

# Preliminary Highway Design with Genetic Algorithms and Geographic Information Systems

Jyh-Cherng Jong

*Civil and Hydraulic Engineering Research Center, Sinotech Engineering Consultants, Inc.,  
Taipei, Taiwan, Republic of China*

Manoj K. Jha

*Maryland Department of Transportation, State Highway Administration, Baltimore, Maryland 21202, USA*

&

Paul Schonfeld\*

*Civil Engineering Department, University of Maryland, College Park, Maryland 20742, USA*

**Abstract:** *A method that integrates geographic information systems (GIS) with genetic algorithms (GAs) for optimizing horizontal highway alignments between two given end points is presented in this article. The proposed approach can be used to optimize alignments in highly irregular geographic spaces. The resulting alignments are smooth and satisfy minimum-radius constraints, as required by highway design standards. The objective function in the proposed model considers land-acquisition cost, environmental impacts such as wetlands and flood plains, length-dependent costs (which are proportional to the alignment length), and user costs. A numerical example based on a real map is employed to demonstrate application of the proposed model to the preliminary design of horizontal alignments.*

\*To whom correspondence should be addressed. E-mail: pschon@eng.umd.edu.

## 1 INTRODUCTION

In a traditional highway design process, engineers usually start by selecting several candidate corridors through given maps and then narrow their focus to the detailed alignment design. Without mathematical models and geographic information systems (GIS), this process is very complex and time-consuming. Engineers must try different combinations of tangents and curves and evaluate in some detail each tentative horizontal alignment. Since highway design is such a complex engineering problem with enormous numbers of possible solutions, a manual design may stop far short of an optimal solution.

The application of mathematical models to highway design would significantly speed up the design process and result in a better solution.<sup>10,13</sup> Over the past three decades, three categories of models have been developed for optimizing horizontal alignments or corridors: calculus of variations,<sup>5,12,13</sup> network optimization,<sup>3,10,11,14–16</sup> and dynamic programming.<sup>4,10,14</sup> Detailed discussions on the advantages and disadvantages of these three approaches

may be found in Jong.<sup>6</sup> The calculus-of-variations method can generate a smooth alignment but requires a continuously differentiable cost function, which may not exist over different land-use patterns and geographic features such as rivers and lakes. Network-optimization approaches, including the shortest-path problem<sup>15,16</sup> and the modified transportation problem,<sup>3</sup> have well-developed solution algorithms, but the resulting alignments are very unsmooth. A dynamic programming approach may need less computer memory than a network approach but also cannot yield a smooth alignment. Both methods have difficulty in considering minimum-radius constraints explicitly. Moreover, their solution sets are only subsets of the problem search space. Thus the best solutions may be missed completely.

In addition to the preceding drawbacks, preprocessing of the cost function or link costs is needed in applying any of these three approaches. This is generally difficult, especially for the calculus-of-variations approach. Moreover, none of these published models explicitly considers user costs. According to OECD,<sup>10</sup> the net present values of vehicle operating cost discounted over 30 years range from 300 to 1000 percent of construction costs. Travel times typically have even higher values than vehicle operating costs. Thus models neglecting user costs may lead to very poor solutions.

It is desirable that a model for optimizing highway alignment should directly exploit a GIS database because most spatial information is becoming available in such computer-readable form. These include realistic shapes of land parcels and land-use patterns. The purpose of this article is to develop a model that directly uses a GIS database in optimizing smooth horizontal alignments satisfying minimum-radius constraints, as required by AASHTO.<sup>2</sup> In addition to land acquisition, environment impact, and construction costs, the proposed model also considers user costs, including vehicle operation, value of travel time, and accident costs. Since the resulting model is complex, lacks differentiability and convexity properties, and may need to search through huge numbers of local optima, a genetic algorithm (GA) with eight problem-specific operators is developed to solve it.

## 2 REPRESENTATION OF ALIGNMENTS

Let  $S(x_S, y_S)$  and  $E(x_E, y_E)$  be the start and end points of the proposed alignment, and  $\overline{SE}$  denotes the line segment connecting  $S$  and  $E$ . Jong<sup>6</sup> proved that any straight line  $L$  that is perpendicular to  $\overline{SE}$  somewhere along  $\overline{SE}$  will intersect the alignment wherever it is. If the optimal alignment is nonbacktracking (i.e., the alignment is oriented more toward the destination than the origin), then  $L$  will intersect the alignment at exactly one point. We call  $L$  a *vertical cutting line*. The idea of defining the decision

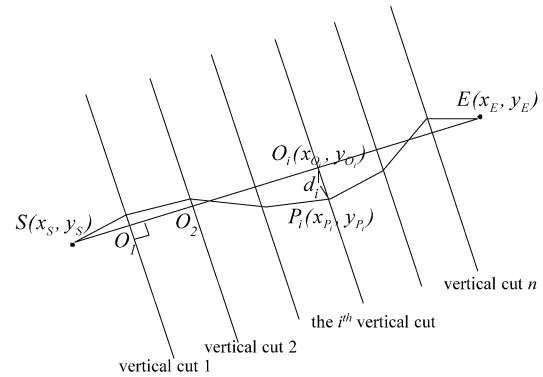


Fig. 1. Decision variables at each vertical cut.

variables to depict the horizontal alignment is based on this “cutting” concept. Suppose that we cut  $\overline{SE}$   $n$  times at equal distance between contiguous cuts, as shown in Figure 1. Then these vertical cutting lines will intersect the alignment at  $n$  points. We define such points as the points of intersections for the alignment, which have to be searched during the optimization process. Instead of directly seeking the  $XY$  coordinates of the intersection points, we simply define the decision variables as the coordinates along the vertical cutting lines in order to minimize the number of variables to be optimized.

The coordinate system at each vertical cut is not well defined yet because the direction is not specified. For maximum generality, we define the upward direction as the positive direction and the downward direction as the negative direction. The only exception occurs when the vertical cuts are orthogonal to the  $X$  axis, in which case the positive direction is defined rightward and the negative direction is leftward.

For each vertical cut, the origin is defined at the intersection point of the cut and the line segment  $\overline{SE}$ . Let  $O_i$  be the origin at the  $i$ th vertical cut,  $\forall i = 1, \dots, n$ . Then the coordinates of  $O_i$ , denoted as  $(x_{O_i}, y_{O_i})$ , are derived as

$$\begin{bmatrix} x_{O_i} \\ y_{O_i} \end{bmatrix} = \begin{bmatrix} x_S \\ y_S \end{bmatrix} + \frac{i}{n+1} \begin{bmatrix} x_E - x_S \\ y_E - y_S \end{bmatrix}. \quad (1)$$

Let  $d_i$  be the coordinate of the intersection point at the  $i$ th vertical cut. Then we must identify the upper and lower bounds of  $d_i$ ,  $\forall i = 1, \dots, n$ , because they act as constraints in the optimization model. Assume that the region of interest is in a rectangle ranging from  $[x_{\min}, x_{\max}]$  and  $[y_{\min}, y_{\max}]$ , consistently with the real coordinate system. Then there are four different cases in determining the bounds of  $d_i$  depending on the angle of the cutting line. Let  $\theta$  denote the angle between the cutting line and the  $X$  axis. Then  $\theta$  ranges from  $0^\circ$  to  $180^\circ$  and can be calculated from

$$\theta = \tan^{-1} \left( \frac{y_E - y_S}{x_E - x_S} \right) + 90^\circ \quad (2)$$

In most computer languages,  $\tan^{-1}$  always returns a value ranging from  $-90^\circ$  to  $90^\circ$ , and therefore, the range of  $\theta$  is between  $0^\circ$  and  $180^\circ$ . We now let  $d_{iU}$  and  $d_{iL}$  be the upper and lower bounds of  $d_i$ , respectively. Then  $d_{iU}$  and  $d_{iL}$  are determined according to the following four cases:

1. Case 1:  $\theta = 0^\circ$  or  $\theta = 180^\circ$ :

$$\begin{aligned} d_{iU} &= x_{\max} - x_{O_i} \\ d_{iL} &= x_{\min} - x_{O_i} \end{aligned} \quad (3a)$$

2. Case 2:  $0^\circ < \theta < 90^\circ$ :

$$\begin{aligned} d_{iU} &= \min \left\{ \frac{x_{\max} - x_{O_i}}{\cos \theta}, \frac{y_{\max} - y_{O_i}}{\sin \theta} \right\} \\ d_{iL} &= \max \left\{ \frac{x_{\min} - x_{O_i}}{\cos \theta}, \frac{y_{\min} - y_{O_i}}{\sin \theta} \right\} \end{aligned} \quad (3b)$$

3. Case 3:  $\theta = 90^\circ$ :

$$\begin{aligned} d_{iU} &= y_{\max} - y_{O_i} \\ d_{iL} &= y_{\min} - y_{O_i} \end{aligned} \quad (3c)$$

4. Case 4:  $90^\circ < \theta < 180^\circ$ :

$$\begin{aligned} d_{iU} &= \min \left\{ \frac{x_{\min} - x_{O_i}}{\cos \theta}, \frac{y_{\max} - y_{O_i}}{\sin \theta} \right\} \\ d_{iL} &= \max \left\{ \frac{x_{\max} - x_{O_i}}{\cos \theta}, \frac{y_{\min} - y_{O_i}}{\sin \theta} \right\} \end{aligned} \quad (3d)$$

The decision variables for delineating the alignment and their associated boundaries have been defined. A given set of  $d_i$  values represents a set of points on different coordinate axes. For consistency, we must convert them into the  $XY$  coordinate system. Let  $P_i$  be the intersection point located at the  $i$ th vertical cut, whose position is determined by  $d_i$ . Then the  $XY$  coordinates of  $P_i$ , denoted by  $(x_{P_i}, y_{P_i})$ , can be obtained from

$$\begin{bmatrix} x_{P_i} \\ y_{P_i} \end{bmatrix} = \begin{bmatrix} x_{O_i} \\ y_{O_i} \end{bmatrix} + d_i \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \quad (4)$$

The set of intersection points  $P_i$ ,  $i = 1, \dots, n$ , generally outlines the track of the alignment. Linking each pair of successive points with a straight-line section will generate a piecewise linear trajectory. Next, circular curves must be fitted to connect the tangent sections at the intersection points with nonzero intersection angles. An iterative computer algorithm for doing this is presented in Jong.<sup>6</sup> In order to keep the resulting alignment continuous for any set of  $P_i$  values, the radius at points with large intersection angles sometimes may be temporarily less than the minimum required by AASHTO.<sup>2</sup> In such a case, the alignment must be penalized during the evaluation process (see next section) until the radius constraint is satisfied. Note that any alignment generated by the algorithm is smooth and composed of tangent sections and circular curves. Although the

spiral transition curves are omitted, the alignment is still precise enough for evaluations needed in preliminary highway design.

Some alignment geometry characteristics such as intersection angle, point of tangency, point of curvature, and center of circular curve are important for describing the alignment and are further used in computing the corresponding alignment cost. The intersection angle at  $P_i$ , denoted by  $\Delta_i$ , is determined by the following equations (see Figure 2):

$$\Delta_i = \cos^{-1} \left( \frac{(\mathbf{P}_i - \mathbf{P}_{i-1}) \cdot (\mathbf{P}_{i+1} - \mathbf{P}_i)}{\|\mathbf{P}_i - \mathbf{P}_{i-1}\| \|\mathbf{P}_{i+1} - \mathbf{P}_i\|} \right) \quad (5)$$

Let  $T_i$  and  $C_i$  denote points of tangency and curvature at  $P_i$ , respectively. Then their coordinates  $(x_{T_i}, y_{T_i})$  and  $(x_{C_i}, y_{C_i})$  are

$$\mathbf{T}_i = \begin{bmatrix} x_{T_i} \\ y_{T_i} \end{bmatrix} = \mathbf{P}_i + \left( R_i \tan \frac{\Delta_i}{2} \right) \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{\|\mathbf{P}_{i+1} - \mathbf{P}_i\|} \quad (6)$$

$$\mathbf{C}_i = \begin{bmatrix} x_{C_i} \\ y_{C_i} \end{bmatrix} = \mathbf{P}_i + \left( R_i \tan \frac{\Delta_i}{2} \right) \frac{\mathbf{P}_{i-1} - \mathbf{P}_i}{\|\mathbf{P}_{i-1} - \mathbf{P}_i\|} \quad (7)$$

where  $R_i$  is the circular curve radius pertaining to  $P_i$ . The center of the circular curve, denoted by  $\delta_i(x_{\delta_i}, y_{\delta_i})$ , can be located in many ways. The simplest one may be vector analysis. Let  $M_i(x_{M_i}, y_{M_i})$  be the middle point of the line segment between  $C_i$  and  $T_i$ , as shown in Figure 3. Then it can be verified by trigonometry that  $M_i$  is also located on the line segment connecting  $P_i$  and  $\delta_i$ . Accordingly,  $\delta_i(x_{\delta_i}, y_{\delta_i})$  can be derived by extending the vector  $\mathbf{M}_i - \mathbf{P}_i$  from point  $P_i$  to point  $\delta_i$ . In mathematical form, we obtain

$$\mathbf{M}_i = \begin{bmatrix} x_{M_i} \\ y_{M_i} \end{bmatrix} = \frac{1}{2}(\mathbf{C}_i + \mathbf{T}_i) = \begin{bmatrix} (x_{C_i} + x_{T_i})/2 \\ (y_{C_i} + y_{T_i})/2 \end{bmatrix} \quad (8)$$

$$\delta_i = \begin{bmatrix} x_{\delta_i} \\ y_{\delta_i} \end{bmatrix} = \mathbf{P}_i + R_i \sec \frac{\Delta_i}{2} \frac{\mathbf{M}_i - \mathbf{P}_i}{\|\mathbf{M}_i - \mathbf{P}_i\|} \quad (9)$$

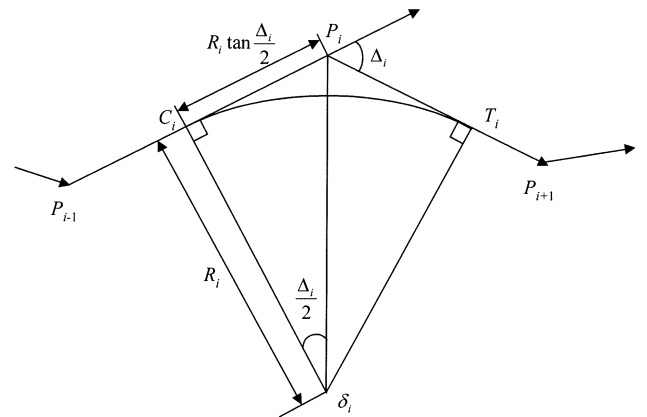


Fig. 2. The geometric specification of a circular curve.

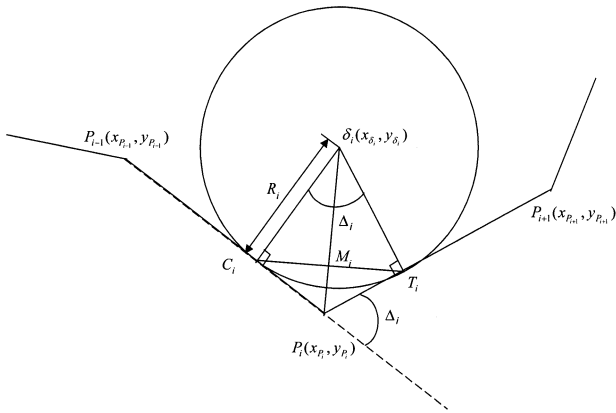


Fig. 3. The geometric relations among  $C_i$ ,  $T_i$ ,  $\delta_i$ , and  $M_i$ .

### 3 COST FUNCTIONS

There are many cost items associated with highway transportation. For mathematical programming, it is better to categorize these cost items according to their relations to the alignment geometry so that they can be easily formulated as functions of optimizable decision variables. Jong and Schonfeld<sup>7</sup> classified highway costs into location-dependent cost, length-dependent cost, VKT (vehicle-kilometer traveled)-dependent cost, and user costs. We follow the same classifications to formulate the cost functions, each of which is discussed below.

#### 3.1 Location-dependent cost

The entire region of interest consists of various land parcels, including private properties, wetlands, and flood plains. The real land parcels have irregular shapes, unlike those in network optimization or dynamic programming models, where the entire region is roughly represented by nodes or grids. To be consistent with the real map in calculating location-dependent cost, an existing GIS database is employed for the proposed model. The minimum segments of land parcels are obtained by overlapping different levels of GIS database such as private properties, wetlands, flood plains, and rivers. Each of the parcels is associated with a location-dependent unit cost for building the highway. In addition to right-of-way cost, the location-dependent unit cost may include environmental impact. For example, if a parcel is a wetland or flood plain, in which the alignment is prohibited, then the location-dependent unit cost for that parcel is set to a very high value so that any alternative passing through the parcel will be unfeasibly costly.

The total location-dependent cost for a highway alignment is computed by summing up the location-dependent cost for all sections of parcels needed for that alignment. Assume that the alignment goes through  $m$  parcels listed in order, and let  $K_j$  be the location-dependent unit cost for

the  $j$ th parcel. Then the total location-dependent cost for the alignment is

$$C_N = \sum_{j=1}^m K_j A_j \quad (10)$$

where  $A_j$  is the area covered by the highway in parcel  $j$ . The computation of Equation (10) is not trivial because identifying the parcels covered by the highway is difficult from a programming point of view. Jong<sup>6</sup> developed a computer algorithm for doing this, but only for rectangular parcels. Fortunately, this task can be performed easily with GIS software (e.g., ArcView GIS). For any given alignment, we can create a buffer surrounding the highway and then superimpose the highway boundary on top of the land parcels. By clipping the boundaries of the highway and the land parcels, we can calculate the area covered by the highway at each parcel and then apply Equation (10) to compute the total location-dependent cost.

#### 3.2 Length-dependent cost

To compute the length-dependent cost for a highway, we simply multiply the highway length by the unit length-dependent cost. Recall that any alignment generated by the proposed model is composed of tangents and circular curves in which  $T_i$  and  $C_{i+1}$  are linked by a straight-line section, whereas  $C_i$  and  $T_i$  are connected by a circular curve with radius  $R_i$ . For notational convenience, we denote  $T_0 = S$  and  $C_{n+1} = E$ . Then the total length of the alignment, denoted by  $L_n$ , is determined by

$$L_n = \sum_{i=0}^n \sqrt{(x_{T_i} - x_{C_{i+1}})^2 + (y_{T_i} - y_{C_{i+1}})^2} + \sum_{i=1}^n R_i \Delta_i \quad (11)$$

The unit length-dependent cost may include unit construction cost, unit maintenance cost, and unit environmental cost. The construction cost per unit length can be broken down into detailed items, such as fences, guardrails, bridges, and pavement cost, provided the information is available. Note that the cost of bridges will depend on several factors, such as the length and type of bridges selected. These factors may depend on detailed geographic characteristics such as soil type, land-use characteristics in the vicinity, and topography. Although the bridge costs are not yet included in the length-dependent costs, they can be included if a detailed geographic database is available. Some environmental impacts such as air and noise pollution are VKT-dependent.<sup>7</sup> If the projected traffic demand is known (since it is usually forecast in the planning stage), such environmental cost can be further transformed to a length-dependent cost. Let  $AADT$  (average annual daily traffic) be the projected two-way traffic volume, which we

assume will increase every year at a rate  $r_t$  throughout the analysis period  $n_y$ . Then the net present value of the environmental cost per unit length is

$$K_{E.L} = \frac{1}{5280} K_{E.V} \cdot 365 \cdot AADT \cdot \sum_{k=1}^{n_y} \left[ \frac{(1+r_t)}{(1+\rho)} \right]^k \quad (12)$$

where  $K_{E.L}$  = the net present value of unit length-dependent cost for environment impact (\$/m)

$K_{E.V}$  = the unit environmental cost per VKT (\$/VKT)

$\rho$  = the assumed interest rate (decimal fraction)

In the preceding equation, the unit environmental cost per VKT is derived by summing up the unit costs for different environmental impacts such as air, noise, and water pollution, oil extraction and distribution, land-use impact, and chemical waste disposal.<sup>7</sup> The computation of Equation (12) is slightly laborious. A different form incorporating smoothly compounded traffic growth at the same annual rate of  $r_t$  is given by<sup>1</sup>

$$K_{E.L} = \frac{1}{5280} K_{E.V} \cdot 365 \cdot AADT \cdot \frac{e^{(r_t-\rho)n_y} - 1}{r_t - \rho} \quad (13)$$

The comprehensive unit length-dependent cost is then

$$K_L = K_C + K_M + K_{E.L} \quad (14)$$

where  $K_L$  = the total unit length-dependent cost (\$/m)

$K_C$  = the construction cost per unit length (\$/m)

$K_M$  = the maintenance cost per unit length (\$/m)

Finally the total length-dependent cost  $C_L$  is

$$C_L = K_L L_n \quad (15)$$

### 3.3 User cost

The major components of user costs usually include vehicle operating costs, the value of travel time, and traffic accident costs. The computation of user costs is less direct because their relations to highway design features are not explicit. In general, the relations were calibrated by statistical analysis. AASHTO<sup>2</sup> provides several figures and tables for estimating user costs of highway and bus transit improvements. This manual has been quoted widely in various studies and served as a primary reference in estimating user costs. However, since those figures and tables are cumbersome to computerize, Jong and Schonfeld<sup>7</sup> derived algebraic functional forms for estimating user costs. They started by estimating average running speeds for different analysis periods (peak or off-peak) and traffic directions (prevalent or nonprevalent), based on alignment geometry

such as curvature, gradient, and the length of the alignment. The average running speeds were further used in estimating fuel-consumption costs and value of travel time for different vehicle types such as medium car, two-axle single-unit truck, and 3-S2 diesel truck. Accident costs were estimated from alignment geometry and traffic conditions. Also, for those alignments which violated minimum radius constraints, a penalty cost was added to the accident costs according to the magnitude of the violation.

## 4 SOLUTION ALGORITHM

Recall that the decision variables in the proposed model are the coordinates  $d_i$  along each vertical cutting line. The objective function is the summation of various cost items. Hence the final model can be formulated as follows:

$$\text{Minimize}_{d_1, d_2, \dots, d_n} C_T = C_N + C_L + C_U \quad (16)$$

$$\text{subject to } d_{iL} \leq d_i \leq d_{iU} \quad \text{for all } i = 1, \dots, n \quad (17)$$

where  $C_U$  is the user cost, including fuel consumption, value of travel time, and accident costs. Since alignments in the proposed model are generated by an iterative algorithm,<sup>6</sup> Equation (16) cannot be expressed directly as a function of decision variables  $d_i$ . Consequently, the problem lacks convexity and differentiability properties, which are required by many gradient-based search methods such as the Newton or steepest-descent methods.

In order to solve the proposed model, a genetic algorithm (GA) was developed. GAs are motivated by the principles of natural selection and "survival of the fittest."<sup>8</sup> They start with a set of possible solutions to the problem, called the *initial population*. Each individual in the population is encoded into a string representation called a *chromosome*. At each generation, the individuals are selected to reproduce offspring based on their fitness to the problem. (In our model, the *fitness value* of a solution is simply defined as its corresponding cost value.) After several generations, the most adapted individuals should survive, whereas poor solutions should die off, and the population will finally converge to an optimal solution. The performance of GAs is ascribed not only to the natural selection process but also to the genetic operators specified by us through which the new offspring are generated from their parents. For notational convenience, we now denote a chromosome by  $\Lambda$  and an individual gene by  $\lambda$  subscripted by its location. Since GAs are probabilistic and involve random number generations, the notation  $r_c[b_L, b_U]$  is used to represent a random number generated from a continuous uniform distribution bounded within the interval  $[b_L, b_U]$ , where the subscripts  $L$  and  $U$  denote the lower and upper bounds. Similarly,  $r_d[b_L, b_U]$  denotes a random number generated from a discrete uniform distribution.

#### 4.1 Genetic encoding

Instead of using binary digits to represent a solution, a floating-point encoding scheme is employed for the proposed GA. The chromosome can be defined easily as the set of decision variables, each of which is a continuous real number confined within its associated boundaries.

$$\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_n] = [d_1, d_2, \dots, d_n] \quad (18)$$

In the preceding equation,  $\lambda_i$  is bounded within the interval  $[d_{iL}, d_{iU}]$ , where  $d_{iL}$  and  $d_{iU}$  are the corresponding lower and upper bounds defined in Equation (3). The boundaries play an important role in generating the initial population and developing genetic operators.

#### 4.2 Initial population

In order to keep the gene pool as large as possible so that the entire search space can be explored, the initial population should be randomly generated. For optimizing highway alignment, however, it is suggested that certain chromosomes that represent a straight alignment be included in the initial population because the alignment has minimal length-dependent and user costs. If somehow an engineer has some initial ideas or guesses about good solutions, they also may be included. Without loss of generality, we assume here that no prior knowledge about the solution is available. Then the population can be generated as follows:

1. Intersection points located on the straight line connecting the start and end points. In this case, the set of intersection points forms a straight alignment. This may not be a good solution, but it may contain some useful information about the problem. In fact, except for location-dependent costs, all other cost items may be reduced by straighter and shorter alignments. Recall that in this case the intersection points lie exactly at the origins of their associated vertical cuts. The chromosome is thus represented by

$$\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_n] = [0, 0, \dots, 0] \quad (19)$$

2. Intersection points located randomly on the vertical cuts. In this case, each gene of the chromosome is randomly generated from a continuous uniform distribution within the corresponding boundaries:

$$\lambda_i = r_c[d_{iL}, d_{iU}] \quad \forall i = 1, \dots, n \quad (20)$$

#### 4.3 Selection/replacement scheme

In our proposed GA, the selection/replacement procedure (alternatively called *sampling mechanism*) has several important aspects. It is characterized as an ordinal-based selection with regular sampling space, generational replacement, and an elitism model. The scheme is modified from the two-step selection algorithm introduced by Michalewicz.<sup>8</sup>

#### 4.4 Genetic operators

It is important to mention that the relation between the genes in a chromosome and the corresponding alignment is indirect. Each gene in a chromosome is an intersection point in a two-dimensional space, where the alignment is obtained by fitting circular curves at each intersection point. Moreover, the genes are not independent of each other because whenever the location of an intersection point is changed, the alignment configuration at other intersection points may change as well due to the curve-fitting process.

The preceding properties make it more difficult to develop appropriate genetic operators. Moreover, the genes in a chromosome are encoded as floating-point (i.e., real) numbers. For these reasons, conventional operators do not work well in this situation. To facilitate the search, we must devise problem-specific genetic operators. Eight kinds of operators are proposed and discussed below.

1. *Uniform mutation.* For most applications of GAs, where the genes are encoded as real numbers, the uniform mutation is performed by randomly selecting a gene and replacing its value with a randomly selected real number. Let the chromosome to be mutated be  $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]$  and the  $k$ th gene be selected to apply the operator (i.e.,  $k = r_d[1, n]$ ). Then  $\lambda_k$  will be replaced by  $\lambda'_k = r_c[d_{kL}, d_{kU}]$ .

Recall that mutations are often seen as background operators to maintain genetic diversity in the population. Unfortunately, the simple mutation operator does not work as expected in our proposed model. To illustrate this, let us consider the situation shown in Figure 4, in which the corresponding alignment of the chromosome passes through a high cost field. Suppose that  $\lambda_k$  is selected to mutate, and the resulting

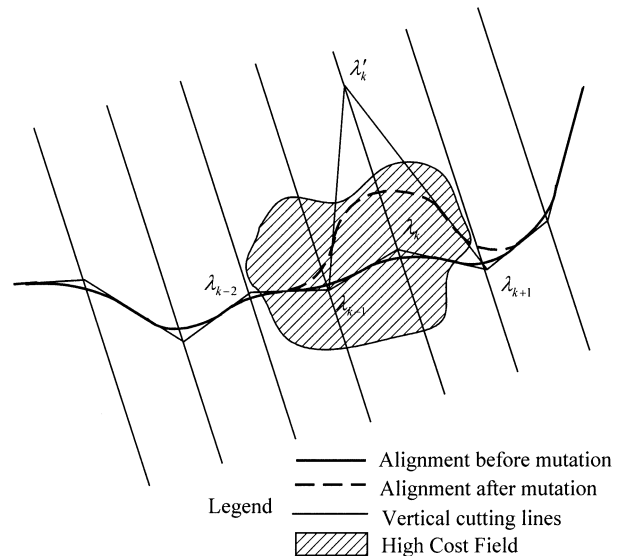


Fig. 4. An example of a failed mutation.

new value of the gene is  $\lambda'_k$ . The figure shows that the new alignment is even worse than the original one. We may expect that next time  $\lambda_{k-2}$ ,  $\lambda_{k-1}$ , and  $\lambda_{k+1}$  will be selected to undergo mutation operators so that the corresponding alignment can jump out of the high cost field. However, before these adjacent genes are selected to mutate, the chromosome may die off because the new chromosome is worse. In fact, as the GA evolves, the final solution will be the one passing through the high cost field at minimal distance. This is obviously just a local rather than a global optimum.

A solution to this problem is to design the operator in such a way that the corresponding alignment of the chromosome has the potential to jump out of the high cost field at once. The operator is then modified by generating another two independent loci  $i$  and  $j$  in addition to  $k$ , where  $i = r_d[0, k - 1]$  and  $j = r_d[k + 1, n + 1]$ , such that the corresponding intersection points of  $\lambda_i$  and  $\lambda_k$  are connected by straight-line segments, and so are the corresponding points of  $\lambda_k$  and  $\lambda_j$ . Note that here locus = 0 represents the starting point  $S$  of the alignment and locus =  $n + 1$  denotes the end point  $E$ . We will name this procedure *elimination* because it is analogous to eliminating the curves between the  $i$ th and  $k$ th intersection points as well as those between the  $k$ th and  $j$ th intersection points.

2. *Straight mutation.* This operator is proposed to straighten the alignment between two randomly selected intersection points. The idea behind this operator is that a straight alignment may result in lower total cost because most cost components are minimized by a more direct and shorter alignment. Let the chromosome to be mutated be  $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]$ . We randomly generate two independent loci  $i$  and  $j$ , where  $i = r_d[0, n + 1]$  and  $j = r_d[0, n + 1]$ ,  $i \neq j$  and  $i < j$ . Then the values of the intermediate genes between the  $i$ th and  $j$ th genes will be replaced by

$$\lambda'_l = \lambda_i + (l - i) \left[ \frac{(\lambda_j - \lambda_i)}{j - i} \right] \quad (21)$$

for all  $l = i + 1, \dots, j - 1$

In the preceding equation, if  $i = 0$  (which represents the start point  $S$ ), then we set  $\lambda_i = 0$ . Similarly, if  $j = n + 1$  (which denotes the end point  $E$ ), then  $\lambda_j$  is also set to 0.

3. *Nonuniform mutation.* Nonuniform mutation was introduced by Michalewicz.<sup>8</sup> The operator is designed for fine-tuning the solution. At early generations, the mutation range is relatively large, while at latter generations, it is tightened for local refinement. For a given parent  $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]$ , in which the  $k$ th gene ( $k = r_d[1, n]$ ) is selected for mutation. We first

generate a binary random digit  $r_d[0, 1]$ . Then  $\lambda_k$  will be replaced according to the following rules:

$$\text{If } r_d[0, 1] = 0, \quad \text{then } \lambda'_k = \lambda_k - f(t, \lambda_k - d_{kL}) \quad (22a)$$

$$\text{If } r_d[0, 1] = 1, \quad \text{then } \lambda'_k = \lambda_k + f(t, d_{kU} - \lambda_k) \quad (22b)$$

In the preceding equations,  $t$  is the current generation number, while  $d_{kL}$  and  $d_{kU}$  are the corresponding lower and upper bounds of the  $k$ th gene as defined in Equation (3). The function  $f(t, y)$  in Equation (22) returns a random value in the range  $[0, y]$  such that the probability of  $f(t, y)$  approaching 0 increases as  $t$  increases.<sup>8</sup> This property causes the operator to search the space uniformly at initial generations and very locally at later stages. In order to prevent the solutions from sticking at a local optimum after a gene is mutated, a curve-elimination procedure such as that used for uniform mutation is applied.

4. *Whole nonuniform mutation.* This operator applies the nonuniform mutation operator to each of the genes in a given chromosome in a randomly generated sequence. The resulting offspring will be totally different from its parent. In other words, the operator forces a chromosome jump to another location of the search space to maintain the diversity of genes in the population. In the early stages of the evolution, the shift in the search space is significant. However, in later generations, the operation perturbs all genes of a chromosome only around the vicinity of the corresponding solution for local refinement. The concept of the operator is similar to some conventional optimization techniques, such as hill-climbing methods, in which the step size is larger in earlier iterations and decreases in later iterations.

5. *Simple crossover.* This operator is analogous to the one-point crossover in binary implementations of GAs. Let two parents  $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{in}]$  and  $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, \dots, \lambda_{jn}]$  be crossed after a randomly generated position  $k$ , where  $k = r_d[1, n]$ . Then the resulting offspring are

$$\Lambda'_i = [\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{ik}, \lambda_{j(k+1)}, \dots, \lambda_{jn}] \quad (23a)$$

$$\Lambda'_j = [\lambda_{j1}, \lambda_{j2}, \dots, \lambda_{jk}, \lambda_{i(k+1)}, \dots, \lambda_{in}] \quad (23b)$$

6. *Two-point crossover.* The idea of this operator is to exchange the genes between two randomly generated positions  $k$  and  $l$  for two given parents  $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{in}]$  and  $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, \dots, \lambda_{jn}]$ , where  $k = r_d[1, n]$ ,  $l = r_d[1, n]$ ,  $k \neq l$  and  $k < l$ . The resulting offspring are

$$\Lambda'_i = [\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{ik}, \lambda_{j(k+1)}, \dots, \lambda_{jl}, \lambda_{i(l+1)}, \dots, \lambda_{in}] \quad (24a)$$

$$\Lambda'_j = [\lambda_{j1}, \lambda_{j2}, \dots, \lambda_{jk}, \lambda_{i(k+1)}, \dots, \lambda_{il}, \lambda_{j(l+1)}, \dots, \lambda_{jn}] \quad (24b)$$

7. *Arithmetic crossover*. The basic concept of such an operator is borrowed from the definition of a convex set: Any linear combination of two points in a convex set also will fall into the set. Therefore, if the feasible region of a constrained optimization problem is a convex set, then any linear combination of two feasible points will be feasible as well. Let  $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{in}]$  and  $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, \dots, \lambda_{jn}]$  be two parents to be crossed. Then, based on the same concept, the resulting offspring are defined as a linear combination of two parent chromosomes, which guarantees the offspring is always feasible:

$$\Lambda'_i = \omega \Lambda_i + (1 - \omega) \Lambda_j \quad (25a)$$

$$\Lambda'_j = \omega \Lambda_j + (1 - \omega) \Lambda_i \quad (25b)$$

where  $\omega = r_c[0, 1]$ .

8. *Heuristic crossover*. The operator is a unique crossover for the following reasons: (a) it uses values of the fitness function in determining the direction of the search, (b) it produces only one offspring, and (c) the resulting offspring may not be feasible. Let  $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{in}]$  and  $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, \dots, \lambda_{jn}]$  be two parents subjected to this operator, where we assume that  $C_T(\Lambda_i) \leq C_T(\Lambda_j)$  (i.e.,  $\Lambda_i$  is better or at least as good as  $\Lambda_j$ ). Then the operator generates a single offspring  $\Lambda'$  according to the following rule:

$$\Lambda' = \omega(\Lambda_i - \Lambda_j) + \Lambda_i \quad (26)$$

where  $\omega = r_c[0, 1]$ . It is possible for this operator to generate an offspring that is not feasible. In such a case, another random number is generated and another offspring is created. If after a certain number of attempts (user defined) no new offspring can meet the boundary constraints defined in Equation (17), the operator gives up and returns  $\Lambda_i$  as the offspring.

## 5 CASE STUDY

A real map from Cecil County, Maryland, is employed to demonstrate the proposed model and algorithm. While the model was capable of handling cases with complex topography and land use, it was difficult to obtain such cases from real databases for Maryland. The Maryland State Highway Administration (MDSHA) has developed a GIS database that stores various spatial information such as land

parcel, boundaries, land cost, wetlands, flood plains, soil conditions, and topography for the entire state. The coordinates of land parcels and costs are stored in an ArcView (trademark of Environmental Systems Research Institute, Inc.) file format called MD Property View. To optimize a highway alignment, the MD Property View is superimposed on the map in which rivers, wetlands, and flood plains are displayed. Figure 5 shows the region over which the alignment of a proposed highway is being optimized. The  $X$  and  $Y$  coordinates of the region range from 340889 to 346764 and from 181058 to 185189, respectively. The darker shades of land parcel in the map represents higher location-dependent unit cost for building a highway. The Great Bohemia Creek runs eastward near the lower end of the map, where location-dependent unit costs are relatively high because expensive bridges would be needed. In addition, some wetlands and flood plains in which the alignment is prohibited are also displayed in a darker shade to represent high environmental impacts.

MD Route 213 crosses the Great Bohemia Creek at the lower left corner of the map (Bohemia River Bridge) and intersects MD Route 310 about 3 km north of the river. On the upper right corner of the map, MD Route 310 intersects MD Route 342. Any vehicles traveling between the Bohemia River Bridge and MD Route 342 must go through the intersection of MD Route 213 and MD Route 310. Assume that the traffic is very heavy during peak hours and causes serious congestion at this intersection. Thus another road is considered to divert traffic between Bohemia River Bridge and the intersection of MD Routes 310 and 342. For this purpose, we use the proposed model and algorithm to optimize the alignment of the new highway. It can be seen from Figure 5 that the straight alignment connecting the start and end points of the highway will go through several high-cost land parcels. Therefore, the optimal alignment must detour from the straight line.

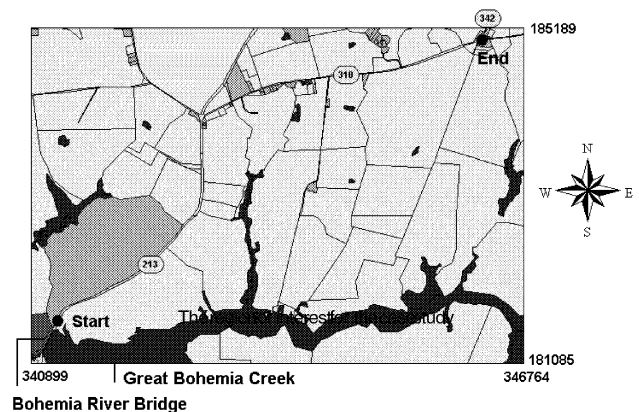


Fig. 5. The region of interest for the case study.

Here we want to run the proposed GA to optimize the alignment and assess the solution. We assume that no particular solution based on engineers' judgments is available for the initial population. Thus the entire initial population is automatically generated from the program. For this problem, the total number of generations is set to 500, and the number of intersection points is 10 because the land-use patterns are not too complex. At each generation, each of the eight genetic operators generates four new offspring, for a total of 32. To evaluate the location-dependent cost for an offspring, the coordinates along the alignment are passed to ArcView GIS, and the result is sent back to the GA optimizer. The communications between ArcView GIS and the proposed GA are bidirectional and programmed so that the process is fully automated.

The optimal horizontal alignment obtained from the program is shown in Figure 6. The alignment is quite straight and smooth. It also satisfies the minimum-radius requirement and avoids high environmental impact areas. The total alignment length is 6050 m with total cost equal to  $\$2.99 \times 10^7$ . The figure shows that a portion of the alignment almost coincides with the existing MD Route 213, whereas the other section just bypasses the flood plain along the branch of the Great Bohemia Creek and a wetland on the south of MD Route 310. Based on the results, it seems appropriate to add two lanes to existing MD Route 213 until it diverges from the new two-lane alignment.

For selecting a highway alignment between two points, there are huge numbers of possible solutions and local optima, which make it impractical to ascertain an exact optimal solution. Since we do not know the exact optimal solution to the problem, we design an experiment to statistically test the goodness of the solution. The experiment is initiated by generating a random sample of solutions to the problem and then fitting a distribution to the objective values for the random sample. The fitness of the distribution can be checked with the chi-square or K-S test (Neter et al.<sup>9</sup>). Since the sample is randomly generated, the fitted distribution should be able to reflect the actual distribution

of the objective value for the real population. Based on this distribution, we can compare the solution found by the proposed algorithm and calculate its cumulative probability in the distribution. A lower cumulative probability indicates a better solution.

Following the experiment, we first create a random sample of 20,000 observations, whereas in the GA approach, only  $32 \times 500 = 16,000$  offspring are generated during the search. The objective value of the best solution in this sample is  $\$4.35 \times 10^7$ , whereas the worst solution yields an objective value of  $\$1.36 \times 10^8$ . The sample mean is  $\$7.26 \times 10^7$ , and the standard deviation is  $\$1.3 \times 10^7$ . After trying different distributions and using the chi-square test, it is found that the following truncated log-normal distribution is the best fitting one:

$$4.35 \times 10^7 + \log \text{ normal } (2.95 \times 10^7, 1.54 \times 10^7) \quad (27)$$

The preceding distribution shows that its best (i.e., lowest) objective value is  $4.35 \times 10^7$ , which is 1.45 times higher than the solution ( $2.99 \times 10^7$ ) found by the proposed algorithm. Figure 7 shows these values on the distribution diagram. It indicates that the objective value ( $3.18 \times 10^7$ ) at the second generation in the GA approach is lower than the lower bound ( $4.35 \times 10^7$ ) of the fitted distribution. At that time, only  $32 \times 2 = 64$  offspring have been generated, whereas in the random sample, 20,000 solutions have been created. Of course, the final solution ( $2.99 \times 10^7$ ) dominates all possible values in the distribution. This analysis indicates that the solution found by the proposed algorithm is remarkably good when compared with possible random solutions to the problem.

The transition of the objective value in the GA approach is plotted in Figure 8. It shows that the objective value drops drastically at the first few generations. The improvements in the objective value then slow down until the 135th generation, when the objective value drops sharply again. After that, the objective value steadily converges to the final value.

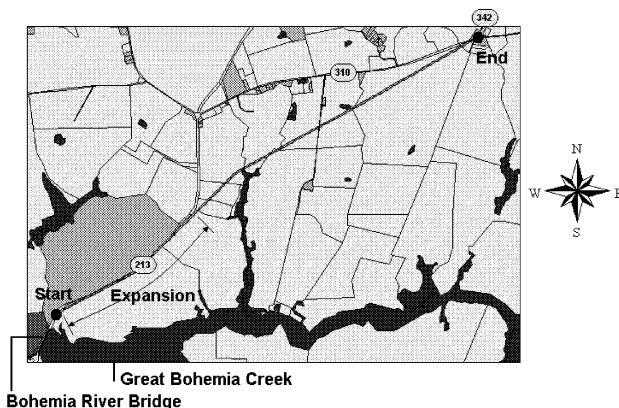


Fig. 6. The optimal alignment for the case study.

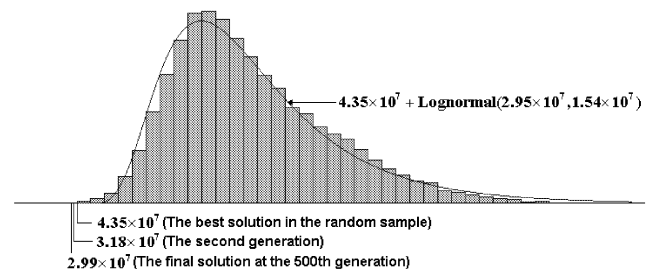


Fig. 7. The fitted distribution of the objective value for the random sample.

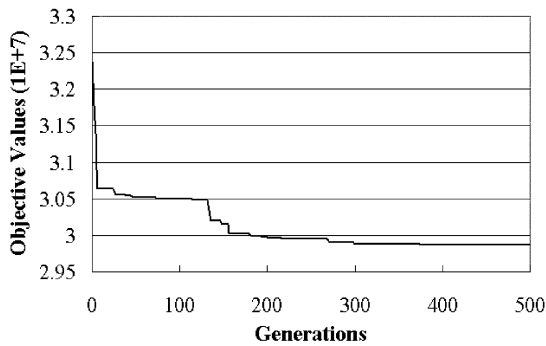


Fig. 8. The transition of objective values during the search.

## 6 CONCLUSIONS

The proposed model and GA for optimizing highway horizontal alignments has several advantages. First, the model can generate smooth alignments based on highway design standards, unlike some previous approaches, such as network optimization and dynamic programming. Second, the GA approach can optimize very complex cost functions including user costs, which have been ignored in many existing models. Third, the model directly exploits the information in a GIS database, which reduces data preprocessing time and allows the model to search through realistic and highly irregular spatial data.

There are basically two ways to apply the proposed model and GA. The first one is to run the program and let it search for a very good alignment, while the other is to use the program as an evaluation and fine-tuning tool. Since humans can easily tire in designing different alternatives, engineers may specify their design solutions in the initial population of the proposed GA and let the program improve their design. The case study shows that application of the proposed model and GA is quite successful and that the resulting alignment is reasonably good.

This work is an attempt to combine GA and GIS in optimizing highway alignments. Although the current model possesses many good features, it leaves much room for future improvements. First, for very irregular land-use patterns or terrain, it is possible that the alignment occasionally may twist backward. Jong<sup>6</sup> called such alignments “backtracking” alignments and developed a different approach to model them. Second, a more complex model that simultaneously optimizes both horizontal and vertical alignments is desirable. Since elevation information is available in a GIS database, extracting such information to estimate earthwork cost based on a proposed vertical profile would be achievable. Such extensions are recommended for future research.

## ACKNOWLEDGMENTS

We express our thanks to Mr. Kirk McClelland and Ms Norie Calvert of the Maryland State Highway Administration for their suggestions. This work was funded by the Highway Design Division of the Maryland State Highway Administration.

## REFERENCES

1. AASHTO, *A Manual on User Benefit Analysis of Highway and Bus-Transit Improvements*, American Association of State Highway and Transportation Officials, Washington, 1977.
2. AASHTO, *A Policy on Geometric Design of Highways and Streets*, American Association of State Highway and Transportation Officials, Washington, 1994.
3. Athanassoulis, G. C. & Calogero, V., Optimal location of a new highway from A to B—A computer technique for route planning, *PTRC Seminar Proceedings on Cost Models and Optimisation in Highways* (session L9), London, 1973.
4. Hogan, J. D., Experience with OPTLOC: Optimum location of highways by computer, *PTRC Seminar Proceedings on Cost Models and Optimisation in Highways* (session L10), London, 1973.
5. Howard, B. E., Bramnick, Z. & Shaw, J. F. B., Optimum curvature principle in highway routing, *Journal of the Highway Division ASCE*, **94** (HW1) (1968), 61–82.
6. Jong, J. C., Optimizing highway alignments with genetic algorithms, Ph.D. dissertation, University of Maryland, College Park, 1998.
7. Jong, J. C. & Schonfeld, P., Cost functions for optimizing highway alignments, *Transportation Research Record 1659*, Washington, 1999, pp. 58–67.
8. Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed., Springer-Verlag, New York, 1996.
9. Neter, J., Wasserman, W. & Whitmore, G. A., *Applied Statistics*, 2nd ed., Allyn and Bacon, Boston, 1982.
10. OECD, *Optimisation of Road Alignment by the Use of Computers*, Organisation of Economic Cooperation and Development, Paris, 1973.
11. Parker, N. A., Rural highway route corridor selection, *Transportation Planning and Technology*, **3** (1977), 247–56.
12. Shaw, J. F. B. & Howard, B. E., Comparison of two integration methods in transportation routing, *Transportation Research Record 806*, Washington, 1981, pp. 8–13.
13. Shaw, J. F. B. & Howard, B. E., Expressway route optimization by OCP, *Transportation Engineering Journal of ASCE*, **108** (TE3) (1982), 227–43.
14. Trietsch, D., A family of methods for preliminary highway alignment, *Transportation Science*, **21** (1) (1987), 17–25.
15. Turner, A. K., A decade of experience in computer aided route selection, *Photogrammetric Engineering and Remote Sensing*, **44** (1978), 1561–76.
16. Turner, A. K. & Miles, R. D., A computer-assisted method of regional route location, *Highway Research Record*, **348** (1971), 1–15.

## NOTATION

The following symbols are used in this article:

$A_j$	= the area covered by the highway in parcel ( $m^2$ )	$L_n$	= total alignment length (m)
$AADT$	= average annual daily traffic (vehicles/day)	$m$	= total number of land parcels covered by the highway (integer)
$C_L$	= total length-dependent cost (\$)	$M_i(x_{M_i}, y_{M_i})$	= the middle point of the line segment connecting $T_i$ and $C_i$
$C_N$	= total location-dependent cost (\$)	$n$	= number of intersection points (integer)
$C_T$	= total cost of the alignment (\$)	$n_y$	= total analysis period (years)
$C_U$	= total user cost (\$)	$O_i(x_{O_i}, y_{O_i})$	= the origin point of the $i$ th vertical cutting line
$C_i(x_{C_i}, y_{C_i})$	= point of curvature pertaining to $P_i$	$P_i(x_{P_i}, y_{P_i})$	= the $i$ th intersection point
$d_i$	= the coordinate of $P_i$ on the $i$ th vertical cutting line (real)	$R_i$	= the radius of circular curve pertaining to $P_i$
$d_{iU}$	= the upper bound of $d_i$ (real)	$r_i$	= annual growth rate of traffic (decimal)
$d_{iL}$	= the lower bound of $d_i$ (real)	$r_c[b_L, b_U]$	= a continuous random number generated from $[b_L, b_U]$
$E(x_E, y_E)$	= the end point of the alignment	$r_d[b_L, b_U]$	= a discrete random number generated from $[b_L, b_U]$
$f(t, y)$	= function that returns a value between $[0, y]$ for nonuniform mutation	$S(x_S, y_S)$	= the start point of the alignment
$i$	= index for intersection points, circular curves, or genes	$t$	= current generation number (integer)
$j$	= index for land parcels or genes	$T_i(x_{T_i}, y_{T_i})$	= point of tangency pertaining to $P_i$
$k$	= index for genes	$X$	= abscissa in Cartesian coordinate system
$K_C$	= the construction cost per unit length (\$/m)	$x_{max}$	= the maximal abscissa of the region of interest (real)
$K_j$	= the unit location-dependent cost of the $j$ th land parcel where the alignment passes ( $$/m2)$	$x_{min}$	= the minimal abscissa of the region of interest (real)
$K_L$	= the total unit length-dependent cost (\$/m)	$Y$	= ordinate in Cartesian coordinate system
$K_M$	= the maintenance cost per unit length (\$/m)	$y$	= parameter used in $f(t, y)$
$K_{E.L}$	= the net present value of unit length-dependent cost for environment impact (\$/m)	$y_{max}$	= the maximal ordinate of the region of interest (real)
$K_{E.V}$	= the unit environmental cost per VKT (\$/VKT)	$y_{min}$	= the minimal ordinate of the region of interest (real)
$L$	= vertical cutting line that is perpendicular to and intersects $\overline{SE}$	$\Lambda$	= chromosome (a vector of real number)
		$\lambda_i$	= the $i$ th gene (real)
		$\theta$	= the angle between the vertical cutting line the $X$ axis (real)
		$\delta_i(x_{\delta_i}, y_{\delta_i})$	= the center point of the $i$ th circular curve
		$\rho$	= the assumed interest rate (decimal)
		$\omega$	= $r_c[0, 1]$
		$\Delta_i$	= the intersection angle at $P_i$ (real)