Horizontal Alignment Optimization in Road Design

by

Sukanto Mondal

B.Sc., Rajshahi University of Engineering & Technology, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE COLLEGE OF GRADUATE STUDIES

(Interdisciplinary Studies - Optimization)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

July 2014

Abstract

The horizontal alignment optimization problem in road design is a complex problem. Usually, classic optimization techniques cannot be used to address the problem. A few studies investigated the problem mainly using heuristics. Unfortunately, all of the previously studied heuristic based methods do not guarantee optimality. In this study, we develop a novel optimization model to solve the horizontal alignment optimization problem in a specified corridor. The cost of a horizontal alignment is significantly affected by the associated vertical alignment cost. So in order to formulate the cost function of the model, we consider both the vertical alignment and earthwork allocation associated with a horizontal alignment. The representation of a horizontal alignment in our model satisfies all of the geometrical specifications used by engineers. Our model is suitable for both backtracking and non-backtracking horizontal alignments. Derivative-free optimization algorithms are used to solve the problem and guarantee the local optimality of our solution. The numerical experiment results of a set of practical problems are reported.

Table of Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, I would like to thank my supervisors, Dr. Yves Lucet and Dr. Solomon Tesfamariam, for their continuous help and encouragement. This thesis would not have become a reality without their guidance and advice. More importantly, thank you to my supervisors for allowing me to knock on their door at any time.

I would like to thank Dr. Warren Hare, who actively advised me during my research. His insightful comments on my research work helped me to develop better ideas.

A big thank you goes out to our industrial partner, Softree Technical Systems Inc., for providing engineering details and practical datasets. It must be mentioned, in the weekly meetings, we had many productive discussions with David Mills, Craig Speirs, and Alexis Guigue on different engineering aspects of the problem. More specifically, during the development of the optimization model, their valuable comments helped me a lot to build a better model.

I would also like to thank my colleagues from different parts of the world, affiliated with the Center for Optimization and Convex Analysis and Nonsmooth Analysis, for our off topics discussions to know their diverse native cultures and lifestyles. My dear colleagues, I really enjoyed your companionship.

This research is partially funded by a Collaborative Research and Development (CRD) Grants from the Natural Sciences and Engineering Research Council (NSERC) sponsored by Softree Technical Systems Inc. The research was performed in the Computer-Aided Convex Analysis (CA^2) laboratory funded by a Leaders Opportunity Fund (LOF) from the Canadian Foundation for Innovation (CFI) and by a British Columbia Knowledge Development Fund (BCKDF).

Dedication

Dedicated to underprivileged talents of the world.

Chapter 1

Introduction

In this chapter, we briefly describe the background of the road design optimization research and our motivation to solve the horizontal alignment optimization problem.

1.1 Motivation

Since the early days of human civilization, the transportation system is considered an integral part of sustainable socioeconomic development. The gradual development of human civilization has led us to invent different modes of transportation, such as land transportation, sea transportation, and air transportation. The invention of the wheel revolutionized the land transportation system and accelerated the economic development manyfold. As of today, the transportation system is continuously contributing to our economy significantly. For example, the transportation sector of Canada contributed about 4.2% of Canada's gross domestic product (GDP) in 2005 [DAAMP06]. In particular, more than one third (about 35%) of the GDP generated by the transportation sector in 2005 came from the truck transportation industry [DAAMP06]. The truck transportation system uses a total of 1,042,300 km of roads in Canada.

The alignment of a road is the route connecting two given end-points. An alignment consists of the vertical and horizontal alignments. Intuitively, a good alignment is one which minimizes the construction costs satisfying the design constraints. In the traditional road design process, engineers use their professional judgment to determine several selected candidate alignments and then manually try to find the best one. In fact, a large number of alternative alignments exist that should be considered in the design process. In the conventional design process, finding the best alignment requires repetitive manual iterations. So it is almost impossible for engineers to consider all of the possible alternative alignments. Hence, engineers cannot ensure that the chosen alignment is (even locally) optimal.

In order to overcome the difficulties in the road design process, it is imperative to develop a computer-aided process to find the optimal alignment.

In the literature, many studies proposed computerized models to find the optimal horizontal alignment. Unfortunately, all of the models have some drawbacks that prevent them to be used effectively in practice (for further discussion see Section 1.3.2). In our research, we solve the horizontal alignment optimization problem and provide a practical model that guarantees (local) optimality. Our approach is validated by computing several real-world alignments.

1.2 Road design optimization

Road design optimization is the problem of finding a curve connecting two given end-points that minimizes the cost while satisfying all of the desired design specifications. The problem is usually divided into three interrelated sub-problems, namely, horizontal alignment optimization, vertical alignment optimization, and earthwork optimization.

In the three dimensional space, a curve can be projected on the horizontal plane (i.e., XY-plane). In road design literature, the curve is called the alignment, and the horizontal projection is called the horizontal alignment. In Figure 1.1, for a three dimensional alignment, the associated horizontal alignment is depicted.

Along the horizontal alignment shown in Figure 1.1, we can measure the distance from the starting point to the end point. Corresponding to any measured distance (using the x-values and the y-values of a three dimensional alignment), we have a z-value (ground elevation). If we draw the z-values with respect to the distance from the starting point in two dimensional space, then the resulting curve is called the vertical alignment.

Finding the optimal alignment connecting the two end-points is hard because of the complex cost structure associated with an alignment and the requirement for satisfying design constraints [JSJ06]. Moreover, the continuous search space of the problem gives an infinite number of alternative alignments. Since the terrain might have an irregular surface, a small change in an alignment may result in a significant change in the total cost.

In order to find a good alignment, an engineer considers five major costs [JSJ06] as follows

– planning and administrative cost,

– construction cost,

– maintenance cost,

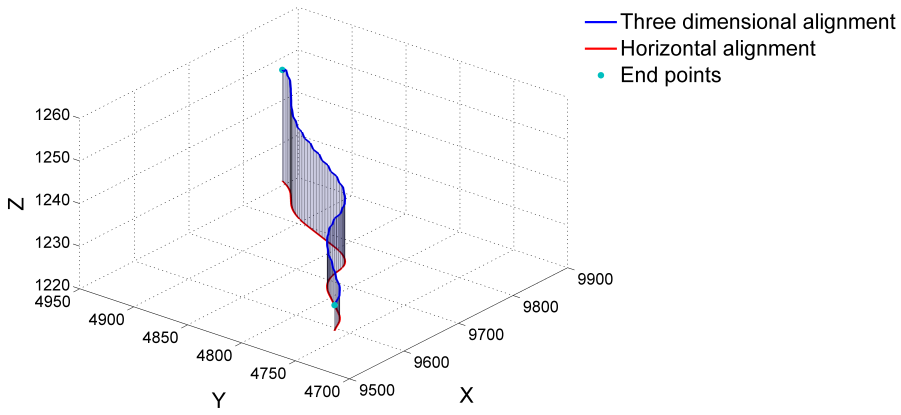

Figure 1.1: A three dimensional alignment (blue curve) showing its projection onto the XY-plane. The projected red curve is the horizontal alignment.

– user cost, and

– social and environment cost.

Planning and administrative costs are not considered in the alignment optimization because these costs are insensitive to alignment alternatives [JSJ06]. In the study of Chew et al. [CGF89], construction costs are classified into six categories. Table 1.1 lists the cost components of the construction costs and the associate approximate contributions toward the total construction cost.

The percentage of each cost component in the total construction cost is not fixed [JSJ06]. Depending on the road location, it could be significantly different. For example, the land acquisition cost might be higher in urban areas, whereas in mountainous areas, the earthwork cost is substantially higher than other costs.

Maintenance costs have many classifications (at least eight) such as roadway surface, bridges, tunnels, roadside features, drainage, shoulders and approaches maintenance, snow and ice control, and traffic control devices. The net maintenance cost over 30 years is about 5% of the total construction cost [OEC73].

User costs (vehicle operating costs) consist of the cost of vehicle maintenance, the value of travel time and the cost of traffic accidents. The net user cost over 30 years varies approximately from 300% to 1000% of the total construction cost [OEC73].

Table 1.1: Classification of the construction costs.

Cost components	Contributions (%)
Land	5%
Miscellaneous items	10 %
Drainage	10 %
Bridges	20 %
Earthwork	25%
Pavement	30%

Social and environment costs are the negative impacts of the road on the environmental and social features of a particular region. In some extreme cases, environmental and social issues might be very critical and even dominates other cost. In practice, social and environment costs are very hard to quantify. Typically, social and environment costs are carefully considered in the planning stage when the preliminary corridor is selected.

However, among the five major cost components of the total cost, in a selected corridor, constructions costs form an important component of the total cost function. In particular, construction costs (excluding land acquisition costs), mainly consist of excavation costs, embankment costs, and hauling costs for the construction materials. In our research, we minimize the total excavation costs, embankment costs, and hauling costs satisfying all design constraints for the road. Note that the design constraints contribute to the user costs and the maintenance costs [BDE10]. For instance, a long and gentle vertical alignment provides a great sight distance which minimizes road accidents.

In a traditional engineering approach, finding a good alignment is a repetitive and complex process. It involves a series of phases, starting from feasibility studies followed by planning, then narrowing down to the selection of several possible corridors, and finally focusing on the details of an alignment including earthwork minimization, and horizontal and vertical design constraints.

1.2.1 Earthwork optimization

Earthwork optimization is the problem of minimizing the total excavation cost, embankment cost, and hauling cost for a fixed alignment. In road construction, earthwork is the major task which involves excavation, embankment and hauling of large quantities of earth materials. The cut and

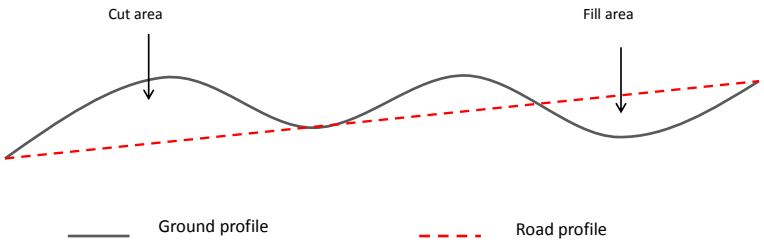

Figure 1.2: Example of cut and fill in road construction.

fill areas are determined by the intersection of the ground profile and the road profile (see Figure 1.2). If the ground profile is below the road profile, then more material is needed to fill. On the other hand, if the ground profile is above the road profile, then additional material has to be cut. The excavation and embankment costs are incurred in the cut and fill areas respectively. The hauling cost is the cost associated with moving the material from the cut areas to fill areas. The earthwork allocation cost is defined as the combination of the excavation, embankment, and hauling costs.

1.2.2 Vertical alignment optimization

A large number of vertical alignments can be built for a fixed horizontal alignment (see Figure 1.3). For each vertical alignment, the earthwork allocation cost (which is the minimum cost corresponding to that fixed alignment) can be calculated by solving the earthwork optimization problem. Thus the vertical alignment optimization problem is to find the vertical alignment which has the minimum earthwork allocation cost satisfying the vertical alignment design constraints for a fixed horizontal alignment. The major design criteria for a vertical alignment are the allowable grades and the rate of curvature [AAS04]. The maximum and minimum allowable grades depend on the design speed (i.e., maximum speed limit) and the traffic composition. The rate of curvature is the length of the vertical curve per percent algebraic difference between the grade at the two end-points of the curve [AAS04].

1.2.3 Horizontal alignment optimization

In a specified corridor, a large number of horizontal alignments can be built (see Figure 1.4). As we described earlier, in the vertical alignment

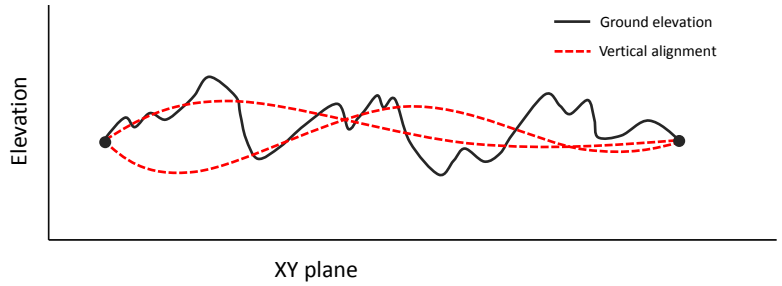

Figure 1.3: Two potential vertical alignments of a fixed horizontal alignment.

optimization process the horizontal alignment is fixed. Thus, for each feasible horizontal alignment, the vertical alignment optimization problem can be solved to get the optimal vertical alignment. The vertical alignment optimization yields the minimum earthwork allocation cost satisfying the vertical alignment design constraints. From the previous two assertions, immediately it follows that each horizontal alignment has a cost (in particular, the minimum earthwork allocation cost) which is computed by solving the corresponding vertical alignment optimization problem. Therefore, the horizontal alignment optimization problem can be defined as the problem of finding the horizontal alignment which has the minimum earthwork allocation cost satisfying the horizontal alignment design constraints and the associated vertical alignment design constraints.

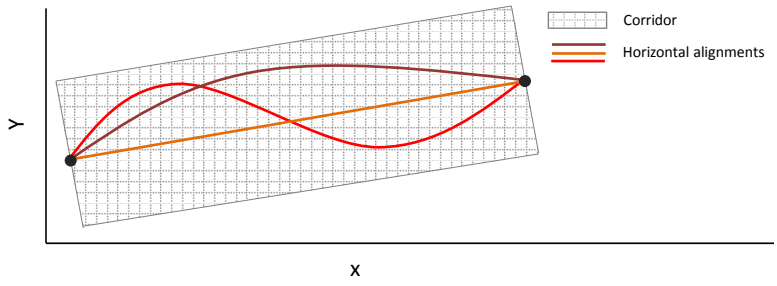

Figure 1.4: Some potential horizontal alignments in a specified corridor.

A horizontal alignment consists of some tangential line segments followed by some circular arcs. The most important design constraint of the horizontal alignment is the minimum radius of curvature of the circular arcs for

the safety requirement.

1.3 Background and literature review

Computer-aided road design optimization started during the 1960's and 1970's. Due to the limited computational power, it was hard to solve optimization problems precisely. The advent of the modern computer and GIS technology has helped us to solve the problem accurately. The road construction cost can be minimized substantially by solving the mathematical models using modern computer programs. Many optimization models have been developed to address the problem from different perspectives. Although existing models work well in certain aspects, they still have some drawbacks that prevent them to be used in practice.

1.3.1 Earthwork and vertical alignment optimization model

Many different approaches have been considered to model the vertical alignment optimization problem. Numerical search [Hay70, Pea73, Rob73, GCF88], and dynamic programming [Hua73, Mur73, Fwa89, GLA05] are notably used to solve the problem.

Numerical search models are the earliest techniques to deal with the vertical alignment optimization problem. Usually, a resulting numerical search model of the vertical optimization problem becomes a non-linear non-convex optimization problem which is hard to solve [JSJ06, page 28, section 2.5.4].

The dynamic programming approach yields a piece-wise linear alignment rather than a smooth alignment. Rahman [Rah12, page 9, section 1.2.3] noted that the dynamic programming approach is not suitable to solve the vertical alignment optimization problem.

In traditional engineering, a mass haul diagram is used to minimize the earthwork allocation. The mass haul diagram cannot be used in practical situations [MS81]. The mass diagram has mainly the following limitations, it cannot handle nonlinear hauling costs and the costs associated with borrow and waste pits. To overcome the previous limitations, a linear programming model was developed by Mayer and Stark [MS81]. Later, the earthwork allocation and the vertical road profile are modelled together by the Transportation and Road Research Laboratory in the United Kingdom and the model was further modified by the Ontario Ministry of Transportation and Communications. This model is the first attempt to integrate the earthwork and vertical alignment optimization together. Unfortunately, the model does not guarantee optimality.

Easa [Eas88] developed a linear programming model considering the earthwork allocation and the vertical alignment. His approach works in three steps: parameterizing all technically feasible vertical alignments, calculating the cut and fill requirements for each vertical alignment, and then using linear programming to optimize the earthwork allocation. The model gives the global optimum through an exhaustive enumeration of possible vertical profile.

Moreb [Mor96] developed the earthwork allocation and vertical road profile in a single linear programming model. Both Easa's model and Moreb's model output a piecewise linear vertical curve. But a smooth alignment is desired. In order to remove the sharp connectivity of the piecewise linear vertical alignment, the optimum result might be tempered. Engineers use quadratic spline for the vertical alignment. Moreb and Aljohani [MA04] modified the model developed in [Mor96] considering the vertical alignment as a quadratic spline. Moreb [Mor09] further improved the previous model by adding some additional constraints to ensure the smoothness with any degree of polynomial spline. However, Koch and Lucet [KL10] proved that the linearity of the model can only be maintained up to quadratic splines.

Recently, Hare, Koch, and Lucet [HKL11] developed a mixed integer linear programming model for the earthwork optimization considering blocks. Rahman [Rah12] extended several models: [Mor09], [KL10], [HKL11] (unpublished manuscript [HHLM11] is also mentioned in [Rah12] but I could not access it) to formulate the vertical alignment optimization problem as a mixed integer linear program model. Hare et al. [HHLR14] further improved the model to reduce the solution time. In our research, we use the model developed in [HHLR14] as the objective function to solve the horizontal alignment optimization problem.

1.3.2 Horizontal alignment optimization model

The horizontal alignment optimization problem is more complicated than the vertical alignment optimization problem [JSJ06]. The main reasons are that the horizontal alignment requires more data, and the cost of the horizontal alignment is dependent on the vertical alignment cost, political, socioeconomic, and environmental issues. In the literature, mainly three basic approaches have been studied: calculus of variation, network optimization, and dynamic programming.

Calculus of variation tries to find a curve connecting two end points in space which minimizes the integral of a function [Wan95]. The nature of the alignment optimization problem allows us to use the concept of calculus of

variation to find the optimal alignment. Howard et al. [HBS68] used the idea of calculus of variation to develop the Optimum Curvature Principle (OCP), which specifies the optimal vertical and horizontal curvatures at any point. In order to apply the OCP, Shaw and Howard [SH81] proposed two numerical integration methods, namely, the arc of circle algorithm and the intrinsic equation procedure. The OCP was applied by Shaw and Howard to find the optimal alignment of an expressway in South Florida [SH82]. Two major requirements to use the OCP are the followings: first, the cost function has to be continuous and second, the cost function has to be twice continuously differentiable. In practice, the cost function might not be continuous [JSJ06, page 8, section 2.4.1]. Although the OCP guarantees global optimality, it requires some assumptions that make it impractical.

Another well-known approach to model the horizontal alignment optimization problem is network optimization. In the network optimization approach, a network is designed to represent a region through which a road could pass. The region is divided into small cells to make a grid. Each cell in the grid represents a node of the network. The nodes are connected through the arcs. Each arc in the network is assigned a weight considering the cost associated with the two connecting cells. An alignment is defined as a set of connecting arcs from the starting node to the ending node.

In the early 1970's, the idea of network optimization was used by Turner and Miles [TM71] to model the route selection problem. This model [TM71] considered the square grid to define the network. Considering all of the cost factors, for each cell in the grid, a smooth surface was constructed. They [TM71] developed the Generalized Computer Aid Route Selection (GCARS) system to generate a set of ranked alignments. Turner [Tur78] further improved the GCARS system by incorporating the environmental impacts as a cost factor.

Athanassoulis and Calogero [AC73] formulated the route selection problem as a modified transportation problem. Note that both Turner's and Athanassoulis's models did not consider the vertical profile. In practice, it is highly expected to incorporate the vertical alignment cost in the horizontal alignment optimization process.

Parker [Par77] and Trietsch [Tri87b, Tri87a] developed the two stages approach considering the vertical profile and used network optimization to find the optimal alignment. While Parker studied only the square search grid, Trietsch studied four different types of search grids: rectangular, square, ellipse, and honeycomb. However, the resulting horizontal alignments of the models ([Par77, Tri87b, Tri87a]) are piece-wise linear curves which are unrealistic.

A basic shortcoming of the network optimization models is that the optimal alignment is a piecewise linear trajectory. In practice, a nonsmooth alignment is to be avoided for safety reasons. Moreover, in order to get a more precise alignment, more nodes are needed which increases the problem size and eventually the problem becomes hard to solve in a reasonable time.

A few studies [OEC73, Hog73, NEW76] applied the dynamic programming approach to optimize the alignment. Similar to the network optimization approach, the dynamic programming approach yields a nonsmooth alignment. Moreover, in the dynamic programming approach, the backward bending feature of the roads (backtracking road) introduces difficulties to handle the alignment [Nic73, page 123, Chapter 5], [JSJ06, page 21, Section 2.4.3], [Par77].

Jong et al. [Jon98, JJS00] developed a horizontal alignment optimization model which was solved by a genetic algorithm. The horizontal alignments represented in [Jon98, JJS00] are very different from a practical alignment [LTL09].

Lee et al. [LTL09] presented a heuristic based method to optimize the horizontal alignment that works in two stages. In the first stage, the heuristic tries to approximate a piecewise linear alignment and then in the second stage, it refines the solution to make the previously generated piecewise linear alignment compatible to a real road alignment. The solution alignment of the model ([LTL09]) yields a practical alignment. Since a heuristic algorithm was used to solve the model, optimality is not guaranteed.

1.3.3 Three dimensional alignment optimization model

In three dimensional alignment optimization, the vertical and horizontal alignments are optimized simultaneously. Modeling three dimensional problem is a complex problem [JSJ06] and most of the studies use heuristic based algorithms.

Chew et al. [CGF89] developed a model to solve the three dimensional alignment using the concept of optimal control theory. Chew's model is the first model that yields a smooth three dimensional alignment. The objective function of Chew's model involves integrals which are hard to compute.

Tat and Tao [TT03] proposed a three dimensional alignment optimization model and used genetic algorithm to solve it. This model [TT03] considers all of the major constraints of the road design. Akay [Aka06] developed a model for three dimensional alignment optimization for forest roads. A simulated annealing algorithm was used to solve the model. A tabu search method was presented by [Aru05] for optimizing the three dimensional align-

ment of forest roads.

A criteria-based decision support system for three dimensional alignment optimization was developed by Jha [Jha03] considering the environmental costs. Jong et al. [JS03] presented an evolutionary model for optimizing the vertical and horizontal alignment simultaneously. The previous two models [Jha03, JS03] were further improved in [JK06] by considering accessibility, proximity, and land-use changes in the road alignment planning process.

Recently, Cheng and Lee [CL06] also proposed a heuristic based model for three dimensional alignment optimization. The heuristic solves the models in three steps: first, it generates a good general horizontal alignment by adding, deleting, or moving the intersection points one by one, then it determines an improved horizontal alignment by adjusting the intersection points based on the previously generated horizontal alignment, and finally, it finds a better three dimensional alignment by tuning the vertical alignment corresponding to the previously obtained horizontal alignment.

All of the above mentioned three dimensional alignment optimization models excluding the model in [CGF89] use a heuristic based algorithm which does not guarantee optimality. Unfortunately, the heuristic based algorithms do not ensure any mathematical proof of convergence.

1.4 Our research approach

In this research, we formulate the horizontal alignment optimization problem as a bi-level optimization problem. Since the horizontal alignment optimization problem is interrelated to the vertical alignment optimization problem, in the inner level, the optimization model solves a vertical alignment optimization problem corresponding to a given horizontal alignment (which comes from the outer level). The outer level of the problem optimizes the horizontal alignment. An alignment in our model considers all of the geometric specifications used by engineers. We used two derivative free optimization algorithms to solve the problem. Our approach requires an initial alignment to start a derivative free optimization algorithm. The resulting solution alignment of our model is locally optimum.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we describe a basic formulation of the horizontal alignment optimization problem. We also discuss the solution approach to solve the problem. Some numerical

experiment results are reported for the basic model. It is shown that, for a road of reasonable length, the basic model cannot be solved in a reasonable time.

In Chapter 3, we develop a model using the concept of the basic model developed in Chapter 2. This model considers all of the geometric specifications used by engineers in practice. The resulting model is a bi-level optimization problem, in which the vertical alignment optimization problem is considered as an inner problem.

In Chapter 4, we report the numerical results obtained by solving the model developed in Chapter 3. The model was solved by two derivative-free optimization solvers. The performance of the two solvers are reported. Finally, in Chapter 5 we summarize the contribution of the thesis and highlight some future works.

Chapter 2

A basic approach to solve the horizontal alignment optimization problem

In this chapter, we describe a basic optimization problem formulation for the horizontal alignment optimization. We report numerical experiments that lead to the improved model described in Chapter 3.

2.1 Terminology

Horizontal alignment optimization consists of finding an optimal curve within a designated corridor. Each corridor has a *baseline*, which defines the horizontal and vertical alignment. There are two given end-points within a specified corridor. The curve connecting the two end-points is the baseline of the corridor. The ground profile data is given for some discrete points within the corridor, which are called *data points*.

There are two types of data points, namely, *base data points* and *offset data points*. The base data points are the points along the baseline, i.e., the engineer's original horizontal alignment. The offset data points represent the horizontal displacement from the base data points. The base data points are selected a few units apart between the two end-points along the baseline. Each of the base data points has some associated offset data points in both the left and the right directions. The red points and the black points in Figure 2.1 are the base data points and offset data points, respectively.

The baseline of a corridor is a piece-wise linear curve connecting the base data points. A base data point together with the associated offset data points is defined as a *station*.

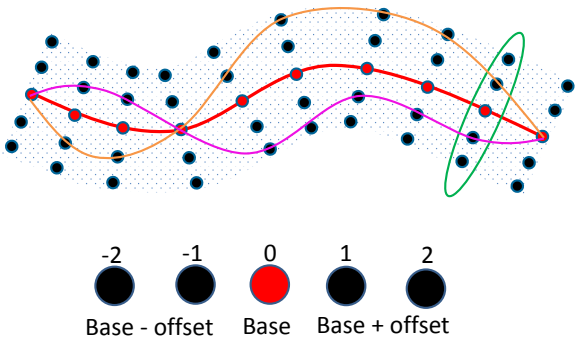

Figure 2.1: Corridor of a horizontal alignment. Base data points are in red, offset data points are in black. Points circled in green constitute an example of a station. The orange and purple curves are two potential horizontal alignments in the corridor.

2.2 Problem Formulation

Each data point within the corridor, either a base data point or an offset data point, has some associated ground profile data. Therefore, for the vertical road profile, we can move vertically up and down for each horizontal offset data point. Altogether a three dimensional alignment data is visualized as shown in Figure 2.2. Thus the horizontal and the vertical displacements from the baseline make a discrete grid for each station (see Figure 2.2). The horizontal displacement allows to move along the x and y axes of the grid points and the vertical displacement allows to move along the z axis. Our goal is to find, for each station, a horizontal offset that generates a horizontal alignment and a vertical alignment which is (locally) optimal.

Since every point in the designated corridor has the ground profile data, we can make an alignment by taking a point from each station and optimizing it as a vertical alignment optimization problem by fixing a horizontal alignment. For instance, the orange curve and the purple curve in Figure 2.1 could make a horizontal alignment and by fixing that particular alignment we get a vertical alignment optimization problem. Each station has some horizontal offset values associated with the offset data points. Note that at the base data point of each station the offset value is zero and at the left and right sides of base data points the offset is positive and negative, respectively. To formulate the horizontal alignment optimization problem, we can now consider the vertical alignment and the horizontal alignment

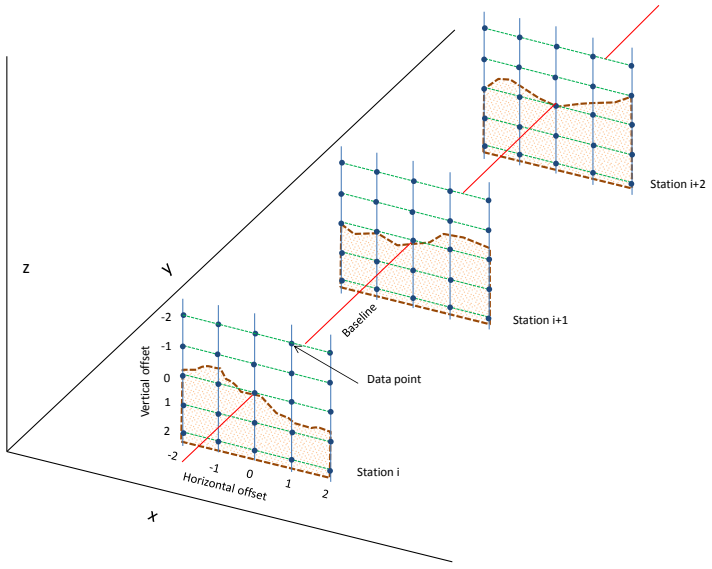

Figure 2.2: Three dimensional alignment.

together in a way that optimizes the vertical alignment cost by varying the horizontal offset value.

Let $\mathcal{S} = \{1, 2, 3, \ldots, n\}$ be the index set for the stations. The decision variable x_i is the horizontal offset value at station $i \in \mathcal{S}$. The lower bound and the upper bound of the horizontal offset value of station $i \in \mathcal{S}$ are defined as l_i and u_i, respectively. In vector form, the horizontal offset X, the lower bound L, and the upper bound U can be written as

$$\begin{aligned} X &= (x_1, x_2, x_3, \cdots, x_n)^\intercal, \\ L &= (l_1, l_2, l_3, \cdots, l_n)^\intercal, \text{ and} \\ U &= (u_1, u_2, u_3, \cdots, u_n)^\intercal. \end{aligned} \tag{2.1}$$

The dimension of the vector X is the number of stations. Now the problem can be written mathematically as follows:

$$\begin{aligned} &\min\ f(X) = \mathcal{C}_{\mathcal{VA}}(X) \\ &\quad \text{s.t} \\ &\qquad L \leq X \leq U, \\ &\qquad X \in \mathbb{R}^n. \end{aligned} \tag{2.2}$$

where $\mathcal{C}_{\mathcal{VA}}(X)$ is a function that returns the cost of an optimal vertical

alignment for a horizontal offset X. So the objective function of the proposed formulation is a vertical alignment optimization problem that can be calculated by the method proposed in [HHLR14].

2.3 Solution Approach

The objective function of the problem is an optimization problem itself that is a large scale mixed integer program. So it is very hard to access the derivative information of the objective function (if it exists). We applied a derivative-free optimization (DFO) approach to deal with the problem. In DFO approach, we can put the objective function and the bound constraints in a blackbox (see Figure 2.3) and optimize the problem without knowing much information on the objective function. Only a vector X is given as an input in the blackbox and an output value is obtained without knowing how the output is computed. To solve our problem we used the NOMAD [LD11] (see `http://www.gerad.ca/nomad`) solver that implements a mesh adaptive direct search (MADS) algorithm developed in [AD06]. The NOMAD solver is an open source solver that is proven to be competitive in solving DFO problems [RS13]. We can integrate it as a static library. The flowchart of the solving technique is illustrated in Figure 2.4.

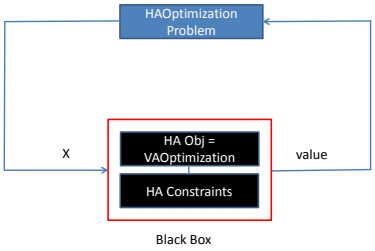

Figure 2.3: Blackbox optimization in horizontal alignment optimization problem.

The algorithm takes a three dimensional alignment data as the input data. The *VAOptimization* block makes the blackbox for the MADS algorithm. The MADS algorithm is an iterative algorithm that continues until the optimal solution (i.e. the mesh is small enough) is found. Every iteration of the MADS algorithm needs to evaluate the blackbox. So we are solving a vertical alignment optimization problem at each iteration. Therefore the optimal solution of the defined problem in Equation (2.2) gives us the optimal vertical and the horizontal alignment.

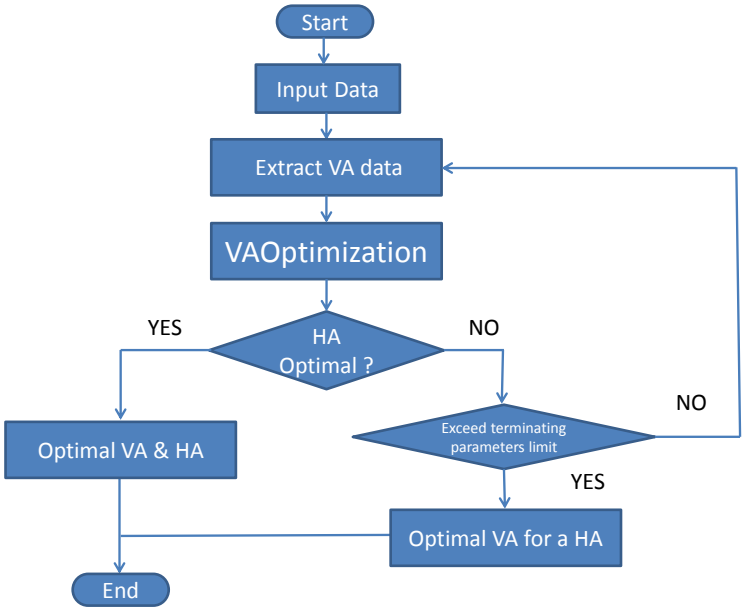

Figure 2.4: Flowchart of the solution approach.

The NOMAD solver has some terminating parameters to stop the MADS algorithm. For instance, we can set the maximum number of iterations for the MADS algorithm. So if the limit of the terminating parameters is exceeded, the algorithm terminates and eventually ends up with a non-optimal horizontal alignment but the optimal vertical alignment is guaranteed to be (globally) optimal for the given horizontal alignment.

The horizontal offsets given by the input data are discrete values but our formulation has continuous offset variables. The lowest and highest horizontal offsets are the lower and higher bounds, respectively. To get the ground profile data for any offset within the bound, linear interpolation is used. Let y_a and y_b be the ground profile data for the horizontal offset x_a and x_b, respectively. For any offset x within x_a and x_b the ground profile data is interpolated using the following equation:

$$y = y_a + (y_b - y_a)\,\frac{x - x_a}{x_b - x_a}. \tag{2.3}$$

2.4 Numerical Results

The experiments were setup with 50 test problems with a different number of stations. We used the derivative free optimization solver NOMAD (version 3.5, available in `http://www.gerad.ca/nomad`) to solve the optimization problems. All of the numerical experiments were performed on a Dell workstation with an Intel(R) Xenon(R) 2.40 GHz (2 cores) processor, 24 GB of RAM and a 64-bit Windows 7 Enterprise operating system.

We performed numerical experiments on five different roads. For each road, we created ten different test problems by varying the number of stations. To analyze the performance, we consider the number of function calls (blackbox evaluations) and the wall-clock time for each test problem. The full set of numerical data is included in Appendix A.1.

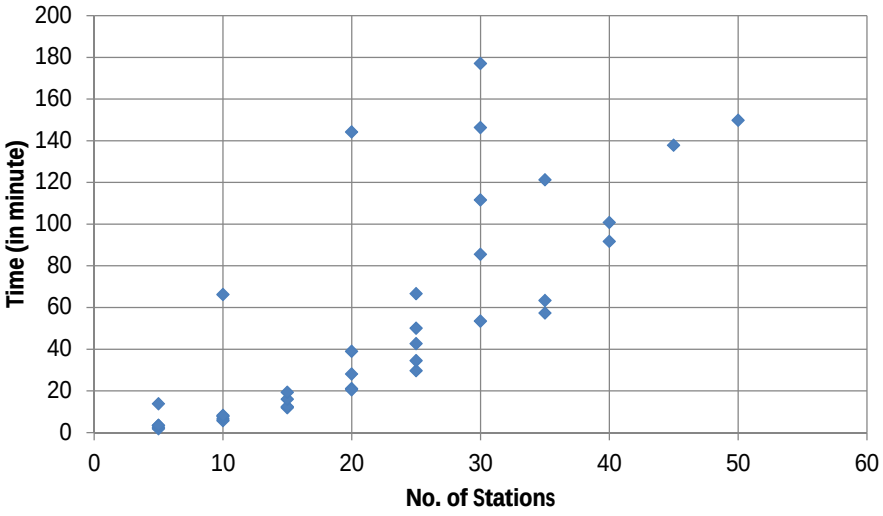

Figure 2.5: Required time with respect to number of stations for solved problem only.

Figures 2.5 and 2.6 shows the wall-clock time and the number of function calls required to solve the test problems. Since the numerical experiments were performed on five different roads, for every number of stations, we have five different test problems. As a terminating condition of the algorithm, we set timeout to 3 hours. When the number of stations is up to 30, all of the five test problems can be solved in 3 hours. From Figures 2.5, we can see that when the number of stations increased to 45 and 50 then only one problem can be solve within the time-limit. From Figure 2.6, we can

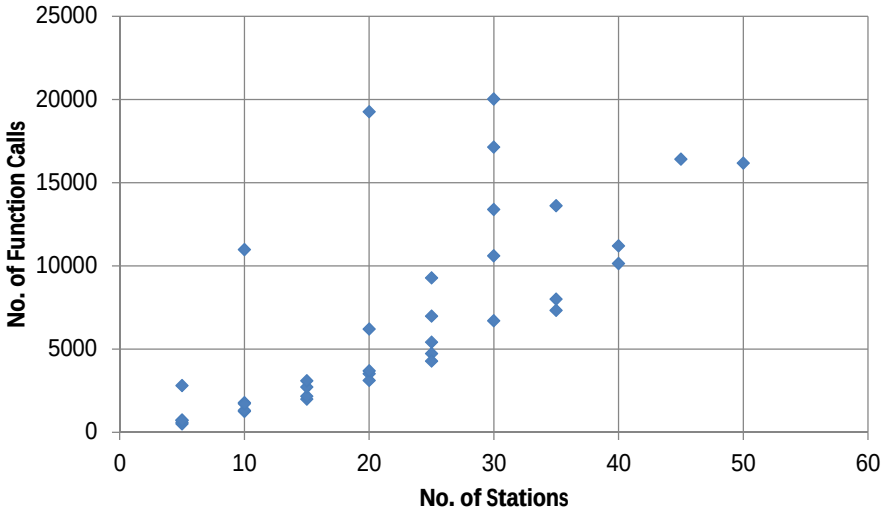

Figure 2.6: Growth of no. of function calls with respect to number of stations.

also observe that for a road of 45 or 50 stations, the solver required around 17250 function calls, which is a large number of function calls to solve a problem corresponding to a small road of 45 or 50 stations. It would be worth mentioning that a typical road is closer to 200 stations.

2.5 Summary

We have discussed a very straight-forward formulation of the horizontal alignment optimization problem. We introduced a derivative-free optimization approach to solve the problem. In this basic optimization model, the problem size increases as the number of stations increases. Usually, the NOMAD solver can only effectively handle a problem of a small number of variables; i.e., the problem size is less than 50 [LD11]. In Addition, the model yields a piece-wise linear curve, which is not used by engineers in practice. Considering all of the engineering specifications of a horizontal alignment, a more precise formulation could be developed for a road of a large number of stations (i.e., more than 100 stations). In the next chapter, we explain how to build a practical model.

Chapter 3

Horizontal alignment optimization model

In this chapter, we describe a horizontal alignment optimization model that produces a piece-wise linear-circular curve instead of the piecewise linear curve outputed by the basic model introduced in Chapter 2. We also describe in detail the geometric specifications of the model.

3.1 Geometric representation of horizontal alignment

In this model, a horizontal alignment consists of a sequence of circular curves and tangential lines. The circular curves and tangential lines are defined by some intersection points and the radius of curvature associated with each intersection point. In Figure 3.1, S and E are the start and end points of the alignment, respectively. The intersection points of the alignment are P_1, P_2, and P_3. Each intersection point has a radius of curvature that defines the circular curve. The radius of curvature associated with the intersection points P_1, P_2, and P_3 are r_1, r_2, and r_3. The purple and red portions of the alignment in Figure 3.1 are the circular curve and tangential line segments, respectively.

Let i be the index of the intersection points and n_p be the number of intersection points. Since the intersection point P_i has an associated radius of curvature r_i, we define an intersection point with radius of curvature as (P_i, r_i), where $P_i \in \mathbb{R}^2$ and $r_i \in \mathbb{R}$. Without loss of generality, we can say that the start and end points are a point in $\mathbb{R}^2$ with zero radius of curvature. The starting and end points are denoted as $(P_0, 0)$ and $(P_{n_p+1}, 0)$. So we represent a horizontal alignment HA as the sequence

$$HA = ((P_0, 0), (P_1, r_1), (P_2, r_2), \dots (P_{n_p}, r_{n_p}), (P_{n_p+1}, 0)). \tag{3.1}$$

To determine the actual horizontal alignment, we need to calculate the circular curves and tangential line segments from the given intersection

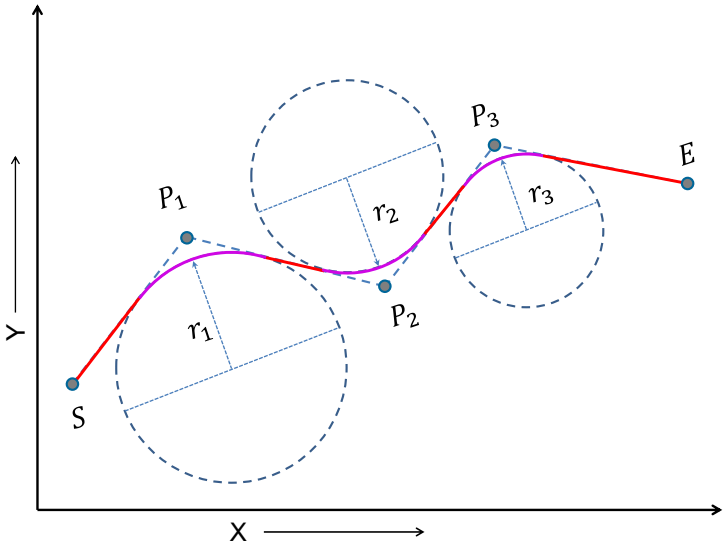

Figure 3.1: Geometric representation of horizontal alignment. The alignment displayed is represented as $((S,0), (P_1,r_1), (P_2,r_2), (P_2,r_2), (E,0))$.

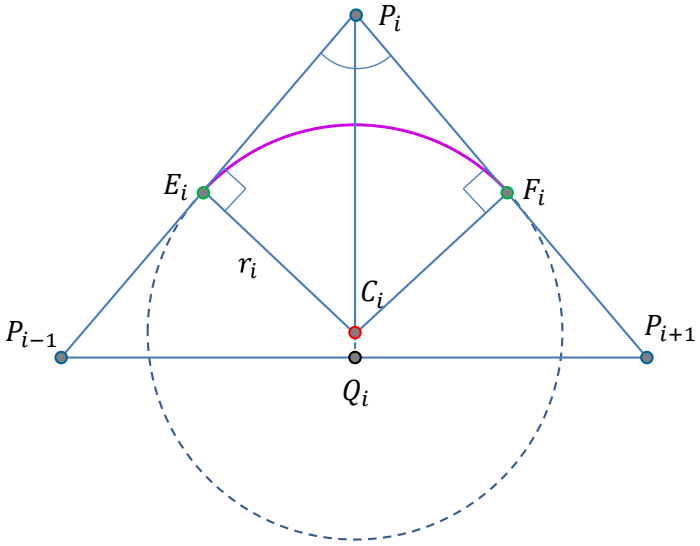

Figure 3.2: Geometric specifications of a circular curve.

points and the associated radius of curvatures. So for each intersection point P_i, we need to find out the two tangential points and the center of curvature. Let E_i and F_i be the left and right tangential point, respectively. In Figure 3.2 the green points are the two tangential points. Let C_i (red point in Figure 3.2) be the center of curvature that corresponds to the intersection point P_i. We can calculate E_i, F_i, and C_i for each intersection point P_i using the radius of curvature r_i and the three consecutive intersecting points P_{i-1}, P_i, and P_{i+1}.

Tangential point calculation

We define the following variables:

- θ_i is the angle at P_i using three consecutive intersection point P_{i-1}, P_i, and P_{i+1}.
- $\mathbf{U}_i$ is the vector from the point P_i to P_{i-1}.
- $\mathbf{V}_i$ is the vector from the point P_i to P_{i+1}.
- $\mathbf{W}_i$ is the vector from the point P_{i-1} to P_{i+1}.
- Q_i is the intersection point of the angle bisector of θ_i and the line joining P_{i-1} and P_{i+1}.

By definition, we have

$$\begin{aligned} \mathbf{U}_i &= P_{i-1} - P_i, \\ \mathbf{V}_i &= P_{i+1} - P_i, \\ \mathbf{W}_i &= P_{i+1} - P_{i-1}. \end{aligned} \tag{3.2}$$

We can calculate the angle θ_i by using the dot product of the vector $\mathbf{U}_i$ and $\mathbf{V}_i$.

$$\mathbf{U}_i \cdot \mathbf{V}_i = \|\mathbf{U}_i\|\|\mathbf{V}_i\| \cos\theta_i. \tag{3.3}$$

Using Equation (3.3) we have

$$\theta_i = \arccos\left(\frac{\mathbf{U}_i \cdot \mathbf{V}_i}{\|\mathbf{U}_i\|\|\mathbf{V}_i\|}\right). \tag{3.4}$$

Since the angle bisector P_iQ_i bisects the angle θ_i, we have

$$\angle P_{i-1}P_iQ_i = \frac{\theta_i}{2} = \angle P_{i+1}P_iQ_i. \tag{3.5}$$

P_iP_{i-1} and P_iP_{i+1} are the tangent to the circle at the tangential point E_i and F_i. Thus we have $C_iE_i \perp P_iP_{i-1}$ and $C_iF_i \perp P_iP_{i+1}$. The triangle $\triangle P_iE_iC_i$ and $\triangle P_iF_iC_i$ are right angle triangles. The segment P_iC_i is the common side of $\triangle P_iE_iC_i$ and $\triangle P_iF_iC_i$. Since $E_iC_i{=}F_iC_i$, we have $P_iE_i = P_iF_i$. Let l_t be the length of P_iE_i. In the triangle $\triangle P_iE_iC_i$, we have

$$\tan\frac{\theta_i}{2} = \frac{E_iC_i}{P_iE_i} = \frac{r_i}{l_t}. \tag{3.6}$$

So

$$l_t = \frac{r_i}{\tan\frac{\theta_i}{2}}. \tag{3.7}$$

Let $\hat{e}_{\mathbf{U}_i}$ and $\hat{e}_{\mathbf{V}_i}$ be the two unit vector of $\mathbf{U}_i$ and $\mathbf{V}_i$. The tangential point E_i and F_i can be calculated as follows:

$$E_i = P_i + l_t\hat{e}_{\mathbf{U}_i}, \tag{3.8}$$

$$F_i = P_i + l_t\hat{e}_{\mathbf{V}_i}. \tag{3.9}$$

Center point calculation

Fact 3.1 ([Byr47, Book VI Proposition III])**.** *The angle bisector of an angle in a triangle divides the opposite side in the same ratio as the sides adjacent to the angle.*

Let l_b be the length of the Q_iP_{i-1}. The length of P_iP_{i-1}, P_iP_{i+1} and $P_{i-1}P_{i+1}$ are $\|\mathbf{U}_i\|$, $\|\mathbf{V}_i\|$, and $\|\mathbf{W}_i\|$, respectively. P_iQ_i is the angle bisector of θ_i in $\triangle P_{i-1}P_iP_{i+1}$. So using Fact 3.1 we have

$$\frac{\|\mathbf{U}_i\|}{\|\mathbf{V}_i\|} = \frac{l_b}{\|\mathbf{W}_i\| - l_b}. \tag{3.10}$$

So

$$l_b = \frac{\|\mathbf{U}_i\|\|\mathbf{W}_i\|}{\|\mathbf{U}_i\| + \|\mathbf{V}_i\|}. \tag{3.11}$$

Let $\hat{e}_{\mathbf{W}_i}$ be the unit vector of $\mathbf{W}_i$. The point Q_i can be calculated as follows:

$$Q_i = P_{i-1} + l_b\hat{e}_{\mathbf{W}_i}. \tag{3.12}$$

Define $\mathbf{X}_i$ as

$$\mathbf{X}_i = Q_i - P_i. \tag{3.13}$$

Let l_x be the length of P_iC_i. In the triangle $\triangle P_iE_iC_i$, we have

$$\cos\frac{\theta_i}{2} = \frac{P_iE_i}{P_iC_i} = \frac{l_t}{l_x}. \tag{3.14}$$

So

$$l_x = \frac{l_t}{\cos\frac{\theta_i}{2}}. \tag{3.15}$$

Let $\hat{e}_{\mathbf{X}_i}$ be the unit vector of $\mathbf{X}_i$. So the center point C_i can be calculated as follows:

$$C_i = P_i + l_x\hat{e}_{\mathbf{X}_i}. \tag{3.16}$$

3.2 Model description

3.2.1 Definitions

In order to model the horizontal alignment, we group a set of consecutive stations to make a segment. We divide the entire corridor into m segments, which are indexed by a set $\mathcal{I}_\mathcal{G} = \{1, 2, 3 \ldots, m\}$. Every segment consists of a set of stations. The g^{th} segment has n_g stations. For all $g \in \mathcal{I}_\mathcal{G}$, the stations associated with the g^{th} segment are indexed by the set $\mathcal{I}_{\mathcal{SG}} = \{1, 2, 3, \ldots, n_g\}$. So the total number of stations is $n = \sum_{g\in\mathcal{I}_\mathcal{G}} n_g$. The g^{th} segment ($g \in \mathcal{I}_\mathcal{G}$) j^{th} station ($j \in \mathcal{I}_{\mathcal{SG}}$) is denoted by $S_{g,j}$. The stations of the corridor are indexed by the set $\mathcal{I}_\mathcal{S} = \{1, 2, 3, \ldots n\}$. The i^{th} ($i \in \mathcal{I}_\mathcal{S}$) station of the corridor is denoted by s_i. The set of all stations is $\mathcal{S} = \{s_1, s_2, s_3, \ldots s_n\}$. We define a function to map a station index of a segment to the actual station index (i.e. station index of the corridor) as follows:

$$\delta : (\mathcal{I}_\mathcal{G}, \mathcal{I}_{\mathcal{SG}}) \mapsto \mathcal{I}_\mathcal{S}. \tag{3.17}$$

Clearly, for all $g \in \mathcal{I}_\mathcal{G}$, $j \in \mathcal{I}_{\mathcal{SG}}$, $i = \delta(g, j) = \sum_{p=1}^{g-1} n_p + j \in \mathcal{I}_\mathcal{S}$. Therefore, at station $s_i \in \mathcal{S}$, we have $S_{g,j} = s_i$ for all $g \in \mathcal{I}_\mathcal{G}$, $j \in \mathcal{I}_{\mathcal{GS}}$. Each station has n_d data points, which are indexed by the set $\mathcal{I}_\mathcal{D} = \{1, 2, 3 \ldots, n_d\}$. The i^{th} station k^{th} ($k \in \mathcal{I}_\mathcal{D}$) data point is denoted by $D_{i,k}$. The set of n_d data points at station s_i is $\mathcal{D}_i = \{D_{i,1}, D_{i,2}, D_{i,3} \ldots D_{i,n_d}\}$. Corresponding to each data point, we have vertical road profile data which is defined by a vector $VA_{i,k}$. The vector $VA_{i,k}$ consists of the cut-fill areas of a material at different heights. For example, the data point (x, y) has cut-fill areas of a material at different heights (i.e., at some different z-values). The cut-fill areas constitute the vector $VA_{i,k}$ corresponding to the data point $D_{i,k}$. At

each station $s_i \in \mathcal{S}$, we have the leftmost and rightmost offset data points that define the boundary of the corridor. The leftmost and the rightmost data points of the i^{th} station are denoted by u_i and v_i and defined as follows:

$$\begin{aligned} u_i &= D_{i,1} \qquad \forall i \in \mathcal{I}_{\mathcal{S}}, \\ v_i &= D_{i,n_d} \qquad \forall i \in \mathcal{I}_{\mathcal{S}}. \end{aligned} \tag{3.18}$$

The set of the leftmost offset data points is $\mathcal{U} = \{u_1, u_2, u_3, \ldots, u_n\}$. The set of the rightmost offset data points is $\mathcal{V} = \{v_1, v_2, v_3, \ldots, v_n\}$. At the g^{th} segment j^{th} station, the leftmost and the rightmost data points are denoted by $U_{g,j}$ and $V_{g,j}$, respectively. So when $\delta(g, j) = i$, we have

$$\begin{aligned} U_{g,j} &= u_i, \\ V_{g,j} &= v_i. \end{aligned} \tag{3.19}$$

Every two consecutive segments share a common station. The last station of a segment is the first station of the next segment (see Figure 3.3). The following equations are satisfied for $g = 1, 2, 3, \ldots, m-1$.

$$\begin{aligned} S_{g,n_g} &= S_{g+1,1}, \\ U_{g,n_g} &= U_{g+1,1}, \\ V_{g,n_g} &= V_{g+1,1}, \\ D_{\delta(g,n_g),k} &= D_{\delta(g+1,1),k}. \end{aligned} \tag{3.20}$$

At each station, the line passing through the leftmost offset data point and the rightmost offset data point is defined as a *cross-section line* of the station, see Figure 2.1. The parametric equation of the cross-section line $L_i(t)$ of the i^{th} station $s_i \in \mathcal{S}$ with the leftmost offset data point u_i and the rightmost offset data point v_i is

$$L_i(t) = (1-t)u_i + tv_i \qquad \text{for } t \in \mathbb{R}. \tag{3.21}$$

Using the mapping in Equation (3.17), for the g^{th} segment j^{th} station, the cross-section line $L_{g,j}(t^l_{g,j})$ is

$$L_{g,j}(t) = (1-t)U_{g,j} + tV_{g,j} \qquad \text{for } t \in \mathbb{R}. \tag{3.22}$$

In Equation (3.22), if the parameter t is restricted to $[0, 1]$, we obtain a segment. The equation of the cross-section segment with end-points $U_{g,j}$ and $V_{g,j}$ is

$$\bar{L}_{g,j}(t) = (1-t)U_{g,j} + tV_{g,j} \qquad \text{for } t \in [0, 1]. \tag{3.23}$$

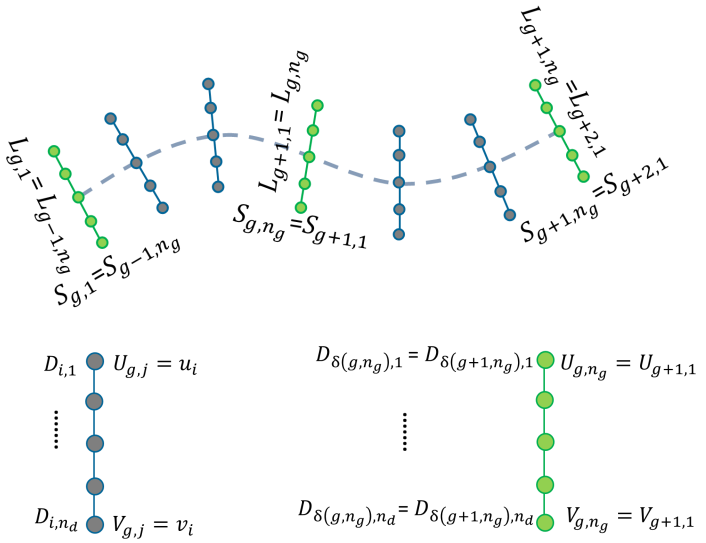

Figure 3.3: Road segment representation in a specified corridor.

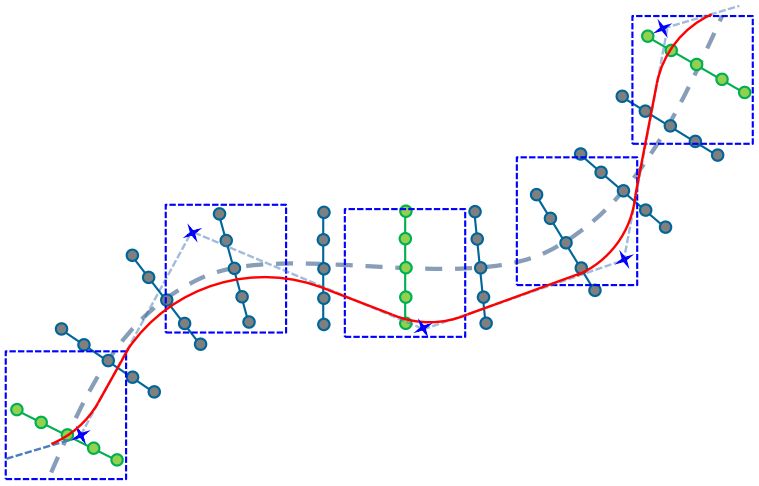

Figure 3.4: Horizontal alignment in a specified corridor. The red curve is an example horizontal alignment. The blue stars are the intersection points (IP) and the dotted blue rectangular boxes are feasible region (for moving) associated with the intersection points.

We assume that each segment has exactly three intersection points (except for the last segment which may have two) named $P_{g,1}$, $P_{g,2}$, and $P_{g,3}$

for the g^{th} segment. Each intersection point of a segment has a radius of curvature. For the g^{th} segment $r_{g,1}$, $r_{g,2}$, and $r_{g,3}$ are defined as the radius of curvatures corresponding to the intersection points $P_{g,1}$,$P_{g,1}$, and $P_{g,3}$, respectively. In Figure 3.4, the blue points are the intersection points and the red curve is the associated horizontal alignment. By moving the intersection points, we can build a wide variety of horizontal alignments.

Each intersection point has a feasible region. We define the feasible region of each intersection point by a rectangular box. The dotted blue rectangles in Figure 3.4, are the feasible region of the intersection points. A rectangular box is defined by the leftmost bottom corner point and the rightmost top corner point. Let $\underline{B_{g,1}}$ and $\overline{B_{g,1}}$ be the leftmost bottom corner point and the rightmost top corner point of the rectangular box associated with the intersection point $P_{g,1}$. So the rectangular box associated with $P_{g,1}$ is defined as $(\underline{B_{g,1}}, \overline{B_{g,1}})$. Similarly, the rectangular boxes associated with $P_{g,2}$ and $P_{g,2}$ are defined as $(\underline{B_{g,2}}, \overline{B_{g,2}})$ and $(\underline{B_{g,3}}, \overline{B_{g,3}})$, respectively.

3.2.2 The optimization model

The road design problem is formulated as a bi-level optimization problem. We can solve a vertical alignment optimization problem for a fixed horizontal alignment. So the main idea is to move the horizontal alignment and then minimize the vertical alignment.

Basic approach: one variable per station

At each station $s_i \in \mathcal{S}$ we are given the data points $D_{i,k}$ and corresponding vertical road profile data $VA_{i,k}$ $(\forall k \in \mathcal{I}_{\mathcal{D}})$ as input. At a station s_i, for an arbitrary data point $D_{i,a}$ the vertical road profile data $VA_{i,a}$ between two consecutive data points $D_{i,k}$ and $D_{i,k+1}$ along the cross-section line segment $\bar{L}_i(t)$ is interpolated using the following equation

$$VA_{i,a} = VA_{i,k} + (VA_{i,k+1} - VA_{i,k}) \frac{\|D_{i,a} - D_{i,k}\|}{\|D_{i,k+1} - D_{i,k}\|}. \tag{3.24}$$

Therefore, we have the vertical road profile data along every cross-section line segment $(\bar{L}_{g,j}(t) \ \ \forall g \in \mathcal{I}_{\mathcal{G}}, \forall j \in \mathcal{I}_{\mathcal{SG}})$. We can build a horizontal alignment by taking a point from each cross-section line segment. For instance, the red and purple piecewise linear curve in Figure 3.5 shows two different horizontal alignments for a road segment. For a fixed horizontal alignment, the optimal vertical alignment cost can be calculated by solving the vertical alignment optimization problem formulated in [HHLR14]. So the cost of a

horizontal alignment is the optimal vertical alignment cost corresponding to that horizontal alignment. Since we are given $D_{i,k}$ and $VA_{i,k}$ for all stations, we could build the cross-section line $L_{g,j}(t)$. Thus, by changing t between 0 to 1 of the $L_{g,j}(t)$, we can obtain any point $D_{i,a}$ on $L_{g,j}(t)$ and the associated $VA_{i,a}$ using Equation (3.24). Let $t^l_{g,j}$ be the parameter of the g^{th} segment j^{th} station. We can obtain all possible horizontal alignments by moving the parameter $t^l_{g,j}$ along the line $L_{g,j}(t^l_{g,j})$. The parameter values of the $L_{g,j}(t^l_{g,j})$ defines a horizontal alignment. So the parameters of the cross-section line can be used as variables. Let T_l be the variable vector defining the horizontal alignment as a piecewise linear function. We define

$$T_l = \langle \underbrace{t^l_{1,1}, t^l_{1,2}, \ldots, t^l_{1,n_1}}_{\text{Segment 1}}, \underbrace{t^l_{2,1}, t^l_{2,2}, \ldots, t^l_{2,n_2}}_{\text{Segment 2}}, \ldots\ldots, \underbrace{t^l_{m,1}, t^l_{m,2}, \ldots, t^l_{m,n_m}}_{\text{Segment m}} \rangle.$$

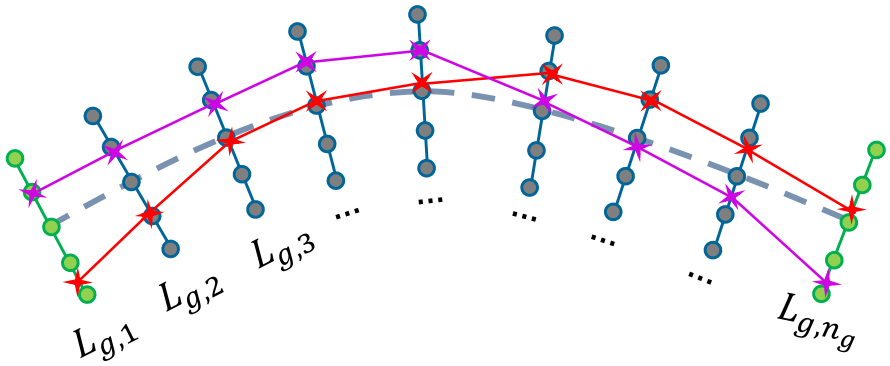

Figure 3.5: Piece-wise linear representation of horizontal alignment segment considering all of the cross-section lines.

The cost function of a horizontal alignment optimization problem associated with variable T_l is defined as $\mathcal{C}_{\mathcal{VA}}(T_l)$. The function $\mathcal{C}_{\mathcal{VA}}(T_l)$ gives the optimal vertical alignment cost for T_l. So the horizontal alignment optimization problem becomes

$$\min_{0 \le T_l \le 1} \mathcal{C}_{\mathcal{VA}}(T_l). \tag{3.25}$$

In practice, if we consider all cross-section lines (i.e., all of the stations), we have n_g variables for each segment. So the total number of variables is $\sum_{g=1}^{m} n_g = n$, which can be large. We can reduce the number of variables by defining the horizontal alignment geometrically.

Advanced approach: three variables per intersection point

Objective function and variables

We define a horizontal alignment using the intersection points and associated radius of curvatures. We have three intersecting points $P_{g,1}$,$P_{g,2}$, and $P_{g,3}$ for the g^{th} segment. By moving the intersection points and with different associated radius of curvatures, a wide variety of horizontal alignments can be built. So we can have the intersection points and the associated radius of curvature as the variables.

All of the intersection points are on the XY-plane. We define the intersection points $P_{g,1}$, $P_{g,2}$, and $P_{g,3}$ in Cartesian coordinate as follows:

$$\begin{aligned} P_{g,1} &= (p_{x_{g,1}}, p_{y_{g,1}}), \\ P_{g,2} &= (p_{x_{g,2}}, p_{y_{g,2}}), \\ P_{g,3} &= (p_{x_{g,3}}, p_{y_{g,3}}). \end{aligned} \tag{3.26}$$

Thus $p_{x_{g,1}}$, $p_{y_{g,1}}$, $p_{x_{g,2}}$, $p_{y_{g,2}}$, $p_{x_{g,3}}$, $p_{y_{g,3}}$, $r_{g,1}$,$r_{g,2}$, and $r_{g,3}$ are the variable for the g^{th} segment. We assume that the starting and end points of the alignment are fixed.

Two adjacent segments share an intersection point. Let $\hat{P}_{g,g+1} = (\hat{p}_{x_{g,g+1}}, \hat{p}_{y_{g,g+1}})$ be the common intersection point between the g^{th} and $(g+1)^{th}$ segments. So for two consecutive segments we have

$$\begin{aligned} P_{g,3} &= P_{g+1,1}, \\ \hat{P}_{g,g+1} &= P_{g,3}, \\ \hat{P}_{g,g+1} &= P_{g+1,1}. \end{aligned} \tag{3.27}$$

Let $\hat{r}_{g,g+1}$ be the radius of curvature corresponding to the point $\hat{P}_{g,g+1}$. So for two consecutive segments we have

$$\begin{aligned} r_{g,3} &= r_{g+1,1}, \\ \hat{r}_{g,g+1} &= r_{g,3}, \\ \hat{r}_{g,g+1} &= r_{g+1,1}. \end{aligned} \tag{3.28}$$

Let X be the variable vector of the optimization problem. We have

$$X = \langle \underbrace{p_{x_{1,2}}, p_{y_{1,2}}, \hat{p}_{x_{1,2}}, \hat{p}_{y_{1,2}}, r_{1,2}, \hat{r}_{1,2}}_{\text{Segment 1}}, \underbrace{p_{x_{2,2}}, p_{y_{2,2}}, \hat{p}_{x_{2,3}}, \hat{p}_{y_{2,3}}, r_{2,2}, \hat{r}_{2,3}}_{\text{Segment 2}} \cdots\cdots$$
$$\underbrace{p_{x_{m-1,2}}, p_{y_{m-1,2}}, \hat{p}_{x_{m,m-1}}, \hat{p}_{y_{m-1,m}}, r_{m-1,2}, \hat{r}_{m-1,m}}_{\text{Segment m-1}}, \underbrace{p_{x_{m,2}}, p_{y_{m,2}}, r_{m,2}}_{\text{Segment m}} \rangle.$$

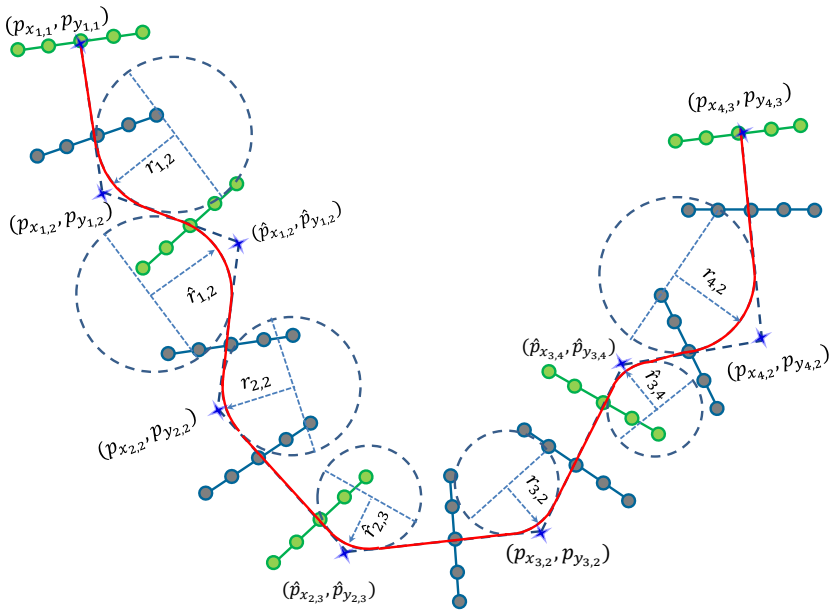

Figure 3.6: An example road of four segments showing the associated variables of the optimization model. The green cross-section lines separate the road segments. The starting point and the end point of the alignment are $(p_{x_{1,1}}, p_{y_{1,1}})$ and $(p_{x_{4,3}}, p_{y_{4,3}})$, respectively, which are fixed.

In Figure 3.6, for a road of four segments, the associated variables are depicted. Note that the last segment might have two intersection points (when the number of intersection points (IP) is not divisible by 3). In that case, the last segment has only variables corresponding to the common intersection point. For instance, if we assume Figure 3.6 does not have the intersection point $(p_{y_{4,2}}, p_{y_{4,2}})$; i.e., the number of intersection points is 8, then the last segment has only the variables $(\hat{p}_{x_{3,4}}, \hat{p}_{y_{3,4}})$ and $\hat{r}_{3,4}$, which are also considered in the previous (third) segment.

A horizontal alignment for a road segment consists of some circular curves and tangential lines. For each segment the horizontal alignment curve has three circular curves and two tangential lines (see Figure 3.7). The purple portions and the red portions of the horizontal alignment curve in Figure 3.7 are circular curves and tangential lines, respectively. Let $E_{g,1}$, $E_{g,2}$, and $E_{g,3}$ be the left tangential points; and $F_{g,1}$, $F_{g,2}$, and $F_{g,3}$ be the right tangential points correspond to the intersection points $P_{g,1}$, $P_{g,2}$, and

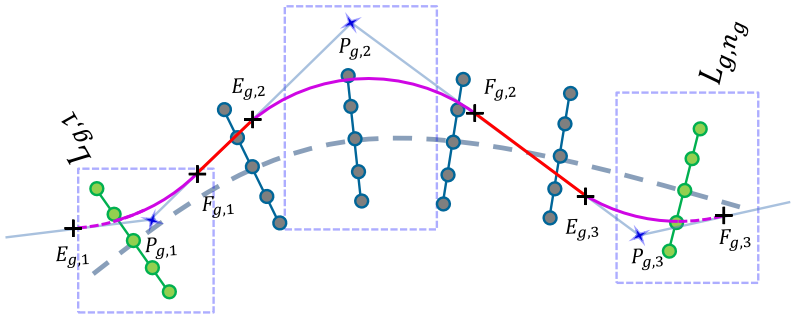

Figure 3.7: A segment of a road showing an associated horizontal alignment.

$P_{g,3}$. The left and right tangential points can be calculated using Equation (3.8) and Equation (3.9) . Let $C_{g,1}$, $C_{g,2}$, and $C_{g,3}$ be the center of curvature corresponds to the intersection point $P_{g,1}$,$P_{g,2}$, and $P_{g,2}$. The center of curvature can be calculated using Equation (3.16). Since all points are in XY-plane, we define

$$\begin{aligned} &C_{g,1} = (c_{x_{g,1}}, c_{y_{g,1}}), C_{g,2} = (c_{x_{g,2}}, c_{y_{g,2}}), C_{g,3} = (c_{x_{g,3}}, c_{y_{g,3}}), \\ &E_{g,1} = (e_{x_{g,1}}, e_{y_{g,1}}), E_{g,2} = (e_{x_{g,2}}, e_{y_{g,2}}), E_{g,3} = (e_{x_{g,3}}, e_{y_{g,3}}), \\ &F_{g,1} = (f_{x_{g,1}}, f_{y_{g,1}}), F_{g,2} = (f_{x_{g,2}}, f_{y_{g,2}}), F_{g,3} = (f_{x_{g,3}}, f_{y_{g,3}}). \end{aligned} \tag{3.29}$$

Let $H_{g,1}$ $H_{g,2}$, $H_{g,3}$, $H_{g,4}$, and $H_{g,5}$ be five parametric pieces of the horizontal alignment curve for the g^{th} segment. $H_{g,1}$, $H_{g,2}$, and $H_{g,3}$ are the circular arc corresponding to the intersection points $P_{g,1}$, $P_{g,2}$, and $P_{g,3}$. $H_{g,4}$ is the tangential lines connecting the two arcs $H_{g,1}$ and $H_{g,2}$. $H_{g,5}$ is the tangential lines connecting the two arcs $H_{g,2}$ and $H_{g,3}$. Let $t^c_{g,1}$, $t^c_{g,2}$, $t^c_{g,3}$, $t^c_{g,4}$, and $t^c_{g,5}$ be the parameters of $H_{g,1}$ $H_{g,2}$, $H_{g,3}$, $H_{g,4}$, and $H_{g,5}$, respectively.

The parametric equation of the circle corresponding to the intersection point $P_{g,1}$ can be written as follows:

$$\begin{bmatrix} x(t^c_{g,1}) \\ y(t^c_{g,1}) \end{bmatrix} = \begin{bmatrix} r_{g,1}\cos(t^c_{g,1}) + c_{x_{g,1}} \\ r_{g,1}\sin(t^c_{g,1}) + c_{y_{g,1}} \end{bmatrix} \quad \text{for } t^c_{g,1} \in [0, 2\pi]\,. \tag{3.30}$$

Equation (3.30) gives the full circle but we need a circular arc with two endpoints. So we need to calculate the bounds of $t^c_{g,1}$ in Equation (3.30) to get the endpoints of a circular arc. The tangential point $E_{g,1}$ and $F_{g,1}$ are the two endpoints of the circular arc associated with the intersection point

$P_{g,1}$. Since $E_{g,1}$ and $F_{g,1}$ are on the circle, we have

$$\begin{bmatrix} r_{g,1}\cos(t^c_{g,1}) + c_{x_{g,1}} \\ r_{g,1}\sin(t^c_{g,1}) + c_{y_{g,1}} \end{bmatrix} = \begin{bmatrix} e_{x_{g,1}} \\ e_{y_{g,1}} \end{bmatrix}, \tag{3.31}$$

and

$$\begin{bmatrix} r_{g,1}\cos(t^c_{g,1}) + c_{x_{g,1}} \\ r_{g,1}\sin(t^c_{g,1}) + c_{y_{g,1}} \end{bmatrix} = \begin{bmatrix} f_{x_{g,1}} \\ f_{y_{g,1}} \end{bmatrix}. \tag{3.32}$$

Equation (3.31) and Equation (3.32) give the two values of t that make the bounds for the circular arc with two end points $E_{g,1}$ and $F_{g,1}$. From Equation (3.31) we deduce

$$t^c_{g,1} = \arccos\frac{e_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} \text{ or } \arcsin\frac{e_{y_{g,1}} - c_{y_{g,1}}}{r_{g,1}}. \tag{3.33}$$

Similarly, from Equation (3.32) we have

$$t^c_{g,1} = \arccos\frac{f_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} \text{ or } \arcsin\frac{f_{y_{g,1}} - c_{y_{g,1}}}{r_{g,1}}. \tag{3.34}$$

The ranges of arccos and arcsin are $[0, \pi]$ and $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, respectively. But the parameter $t^c_{g,1}$ of the full circle in Equation (3.30) varies from 0 to 2π. In order to handle this issue, first we need to identify the quadrant of the circle in which the two endpoints $E_{g,1}$ and $F_{g,1}$ lie on, and then adjust the value of $t^c_{g,1}$ to calculate the actual value with respect to the full circle. Let $t^e_{g,1}$ and $t^f_{g,1}$ be the two parameter values corresponding to the two endpoints $E_{g,1}$ and $F_{g,1}$, respectively. The value of the parameter associated with the endpoint $E_{g,1}$ can be written as follows:

$$t^e_{g,1} = \begin{cases} \arcsin\frac{e_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{e_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} \geq 0 \text{ and } \frac{e_{y_{g,1}} - c_{y_{g,1}}}{r_{g,1}} \geq 0, \\ \pi - \arcsin\frac{e_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{e_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} < 0 \text{ and } \frac{e_{y_{g,1}} - c_{y_{g,1}}}{r_{g,1}} \geq 0, \\ \pi - \arcsin\frac{e_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{e_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} < 0 \text{ and } \frac{e_{y_{g,1}} - c_{y_{g,1}}}{r_{g,1}} < 0, \\ 2\pi + \arcsin\frac{e_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{e_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} \geq 0 \text{ and } \frac{e_{y_{g,1}} - c_{y_{g,1}}}{r_{g,1}} < 0. \end{cases} \tag{3.35}$$

Similarly, associated with the endpoint $F_{g,1}$, the value of the parameter is

$$t^f_{g,1} = \begin{cases} \arcsin \frac{f_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{f_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} \geq 0 \text{ and } \frac{f_{y_{g,1}} - c_{y_{g,1}}}{r_{g,1}} \geq 0, \\ \pi - \arcsin \frac{f_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{f_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} < 0 \text{ and } \frac{f_{y_{g,1}} - c_{y_{g,1}}}{r_{g,1}} \geq 0, \\ \pi - \arcsin \frac{f_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{f_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} < 0 \text{ and } \frac{f_{y_{g,1}} - c_{y_{g,1}}}{r_{g,1}} < 0, \\ 2\pi + \arcsin \frac{f_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{f_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} \geq 0 \text{ and } \frac{f_{y_{g,1}} - c_{y_{g,1}}}{r_{g,1}} < 0. \end{cases} \tag{3.36}$$

Note that the four cases in Equation (3.35) and Equation (3.36) represent the first, second, third and fourth quadrants of the circle, respectively.

In Equation (3.35) and Equation (3.36), we do not know which value is the upper bound or lower bound of the circular arc connecting the two endpoints $E_{g,1}$ and $F_{g,1}$. So the lowest and highest of the two values are the lower bound and the upper bound, respectively. Let $\underline{t_{g,1}}$ and $\overline{t_{g,1}}$ be the upper bound and the lower bound for the circular arc $H_{g,1}$. We have

$$\begin{aligned} \underline{t_{g,1}} &= \min\left\{t^e_{g,1}, t^f_{g,1}\right\}, \\ \overline{t_{g,1}} &= \max\left\{t^e_{g,1}, t^f_{g,1}\right\}. \end{aligned} \tag{3.37}$$

Note that if any of the two endpoints lies on the fourth quadrant and the other endpoint lies on the first quadrant then the value of the parameter corresponding to the first quadrant has to be added by 2π, otherwise a wrong arc will be generated. In Figure 3.8 an example of this issue is illustrated. For the example shown in Figure 3.8, using the formulas in Equations (3.35), (3.36), and (3.37), we can calculate the lower bound and upper bound of the arc as $\frac{\pi}{4}$ and $\frac{7\pi}{4}$ which generates the red arc. In Figure 3.8, we can observe that the correct arc is the green arc rather than the red arc. So in order to generate the green arc, we have to add 2π to the associated parameter value of the endpoint $E_{g,1}$ (which lies on the first quadrant). After adding 2π to $\frac{\pi}{4}$ (the value associated with $E_{g,1}$), we get the new lower bound and upper bound as $\frac{7\pi}{4}$ and $\frac{9\pi}{4}$, respectively, which generates the green arc in Figure 3.8.

Let $\underline{t_{g,2}}$, $\underline{t_{g,3}}$ be the lower bounds and $\overline{t_{g,2}}$, $\overline{t_{g,3}}$ be the upper bounds for the circular arcs $H_{g,2}$ and $H_{g,3}$, respectively. The lower and upper bound for $H_{g,2}$ and $H_{g,3}$ can be calculated similarly as in Equation (3.37). So the parametric equation of the circular arcs $H_{g,1}$, $H_{g,2}$, and $H_{g,3}$ are

$$H_{g,1}(t^c_{g,1}) = \begin{bmatrix} r_{g,1}\cos(t^c_{g,1}) + c_{x_{g,1}} \\ r_{g,1}\sin(t^c_{g,1}) + c_{y_{g,1}} \end{bmatrix} \quad \text{for } t^c_{g,1} \in \left[\underline{t_{g,1}}, \overline{t_{g,1}}\right], \tag{3.38}$$

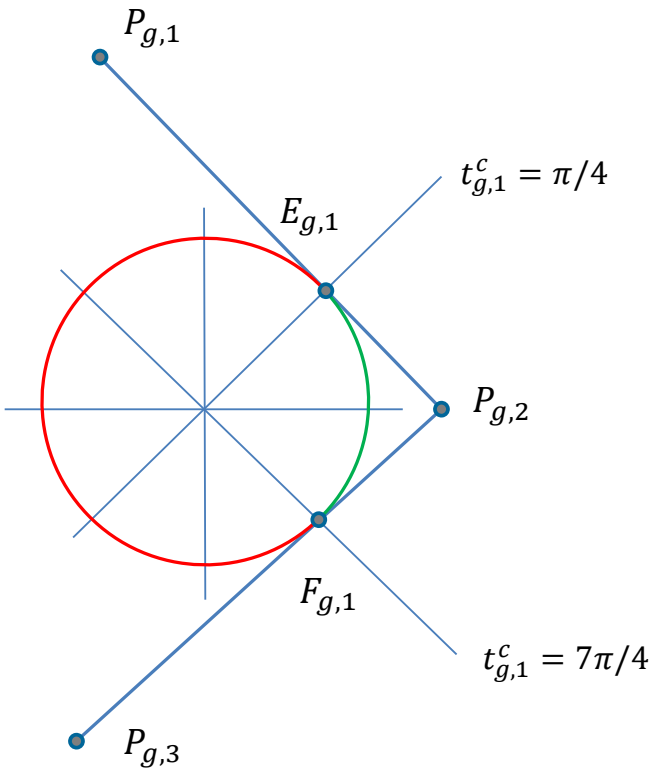

Figure 3.8: An example of the quadrants issue in generation of a circular arc.

$$H_{g,2}(t^c_{g,2}) = \begin{bmatrix} r_{g,2}\cos(t^c_{g,2}) + c_{x_{g,2}} \\ r_{g,2}\sin(t^c_{g,2}) + c_{y_{g,2}} \end{bmatrix} \quad \text{for } t^c_{g,2} \in \left[\underline{t_{g,2}}, \overline{t_{g,2}}\right], \tag{3.39}$$

and

$$H_{g,3}(t^c_{g,2}) = \begin{bmatrix} r_{g,3}\cos(t^c_{g,3}) + c_{x_{g,3}} \\ r_{g,3}\sin(t^c_{g,3}) + c_{y_{g,3}} \end{bmatrix} \quad \text{for } t^c_{g,3} \in \left[\underline{t_{g,3}}, \overline{t_{g,3}}\right]. \tag{3.40}$$

The tangential line segments $H_{g,4}$ connects the endpoints $F_{g,1}$ and $E_{g,2}$. The parametric equation of $H_{g,4}$ is

$$H_{g,4}(t^c_{g,4}) = (1 - t^c_{g,4})F_{g,1} + t^c_{g,4}E_{g,2} \quad \text{for } t^c_{g,4} \in [0,1]. \tag{3.41}$$

Similarly, the tangential line segments $H_{g,5}$ connects the endpoints $F_{g,2}$ and $E_{g,3}$. The parametric equation of $H_{g,5}$ is

$$H_{g,5}(t^c_{g,5}) = (1 - t^c_{g,5})F_{g,2} + t^c_{g,5}E_{g,3} \quad \text{for } t^c_{g,5} \in [0,1]\,. \tag{3.42}$$

In order to calculate the cost of the horizontal alignment, we need to compute parameter $t^l_{g,j}$ of the cross-section lines. We have two different cases: finding the intersection parameters for a circular arc and a line segment and finding the intersection parameters for two line segments (i.e., a cross section line segment and a tangential line segment).

For each segment, we have a set of cross-section lines $\left\{L_{g,1}, \ldots, L_{g,n_g}\right\}$. A cross-section line $L_{g,j}(t^l_{g,j}), j \in \mathcal{I}_{\mathcal{SG}}$ of the g^{th} segment is

$$L_{g,j}(t^l_{g,j}) = (1 - t^l_{g,j})U_{g,j} + t^l_{g,j}V_{g,j} \quad \text{for } t^l_{g,j} \in \mathbb{R}. \tag{3.43}$$

Since $U_{g,j}$ and $V_{g,j}$ are in the XY-plane, we define $U_{g,j} = (u_{x_{g,j}}, u_{y_{g,j}})$ and $V_{g,j} = (v_{x_{g,j}}, v_{y_{g,j}})$. Equation (3.43) can be written as follows:

$$\begin{aligned} x &= u_{x_{g,j}} + (v_{x_{g,j}} - u_{x_{g,j}})t^l_{g,j}, \\ y &= u_{y_{g,j}} + (v_{y_{g,j}} - u_{y_{g,j}})t^l_{g,j}. \end{aligned} \tag{3.44}$$

The equation of the circle that corresponds to the intersection point $P_{g,1}$ in implicit form is

$$(x - c_{x_{g,1}})^2 + (y - c_{y_{g,1}})^2 - {r_{g,1}}^2 = 0. \tag{3.45}$$

Substituting for x and y from Equation (3.44) into the Equation (3.45) gives a quadratic equation of $t^l_{g,j}$:

$$(u_{x_{g,j}} + (v_{x_{g,j}} - u_{x_{g,j}})t^l_{g,j} - c_{x_{g,1}})^2 + (u_{y_{g,j}} + (v_{y_{g,j}} - u_{y_{g,j}})t^l_{g,j} - c_{y_{g,1}})^2 - {r_{g,1}}^2 = 0. \tag{3.46}$$

The two roots of Equation (3.46) give the points on the line that cuts the circle. We deduce

$$t^l_{g,j} = \frac{a \pm \sqrt{\Delta}}{b}, \tag{3.47}$$

where

$$\begin{aligned} a =& (v_{x_{g,j}} - u_{x_{g,j}})(c_{x_{g,1}} - u_{x_{g,j}}) + (v_{y_{g,j}} - u_{y_{g,j}})(c_{y_{g,1}} - u_{y_{g,j}}), \\ \Delta =& {r_{g,1}}^2((v_{x_{g,j}} - u_{x_{g,j}})^2 + (v_{y_{g,j}} - u_{y_{g,j}})^2) \\ & - ((v_{x_{g,j}} - u_{x_{g,j}})(c_{y_{g,1}} - u_{y_{g,j}}) - (v_{y_{g,j}} - u_{y_{g,j}})(c_{x_{g,1}} - u_{x_{g,j}}))^2, \text{ and} \\ b =& (v_{x_{g,j}} - u_{x_{g,j}})^2 + (v_{y_{g,j}} - u_{y_{g,j}})^2. \end{aligned}$$

The two different values of $t^l_{g,j}$ give the two intersection points of Equation (3.44) and Equation (3.45) on the line $L_{g,j}(t^l_{g,j})$. The roots maybe similar, in that case, the line intersect only in a single point. If the roots are imaginary then there is no intersection. Let $S_{g,j} = (s_{x_{g,j}}, s_{y_{g,j}})$ be an intersection point obtained by $t^l_{g,j}$. To calculate the parameter of the circle in Equation (3.30) corresponding to the point $S_{g,j}$, we deduce

$$\begin{bmatrix} r_{g,1}\cos(t^c_{g,1}) + c_{x_{g,1}} \\ r_{g,1}\sin(t^c_{g,1}) + c_{y_{g,1}} \end{bmatrix} = \begin{bmatrix} s_{x_{g,1}} \\ s_{y_{g,1}} \end{bmatrix}. \tag{3.48}$$

So

$$t^c_{g,1} = \arccos\frac{s_{x_{g,1}} - c_{x_{g,1}}}{r_{g,1}} \text{ or } \arcsin\frac{s_{y_{g,1}} - c_{y_{g,1}}}{r_{g,1}}. \tag{3.49}$$

As we described earlier, due to the quadrant issue of the circle, the value of the parameter $t^c_{g,1}$ has to be adjusted. Let $t^s_{g,1}$ be the parameter value corresponding to the point $S_{g,j}$. We have

$$t^s_{g,1} = \begin{cases} \arcsin\frac{s_{x_{g,1}}-c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{s_{x_{g,1}}-c_{x_{g,1}}}{r_{g,1}} \geq 0 \text{ and } \frac{s_{y_{g,1}}-c_{y_{g,1}}}{r_{g,1}} \geq 0, \\ \pi - \arcsin\frac{f_{x_{g,1}}-c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{s_{x_{g,1}}-c_{x_{g,1}}}{r_{g,1}} < 0 \text{ and } \frac{s_{y_{g,1}}-c_{y_{g,1}}}{r_{g,1}} \geq 0, \\ \pi - \arcsin\frac{f_{x_{g,1}}-c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{s_{x_{g,1}}-c_{x_{g,1}}}{r_{g,1}} < 0 \text{ and } \frac{s_{y_{g,1}}-c_{y_{g,1}}}{r_{g,1}} < 0, \\ 2\pi + \arcsin\frac{f_{x_{g,1}}-c_{x_{g,1}}}{r_{g,1}} & \text{if } \frac{s_{x_{g,1}}-c_{x_{g,1}}}{r_{g,1}} \geq 0 \text{ and } \frac{s_{y_{g,1}}-c_{y_{g,1}}}{r_{g,1}} < 0. \end{cases} \tag{3.50}$$

If $t^l_{g,j} \in [0,1]$ and $t^s_{g,1} \in \left[\underline{t_{g,1}}, \overline{t_{g,1}}\right]$ then the intersection point is in the corridor. In this case, we accept the value of the parameter $t^l_{g,j}$, otherwise we reject the value. Similarly, for all other circular arcs we can calculate the value of the parameter $t^l_{g,j}$.

Now we need to calculate the intersection point between the cross-sectional line and the tangential line segment. The equation of the tangential line segment in Equation (3.41) can be written as follows:

$$\begin{aligned} x &= f_{x_{g,1}} + (e_{x_{g,2}} - f_{x_{g,1}})t^c_{g,4}, \\ y &= f_{y_{g,1}} + (e_{y_{g,2}} - f_{y_{g,1}})t^c_{g,4}. \end{aligned} \tag{3.51}$$

To calculate the intersection parameter of a cross-section line and a tangential line, from Equation (3.51) and Equation (3.44) we have the following linear system of equations:

$$\begin{aligned} u_{x_{g,j}} + (v_{x_{g,j}} - u_{x_{g,j}})t^l_{g,j} &= f_{x_{g,1}} + (e_{x_{g,2}} - f_{x_{g,1}})t^c_{g,4}, \\ u_{y_{g,j}} + (v_{y_{g,j}} - u_{y_{g,j}})t^l_{g,j} &= f_{y_{g,1}} + (e_{y_{g,2}} - f_{y_{g,1}})t^c_{g,4}. \end{aligned} \tag{3.52}$$

Equations (3.52) in matrix form are

$$\begin{bmatrix} v_{x_{g,j}} - u_{x_{g,j}} & e_{x_{g,2}} - f_{x_{g,1}} \\ v_{y_{g,j}} - u_{y_{g,j}} & e_{y_{g,2}} - f_{y_{g,1}} \end{bmatrix} \begin{bmatrix} t^l_{g,j} \\ t^c_{g,4} \end{bmatrix} = \begin{bmatrix} f_{x_{g,1}} - u_{x_{g,j}} \\ f_{y_{g,1}} - u_{y_{g,j}} \end{bmatrix}. \tag{3.53}$$

The solution for $t^l_{g,j}$ and $t^c_{g,4}$ is

$$\begin{bmatrix} t^l_{g,j} \\ t^c_{g,4} \end{bmatrix} = \begin{bmatrix} v_{x_{g,j}} - u_{x_{g,j}} & e_{x_{g,2}} - f_{x_{g,1}} \\ v_{y_{g,j}} - u_{y_{g,j}} & e_{y_{g,2}} - f_{y_{g,1}} \end{bmatrix}^{-1} \begin{bmatrix} f_{x_{g,1}} - u_{x_{g,j}} \\ f_{y_{g,1}} - u_{y_{g,j}} \end{bmatrix}. \tag{3.54}$$

In order to calculate the solution defined in Equation (3.54), the coefficient matrix has to be invertible. Thus in Equation (3.52), if the coefficient matrix is not full-rank, the system has no solution (i.e. the two lines are parallel). If the value of both $t^l_{g,j}$ and $t^c_{g,4}$ are in $[0, 1]$, we accept the solution. Otherwise, the solution point is outside of the corridor.

If all $t^l_{g,j} \in [0, 1]$ we can compute the optimal vertical alignment cost for a horizontal alignment defined by X. On the other hand, if any $t^l_{g,j} \notin [0, 1]$, the alignment is outside of the corridor. In this case, we set the optimal vertical alignment cost to infinity. Finally, the objective function of the horizontal alignment optimization can be written as

$$f(X) = \begin{cases} \mathcal{C}_{\mathcal{VA}}(X) & \text{if } t^l_{g,j} \in [0, 1] \ \ \forall g \in \mathcal{I}_{\mathcal{G}}, j \in \mathcal{I}_{\mathcal{SG}}, \\ \infty & \text{otherwise.} \end{cases} \tag{3.55}$$

Constraints

A horizontal curve consists of the tangential line segments followed by the circular arcs. Two consecutive circular arcs are connected by a tangential line. The horizontal alignment will be discontinuous when two circular arcs overlap along the tangential line (see Figure 3.9).

For the g^{th} segment, we have two tangential lines. The line passing through the intersection points $P_{g,1}$ and $P_{g,2}$ and the line passing through the intersection points $P_{g,2}$ and $P_{g,3}$ are the two tangential lines. In order to maintain continuity on the line passing through the intersection points $P_{g,1}$ and $P_{g,2}$, the length of $P_{g,1}P_{g,2}$ must be greater than or equal to the summation of the length of $P_{g,1}F_{g,1}$ and the length of $P_{g,2}E_{g,2}$. So we can write the continuity constraints as follows:

$$\|P_{g,2} - P_{g,1}\| \geq \|P_{g,1} - F_{g,1}\| + \|P_{g,2} - E_{g,2}\|. \tag{3.56}$$

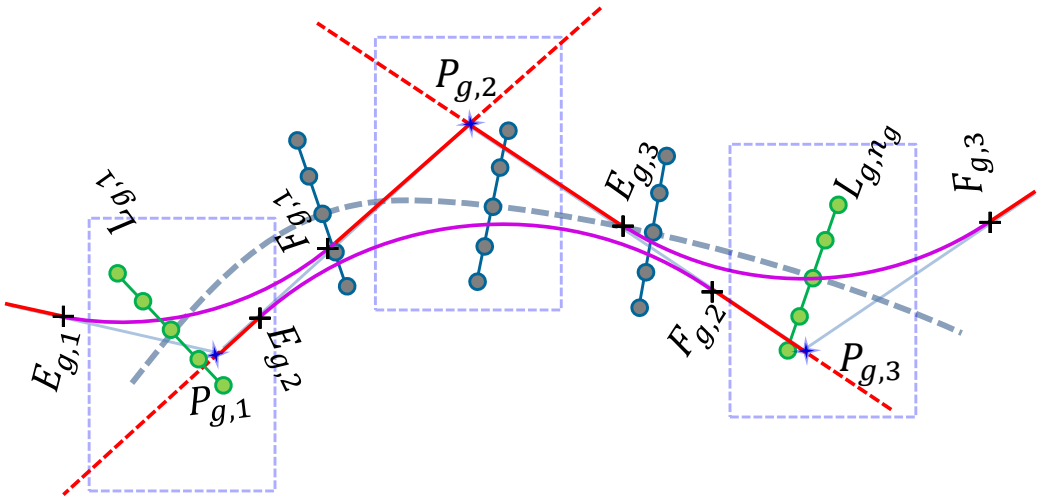

Figure 3.9: Discontinuity in a horizontal alignment.

Similarly, on the line passing through the intersection points $P_{g,3}$ and $P_{g,2}$, the continuity constraint becomes

$$\|P_{g,2} - P_{g,3}\| \geq \|P_{g,3} - E_{g,3}\| + \|P_{g,2} - F_{g,2}\|. \tag{3.57}$$

Each intersection point has a feasible region. The feasible region is defined by a rectangular box. We define the box corner points in Cartesian coordinate as follows:

$$\begin{aligned}
\underline{B_{g,1}} &= (\underline{b}_{x_{g,1}}, \underline{b}_{y_{g,1}}), \overline{B_{g,1}} = (\overline{b}_{x_{g,1}}, \overline{b}_{y_{g,1}}); \\
\underline{B_{g,2}} &= (\underline{b}_{x_{g,2}}, \underline{b}_{y_{g,2}}), \overline{B_{g,2}} = (\overline{b}_{x_{g,2}}, \overline{b}_{y_{g,2}}); \\
\underline{B_{g,3}} &= (\underline{b}_{x_{g,3}}, \underline{b}_{y_{g,3}}), \overline{B_{g,3}} = (\overline{b}_{x_{g,3}}, \overline{b}_{y_{g,3}}).
\end{aligned} \tag{3.58}$$

So, in order to bound the intersection points inside of the rectangular boxes we have the following constraints for the g^{th} segment:

$$\begin{aligned}
\underline{b}_{x_{g,1}} \leq p_{x_{g,1}} \leq \overline{b}_{x_{g,1}}, \\
\underline{b}_{y_{g,1}} \leq p_{y_{g,1}} \leq \overline{b}_{y_{g,1}}, \\
\underline{b}_{x_{g,2}} \leq p_{x_{g,2}} \leq \overline{b}_{x_{g,2}}, \\
\underline{b}_{y_{g,2}} \leq p_{y_{g,2}} \leq \overline{b}_{y_{g,2}}, \\
\underline{b}_{x_{g,3}} \leq p_{x_{g,3}} \leq \overline{b}_{x_{g,3}}, \\
\underline{b}_{y_{g,3}} \leq p_{y_{g,3}} \leq \overline{b}_{y_{g,3}}.
\end{aligned} \tag{3.59}$$

Two adjacent segments have a common intersection point $\hat{P}_{g,g+1}$. Let $(\underline{\hat{B}_{g,g+1}}, \overline{\hat{B}_{g,g+1}})$ be the rectangular box corresponding to the common intersection point $\hat{P}_{g,g+1}$. So for two consecutive segments we have

$$\begin{aligned}(\underline{\hat{B}_{g,g+1}}, \overline{\hat{B}_{g,g+1}}) &= (\underline{B_{g,1}}, \overline{B_{g,1}}),\\ (\underline{\hat{B}_{g,g+1}}, \overline{\hat{B}_{g,g+1}}) &= (\underline{B_{g,3}}, \overline{B_{g,3}}).\end{aligned} \tag{3.60}$$

We define the rectangular box corner points $\underline{\hat{B}_{g,g+1}}$ and $\overline{\hat{B}_{g,g+1}}$ in Cartesian coordinate as follows:

$$\begin{aligned}\underline{\hat{B}_{g,g+1}} &= (\underline{\hat{b}}_{x_{g,g+1}}, \underline{\hat{b}}_{y_{g,g+1}}),\\ \overline{\hat{B}_{g,g+1}} &= (\overline{\hat{b}}_{x_{g,g+1}}, \overline{\hat{b}}_{y_{g,g+1}}).\end{aligned} \tag{3.61}$$

Since the starting and end points are fixed the constraints (3.59) for the g^{th} segment can be rewritten as follows:

$$\begin{aligned}\underline{b}_{x_{g,2}} \le p_{x_{g,2}} \le \overline{b}_{x_{g,2}} \quad &\forall g \in \{1, 2, ...m\},\\ \underline{b}_{y_{g,2}} \le p_{y_{g,2}} \le \overline{b}_{y_{g,2}} \quad &\forall g \in \{1, 2, ...m\},\\ \underline{\hat{b}}_{x_{g,g+1}} \le \hat{p}_{x_{g,g+1}} \le \overline{\hat{b}}_{x_{g,g+1}} \quad &\forall g \in \{1, 2, ...m-1\},\\ \underline{\hat{b}}_{y_{g,g+1}} \le \hat{p}_{y_{g,g+1}} \le \overline{\hat{b}}_{y_{g,g+1}} \quad &\forall g \in \{1, 2, ...m-1\}.\end{aligned} \tag{3.62}$$

The radius of curvature associated with each intersection point has a minimum value. If the radius of curvature is too small (or zero) then a horizontal alignment might get a sharp turn. Let $R_{\min}$ be the minimum radius of curvature. For the radius of curvatures $r_{g,1}$, $r_{g,2}$, and $r_{g,3}$ of the g^{th} segment, we have the following constraints:

$$\begin{aligned}r_{g,1} &\ge R_{\min},\\ r_{g,2} &\ge R_{\min},\\ r_{g,3} &\ge R_{\min}.\end{aligned} \tag{3.63}$$

For the starting and end points of the alignment, the radius of curvatures are zero (i.e. $r_{1,1} = 0$ and $r_{m,3} = 0$). Since for two adjacent segments $\hat{r}_{g,g+1} = r_{g,3} = r_{g+1,1}$, we can rewrite the radius of curvature constraints as follows:

$$\begin{aligned}\hat{r}_{g,g+1} \ge R_{\min} \quad &\forall g \in \{1, 2, ...m-1\},\\ r_{g,2} \ge R_{\min} \quad &\forall g \in \{1, 2, ...m\}.\end{aligned} \tag{3.64}$$

3.3 Model Summary

The summary of the optimization model considering all of the geometric specifications is stated as follows.

Objective function

$$f(X) = \begin{cases} \mathcal{C}_{\mathcal{VA}}(X) & \text{if } t^l_{g,j} \in [0,1] \ \ \forall g \in \mathcal{I}_{\mathcal{G}}, j \in \mathcal{I}_{\mathcal{SG}}, \\ \infty & \text{otherwise.} \end{cases} \tag{3.65}$$

Continuity constraints

$$\|P_{g,2} - P_{g,1}\| \geq \|P_{g,1} - F_{g,1}\| + \|P_{g,2} - E_{g,2}\| \quad \forall g \in \{1, 2, ...m\}, \tag{3.66}$$

$$\|P_{g,2} - P_{g,3}\| \geq \|P_{g,3} - E_{g,3}\| + \|P_{g,2} - F_{g,2}\| \quad \forall g \in \{1, 2, ...m\}, \tag{3.67}$$

IP bound constraints

$$\begin{aligned} \underline{b}_{x_{g,2}} \leq p_{x_{g,2}} \leq \overline{b}_{x_{g,2}} \quad & \forall g \in \{1, 2, ...m\}, \\ \underline{b}_{y_{g,2}} \leq p_{y_{g,2}} \leq \overline{b}_{y_{g,2}} \quad & \forall g \in \{1, 2, ...m\}, \\ \hat{\underline{b}}_{x_{g,g+1}} \leq \hat{p}_{x_{g,g+1}} \leq \hat{\overline{b}}_{x_{g,g+1}} \quad & \forall g \in \{1, 2, ...m-1\}, \\ \hat{\underline{b}}_{y_{g,g+1}} \leq \hat{p}_{y_{g,g+1}} \leq \hat{\overline{b}}_{y_{g,g+1}} \quad & \forall g \in \{1, 2, ...m-1\}, \end{aligned} \tag{3.68}$$

Minimum radius of curvature constraints

$$\begin{aligned} \hat{r}_{g,g+1} \geq R_{\min} \quad & \forall g \in \{1, 2, ...m-1\}, \\ r_{g,2} \geq R_{\min}. \quad & \forall g \in \{1, 2, ...m\}. \end{aligned} \tag{3.69}$$

Chapter 4

Numerical results

In this chapter, we present our experimental data and discuss the model performance to solve the real-world problems. We solve the test problem set using the two different derivative free optimization solvers: NOMAD [ALT09] and HOPSPACK [Pla09]. Finally, we compare the results of the two solvers.

4.1 Experimental setup

We performed numerical experiments on five different roads listed in Table 4.1. The road profile data are given by our industry partner Softree Technical System Inc. Note that Road D is a backtracking road and the other four roads i.e., Road A, Road B, Road C, and Road E are non-backtracking roads.

Table 4.1: Specifications of the test problems

Road Name	No. of stations	No. of IPs
Road A	73	8
Road B	361	5
Road C	258	14
Road D	118	22
Road E	74	9

All of the experiments were carried out in a Dell workstation with an Intel(R) Xenon(R) 2.40 GHz (2 cores) processor, 24 GB of RAM and a 64-bit Windows 7 Enterprise operating system. We used two derivative free optimization solvers, NOMAD [ALT09] (version 3.5, available at `http://www.gerad.ca/nomad`) and HOPSPACK [Pla09] (version 2.0.2, available at `http://www.sandia.gov/hopspack`) to solve the test problems. The optimization model was implemented in C++ using Microsoft Visual Studio 2010 Professional Edition.

4.2 The NOMAD and HOPSPACK solvers

The NOMAD and HOPSPACK solvers use two different derivative free optimization algorithms of the same category. Both of the solvers use the pattern search to find an optimal solution. The NOMAD and HOPSPACK solvers use the Mesh Adaptive Direct Search (MADS) algorithm [AD06] and the Asynchronous Parallel Pattern Search (APPS) algorithm [Kol05, GK06], respectively. Both algorithms convergence to a locally optimal point [AD06], [Kol05]. When the mesh size of the MADS algorithm goes to zero, it converges to a local minimum [AD06]. Similarly, if the step length of the APPS algorithm goes to zero, it converges to a local minimum [Kol05]. It is worth mentioning that both MADS and APPS algorithm are globally convergent to a locally optimum point and a solution (i.e., a minimizer) obtained by the algorithms depends on the starting point.

As a stopping condition of the algorithms, we use minimum mesh size and minimum step length for the MADS and APPS algorithms, respectively. Since we are interested in a solution close to a local minimum, we set both the minimum mesh size and minimum step length to 0.1. The input data of our model are given in meters. The final scaling (see [ALT09] and [Pla09]) of the variables of our model goes down below 10 cm, which means a local minimum should exist in less than 10 cm distance. However, we can set these parameter values of the algorithms to a value less than 0.1 for a better precision, but most of the agents (including our industry partner) only require a 10 cm precision in the optimal solution.

Both solvers, NOMAD and HOPSPACK need an initial starting point to run the algorithm. We used the baseline alignment of a corridor as an initial starting point for both solvers. However, the NOMAD solver gives a deterministic solution (i.e., different independent runs of the algorithm yield the same solution), while the HOPSPACK solver gives a non-deterministic solution (i.e., different independent runs of the algorithm might yield different solutions). So first, we solved the test problems using the NOMAD solver and then compare with the HOPSPACK solver solutions obtained by different independent runs.

4.3 Results for the test problems

Our model works well for both backtracking and non-backtracking alignments. Figure 4.1 and Figure 4.2 illustrate an initial alignment and an optimum alignment for a non-backtracking road and a backtracking road,

respectively.

Table 4.2 shows the cost improvement of the objective functions, the number of black-box evaluation and wall-clock time required to solve the test problems using the NOMAD solver.

Table 4.2: Cost improvement, no. of black-box evaluations and wall-clock time required to solve the test problems using the NOMAD solver.

Road Name	Initial alignment cost	Optimized alignment cost	Cost Improvement (%)	No. of Black-box evaluations	Wall-clock time (seconds)
Road A	1,897	1,361	28%	2,073	1,445
Road B	17,036	15,198	11%	2,528	1,770
Road C	87,829	69,621	21%	37,165	45,647
Road D	31,031	14,418	54%	90,535	47,613
Road E	8,054	6,498	19%	11,101	5,588

Now we compare the HOPSPACK solver results with the results obtained by the NOMAD solver. We solved each test problem five times independently using the HOPSPACK solver. Table 4.3 lists the optimum values of the objective functions obtained by five independent executions of the HOPSPACK solver for each test problem. The differences in the optimum objective function values are calculated with respect to the value obtained by the NOMAD solver. So in Table 4.3, a " + " value in the *Difference in the optimum costs* column indicates the HOPSPACK solver yields a better solution than the NOMAD solver and a " − " value indicates the opposite.

Combining the results obtained for the different roads listed in Table 4.3, we make an overall comparison between the two solvers. We observed that the HOPSPACK solver might yield a better or a worse solution than the solution obtained by the NOMAD solver. Thus, considering the tolerance of the difference in the optimum objective values obtained by two solver, we count the number of times a solver wins with respect to the other solver. Table 4.4 shows the comparison of the solvers for different tolerance values of the difference in the optimum objective values. The $x\%$ tolerance of the difference in optimum objective values means if the optimum objective values obtained by the two solvers are in between $-x\%$ to $+x\%$, then the solvers yield the same solution (i.e., the two solvers tie), otherwise a positive percentage value indicates the HOPSPACK solver wins and a negative

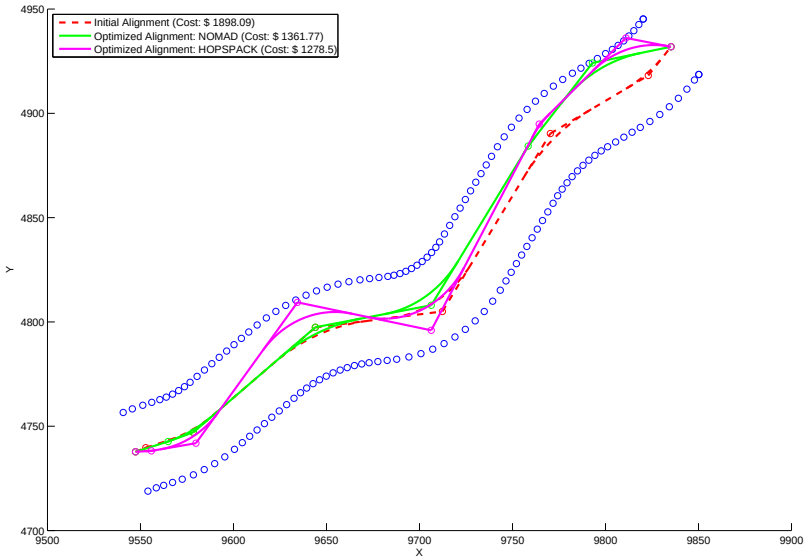

Figure 4.1: A non-backtracking alignment (the test problem associated with Road A) showing the initial alignment and the optimized alignments obtained by the NOMAD and HOPSPACK solvers.

percentage value indicates the NOMAD solver wins.

In Table 4.4, we see that if the tolerance of difference in the optimum objective value is ±3% or above, then the two solver tie for more than 50% test runs (i.e., more than 13 test runs among 25 test runs). We can also observe that for any case of the tolerance change in the optimum objective function value, the difference in the number of times the NOMAD solver wins and the number of times the HOPSPACK solver wins is at most 2. So in terms of optimum objective values obtained by the two solvers, the performance of both solvers are roughly equivalent.

We also recorded the number of black-box evaluations required for the HOPSPACK and NOMAD solvers to get the optimum solutions. We calculated the difference in number of black-box evaluations with respect to the number of black-box evaluations required by the NOMAD solver. So in Table 4.5, a " + " value the *Difference in no. of black-box evaluations* column indicates the HOPSPACK solver required less black-box evaluations than the NOMAD solver and a " − " value indicates the opposite.

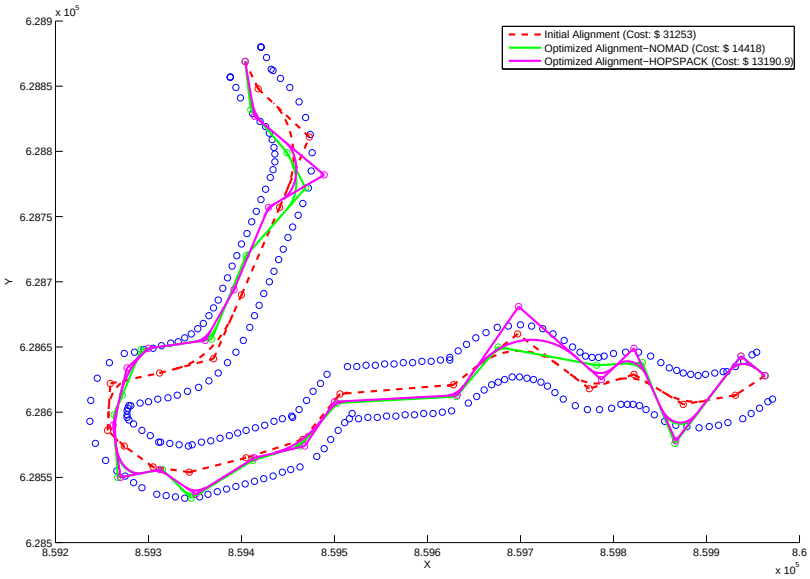

Figure 4.2: A backtracking alignment (the test problem associated with Road D) showing the initial alignment and the optimized alignments obtained by the NOMAD and HOPSPACK solvers.

In Table 4.5, we can see that for all of the 25 test runs, the HOPSPACK solver required less black-box evaluation than the NOMAD solver. For this particular problem set, on average, the HOPSPACKS solver took 78% less blackbox evaluation than the NOMAD solver. So the HOPSPACK solver is roughly five time faster than the NOMAD solver to compute the optimum solution.

Table 4.3: Comparison of optimum objective function values obtained by the HOPSPACK and NOMAD solvers to solve the test problems.

Road Name	Execution No.	Optimum cost function value-NOMAD	Optimum cost function value-HOPSPACK	Difference in the optimum costs (%)
Road A	Test run 1	1,361	1,291	+5.2%
	Test run 2	1,361	1,423	-4.6%
	Test run 3	1,361	1,418	-4.2%
	Test run 4	1,361	1,278	+6.1%
	Test run 5	1,361	1,486	-9.2%
Road B	Test run 1	15,198	15,510	-2.1%
	Test run 2	15,198	15,141	+0.4%
	Test run 3	15,198	15,128	+0.5%
	Test run 4	15,198	15,172	+0.2%
	Test run 5	15,198	15,529	-2.2%
Road C	Test run 1	69,621	70,161	-0.8%
	Test run 2	69,621	70,378	-1.1%
	Test run 3	69,621	69,995	-0.5%
	Test run 4	69,621	67,301	+3.3%
	Test run 5	69,621	67,045	+3.7%
Road D	Test run 1	14,418	13,190	+8.5%
	Test run 2	14,418	15,154	-7.6%
	Test run 3	14,418	14,155	+1.8%
	Test run 4	14,418	14,016	+2.8%
	Test run 5	14,418	15,384	-6.7%
Road E	Test run 1	6,497	6,524	-0.4%
	Test run 2	6,497	6,475	+0.3%
	Test run 3	6,497	6,497	0.0%
	Test run 4	6,497	6,502	-0.1%
	Test run 5	6,497	6,476	+0.3%

Table 4.4: Overall comparison of the HOPSPACK solver and the NOMAD solver with the optimum objective function values.

Tolerance of the difference in the optimum costs	No. of times the NOMAD solver wins	No. of times the HOPSPACK solver wins	No. of times the two solvers ties
± 1%	8	7	10
± 2%	7	6	12
± 3%	5	5	15
± 4%	5	3	17
± 5%	3	3	19
± 6%	3	2	20
± 7%	2	1	22
± 8%	1	1	23
± 9%	1	0	24
± 10%	0	0	25

Table 4.5: Comparison of the no. of black-box evaluations required for the HOPSPACK and NOMAD solvers to solve the test problems.

Road Name	Execution No.	No. of black-box evaluations-NOMAD	No. of Black-box evaluations-HOPSPACK	Difference in no. of black-box evaluations (%)
Road A	Test run 1	2,073	325	+84.3%
	Test run 2	2,073	336	+83.8%
	Test run 3	2,073	697	+66.4%
	Test run 4	2,073	486	+76.6%
	Test run 5	2,073	665	+67.9%
Road B	Test run 1	2,528	316	+87.5%
	Test run 2	2,528	309	+87.8%
	Test run 3	2,528	286	+88.8%
	Test run 4	2,528	392	+88.5%
	Test run 5	2,528	485	+80.8%
Road C	Test run 1	37,165	7,547	+79.7%
	Test run 2	37,165	31,418	+15.5%
	Test run 3	37,165	3,392	+90.9%
	Test run 4	37,165	3,213	+91.4%
	Test run 5	37,165	4,661	+87.5%
Road D	Test run 1	90,535	15,852	+82.5%
	Test run 2	90,535	19,997	+77.9%
	Test run 3	90,535	21,194	+76.6%
	Test run 4	90,535	17,816	+80.3%
	Test run 5	90,535	21,339	+76.4%
Road E	Test run 1	11,101	2,019	+81.8%
	Test run 2	11,101	4,332	+60.1%
	Test run 3	11,101	1,996	+82.0%
	Test run 4	11,101	2,501	+77.5%
	Test run 5	11,101	3,120	+71.9%

4.4 Summary of the result

Using our advanced model developed in Chapter 3, the NOMAD solver and the HOPSPACK solver can solve all of the test problems. In our advanced model, the number of variables in the optimization problem depends on the number of intersection points rather than the number of stations. For a small non-backtracking road of 73 stations with 8 intersection points, the NOMAD solver and the HOPSPACK solver required 2,073 and 325 function calls, respectively. In the meanwhile, for a backtracking road of 118 stations with 22 intersection points, the NOMAD solver and the HOPSPACK solver required 90535 and 15852 function calls, respectively, to solve the problem. So our advanced model can solve a reasonably large problem in a reasonable number of function calls.

Chapter 5

Conclusion

5.1 Contributions

Geometric specifications of a horizontal alignment is one of the most important considerations in the road design process. In our model, a horizontal alignment is represented using the geometric specifications which are used by engineers in practice. Thus the solution of the optimization model yields a practical horizontal alignment which satisfies geometric specifications and engineering requirements.

In this research, we pursued a new approach to address the horizontal alignment optimization problem. While most of the studies in the literature used heuristic based methods, we used derivative-free optimization approach. The rationale to use the derivative-free optimization approach is, it converges to a locally optimum solution. Thus our model always gives us a mathematically proven local optimum solution.

It is well known that the backward bends of a horizontal alignment (i.e., in the case of backtracking roads) might give rise to some difficulties in the optimization process [Nic73, page 123, Chapter 5], [JSJ06, page 21, Section 2.4.3], [Par77]. Our model effectively handles backward bends in a horizontal alignment. So our model can generate both backtracking and non-backtracking alignments.

Our optimization model is a bi-level optimization problem. Our model integrates the vertical alignment optimization problem and the horizontal alignment optimization problem together. In our model, the cost of a horizontal alignment is the cost of the optimized vertical alignment which corresponds to that specific horizontal alignment. So our model yields a solution which has not only a locally optimum horizontal alignment, but also the corresponding optimum vertical alignment.

5.2 Recommendations for future research

Although our proposed model works well for solving practical horizontal alignment optimization problems, it can be improved further for better precision and performance.

In our model formulation, we considered that the cross sections in a corridor are fixed, which are taken corresponding to the baseline alignment. However, cross sections should be always perpendicular to a horizontal alignment. When a horizontal alignment is significantly different from the baseline alignment, a set of new cross sections should be generated to increase the precision before calculating the corresponding vertical road profile.

A surrogate cost function is an approximation of the original cost function which is cheaper to compute. Two types of surrogates can be developed: adaptive and non-adaptive surrogates. The surrogate function of an original cost function might reduce the solution time required to get a solution. For the development of an adaptive surrogate cost function for our original cost function, we are motivated by the research work done in [BDF$^+$99]. They developed an adaptive surrogate for solving the helicopter rotor blade design problem. The framework for optimization of expensive functions proposed in [BDF$^+$99] can be applied to our problem effectively. We could also develop a non-adaptive surrogate cost function to accelerate the optimization process. The NOMAD solver can exploit the usage of a non-adaptive surrogate.

In our model, we only consider the construction costs to formulate our cost function. In the future, land acquisition costs could be incorporated by considering the unit cost of a piece of land corresponding to two consecutive cross sections in a corridor. We can also include pavement costs by considering the unit pavement cost between two consecutive stations.

We mentioned in Chapter 4 that both solvers need an initial starting point. Since solutions obtained by both solvers are locally optimum, we can use multiple starting points (i.e.; multiple initial alignments) to start the algorithms to obtain a better solution quickly. How to choose multiple alternative good alignments in a specified corridor can be a potential future research direction.

The HOPSPACK solver can use a parallel computing environment effectively. The resulting optimization model can be solved using the parallel version of the HOPSPACK solver to reduce the solution time.

During the optimization process, at each iteration, both of the derivative free optimization solvers solve a large number of vertical alignment optimization problems (i.e., a large scale mixed linear programming (MILP) problem). At the earlier stage of the optimization process (i.e., when the

mesh size is coarse) we can relax some of the parameters of the vertical alignment optimization problem to get an approximation cost to go forward and then at the later stage (i.e., when the mesh size becomes relatively small) we can again tight the parameters to get the accurate costs. This policy might reduce the solution time significantly. We can also use a warm start of the vertical alignment optimization problem when the horizontal alignments are close to each other to accelerate the vertical alignment optimization process. So the interconnection between the derivative free optimization solver and the MILP solver can be a potential way to reduce the solution time.

Bibliography

[AAS04] AASHTO. A policy on geometric design of highways and streets, 2004. American Association of State Highway and Transportation Officials. → pages 5

[AC73] G. C. Athanassoulis and V. Calogero. Optimal location of a new highway from A to B: a computer technique for route planning. In *PTRC Seminar Proceedings on Cost Models & Optimization in Highway*, 1973. → pages 9

[AD06] C. Audet and J. E. Dennis, Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006. → pages 16, 42

[Aka06] A. Akay. Minimizing total costs of forest roads with computeraided design model. *Sadhana*, 31(5):621–633, 2006. → pages 10

[ALT09] C. Audet, S. Le Digabel, and C. Tribes. NOMAD user guide. Technical Report G-2009-37, Les cahiers du GERAD, 2009. → pages 41, 42

[Aru05] K. Aruga. Tabu search optimization of horizontal and vertical alignments of forest roads. *Journal of Forest Research*, 10:275–284, 2005. → pages 10

[BDE10] BDE. Bureau of design and environment manual, 2010. Illinois Department of Transportation, USA. → pages 4

[BDF+99] A. J. Booker, Jr. Dennis, J. E., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural optimization*, 17(1):1–13, 1999. → pages 51

[Byr47] O. Byrne. *The First Six Books of the Elements of Euclid.* William Pickering, London,UK, 1847. → pages 23

[CGF89] E. P. Chew, C. J. Goh, and T. F. Fwa. Simultaneous optimization of horizontal and vertical alignments for highways. *Transportation Research Part B: Methodological*, 23(5):315–329, 1989. → pages 3, 10, 11

[CL06] J. Cheng and Y. Lee. Model for three-dimensional highway alignment. *Journal of Transportation Engineering*, 132(12):913–920, 2006. → pages 11

[DAAMP06] J. Dunlavy, A. Akuoko-Asibey, R. Masse, and D. Pilon. An analysis of the transportation industry in 2005. Technical report, Statistics Canada, 2006. → pages 1

[Eas88] S. M. Easa. Selection of roadway grades that minimize earthwork cost using linear programming. *Transportation Research Part A: General*, 22(2):121 – 136, 1988. → pages 8

[Fwa89] T. F. Fwa. Highway vertical alignment analysis by dynamic programming. *Transportation Research Record*, 1239:1–9, 1989. → pages 7

[GCF88] C. J. Goh, E. P. Chew, and T. F. Fwa. Discrete and continuous models for computation of optimal vertical highway alignment. *Transportation Research Part B: Methodological*, 22(6):399–409, 1988. → pages 7

[GK06] G. A. Gray and T. G. Kolda. Algorithm 856: APPSPACK 4.0: Asynchronous parallel pattern search for derivative-free optimization. *ACM Transactions on Mathematical Software*, 32(3):485–507, September 2006. → pages 42

[GLA05] A. B. Goktepe, A. H. Lav, and S. Altun. Dynamic optimization algorithm for vertical alignment of highways. *Mathematical and Computational Applications*, 10:341–350, 2005. → pages 7

[Hay70] R. W. Hayman. Optimization of vertical alignment for highways through mathematical programming. *Highway Research Record*, pages 1–9, 1970. → pages 7

[HBS68] B. E. Howard, Z. Bramnick, and J. F. Shaw. Optimum curvature principle in highway routing. In *Proceeding of the American Society of Civil Engineers*, volume 94, pages 61–82, 1968. → pages 9

[HHLM11] D. Hare, W. Hare, Y. Lucet, and S. Mehrotra. An arc-flow model for earthwork operations. Technical report, (undisclosed), 2011. → pages 8

[HHLR14] W. Hare, S. Hossain, Y. Lucet, and F. Rahman. Models and strategies for efficiently determining an optimal vertical alignment of roads. *Computers & Operations Research*, 44:161 – 173, 2014. → pages 8, 16, 27

[HKL11] W. L. Hare, V. R. Koch, and Y. Lucet. Models and algorithms to improve earthwork operations in road design using mixed integer linear programming. *European Journal of Operational Research*, 215(2):470 – 480, 2011. → pages 8

[Hog73] J. D. Hogan. Exprience with OPTLOC - optimum location of highways by computer. In *PTRC Seminer Proceddings on Cost Models and Optimisation in Highways*, 1973. → pages 10

[Hua73] J. P. Huarte. Opygar: Optimization and automatic design of highway profiles. In *PTRC Seminar Proceedings on Cost Models and Optimization in Highways*, 1973. → pages 7

[Jha03] M. K. Jha. Criteria-based decision support system for selecting highway alignments. *Journal of Transportation Engineering*, 129(1):33–41, 2003. → pages 11

[JJS00] J.-C. Jong, M. K. Jha, and P. Schonfeld. Preliminary highway design with genetic algorithms and geographic information system. *Computer Aided Civil Infrastructure Engineering*, 15(4):261–271, 2000. → pages 10

[JK06] M. K. Jha and E. Kim. Highway route optimization based on accessibility, proximity, and land-use changes. *Journal of Transportation Engineering*, 132:435–439, 2006. → pages 11

[Jon98] J.-C. Jong. *Optimizing highway alignments with genetic algorithms*. PhD thesis, University of Maryland, College Park, 1998. → pages 10

[JS03] J.-C. Jong and P. Schonfeld. An evolutionary model for simultaneously optimizing 3d highway alignments. *Tranportation Research, Part B: Methodology*, 37B(2):107–128, 2003. → pages 11

[JSJ06] M. K. Jha, P. Schonfeld, and J.-C. Jong. *Intelligent Road Design*, volume 19. WIT Press, 2006. → pages 2, 3, 7, 8, 9, 10, 50

[KL10] V. R. Koch and Y. Lucet. A note on: Spline technique for modeling roadway profile to minimize earthwork cost. *Journal of Industrial and Management Optimization*, 6(2):393 – 400, 2010. → pages 8

[Kol05] T. G. Kolda. Revisiting asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal on Optimization*, 16(2):563–586, 2005. → pages 42

[LD11] S. Le Digabel. Algorithm 909: Nomad: Nonlinear optimization with the mads algorithm. *ACM Transactions on Mathematical Software*, 37(4):44:1–44:15, 2011. → pages 16, 19

[LTL09] Y. Lee, Y. Tsou, and H. Liu. Optimization method for highway horizontal alignment design. *Journal of Transportation Engineering*, 135(4):217–224, 2009. → pages 10

[MA04] A. A. Moreb and M. S. Aljohani. Quadratic representation for roadway profile that minimizes earthwork cost. *Journal of Systems Science and Systems Engineering*, 13(2):245–252, 2004. → pages 8

[Mor96] A. A. Moreb. Linear programming model for finding optimal roadway grades that minimize earthwork cost. *European Journal of Operational Research*, 93(1):148 – 154, 1996. → pages 8

[Mor09] A. A. Moreb. Spline technique for modeling roadway profile to minimize earthwork cost. *Journal of Industrial and Management Optimization (JIMO)*, 5(2):275–283, 2009. → pages 8

[MS81] R. H. Mayer and R. M. Stark. Earthmoving logistics. *Journal of the Construction Division*, 107(2):297–312, 1981. → pages 7

[Mur73] J. D. Murchland. Methods of vertical profile optimisation for an improvement to an existing road. In *PTRC Seminar Proceedings on Cost Models and Optimisation in Highways*, 1973. → pages 7

[NEW76] A. J. Nicholson, D.G. Elms, and A. Williman. A variational approach to optimal route location. *Highway Engineers*, 23:22–25, 1976. → pages 10

[Nic73] A. J. Nicholson. *A variational approch to optimal route location*. PhD thesis, University of Canterbury, 1973. → pages 10, 50

[OEC73] OECD. Optimisation of road alignment by the use of computers, 1973. Organisation for Economic Co-operation and Development (Road Research Group), Paris. → pages 3, 10

[Par77] N. A. Parker. Rural highway route corridor selection. *Transportation Planning and Technology*, 3(4):247–256, 1977. → pages 9, 10, 50

[Pea73] A. D. Pearman. Investigation of the objective function in problems of optimising highway vertical alignment. In *PTRC Seminar Proceedings Cost Models and Optimisation in Highways*, 1973. → pages 7

[Pla09] T. D. Plantenga. HOPSPACK 2.0 user manual. Technical Report SAND2009-6265, Sandia National Laboratories, Albuqerque, NM and Livermore, CA, October 2009. → pages 41, 42

[Rah12] F. Rahman. Optimizing the vertical alignment under earthwork block removal constraints in road construction. Master's thesis, The University of British Columbia, 2012. → pages 7, 8

[Rob73] R. Robinson. Automatic design of the road vertical alignment. In *PTRC Seminar Proceedings on Cost Models and Optimization in Highways*, 1973. → pages 7

[RS13] L. M. Rios and N. V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013. → pages 16

[SH81] J. F. B. Shaw and B. E. Howard. Comparison of two integration methods in transportation routing. *Transportation Research Record*, pages 8–3, 1981. → pages 9

[SH82] J. F. B. Shaw and B. E. Howard. Expressway route optimization by OCP. *Transportation Engineering Journal*, 108:227–243, 1982. → pages 9

[TM71] A. K. Turner and R. D. Miles. The gcars system: A computer assisted method of regional route location. *Highway Research Record*, pages 1–15, 1971. → pages 9

[Tri87a] D. Trietsch. Comprehensive design of highway networks. *Tranportation Science*, 21(1):1, 1987. → pages 9

[Tri87b] D. Trietsch. A family of methods for preliminary highway alignment. *Transportation Science*, 21(1):17–25, 1987. → pages 9

[TT03] C. W. Tat and F. Tao. Using GIS and genetic algorithm in highway alignment optimization. In *Intelligent Transportation Systems*, volume 2, pages 1563–1567, 2003. → pages 10

[Tur78] A. K. Turner. A decade of experience in computer aided route selection. *Photogrammetric Engineering and Remote Sensing*, pages 1561–1576, 1978. → pages 9

[Wan95] F. Y. M. Wan. *Introduction to the calculus of variations and its applications*. Chapman & Hal, 1995. → pages 8

Appendix

Appendix A

Tables

A.1 Results for basic model

The numerical experiments were performed on the five different roads. Tables A.1, A.2, A.3, A.4, and A.5 list the computational data for five different input files (i.e., different roads). " * " indicates the problem cannot be solved in 3 hours.

Table A.1: Computational Experience (Hart Rd Small.csv)

# of station	# of Fun. call	# of itr.	Wall-clock Time (in Min)	Memory (in Mb)
5	509	53	1.8	0.10
10	1,256	70	5.6	0.35
15	1,998	81	11.9	0.73
20	3,517	1,119	21.0	1.56
25	4,734	131	29.7	2.48
30	6,702	175	53.4	4.09
35	8,011	173	57.3	5.56
40	11,213	211	91.7	8.57
45	16,413	285	137.9	13.86
50	16,182	257	149.8	14.78

Table A.2: Computational Experience (Diamond Road align-1.csv)

# of station	# of Fun. call	# of itr.	Wall-clock Time (in Min)	Memory (in Mb)
5	600	73	2.5	0.11
10	1,318	81	6.4	0.37
15	2,169	89	12.4	0.79
20	3,119	105	20.4	1.39
25	4,286	111	34.5	2.24
30	13,391	307	111.5	8.29
35	7,329	147	63.3	5.02
40	*	*	*	*
45	*	*	*	*
50	*	*	*	*

Table A.3: Computational Experience (Diamond Road align-2.csv)

# of station	# of Fun. call	# of itr.	Wall-clock Time (in Min)	Memory (in Mb)
5	2,810	359	13.8	0.57
10	1,787	107	7.7	0.50
15	2,715	117	16.0	0.98
20	3,690	115	28.0	1.60
25	5,424	147	42.6	2.87
30	20,032	447	177.3	12.60
35	*	*	*	*
40	10,146	100.7	6,047	7.70
45	*	*	*	*
50	*	*	*	*

Table A.4: Computational Experience (bluff road.csv)

# of station	# of Fun. call	# of itr.	Wall-clock Time (in Min)	Memory (in Mb)
5	718	359	3.4	0.14
10	10,980	107	66.2	3.17
15	*	*	*	*
20	6,207	207	38.9	2.74
25	9,281	275	66.6	4.87
30	17,141	407	146.3	10.60
35	*	*	*	*
40	*	*	*	*
45	*	*	*	*
50	*	*	*	*

Table A.5: Computational Experience (spur 3 demo.csv)

# of station	# of Fun. call	# of itr.	Wall-clock Time (in Min)	Memory (in Mb)
5	728	83	3.4	0.14
10	1,727	101	8.2	0.48
15	3,099	127	19.3	1.12
20	19,268	627	144.1	8.79
25	6,987	189	50.0	3.61
30	10,609	245	85.5	6.41
35	13,625	285	121.21	9.34
40	*	*	*	*
45	*	*	*	*
50	*	*	*	*

A.2 Optimized alignments of the test problems

The minimizer of the optimization problems (developed using the model described in Chapter 3) are listed in Tables A.6, A.7, A.8, A.9, and A.10.

Table A.6: The optimal alignments of the Road A obtained by the NOMAD and HOPSPACK solvers.

Spec.	Initial alignment	Opt. align.-NOMAD	Opt. align.-HOPSPACK (run 1)	Opt. align.-HOPSPACK (run 2)	Opt. align.-HOPSPACK (run 3)	Opt. align.-HOPSPACK (run 4)	Opt. align.-HOPSPACK (run 5)
IP1-x	9,747.36	9,747.36	9,747.36	9,747.36	9,747.36	9,747.36	9,747.36
IP1-y	4,737.74	4,737.74	4,737.74	4,737.74	4,737.74	4,737.74	4,737.74
IP1-r	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IP2-x	9,552.88	9,564.89	9,558.89	9,557.24	9,562.79	9,555.89	9,557.09
IP2-y	4,739.69	4,742.69	4,738.19	4,739.69	4,739.69	4,738.19	4,738.49
IP2-r	50.00	50.00	62.50	50.00	150.00	62.50	55.00
IP3-x	9,578.33	9,578.42	9,587.33	9,578.33	9,584.33	9,579.83	9,583.13
IP3-y	4,747.11	4,747.11	4,747.11	4,744.11	4,741.86	4,741.86	4,745.91
IP3-r	75.00	75.00	75.00	50.00	50.00	50.00	87.50
IP4-x	9,644.02	9,644.03	9,650.03	9,644.03	9,638.63	9,634.43	9,647.03
IP4-y	4,797.37	4,797.47	4,803.38	4,803.38	4,803.38	4,809.38	4,806.23
IP4-r	100.00	100.00	50.00	50.00	50.00	50.00	62.50
IP5-x	9,712.34	9,706.34	9,712.34	9,712.34	9,706.34	9,706.34	9,703.34
IP5-y	4,804.99	4,808.00	4,805.00	4,803.38	4,802.00	4,796.00	4,805.60
IP5-r	60.00	60.00	50.00	50.00	50.00	50.00	50.00
IP6-x	9,770.36	9,758.46	9,764.37	9,764.37	9,770.37	9,764.37	9,764.37
IP6-y	4,890.32	4,884.32	4,905.32	4,905.32	4,908.92	4,894.82	4,899.32
IP6-r	100.00	100.00	100.00	50.00	50.00	50.00	50.00
IP7-x	9,822.99	9,793.00	9,823.00	9,820.00	9,811.00	9,811.00	9,821.20
IP7-y	4,918.13	4,924.13	4,930.13	4,915.13	4,918.13	4,936.13	4,915.13
IP7-r	75.00	75.00	50.00	50.00	135.00	50.00	50.00
IP8-x	9,835.23	9,835.23	9,835.23	9,835.23	9,835.23	9,835.23	9,835.23
IP8-y	4,931.89	4,931.89	4,931.89	4,931.89	4,931.89	4,931.89	4,931.89
IP8-r	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.7: The optimal alignments of the Road B obtained by the NOMAD and HOPSPACK solvers.

Spec.	Initial alignment	Opt. align.- NOMAD	Opt. align.- HOPSPACK (run 1)	Opt. align.- HOPSPACK (run 2)	Opt. align.- HOPSPACK (run 3)	Opt. align.- HOPSPACK (run 4)	Opt. align.- HOPSPACK (run 5)
IP1-x	392,201.60	392,201.60	392,201.60	392,201.60	392,201.60	392,201.60	392,201.60
IP1-y	6,983,091.87	6,983,091.87	6,983,091.87	6,983,091.87	6,983,091.87	6,983,091.87	6,983,091.87
IP1-r	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IP2-x	392,282.65	392,314.00	392,313.00	392,325.00	392,313.00	392,283.00	392,313.00
IP2-y	6,983,568.99	6,983,620.00	6,983,680.00	6,983,630.00	6,983,610.00	6,983,530.00	6,983,680.00
IP2-r	200.00	4,751.10	3,800.00	2,800.00	4,650.00	3,200.00	3,800.00
IP3-x	392,923.43	392,891.00	392,923.00	392,890.00	392,893.00	392,893.00	392,923.00
IP3-y	6,984,840.69	6,984,860.00	6,984,840.00	6,984,860.00	6,984,860.00	6,984,870.00	6,984,840.00
IP3-r	2,000.00	583.74	1,700.00	900.00	1,200.00	500.00	2,500.00
IP4-x	393,378.71	393,370.00	393,409.00	393,409.00	393,409.00	393,379.00	393,379.00
IP4-y	6,985,977.54	6,985,840.00	6,986,010.00	6,985,960.00	6,985,960.00	6,985,870.00	6,985,940.00
IP4-r	2,000.00	4,967.77	4,500.00	3,600.00	3,500.00	4,650.00	4,450.00
IP5-x	393,495.89	393,495.89	393,495.89	393,495.89	393,495.89	393,495.89	393,495.89
IP5-y	6,986,433.99	6,986,433.99	6,986,433.99	6,986,433.99	6,986,433.99	6,986,433.99	6,986,433.99
IP5-r	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.8: The optimal alignments of the Road C obtained by the NOMAD and HOPSPACK solvers.

Spec.	Initial alignment	Opt. align.-NOMAD	Opt. align.-HOPSPACK (run 1)	Opt. align.-HOPSPACK (run 2)	Opt. align.-HOPSPACK (run 3)	Opt. align.-HOPSPACK (run 4)	Opt. align.-HOPSPACK (run 5)
IP1-x	405,181.86	405,181.86	405,181.86	405,181.86	405,181.86	405,181.86	405,181.86
IP1-y	7,016,733.12	7,016,733.12	7,016,733.12	7,016,733.12	7,016,733.12	7,016,733.12	7,016,733.12
IP1-r	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IP2-x	405,446.69	405,494.00	405,484.00	405,466.00	405,447.00	405,454.00	405,472.00
IP2-y	7,016,896.66	7,016,930.00	7,016,930.00	7,016,910.00	7,016,90.00	7,016,900.00	7,016,920.00
IP2-r	500.00	200.00	200.00	200.00	200.00	200.00	200.00
IP3-x	405,721.39	405,716.00	405,723.00	405,751.00	405,706.00	405,706.00	405,714.00
IP3-y	7,017,215.30	7,017,190.00	7,017,200.00	7,017,250.00	7,017,200.00	7,017,160.00	7,017,170.00
IP3-r	500.00	200.00	6,680.00	24,130.00	11,600.00	3,355.00	3,895.00
IP4-x	405,941.14	405,939.00	405,977.00	405,949.00	405,956.00	405,971.00	405,956.00
IP4-y	7,017,573.02	7,017,510.00	7,017,540.00	7,017,480.00	7,017,480.00	7,017,550.00	7,017,530.00
IP4-r	500.00	200.00	200.00	1,915.00	1,840.00	200.00	700.00
IP5-x	406,182.87	406,194.00	406,153.00	406,147.00	406,150.00	406,153.00	406,153.00
IP5-y	7,017,925.85	7,017,890.00	7,017,910.00	7,017,890.00	7,017,910.00	7,017,900.00	7,017,900.00
IP5-r	500.00	200.00	480.00	380.00	455.00	450.00	450.00
IP6-x	406,363.56	406,354.00	406,358.00	406,361.00	406,358.00	406,364.00	406,364.00
IP6-y	7,018,063.80	7,018,020.00	7,018,020.00	7,018,030.00	7,018,020.00	7,018,030.00	7,018,030.00
IP6-r	500.00	200.00	200.00	200.00	200.00	200.00	200.00
IP7-x	406,498.87	406,470.00	406,466.00	406,469.00	406,469.00	406,466.00	406,469.00
IP7-y	7,018,292.85	7,018,310.00	7,018,310.00	7,018,320.00	7,018,320.00	7,018,310.00	7,018,320.00
IP7-r	200.00	200.00	200.00	200.00	200.00	200.00	200.00
IP8-x	406,356.24	406,362.00	406,359.00	406,359.00	406,359.00	406,359.00	406,362.00
IP8-y	7,018,576.57	7,018,550.00	7,018,550.00	7,018,550.00	7,018,550.00	7,018,550.00	7,018,540.00
IP8-r	500.00	200.00	200.00	200.00	200.00	200.00	200.00
IP9-x	406,282.98	406,325.00	406,328.00	406,326.00	406,328.00	406,326.00	406,326.00
IP9-y	7,018,963.58	7,018,970.00	7,018,970.00	7,018,970.00	7,018,970.00	7,018,980.00	7,018,970.00
IP9-r	200.00	200.00	200.00	200.00	200.00	200.00	200.00
IP10-x	406,390.42	406,383.00	406,383.00	406,387.00	406,383.00	406,360.00	406,371.00
IP10-y	7,019,157.70	7,019,170.00	7,019,170.00	7,019,180.00	7,019,160.00	7,019,070.00	7,019,120.00
IP10-r	500.00	200.00	250.00	240.00	200.00	320.00	200.00
IP11-x	406,407.51	406,351.00	406,390.00	406,379.00	406,348.00	406,376.00	406,364.00
IP11-y	7,019,372.57	7,019,390.00	7,019,370.00	7,019,290.00	7,019,380.00	7,019,460.00	7,019,450.00
IP11-r	500.00	200.00	780.00	310.00	400.00	260.00	330.00
IP12-x	406,639.48	406,700.00	406,687.00	406,678.00	406,669.00	406,719.00	406,719.00
IP12-y	7,020,074.57	7,020,020.00	7,020,090.00	7,020,100.00	7,020,140.00	7,019,940.00	7,019,920.00
IP12-r	500.00	500.00	200.00	200.00	200.00	1,100.00	1,075.00
IP13-x	406,581.99	406,582.00	406,582.00	406,582.00	406,582.00	406,582.00	406,582.00
IP13-y	7,020,629.05	7,020,620.00	7,020,620.00	7,020,620.00	7,020,620.00	7,020,630.00	7,020,620.00
IP13-r	200.00	200.00	200.00	200.00	200.00	200.00	200.00
IP14-x	406,997.27	406,997.27	406,997.27	406,997.27	406,997.27	406,997.27	406,997.27
IP14-y	7,021,135.64	7,021,135.64	7,021,135.64	7,021,135.64	7,021,135.64	7,021,135.64	7,021,135.64
IP14-r	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.9: The optimal alignments of the Road D obtained by the NOMAD and HOPSPACK solvers.

Spec.	Initial alignment	Opt. align.-NOMAD	Opt. align.-HOPSPACK (run 1)	Opt. align.-HOPSPACK (run 2)	Opt. align.-HOPSPACK (run 3)	Opt. align.-HOPSPACK (run 4)	Opt. align.-HOPSPACK (run 5)
IP1-x	859,962.73	859,962.73	859,962.73	405,181.86	405,181.86	405,181.86	405,181.86
IP1-y	628,627.91	628,627.91	628,627.91	7,016,733.12	7,016,733.12	7,016,733.12	7,016,733.12
IP1-r	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IP2-x	859,930.76	859,937.00	859,937.00	859,961.00	859,918.00	859,927.00	859,961.00
IP2-y	628,612.74	628,643.00	628,643.00	628,605.00	628,633.00	628,637.00	628,604.00
IP2-r	25.00	25.00	25.00	25.00	25.00	25.00	25.00
IP3-x	859,874.81	859,866.00	859,867.00	859,858.00	859,872.00	859,868.00	859,849.00
IP3-y	628,605.96	628,576.00	628,578.00	628,599.00	628,576.00	628,576.00	628,601.00
IP3-r	90.00	25.00	25.00	25.00	25.00	25.00	45.00
IP4-x	859,822.25	859,831.00	859,822.00	859,820.00	859,827.00	859,826.00	859,822.00
IP4-y	628,629.40	628,638.00	628,649.00	628,641.00	628,637.00	628,641.00	628,642.00
IP4-r	60.00	25.00	25.00	25.00	25.00	25.00	25.00
IP5-x	859,773.55	859,782.00	859,787.00	859,789.00	859,790.00	859,780.00	859,788.00
IP5-y	628,618.10	628,636.00	628,625.00	628,624.00	628,630.00	628,633.00	628,625.00
IP5-r	60.00	60.00	25.00	25.00	48.00	25.00	25.00
IP6-x	859,696.78	859,676.00	859,698.00	859,702.00	859,699.00	859,674.00	859,701.00
IP6-y	628,659.71	628,650.00	628,681.00	628,690.00	628,681.00	628,651.00	628,690.00
IP6-r	90.00	25.00	92.50	99.00	115.00	25.00	101.50
IP7-x	859,627.65	859,631.00	859,632.00	859,629.00	859,629.00	859,629.00	859,629.00
IP7-y	628,620.61	628,612.00	628,613.00	628,614.00	628,612.00	628,612.00	628,613.00
IP7-r	25.00	25.00	25.00	25.00	25.00	25.00	25.00
IP8-x	859,505.79	859,503.00	859,500.00	859,499.00	859,500.00	859,499.00	859,499.00
IP8-y	628,614.11	628,607.00	628,608.00	628,609.00	628,608.00	628,608.00	628,609.00
IP8-r	25.00	25.00	25.00	25.00	25.00	25.00	25.00
IP9-x	859,466.24	859,463.00	859,468.00	859,470.00	859,468.00	859,470.00	859,471.00
IP9-y	628579.36	628,574.00	628,574.00	628,576.00	628,573.00	628,575.00	628,577.00
IP9-r	25.00	25.00	96.25	93.75	97.50	90.00	90.00
IP10-x	859,404.57	859,412.00	859,414.00	859,412.00	859,414.00	859,414.00	859,412.00
IP10-y	628,565.11	628,563.00	628,565.00	628,565.00	628,565.00	628,565.00	628,565.00
IP10-r	25.00	25.00	25.00	25.00	27.50	27.50	25.00
IP11-x	859,343.95	859,346.00	859,351.00	859,360.00	859,347.00	859,348.00	859,359.00
IP11-y	628553.75	628,534.00	628,537.00	628,543.00	628,535.00	628,535.00	628,543.00
IP11-r	25.00	25.00	25.00	25.00	25.00	25.00	25.00
IP12-x	859,305.28	859,315.00	859,312.00	859,277.00	859,316.00	859,313.00	859,281.00
IP12-y	628,557.69	628,556.00	628,556.00	628,560.00	628,556.00	628,558.00	628,559.00
IP12-r	25.00	25.00	25.00	25.00	25.00	28.75	25.00
IP13-x	859,273.63	859,267.00	859,270.00	859,265.00	859,269.00	859,268.00	859,266.00
IP13-y	628573.70	628,550.00	628,550.00	628,571.00	628,551.00	628,548.00	628,566.00
IP13-r	25.00	25.00	25.00	25.00	25.00	25.00	25.00
IP14-x	859,255.50	859,263.00	859,262.00	859,262.00	859,262.00	859,263.00	859,263.00
IP14-y	628,585.89	628,598.00	628,590.00	628,601.00	628,595.00	628,590.00	628,587.00
IP14-r	25.00	25.00	55.00	25.00	38.75	25.00	42.50

continue to next page

continue from previous page

Spec.	Initial alignment	Opt. align.-NOMAD	Opt. align.-HOPSPACK (run 1)	Opt. align.-HOPSPACK (run 2)	Opt. align.-HOPSPACK (run 3)	Opt. align.-HOPSPACK (run 4)	Opt. align.-HOPSPACK (run 5)
IP15-x	859,258.87	859272.00	859,277.00	859,271.00	859,271.00	859,271.00	859,266.00
IP15-y	628,622.27	628,613.00	628,634.00	628,612.00	628,610.00	628,624.00	628,604.00
IP15-r	25.00	25.00	62.50	68.75	25.00	25.00	32.50
IP16-x	859,312.40	859,292.00	859,300.00	859,294.00	859,289.00	859,294.00	859,295.00
IP16-y	628,630.18	628,648.00	628,649.00	628,650.00	628,648.00	628,648.00	628,650.00
IP16-r	25.00	25.00	25.00	37.50	37.50	38.75	35.00
IP17-x	859,370.11	859,368.00	859,361.00	859,358.00	859,362.00	859,371.00	859,361.00
IP17-y	628,640.65	628,656.00	628,655.00	628,653.00	628,655.00	628,657.00	628,653.00
IP17-r	90.00	25.00	25.00	25.00	25.00	25.00	25.00
IP18-x	859,400.45	859,405.00	859,392.00	859,391.00	859,393.00	859,397.00	859,396.00
IP18-y	628,690.04	628,720.00	628,694.00	628,695.00	628,699.00	628,720.00	628,697.00
IP18-r	25.00	25.00	56.25	36.25	25.00	203.12	561.25
IP19-x	859,440.98	859,469.00	859,429.00	859,430.00	859,425.00	859,452.00	859,429.00
IP19-y	628,757.15	628,772.00	628,757.00	628,759.00	628,758.00	628,763.00	628,756.00
IP19-r	25.00	25.00	25.00	25.00	25.00	25.00	25.00
IP20-x	859,472.96	859,449.00	859,489.00	859,496.00	859,461.00	859,473.00	859,488.00
IP20-y	628,811.20	628,799.00	628,782.00	628,781.00	628,800.00	628,807.00	628,781.00
IP20-r	50.00	25.00	25.00	25.00	50.00	32.50	25.00
IP21-x	859,418.43	859,410.00	859,414.00	859,404.00	859,412.00	859,415.00	859,415.00
IP21-y	628,848.14	628,832.00	628,827.00	628,830.00	628,837.00	628,822.00	628,826.00
IP21-r	25.00	25.00	25.00	63.75	25.00	25.00	25.00
IP22-x	859,404.20	859,404.20	859,404.20	859,404.20	859,404.20	859,404.20	859,404.20
IP22-y	628,868.81	628,868.81	628,868.81	628,868.81	628,868.81	628,868.81	628,868.81
IP22-r	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.10: The optimal alignments of the Road E obtained by the NOMAD and HOPSPACK solvers.

Spec.	Initial alignment	Opt. align.-NOMAD	Opt. align.-HOPSPACK (run 1)	Opt. align.-HOPSPACK (run 2)	Opt. align.-HOPSPACK (run 3)	Opt. align.-HOPSPACK (run 4)	Opt. align.-HOPSPACK (run 5)
IP1-x	860,254.98	859,962.73	859,962.73	859,962.73	859,962.73	859,962.73	859,962.73
IP1-y	628,612.83	628,627.91	628,627.91	628,627.91	628,627.91	628,627.91	628,627.91
IP1-r	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IP2-x	860,219.54	860,207.00	860,202.00	860,203.00	860,202.00	860,202.00	860,202.00
IP2-y	628,747.01	628,777.00	628,788.00	628,788.00	628,788.00	628,788.00	628,788.00
IP2-r	50.00	50.00	50.00	50.00	50.00	50.00	50.00
IP3-x	860,179.37	860,179.00	859,866.00	860,179.00	860,179.00	860,179.00	860,179.00
IP3-y	628,809.75	628,817.00	628,811.00	628,813.00	628,812.00	628,811.00	628,812.00
IP3-r	50.00	50.00	50.00	50.00	50.00	50.00	50.00
IP4-x	860177.93	860,180.00	860,178.00	860,178.00	860,178.00	860,178.00	860,178.00
IP4-y	628,861.00	628,856.00	628,849.00	628,858.00	628,854.00	628,849.00	628,865.00
IP4-r	50.00	50.00	117.50	170.00	100.00	102.50	110.0
IP5-x	860,181.57	860,190.00	860,182.00	860,188.00	860,188.00	860,182.00	860,188.00
IP5-y	628,896.38	628,897.00	628,878.00	628,893.00	628,896.00	628,878.00	628,892.00
IP5-r	50.00	50.00	220.00	112.50	50.00	235.00	122.50
IP6-x	860199.29	860,204.00	860,199.00	860,205.00	860,199.00	860,199.00	860,205.00
IP6-y	628,959.03	628,967.00	628,935.00	628,971.00	628,935.00	628,935.00	628,973.00
IP6-r	50.00	50.00	330.00	198.75	320.00	312.50	205.0
IP7-x	860,181.57	860,203.00	860,206.00	860,202.00	860206.00	860206.00	860203.00
IP7-y	629059.67	629,050.00	629,030.00	629,028.00	629,036.00	629,039.00	629,029.00
IP7-r	50.00	50.00	665.00	237.50	412.50	340.0	247.50
IP8-x	860,225.87	860,230.00	860,226.00	860,227.00	860,227.00	860,227.00	860,227.00
IP8-y	629,164.10	629,159.00	629,164.00	629,163.00	629,163.00	629,163.00	629,163.00
IP8-r	50.00	50.00	50.00	50.00	50.00	50.00	50.00
IP9-x	860,146.12	860,146.12	860,146.12	860,146.12	860,146.12	860,146.12	860,146.12
IP9-y	629,280.56	629,280.56	629,280.56	629,280.56	629,280.56	629,280.56	629,280.56
IP9-r	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Appendix B

Figures

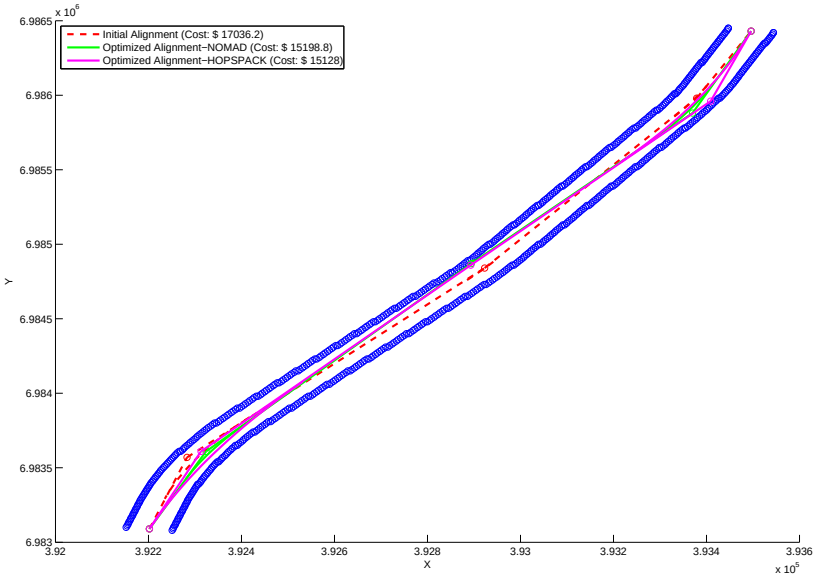

Figure B.1: Optimum alignments of the Road B obtained by the NOMAD solver and the HOPSPACK solver.

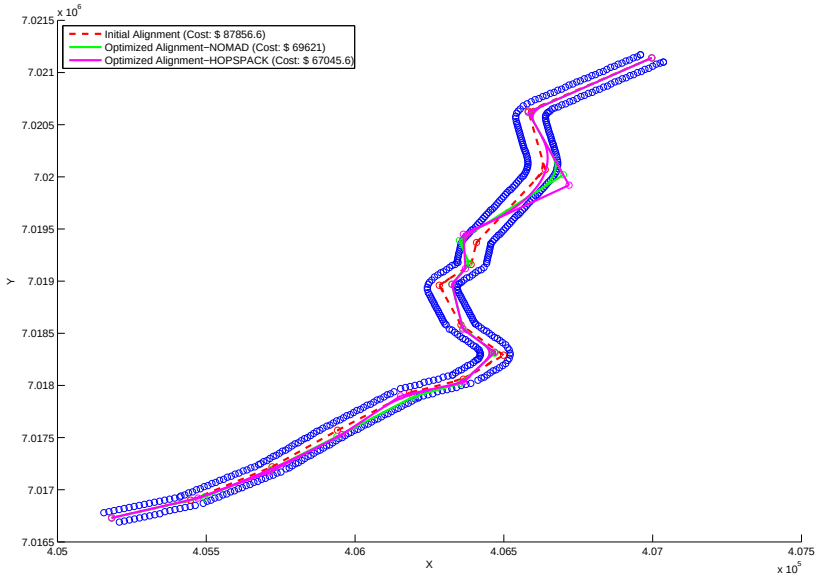

Figure B.2: Optimum alignments of the Road C obtained by the NOMAD solver and the HOPSPACK solver.

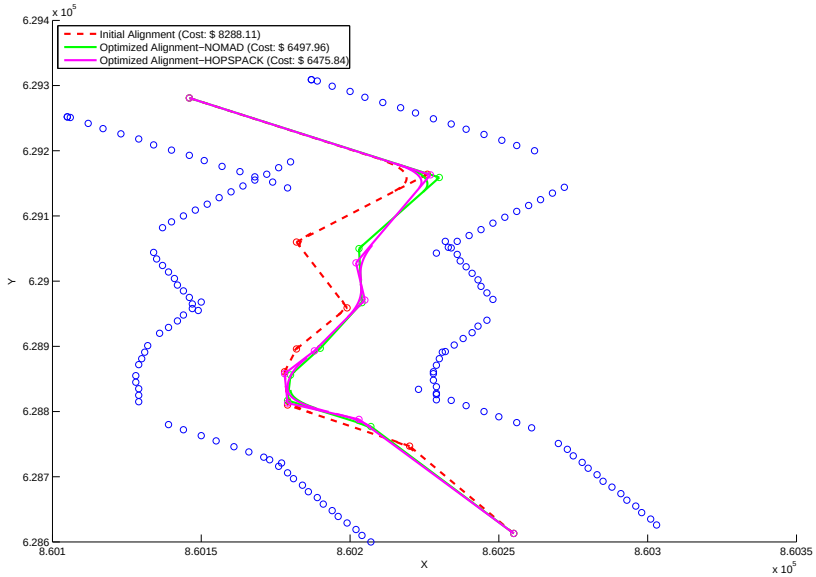

Figure B.3: Optimum alignments of the Road E obtained by the NOMAD solver and the HOPSPACK solver.