DEVELOPMENT OF GENETIC-TAXONOMY EVALUATOR FOR FINDING SHORTEST PATH IN TRANSPORTATION SYSTEMS

Sutanto SOEHODHO
Department of Civil Engineering
Faculty of Engineering
University of Indonesia
Depok 16424, Indonesia
Fax. : 62-21-7270028

abstract : This research is aimed at elaborating a new methodology of shortest path finding by using taxonomy systems and genetic algorithms. Combination of the two is called Genetic Taxonomy Evaluator (GTE) which is expected to be an effective and efficient tool compared to conventional method in finding shortest path within the transport network. While keeping the characteristics of transportation networks Taxonomy Reconstructor (TR) transforms the network into taxonomy structure which is hierarchically shaped based on problem to be solved. In the process TR also creates classification of nodes in the network. This classification provides facilities to isolate the problem to the core, and criteria that can be inserted in the Genetic Algorithm (GA). A package program for GTE is then developed in C Language and performance of model is analyzed. Furthermore, results of analysis are concluded that several optimal parameters should be determined prior to searching the shortest path, and the findings are quite promising as a threshold for further researches.

1. INTRODUCTION

One of the biggest computational efforts devoted to transportation planning is undertaking the step of network assignment in which flow on each link of the network is to be determined. The big computational effort in network assignment is caused by iteration in searching for shortest path that connect each of origin-destination pair in the network, which is done by enumerating possible combination of links such as *Djakstra, Moore* and others. This problem has stimulated the research in which a method is to be sought for solving the shortest path finding with more efficient way.

The original idea of this research in developing efficient method is conducted by learning the characters of transportation network as the object and characters of human in doing their optimal decision for route choice as the subject. Learning the two has evolved with two step solution which consists of two steps. The first step is to simplify the structure of network shape by maintaining the original characters such as distance. This simplification is necessary since it eases human recognition for route choice. The second step is to adopt an optimization method with closed characters to the simplified network structure.

The ensuing chapters will explain the methodology step by step, development of the algorithm, and some computational experience with prior determination of several optimal parameters.

2. GENERAL METHODOLOGY

As already mentioned above that there are two main tasks in the methodology proposed in this research. The first task is to restructure shape of transportation network into new network where its nodes arranged hierarchically to connect their closest nodes which are located at different level. As what shown in the original network, restructured network should still reflects the relative position of each node to the others including their connecting links. This sort of network characters is owned by one of the paradigms of computer learning called taxonomy formation. Furthermore, the closeness of any origin node to destination node of a trip in transportation system is the most important perception for route choice. So it is emphasized in the hierarchical taxonomy formation of the network.

As for the second task, it has to deal with the solution of non-linear optimization problems since the impedance or link performance function is indicated by non-linear function such flow-dependent travel time of road performance. Unlike conventional method, the simulated annealing method is known to be able to provide global solution to such problems, however its original physics nature imposes lengthy solution for mapping problem of transportation systems. It is then chosen in this approach of using the genetic algorithm which has proven to give good solution in many disciplines. Its properties with parallel solution are very suitable for transportation problems and further development of parallel processor in computing systems.

It is the general methodology adopted by this research in which the hybrid of both taxonomy re-constructor and genetic algorithm is developed as a method, latter called as Genetic-Taxonomy Evaluator (GTE), in which a set of source code is developed in C-Language for further investigation of its performance regarding optimal parameters.

3. TAXONOMY RECONSTRUCTOR (TR) AND GENETIC ALGORITHM (GA)

The task of taxonomy re-constructor (TR) is to arrange the original road network into representative hierarchical network in which a set of certain nodes will be located in a layer showing their closeness and directness to the origin node being considered. The following explanation is contrived.

3. 1. Formation of Taxonomy Structure

To provide clear understanding in the taxonomy re-constructor, simple network illustrated in Figure 1 is used.

Supposed node 17 is chosen as origin node, then taxonomy re-constructor will put node 17 as the paramount in hierarchical network, while there several vertical layers consist of certain nodes for each layer. The first layer under the paramount comprises of a set of nodes that has degree of closeness one in which they are separated from the origin node only by one direct link. In the same manner the ensuing layers denote their closeness by their layer, so nodes in the second layer are separated from the origin node by two links and so forth. By doing so, simple network in Figure 1 is restructured as shown in Figure 2.

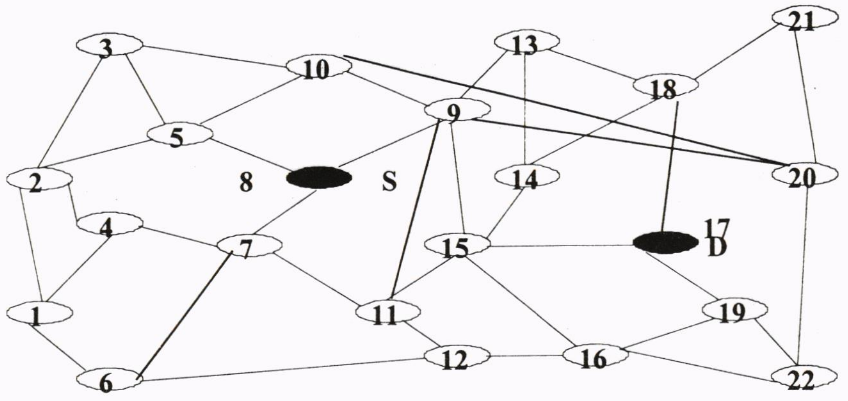


Figure 1. Simple Transportation Network

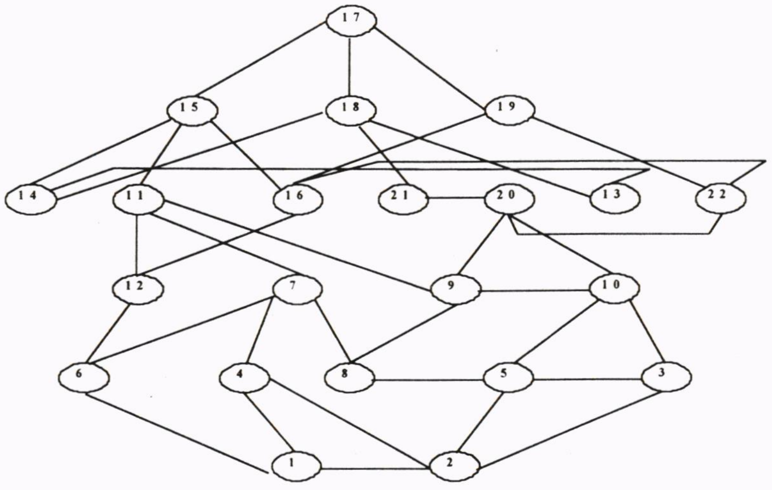


Figure 2. Taxonomy Structure of Simple Transportation Network

Localized Problem

Route choice behavior of individual appear to be various, one may chooses the route that directly connects his/her destination, by other my choose indirect route. If all trip maker will take only direct route, the problem of route choice may reduce a lot in computation effort, however, the practice always shows many possibilities since performance of road changes by time and flow condition (e.g., congestion). In the proposed GTE problem can be generalized for all possibilities, so that several route types are defined as follows;

- ***Direct Route*** is defined as a route that connects origin node and destination with sequential nodes in which only one node in each row is connected e.g., route 8-9-11-15-17 shown by Figure 2.

- ***Indirect Route*** is defined as a route that connects origin node to destination node with possibility of having horizontal connection of minimal two nodes at the same row, e.g., route 8-9-10-13-18-17 shown by Figure 2.

- ***Round Route*** is defined as a route that connects origin node and destination node with possibility of having vertical connection of minimal two nodes of different rows, e.g., route 8-7-6-12-16-15-17 shown in Figure 2.

By having route types, the process of route choice in GTE could be controlled to a certain level of complexity such of finding route with direct route with least size of route set, and forth to the biggest set with round routes. For example, to find the shortest route connecting node 8 to 17 with direct route can be focused or localized by inventory of direct routes available between rows 5 and 1. While for indirect and round routes, the search may move within the same row and/or to lower level row and back to upper row and so forth depend on the degree of search chosen.

3. 2. Genetic Algorithm (GA)

The original Genetic Algorithm (GA) (Holland, 1992) is utilized to finding the shortest path by cross-matching and mutation processes. GA uses a population of strings or character harnesses to represent data. The character harnesses used in the presentation denote the performance of road segments such as travel time, and they can be assumed as genetic entity that can be cross-matched and mutated to produce new entities. Similar to biologically genetic process, the best entity or the best route/path is sought by iteration. The worse entity is eliminated, and the better entity is processed further until no better entity is found and iteration is stopped.

4. DEVELOPMENT OF GENETIC TAXONOMY EVALUATOR (GTE)

The development of GTE comprises of two tasks. First task is to transform original transport network into taxonomy network, and second task is to genetic process to produce best route/path which incorporate the following rules :

1. Inventory of complete nodes and group of nodes at each taxonomy row
2. Choose string integers of nodes and/or group of nodes as symbols
3. Consider representation of all types of routes within the population
4. Enable probability concept for various route choice
5. Strings produced by genetic process that do not represent real network should be eliminated
6. Intensity of links or road segments should represent the road performance such as travel time for evaluation

By having the rules and various search scheme can be developed, and used for single or multi-path assignment in transportation problems.

String and Integer as Data Representation

String is used to represent taxonomy node. As for simple example given above Table 1 illustrates some integer strings that compose the inventory of taxonomy row 5 to 1 with degree of search 1.

Taxonomy Row	5	4	3	2	1
String Location	1	2	3	4	5
Actual / Symbolic					
1	8	7	11	15	17
2	8-5	9	13	18	
3		10	11-9-13	15-14-18	
4		9-5-10	11-9-10-13		
5		7-8-9			
6		9-10			
7		7-8-5-10			
8		9-8-5-10			

Table 1. Inventory of Taxonomy Row for Degree of Search 1

1)

String location	: 1	2	3	4	5
Taxonomy row	: 5	4	3	2	1
Integer Symbol	: 1	1	1	1	1
Actual node	: 8	7	11	15	1

2)

String location	: 1	2	3	4	5
Taxonomy row	: 5	4	3	2	1
Integer symbol	: 2	6	2	2	1
Actual node	: 8-5	9-10	13	18	17

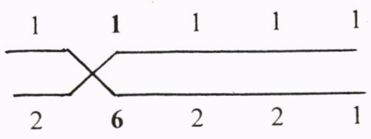
3)

String location	: 1	2	3	4	5
Taxonomy row	: 5	4	3	2	1
Integer symbol	: 1	1	3	2	1
Actual node	: 8	7	11-9-13	18	17

First example shows string 11111 that represent direct route 8-7-11-15-17, string 26221 of second example represent indirect route 8-5-9-10-13-18-17, while string 11321 of third example represent round route 8-7-11-9-13-18-17. So it is clearly seen that length of string reflects number of taxonomy rows.

Cross-Matching Process

Cross-matching process of GTE is adopted from GA. To give an illustration the strings given above can be used. For example string 11111 is crossed with string 26221 with a certain value probability, and random process of choosing location 2 as crossing location.



By this process some new strings will be produced. Cross-matching of the string 11111 (route 8-9-10-13-18-17) and string 21111 (route 8-5-7-11-15-17) produces new route 8-9-10-13-18-17 which is valid, and new route 8-5-7-11-15-17 which is not valid since no connection between nodes 5 and 7 in the real network.

Mutation Process

In taxonomy the mutation process may increase possibility of improving the route. It is then necessary to do the process before discarding the invalid route. Again probability concept is helpful to adjust the possibility of discarding route. The valid route may have high probability of mutation, while invalid route may have low probability. Further, the mutation process can be done at the location in which inconsistency appears such as route 8-5-11-15-17 that may experience mutation process at location 2 with two possibilities ;

a) 8-5-7-11-15-17 mutates into 8-5-10-11-15-17

or

b) 8-5-7-11-15-17 mutates into 8-5-9-11-15-17

The possibility a) may repeat several times the process since the produced route is still invalid. The repetition could be done until valid route is gained, or stopped by giving some penalty to the string. While possibility b) has certainly created a new route.

5. DEVELOPMENT OF ALGORITHM AND SOURCE CODE

In order to comprehend GTE mechanism in searching shortest path an algorithm is developed. The algorithm is structured based on C-Language which is widely used, and the illustrated as followings :

```
****************************************************************

#include <stdlib.h>

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <alloc.h>
#include <dir.h>
#include <dos.h>
#include <time.h>

#include "global.h"
#include "gte.h"
#include "fileinit.h"
#include "taxonomy.h"
#include "inventrs.h"
#include "genetic.h"
*************************************************************************
NETWORK struct _STC *network;
int gteTimerId;
int processLevel = 2;
int main (int argc, char *argv [])
{
FILE *infile;
char outname [MAXPATH];
char *cptr;
int argCnt;

struct FILENODE_STC fileNodes;
struct TAXONOMY_ROW_STC *TaxRow;
int startPoint,
   endPoint;


 if (argc < 2) {
   cprintf ("%s [/q][/g[d|s|n]][/p[t|i|g]] <route-filename>\r\n", argv [0]);
   exit (1);
 }

 argCnt = 1;
 argCnt = processOptions (argc, argv, argCnt);

 if (argCnt >= argc) {
   cprintf ("%s [/q][/d] <route-filename>\r\n", argv [0]);
   exit (1);
 }
 if ((infile = fopen (argv [argCnt], "rt")) == NULL) {
   cprintf ("Cannot open %s\r\n", argv [1]);
   exit (1);
 }
 strcpy (outname, argv [argCnt]);
 if ((cptr = strrchr (outname, '.')) == NULL)
   cptr = outname + strlen (outname);
 strcpy (cptr, (strcmp (cptr, ".out") ? ".out" : ".ou1"));

 gteTimerId = initTimer ("GTE");
 stopTimer (gteTimerId);

 if (!readDataFile (infile, &fileNodes))
   exit (1);
 if (!getStartEndData (&fileNodes, &startPoint, &endPoint))
   exit (1);
 if (!taxonomy (&fileNodes, &endPoint)) {
   exit (1);
}
if (processLevel >= 1) {
  if (!inventarisLajur (&lajurTax, startPoint, endPoint)) {
    exit (1);
  }
  if (processLevel >= 2) {
    genetic (TaxRow);
  }
}

 gteTimerId = endTimer ("GTE", gteTimerId);

 return 0;}
*************************************************************
```

GTE Sub-routines

Furthermore, the algorithm has several sub-routines with the following explanations :

- GTE.C is main program
- TAXONOMY.C contains routines for taxonomy process
- INVENTRS.C contains routines for inventory
- GENETIC1.C contains routines for genetic process
- FILEINIT.C contains routines for reading input file and sending to memory
- GLOBAL.C contains global routines and supports the main program
- ERRORS.C contains routines for printing when errors occur

6. COMPUTATIONAL EXPERIENCES

To see the performance of GTE algorithm in solving the shortest path problems, a medium size of transport network given by Sioux-Falls city (LeBlanc, et.al., 1975) is used. The network has 24 nodes and 76 links in total which is shown in Figure 4, and the link performance function is given in Table 2.

There are six (6) parameters involved in the GTE algorithm. These parameters are necessary to be set up prior to the search since they determine the optimal search, and have the following definitions :

- **Population Size (PS)** - denotes the size of string population used in the genetic process
- **Best Paths to Keep (BP)** - denotes number of the best strings that will generate next strings without cross-over and/or mutation
- **Number of Generations (NG)** - denotes number of generations to be processed
- **Cross-Over Probability (COP)** - denotes probability of cross-over process at each generation

- **Mutation Probability (MP)** - denotes probability of mutation process at each generation for valid strings or routes
- **Invalid Mutation Rate (IMR)** - denotes the number of trials to be mutated to invalid string to obtain valid strings

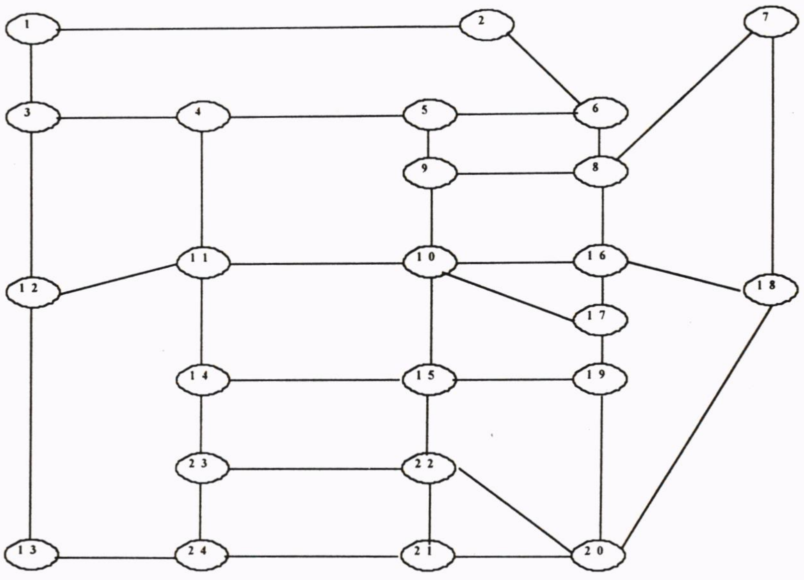


Figure 4. Sioux-Falls City Network

Prior to conducting the search for shortest path the six parameters mentioned above are to be determined first, and the popular ***Hill Climbing*** method is used. Parameter determination is done by searching one parameter as variable and holding the others as constant within a certain range with decreasing/increasing move step to converge to real shortest path (obtained by conventional ***Moore Algorithm***) and reach minimal computing time. Similar process is applied for each other parameter intermittently until optimal value is obtained, and so on.

To have conclusive optimal parameters, the Sioux-Falls network is utilized by testing two Origin-Destination pairs, namely (11-16) and (1-18) with feasible paths of 286 and 14,256 consecutively. The results of parameters determination is summarized in Table 3.

LINK	TRAVEL TIME (HOUR)
(7,18) & (18,7)	0.02
(10,15) & (15,10)	0.06
(3,4) & (4,3)	0.04
(1,3) & (3,1)	0.04
(1,2) & (2,1)	0.06
(3,12) & (12,3)	0.04
(12,13) & (13,12)	0.03
(18,20) & (20,18)	0.04
(10,11) & (11,10)	0.05
(16,18) & (18,16)	0.03
(15,19) & (19,15)	0.03
(10,17) & (17,10)	0.08
(8,9) & (9,8)	0.1
(4,11) & (11,4)	0.06
(5,9) & (9,5)	0.05
(9,10) & (10,9)	0.03
(15,22) & (22,15)	0.03
(11,12) & (12,11)	0.06
(4,5) & (5,4)	0.02
(13,24) & (24,13)	0.04
(24,21) & (21,24)	0.03
(20,21) & (21,20)	0.06
(2,6) & (6,2)	0.05
(6,8) & (8,6)	0.02
(7,8) & (8,7)	0.03
(5,6) & (6,5)	0.04
(23,24) & (24,23)	0.02
(21,22) & (22,21)	0.02
(14,23) & (23,14)	0.04
(22,23) & (23,22)	0.04
(14,15) & (15,14)	0.05
(16,17) & (17,16)	0.02
(17,19) & (19,17)	0.02
(19,20) & (20,19)	0.04
(11,14) & (14,11)	0.04
(20,22) & (22,20)	0.05
(8,16) & (16,8)	0.05
(10,16) & (16,10)	0.04

Table 2. Sioux-Falls Link Performance Function

O-D Pair	PS (paths)	BP (paths)	NG (paths)	COP (%)	MP (o/oo)	IMR (times)
11-16	50	4	10	60	15	10
1-18	400	4	10	60	15	10

Table 3. Optimal Parameters for Different O-D Pairs

It can be concluded from Table 3 that almost all parameters for different O-D pairs have the same values except for PS in which the numbers differ reciprocally with their feasible paths. O-D pair 11-16 with 286 feasible paths has PS value of 50 which is about 17%, while O-D pair 1-18 with 14,256 feasible paths has PS value of only 400 which is 2.8%. It seems that the ratio is reciprocal, hence the greater the number of feasible paths the less the value of PS.

It is recorded that once the optimal parameters are determined the path finding process could rapidly converge to the shortest one. In the case of O-D pair 11-16 path finding process is completed in 0.17 second, while O-D pair 1-18 is 6.09 seconds. Furthermore, computation experience has shown that the convergence of shortest path finding is achieved within less number of generation iterations. Table 4 shows the convergence as compared with number of iterations for O-D pair 11-16, and the illustration is given in Figure 5 in which convergence is achieved at the 7th iteration. The similar situation is also shown by Table 5 and Figure 6 in which convergence is achieved at the 7th iteration.

PS (path)	50	**Shortest Path**									
BP (path)	4	**# of Generation Iterations**									
NG (path)	10	1	2	3	4	5	6	7	8	9	10
COP (%)	60										
MP (o/oo)	15	22	18	18	17	9	9	9	9	9	9
IMR (times)	10										

Table 4. Convergence of O-D 11-16

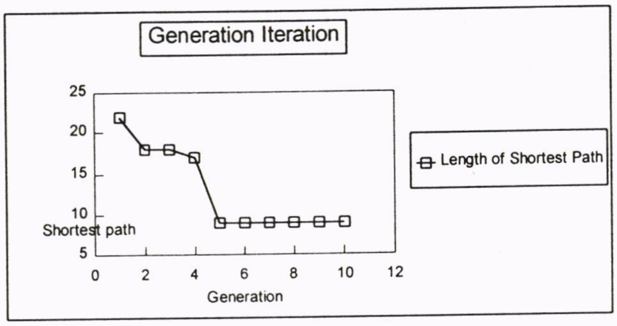


Figure 5. Convergence of O-D 11-16

		Shortest Path									
		# of Generation Paths									
PS (path)	400										
BP (path)	4										
NG(path)	10	1	2	3	4	5	6	7	8	9	10
COP (%)	60										
MP (o/oo)	15	50	30	27	21	21	21	18	18	18	18
IMR (times)	10										

Table 5. Convergence of O-D 1-18

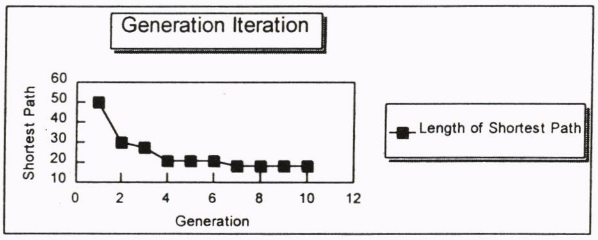


Figure 6. Convergence of O-D 1-18

6. CONCLUSIONS

A new approach of way-finding in transportation systems, namely shortest path, is proposed in this paper. The approach is a hybrid of both taxonomy re-constructor and genetic algorithm which is called Genetic-Taxonomy Evaluator (GTE).

The GTE is coded in C-Language as a tool to evaluate the performance of the proposed method. The core of GTE comprises of six parameters that should be determined prior to the searching process. These parameters are essentially embedded from the process of cross-matching and mutation as the nature of genetic algorithm, and experience of determining the parameters shows that they have specific characteristics. Having optimal value of these parameters lead the computation to rapid convergence to real shortest path sought. Limited computation effort is evaluated by utilizing real data of Sioux-Falls network which consists of 24 nodes and 76 links and is considered as medium size network.

Findings of new approach (GTE) and its performance is very promising and could be a threshold for further research and refinement.

Acknowledgment

This research is funded by Indonesia-Japan Toray Foundation, and to the foundation the author is grateful.

REFERENCES

Carbonell J. (1990), **Paradigms for Machine Learnings, Machine Learning Paradigms & Methods**, Carbonell J (Ed.)., The MIT Press, page 1-9.

Davis L., & Steenstrup M. (1987), **Genetic Algorithms & Simulated Annealing : An Overview, Genetic Algorithms & Simulated Annealing**, Davis L. And Morgan Kaufman (Eds.), page 1-11.

Dial, R.B. (1971), **Probabilistic Multipath Traffic Assignment Algorithm Which Obviates Path Enumeration**, Transportation Research 5(2), page 83-111.

Fisher Jr., D.H., Pazzani M.J. and Langley P. (1991), **Concept Formation : Knowledge & Experience in Supervised Learning**, Morgan Kaufman (Ed.).

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

Grefenstette J.J. (1989), Incorporating Problem Specific Knowledge into Genetic Algorithms & Simulated Annealing, Davis L., **Proceedings of The 3rd International Conference on Genetic Algorithms**, Morgan Kaufman (ed.), page 42-60.

Grefenstette J.J., & Baker J.E. (1989), How Genetic Algorithms Work : A Critical Look at Implicit Parallelism, **Proceedings of The 3rd International Conference on Genetic Algorithms**, Morgan Kaufman (ed.), page 20-27.

Holland J. (1992), **Adaptation in Natural & Artificial Systems**, Second Edition., The MIT Press.

LeBlanc L.J., Morlok E.K. and Pierskalla W.P. (1975), **An Efficient Approach to Solving The Road Network Equilibrium Traffic Assignment Problems**, Transportation Research, Vol. 15, No. 2/4, page 115-124.

Michaelwics Z. (1992), **Genetic Algorithms + Data structures = Evolution Programs**, Springer-Verlag.

Sheffi Y. (1985), **Urban Transportation Networks : Equilibrium Analysis with Mathematical Programming Methods**, Prentice-Hall, Inc.

Michaelwics Z. (1992), **Genetic Algorithms + Data structures = Evolution Programs**, Springer-Verlag.
Sheffi Y. (1985), **Urban Transportation Networks : Equilibrium Analysis with Mathematical Programming Methods**, Prentice-Hall, Inc.
Sudarbo D.H. & Gorostiza Z.C. (1992), Use of Genetic Algorithms to Optimize Vehicle Wire Harness Costs, **Internal Report, Condumes Automotive Co.**, Livonia MI, USA.