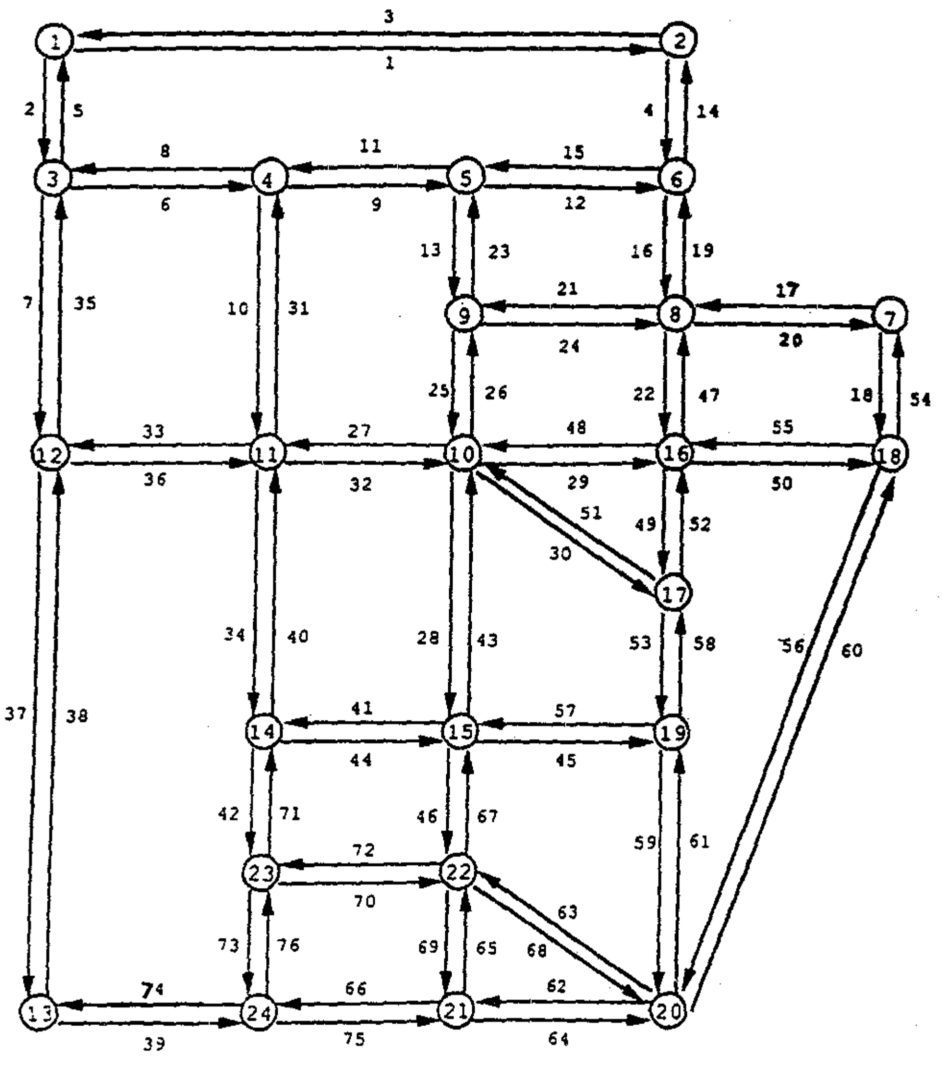
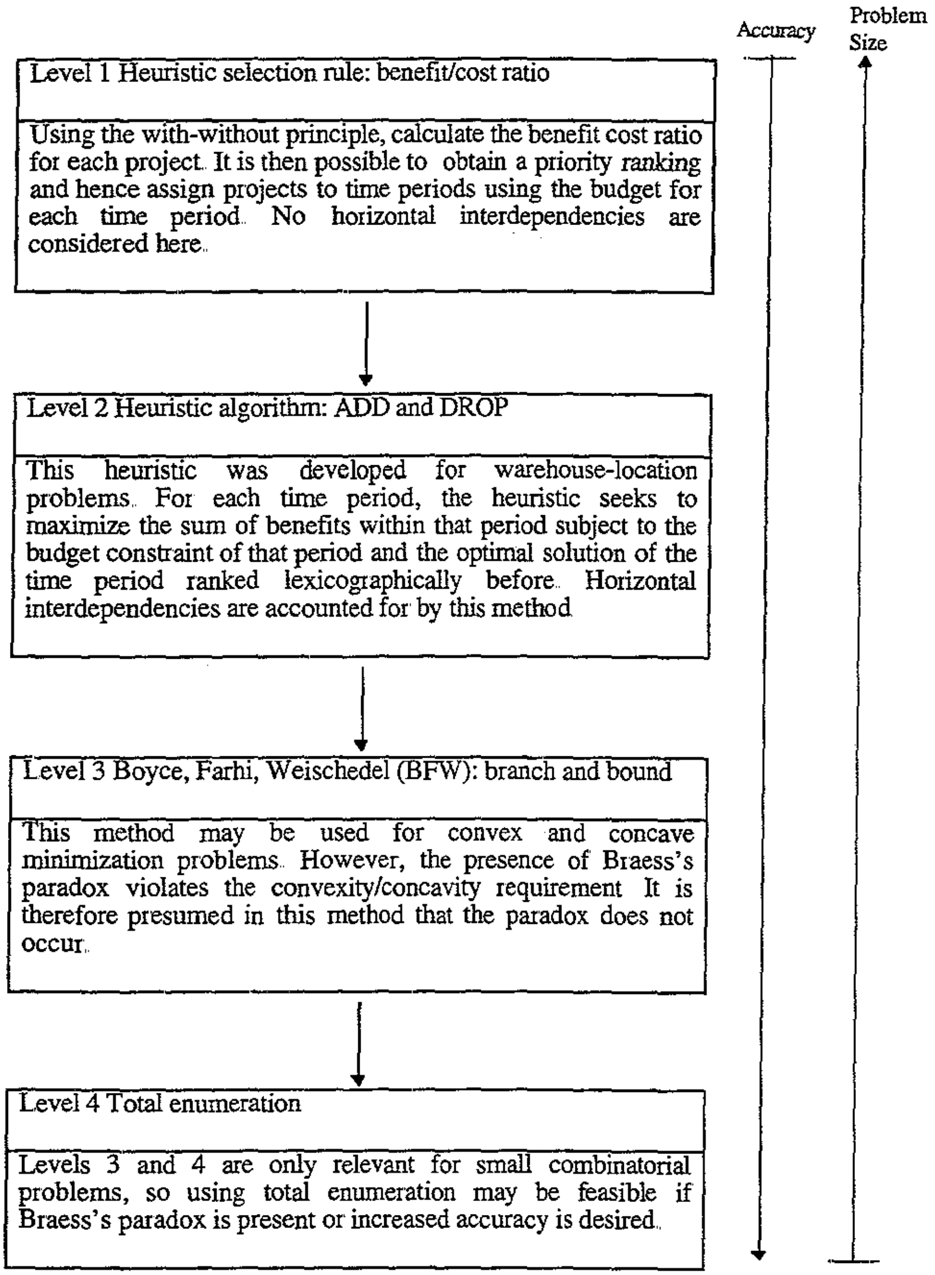


Figure 2 Sioux Falls, South Dakota



Level 1 Heuristic selection rule: benefit/cost ratio

Using the with-without principle, calculate the benefit cost ratio for each project. It is then possible to obtain a priority ranking and hence assign projects to time periods using the budget for each time period. No horizontal interdependencies are considered here.

Level 2 Heuristic algorithm: ADD and DROP

This heuristic was developed for warehouse-location problems. For each time period, the heuristic seeks to maximize the sum of benefits within that period subject to the budget constraint of that period and the optimal solution of the time period ranked lexicographically before. Horizontal interdependencies are accounted for by this method.

Level 3 Boyce, Farhi, Weischedel (BFW): branch and bound

This method may be used for convex and concave minimization problems. However, the presence of Braess's paradox violates the convexity/concavity requirement. It is therefore presumed in this method that the paradox does not occur.

Level 4 Total enumeration

Levels 3 and 4 are only relevant for small combinatorial problems, so using total enumeration may be feasible if Braess's paradox is present or increased accuracy is desired.

Figure 3 Procedure used to solve the time-dependent NDP in Rothengatter (1986).

	Project 1				**Project 2**	**....**
	capacity 1	capacity 2	capacity 3	.....	capacity 1	.....
year 1	0	1	0	.....	0	.....
year 2	0	0	0	.....	1	.....
year 3	0	0	0	.....	0	.....
.....	⋮	⋮	⋮	⋮	⋮	.....

Figure 4 Binary matrix representation of a project timetable. Capacities 1, 2, 3, ... are pre-determined additional capacities to be added on the link.

References

Chang, C.-J and Chang, S.-H (1993), "A Heuristic Algorithm for Solving the Discrete Network Design Problem" , Transportation Planning and Technology, 17, 39-50.

Cree, N. D., Maher, M. J. and Paechter, B. (1996) In *4th Meeting of the Euro Working Group on Transportation* University of Newcastle-upon-Tyne, September 1996.

Friesz, T. L., Cho, H. S., Mehta, N. J., Tobin, R. L. and Anandalingam, G. (1992), "A Simulated Annealing Approach to the Network Design Problem with Variational Inequality Constraints" , *Transportation Science,* **26,** 18 - 26.

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

Leblanc, L. J. (1975), "An Algorithm for the Discrete Network Design Problem" , *Transportation Science,* **9,** 183 - 199.

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

Michalewicz, Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs,* Springer-Verlag, New York.

Poorzahedy, H. and Turnquist, M. A. (1982), "Approximate algorithms for the discrete network design problem" , Transportation Research B, 16B, 45-56.

Rothengatter, W. (1979), "Application of optimal subset selection to problems of design and scheduling in urban transportation networks" , *Transportation Research,* **13B,** 49 - 63

Rothengatter, W. (1986) "Scheduling of interstate road and railway investments", *Environment and Planning A*, **18**, 465-483.

Salim, V. (1997) "Optimising the sequence of urban road improvement projects by genetic algorithm", pp 567-581 of *Papers of the Australasian Transport Research Forum 21(2)* Adelaide, South Australia.

Salim, V. (1998) "Scheduling Road Investments using Genetic Algorithms" forthcoming in the proceedings of The 4th International Conference on Optimization: Techniques and Applications, July 1998, Perth, Western Australia.

Suwansrikul, C., Friesz, T. and Tobin, R. (1987), "Equilibrium decomposed optimization: A heuristic for the continuous equilibrium network design problem" , *Transportation Science,* **21,** 254-263.

Xiong, Y. (1993) *Optimization of Transportation Network Design Problems Using a Cumulative Genetic Algorithm and Neural Networks* , PhD dissertation, Department of Civil Engineering, University of Washington, Washington.