BI-LEVEL GENETIC ALGORITHM APPROACH FOR 3D ROAD ALIGNMENT OPTIMIZATION

FAN TAO

(B.Eng., South East University)

A THESIS SUBMITTED FOR THE DEGREE OF MASTER OF ENGINEERING
DEPARTMENT OF CIVIL ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2004
First and foremost, the author wished to express his heartfelt appreciation and gratitude to Associate Prof. Chan Weng Tat for his patient guidance, support and encouragement given throughout the course of the research. He has developed confidence to face the challenges in life after spending about two years in the National University of Singapore.

Sincere gratitude to the lab supervisor, Associate Prof. Cheu Ruey Long, and lab technicians Mr. Foo and Mr. Ooh for their assistance in providing excellent laboratory equipment and environment.

The author gratefully acknowledges the financial support provided by the National University of Singapore. He would also like to thank all his friends in the ITVS lab, Pan Xiaohong, Liu Nan, Wu Lan, Sun Yueping, Yao Li, Liu Daizong, Lin Xiaoying, Song Liying, Li Yitong, Zheng Weizhong, Wang Hao, Liu Qun, Brandon, Huang Yongxi, Deng Weijia, Cao Zhi, Xie Chenglin, Pierre, Dong Meng, Huang Yikai, Chen Shihua and Xiong Yue for accompanying and helping him during his study period.

Last but not least, the author is profoundly grateful to his beloved wife and son, his parents, his parents-in-law, brother for their unceasing understanding, love, concern and support throughout the dissertation.
SUMMARY

Determining the best road alignment in 3D space is a difficult road engineering problem for computers to solve without human guidance. Computer methods are necessary to automate the search through many feasible solutions to determine one that incurs the minimal total costs. The search space increases exponentially from 2D to 3D space; this has motivated the decomposition of the 3D road alignment problem into two separate horizontal and vertical alignment sub-problems.

Genetic algorithms (GA) are an optimization method based on evolutionary principles. In the first part of the research, the GA has been used as the basis to develop methods to optimize the horizontal and vertical alignments separately. In the horizontal alignment problem, the objective is to determine the best road alignment in 2D horizontal space. For each horizontal road alignment, it is necessary to determine the best vertical alignment among the many possible vertical alignments. The 3D alignment is obtained by combining the horizontal and vertical alignments. The case studies show that the proposed approach can very quickly and consistently improve the quality of the solutions for both the horizontal and vertical alignment problems using an iterative procedure.

Due to the non-linear interaction between horizontal and vertical alignments, and elements of the total cost, the best 3D alignment cannot be obtained by combining the best horizontal alignment and the best vertical alignment. Therefore, a bi-level GA approach is developed in this thesis to optimize the 3D alignment. The example include in the study shows that the proposed bi-level GA programming quickly identifies combinations of horizontal and vertical alignments to give high quality 3D alignments.
based on the total cost. Several noteworthy points about the final alignment obtained are 
(a) the alignment is continuous both in the horizontal and vertical planes; (b) the number 
of horizontal and vertical intersection points that define the alignment need not be the 
same; and (c) the number of intersection points is determined by the bi-level GA 
depending on the terrain condition.

**Keywords:** 3D road alignment, bi-level algorithm, horizontal alignment, vertical 
alignment, genetic algorithms.
# TABLE OF CONTENTS

Acknowledgements
Summary
Table of Contents
List of Tables
List of Figures

## CHAPTER 1 – INTRODUCTION

1.1 OVERVIEW OF THE ROAD ALIGNMENT OPTIMIZATION 1
1.2 OBJECTIVES AND SCOPE OF THE RESEARCH 3
1.3 ORGANISATION OF THE THESIS 4

## CHAPTER 2 – LITERATURE REVIEW

2.1 OVERVIEW 5
2.2 MODELS FOR OPTIMIZING THE VERTICAL ROAD ALIGNMENT 5
   2.2.1 Explicit Enumeration 6
   2.2.2 Dynamic Programming 7
   2.2.3 Linear Programming 8
   2.2.4 Numerical Search 9
   2.2.5 Genetic Algorithms 10
2.3 MODELS FOR OPTIMIZING THE HORIZONTAL ROAD ALIGNMENT 12
   2.3.1 Dynamic Programming 12
CHAPTER 3 – FORMULATION OF THE ROAD ALIGNMENT OPTIMIZATION PROBLEM

3.1 DATA FORMAT FOR DESCRIBING THE REGION OF INTEREST 28

3.2 OVERVIEW OF COST MODELLING 30

3.2.1 Supplier Costs 30

3.2.2 User Costs 30

3.2.3 Summary of Cost Considerations 31

3.3 COST MODELLING IN THE ROAD ALIGNMENT ANALYSIS 31
3.3.1 Earthwork Cost 31
3.3.2 Land Use Cost 34
3.3.3 Pavement Cost 35

3.4 DESIGN CONSTRAINTS 35
3.4.1 Vertical Alignment 35
3.4.2 Horizontal Alignment 39

3.5 REPRESENTATION OF THE ALIGNMENT 40
3.5.1 Representation of the Horizontal Alignment 41
3.5.2 Representation of the Vertical Alignment 45

3.6 SUMMARY 47

CHAPTER 4 – OPTIMIZING THE ROAD ALIGNMENT 49

4.1 GENETIC ALGORITHMS FOR OPTIMIZING THE HORIZONTAL ALIGNMENT 49
4.1.1 Genetic Encoding 49
4.1.2 Initial Population 51
4.1.3 Fitness Function 51
4.1.4 Selection and Replacement 52
4.1.5 Genetic Operators 52
4.1.6 Convergence 53
4.1.7 Case Study 53

4.2 GENETIC ALGORITHMS FOR OPTIMIZING THE VERTICAL ALIGNMENT 58
4.2.1 Data Preparation 59
4.2.2 Genetic Encoding 59
4.2.3 Initial Population 60
4.2.4 Fitness Function 61
4.2.5 Genetic Operators 62
4.2.6 Convergence 65
4.2.7 Case Study 65

4.3 BI-LEVEL GENETIC ALGORITHMS FOR OPTIMIZING THE 3D ROAD ALIGNMENT 69
4.3.1 Bi-level Formulation of the 3D road alignment Optimization Problem 70
4.3.2 Performance of the Bi-level Program 73
4.3.3 Comparison of Jong’s Model and the Proposed Model for Vertical Alignment Optimization 80

CHAPTER 5 – CONCLUSIONS AND RECOMMENDATIONS 86
5.1 SUMMARY AND CONCLUSION 86
5.2 RECOMMENDATIONS FOR FUTURE RESEARCH 87
5.2.1 Improvements in Cost Estimation 87
5.2.2 Extensions of Model Capabilities 88

APPENDIX A – CALCULATION OF FITNESS FUNCTION FOR HORIZONTAL ALIGNMENT 89
LIST OF TABLES

Table 3.1  Cost Items in Different Road Alignment Analysis  31
Table 4.1  GA Parameters for the Horizontal Alignment Test Case  56
Table 4.2  Parameters for the Vertical Alignment Test Case  68
Table 4.3  Parameters of the Upper Level for Test Case  75
Table 4.4  Parameters of the Lower Level for Test Case  75
Table 4.5  Cost Components for the best Alignment (S$)  79
Table 4.6  Parameters of the two Programs for Vertical Alignment Optimization  81
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1</td>
<td>A One-point Crossover</td>
<td>24</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>An Example of Mutation</td>
<td>24</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Basic Structure of Genetic Algorithms</td>
<td>25</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>An Example of Study Area for Alignment Optimization</td>
<td>29</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>An Example of Transformation</td>
<td>29</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Typical Cross Section</td>
<td>32</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>Typical Vertical Curves</td>
<td>37</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>Decision Variables at each Vertical Cut</td>
<td>40</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Geometric Specification of a Circular Curve</td>
<td>42</td>
</tr>
<tr>
<td>Figure 3.7</td>
<td>An Example of Horizontal Alignment Discontinuity</td>
<td>43</td>
</tr>
<tr>
<td>Figure 3.8</td>
<td>A Typical Vertical Alignment</td>
<td>46</td>
</tr>
<tr>
<td>Figure 3.9</td>
<td>Discontinuous Situation of Vertical Alignment</td>
<td>47</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>GA-based Procedure for Horizontal Alignment Optimization</td>
<td>49</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>The Test Domain</td>
<td>54</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Sensitivity Study of Population Size on Horizontal Alignment Analysis</td>
<td>55</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Sensitivity Study of Mutation Rate on Horizontal Alignment Analysis</td>
<td>55</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Sensitivity Study of Crossover Rate on Horizontal Alignment Analysis</td>
<td>56</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>The Best Horizontal Alignment at the 200th Generation</td>
<td>56</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>Objective Value through successive Generations</td>
<td>57</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>GA-based Procedure for Vertical Alignment Optimization</td>
<td>58</td>
</tr>
<tr>
<td>Figure 4.9</td>
<td>Envelope of Feasible Zone Subject to Maximum Allowable Gradient</td>
<td>61</td>
</tr>
<tr>
<td>Figure 4.10</td>
<td>3D View of the Test Domain</td>
<td>65</td>
</tr>
</tbody>
</table>
Figure 4.11  Ground Elevation of the Test Domain  
Figure 4.12  Sensitivity Study of Population Size on Vertical Alignment Analysis  
Figure 4.13  Sensitivity Study of Mutation Rate on Vertical Alignment Analysis  
Figure 4.14  Sensitivity Study of Crossover Rate on Vertical Alignment Analysis  
Figure 4.15  Horizontal Alignment and its associated Optimal Vertical Alignment  
Figure 4.16  Bi-level GA-based Procedure for 3D Alignment Optimization  
Figure 4.17  Objective Values (of earthwork costs) through successive Generations  
Figure 4.18  The Best Alignment in the First Generation  
Figure 4.19  The Best Alignment in the 50\textsuperscript{th} Generation  
Figure 4.20  The Best Alignment in the 100\textsuperscript{th} Generation  
Figure 4.21  Objective Value through successive Generations  
Figure 4.22  Case Study 1  
Figure 4.23  Case Study 2  
Figure 4.24  Case Study 3  
Figure 4.25  Case Study 4  
Figure 4.26  Case Study 5  
Figure 4.27  Comparison of Results (Earthwork Cost S$)  
Figure A1  Cell Definition of the Study Region for Land Use  
Figure A2  An Example of Points of Tangency and Curvature  
Figure A3  Sorted Intersection points of A tangent Section  
Figure A4  Intersection Points of Grids and Circle  
Figure A5  Sorted Intersection points of A Circular Curve  
Figure A6  The Geometric Representation of Equation 4.16
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A7</td>
<td>Geometric Representation of the Direction for Vectors</td>
<td>99</td>
</tr>
<tr>
<td>A8</td>
<td>Cell Definition of the Study Region for Ground Elevation</td>
<td>100</td>
</tr>
<tr>
<td>A9</td>
<td>Geometric Representation of $S_C$ and $S_T$</td>
<td>102</td>
</tr>
<tr>
<td>A10</td>
<td>Geometric Representation of equation 4.18</td>
<td>103</td>
</tr>
<tr>
<td>A11</td>
<td>Station point on a parabolic curve</td>
<td>105</td>
</tr>
<tr>
<td>A12</td>
<td>Station point on a tangent section</td>
<td>106</td>
</tr>
</tbody>
</table>
CHAPTER 1 INTRODUCTION

1.1 Overview of the Road Alignment Optimization

Optimizing road alignments is a difficult combinatorial problem from road engineering. A road is described in plan and elevation by horizontal and vertical alignments respectively. For a proposed new road or relocation of an existing road, one of the first tasks in design is to determine the road alignment. Road alignments optimization is to find a feasible road alignment connecting two given end points such that the alignment incurs minimal total costs.

The final optimal alignment must also satisfy a set of design constrains and operational requirements. The task of identifying such an alignment which is so called optimal alignment is complex and challenging. It involves the evaluation of a possibly infinitely large number of alternative alignments in order to select one which results in minimal total costs. The alignment selection process is one of the most important tasks in road design because it is extremely difficult and costly to correct alignment deficiencies after the road has been constructed [AASHTO, 1994].

The traditional road design process usually consists of three different stages, namely route location, preliminary design, and final design. Firstly, the engineers will choose a broad corridor for the proposed road alignment. This is followed by studies to narrow down to several preliminary alignments. Finally, detailed analyses of both horizontal and vertical alignments are performed to select the final road alignment. The procedure, which requires professional judgment in various fields including transportation, economics, ecology, geology, environment, and politics, has proven to be lengthy and elaborate [Jong, 1998].
It is often desirable to pose the design problem at the design phase as an optimization problem. With reasonable mathematical models and high-speed computers, engineers can speed up the design process and get a good design rather than a merely satisfactory solution. In fact, the road alignment optimization problem has attracted a lot of research attention over the past thirty years. OECD [1973], Shaw and Howard [1982], Fwa [1988], and Jong [1998] have developed mathematical models and computer programs to optimize the road alignment. The results obtained from these previous studies have shown that optimization models can yield considerable improvement in construction cost compared with the conventional manual design. For example, [Stott, 1972] found that about 15% of construction cost saving can be achieved by using computers and mathematical programming techniques as compared to the conventional manual design method.

However, these existing models are not widely used in real engineering projects and can be improved in certain respects. A realistic model together with an effective search algorithm and an accurate total cost calculation is needed for the road alignment optimization problem. The difficulties in developing an efficient and accurate model are mainly because of the complex representation of a three-dimensional (3D) road alignment. The problem itself has a continuous search space and thus makes the number of alternative alignments infinitely large. Furthermore, the total cost associated with a road alignment is complex. Some of them are explicit (such as land use cost, earthwork cost, pavement cost and so on) while others may implicit (e.g. vehicle operating cost, travel time cost and accident cost). Any change in the alignment will incur corresponding changes in the total cost, especially if the terrain over which the alignment is optimized is irregular and fluctuate greatly. Finally, the
proposed alignment must also satisfy a set of design constraints and operational requirements.

There are three major types of road alignment optimization:

a) Horizontal alignments optimization

b) Vertical alignments optimization

c) 3D alignments optimization

The horizontal alignment usually consists of a series of straight (tangent) lines, circular curves, and possible spiral transition curves. Optimizing the horizontal alignment is important in relatively flat terrain or built-up areas. The main reason may be that vertical alignment will not change very much in such kind of areas. On the other hand, the vertical alignment usually consists of a series of straight lines (tangents) joined to each other by parabolic curves. Vertical alignment optimization is commonly performed for a cross-country road that traverses across different types of terrain. Horizontal alignment optimization is more complex and requires substantially more data than vertical alignment optimization [OECD, 1973]. Most agencies handle the road alignment problem as two separate tasks. The first one is optimizing the horizontal alignment while the second one is optimizing the vertical alignment for the horizontal alignment selected by the first task. The most difficult form of road alignment analysis is the 3D alignment optimization that involves both horizontal and vertical alignment optimization simultaneously. 3D alignment optimization to choose the best combined horizontal and vertical alignments can be attempted when the broad corridor of a new road has been defined.

1.2 Objectives and Scope of Research
Chapter One

The main objective of this research is to find a 3D alignment connecting two given end points which minimizes total costs and satisfies the design and operational constrains. Four research goals will be pursued to achieve this objective:

a) Develop a model for optimizing the vertical road alignment
b) Develop a model for optimizing the horizontal road alignment
c) Develop a model for optimizing the 3D road alignment
d) Design a efficient search algorithm for solving the proposed models

Road alignments optimization is a very complicated problem. The two critical successful factors in the optimization of road alignments should be a good search algorithm, and an efficient and accurate way to calculate the total costs of the road [Chan & Fan, 2003]. This research will attempt to design a good search algorithm as well as identify the elements of a realistic cost model to optimize road alignment.

1.3 Organisation of the Thesis

This thesis consists of five chapters. Chapter One defines the objectives and scope of the research. Chapter Two presents a literature review on the research area and some background of the optimization technique used in this area.

Chapter Three illustrates the key theoretical basis behind this study including the representation of the road alignment, the cost modelling in road alignment analysis and the constraints formulation for both vertical and horizontal alignment analysis.

Chapter Four first describes the models and solution techniques based on genetic algorithms for horizontal and vertical road alignments separately. These two approaches are then combined together as a bi-level genetic algorithm programming to optimize the 3D road alignment.

Finally, Chapter Five concludes, summarizes all the major findings and provides recommendations for future research.
CHAPTER 2 LITERATURE REVIEW

2.1 Overview

Optimization of road alignments has attracted much research interest since the early 70s because of the improvement of the computer’s capabilities and mathematical programming techniques. Many different models for optimizing road alignments have been developed. Although existing models have performed well in some aspects, most of them were developed based on some unrealistic assumptions or overlook some important aspects of the problem. For example, some of the existing models consider the road alignment as piecewise linear segment which is too rough for road alignment representation [e.g. Easa, 1988; Puy Huarte, 1973; Fwa, 1989; Hogan, 1973; Nicholson, 1976]. We will give a detailed review of the existing models in the following sections.

The literature review for this research is divided into six sections. Section 2.1 gives a brief overview of the optimization models. In sections 2.2 through 2.4, the advantages and disadvantages of existing models for vertical, horizontal and 3D road alignment optimization are reviewed respectively. Section 2.5 gives a brief introduction of Genetic Algorithms. Finally, in the last section 2.6, a brief summary about road alignment optimization and some characteristics of a good optimization model for road alignment to be addressed in this research are outlined.

2.2 Models for Optimizing the Vertical Road Alignment

A survey of the literature revealed that there were more models for optimizing the vertical alignment than there were for optimizing the horizontal alignment; there were fewer still optimizing the alignment in three dimension. It is postulated that one reason for this could be that the fewer costs (e.g. earthwork cost) are significantly influenced by the vertical alignment. Existing models for vertical alignment
optimization can be classified into five categories based on the research models and search algorithms.

2.2.1 Explicit Enumeration

Easa [1988] presents a model which selects the roadway grades that minimize the cost of earthwork and satisfy the geometric specifications. His model determines the elevations at predetermined stations along the horizontal alignment set at equal intervals. The search procedure employed to determine the station elevations is quite straightforward - all the possible combinations of elevation were enumerated and checked. For each combination of elevation, the following steps are taken: (i) check against design constraints and discard the combination if any constraint violation is detected; (ii) if feasible, determine the earthwork volumes for that elevation combination; (iii) check whether the borrow or disposal volumes do not exceed the capacities of the borrow pit or landfill. If this constraint is violated, the alignment is deemed infeasible and discarded; otherwise, linear programming is used to derive the most economic earth-moving plan. The above procedure is repeated until all combinations of intersection points have been investigated. The final optimal alignment is the elevation combination which has the lowest total cost which consists of earthwork cost and earthwork allocation cost.

Easa’s model includes most of the important geometric constraints such as minimum slope, maximum gradient, minimum distance between reverse curves, range of elevation at each station, etc. as well as constraints on the capacities of the borrow pit and landfill. The main limitation of Easa’s approach is the exhaustive nature of the search and was time consuming because all possible combinations are explored. Furthermore, only a discrete set of elevations was considered at each station. Although this helped to limit the size of the search, it also meant that only a subset of the
problem’s actual search space was included. Therefore, there is doubt about the accuracy of the earthwork volumes (and cost) calculated, and the resulting solution cannot be considered a globally or nearly globally optimal solution. Another weakness of Easa’s approach is that the model only considered earthwork costs; other important costs such as pavement cost and vehicle operating cost are not considered.

### 2.2.2 Dynamic Programming

Dynamic programming is the most widely used method for optimizing vertical alignments as this method is well suited to the problem structure. Each successive station on the alignment route is considered as a stage in a dynamic programming model while the different possible elevations at each station are deemed to be the states at each stage.

Most dynamic programming models for optimizing the vertical alignment generate the alignment as a series of piecewise linear segments [e.g. Puy Huarte, 1973; Goh, Chew, and Fwa, 1988; Fwa, 1989]. The common approach adopted in these models first constructs vertical lines (called cut lines) perpendicular to the road axis at equal intervals along the horizontal alignment. The trial road profile can pass at any one of the several elevations on each cut line. The objective function usually considers the minimization of the sum of the earthwork and operating costs. Constraints on gradient and curvature are imposed by restricting elevation differentials between the levels at adjacent cut lines during the search. The costs of all feasible road alignments are compared to find the lowest total cost and the corresponding route from the end stage to the start stage of the scheme. The gradient constraints can be treated more efficiently in comparison to the curvature constraints.

Murchland [1973] also used the dynamic programming approach to optimize the vertical alignment by minimizing the earthwork cost. Unlike the models discussed
above, Murchland used a set of quadratic spline functions with points at equal intervals to specify the alignment. The proposed alignment is smooth everywhere. The first and second derivatives of the alignment can be obtained at any point along the alignment, making it easy to formulate the gradient and gradient change constraints. However, the alignment is still restricted to pass through a limited finite set of points at each station.

The dynamic programming approach for vertical alignment optimization has been the most successful one to-date. However, only a finite set of points is considered at each station. Thus only a subset of the problem’s search space is considered and this cannot guarantee a global or nearly global optimum. Furthermore, the use of piecewise linear segments to represent the vertical alignment is too coarse for alignment applications, although, the final road profile can be smoothed by fitting a binomial curve. However, this detracts from the elegance of the dynamic programming search.

### 2.2.3 Linear Programming

ReVelle, Whitlatch, and Wright [1997] report the use of a linear programming approach to optimize the vertical alignment to minimize the earthwork cost. They use a 5th order polynomial function to represent the vertical alignment. The first and second derivatives can be easily obtained at any point along the horizontal alignment since the vertical alignment is a 5th order continuous function. Again, the use of a functional representation for the alignment allows the gradient and gradient change constraints to be easily formulated. A linear programming approach is employed to optimize the coefficients of the 5th polynomial function so that the total earthwork volume is minimized.

This model differs from the previous models in several aspects. Firstly, the elevation of any point along the vertical alignment can be easily calculated using the 5th order polynomial function. Secondly, there exist well-developed algorithms, such
as the simplex method, to solve the linear programming problem. However, Jong [1998] pointed out that the 5th order polynomial cannot represent road alignments realistically. Moreover, earth-work volumes are calculated using a simplified way without considering the side slopes. Omitting the side slopes in the calculation of the earth-work volumes maintains the linearity of the objective function, a requirement if the linear programming approach is to be used. Finally, only some of the points along the alignment are checked against the gradient and gradient change constraints and there is no guarantee that all the other points satisfy the constraints.

2.2.4 Numerical Search

An approach using numerical search for optimizing the vertical alignment has been proposed to overcome some obvious disadvantages of the other approaches. The search space defined in this approach is continuous rather than a discrete solution set.

Hayman[1970] suggested a model where the decision variables are defined as the elevations at each station and are continuous in nature. The alignment is then generated by connecting these points with straight line segments. In this model, the gradient and curvature constraints are formulated in the same way as Goh et al[1988] and Fwa[1989]. Hayman also considered additional constraints such as slope stability and material balance constraints.

The search method employed in Hayman’s study can be characterized as a line search method. It starts with an initial guess of the solution. A new point is formed by moving the original point towards its gradient direction with a step size. This procedure is repeated until no non-zero step size is found. The computational sequence is then altered to solve an auxiliary problem that seeks a new feasible direction. The entire algorithm will finally end up with a solution better than any other nearby points in the search space. Due to the local nature of the search procedure employed, the
solution found cannot be guaranteed to be a global optimum. In practice, several
different initial solutions are tried to increase the possibility of finding a good solution.

Goh, Chew, and Fwa[1988] also adopted continuous models for optimizing a
vertical alignment. The model is first formulated as a calculus of variations problem.
Then, this model is converted into an optimal control problem by some mathematical
techniques from optimal control theory [Goh and Teo, 1988]. The alignment is
parameterized by a set of cubic spline functions. The gradient and curvature constraints
can be easily formulated because of the availability of the first and second derivatives
of the cubic spline function. These constraints are then transformed further into one-
dimensional constraints via constraint transcriptions defined in optimal control theory.
The final model thus becomes a general constrained nonlinear optimization problem
with the coefficients of spline functions as its decision variables. The model can be
solved by a numerical search method and has several local minima.

In general, a well-formulated continuous model provides more flexibility in
defining the alignment configurations, and has the potential to yield a realistic
alignment. However, both formulation and the solution of the model are difficult.
Moreover, the problems are usually nonlinear and non-convex and many local optima
exist in the search space, making it difficult to find a globally optimal solution.

2.2.5 Genetic Algorithms

The genetic algorithm is search method motivated by the principles of natural
selection and “survival of the fittest”. A genetic algorithm performs a multi-directional
search by maintaining a population of potential solutions and encourages information
formation and exchange between these directions [Michalewiz, 1996]. Due to the
difficulties of general representation of road alignment as well as the complexity of
costs and constraints associated with road alignment, it seems to be very suitable for solving road alignment optimization problem.

Fwa et al[2002] present a model to solve the vertical alignment optimization problem with genetic algorithms. This model utilizes grids with data values defined at equal intervals, in directions vertical and perpendicular to the road axis. The trial road profile can pass through one of several elevations at each grid point. In the genetic algorithm solution process, a set of solutions, known as the parent pool, is first created by randomly selecting data values. A pool of solutions, known as the offspring solution pool, is then generated from the initial parent solution pool through genetic operators such as reproduction, crossover and mutation. A new pool of parent solutions is formed from the initial parent pool and the offspring pool by selecting the best solution. This procedure is repeated to obtain better solutions. It is stopped when negligible differences are observed between successive generational pools of the solutions. The best solution in the last iteration is taken as the optimal vertical alignment.

This genetic algorithm model was flexible enough to be able to include a variety of constraints. Besides the gradient and curvature constraints, it also considers the critical length of grade control, fixed-elevation points, and non-overlapping of horizontal and vertical curves; these constraints are not usually considered in models using conventional methods because of the difficulty in modelling them. However, the elevation at each intersection is only allowed to pass through a finite set of points, which is a subset of the whole search space and cannot guarantee the global or nearly global optima. Finally, the resulting alignment is still a piecewise linear segment, which is not accurate enough for application purposes.

2.3 Models for Optimizing the Horizontal Road Alignment
Models for optimizing horizontal alignments are more complex and require substantially more data than those for optimizing vertical alignments [OECD, 1973]. There is not much work on the optimization of horizontal alignments compared to the research on the optimization of vertical alignments. The optimization of horizontal alignments needs to consider political, socioeconomic, and environmental issues because of the interaction between the route of the road and land-use. The major cost components such as land cost, construction cost, social cost and environmental cost are very sensitive to changes in the horizontal alignment.

Generally, work on the optimization of the horizontal alignment adopts one of four approaches: dynamic programming, calculus of variations, network optimization, or genetic algorithms.

2.3.1 Dynamic Programming

Dynamic programming has been widely used for optimizing road alignments, especially vertical alignments as seen in section 2.2. The dynamic programming procedure for optimizing horizontal alignments is similar to that employed for vertical alignments. Firstly, the route between the start and end points of the alignment is divided into equal sections and straight lines perpendicular to the axis of the alignment are placed at stations located between these sections. Each station represents a stage of the dynamic programming problem, whilst nodes on the perpendicular line represent the state of each. The search procedure usually starts from the last stage, and proceeds backwards along the route towards the first stage. Trietsch [1987], Hogan[1973], and Nicholson[1976] are some of the researchers who used dynamic programming in horizontal alignment optimization.

Dynamic programming is efficient at optimizing the horizontal alignment. It needs lower storage requirements compared to the other approaches. However, during
the search procedure, only a limited number of nodes in the next stage are permitted to connect to the node at the current stage. That means only a subset of the whole search space is investigated and thus, the method cannot guarantee that any solution found is the global or nearly global optima. However, this is a drawback shared by all approaches which using a discrete search space. Moreover, the final alignment obtained by dynamic programming is composed of piecewise linear segments, which is not good enough for real applications as a typical horizontal road design consists of geometric curves and tangent lines.

2.3.2 Calculus of Variations

The calculus of variations seeks a curve connecting two end points in space which minimizes the integral of a function [Wan, 1995]. Howard, Bramnick, and Shaw[1968] developed a model that used the Optimum Curvature Principle (OCP). The principle states that the curvature of the optimal road location at each point on the road is equal to the logarithm of the directional derivative (percentage rate of change) of the criterion function perpendicular to the route. In other words, it was assumed that there existed a continuous cost surface above the two-dimensional region of interest. This principle was a necessary condition that an optimal route must satisfy in any region. This was achieved by minimizing the path integral of the criterion function. The optimization began with a search where several routes were initiated from the start point in several directions. The route that arrived at the end point was considered to be the optimal because it had traversed the field from the start point to the end point whilst obeying the optimum curvature principle.

The optimum alignment derived by the OCP is continuous and a global optimum is guaranteed; this is the main advantage of the method. The determination of the local cost function is a crucial point of the OCP which requires that the local cost
function be continuous over the region of interest. However, this is not necessarily so as the land use cost is usually not continuous between different zones. There are some approximations and assumptions behind the determination of the local cost function in the OCP.

2.3.3 Network Optimization

The basic idea of this approach is to formulate the optimization of horizontal alignment as a network problem, in which the alignment is represented by the arcs connecting the start point to the end point. Then, a well-developed network optimization technique such as the shortest path algorithm can be used to solve the problem.

The Generalized Computer-Aided Route Selection (GCARS) system, developed by Turner and Miles [1971], employed the shortest path algorithm. It borrowed the basic idea of network optimization where the route was represented by a series of arcs connecting the start and end points. A cost surface was prepared for each factor in the route selection problem. The total cost is calculated as the linear weighted combination of the different cost components. Finally, a grid network is formed from the cost model matrix by joining all nodes and assigning the cost to each link.

Athanassoulis and Calogero [1973] also employed network optimization techniques to solve the horizontal alignment problem. Unlike Turner’s model, where link costs are calculated by averaging the cost of the two end nodes of a link, all the costs in Athanassoulis’s model are mapped as “cost line” (like river, bridge) and “cost area” (such as lake, wetland) which formed a basis for calculating link costs. Then the cost between any pair of nodes was calculated by the summation of the length in each cost area multiplying the associated unit cost. The model comprised two phases. Phase I was a matrix generator program that calculated the elements of the cost matrix.
Phase II used a modified transportation problem program, which used the cost matrix to identify the optimal route as a sequence of straight segments. The cost between any pair of nodes was calculated as the summation of the product of length in each cost area and the unit cost.

There are several disadvantages associated with this approach. Firstly, the alignment is only allowed to pass through discrete points of the search space; thus searching only a subset of the real search space is included and there is a possibility of missing the global optima. Secondly, the optimal alignment derived by the network approach is made up of piecewise linear segments, which is not realistic for actual alignments. Finally, the calculation and storage requirements for link costs are high; if the resulting network is large, the computational time and computer storage space needed for the cost matrix are considerable.

### 2.3.4 Genetic Algorithms

Jong [1998] employed a genetic algorithm model to optimize the horizontal alignment. This model first randomly generates a route made up of a succession of piecewise linear segments. A curve with a fixed radius (for example, the minimal radius specified by AASHTO [1994]) is added at each point of intersection between two successive segments to define the horizontal alignment. The genetic algorithm actually generates a pool of such candidate alignments. Each of the candidate alignments of the current population pool will undergo selection, crossover, and mutation operators to form the next generation. This procedure will be repeated until there is no improvement between successive generations. Jong also defined eight problem-based genetic operators to speed up the convergence of the algorithm.

Unlike the above mentioned models, the optimal alignment derived by this approach is not a piece-wise straight line and represents a realistic alignment. The cost
items included in Jong’s model are more elaborate compared with the other models. However, the number of the horizontal intersection points between the given two end points is fixed in Jong’s model, while in real engineering project it should be variable depending on the terrain condition [Chan & Fan, 2003].

2.4 Models for Optimizing the 3D Road Alignment

Although several mathematical models have been developed to solve the road alignment optimization problem, most of them only emphasize either horizontal or vertical alignments. Models that simultaneously optimize both horizontal and vertical alignments are seldom found in the literature. The main reason may be that the 3D alignment optimization involves more factors and its geometric specification is more complex.

2.4.1 Dynamic Programming

The dynamic programming model for optimizing 3D alignment involves setting the stages of the model as equally spaced vertical planes between the start and end points i.e. in the top view, the stage planes are perpendicular to the line segment connecting the two end points of the alignment. The states of each stage are defined on a two-dimensional grid. The 3D alignment is obtained by connecting the grids at each stage. Studies using dynamic programming for optimizing 3D alignments include Hogan [1973] and Nicholson[1976].

The disadvantages of application of dynamic programming for optimizing 3D alignment are obviously. Firstly, the search area is discrete, which is only a subset of the whole search space. Secondly, the final alignment is a piecewise linear segment for both horizontal and vertical alignment, which is too rough for application. Finally, the computational time and the computer storage requirement for this approach are considerable.
2.4.2 Numerical Search

Chew, Goh, and Fwa[1989] developed a model which can optimize a “smooth” 3D alignment. This is the extension of their continuous model for vertical alignment optimization [Goh, Chew, Fwa, 1988].

The model utilized a set of cubic spline functions to interpolate the alignment. Then the authors transformed the constraints into one-dimensional constraints by the method of constraint transcription used in the optimal control theory. Finally the model becomes a constrained nonlinear program structure with the coefficient vectors of spline functions as its decision variables.

The optimal 3D alignment derived by this approach is smooth everywhere. However, like other models for optimizing vertical alignments by numerical search, the solution found by this model only guarantees a local optimum. In practice, different initial solutions with human judgement will be used for running the model. Moreover, this model is developed based on the assumption that all the cost functions associated with the road are continuous within the region of interest. It is difficult for this model to deal with the discontinuous local cost function (for example land use cost) into the objective function because the algorithm requires a differentiable objective function.

2.4.3 Genetic Algorithms

Jong[1998] develops an evolutionary model for solving 3D alignment optimization problem. It overcomes some drawbacks in existing models. The proposed GA model for the 3D road alignment optimization problem is as follows. Firstly, a piecewise straight line, which connects the start point and the end point of the alignment, is randomly generated. This piecewise straight line is a spatial line (line in 3D space). The projection of the spatial line onto the $XY$ plane becomes the horizontal alignment. The author completes the horizontal alignment by adding a curve with a
fixed radius (the minimal allowable radii according to AASHTO [1994] is used in this study) at each point of intersection in the horizontal alignment. The projection of this spatial line onto the surface orthogonal to the $XY$ horizontal plane containing the horizontal alignment determines the vertical alignment. Adding minimal allowable length of parabolic curves to the vertical intersection points completes the vertical alignment. This model can therefore determine the 3D alignment of the road.

Genetic algorithm is used in this study to optimize the 3D alignment. The initial population of the problem is randomly generated in order to keep the diversity of the problem. Then the parent population will undergo selection, crossover, and mutation operators to generate some offspring population. The best chromosomes (solutions) from both the initial parent population and the offspring population will form a new parent population for the next iteration. This procedure will repeated until the predefined condition of termination is satisfied.

Jong’s model considers most of the cost associated with road alignment such as earthwork cost, land use cost, user cost and so on. However, his model for computing the land use cost is developed for grids of rectangular cells with uniform interval characteristics. This prevents its application to irregularly shaped geographic features. Furthermore, it is based on piecewise linear approximations of the alignment, which reduce its precision. Jha [2000, 2001] extends Jong’s work by linking GIS database to the optimization operations. A GIS based comprehensive road cost model is used for optimizing road alignment in Jha’s work. An integrated model is developed by linking a GIS model with an optimization model employing genetic algorithm. The GIS model provides accurate geographical features, computes land use costs, and transmits theses costs to an external program. That program computes the other costs and then, using genetic algorithm, to optimize the road alignment.
The proposed algorithm can optimize complex, comprehensive, and non-differentiable objective function. The model can also exploit detailed geographical information for road analysis. The resulting alignment are smooth everywhere and can have backward bends (i.e., “backtracking”) to better fit terrain and land-use patterns.

The application of genetic algorithm in optimizing 3D alignment still has several defects. First, there is a tendency for horizontal and vertical curves to coincide in the resulting 3D alignment in Jong’s model while it is not the real condition in practice. This occurs because the same points of intersection control both the vertical and horizontal alignments. In other words, for a horizontal alignment, Jong only consider a particular group of vertical alignments which has the same intersection point position as the horizontal one. Thus, only the subset of the whole search space is investigated. Although Jong [1998] states in his dissertation:

“To avoid this problem, after completing the search program, a further refinement on the vertical alignment is performed by another genetic procedure in which the vertical control points are reset so that the vertical curves are located in different positions from the horizontal curves”

However, if we do the refinement, that is inconsistent with the alignment during search. Some case studies will be presented in Chapter 4 to illustrate this limitation of Jong’s model.

Furthermore, the number of intersection points of the proposed horizontal and vertical alignment is fixed in Jong’s model which restricts the configuration of the road alignment. It should be variable depending on the terrain condition.

2.5 Overview of Genetic Algorithms

The traditional theoretical optimization techniques require the problem to be formulated mathematically. However, in a real-life road project, it is very difficult to
represent the 3D alignment mathematically. The very large number of feasible
solutions in a typical road design problem also renders most conventional optimization
techniques unsuitable for practical applications of road alignment analysis.

A relatively new optimization technique known as genetic algorithms (GAs) is
adopted for the present research to overcome the problems described in the preceding
paragraph. Genetic algorithms are evolutionary methods motivated by the principles of
natural selection and “survival of the fittest”. It is a directed random search technique,
invented by Holland [Holland, 1975]. The GAs perform a multi-directional search by
maintaining a population of potential solutions and encourages information formation
and exchange between these directions [Michalewiz, 1996]. GAs are stochastic
algorithms that can be used to find approximate solutions for complex problems. The
problems usually have a search space that typically is much too large to be searched by
means of enumerative methods.

GAs work with an evolving set of solutions (represented by chromosomes)
called the population. Solutions from the current population are taken and used to form
a new population to replace the current population. This is motivated by expectation
that the quality of solutions in the new population will be better than that in the
previous one. Solutions are selected to form new offspring according to their fitness.
The fitter they are, the more chances these solutions will have to be selected. The basic
steps of the GAs are as follows:

Step1: Determine a genetic representation for potential solutions to the problem.
Step2: Generate an initial population of candidate solutions.
Step3: Compute the fitness of each individual.
Step4: Select individuals from the parent population according to their fitness.
Step5: Apply both the crossover and mutation operators to each selected
individual to form the offspring population.

Step 6: If a pre-specified stopping condition is satisfied, stop the algorithm; otherwise, return to step 3

The application of GA to a specific problem includes several steps. A suitable encoding for the solution must be devised first. We also require a fitness function through which the individuals are selected to reproduce offspring by undergoing genetic operators. Each of the steps is described below:

2.5.1 Genetic Encoding

To apply GA to a specific problem, we must first devise an appropriate genetic representation for the solution. Originally, a potential solution to the problem is encoded into a string of a given length, which is referred as a chromosome or genotype. The method of representation has a major impact on the performance of the GA. Different representation schemes might cause different performance in terms of accuracy and computation time.

There are two common representation methods for numerical optimization problems [Michalewiz, 1996; Davis, 1991]. The preferred method is the binary string representation method. The second representation method is to use a vector of integers or real numbers, with each integer or real number representing a single parameter.

2.5.2 Fitness Function

The fitness evaluation unit acts as an interface between the GA and the optimization problem. The GA assesses solutions for their quality according to the information produced by this unit and not by using direct information about their structure. Given a particular chromosome, the fitness function returns a single value, which represents the merit of the corresponding solution to the problem.

Fitness evaluation functions might be complex or simple depending on the
optimization problem at hand. Where a mathematical equation cannot be formulated for this task, a rule-based procedure can be constructed for use as a fitness function or in some cases both can be combined. Where some constraints are very important and cannot be violated, the structures or solutions which do so can be eliminated in advance by appropriately designing the representation scheme. Alternatively, they can be given low probabilities by using special penalty functions.

2.5.3 Selection and Replacement

The individuals in the population are selected to reproduce offspring according to their fitness values. The higher the fitness function, the more chance an individual has to be selected. There are two different types of selection schemes: proportionate selection and ordinal-based selection. The concept behind these two approaches is the selective pressure, which is defined as the degree to which the better individuals are favoured in the selection process. A strong selective pressure may lead to premature convergence (i.e., converge to a local optimum), while a weak selective pressure tends to reduce the convergence of a GA.

Once offspring are produced, we must determine which of the current members of the population should be replaced by the new offspring. Replacement is strongly related to the selection process, where we decide which of the current members of the population is going to reproduce offspring. There are many kinds of classifications of replacements. From the sampling space point of view, we can basically categorize them as either regular sampling space or enlarged sampling space. Note that it is not guaranteed that the newly born offspring will dominate their parents, and that the best chromosome in the current generation will not be selected to die. An elitism model is thus developed for preventing the best individual from dying off. In this policy, the best chromosome is always passed on to the next generation.
2.5.4 Genetic Operators

In classical GA, offspring are generated from their parents by two typical types of genetic operators: mutation and crossover.

1) Crossover

This operator is considered the one that makes the GA different from other algorithms, such as dynamic programming. It is used to create two new individuals (children) from two existing individuals (parents) picked from the current population by the selection operation. The intuition behind the applicability of the operator is information exchange between potential solutions. The mechanism is similar to sexual mating in nature. The crossover operator is supposed to help in exploiting the information of the better individuals in the population.

There are several ways of doing this. Some common crossover operations are one-point crossover, two-point crossover, cycle crossover and uniform crossover. Figure 2.1 shows an illustration of one-point crossover, which is the simplest crossover operator in GAs.

![Figure 2.1 A One-point Crossover](image)

2) Mutation

In this procedure, all individuals in the population are checked bit by bit values are randomly reversed according to a specified rate. Unlike crossover, this is a monadic
operation. That is, a child string is produced from a single parent string. The mutation operator forces the algorithm to search new areas. Eventually, it helps the GA avoid premature convergence and find the global optimal solution. Figure 2.2 shows an example of mutation.

Figure 2.2 An Example of Mutation

2.5.5 Convergence

If a GA has been correctly implemented, the population will evolve over successive generations so that it will converge toward the global optimum. However, GA cannot be expected to stop spontaneously, nor guaranteed to find the global optimum. The evolution has to be stopped at some point according to a predetermined criterion. There are usually three stopping rules to stop the evolution: 1) iteration limit exceeded, 2) population too similar, and 3) no change in the best solution found in a given number of iterations.

Figure 2.3 shows the basic flowchart of a general genetic algorithms search procedure.
2.6 Summary

Road alignments optimization is one of the most complex and challenging problems in road design. The main objective of this problem is to minimize the total costs (for example, land use cost, earthwork cost, pavement cost, etc.) while satisfy a set of design constraints and operational requirements. The conventional manually design procedure for road alignment is as follows. Firstly, the engineers select the most suitable horizontal alignment while the costs which are sensitive to vertical alignment are considered roughly depending on the experiences of the engineers. Vertical alignment analysis is then performed to minimize the total costs which are sensitive to the vertical alignment for the selected horizontal alignment. This procedure can not guarantee the global optima obviously. Most of the models found in the literature review optimize either vertical or horizontal alignment separately. Only a few models
Chapter Two

are developed to solve the 3D alignment. The advantages and disadvantages of the existing models are discussed in the previous sections.

The problem of optimizing vertical alignment can be stated as: given a fixed horizontal alignment, find the optimal vertical alignment to minimize the total cost associated with this particular alignment. Models for vertical alignment optimization are widely found in the literature review. It is the easiest one compared with the horizontal and 3D alignment optimization. The main reason may be that there is only a few costs (such as earthwork cost) are sensitive to vertical alignment so that other cost items can be ignored during optimization.

Horizontal alignment analysis is more complicated than vertical alignment analysis. Among all the models found in the literature review, Jong's model [1998] seems to have the most reasonable solution for the problem. However, horizontal alignment analysis seems to only be available in relatively flat terrains or a built-up area since the earthwork volume within this region will not vary very much according to different configuration of horizontal alignment. All of the above models have not considered the earthwork cost or just given an approximation of the earthwork cost. According to the studies by OECD [1973] and Chew et al. [1989], earthwork costs reach up to about 25% of all construction costs. It is insignificance to optimize the horizontal alignment without considering the earthwork cost. Furthermore, earthwork volume will change considerably with different type of vertical alignment even with the same horizontal alignment. Therefore, we should also consider the vertical alignment during the optimization of horizontal alignment, which lead the optimization of horizontal alignment to the 3D alignment optimization.

3D alignment optimization is the most difficult problem among the alignment optimization problems. Fewer models are found in the literature review to solve this
problem. Among these existing models, Jong’s model [1998] seems to be the most reasonable one. However, there still exist some defects about his model as stated in subsection 2.4.3.

Apparently, none of the approaches discussed in the previous sections dominates the others, and there is always some trade off between them. As a summary, a good model for optimizing road alignment should have the following necessary conditions:

1. A good model for road alignment representation so that the resulting alignment is realistic.

2. Formulate the design constraints and operational requirements the more the better.

3. Optimize 3D alignments.

4. Find globally or nearly globally optimal solutions.

5. The search algorithm should be efficient.

6. The number of both horizontal and vertical intersection points should be variable depending on the terrain condition.
This chapter starts with the data format for describing the region of interest. The cost modelling for road alignment is then briefly outlined in section 3.2 and discussed in more detail in section 3.3. The design constraints and operational requirement are discussed in section 3.4. Section 3.5 presents the modelling approach for representing the alignment in the horizontal and vertical planes. Finally, the complete models for each optimization problem (including vertical, horizontal and 3D alignment) are presented in section 3.6.

3.1 Data organization to describe the Region of Interest

Certain assumptions are made when selecting and representing information about the region of interest for the purpose of solving the alignment problem computationally. These include:

1) The study region is rectangular in shape and two of the edges of this rectangle are parallel to the straight line connecting the two given end points of the proposed alignment. Other shapes can be transformed to the required rectangular shape mentioned above in the manner discussed below.

2) The study region is abstracted as a matrix of uniform cells, each cell containing a discrete value on some aspect of the region relevant to the alignment problem such as land acquisition cost, land-use cost, and soil condition. The area represented by each cell need not be the same for the different thematic matrices.

Figure 3.1 provides an example of the format used to describe the study area,
in which the coordinates of the origin (bottom left corner) are labeled as \( O(x_o, y_o) \).

We further denote \( x_{\text{max}} \) and \( y_{\text{max}} \) as the maximal \( X \) and \( Y \) coordinates of the study area. The straight line \( SE \), connecting two end points (\( S \) for start point and \( E \) for end point) of the proposed alignment, is parallel to the \( X \) coordinate axes.

![Figure 3.1 An Example of Study Area for Alignment Optimization](image)

![Figure 3.2 An Example of Transformation](image)

If the region of interest is not rectangular and the straight line \( SE \) connecting the two end points (\( S \) and \( E \)) of the proposed alignment is not parallel to the \( X \) coordinate axes, a transformation can be made to satisfy the above two assumptions as follows. Firstly, rotate the region of interest so that the straight line \( SE \) is parallel to the \( X \) coordinate axes. The study area can then be modeled as a collection of cells in a rectangular grid, where inaccessible regions are represented by cells with very high availability cost. Thus, any study region could be similarly transformed into a format
acceptable to the proposed model. Figure 3.2 shows an example of such a mapping.

3.2 Overview of Cost Modelling

In the optimization analysis of road alignments, all costs need to be suitably modelled to be included in the computerised calculation. Costs can be presented with different degrees of accuracy depending on the quality of data and the complexity of the modelling. The costs associated with road alignment design can be categorized as either supplier costs or user costs.

3.2.1 Supplier Costs

The supplier costs consist of length-dependent cost, location-dependent cost and earthwork volume cost.

\[ C_{\text{sup}} = C_L + C_N + C_V \]  

(3.1)

where \( C_{\text{sup}} = \) total supplier costs

\( C_L = \) length-dependent cost

\( C_N = \) location-dependent cost

\( C_V = \) earthwork volume cost

3.2.2 User Costs

The user cost considered is defined as the sum of the costs associated with vehicle operation, travel time, and accidents:

\[ C_U = \sum C_F + \sum C_T + \sum C_A \]  

(3.2)

where \( C_U = \) total user costs

\( C_F, C_T \text{ and } C_A = \) fuel consumption cost, travel time cost and accident cost respectively.

The computation of user costs is less straightforward and various models have been developed to estimate various user costs including vehicle operating, travel time,
and accident costs. These models were all derived from historical data using statistical regression. Different historical data in different country or region will lead to different results. User costs are significantly influenced by estimates of future traffic volumes. Due to the difficulties mentioned above, there is still no robust model to estimate the user cost. Therefore, user costs are not included in this study.

3.2.3 Summary of Cost Considerations

Different types of costs will favour different alignment configurations. Table 3.1 shows the cost items included in the calculation of costs for both the horizontal and vertical alignments. For vertical alignments, earthwork costs are dominant whilst for horizontal alignments the main consideration is land related and other location-dependent costs. The various cost items considered in this study are discussed in more detail in the next section.

<table>
<thead>
<tr>
<th>Cost items</th>
<th>Alignment type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earthwork cost</td>
<td>☒</td>
</tr>
<tr>
<td>Land use cost</td>
<td>☒ ☒</td>
</tr>
<tr>
<td>Pavement cost</td>
<td>☒ ☒ ☒</td>
</tr>
</tbody>
</table>

3.3 Cost Modelling in the Road Alignment Analysis

3.3.1 Earthwork Cost

The surface elevation model can be used to determine the elevation of the existing ground along designated points of the chosen alignment. The ground profile perpendicular to the alignment can also be determined from the surface elevation model. Figure 3.3 shows the typical cut and fill cross sections along the road alignment.
where \( w \) = the width of the alignment

\( h \) = different between the road and ground level, positive for filling and negative for cut

\( a \) and \( b \) = angle of side slope of cross section, \( a \) for fill cross section and \( b \) for cut cross section

Among a number of methods available, the two methods in general use for obtaining the earthwork volume in road construction work are known as the Average End Area Method and the Prismoidal Method. The Average End Area Method assumes that the earthwork volume between two successive cross sections is the average of their areas multiplied by the distance between them. The Prismoidal Method is sometimes called “Simpson’s Rule” for Volumes. It is a modification of the End Areas Formula. The Average End Area Method is the simplest method to estimate earthwork volume. However, for linear ground profiles, the Prismoidal Method gives the exact volume, while the Average End Area method generally
overestimates the earthwork volume. Therefore, the proposed method used here is Prismoidal Method. The formulation of the Prismoidal Method is as follows:

\[
V = (A_1 + 4 \times A_m + A_2) \times L / 6
\]  

(3.3)

where, \( V \) = volume between two cross sections

\( A_1, A_2 = \) area of the two end cross sections

\( A_m = \) area of the middle cross section

\( L = \) distance between the two end cross sections

We derive the formula for the volume between two successive cross sections from equation 3.3 based on the assumption that the longitudinal ground profile between two successive stations is linear and the ground cross slope is level. There are generally four cases that need to be considered in computing the earthwork volume between two successive cross sections.

a) Consecutive cross-sections are cut sections and there is no crossing of the ground and road profiles

If \( h_1 < 0 \) and \( h_1 \times h_2 \geq 0 \), then

\[
\begin{align*}
V_{\text{fill}} &= 0 \\
V_{\text{cut}} &= \left( w + h_1 \cdot \cot(b) \right)h_1 + 4 \left( w + (h_1 + h_2) \cdot \cot(b) \right)(-h_1 - h_2) + (w - h_2 \cdot \cot(b))(h_2) \cdot L
\end{align*}
\]

(3.4a)

b) Consecutive cross-sections are fill sections and there is no crossing of the ground and road profiles

If \( h_1 \geq 0 \) and \( h_1 \times h_2 \geq 0 \), then

\[
\begin{align*}
V_{\text{fill}} &= \left( w + h_1 \cdot \cot(b) \right)h_1 + 4 \left( w + (h_1 + h_2) \cdot \cot(b) \right)(h_1 + h_2) + (w + h_2 \cdot \cot(b))h_2 \cdot L \\
V_{\text{cut}} &= 0
\end{align*}
\]

(3.4b)

c) Consecutive cross-sections are cut and fill sections, respectively

If \( h_1 < 0 \) and \( h_1 \times h_2 \leq 0 \), then
d) Consecutive cross-sections are fit and cut sections, respectively

If \( h_1 \geq 0 \) and \( h_1 \times h_2 \leq 0 \), then

\[
\begin{aligned}
V_{\text{fill}} &= \frac{(w + h_2 \cdot \cot(a))h_2 + 4(w + h_2 / 2 \cdot \cot(a)) \cdot h_2 / 2 \cdot L \times h_2}{6 (h_2 - h_1)} \\
V_{\text{cut}} &= \frac{(w - h_1 \cdot \cot(b)) \cdot (-h_1) + 4(w - h_1 / 2 \cdot \cot(b)) \cdot (-h_1) / 2 \cdot L \times (-h_1)}{6 (h_2 - h_1)}
\end{aligned}
\] (3.4c)

where \( V_{\text{fill}} \) = filling volume between two cross sections 1 and 2

\( V_{\text{cut}} \) = cut volume between two cross sections 1 and 2

\( w, h, a, b \) and \( L \) = same definition as Figure 3.3 and equation 3.3. While the subscript of \( h \) represent position of the cross sections.

The total earthwork volume can be obtained by summing up the cut and fill volumes along the horizontal alignment. Therefore, the total earthwork cost can be represented as follows:

\[
C_v = U_{\text{fill}} \times \sum V_{\text{fill}} + U_{\text{cut}} \times \sum V_{\text{cut}}
\] (3.5)

where \( U_{\text{fill}} \) and \( U_{\text{cut}} \) = unit cost of fill and cut volume

\( \sum V_{\text{fill}} \) and \( \sum V_{\text{cut}} \) = the total filling and cut volume

3.3.2 Land Use Cost

Land cost is defined as the product of unit land cost and the area of land required for the road right-of-way.

\[
C_{\text{Land}} = \sum A_i \times U_{\text{Land} - i}
\] (3.6)

where \( C_{\text{Land}} \) = total land use cost

\( U_{\text{Land} - i} \) = unit cost of land type \( i \)
$A_i = \text{total area of land type } i$

### 3.3.3 Pavement Cost

The Pavement cost can be represented as follows:

$$C_p = U_p \times L \quad (3.7)$$

where $C_p = \text{pavement cost}$

$U_p = \text{unit pavement cost}$

$L = \text{total length of the alignment}$

### 3.4 Design Constraints

There are a great number of constraints and operational requirements that need to be met when designing a road. These constraints have been developed over a long time and published in many handbooks and reports (for example [AASHTO, 1994]). In this study, only the most important constraints are included; these are discussed in the following sections.

#### 3.4.1 Vertical Alignment

The important design considerations with regards to vertical alignment include the design speed, sight distance, curvature control and the maximum allowable gradient.

**i) Maximum Allowable Gradient**

The vertical profile of a road is constrained by geometric design standards which are largely determined by the design speed of the road. The grade of a road is the vertical rise (or fall) per unit of horizontal distance, expressed as a percentage. The maximum grade to be adopted will depend on factors such as the design controls for vehicular operations, and whether the road is in a rural or urban area. This maximum gradient is imposed so that heavy vehicles can maintain reasonable speeds when
traveling up-hill. This reduces congestion caused by heavy vehicles and vehicle operating costs. It is generally accepted that a maximum grade of 4% to 5% could be applied without appreciable loss in vehicular speeds.

A road designed with a smaller value for the maximum vertical gradient constraint will enable smoother traffic flow. Savings in vehicle operating costs could also be achieved by stricter gradient control. However, it will increase the cost of earthworks.

The grade effect is more pronounced on truck operations. On upgrades the maximum speed that can be maintained by a truck is dependent primarily on the length and steepness of the grade as well as the weight/horsepower ratio. Other factors that affect the average speed over the entire length of grade include the entering speed, wind resistance, and the skill of the operator.

In this study, the maximum allowable gradient followed the AASHTO guidelines and was arbitrarily set at 5%.

ii) Vertical Curvature Requirements

Typical vertical curves are shown in Figure 3.4. The notations in the figure are defined as follows:

\[ VPI = \text{Vertical intersection point, or the point at which two grades join} \]

\[ g = \text{Percent grade. Positive for up-grade and negative for down-grade} \]

\[ L = \text{Length of vertical curve measured horizontally} \]

\[ VPC \text{ and } VPT = \text{Start and end points of the vertical curve} \]

\[ A = \text{Algebraic difference of consecutive grade. Positive for sag vertical curve and negative for crest vertical curve} \]

A vertical profile is made up of a series of tangent sections joined by parabolic vertical curves. The vertical curves may be classified as crest and sag types as
depicted in Figure 3.4. Vertical curves should be simple in application, safe in design, comfortable in operation, pleasing in appearance, and adequate for drainage. The major concern for safe operation on crest vertical curves is enough sight distance for the design speed. The rate of change of grade affects the comfort level of the motorists. This consideration is most important in sag vertical curves where gravitational and vertical centrifugal forces act in the same direction. Appearance is another important factor that needs to be considered. A long curve has a more pleasing appearance than a short one [AASHTO,1994]. In practice, these considerations are addressed by a careful choice of the minimum length of the vertical curve.

Figure 3.4 Typical Vertical Curves

a) Crest Vertical Curve

The minimum length of crest vertical curve as determined by sight distance
requirements generally is satisfactory from the standpoint of safety, comfort and appearance. The basic formulas for the length of a parabolic vertical curve in terms of sight distance and the algebraic difference in grade are as follows [AASHTO, 1994]:

i) When \( S \) is less than \( L \),

\[
L_{\text{min}} = \frac{AS^2}{1329} \quad \text{(in imperial units)}
\]  
\( (3.8a) \)

\[
L_{\text{min}} = \frac{AS^2}{405} \quad \text{(in SI units )}
\]  
\( (3.8b) \)

ii) When \( S \) is greater than \( L \),

\[
L_{\text{min}} = 2S - \frac{1329}{A} \quad \text{(in imperial units)}
\]  
\( (3.8c) \)

\[
L_{\text{min}} = 2S - \frac{405}{A} \quad \text{(in SI units)}
\]  
\( (3.8d) \)

where \( L \) = length of vertical curve, ft (imperial units) or meter (SI units)
\( L_{\text{min}} \) = the minimal length of vertical curve
\( S \) = sight distance, ft (Imperial units) or meter (SI units)
\( A \) = algebraic difference in grades, percent (%)

b) Sag Vertical Curve

There are at least four criteria for establishing the length of a sag vertical curve. They include headlight sight distance, rider comfort, drainage control, and requirements for general appearance. The simplified formulas for the length of a parabolic vertical curve in terms of sight distance and the algebraic difference in grade is as follows [AASHTO, 1994]:

i) When \( S \) is less than \( L \),

\[
L_{\text{min}} = \frac{AS^2}{400 + 3.5S} \quad \text{(in imperial units)}
\]  
\( (3.8e) \)
\[ L_{\text{min}} = \frac{AS^2}{122 + 3.5S} \] (in SI units) \hspace{1cm} (3.8f)

ii) When \( S \) is greater than \( L \),

\[ L_{\text{min}} = 2S - \frac{400 + 3.5S}{A} \] (in imperial units) \hspace{1cm} (3.8g)

\[ L_{\text{min}} = 2S - \frac{122 + 3.5S}{A} \] (in SI units) \hspace{1cm} (3.8h)

where \( L \) = length of vertical curve, ft (imperial units) or meter (SI units)
\( L_{\text{min}} \) = the minimal length of vertical curve
\( S \) = light beam distance, ft (imperial units) or meter (SI units)
\( A \) = algebraic difference in grades, percent (%)

For overall safety on roads, a sag vertical curve should be long enough so that the light beam distance is nearly the same as the stopping sight distance.

### 3.4.2 Horizontal Alignment

In the design of horizontal road curves it is necessary to establish the proper relation between the design speed and curvature and their relationship with the rate of super-elevation and side friction. When a vehicle moves in a circular path, it is forced radially outwards by centrifugal force. The centrifugal force can be counterbalanced by the vehicle weight component which is determined by the roadway super-elevation, or the side friction developed between tires and the road surface, or by a combination of the two. From the laws of mechanics, the basic point mass formula for vehicle operation on a curve is [AASHTO, 1994]:

\[ \frac{e + f}{1 - ef} = \frac{V^2}{15R} \] (in imperial units) \hspace{1cm} (3.9a)

\[ \frac{e + f}{1 - ef} = \frac{7.864 \times 10^{-3}V^2}{R} \] (in SI units) \hspace{1cm} (3.9b)

The minimum safe radius \( R_{\text{min}} \) can be calculated directly using [AASHTO,
Chapter Three

1994]:

\[ R_{\text{min}} = \frac{V^2}{15(e + f)} \] (in imperial units) \hspace{1cm} (3.10a)

\[ R_{\text{min}} = \frac{7.864 \times 10^{-3} V^2}{(e + f)} \] (in SI units) \hspace{1cm} (3.10b)

where \( e \) = rate of roadway super-elevation, ft/ft (in imperial units) or m/m (SI units)

\( f \) = side friction factor

\( V \) = vehicle speed, mph (in imperial units) or kmph (SI units)

\( R \) = radius of curve, ft (in imperial units) or m (SI units)

3.5 Representation of the Alignment

3.5.1 Representation of the Horizontal Alignment

The method of representing the horizontal alignment is based on that described in Jong [1998]. Let \( S(x_s, y_s) \) and \( E(x_E, y_E) \) be the start and end points of the proposed alignment and \( SE \) denotes a line connecting these two end points. The choice of decision variables to represent the horizontal alignment is based on the so
called “cut” concept. Suppose that we cut the line $SE$ $n$ times at equal intervals by a series of vertical lines as shown in Figure 3.5. The intersection points between the alignment and each vertical cut are the points defining the road alignment. Instead of directly searching for the $x_i$ and $y_i$ of the $i^{th}$ intersection point, we only need the offset $d_i$ between line $SE$ and the point of intersection. In the definition of $d_i$ the upward direction is taken as the positive direction.

For each vertical cut, the origin is defined at the intersection point of the cut line and the line $SE$. Let $O_i$ be the origin at the $i^{th}$ vertical cut, then the coordinates of $O_i$ denoted as $(x_{Oi}, y_{Oi})$ are derived as:

$$
\begin{align*}
\begin{bmatrix} x_{Oi} \\ y_{Oi} \end{bmatrix} &= \begin{bmatrix} x_s \\ y_s \end{bmatrix} + i \times D = \begin{bmatrix} x_s \\ y_s \end{bmatrix} + \frac{i}{n+1} \begin{bmatrix} x_E - x_s \\ y_E - y_s \end{bmatrix} \\
&= \begin{bmatrix} x_{Oi} \\ y_{Oi} \end{bmatrix} + 0 \\
&= \begin{bmatrix} x_{Oi} \\ y_{Oi} \end{bmatrix} + d_i
\end{align*}
$$

(3.11)

where $D = \text{length of the interval (shown in Figure 3.5)}$

$n = \text{the number of intersection points}$

Let $P_i$ be the $i^{th}$ intersection point and $d_i$ be the offset between $P_i$ and $O_i$ (upward for positive and downward for negative). Then the coordinates of $P_i$ denoted by $(x_{Pi}, y_{Pi})$ can be expressed as:

$$
\begin{align*}
\begin{bmatrix} x_{Pi} \\ y_{Pi} \end{bmatrix} &= \begin{bmatrix} x_{Oi} \\ y_{Oi} \end{bmatrix} + 0 \\
&= \begin{bmatrix} x_{Pi} \\ y_{Pi} \end{bmatrix} + d_i
\end{align*}
$$

(3.12)

The set of points $P[i]$ $i = 1, ..., n$ generally outlines the track of the alignment. For notational convenience, let $P_0$ and $P_{n+1}$ denote $S$ and $E$ respectively. Linking these intersection points by straight line sections will generate a piecewise linear trajectory. Next, circular curves tangential to each pair of adjacent straight line sections at the intersection point $P_i$ are fitted. The circular curves address the safety considerations for horizontal curves discussed in the previous section. We further
assume that the minimal allowable radius for a given design speed is used to fit the tangent sections.

Figure 3.6 Geometric Specification of a Circular Curve

Figure 3.6 shows the geometric specification of a circular curve. The geometric meaning of each variable in Figure 3.6 is shown below:

- \( P \) = intersection point
- \( C_i \) = point of curvature (beginning of the curve)
- \( T_i \) = point of tangency (end of the curve)
- \( R_i \) = radius of circular curve
- \( \Phi_i \) = centre of the circular curve
- \( \Delta_i \) = intersection angle of \( P_i \)
- \( L_{r_i} \) = tangent length from \( C_i \) to \( P_i \)
- \( L_i \) = the distance between two successive intersection points \( P_i \) and \( P_{i+1} \)
By trigonometry, we have:

\[ \Delta_i = \tan^{-1}\left(\frac{y_{P_{i+1}} - y_{P_i}}{x_{P_{i+1}} - x_{P_i}}\right) - \tan^{-1}\left(\frac{y_{P_i} - y_{P_{i-1}}}{x_{P_i} - x_{P_{i-1}}}\right) \]  

(3.13)

\[ L_{\Delta_i} = R_i \times \tan\left|\frac{\Delta_i}{2}\right| \]  

(3.14)

\[ L_i = \sqrt{(x_{P_i} - x_{P_{i+1}})^2 + (y_{P_i} - y_{P_{i+1}})^2} \]  

(3.15)

---

**Figure 3.7 An Example of Horizontal Alignment Discontinuity**

To determine the circular curve at each intersection point, we must calculate the intersection angle \( \Delta_i \) first. The tangent length \( L_{\Delta_i} \) is then computed using the minimal allowable radius \( R_{\text{min}} \). If the length \( L_i \) between any two consecutive intersection points (say \( P_i \) and \( P_{i+1} \)) is less than the sum of the tangent lengths \( L_{\Delta_i} + L_{\Delta_{i+1}} \), then a discontinuity occurs, as shown in Figure 3.7. Therefore, the radius for these two intersection points must be reduced so that the continuity condition can hold even though this might violate the minimum safety radius requirement (more on how this violation is handled within the optimization procedure, later). This verification step for every tangent segment is necessary in order to keep the continuity of the whole alignment. The determination of the horizontal curve radius is as follows:
1) Step 1: Initialization

Set $R_i = R_{\text{min}}$ for $i = 1, \ldots, n$

Calculate the intersection angle $\Delta_i$ with equation (3.13)

Calculate the tangent length $L_{T_i}$ with equation (3.14)

Calculate $L_i$ with equation (3.15)

Set $L_{T_0} = L_{T_{n+1}} = 0$

Set $i = 0$

2) Step 2: Identify discontinuous tangent sections

2.1 If $i \leq n$, then continue; otherwise STOP

2.2 If $L_i < L_{T_i} + L_{T_{i+1}}$, then continue; otherwise go to step 2.4

2.3 $\Delta t_1 = L_{T_i} + L_{T_{i+1}} - L_i$

$\Delta t_2 = L_{T_i} + L_{T_{i+1}}$

$L_{T_i} = \frac{\Delta t_1 \times L_{T_i}}{\Delta t_2}$

$L_{T_{i+1}} = \frac{\Delta t_1 \times L_{T_{i+1}}}{\Delta t_2}$

$R_i = \frac{L_{T_i}}{\tan|\Delta_i/2|}$

$R_{i+1} = \frac{L_{T_{i+1}}}{\tan|\Delta_{i+1}/2|}$

2.4 Set $i = i + 1$; go to step 2.1

With the above procedure, we can then generate a unique horizontal alignment for a given set of decision variables $d_i, (i = 1, \ldots, n)$. The resulting alignment is composed of tangent sections and circular curves. It is important to reiterate that this
alignment may violate the minimal allowable radius constraint. However, the optimization procedure includes a mechanism to penalize such violations and a final check to flag such violations in any particular proposed alignment.

3.5.2 Representation of the Vertical Alignment

In general, the vertical alignment usually consists of a series of straight lines (tangent) joined to each other by parabolic curves. The starting point for the determination of the vertical alignment (or profile) is a candidate horizontal alignment. The vertical alignment is defined in a curvilinear orthogonal plane running longitudinally along the proposed horizontal alignment. The representation and procedure of construction of the vertical alignment follows very much that of the horizontal alignment discussed previously. A series of vertical lines, like AB in Figure 3.8, which are perpendicular to the proposed horizontal alignment and spaced at equal intervals apart is introduced for the purpose of determining the vertical profile. The vertical alignment is defined by a series of vertical intersection points $VPI_i$ along these vertical lines. Instead of directly use the vertical elevation of $VPI_i$, we use the gradient $g_i$ between two consecutive intersection points $VPI_{i-1}$ and $VPI_i$. Using the connecting gradient instead of the absolute elevations reduces the search space and simplifies the checking for infeasible vertical profiles. Connecting the start and end points of the alignment with straight lines through the series of intersection points will then yield a piecewise linear vertical trajectory. An iterative procedure is then employed to fit parabolic curves at each intersection point so that the alignment is smooth and continuous. A typical vertical alignment is shown in Figure 3.8.

Where $S$ and $E$ = the start and end points of the alignment

$$VPI_i = \text{the } i^{\text{th}} \text{ vertical intersection point}$$
\[ g_i = \text{the gradient between intersection points } VPI_{i-1} \text{ and } VPI_i \]

At the intersection point where the intersection angle is not zero, a parabolic curve is inserted. Since a tangent segment is bounded by two adjacent intersection points, their curve lengths are interdependent. Ideally, a tangent must be long enough to accommodate the parabolic curve lengths required by design standards. However, in some situations, the length between two successive intersection points (say \( VPI_i \) and \( VPI_{i+1} \)) may not be long enough to accommodate the minimal length of vertical curve at \( VPI_i \) and \( VPI_{i+1} \) (see Figure 3.9 as an example). Then a discontinuity occurs, which violates the alignment definition. To avoid such a condition, additional constraints are required:

\[ \frac{L_{\text{min, }i}}{2} + \frac{L_{\text{min, }i+1}}{2} \leq l_i \]  

(3.16)

where \( L_{\text{min, }i} = \) the minimal length of vertical curve at \( VPI_i \) calculated by equation 3.8

\[ l_i = \text{the horizontal distance between } VPI_i \text{ and } VPI_{i+1} \]
Chapter Three

Figure 3.9 Discontinuous Situation of Vertical Alignment

If the tangent is too short, the parabolic curve lengths at both ends must be reduced to avoid a discontinuous vertical alignment. The new parabolic curve lengths of two adjacent intersection points are:

\[
L_i = L_{\text{min}, i} - \left( \frac{L_{\text{min}, i} + L_{\text{min}, i+1}}{2} - d_i \right) \times \frac{L_{\text{min}, i}}{L_{\text{min}, i} + L_{\text{min}, i+1}} \quad (3.17a)
\]

\[
L_{i+1} = L_{\text{min}, i+1} - \left( \frac{L_{\text{min}, i} + L_{\text{min}, i+1}}{2} - d_i \right) \times \frac{L_{\text{min}, i+1}}{L_{\text{min}, i} + L_{\text{min}, i+1}} \quad (3.17b)
\]

The parabolic curve fit procedure is similar as the horizontal one mentioned in the previous section. With the above procedure, we can then generate a unique vertical alignment for a given set of decision variables \( g_i, (i = 1, \ldots, n) \). The result vertical alignment is smooth and continuous everywhere. It is important to reiterate that this alignment may violate the minimal length of vertical curve constraint. However, the optimization procedure includes a mechanism to penalize such violations and a final check to flag such violations in any particular proposed alignment.

3.6 Summary

The purpose of this section is to give a concise description of the mathematical optimization problem for 3D road alignment addressed by this research. The description consists of the relevant decision variables, objective functions and
Chapter Three

constraints.

Minimize: \( C_{V}(d_i, g_j) + C_{\text{Land}}(d_i) + C_{p}(d_i) \)

Subject to: 1) \( -g_{\text{max}} \leq g_j \leq g_{\text{max}} \), for \( j = 1, \ldots, m \)

2) \( L_j \geq L_{\text{min}} \), for \( j = 1, \ldots, m \)

3) \( R_i \geq R_{\text{min}} \), for \( i = 1, \ldots, n \)

where \( C_{V}(d_i, g_j) \) = the earthwork volume cost, which is a function of variables \( d_i \) and \( g_j \)

\( C_{\text{Land}}(d_i) \) = the land use cost, which is a function of variable \( d_i \)

\( C_{p}(d_i) \) = the pavement cost, which is a function of variable \( d_i \)

\( d_i \) = offset along the cut line for horizontal intersection point

\( R_i \) = the radius of \( i^{\text{th}} \) horizontal intersection point

\( R_{\text{min}} \) = minimal allowable radius

\( g_j \) = the gradient between two consecutive vertical intersection point

\( g_{\text{max}} \) = maximal allowable vertical gradient

\( L_j \) = the length of vertical curve

\( L_{\text{min}} \) = minimal length of vertical curve
CHAPTER 4 OPTIMIZING ROAD ALIGNMENTS

In this chapter, a solution algorithm is presented to solve the 3D road alignment problem. This algorithm is composed of separate algorithms for the horizontal and vertical alignment optimization. These two algorithms (discussed in Sections 4.1 and 4.2) can be used on their own or in combination to solve the 3D road alignment problem. The method used to combine the two algorithms is based on a bi-level optimization scheme discussed in Section 4.3. Case studies are presented at each of the three sections to gauge the performance of the algorithms. All the GA optimization program was developed based on a GA library “PGAPack” [Levine,1996].

4.1 Genetic Algorithms for Optimizing the Horizontal Alignment

The following sections discuss the key steps of the GA-based procedure (shown in Figure 4.1) that was adopted for horizontal alignment optimization.

![Figure 4.1 GA-based Procedure for Horizontal Alignment Optimization](image-url)
4.1.1 Genetic encoding

In the GA representation, the solution to the problem is represented as a string of genes called a chromosome. Each gene represents one of the intersection points of the proposed horizontal alignment and the content of that gene encodes a value of $d_i$ (defined in Chapter 3). Therefore, the length of the chromosome string is as long as the maximum number of intersection points allowed for the alignment.

For notational convenience, in this thesis we refer to a chromosome by $\Omega$ and an individual gene by $\tau$ subscripted by its location. For example, a six-gene chromosome may be represented by $\Omega = [\tau_1, \tau_2, \ldots, \tau_6]$.

For the horizontal alignment optimization problem, an integer point encoding is employed to represent the offset along the cut line.

$$\Omega = [\tau_1, \tau_2, \ldots, \tau_n] = [d_1, d_2, \ldots, d_n]$$

(4.1)

where $d_i$ = the offset along the cut line of the $i^{th}$ intersection point, positive upwards

$n$ = total number of the intersection points

In the above equation, the alleles of the $i^{th}$ gene will be selected within the interval $[d_{low}, d_{up}]$, where $d_{low}$ and $d_{up}$ are the lower and upper bound of $d_i$. We can obtain $d_{low}$ and $d_{up}$ from Figure 3.5:

$$\begin{align*}
    d_{up} &= y_{max} - y_S \\
    d_{low} &= y_O - y_S
\end{align*}$$

(4.2)

The maximum number of intersection points of the proposed horizontal alignment is fixed in equation 4.1. However, the actual number of intersection points needed to define the alignment varies depending on the terrain condition and land-use patterns [Chan & Fan,
Chapter Four

2003]. This is achieved by adopting the use of special empty / dummy cells to indicate the absence of an intersection point at a particular cut line:

\[
\Omega = [r_1, r_2, \ldots, r_{n-1}, r_n] = [d_1, d_2, \ldots, X, \ldots, X, \ldots, d_{n-1}, d_n]
\]  

(4.3)

where \( X \) = an empty gene

The value for empty genes \( X \) should be chosen so that it is not within the range \([d_{\text{low}}, d_{\text{up}}]\).

4.1.2 Initial Population

In order that as much of the search space is explored, the initial population is randomly generated to keep the gene pool as diverse as possible. However, if an engineer has some initial guesses about the solution, they might be included as well. Without loss of generality, we assume that no prior knowledge about the solution is available in this research. Therefore, the population can be generated as follows:

\[
d_i = \begin{cases} 
    r_c [d_{\text{low}}, d_{\text{up}}], & \forall i = 1, \ldots, n \\
    X & \end{cases}
\]

(4.4)

4.1.3 Fitness Function

As stated in section 3.2.3, the costs included in horizontal alignment problem are land use cost and pavement cost. Therefore, the fitness function can be defined as:

\[
C_{\text{hor}} = C_{\text{Land}} + C_p + C_{\text{pen}_\text{hor}}
\]

(4.5)

where \( C_{\text{hor}} \) = fitness function of the proposed horizontal alignment

\( C_{\text{Land}} \) = land use cost of the proposed horizontal alignment

\( C_p \) = pavement cost of the proposed horizontal alignment

\( C_{\text{pen}_\text{hor}} \) = penalties for the violation of horizontal alignment constraints

The computation of the fitness function involves a complicated procedure to determine the domain cells through which the horizontal alignment passes (for the purpose
of determining the land use costs). First, the corresponding intersection points of the chromosome must be decoded by equation 3.12. Next, given the set of intersection points, we generate the corresponding alignment (consisting of tangent sections and circular curves) using the procedure described in section 3.5.1. Once the horizontal alignment elements have been determined, we can calculate its associated cost using the procedure described in Appendix A.

4.1.4 Selection and replacement

The reproductive chance of each individual is determined by its fitness function – in this study, lower values for the objective function $C_{hor}$ denote fitter individuals which will have a higher probability of being selected to reproduce offspring. There are many methods to select chromosomes and allocate reproductive chances including roulette wheel selection, Boltzmann selection, tournament selection and ranking selection. We use ranking selection in this study because it avoids both pre-convergence during the early generations and random search in later generations [Michalewicz 1996].

4.1.5 Genetic operators

The performance of evolutionary programs is highly dependent on their genetic operators through which the population evolves to become increasingly adapted to the problem. Crossover operators combine the features of two parent chromosomes to form two offspring, while mutation operators arbitrarily alter one or more genes of a selected chromosome to create a new chromosome. Three genetic operators are used in this model.

i) One-point crossover.

Let two parents $\Omega_i = [d_{i1}, d_{i2}, \ldots, d_{i(n-1)}, d_{in}]$ and $\Omega_j = [d_{j1}, d_{j2}, \ldots, d_{j(n-1)}, d_{jn}]$ be crossed after a randomly generated position $k$, then the resulting offspring are:

$$\Omega_i' = [d_{i1}, d_{i2}, \ldots, d_{i(k-1)}, d_{j(k)}, \ldots, d_{j(n-1)}, d_{in}]$$

$$\Omega_j' = [d_{j1}, d_{j2}, \ldots, d_{j(k-1)}, d_{i(k)}, \ldots, d_{i(n-1)}, d_{jn}]$$
ii) Uniform mutation [Michalewicz, 1996]

Let \( \Omega = [d_1, d_2, \ldots, d_{n-1}, d_n] \) be the chromosome to be mutated at the encoded genes of the \( i^{th} \) intersection point. Then \( d_i \) will be replaced by:

\[
d'_i = \begin{cases} 
[d_{\text{low}}, d_{\text{up}}] \\
X 
\end{cases}
\]

iii) Non-uniform mutation [Michalewicz, 1996]

Let \( \Omega = [d_1, d_2, \ldots, d_{n-1}, d_n] \) be the chromosome to be mutated at the encoded genes of the \( i^{th} \) intersection point. Then \( d_i \) will be replaced by:

\[
d'_i = \begin{cases} 
d_i + \Delta(t, d_{\text{up}} - g_i) \\
d_i - \Delta(t, d_i + d_{\text{low}}) \\
X 
\end{cases}
\]

\[
\Delta(t, y) = y \times r \times (1 - \frac{t}{T})^b
\]

where \( t = \) current generation number

\( T = \) maximum generation number

\( r = \) random number within the region \([0,1]\)

\( b = \) degree of non-uniformity, we use \( b = 1 \) in this research.

4.1.6 Convergence

There are three candidate conditions typically used as stopping criteria: 1) iteration limit exceeded, 2) population too similar, and 3) no change in the best solution found in a given number of iterations; this study used the first criterion as it was the simplest of the three stopping criteria.

4.1.7 Case study

In this section, we intend to investigate the performance of the proposed solution algorithm for horizontal alignment by running a test case. The domain for the test case is
designed such that the optimal or near optimal alignment is fairly obvious. A map of the
test domain is shown in Figure 4.2.

![Figure 4.2 The Test Domain](image)

The test domain is a $2100m \times 1000m$ area which is partitioned into equal sized cells
$100m \times 100m$ in dimension. The two dots in the map represent the two given end points of
the proposed horizontal alignment. In Figure 4.2, the darker shaded cells represent
locations where the land use cost is higher. A visual inspection of the map indicates that
the final alignment must skirt the high cost cells to minimize the total cost. The minimal
allowable radius used in this study is 300 meters and follows the AASHTO [1994]
guidelines.

### 4.1.7.1 Sensitivity study of GA parameters

There are three important control parameters of a single GA which include
population size (number of individuals in the population), crossover rate and mutation rate.
A sensitivity study was carried out in this study to find the optimum GA parameters for this
example problem on horizontal alignment analysis.

An important GA parameter is the population size. A total of ten pool sizes were
considered in this study. The parent pool size ranged from 20 to 200 in increments of 20.
The results are shown in Figure 4.3. It can be observed from Figure 4.3 that the total cost
fell rather quickly with the increase in population size until a pool size of 80. GA solutions showed little variation for pool sizes beyond 100. Hence, a population size of 100 was adopted for this problem.

![Figure 4.3 Sensitivity Study of Population Size on Horizontal Alignment Analysis](image)

Studies were also done on different values of mutation rate and crossover rate. The results are presented in Figure 4.4 and Figure 4.5. All the curves obtained are relatively flat. Based on the above results, the mutation rate and crossover rate selected for this study were 0.2 and 0.7 respectively.

![Figure 4.4 Sensitivity Study of Mutation Rate on Horizontal Alignment Analysis](image)
Chapter Four

Figure 4.5 Sensitivity Study of Crossover Rate on Horizontal Alignment Analysis

Settings for the GA used for the optimization are given in Table 4.1.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Maximum number of intersection points</td>
<td>6</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.2</td>
</tr>
<tr>
<td>Uniform mutation proportion</td>
<td>0.4</td>
</tr>
<tr>
<td>Non-uniform mutation proportion</td>
<td>0.6</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>0.7</td>
</tr>
<tr>
<td>Maximum number of generations</td>
<td>100</td>
</tr>
</tbody>
</table>

4.1.7.2 Result of the case study

Figure 4.6 The Best Horizontal Alignment at the 100th Generation
Figure 4.6 shows the best horizontal alignment obtained at the 100\textsuperscript{th} generation. There are no constraint violations for this solution. The final horizontal alignment winds its way through the low cost cells and skirts the high cost cells to minimize the total cost. The number of actual intersection points in this alignment is four (indicated by IP1, IP2, IP3 and IP4 in the figure) even though a total of six possible intersection points (spaced 300m apart) was defined in the chromosome. This shows that the algorithm has the ability to vary the number of intersection points used depending on the land-use cost encountered.

The objective function values in each generation for a typical run are plotted in Figure 4.7. It shows that in the initial stage of the search, the objective value is extremely high, possibly due to poor choice of alignment alignments and imposition of penalty costs for constraint violations. After about five generations, the objective value drops sharply from about $1.3 \times 10^6$ to $2.3 \times 10^5$. After about 65 generations, the objective value is very close to the optimal solution found at the 100\textsuperscript{th} generation.
The example shows that the proposed GA-based algorithm works very well. It can very quickly and consistently improve the quality of the solutions for the horizontal alignment problem with 100 generations. The final horizontal alignment is continuous everywhere. The number of intersection points is variable depending on the terrain condition and land use patterns.

4.2 Genetic Algorithms for Optimizing the Vertical Alignment

The purpose of this section is to describe the method used to find the optimal vertical alignment for any pre-determined horizontal alignment. This section begins with a description of the data preparation steps for the proposed method followed by a description of the GA-based procedure (shown in Figure 4.8) for optimizing vertical alignment in the subsequent sections. A case study is presented in section 4.2.7 to investigate the performance of the proposed solution algorithm.

![Figure 4.8 GA-based Procedure for Vertical Alignment Optimization](image-url)
4.2.1 Data preparation

The data needed for vertical alignment optimization along a pre-selected horizontal alignment includes the length of the horizontal alignment and the ground profile along the selected alignment.

i) Length of the horizontal alignment

The length of any pre-determined horizontal alignment can be computed using equation A11 once the intersection points along the horizontal alignment are determined. This length is used to mark intermediate positions along the alignment by referring to the elapsed distance between a fixed starting point and the intermediate point.

ii) Ground profile along the horizontal alignment

The horizontal alignment is composed of a series of tangent lines and circular curves determined using the 2-step procedure described previously in section 3.5.1. The ground profile is determined by finding the height of the terrain at intermediate ground points, located between the start and end points of the horizontal alignment, which are spaced at equal distances apart. Cross sections at these selected intermediate locations are used to calculate the earthwork volume. This requires a procedure to determine the coordinates of these intermediate ground points located on the tangent lines and circular curves of the horizontal alignment. The details of the procedure to do this are described in Appendix C.

4.2.2 Genetic encoding

For the vertical alignment optimization problem, the chromosome represents a set of variables $g_j$:

$$\Omega = [\tau_1, \tau_2, \ldots, \tau_{m-1}, \tau_m] = [g_1, g_2, \ldots, g_{m-1}, g_m] \quad (4.6)$$

where $g_j$ = the gradient of $j^{th}$ segment connecting $VPI_j$ and $VPI_{j+1}$

$m$ = total number of vertical intersection points
Each gene represents one of the intersection points of the proposed vertical alignment and the content of that gene encodes a value of $g_j$ (defined in Chapter 3). The intersection points are spaced equally apart along the horizontal alignment. Vertical cut lines are imagined at each of these intersection points. The required elevation of the vertical alignment at an intersection point $j$ is determined by the point at which a line extending from the previous vertical elevation point with gradient $g_j$ intersects the vertical cut line. The length of the chromosome string is as long as the maximum number of intersection points allowed for the alignment. In Equation 4.6, the alleles of the $j^{\text{th}}$ gene will be selected within the interval $[-g_{\text{max}}, g_{\text{max}}]$, where $g_{\text{max}}$ is the maximum allowable gradient according to AASHTO [1994] guidelines.

The maximum number of intersection points of the proposed vertical alignment is fixed at the beginning of the optimization. However, the actual number of intersection points that are eventually used to define the vertical alignment is allowed to vary depending on the terrain condition and land-use patterns [Chan & Fan, 2003] and is determined dynamically by the GA procedure, much in the same way as the number of intersection points used for the horizontal alignment. This is achieved by adopting the use of special empty / dummy cells to indicate the absence of an intersection point at a particular cut line:

$$\Omega = [r_1, r_2, \ldots, r_{m-1}, r_m] = [g_1, g_2, \ldots, Y, \ldots, Y, \ldots, g_{m-1}, g_m]$$

(4.7)

where $Y$ = an empty gene

A special value that is not within the range $[-g_{\text{max}}, g_{\text{max}}]$ is used to encode for the empty gene $Y$.

4.2.3 Initial population
In order that as much of the search space is explored, the initial population is randomly generated to keep the gene pool as diverse as possible. Following the example of [Fwa et al 2002], a “big envelope” (shown in Figure 4.9) which represents the feasible search space for the proposed vertical alignment is defined in order to keep all the chromosomes feasible in the initial population.

### 4.2.4 Fitness Function

The calculation procedure for each individual’s value of the fitness function consists of 4 steps:

i) Determine the road design elevation \( d_{E_j} \) at the location of the selected ground points using the procedure described in Appendix D.

ii) Determine the depth of cut / fill at the location of the selected ground points

\[
h_j = d_{E_j} - gp_{E_j}
\]

where \( h_j \) = depth of cut / fill

\( d_{E_j} \) = the design elevation

\( gp_{E_j} \) = the ground elevation
iii) Determine the earthwork volume

The fit volume \( V_{fill} \) and cut volume \( V_{cut} \) can be calculated by equation 3.4 when the depth of cut / fill at the location of the selected ground points is obtained. The earthwork cost \( C_v \) can then be obtained from equation 3.6.

iv) Calculate the fitness value of each individual

The final value of the proposed vertical alignment is obtained by:

\[
C = \sum C_v + \sum C_{pen \_ver}
\]  

(4.11)

where \( C_v \) = the cost of the earthwork

\( C_{pen \_ver} \) = penalties for the violation of vertical constraints

The vertical constraints considered in this research are the maximum allowable gradient and vertical curvature requirements.

### 4.2.5 Genetic operators

The performance of evolutionary programs is highly dependent on their operators, through which the population evolves to become increasingly adapted to the problem. Six problem-specific genetic operators are developed in this study to help the performance of the problem:

i) Uniform mutation [Michalewicz, 1996]

Let \( \Omega = [g_1, g_2, \ldots, g_{n-1}, g_n] \) be the chromosome to be mutated at the encoded genes of the \( i^{th} \) intersection point. Then \( g_i \) will be replaced by:

\[
g_i = \begin{cases} 
-g_{\max}, & g_{\max} \\
X, & \text{random value}
\end{cases}
\]

ii) Non-uniform mutation [Michalewicz, 1996]

Let \( \Omega = [g_1, g_2, \ldots, g_{n-1}, g_n] \) be the chromosome to be mutated at the encoded genes of the \( i^{th} \) intersection point. Then \( g_i \) will be replaced by:
\[ g_i' = \begin{cases} 
  g_i + \Delta(t, g_{\max} - g_i) \\
  g_i - \Delta(t, g_i + g_{\max}) \\
  X
\end{cases} \]

where \( \Delta(t, y) = y \times r \times (1 - \frac{t}{T})^{b} \)

- \( t \) = current generation number
- \( T \) = maximum generation number
- \( r \) = random number within the region \([0,1]\)
- \( b \) = degree of nonuniformity, we use 1 in this research

iii) One point crossover

Let two parents \( \Omega_i = [g_{i1}, g_{i2}, \ldots, g_{i(n-1)}, g_{i(n)}] \) and \( \Omega_j = [g_{j1}, g_{j2}, \ldots, g_{j(n-1)}, g_{j(n)}] \) be crossed after a randomly generated position \( k \), then the resulting two offspring are:

- \( \Omega_i' = [g_{i1}, g_{i2}, \ldots, g_{i(k)}, g_{j(k+1)}, \ldots, g_{j(n-1)}, g_{j(n)}] \)
- \( \Omega_j' = [g_{j1}, g_{j2}, \ldots, g_{j(k)}, g_{i(k+1)}, \ldots, g_{i(n-1)}, g_{i(n)}] \)

iv) Two point crossover

Let \( \Omega_i = [g_{i1}, g_{i2}, \ldots, g_{i(n-1)}, g_{i(n)}] \) and \( \Omega_j = [g_{j1}, g_{j2}, \ldots, g_{j(n-1)}, g_{j(n)}] \) be the two parents to be crossed between positions \( k \) and \( l \). The resulting two offspring are:

- \( \Omega_i' = [g_{i1}, g_{i2}, \ldots, g_{i(k)}, g_{j(k+1)}, \ldots, g_{j(l)}, g_{i(l+1)}, \ldots, g_{i(n-1)}, g_{i(n)}] \)
- \( \Omega_j' = [g_{j1}, g_{j2}, \ldots, g_{j(k)}, g_{i(k+1)}, \ldots, g_{i(l)}, g_{j(l+1)}, \ldots, g_{j(n-1)}, g_{j(n)}] \)

v) Arithmetical crossover

The operator is introduced in Michalewicz’s GENOCOP system [1996] for numerical optimization. The offspring are generated through linear combinations of their parents. Let \( \Omega_i \) and \( \Omega_j \) be two parents for the arithmetic crossover operator, \( \lambda_{i(k)} \) and
\( \lambda_{j(k)} \) be the \( k^{th} \) gene of the chromosome. There are 3 different cases because of the specific representation of the chromosome:

Case 1: \( \lambda_{i(k)} \neq X \) and \( \lambda_{j(k)} \neq X \), then

\[
\begin{align*}
\lambda'_{i(k)} &= k_1 \lambda_{i(k)} + k_2 \lambda_{j(k)} \\
\lambda'_{j(k)} &= k_2 \lambda_{i(k)} + k_1 \lambda_{j(k)}
\end{align*}
\]

where \( k_1 + k_2 = 1 \), \( k_1 \geq 0 \) and \( k_2 \geq 0 \)

Case 2: \( \lambda_{i(k)} = X \) or \( \lambda_{j(k)} = X \), then

\[
\begin{align*}
\lambda'_{i(k)} &= \lambda_{i(k)}, \quad \lambda'_{j(k)} = \lambda_{j(k)}, \quad \text{or} \\
\lambda'_{i(k)} &= \lambda_{j(k)}, \quad \lambda'_{j(k)} = \lambda_{i(k)}
\end{align*}
\]

Case 2: \( \lambda_{i(k)} = X \) and \( \lambda_{j(k)} = X \), then

\[
\begin{align*}
\lambda'_{i(k)} &= \lambda_{i(k)}, \quad \lambda'_{j(k)} = \lambda_{j(k)}
\end{align*}
\]

vi) Direction-based crossover

Let the two parents to be crossed by this operator be denoted by \( \Omega_i \) and \( \Omega_j \), where we assume that \( f(\Omega_i) \leq f(\Omega_j) \) (i.e. \( \Omega_i \) is at least as good as \( \Omega_j \)). Intuitively, one may think that moving \( \Omega_i \) along \( \Omega_i - \Omega_j \) may yield a better solution. Using this idea, the operator generates offspring according to the following rule. The operator is also divided into 3 different cases because of the same reason as the former operator.

Case 1: \( \lambda_{i(k)} \neq X \) and \( \lambda_{j(k)} \neq X \), then

\[
\lambda'_{k} = \lambda_{i(k)} + r(\lambda_{i(k)} - \lambda_{j(k)})
\]

where \( r \) is a random number within the region \([0,1]\)

Case 2: \( \lambda_{i(k)} = X \) or \( \lambda_{j(k)} = X \), then

\[
\begin{align*}
\lambda'_{i(k)} &= \lambda_{i(k)}, \quad \lambda'_{j(k)} = \lambda_{j(k)}, \quad \text{or}
\end{align*}
\]
$\lambda_{\alpha(k)} = \lambda_{\alpha(k)}, \hat{\lambda}_{\alpha(k)} = \hat{\lambda}_{\alpha(k)}$

Case 2: $\lambda_{\alpha(k)} = X$ and $\lambda_{\beta(k)} = X$, then

$\tilde{\lambda}_{\alpha(k)} = \tilde{\lambda}_{\alpha(k)}, \tilde{\lambda}_{\beta(k)} = \tilde{\lambda}_{\beta(k)}$

In case 1, this operator may generate an offspring away from the feasible region $[-g_{\text{max}}, g_{\text{max}}]$. In such a case, the upper bound $g_{\text{max}}$ or the lower bound $-g_{\text{max}}$ is used to in case of $\lambda_{k} > g_{\text{max}}$ or $\lambda_{k} < -g_{\text{max}}$, respectively.

### 4.2.6 Convergence

The stopping criterion of the proposed algorithm for the vertical alignment optimization is the same as the horizontal one.

### 4.2.7 Case study

![3D View of the Test Domain](image)

**Figure 4.10 3D View of the Test Domain**

The same test domain as the one for horizontal alignments optimization is used in order to keep the continuity of the research. The 3D view of the test domain is shown in Figure 4.10. It can be seen that there are two small hills in the study region. The region is divided into equal sized cells $(50m \times 50m)$ to store different height data so that it can be
used in the proposed approach. In Figure 4.11, the darker shaded cells represent locations where the ground elevation is higher.

![Figure 4.11 Ground Elevation of the Test Domain](image)

A horizontal alignment is randomly generated within the study region and checked for feasibility with respect to the horizontal alignment constraints. This alignment is then used as the basis to test the approach for optimizing the vertical alignment.

### 4.2.7.1 Sensitivity analysis of GA parameter

A sensitivity study was carried out in this section to find the optimum GA parameters for this example problem on vertical alignment analysis.

A total of twenty pool sizes were considered in this study to determine the optimum population size. The parent pool size ranged from 50 to 1000 in increments of 50. The results are shown in Figure 4.12. It can be observed from Figure 4.12 that the total cost fell rather quickly with the increase in population size until a pool size of 450. GA solutions showed little variation for pool sizes beyond 500. Therefore, a population size of 500 was adopted for this problem.
Studies were also done on different values of mutation rate and crossover rate. The results are presented in Figure 4.13 and Figure 4.14. All the curves obtained are relatively flat. Based on the above results, the mutation rate and crossover rate selected for this study were 0.2 and 0.7 respectively.
Figure 4.14 Sensitivity Study of Crossover Rate on Vertical Alignment Analysis

Settings for the program used for the optimization are given in Table 4.2.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Proposed Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum allowable gradient</td>
<td>5%</td>
</tr>
<tr>
<td>Sight distance</td>
<td>122m</td>
</tr>
<tr>
<td>Population size</td>
<td>500</td>
</tr>
<tr>
<td>Number of vertical intersection points</td>
<td>m</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.2</td>
</tr>
<tr>
<td>Uniform mutation proportion</td>
<td>0.4</td>
</tr>
<tr>
<td>Non-uniform mutation proportion</td>
<td>0.6</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>0.7</td>
</tr>
<tr>
<td>One point crossover proportion</td>
<td>0.48</td>
</tr>
<tr>
<td>Two point crossover proportion</td>
<td>0.48</td>
</tr>
<tr>
<td>Arithmetical crossover proportion</td>
<td>0.02</td>
</tr>
<tr>
<td>Direction-based crossover proportion</td>
<td>0.02</td>
</tr>
<tr>
<td>Maximum number of generations</td>
<td>200</td>
</tr>
</tbody>
</table>

Note: $m = \lfloor (L_{\text{total}} + d / 2) / d \rfloor + 1$ the maximum number of vertical intersection points

$d$ = the interval between two consecutive vertical intersection points in the proposed model. ($d = 50m$ in this example)

$\lfloor \cdot \rfloor$ denotes the truncated integer value of its argument

$L_{\text{total}}$ = the length of the particular horizontal alignment
Chapter Four

The horizontal alignment and its associated optimal vertical alignment obtained by the proposed approach are shown in Figure 4.15. There are no constraint violations for this solution based on the fact that no penalty terms were included in the final value of the objective function. The final vertical alignment follows the ground profile very closely thus minimizing the amount of earthwork excavation and embankment.

![Figure 4.15 Horizontal Alignment and its associated Optimal Vertical Alignment](image)

4.3 Bi-level Genetic Algorithm for Optimizing the 3D Road Alignment
4.3.1 Bi-level Formulation of the 3D Road Alignment Optimization Problem

The 3D road alignment is a line defined in 3D space. The projection of the 3D alignment onto the $XY$ plane becomes the horizontal alignment whilst its projection onto the surface orthogonal to the $XY$ horizontal plane containing the horizontal alignment determines the vertical alignment. Most of the existing approaches [for example, Jong 1998; Hogan 1973; Goh, Chew, Fwa, 1988] optimize both the horizontal and vertical alignments simultaneously. Any approach which tries to optimize both the horizontal and vertical alignments simultaneously must have some assumptions of the relation between the horizontal and its associated vertical alignments. For example, Jong [1998] assumed that the number and position of the vertical intersection points are the same as the horizontal intersection points. However, in real engineering problem, the number and position of the vertical intersection points can vary depending on the terrain condition after the horizontal alignment is determined [Chan & Fan, 2003]. Assuming that the two sets of intersection points are the same has the effect of restricting the search area to a subset of the entire feasible solution area which may result in a solution of lower quality.

Due to the non-linear interaction between horizontal and vertical alignments, and elements of the total cost, the best 3D alignment cannot be obtained by combining the best horizontal alignment and the best vertical alignment. It is necessary to search among the possible combinations of vertical and horizontal alignments for the best combination. This is the purpose of the bi-level GA approach developed in this study to optimize the 3D alignment. The bi-level optimization problem is a hierarchical optimization problem where a subset of the variables is constrained to be the solution of another optimization problem parameterized by the remaining variables. A bi-level optimization problem is a multilevel problem with two levels.
In mathematical terms, the bi-level program for the 3D road alignment optimization problem can be expressed as:

\[
\begin{align*}
\text{Minimize: } & C_v(d_i, g_j) + C_{\text{Land}}(d_i) + C_p(d_i), \text{ for } i = 1, \ldots, n \quad j = 1, \ldots, m \\
\text{Subject to: } & 1) \quad R_i \geq R_{\min}, \text{ for } i = 1, \ldots, n \\
\end{align*}
\]  

(4.12)

Where \( g_j \), for each set of value of \( d_i \), is the solution of the lower level problem:

\[
\begin{align*}
\text{Minimize: } & C_v(d_i, g_j) \quad \text{for } i = 1, \ldots, n \quad j = 1, \ldots, m \\
\text{Subject to: } & 1) \quad -g_{\max} \leq g_j \leq g_{\max}, \text{ for } j = 1, \ldots, m \\
& 2) \quad L_j \geq L_{\min}, \text{ for } j = 1, \ldots, m \\
\end{align*}
\]  

(4.13)

where \( C_v(d_i, g_j) + C_{\text{Land}}(d_i) + C_p(d_i) = \) total cost of the alignment which is determined by \( d_i \) and \( g_j \)

\( C_v(d_i, g_j) = \) earthwork volume cost of the alignment which is determined by \( d_i \) and \( g_j \)

\( C_{\text{Land}}(d_i) = \) the land use cost of the alignment which is determined by \( d_i \)

\( C_p(d_i) = \) the pavement cost of the alignment which is determined by \( d_i \)

\( d_i = \) offset along the cut line for \( i^{th} \) horizontal intersection point

\( R_i = \) the radius of \( i^{th} \) horizontal intersection point

\( R_{\min} = \) minimal allowable radius

\( g_j = \) the gradient between two consecutive vertical intersection point

\( g_{\max} = \) maximum allowable vertical gradient

\( L_j = \) the length of vertical curve

\( L_{\min} = \) minimal length of vertical curve

The bi-level GA-based procedure for 3D road alignment optimization is shown in
Figure 4.16 Bi-level GA-based Procedure for 3D Alignment Optimization

Where $C = \text{the fitness function of the particular horizontal with optimal vertical alignment } V_{optimal \_i}$

$C_{\text{ver}} = \text{the fitness function of the vertical alignment for the particular horizontal alignment}$

$C_{\text{Land}} = \text{land use cost of the alignment}$

$C_{\rho} = \text{pavement cost of the alignment}$

$C_{V} = \text{earthwork volume cost of the alignment}$

$C_{\text{pen\_ver}} = \text{penalties for the violation of vertical alignment constraints}$

$C_{\text{pen\_hor}} = \text{penalties for the violation of horizontal alignment constraints}$
The search procedure of the upper level is similar to the procedure for horizontal alignment optimization described in section 4.1. The only difference is the calculation of the fitness function of each individual. In section 4.1, only the land use cost $C_{\text{land}}$ and pavement cost $C_p$ are included whereas for the upper level bi-level program, the earthwork cost $C_{\text{ver}}$ is included after the conclusion of the lower level program. The bi-level program proceeds by transferring to the lower level program the set of horizontal alignment data $d_i$ (for $i = 1, \ldots, n$) for each and every horizontal alignment in the upper level program. The lower level program works out the length and ground profile for any particular horizontal alignment. A GA-based program is then used to obtain the optimal vertical alignment (in terms of earthwork volume costs) for this horizontal alignment. At the end of the GA search, the lower level program will transfer the earthwork volume cost of the best vertical alignment $C_{\text{ver}}$ obtained to the upper level. This is repeated for all the other horizontal alignments in the population pool of the upper level program. With a sequential processor, it is only possible to do this one at a time but with parallel processing, several lower level programs can be started to do the lower level search simultaneously. Finally, the fitness function of the individuals in the upper level can be computed as:

$$C = C_{\text{land}} + C_p + C_{\text{ver}} + C_{\text{pen}_\text{ver}} + C_{\text{pen}_\text{hor}}$$

(4.14)

The reproductive chance of each individual is determined by its fitness function – in this study, lower values for the objective function $C$ computed by equation 4.14 denote fitter individuals which will then have a higher probability of being selected to reproduce offspring. Some selected individuals will then undergo reproduction by means of crossover and mutation to form new solutions. A user specified generation number is used to stop the program.

4.3.2 Performance of the Bi-level Program
This section describes the results of some test runs to investigate the performance of the proposed bi-level GA program for 3D alignment optimization. The test domain is the same as that used in the previous sections.

![Figure 4.17](image)

**Figure 4.17 Objective Values (of earthwork costs) through successive Generations**

It was felt that the convergence of the lower-level program (on earthwork volume costs) would depend on the horizontal alignment adopted. To test this conjecture, several feasible horizontal alignments were randomly generated by a program within the search area and the GA-search procedure in the lower level is used to optimize the vertical alignment for these horizontal alignments. Figure 4.17 shows the objective values in each generation for these horizontal alignments marked as Alignments 1-6. It shows that the convergence in all these six cases was largely similar although they all converged to different asymptotic values. In the initial stage of the search, both the objective values are extremely high. The objective values then drop sharply after about 5-10 generations. Finally, the objective values converge to their respective asymptotic values. The iteration / generation beyond which there is no significant improvement in the objective function value is different for each of the different horizontal alignments. However, an inspection of
Figure 4.17 indicates that substantial convergence for all six alignments is achieved after about 110 generations. Therefore, the maximum number of generations for the lower level GA procedure is set at 200.

The complete bi-level program is then tested with values for the parameters summarized in Table 4.3 and 4.4.

<table>
<thead>
<tr>
<th>Table 4.3 Parameters of the Upper Level for Test Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
</tr>
<tr>
<td>Minimal allowable radius</td>
</tr>
<tr>
<td>Population size</td>
</tr>
<tr>
<td>Maximum number of intersection points</td>
</tr>
<tr>
<td>Mutation probability</td>
</tr>
<tr>
<td>Uniform mutation proportion</td>
</tr>
<tr>
<td>Non-uniform mutation proportion</td>
</tr>
<tr>
<td>One-point crossover probability</td>
</tr>
<tr>
<td>Maximum number of generations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4.4 Parameters of the Lower Level for Test Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
</tr>
<tr>
<td>Maximum allowable gradient</td>
</tr>
<tr>
<td>Sight distance</td>
</tr>
<tr>
<td>Population size</td>
</tr>
<tr>
<td>Number of vertical intersection points</td>
</tr>
<tr>
<td>Mutation probability</td>
</tr>
<tr>
<td>Uniform mutation proportion</td>
</tr>
<tr>
<td>Non-uniform mutation proportion</td>
</tr>
<tr>
<td>Crossover probability</td>
</tr>
<tr>
<td>One-point crossover proportion</td>
</tr>
<tr>
<td>Two-point crossover proportion</td>
</tr>
<tr>
<td>Arithmetical crossover proportion</td>
</tr>
<tr>
<td>Direction-based crossover proportion</td>
</tr>
<tr>
<td>Maximum number of generations</td>
</tr>
</tbody>
</table>

where $N_1 = \lceil \left( L_{\text{total}} + d/2 \right)/d \rceil + 1$ is the maximum number of vertical intersection points

$d = \text{the distance between two consecutive vertical intersection points}$

($d = 50m$ in this example)

$\lceil \cdot \rceil$ denotes the truncated integer value of its argument
Figure 4.18 The Best Alignment in the First Generation of the Upper Level program
Figure 4.19 The Best Alignment in the 50th Generation
Figure 4.20 The Best Alignment in the 100th Generation
To visualize the evolution of the program, the best horizontal alignment and its associated vertical alignment found in the 1\textsuperscript{st}, 50\textsuperscript{th}, and 100\textsuperscript{th} generations of the upper level program are shown in Figures 4.18-4.20.

Table 4.5 Cost Components for the best Alignment (S$)

<table>
<thead>
<tr>
<th>generation number</th>
<th>Cost items</th>
<th>land use cost</th>
<th>pavement cost</th>
<th>earthwork cost</th>
<th>vertical penalty</th>
<th>horizontal penalty</th>
<th>total cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.59×10^5</td>
<td>1.56×10^5</td>
<td>4.65×10^5</td>
<td>0</td>
<td>0</td>
<td>7.80×10^5</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>0.843×10^5</td>
<td>1.816×10^5</td>
<td>0.381×10^5</td>
<td>0</td>
<td>0</td>
<td>3.04×10^5</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0.840×10^5</td>
<td>1.810×10^5</td>
<td>0.380×10^5</td>
<td>0</td>
<td>0</td>
<td>3.03×10^5</td>
<td></td>
</tr>
</tbody>
</table>

The alignment in the first generation is not good since it passes through four very expensive cells and the last peak on the right side of the test domain. By the 50\textsuperscript{th} generation, the horizontal alignment is almost in its final position. Table 4.5 shows the cost components for the best alignment obtained in the 1\textsuperscript{st} 50\textsuperscript{th} and 100\textsuperscript{th} generations. The final alignment obtained in the 100\textsuperscript{th} generation seems to be a very reasonable solution for the test domain. It can be found in Figure 4.20 that the final optimal horizontal alignment skirts the two small hills to minimize earthwork cost and avoids high cost cells to minimize land use cost. The optimal vertical alignment in Figure 4.20 also shows that the road is very close to the ground profile in order to minimize earthwork excavation and embankment.

The objective function values in each generation are plotted in Figure 4.21. It shows that in the initial stage of the search, the objective value is extremely high. After about ten generations, the objective value drops sharply from about 8.0×10^5 to 3.6×10^5. After about 40 generations, the objective value is very close to the optimal solution found at the 100\textsuperscript{th} generation.

The example shows that the proposed bi-level GA program quickly identifies combinations of horizontal and vertical alignments to give high quality 3D alignments based on the total cost. Several noteworthy points about the final alignment obtained are (a) the alignment is continuous both in the horizontal and vertical planes; (b) the number of horizontal and vertical intersection points that define the alignment need not be the same;
and (c) the number of intersection points is determined by the bi-level GA depending on the terrain condition.

![Figure 4.21 Total Objective Value through successive Generations](image)

**4.3.3 Comparison of Jong’s Model and the Proposed Model for Vertical Alignment Optimization**

As discussed in the previous section, most of the existing approaches for the 3D road alignment optimization tend to optimize the horizontal and vertical alignment simultaneously. Therefore, they must have some assumption about the relationship between the horizontal alignment and its associated vertical alignment. For example, Jong [1998] assumed that the number and position of the vertical alignment are the same as the horizontal alignment. However, in real engineering project, the number and position of the vertical intersection points should vary depending on the terrain condition after the horizontal alignment is determined. In this sub-section, we intend to compare the resulting optimal vertical alignment obtained by both Jong’s and the proposed approaches for a particular horizontal alignment in the 3D road alignment optimization model.
The five candidate horizontal alignments are obtained by the following ways. Four of the five horizontal alignments are randomly generated within the study region while the remaining one is the optimal horizontal alignment obtained by the proposed bi-level genetic algorithm program in section 4.3.4. Both Jong’s [1998] model and the proposed model are used to optimize the vertical alignment for these five particular horizontal alignments. The parameters of these two programs are summarized as follows:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Proposed Model</th>
<th>Jong’s Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum allowable gradient</td>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td>Sight distance</td>
<td>122m</td>
<td>122m</td>
</tr>
<tr>
<td>Population size</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Number of vertical intersection points</td>
<td>$N_1$</td>
<td>$N_2$</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>Maximum number of generations</td>
<td>200</td>
<td>200</td>
</tr>
</tbody>
</table>

Note: $N_1 = \left\lfloor \frac{L_{\text{total}} + d/2}{d} \right\rfloor + 1$ the maximum number of vertical intersection points

$d$ = the distance between two consecutive vertical intersection points in the proposed model. ($d = 50m$ in this example)

$\lfloor \cdot \rfloor$ denotes the truncated integer value of its argument

$N_2$ = the number of vertical intersection points in Jong’s model, which is same as the number of horizontal intersection points

The five horizontal alignments together with the optimal vertical alignment obtained by both Jong’s model and the proposed model are shown as Figure 4.22 to Figure 4.26. The comparison of the results obtained by both the two models is shown as Figure 4.27.
Case Study 1

Figure 4.22 Case Study 1

Case Study 2

Figure 4.23 Case Study 2
Case 3

Figure 4.24 Case Study 3

Case 4

Figure 4.25 Case Study 4
It is obvious from Figure 4.22 to 4.27 that the final vertical alignments obtained by the proposed model are much closer to the ground profile than the one obtained by Jong’s model and thus reduce the earthwork volume cost for the particular horizontal alignment. This is mainly due to the fact that the number and position of the vertical intersection points in the proposed model can vary depending on the terrain condition whilst in Jong’s model, the number of position of the vertical intersection points are fixed. From the comparison from these two models, we can find that the proposed model has the advantages for 3D road alignment optimization compared with Jong’s model.
Figure 4.27 Comparison of Results (Earthwork Cost S$)
CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary and Conclusion

Determining the best road alignment in 3D space is a difficult road engineering problem for computers to solve without human guidance. Computer methods are necessary to automate the search through many feasible solutions to determine one that incurs the minimal total costs. The search space increases exponentially from 2D to 3D space; this has motivated the decomposition of the 3D road alignment problem into two separate horizontal and vertical alignment sub-problems.

Genetic algorithms (GA) are an optimization method based on evolutionary principles. In the first part of the research, the GA has been used as the basis to develop methods to optimize the horizontal and vertical alignments separately. In the horizontal alignment problem, the objective is to determine the best road alignment in 2D horizontal space. For each horizontal road alignment, it is necessary to determine the best vertical alignment among the many possible vertical alignments. The 3D alignment is obtained by combining the horizontal and vertical alignments. The case studies show that the proposed approach can very quickly and consistently improve the quality of the solutions for both the horizontal and vertical alignment problems using an iterative procedure.

Due to the non-linear interaction between horizontal and vertical alignments, and elements of the total cost, the best 3D alignment cannot be obtained by combining the best horizontal alignment and its associated best vertical alignment. Therefore, a bi-level GA approach is developed in this thesis to optimize the 3D alignment. The examples included in the study show that the proposed bi-level GA programming quickly identifies combinations of horizontal and vertical alignments to give high quality 3D alignments based on the total cost. Several noteworthy points about the final alignment obtained are (a)
the alignment is continuous both in the horizontal and vertical planes; (b) the number of horizontal and vertical intersection points that define the alignment need not be the same; and (c) the number of intersection points is determined by the bi-level GA depending on the terrain condition.

5.2 Recommendations for Future Research

Although the proposed models perform well in optimizing road alignments, there is considerable room for further improvements and further research.

5.2.1 Improvements in Cost Estimation

The cost function formulated in the proposed models only includes the most dominating and sensitive cost components. However, there are still some costs may be considered in future research. The possible improvements in cost estimation are summarized below:

i) User cost

The computation of user costs is less straightforward and various models have been developed to estimate various user costs including vehicle operating, travel time, and accident costs. These models were all derived from historical data using statistical regression. Different historical data in different country or region will lead to different results. User costs are significantly influenced by estimates of future traffic volumes. Due to the difficulties mentioned above, there is still no robust model to estimate the user cost. It should be added to the total costs if a robust model for accurately estimating user cost is available.

ii) Structure Cost

In some situations, a road may be constructed at less cost with tunnels or bridges instead of heavy earth cutting and filling. This option is not considered in the proposed models.
models. A way to incorporate tunnels and bridges into the models is to add some logic to the program.

5.2.2 Extensions of Model Capabilities

The proposed models only optimize the location of the centreline of a newly built road alignment. Possible extensions of the models are identified as follows:

i) Adding more design variables into the models

In designing a road, the decisions include not only the location of the alignment, but also other controls such as road width, radius, and super-elevations. It is possible to add design parameters other than location of the road centreline to achieve a better design.

ii) Considering more design constraints

The design constraints considered in the proposed models are horizontal curvature, maximal allowable gradient, and minimal length of vertical curves. Other constraints such as horizontal sight distance, critical length of vertical grade, and fixed levels controls may be included in the future research.
APPENDIX A CALCULATION OF FITNESS FUNCTION FOR HORIZONTAL ALIGNMENT

Land Use Cost

<table>
<thead>
<tr>
<th>Y = Y_{Y_{max}}</th>
<th>Y = Y_{Y+v<em>Y</em>D}</th>
<th>C(u, v)</th>
<th>Y = Y_{Y היום+Y*D}</th>
<th>Y = Y_{Y+Y*D}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>C(0, 1)</td>
<td>C(0, 0)</td>
<td>C(1, 0)</td>
</tr>
<tr>
<td>(X_{o}, Y_{o})</td>
<td>X = X_{o+D}</td>
<td>X = X_{o+Y*D}</td>
<td>X = X_{o+Y+h*D}</td>
<td>X = X_{max}</td>
</tr>
</tbody>
</table>

Figure A1 Cell Definition of the Study Region for Land Use

Based on the assumptions mentioned in section 3.1, we partition the study region with equal sized cells to store the land use cost. Let $C(u, v)$ denote the cell bounded by $x = x_o + u \times D$, $x = x_o + (u + 1) \times D$, $y = y_o + v \times D$, and $y = y_o + (v + 1) \times D$ (as shown in Figure A1). We further define $C_{Land}(u, v)$ as the unit land use cost for the cell $C(u, v)$.

Therefore the associated land use cost for a proposed alignment can be determined as:

$$C_{Land} = W \times \left[ \sum_{u=0}^{u_{max}-1} \sum_{v=0}^{v_{max}-1} L(u, v) \times C_{Land}(u, v) \right]$$  \hspace{1cm} (A1)

where: $u_{max} = (x_{max} - x_o) / D$ is the maximal cell index in X coordinate

$v_{max} = (y_{max} - y_o) / D$ is the maximal cell index in Y coordinate
\[ L(u,v) = \text{the length of the alignment in cell } C(u,v) \]

\[ W = \text{the width of the road, which is assumed to be fixed along the alignment} \]

\[ C_{\text{Land}}(u,v) = \text{the unit land use cost of cell } C(u,v) \]

\[ D = \text{the cell size} \]

In equation A1, \( L(u,v) \) is determined from the set of decision variables \( d_i \) defining the offsets along the cut-lines defined in section 3.5. There is no functional form relating \( L(u,v) \) explicitly to \( d_i \). \( L(u,v) \) can be computed only through a two-step procedure: (a) Step 1: Determine the individual elements of the horizontal alignment using the \( d_i \); (b) Step 2: Identify the cells that the alignment passes through and calculate the length of the alignment in these cells. This step involves further subdividing the individual alignment elements into shorter segments that lie wholly within a cell. Details of both steps are discussed in the subsequent sections.

**Step 1:** As discussed in the Chapter 3, the horizontal alignment contains tangent sections and circular curves. For notational convenience, we note that \( T_0 = C_0 = S \) and \( T_{n+1} = C_{n+1} = E \) at the start and end points of the horizontal alignment. Then, as illustrated in Figure A2, we observe that \( T_i \) and \( C_{i+1} \) are connected by a straight-line section (tangent section) while \( C_i \) and \( T_i \) are linked by a circular curve with radius \( R_i \) (circular curve section). The coordinates of points \( C_i \) and \( T_i \) can be obtained by trigonometric analysis:

\[
C_i = \begin{bmatrix} x_{C_i} \\ y_{C_i} \end{bmatrix} = \begin{bmatrix} x_{P_i} - L_{T_i} \times (x_{P_i} - x_{P_{i+1}})/L_{T_{i-1}} \\ y_{P_i} - L_{T_i} \times (y_{P_i} - y_{P_{i+1}})/L_{T_{i-1}} \end{bmatrix}
\]

(A2-a)
\[ T_i = \begin{bmatrix} x_{T_i} \\ y_{T_i} \end{bmatrix} = \begin{bmatrix} x_{P_i} + L_{T_i} \times (x_{P_{i-1}} - x_{P_i}) / L_i \\ y_{P_i} + L_{T_i} \times (y_{P_{i-1}} - y_{P_i}) / L_i \end{bmatrix} \]  

(A2-b)

where \( x_{P_i}, y_{P_i} \) = the coordinates of the \( i^{th} \) intersection point determined by \( d_i \) using equation 3.12

\( L_{T_i} \) = tangent length of \( i^{th} \) intersection point

\( L_i \) = the distance between two successive intersection points \( P_i \) and \( P_{i+1} \)

Figure A2 An Example of Points of Tangency and Curvature

Having determined the coordinates of \( C_i \) and \( T_i \), it is now possible to determine the length of alignment in each cell by subdividing the circular and tangent sections further using Step 2.

**Step 2:** The procedures to determine the coordinates of the subdivisions for circular and tangent sections are different and are discussed separately.

\( \text{a) Subdivision of tangent sections} \)

The purpose of the subdivision is to define shorter segments that lie wholly within a land-use cost cell. This is achieved by finding the coordinates of the entry and exit points of the tangent for a particular cell. Each subdivided tangent section will intersect 2 grid lines forming the boundaries of the cell - either horizontal grid lines (parallel to the \( X \) axis) and/or vertical grid lines (parallel to the \( Y \) axis). There is another possibility
where the tangent section lies wholly within a cell and there will not intersect any of the cell’s boundaries.

Let \( \text{link}(i) \) be the link connected by \( T_i \) and \( C_{i+1} \) for all \( i = 0, \ldots, n \) (where \( n \) is the number of the horizontal intersection points). The function of the tangent segment \( \text{link}(i) \) can be derived as:

\[
\frac{x - x_{T_i}}{y - y_{T_i}} = \frac{x - x_{C_{i+1}}}{y - y_{C_{i+1}}}
\]

(A3)

The coordinates of the intersection points can be obtained using equation A3. The ranges to be considered in solving the above equation are:

\[
[\min(x_{T_i}, x_{C_{i+1}}), \max(x_{T_i}, x_{C_{i+1}})] \text{ for the } X \text{ interval, and}
\]

\[
[\min(y_{T_i}, y_{C_{i+1}}), \max(y_{T_i}, y_{C_{i+1}})] \text{ for the } Y \text{ interval}
\]

Sorting the intersection points by their X or Y coordinates will order the points in the correct sequence. The midpoint of any two consecutive points will indicate the cell through which the line segment passes.

Let \( S_1^i, S_2^i, \ldots, S_j^i \) be the ordered set of intersection points after subdividing \( \text{link}(i) \), including the two end points of the \( \text{link}(i) \) as in Figure A3. The coordinates of \( S_j^i \) are represented by \((x_{S_j^i}, y_{S_j^i})\). The line segment between two consecutive points \( S_j^i \) and \( S_{j+1}^i \) will fall within cell\((u,v)\) determined by:

\[
\text{Index } u: \text{Index}_u = \left\lfloor \frac{(x_{S_j^i} + x_{S_{j+1}^i})/2 - x_o}{D} \right\rfloor \quad (A4-a)
\]

\[
\text{Index } v: \text{Index}_v = \left\lfloor \frac{(y_{S_j^i} + y_{S_{j+1}^i})/2 - y_o}{D} \right\rfloor \quad (A4-b)
\]
where \( \lfloor \cdot \rfloor \) denotes the truncated integer value of its argument.

\[
C_{\text{Land}}^T = W \times \left[ \sum_{i=0}^{n} \sum_{j=1}^{J} C_{\text{Land}}(\text{Index}_u, \text{Index}_v)_j \times L_i \right] \quad (A5)
\]

where: \( C_{\text{Land}}^T \) = the land use cost of the alignment along all tangent sections

\[
C_{\text{Land}}(\text{Index}_u, \text{Index}_v)_j = \text{the unit land use cost where } j^{\text{th}} \text{ segment of link}(i) \text{ is located}
\]

\[
L_j = \left( (x_{S_j}^i - x_{S_{j+1}}^i)^2 + (y_{S_j}^i - y_{S_{j+1}}^i)^2 \right)^{1/2} \text{ is the distance between } S_j^i \text{ and } S_{j+1}^i
\]

\( J \) = number of intersection points of link\((i)\)

\( b) \) Land use cost of circular curves
The computation of land use cost for the circular curves of a given alignment is relatively difficult compared with tangent sections. Let $\text{Arc}(i)$ be the circular curve from $C_i$ to $T_i$ for all $i = 1, \ldots, n$ (where $n$ is the number of the horizontal intersection points). Three parameters are required for completely describing $\text{Arc}(i)$. They are the point of curvature $C_i(x_{C_i}, y_{C_i})$, the point of tangency $T_i(x_{T_i}, y_{T_i})$, and the center of the circular curve $\Phi_i(x_{\Phi_i}, y_{\Phi_i})$. The coordinates of $C_i$ and $T_i$ can be obtained by equation A2. As to $\Phi_i$, we can obtain its coordinates by trigonometric analysis:

If $\alpha_{i+1} - \alpha_i \leq 0$, then

$$\Phi_i = \begin{bmatrix} x_{\Phi_i} \\ y_{\Phi_i} \end{bmatrix} = \begin{bmatrix} x_{C_i} \\ y_{C_i} \end{bmatrix} + \begin{bmatrix} R_i \times \cos(\Delta_i - \pi/2) \\ R_i \times \sin(\Delta_i - \pi/2) \end{bmatrix}$$

(A6-a)

else, $\Phi_i = \begin{bmatrix} x_{\Phi_i} \\ y_{\Phi_i} \end{bmatrix} = \begin{bmatrix} x_{C_i} \\ y_{C_i} \end{bmatrix} + \begin{bmatrix} R_i \times \cos(\Delta_i + \pi/2) \\ R_i \times \sin(\Delta_i + \pi/2) \end{bmatrix}$

(A6-b)

where: $\alpha_i$ = the direction of vector $P_{i-1}P_i$ which is obtained by connecting two consecutive intersection points $P_{i-1}$ and $P_i$ (see Appendix B for the calculation of the direction of vectors)

The formulation of the circular curve can then be derived as:

$$(x - x_{\Phi_i})^2 + (y - y_{\Phi_i})^2 = R_i^2$$

(A7)

The circle will intersect each grid line at two distinct points due to the symmetric property of the circle unless it is just tangent to the grids. The coordinates of these intersection points can be obtained by equation A7. The ranges to be considered in solving the above equation are:

$$[\min(x_{C_i}, x_{T_i}), \max(x_{C_i}, x_{T_i})]$$ for the $X$ interval, and

$$[\min(y_{C_i}, y_{T_i}), \max(y_{C_i}, y_{T_i})]$$ for the $Y$ interval
For each $X/Y$ within the above interval, two distinct points can be obtained by equation A7. However, what we need to know is the intersection points which belong to $Arc(i)$. Figure A4 shows an example of this instance. Suppose that the $X$ coordinate is $x_i$, then two distinct $Y$ coordinates $y_i$ and $y_i'$ can be obtained by the equation A7. In other word, there two intersection points $O_i$ and $O_i'$ on the circle which have the same $X$ coordinate $x_i$. A criterion is used here to judge whether the intersection point belongs to $Arc(i)$. First, let $\alpha_{\Phi,C_i}$ and $\alpha_{\Phi,T_i}$ be the direction of vectors $\Phi_iC_i$ and $\Phi_iT_i$. We also need to calculate the direction of vector $\Phi_iO_i/\Phi_iO_i'$, say $\beta$ (see Appendix B for the calculation of the direction of vector). The intersection point belongs to $Arc(i)$ if and only if $\beta$ is within the range $[\alpha_{\Phi,C_i}, \alpha_{\Phi,T_i}]$. 

Figure A4 Intersection Points of Grids and Circle
Sorting the intersection points by their X or Y coordinates will order the points in the correct sequence. Let \( O_{1}^{j}, O_{2}^{j}, \ldots, O_{k}^{j} \) be the set of intersection points after sorting, including two end points of \( Arc(i) \), where \( O_{j}^{j} \) denotes the \( j^{th} \) intersection points of \( Arc(i) \) as in Figure A5.

![Figure A5 Sorted Intersection points of A Circular Curve](image)

Suppose that the middle point of the arc segment, denoted by \( M_{j}^{i} \), is used to indicate the cell. Then the coordinates of \( M_{j}^{i} \) can be obtained by trigonometric analysis as equation A8. The geometric representation of the analysis is illustrated in Figure A6.

\[
M_{j}^{i} = \begin{bmatrix} x_{M_{j}^{i}} \\ y_{M_{j}^{i}} \end{bmatrix} = \begin{bmatrix} x_{j}^{i} \\ y_{j}^{i} \end{bmatrix} + \begin{bmatrix} \frac{(x_{m_{j}}^{i} - x_{j}^{i}) \times R_{j} / \sqrt{(x_{m_{j}}^{i} - x_{j}^{i})^2 + (y_{m_{j}}^{i} - y_{j}^{i})^2}} \\ \frac{(y_{m_{j}}^{i} - y_{j}^{i}) \times R_{j} / \sqrt{(x_{m_{j}}^{i} - x_{j}^{i})^2 + (y_{m_{j}}^{i} - y_{j}^{i})^2}} \end{bmatrix}
\]  

(A8)

where: \( m_{j}^{i} = \begin{bmatrix} x_{m_{j}}^{i} \\ y_{m_{j}}^{i} \end{bmatrix} = \begin{bmatrix} (x_{j}^{i} + x_{j+1}^{i})/2 \\ (y_{j}^{i} + y_{j+1}^{i})/2 \end{bmatrix} \) is the middle point of the straight line connecting \( O_{j}^{j} \) and \( O_{j+1}^{j} \)
Figure A6 The Geometric Representation of Equation A7

With the coordinates of $M_j^i$, the indexes of the cell through which an arc segment connects $O_j^i$ and $O_{j+1}^i$ are as follows:

Index $u$: $Index_u = \left| \frac{x_{M_j^i} - x_o^i}{D} \right| \quad (A9-a)$

Index $v$: $Index_v = \left| \frac{y_{M_j^i} - y_o^i}{D} \right| \quad (A9-b)$

Then the land use cost of the alignment along all circular curves is:

$$C_{Land}^A = W \times \left[ \sum_{i=0}^{n-1} \sum_{j=1}^{K-1} C_{Land}(Index_u, Index_v)^j \times A_j^i \right] \quad (A10)$$

where: $C_{Land}^A$ = the land use cost of the alignment along all circular curves

$$C_{Land}(Index_u, Index_v)^j_i = \text{the unit land use cost where } j^{th} \text{ segment of } arc(i)$$

is located
\[ A_i^j = 2 \times R_i \times \sin^{-1}\left(\frac{\left(\left(x_{O_i} - x_{O_{i+1}}\right)^2 + \left(y_{O_i} - y_{O_{i+1}}\right)^2\right)^{1/2}}{R_i}\right) \] is the length of the \( j \)th segment of \( arc(i) \)

\[ K = \text{number of intersection points of } arc(i) \]

**Pavement Cost**

The computation of pavement cost for a road is relatively straightforward. The pavement cost is the product of the total road length and the unit pavement cost. The total length of the proposed road alignment, denoted by \( L_{\text{total}} \), is expressed as:

\[
L_{\text{total}} = \sum_{i=0}^{n} \sqrt{(x_{R_i} - x_{C_{i+1}})^2 + (y_{R_i} - y_{C_{i+1}})^2} + \sum_{i=1}^{n} R_i \Delta_i
\]  

(A11)

Then the pavement cost can be obtained as:

\[
P_{\text{total}} = L_{\text{total}} \times U_p
\]  

(A12)

where: \( C_p = \) pavement cost of the proposed road

\( U_p = \) unit cost of pavement

The fitness function can then be calculated by equation 4.5. For any horizontal constraint violation, a user specified penalty is added to the fitness function in order to prevent this situation. The horizontal constraint considered in this research is the minimal allowable radius.
APPENDIX B  CALCULATION FOR DIRECTION OF VECTORS

Vector is a quantity that has two aspects. It has a size, or magnitude, and a direction. Vectors are usually drawn as arrows. The direction of vectors in this research refers to the angle measured counterclockwise between the $X$ axes and the vectors. Figure A7 shows the geometric representation of the direction for vectors.

Figure A7 Geometric Representation of the Direction for Vectors

Suppose that we know the coordinates of both the two end points of vector $\vec{SE}$.

Then the direction of vector $\vec{SE}$ can be obtained as follows:

Case 1: $y_E \geq y_S$

$$\alpha_{\vec{SE}} = \cos^{-1}\left(\frac{x_E - x_S}{\sqrt{(x_E - x_S)^2 + (y_E - y_S)^2}}\right)$$  \hspace{1cm} (A13-a)

a)

Case 2: $y_E < y_S$

$$\alpha_{\vec{SE}} = 2\pi - \cos^{-1}\left(\frac{x_E - x_S}{\sqrt{(x_E - x_S)^2 + (y_E - y_S)^2}}\right)$$  \hspace{1cm} (A13-b)

where: $\alpha_{\vec{SE}} =$ the direction of vector $\vec{SE}$

$x_S, y_S =$ the coordinates of the start point $S$

$x_E, y_E =$ the coordinates of the end point $E$
APPENDIX C CALCULATION OF GROUND ELEVATION ALONG THE HORIZONTAL ALIGNMENT

The ground profile is determined by finding the height of the terrain at intermediate ground points, located between the start and end points of the horizontal alignment, which are spaced at equal distances apart. Cross sections at these selected intermediate locations are used to calculate the earthwork volume. Based on the assumption mentioned in section 3.1, we partition the study region with equal sized cells \((D_e \times D_e)\) to store different ground elevation data. Let \(C(u, v)\) denote the cell bounded by \(x = x_o + u \times D_e\), \(x = x_o + (u+1) \times D_e\), \(y = y_o + v \times D_e\), and \(y = y_o + (v+1) \times D_e\) (as shown in Figure A8).

Suppose that we decide to obtain the elevation of the intermediate ground points along the horizontal alignment with equal interval \(d_e\), then the total number of ground points can be obtained as:

![Figure A8 Cell Definition of the Study Region for Ground Elevation](image-url)
\[ N = \left\lfloor \left( L_{\text{total}} + \frac{d_E}{2} \right) / d_E \right\rfloor + 1 \]  

(A14)

where:  

- \( N \) = the total number of the intermediate ground points
- \( L_{\text{total}} \) = the length of the proposed alignment (obtained by equation A11)
- \( d_E \) = horizontal interval between two successive intermediate ground points

\( \lfloor \cdot \rfloor \) denotes the truncated integer value of its argument.

The coordinates of the ground point along the horizontal alignment will indicate the cell through where the ground point exists. The ground points locate at either tangent segments or circular curves of the horizontal alignment. Different equation will be used to compute the coordinates of the ground points for different point locations. The station of the ground points \( S_{gp} \), tangent point \( S_T \), and curvature point \( S_C \) is needed first in order to calculate the coordinates of any ground point. The geometric representation of the station of tangent point and curvature point is shown as Figure A9.

![Figure A9 Geometric Representation of \( S_C \) and \( S_T \)](image)

The station of the ground points to be calculated along the horizontal alignment can be expressed as:
The station of the points \( T_j \) and \( C_i \) can be expressed as:

\[
\begin{align*}
S_{gp_0} &= 0 \\
S_{gp_{N-1}} &= L_{total} \\
S_{gp_j} &= d \times j, \ j = 1, \ldots, N - 2
\end{align*}
\]  

(A15)

The location of the ground points can be determined when the station of the ground points, tangent point \( T_j \), and curvature point \( C_i \) are known. The coordinates of the ground points with different locations are different and will be discussed separately.

- Coordinates of ground points at tangent segment

Suppose that the ground point \( gp_j \) is located between \( T_j \) and \( C_{i+1} \), then the coordinate of this ground point is:

\[
gp_j = \begin{bmatrix} x_{gp_j} \\ y_{gp_j} \end{bmatrix} = \begin{bmatrix} x_{T_j} + (x_{C_{i+1}} - x_{T_j}) \times \frac{S_{gp_j} - S_{T_j}}{S_{C_{i+1}} - S_{T_j}} \\ y_{T_j} + (y_{C_{i+1}} - y_{T_j}) \times \frac{S_{gp_j} - S_{T_j}}{S_{C_{i+1}} - S_{T_j}} \end{bmatrix}
\]  

(A17)

- Coordinates of ground points at circular curve

Suppose that the ground point \( gp_j \) is located between \( C_i \) and \( T_j \), then the coordinate of this ground point is:

\[
gp_j = \begin{bmatrix} x_{gp_j} \\ y_{gp_j} \end{bmatrix} = \begin{bmatrix} x_{C_i} + R_i \times \cos(\alpha + \frac{\alpha_{C_i}}{C_i} + \alpha) \\ y_{C_i} + R_i \times \sin(\alpha + \frac{\alpha_{C_i}}{C_i} + \alpha) \end{bmatrix}
\]  

(A18)
where: $\alpha_{\Phi_i C_i}$ = the direction of vector $\Phi_i C_i$ (see Appendix B for the calculation of the direction of vector)

$$\alpha = (S_{gp_j} - S_{c_i}) / R_i$$ is the angle between vectors $\Phi_i C_i$ and $\Phi_i gp_j$

The geometric representation of equation A18 is as follows:

![Geometric Representation of equation A18](image)

Figure A10 Geometric Representation of equation A18

Let $gp_1$, $gp_2$, …, $gp_N$ be the ground points along the horizontal alignment. The coordinates of $gp_j$ can be computed using equation A17 and A18. The indexes of the cell at which the ground point $gp_j$ located are as follows:

Index $u$: $index_u = \lfloor x_{gp_j} / d_x \rfloor$ \hspace{1cm} (A19-a)

Index $v$: $index_v = \lfloor y_{gp_j} / d_y \rfloor$ \hspace{1cm} (A19-b)

where $\lfloor \cdot \rfloor$ denotes the truncated integer value of its argument.
The indexes calculated with equation A19 indicate the cell where \( g_{pij} \) is located.

We further define \( C_{Ele}(u,v) \) as the ground elevation data for the cell \( C(u,v) \). Then the ground elevation along the horizontal alignment can be calculated by:

\[
gp_{Ej} = C_{Ele}(index_u, index_v)
\]

where: \( g_{pEj} \) = the ground elevation along the horizontal alignment, for

\[ j = 0,1, \ldots, N - 1 \]
APPENDIX D DETERMINATION OF THE ROAD DESIGN ELEVATION

As discussed in the Chapter 3, the vertical alignment contains tangent sections and parabolic curves. The logical and mathematical requirements for determining the road design elevation of alignment are different for tangent sections and parabolic curves. They will be discussed separately as follows:

Station point located on a parabolic curve (shown in Figure A11)

\[ d_{E_j} = E_{VPC_{-i}} + \frac{g_i}{100} + \frac{1}{2} \frac{g_{i+1} - g_i}{100 \times L_i} x^2 \]  \hspace{1cm} \text{(A20)}

where: \[ d_{E_j} \] = the design elevation of the selected ground point \[ gp_j \]

\[ E_{VPC_{-i}} \] = the elevation of the point \[ VPC \] of \( i^{th} \) intersection point

Figure A11 Station point on a parabolic curve
\[ L_i = \text{the length of the } i^{th} \text{ vertical curve} \]
\[ x = \text{the distance between } VPC \text{ and the selected ground point } gp_j \]

- Station point located on a tangent section (shown in Figure A12)

\[ \begin{align*}
  d_{E_j} &= E_{VPT-i} + \frac{g_{i+1}}{100} x \\
  \text{where: } d_{E_j} &= \text{the design elevation of the ground point } gp_j \\
  E_{VPT-i} &= \text{the elevation of the point } VPT \text{ of } i^{th} \text{ intersection point} \\
  x &= \text{the distance between } VPT \text{ and the ground point } gp_j
\end{align*} \]
REFERENCES


Holland, J. H. Adaptation in Natural and Artificial System, University of Michigan Press, Ann Arbor, MI. 1975


Jong, J. C. Optimizing Highway Alignments with Genetic Algorithms, Ph.D. Dissertation, Department of Civil Engineering, University of Maryland, College Park, Maryland. 1998


Murchland, J. D. Methods of Vertical Profile Optimisation for an Improvement to an Existing Road, Planning and Transport Research and Computation Seminar Proceedings on Cost Models and Optimisation in Highways (Session L12), London. 1973


