Genetic Algorithm Approach for Locating Automatic Vehicle Identification Readers

Mazen Arafeh[1] and Hesham Rakha[2]

I. ABSTRACT

The paper develops an algorithm for optimally locating surveillance technologies with an emphasis on Automatic Vehicle Identification tag readers by maximizing a travel time reliability objective function. The problem is formulated as a quadratic 0-1 optimization problem where the objective function parameters represent benefit factors that capture travel time variability along specified trips. A Genetic Algorithm is developed to solve the problem and the computational results are presented using data pertaining to a freeway section in San Antonio, Texas, as well as a number of synthetic test cases, to demonstrate the efficacy of the proposed approach.

II. INTRODUCTION

The primary application of Automatic Vehicle Identification (AVI) has been electronic toll collection. However, the use of AVI technology for travel time estimation is gaining popularity; (e.g. Houston and San Antonio metropolitan areas use AVI technology to monitor traffic operations in real-time). Consequently, different approaches have been developed for locating AVI tag readers for the purpose of estimating dynamic roadway travel times.

Sherali *et al.* (2003) developed an algorithm for optimally locating AVI tag readers by maximizing the benefit that would accrue from measuring travel times on a transportation network. The authors formulated the underlying reader location problem by considering a graph, denoted by $G(N, A)$, with a node set N and a set of directed arcs $(i, j) \in A$ connecting two potential reader locations. Sherali *et al.* formulated the problem as a quadratic 0-1 optimization problem where the objective function parameters represent benefit factors (b_{ij}) that capture the variability of travel time estimates as measured by the Coefficient of Variation of the link travel time (CV) as

$$CV_l = \frac{\sigma_l}{\mu_l}, \qquad [1]$$

Where σ_l is the standard deviation of link l, and μ_l is the expected travel time along link l. In addition, Sherali *et al.* defined C_j as site-specific costs of installing a reader at location $j \in N$, and R as the maximum number of available readers. In addition, they imposed a maximum budgetary limitation equal to B. Then, they defined a binary variable y_j as

$$y_j = \begin{cases} 1, & \text{reader located} \\ 0, & \text{otherwise} \end{cases} \quad \forall\ j \in N. \qquad [2]$$

The complete formulation of the reader location problem can be written as

$$\text{Max.} \quad \sum_{(i,j)\in A} \sum b_{ij} y_i y_j \qquad [3]$$

Subject to:

$$\sum_{j\in N} y_j = R \qquad [4]$$

$$\sum_{j\in N} C_j y_j \le B \qquad [5]$$

Sherali *et al.* designed an optimization approach based on the Reformulation-Linearization Technique coupled with semi-definite programming concepts to solve the formulated reader location problem. This solution approach was demonstrated to be sufficient for small problems (number of reader locations less than 10), but would not be ideal for large networks given the computational load of the solution approach.

In developing the benefit factors, the freeway networks were divided into four segment types that include merge, diverge, weave, and basic freeway segments (Sherali *et al.*, 2003). The INTEGRATION software was utilized to derive the benefit factors for the four freeway section types by conducting numerous simulation runs for different O-D demand configurations, roadway section lengths, and random number seeds. The simulation results were recorded as generic look-up tables that can be used for any freeway section for the purpose of computing section-specific benefit factor coefficients. These segments were then aggregated to form links, which represent a roadway section between any two reader locations.

In developing the link CV, Sherali *et al.* derived two mathematical formulations. Considering a link between two reader pairs, say (i, j), to be composed of several roadway segments, they defined the benefit factors (b_{ij}) as

$$b_{ij}^{(1)} = \left\{ \sum_{l=1}^{m} \sigma_l^2 \Big/ \sum_{l=1}^{m} \mu_l^2 \right\}^{1/2}. \qquad [6]$$

---

[1] Mazen Arafeh is a Research Assistant Professor with the Grado Dept. of Industrial Systems Engineering at Virginia Tech and a Senior Research Associate with the Virginia Tech Transportation Institute, Blacksburg, VA 24061 (e-mail: amazen@vt.edu).

[2] Hesham Rakha is an Associate Professor with the Charles Via Jr. Dept. of Civil and Environmental Engineering at Virginia Tech and the Director of the Center for Sustainable Mobility at the Virginia Tech Transportation Institute, Blacksburg, VA 24061 (e-mail: hrakha@vt.edu).

Alternatively, assuming the independence of travel times along the various links, the trip benefit function can be computed as

$$b_{ij}^{(2)} = \left\{ \sum_{l=1}^{m} \sigma_l^2 \right\}^{1/2} \Bigg/ \sum_{l=1}^{m} \mu_l \ . \qquad [7]$$

A more detailed analysis of trip travel time variability is discussed by Rakha *et al.* (2005).

Teodorovic *et al.* (2002) used a different approach to locating AVI tag readers. They proposed a composite objective function that is comprised of a weighted average of the total number of AVI tag readings and the number of O-D pairs that are at least partially covered by these readings. They developed a genetic algorithm to heuristically maximize this function. This formulation assumes that the O-D table is known *a priori* and that vehicles follow a static shortest path over the network between each O-D pair. However, the reader locations might measure travel times over any arbitrary subsets of the shortest paths between the O-D pairs, and the model does not distinguish between the benefits accruing from obtaining information regarding travel times over one portion of an O-D shortest path from another.

III. GENETIC ALGORITHM DESCRIPTION

We extend the work done by Sherali et al. (2003) to use Genetic Algorithms to solve the AVI location problem. Specifically, we develop a binary vector as a chromosome to represent the reader location problem, where a 1 indicates a reader is present at a location, and a 0 indicates otherwise. For example, for the chromosome 10111010, there are eight locations with readers assigned to locations 1, 3, 4, 5, and 7. The chromosome size depends on the size of the problem (i.e. number of potential reader locations).

The construction of a GA as described by Figure 1 requires the creation of an initial population, evaluating the fitness of the initial population (objective function), sorting the various chromosomes based on their degree of fitness (objective function value), creating new populations (children chromosomes) by using the GA operators to alter the composition of the parents, evaluating the fitness of various populations, keeping the best chromosomes, removing the unfit chromosomes, and continuing until the stopping criteria is met; specified by the number of generations.

The initial population of chromosomes was generated to include up to a total of 8 batches that were created in sequence and repeated every 8 chromosomes. Chromosomes were initialized by setting all genes to 0's and allocating readers using different schemes, as will be discussed. The first batch of chromosomes was constructed by randomly assigning readers (1's) within a chromosome, until the desired number of readers was satisfied. The second and third batches were constructed by allocating the desired number of readers starting from left to right and right to left, respectively. The forth batch of the initial population chromosomes was generated by locating readers (1's) at locations that incur minimum installation costs after ranking reader locations based on their installation cost. The fifth and sixth batches were constructed by starting from an end gene (i.e. first or last potential reader location), and assigning 1's every other gene. In the case of the fifth batch the assignment proceeded from left to right while in the second batch it proceed from right to left. Batches seven and eight were similar to batches five and six except that readers were allocated every third reader location. It should be noted that the chromosomes produced were ensured to be feasible (i.e. they all satisfied the budgetary constraint).

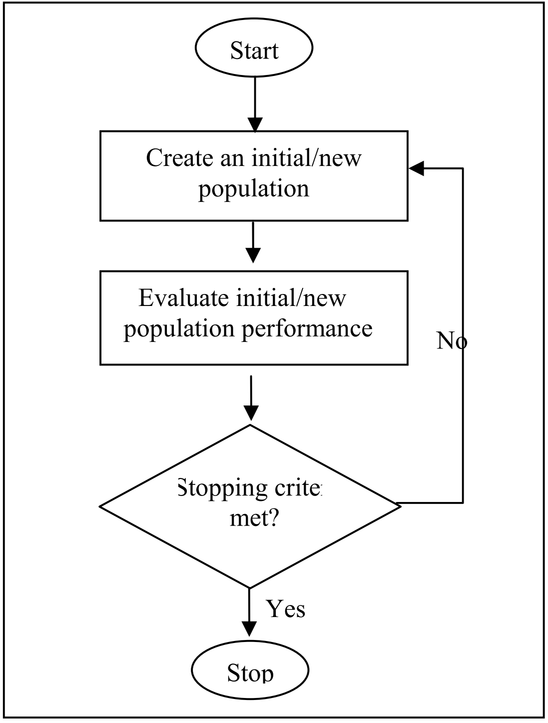


Figure 1: General Flow Chart of Genetic Algorithms

The creation of the new population involved selecting the parent chromosomes, selecting the genetic operators, applying the operators, evaluating the child chromosomes, and updating the list of best chromosomes. Parent chromosomes were selected from the list of best chromosomes (highest objective function values), which is known as Elitism. Chromosome selection was accomplished using rank selection. In rank selection, individuals in the population are ranked according to their fitness as

$$P_n = \frac{N - n + 1}{\sum_{l=1}^{N} n} \ .$$

Where N is total number of chromosomes and n is the chromosome rank. The probability a chromosome is chosen for mating is directly proportional to its rank. For example, suppose we have five chromosomes. The probability of choosing the first chromosome is

$$P_1 = \frac{5-1+1}{1+2+3+4+5} = 0.333 = 33.3\% .$$

Table 1 summarizes the probability of selecting each chromosome for the five chromosomes that were analyzed. The cumulative probabilities listed in the table are used in the selection process by generating a random number between 0 and 1. Starting at the top of the list, the first chromosome with a cumulative probability that is greater than the random number is selected. For instance, if the random number is 0.465 (r = 0.465), then 0.33< r ≤ 0.600, so chromosome 2 is selected.

Table 1: Rank Selection

n	P_i	$\sum_{i=1}^{n} P_i$
1	0.333	0.333
2	0.267	0.600
3	0.200	0.800
4	0.133	0.933
5	0.067	1.000

There are several types of genetic operators that may be used in the reproduction stages of a GA. The most commonly used classical operators are crossover operators and mutation operators. Crossover operators in general combine the features of two parent chromosomes to form two child chromosomes by swapping corresponding segments of the parents. The intuition behind the applicability of crossover operators is information exchange between different potential solutions. A chromosome can be divided into two or more segments, for example the following chromosome is divided into two segments using a single crossing point as $101|11010$. A random integer from the range $\lceil 0,\ l \rfloor$ where l is the length of the chromosome, is generated to indicate the position of the crossing point. In a two–crossing–point crossover operator, a chromosome is divided into three sections; a front section, a middle section, and a rear section. The front and rear sections are introduced from one parent and the middle section is introduced from the other parent to create two child chromosomes. The details of the how the optimal crossover rate was derived is beyond the scope of this paper, but described elsewhere (Arafeh and Rakha, 2005). It should be noted, however, that the findings of the study will be described in the algorithm results section.

Note that in some cases, the two child chromosomes may be infeasible chromosomes, i.e. a child chromosome may end up with a number of readers more than the required number of readers, while the other child chromosome ends up with a number of readers less than the required number of readers. In this case, a repair mechanism is applied through a single bit mutation operator, where the child chromosome with the higher number of readers undergoes a mutation operation by converting a 1 gene that is randomly selected to 0, until the desired number of readers is satisfied (i.e. solution is feasible). Similar operations are applied to other chromosomes with more readers than desired, except that the mutation operation converts 0's to 1's.

Generally mutation arbitrarily alters one or more genes of a chromosome, by a random change with a probability equal to the mutation rate. The intuition behind the mutation operator is the introduction of some extra variability into the population. This operator is applied in two stages: (a) randomly select two dissimilar genes to mutate, ***1***0111***0***10, (b) exchange the genes ***0***0111***1***10. The reason for choosing two dissimilar genes each time the operator is applied is to guarantee a legal child chromosome (i.e. a specific number of readers). The feasibility of the chromosomes is also ensured (i.e. the chromosome is guaranteed to satisfy the budgetary constraint). The mutation operator can be applied after each crossover operation or by itself as an independent operator, which can be described as a random search. We chose to apply the mutation operator after each crossover operation with probability equal 1 for reasons to be discussed in the algorithm results section.

In general, as the pool of chromosomes increases, the performance of the GA is enhanced. The total number of chromosomes created depends on the population size and the number of generations before the algorithm is terminated. Consequently, the chromosome pool can be increased either by increasing the population size or by increasing the number of generations. The reader location problem has a finite number of solutions. For example the L8R4 (8 potential reader locations with 4 readers considered) problem has $C_4^8 = 70$ possible solutions assuming that all solutions satisfy the budgetary constraint. If we were to perform a complete enumeration of the solution space, we would evaluate a total of 70 solutions. With this fact in mind, we determined that the total number of chromosomes created to solve each problem should not exceed the number of possible solution enumerations for the problem. Otherwise a direct enumeration of all solutions would offer a more efficient approach to solving the problem.

IV. EXAMPLE APPLICATIONS

For the purpose of illustrating the GA, we consider the northbound direction of Interstate-35, also known as the North Panama Freeway, in San Antonio, Texas. We consider two instant problems: L8R4 (eight possible locations with four readers) and L8R5 (eight possible locations with five readers).

Note that other problems such as the L8R3 problem is not considered because it is similar to the L8R5 problem and the L8R2 problem is trivial.

The benefit factors for the various links were computed using Equation 6 and are summarized in Table 2. Table 3 summarizes the estimated cost of installing a reader at each of the eight possible locations.

The solution space for these two problems is 70 and 56, respectively. For the L8R4 problem, the total number of chromosomes created over all the generations varied from 20 to 70 chromosomes at increments of 5 chromosomes. Similarly, the L8R5 problem was solved using a population size and number of generations that was less than the total solution space.

Table 2: Benefit factors $b^1_{pq},\ \forall\ (p,q) \in A$

	1	2	3	4	5	6	7	8
1	0.000	0.334	0.287	0.961	0.800	0.749	0.651	0.546
2	0.000	0.000	0.240	1.152	0.908	0.824	0.696	0.567
3	0.000	0.000	0.000	1.550	1.110	0.951	0.772	0.602
4	0.000	0.000	0.000	0.000	0.119	0.349	0.299	0.289
5	0.000	0.000	0.000	0.000	0.000	0.535	0.366	0.318
6	0.000	0.000	0.000	0.000	0.000	0.000	0.223	0.260
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.277
8	0.000	0.000	0.000	0.00	0.000	0.000	0.000	0.000

Table 3: Reader Cost ($1000) along I–35.

Node j	Cost C_j
1	6.32
2	9.16
3	7.00
4	3.63
5	9.11
6	1.24
7	3.68
8	5.15

To further test and analyze the proposed genetic algorithm, we constructed two additional larger test cases. Specifically, we analyzed a 15-mile section of the eastbound direction of Interstate-66 between exits 47 and 62. The constructed network has a total of 22 possible reader locations. We considered two sub-problems with 8 and 16 readers, respectively (L22R8 and L22R16). The solution space ranges from 319,770 to 74,613 for the L22R8 and L22R16 problems, respectively if the budget is assumed to be very high that it is not binding. The solution space can be reduced as the total budget is reduced.

It should be mentioned that the benefit factors (b_{ij}) for the I-66 problems were generated randomly (not derived by the INTEGRATION software) in order to study the effect of (b_{ij}) on the performance of the GA. The two problems with the new (b_{ij}'s) are identified as L22R8` and L22R16`.

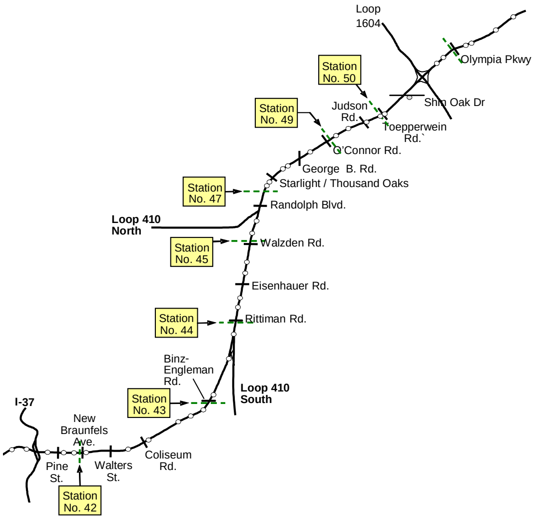


Figure 2: Study Corridor

V. ALGORITHM RESULTS

The algorithm was coded and implemented in Microsoft Visual C++ 6.0 and run on an AMD Athlon processor.

An experiment was performed to study the effect of the mutation rate, the effect of the total number of chromosomes created, the genetic operators, the chromosome selection method, and elitism, on the performance of the GA. In addition to studying the effects of the different parameters listed above on the performance of the GA and their interactions, the study also attempted to establish the optimal parameters for the GA.

A complete discussion of our findings of the experiment can be found in the literature (Arafeh and Rakha, 2005). However, it should be mentioned here that the sensitivity analysis demonstrated that minimum differences were observed in the performance of one-crossing-point versus two-crossing-point crossover operators and that the crossover operator performs better than the use of standalone mutation operators. We also found that the optimal mutation rate is 1. The optimal solutions for each of the problems are summarized in Table 4.

Table 4: Optimal Solutions for Test Problems

Problem	Optimal Reader Locations	Obj. Function
L8R4	00111100	5.0783
L8R5	10111100	7.4117
L22R8	0000111100101110000000	3.8364
L22R16	0011111111111111010100	15.8603
L22R8`	0000111100101110000000	3.8376
L22R16`	0011111111111111110000	15.8664

Figure 3 illustrates the performance of the GA versus the total number of chromosomes created presented as a percentage of the total enumeration for the L8R4 problem. The horizontal line shows the optimum (objective function value of 5.0783). The numbers on the graph indicate the percentage of times the optimum solution was achieved. The figure demonstrates that as the total number of chromosomes created increases, the performance of the algorithm improves. The effect from increasing the total number of chromosomes created is drastic until we reach 50% of total enumeration space (35 chromosomes). After this point, increasing the total number of chromosomes has a smaller effect on the performance of the GA.

Similarly, Figure 4 shows the results for the L8R5 problem. Note the similar behavior of the algorithm in solving the two problems.

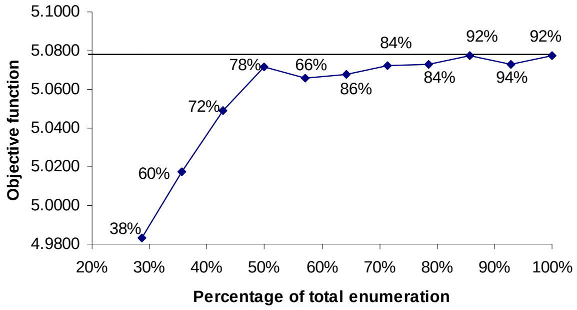


Figure 3: Optimal GA performance for L8R4 Problem.

The effectiveness of the GA is affected by the initial population. If the initial population includes chromosomes that are already close to the optimal solution, the algorithm will reach the optimal solution in fewer generations. However, in this case we increase the chance of being trapped in a local maximum. We should note here that the optimal solution (readers' location) is affected by the b_{ij} (benefit factor) values, so for different b_{ij} values, we may have a different solution for the same problem, as demonstrated in Figure 5.

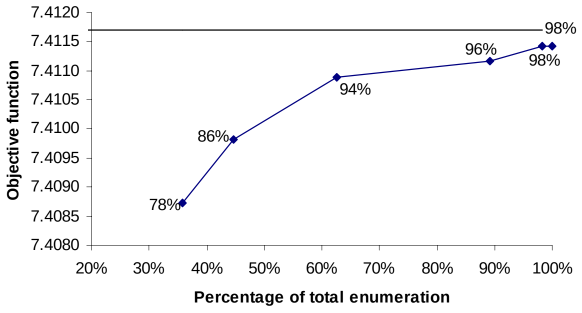
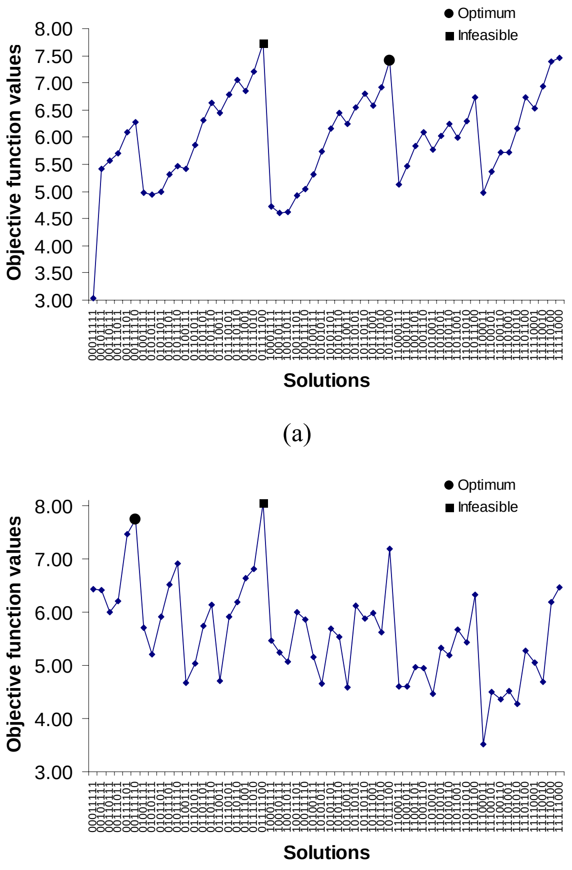


Figure 4: Optimal GA performance for L8R5 Problem.

Figure 6 presents the progress of the GA (objective function value) while solving the L8R5 problem from one generation to the next for three chromosome scenarios: 56 (100% total enumeration), 35, and 20 chromosomes (before the algorithm is stopped). The algorithm reaches the optimal solution in two generations for the 56 and 35 chromosome scenarios. When using 20 chromosomes the algorithm is trapped in a local maximum and thus is unable to find the optimum solution.



(a)

(b)

Figure 5: Objective Function Values of Solutions.

Before we conclude, we present the results for problems L22R16 and L22R8 in figures 6 and 7 respectively. It is worth noting that the algorithm's performance is stronger for larger network problems. To guarantee optimality of the solution 92% of the time for problem L8R4, we needed to create at least 60 chromosomes (86% of the total enumeration space); however, the algorithm finds the optimal solution 100% of the time with only 1% of the total enumerations for the L22R16 problem and with 0.43% of the total enumeration for the L22R8 problem. Similar results were observed for problems L22R8' and L22R16', however because of the limited space these results are not presented.

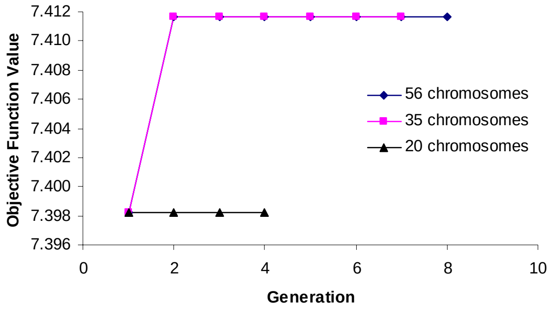


Figure 6: Variation in Objective Function as Chromosome Size Varies (L8R5 Problem)

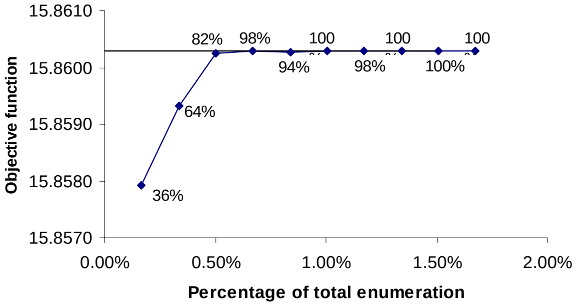


Figure 7: Variation in Objective Function for the L22R16 Problem

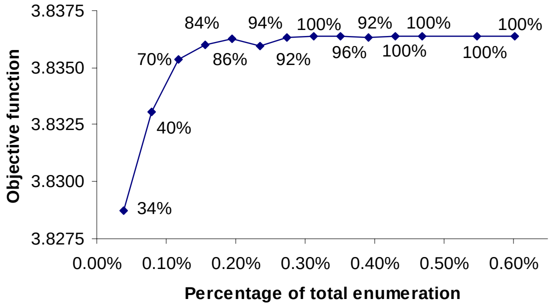


Figure 8: Variation in Objective Function for the L22R8 Problem

VI. CONCLUSIONS

The paper develops an algorithm for optimally locating surveillance technologies with an emphasis on Automatic Vehicle Identification tag readers by maximizing a travel time reliability objective function. The problem is formulated as a quadratic 0-1 optimization problem where the objective function parameters represent benefit factors that capture travel time variability along specified trips. A Genetic Algorithm is developed to solve the problem and the computational results are presented using six problems ranging from small to large networks. The study demonstrates that the algorithm is more efficient as the size of the network and problem increases. For small networks (L8R4 problem), the algorithm required at least 86% of the total enumeration space to converge to the optimum solution, while for large networks (L22R16 problem), the algorithm required only 1 percent of the total enumeration space to converge to the optimal solution. Over all, the genetic algorithm proved to be a good approach for locating automatic vehicle identification readers.

VII. REFERENCES

Arafeh M. and H. Rakha, Systematic Analysis of GA Parameters' Effect on GA Performance. Paper in Progress, 2005.

Goldberg, David E. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison Wesley Longman, Inc., 1989.

Haupt, R., and S. Haupt. *Practical Genetic Algorithms*. 2nd Edition, John Wiley & Sons, Inc., 2004.

M. Van Aerde and Assoc., Ltd., *INTEGRATION Release 2.30 for Windows*, 2004.

Rakha H., Arafeh M., Dion F., and El-Shawarby I., Trip Travel Time Variability Analysis on Series Transportation Networks, Paper submitted to IEEE ITS Congress, 2005.

Sherali H., Desai J., Rakha H., and El-Shawarby I. A Discrete Optimization Approach for Locating Automatic Vehicle Identification Readers for the Provision of Roadway Travel Times. Transportation Research Board 82nd Annual Meeting, Washington D.C., 2003.

Teodorovic D., M. Van Aerde , F. Zhu, F. Dion. Genetic Algorithms Approach to the Problem of the Automated Vehicle Identification Equipment Locations. *Journal of Advanced Transportation*, 36 (1), pp. 1-21, 2002.

VIII. ACKNOWLEDGEMENTS

This research effort was conducted under the auspices of the Intelligent Transportation Systems (ITS) Center.