Road Design Optimization with a Surrogate Function

by

Yasha Pushak

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
B.SC. HONOURS IN COMPUTER SCIENCE HONOURS IN MATHEMATICS
in
Irving K. Barber School of Arts and Sciences
(Computer Science and Mathematics)

THE UNIVERSITY OF BRITISH COLUMBIA
(Okanagan)
April 2015
© Yasha Pushak, 2015
Planning new highways is a complex process requiring consideration of several non-trivial cost factors. An initial path is first refined by adjusting its vertical alignment to reduce earthwork cutting, filling, and hauling costs. This optimization problem is then used as a sub-problem that needs to be repeatedly solved with different horizontal alignments, to find an optimal path. The complete optimization process requires significant computation time to solve real-world problems. A multi objective surrogate function, which approximates the earthwork and paving costs and can be quickly evaluated, can be used to reduce the running time of the full optimization algorithm. This thesis builds upon an existing surrogate function, using a “Mass Haul” to approximate the earthwork hauling costs. Other constraints made to simplify the former surrogate are relaxed to improve the quality of the solutions returned by the surrogate. Numerical results compare solutions returned by the two versions of the surrogate and investigate the Pareto front between the earthwork and paving costs.
Table of Contents

Abstract ................................................................. ii
Table of Contents ....................................................... iii
Listings ................................................................. v
List of Figures .......................................................... vi
Acknowledgments ....................................................... ix

Chapter 1: Introduction .................................................. 1
  1.1 Road Design in the Literature ................................. 1
  1.2 Research Goal ................................................... 2
  1.3 Thesis Outline ................................................... 3

Chapter 2: Existing Models ............................................. 4
  2.1 The Three-Dimensional Model ................................. 4
    2.1.1 The Earthwork Problem ................................ 4
    2.1.2 The Vertical Alignment Problem ....................... 5
    2.1.3 The Horizontal Alignment Problem ................... 7
  2.2 The Surrogate Function Hirpa-2014 ......................... 9
    2.2.1 Road Alignment Model ................................ 9
    2.2.2 Terrain Model ............................................ 11
    2.2.3 Cost Formulation ....................................... 12

Chapter 3: Implementing the Surrogate .............................. 15
  3.1 Improving the Surrogate .................................... 15
    3.1.1 The Jacobian ............................................ 15
    3.1.2 Computing the Integral ................................ 17
    3.1.3 Relaxing the Constant Curve Elevation Constraint .. 19
    3.1.4 Mass-Haul ................................................ 22
# TABLE OF CONTENTS

3.2 Converting between the Full Model and the Surrogate . . . . 23
   3.2.1 Converting the Terrain . . . . . . . . . . . . . . . . . . 24
   3.2.2 Converting the Road Alignment . . . . . . . . . . . . 25

Chapter 4: Numerical Tests . . . . . . . . . . . . . . . . . . . . . 27
   4.1 Comparing the Mass-Haul Algorithm . . . . . . . . . . . . 27
   4.2 Pareto Front . . . . . . . . . . . . . . . . . . . . . . . . . . 28
   4.3 Cost Parameter Stability . . . . . . . . . . . . . . . . . . . . 31

Chapter 5: Conclusion . . . . . . . . . . . . . . . . . . . . . . . . 33

Bibliography . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

Appendix . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
   Appendix A: Source Code . . . . . . . . . . . . . . . . . . . . . 39
      A.1 Mass Haul Algorithm . . . . . . . . . . . . . . . . . . . . . 39
      A.2 Mass Haul Costs . . . . . . . . . . . . . . . . . . . . . . . 42
Listings

3.1 Source code written to calculate the closed form integrals for the volume of cutting and filling of tangent sections in the surrogate function Hirpa-2014. 18

3.2 Source code written to calculate the closed form integrals for the volume of cutting and filling of circular sections in the surrogate function Hirpa-2014. 18

3.3 Source code written to calculate the closed form integrals for the volume of cutting and filling of curve sections in the surrogate function Pushak-2015. 20

A.1 An Implementation of the Mass Haul algorithm. An emphasis was made on code readability and documentation for future use. 39

A.2 A helper function used by the Mass Haul Algorithm in listing A.1 to determine the costs of hauling some volume of earth over a set distance. 42
List of Figures

Figure 2.1 A sample solution for the Earthwork Problem. The elevation profile of the ground is the solid black line. The dotted black line is the elevation profile of the road. ................................................................. 5

Figure 2.2 A summary of the Earthwork Problem. ............................. 5

Figure 2.3 A sample solution for the Vertical Alignment Problem. The elevation profile of the ground is the solid black line. The dotted black line is the elevation profile of the road. We see that in comparison to Figure 2.1, the modified vertical alignment has reduced the total volume of cutting and filling, which decreases the total cost of the vertical alignment. .......................... 6

Figure 2.4 A summary of the Vertical Alignment Problem. The Earthwork Problem is used as a component of the Vertical Alignment Problem. ................................................................. 7

Figure 2.5 A sample solution for the Horizontal Alignment Problem. The solid black curvy-linear line represents the initial alignment selected by engineers. The gray lines are the cross sections which define the horizontal corridor. The elevation of the ground is given at each of the points along the cross sections. The dotted black line represents a hypothetical optimal solution to the Horizontal Alignment Problem. ................................................................. 7

Figure 2.6 A summary of the Horizontal Alignment Problem. The Vertical Alignment Problem is treated as a black-box, for which the optimal solution is used as the objective function for the Horizontal Alignment problem. ................................................................. 8

Figure 2.7 The horizontal alignment model used by the surrogate function. ................................................................. 9
LIST OF FIGURES

Figure 2.8  The vertical alignment model used by the surrogate function. Here, the number of vertical segments per horizontal tangent is $m_i = 5$ for $1 \leq i \leq n$. Note that, when projected onto the $xy$-plane, the length of each vertical segment is exactly $1/m_i$ the length of the corresponding horizontal tangent. The height of the road is constant over the length of each circular curve, hence $z_5 = z_6$. .......................... 10

Figure 2.9  Visualization of the piecewise-linear function used to model the elevation of the terrain. Here $\delta x = \delta y = 10$ (meters). ................................................. 12

Figure 2.10 On the left, the cross-sectional area of cut for a point $s_1$ that is in a cut region. On the right, the cross-sectional area of fill for a point $s_2$ that is in a fill region. ................................................. 14

Figure 3.1  A sample of an optimal solution found using the surrogate function with Nomad [AAC$^+$. The test was run with and without the constraint which imposes a constant elevation over the circular sections of the road. The black line is the original surrogate with the constraint and the white line is the relaxed one. The $x$ and $y$ axes are in meters, the $z$ axis is in decameters. 20

Figure 3.2 The volume of earthwork that needs to be done along the length of the roads in Figure 3.1. The original surrogate is in black and the version with the relaxed curve elevation constraint is in gray. A value above zero indicates a region of cut and a value below zero indicates a region of fill. ......................... 21

Figure 3.3  The visual representation of using the distance and offset coordinates of the terrain coordinate as the input to the surrogate function. ......................... 24

Figure 4.1  A comparison of the surrogate function with and without the Mass-Haul algorithm. The original optimal surrogate road alignment is in white, the alignment found with the surrogate function using the Mass-Haul algorithm is in black. The $x$ and $y$ axes are in meters, the $z$ axis is in decameters. ......................... 28
Figure 4.2  The volume of earthwork that needs to be done along the road alignments in Figure 4.1. A positive value indicates a region of cut, while a negative value indicates the region of fill. The original optimal surrogate road alignment is in gray, the alignment found with the surrogate function using the Mass-Haul algorithm is in black. ........................................... 29

Figure 4.3  The plot of all solutions found by the weighted sum method. The boxed points are the ones that belong to the pareto front. Note the vertical scale does not start at zero and all costs are within 10%. ............ 30

Figure 4.4  The plot of the pareto fronts found by the various algorithms with their respective parameters. ........ 31

Figure 4.5  Five different optimal road alignments found with different weights for the cost parameters. The white line represents a value of $\theta = 0$ (all user costs) and the black line represents a value of $\theta = 1$ (all earthwork costs). The three shades of the grey lines represents the values 0.25, 0.5 and 0.75. The $x$ and $y$ axes are in meters, the $z$ axis is in decameters. ............... 32
Acknowledgments

I would like to thank my supervisors Dr. Yves Lucet and Dr. Warren Hare, as well as Softree Technical Systems Inc. for the input data used to gather results and engineering advise on road design. Thank you for the funding received through CFI for the CA² lab, and the NSERC funding for the computer used to gather the results.
Chapter 1

Introduction

Modern road design is a complex process requiring consideration of several different cost factors, be them political, environmental or financial. Determining the optimal road alignment for new highways or forestry roads is a non-trivial problem which cannot be solved by hand. In order to accurately model and capture the different costs and constraints computers are required to aid the process and determine the optimal solution.

Current numerical optimization techniques exist to find locally optimal solutions to the road design problem. These methods yield satisfactory solutions, however require a significant amount of computation time to converge. This thesis is primarily focused with developing a surrogate function which approximates the costs of the full optimization problem, yet is quick to evaluate. The intended use for the surrogate function is to be combined with existing methods to speed up the overall optimization process.

1.1 Road Design in the Literature

The first problem which needs to be solved when designing a new highway or forestry road, is planning an initial alignment. When connecting a start and destination point with a road, it can be a nontrivial choice of deciding if the road should go around one side or another of a mountain, lake, or other obstacle. Currently this problem is done by hand, with engineers sketching an initial plan for a road alignment. Most existing work in road design optimization has been done in refining the road alignments planned by engineers. There has been some work done [PLH14] which addresses this problem by finding a set of locally optimal paths which serve as candidates corridors for further refinement. Similar methods for finding spatially distinct corridors have been done by Church et al. [CLL92, LC93], which was later improved by Scaparra et al. [SCM14]. Another approach for this type of problem was used by Zhang et al. [ZA05] using genetic algorithms.

Once an initial corridor is selected, the alignment needs to be refined. Typically this is treated only as a 2-dimensional problem, seeking to find the optimal horizontal alignment within a corridor. There have been sev-
eral different approaches used to solve the horizontal alignment problem. The typical formulation begins with some form of a locally optimal corridor and aims to find the optimal road alignment within the corridor. This has been done using, for example, genetic algorithms as done by Kang et al. [Kan08, KJS12] or Jha et al. [Jha03, JS04, JK06]. Mondal [Mon14] solved the problem using derivative-free optimization solvers as NOMAD [AAC+] and HOPSPACK [Lib]. The cost formulation for the road varies, but a common approach is to separate the horizontal and vertical alignment problems. One way of doing this is as a bi-level optimization problem using the vertical alignment as a sub-optimization problem to the horizontal alignment optimization problem. One example of a bi-level approach was done by Mondal [Mon14]. In a similar method to ones describes in this thesis, Lee et al. [LTL09] modeled the horizontal alignment as a piecewise-linear line and applied safety constraints after finding the optimal piecewise-linear alignment. Another take to improving the efficiency of the horizontal alignment problem was done by Kang et al. [KSJ07] using feasible gates to reduce the search space.

Closely linked to the cost of the horizontal alignment, is the road’s vertical alignment, which is needed to determine the earthwork costs. In order to simplify models, the vertical alignment problem is typically treated separately from the horizontal problem, or as a subproblem of the full alignment problem. It is primarily concerned with minimizing the earthwork costs associated with constructing new roads, which are a significant cost factor in road construction [JSJ06]. There are three components to the earthwork costs: cutting, filling and hauling. The first two are static for a given road alignment and are proportional to the volume of earthwork that needs to be done along the road. The third cost is determined by the strategy used to move the earth along the road. The goal is to minimize the cost of excavation and embankment, while finding the earth-hauling plan which minimizes the cost of transporting the material. Some work on minimizing the earthwork costs was done by Hare et al. [HIKL11], which was later expanded upon in a method for solving the vertical alignment [HHLR14].

1.2 Research Goal

This thesis builds upon an existing surrogate model developed by Hirpa [Hir14]. The surrogate function is designed to serve as a quick approximation of the construction costs for a given three-dimensional road alignment. The ultimate goal for the surrogate is to be combined with a bi-level optimization
1.3. Thesis Outline

approach developed by Mondal [Mon14]. Since the full three-dimensional problem is structured as a bi-level optimization problem, it requires fully solving the vertical alignment for every horizontal alignment, which yields a slow algorithm. The goal is to combine the surrogate with this method in a way that uses the surrogate function to quickly make progress, while occasionally using the full model to validate that the surrogate function is making steps in the correct direction.

1.3 Thesis Outline

In Chapter 1 we introduce the road design problems and provide motivation for construction and use of the surrogate function. Chapter 2 begins with an overview of the full three-dimensional alignment problem and one method for how it has been solved in literature. Chapter 2.1 provides some background into how the earthwork, vertical and horizontal problems are used to form the full, three-dimensional model. In Chapter 2.2 we discuss the formulation for an existing surrogate function by Hirpa [Hir14]. Chapter 3 describes the work done for this thesis to implement the surrogate function described in Chapter 2.2. Chapter 3.1 contains several modifications and improvements made to the existing surrogate. Chapter 3.2 discusses possible methods for converting between the full model and the surrogate model. In Chapter 4 we briefly discuss some preliminary results obtained with the surrogate and provide concluding remarks and ideas for future work in Chapter 5.
Chapter 2

Existing Models

This chapter provides an overview of the specific models that the lab is using. We describe how the road and terrain are modeled and briefly cover the cost formulation.

2.1 The Three-Dimensional Model

This model is used to refine an initial alignment provided by an engineer. An initial corridor is selected manually and the elevation data within the corridor is used as input for this model. A derivative-free solver is used to find the optimal road alignment within the corridor provided by engineers. This model has been developed as part of an ongoing research project [HKL11, HHLR14]. Most recently the horizontal component was completed by Mondal [Mon14].

2.1.1 The Earthwork Problem

The road design problem requires consideration of several different cost factors. Once a horizontal and vertical alignment have been chosen, we then need to find the optimal earthwork solution. Suppose we begin with the vertical alignment seen in Figure 2.1. The goal of the Earthwork Problem is to find the optimal strategy to move the earth along the road.

Some of the material that needs to be cut can be used to fill in ground to bring it to the level of the road elsewhere. How far and how much of this material is moved affects the cost of the earthwork solution. In some cases, the type of material that exists along the road needs to be replaced with better quality earth, which will increase the cost of the earthwork solution as well. In order to build the final road, some material is required to be brought to the road, while other material must be removed and deposited at waste pits. Depending on the distance a given volume of material needs to be moved, different types of trucks or bulldozers will be used, which again alters the cost of the final earthwork solution.
2.1. The Three-Dimensional Model

A summary of the earthwork problem is given in Figure 2.2. For more details on its formulation and how it can be solved the reader is referred to [HKL11].

The Earthwork Problem

- **Input:** A vertical alignment and ground profile
- **Minimize:** The earth hauling costs
- **Output:** The cost of cutting, filling, and hauling

Figure 2.2: A summary of the Earthwork Problem.

2.1.2 The Vertical Alignment Problem

The model used for the vertical alignment is a piecewise-quadratic function. In order to meet minimum safety criteria, various constraints are imposed upon the vertical alignment. These constraints include a restriction on the maximum curvature of the vertical alignment and require that the alignment is both continuous and smooth.
2.1. The Three-Dimensional Model

The cost formulation for the Vertical Alignment Problem is closely linked with the Earthwork Problem. By altering the vertical alignment, we can adjust the volume of cutting and filling that is required along the road. Minimizing these volumes reduces the excavation and embankment costs, as well as the volume of earthwork that needs to be hauled, see Figure 2.3. Hence, the cost of a given vertical alignment can be found by solving the Earthwork Problem.

While the Earthwork Problem could then be treated as a black box and solved as a subproblem of the Vertical Alignment problem, it is faster in practice to solve the two by simultaneously optimizing the vertical alignment and the earth hauling costs. The final output of the Vertical Alignment problem yields the vertical alignment with the optimal earthwork costs. A visual summary of the Vertical Alignment Problem is presented in Figure 2.4.

For more details on how the Vertical Alignment Problem is modeled and solved, the reader is referred to [HHLR14].
2.1. The Three-Dimensional Model

2.1.1 The Vertical Alignment Problem

- **Input:** A ground profile
- **Minimize:** The Earthwork costs
- **Output:** The vertical alignment with minimal cost

Update

The Earthwork Problem
- **Input:** A vertical alignment and ground profile
- **Minimize:** The earth hauling costs
- **Output:** The cost of cutting, filling, and hauling

2.1.3 The Horizontal Alignment Problem

The goal for the Horizontal Alignment Problem is to refine an initial alignment provided by engineers. To this end, a corridor around the alignment is given which provides the feasible region for the road and the terrain model for the Horizontal Alignment Problem.

Figure 2.4: A summary of the Vertical Alignment Problem. The Earthwork Problem is used as a component of the Vertical Alignment Problem.

Figure 2.5: A sample solution for the Horizontal Alignment Problem. The solid black curvy-linear line represents the initial alignment selected by engineers. The gray lines are the cross sections which define the horizontal corridor. The elevation of the ground is given at each of the points along the cross sections. The dotted black line represents a hypothetical optimal solution to the Horizontal Alignment Problem.
2.1. The Three-Dimensional Model

The Horizontal Alignment Problem

- Input: A horizontal corridor and terrain data
- Minimize: The Earthwork and Paving Costs
- Output: The horizontal and corresponding vertical alignment with minimal cost

The Vertical Alignment Problem

- Input: A ground profile
- Minimize: The Earthwork costs
- Output: The vertical alignment with minimal cost

Figure 2.6: A summary of the Horizontal Alignment Problem. The Vertical Alignment Problem is treated as a black-box, for which the optimal solution is used as the objective function for the Horizontal Alignment problem.

The initial, horizontal road alignment is modeled as a piecewise curvy-linear line. At equally spaced intervals along this line we are given a cross-section which is perpendicular to the road. The feasible corridor for the Horizontal Alignment Problem is then defined by these cross-sections. Additionally, each cross-section is given a set of equally spaced points, which are used to define the terrain model, see Figure 2.5. At each point, information such as the elevation of the ground, the type and percentage of materials present are specified.

Given a horizontal alignment, the terrain data along the alignment can be used as an input to the Vertical Alignment Problem. The cost of the Horizontal Alignment Problem includes both the earthwork costs from the vertical solution and the cost of paving the road. The Vertical Alignment Problem can then be treated as a black box used in the objective function of the Horizontal Alignment Problem. A derivative-free method is applied to the Horizontal Alignment Problem to find the optimal solution. A visual summary of the Horizontal Alignment Problem is presented in Figure 2.6.

For more details on the model and implementation of the Horizontal Alignment problem, the reader is referred to Mondal [Mon14].
2.2 The Surrogate Function Hirpa-2014

The surrogate function described in this section was designed by Hirpa [Hir14] to provide a quick approximation of the costs associated with building a new road.

2.2.1 Road Alignment Model

Similar to the full model, the surrogate splits the road’s alignment into two parts: the horizontal alignment and the vertical alignment.

The horizontal alignment is a piecewise circular-linear curve. The coordinates of the start and end of the road are fixed and a set of \( n \) intersection points and radii define the road’s horizontal alignment. Let \((x_s, y_s)\) and \((x_e, y_e)\) be the coordinates for the start and end of the road respectively and let \(\{(x_1, y_1, r_1), (x_2, y_2, r_2)\ldots (x_n, y_n, r_n)\}\) be the set of intersection points and their radii, as seen in Figure 2.7.

![Horizontal Road Alignment](image)

Figure 2.7: The horizontal alignment model used by the surrogate function.

We denote the segments between circular pieces as tangents. The transition points from tangents to circular pieces form the set

\[\{TC_1, TC_2, \ldots, TC_n\}\]
and the transition points from circular pieces to tangents the set
\[ \{CT_1, CT_2, \ldots, CT_n\} \].

These sets are not inputs to the model, as they can be computed from the
intersection points and their radii.

Also included in the horizontal model is the width of the road, \( w \) in
meters, which is assumed to be constant for the length of the road.

The vertical alignment model adds elevation data to the horizontal align-
ment to create the 3 dimensional alignment. The \( i \)th horizontal tangent
section is split into \( m_i \) segments of equal length. These \( m_i \) segments de-
fine a piecewise-linear vertical alignment for each horizontal tangent, as in
Figure 2.8.

![Vertical Road Alignment](image)

Figure 2.8: The vertical alignment model used by the surrogate function. Here, the number of vertical segments per horizontal tangent is \( m_i = 5 \) for
\( 1 \leq i \leq n \). Note that, when projected onto the \( xy \)-plane, the length of each
vertical segment is exactly \( 1/m_i \) the length of the corresponding horizontal
tangent. The height of the road is constant over the length of each circular
curve, hence \( z_5 = z_6 \).

**Alignment Constraints**

There are several constraints imposed on the surrogate alignment, either
for simplicity, or to ensure that the road produced meets safety criteria. An
implicit constraint enforced by the alignment’s formulation ensures that
2.2. The Surrogate Function Hirpa-2014

The road alignment is continuous. In practice a full road alignment must also be smooth, however this constraint is relaxed in the vertical model for simplicity.

In the horizontal alignment the transition point \( CT_i \) must not pass the point \( TC_{i+1} \), in order to ensure a smooth transition between the curves. A minimum radius constraint is imposed on the horizontal alignment for safety. To reduce the search space in the optimization process, a maximum curvature radius is also specified.

In the vertical alignment, the elevation distance between two consecutive points is bounded to ensure a maximum road grade. For simplicity in implementation, each horizontal tangent section is assumed to have the same \( m_i = m \) number of vertical segments. One constraint of note was made to simplify the cost calculations, which was to make the road elevation constant over the circular segments.

2.2.2 Terrain Model

The terrain model used for the surrogate function is a piecewise-linear function. This model was chosen for its simplicity and because a piecewise continuous function was needed to calculate the integrals in Chapter 2.2.3. Each linear plane of the model is a rectangle with fixed dimensions, \( \delta x \) and \( \delta y \).

The model, \( M(x, y) = z \), is stored using three matrices, \( A \), \( B \) and \( C \), where

\[
A = \frac{\partial M}{\partial x}, \\
B = \frac{\partial M}{\partial y}, \\
C = z(x, y) - x \frac{\partial M}{\partial x} - y \frac{\partial M}{\partial y}.
\]

We introduce the variables

\[
i_x := \text{floor}\left(\frac{x}{\delta x}\right) + 1, \\
j_y := \text{floor}\left(\frac{y}{\delta y}\right) + 1,
\]

such that

\[
A(i_x, j_y) = \frac{\partial M}{\partial x}\bigg|_{(x,y)},
\]
2.2. The Surrogate Function Hirpa-2014

Figure 2.9: Visualization of the piecewise-linear function used to model the elevation of the terrain. Here $\delta x = \delta y = 10$ (meters).

$$B(i_x, j_y) = \frac{\partial M}{\partial y} \bigg|_{(x,y)}.$$  

The value of $z$ for the model can then be found using

$$z = M(x, y) = xA(i_x, j_y) + yB(i_x, j_y) + C(i_x, j_y).$$

2.2.3 Cost Formulation

Road construction contains many different factors contributing to the final cost of the road. For a typical new highway, two of the dominating costs are the earthwork and paving costs [JSJ06]. In order to capture both of these cost factors, the surrogate is a multi-objective function that returns two cost components, $Cost_{earth}$ and $Cost_{paving}$.  

\[ \text{Surrogate Terrain Model} \]
The cost calculations uses a parametric form of the road alignment given by \( r(s) \), where
\[
    r(s) = (x(s), y(s), z(s)).
\]

Earthwork Costs

The surrogate earthwork costs include three parts: the cost of cutting, filling and the waste cost. The total earthwork costs are calculated as
\[
    \text{Cost}_{\text{earth}} = V_c C_c + V_f C_f + V_w C_w,
\]
where \( V_c, V_f \) and \( V_w \) are the volume of cutting, filling and earth wasted in cubic meters, respectively, and the user specified parameters \( C_c, C_f \) and \( C_w \) are the costs of cutting, filling and waste costs per cubic meter, respectively.

The waste costs are a simplification used in place of the earth hauling costs. The waste volume is simply the difference of the volume of cutting and filling, i.e.,
\[
    V_w = |V_c - V_f|.
\]
This waste cost then applies a penalty to any excess earth that needs to be brought to or removed from the road. We note that in this cost formulation there are no earthwork hauling costs.

In order to calculate the volume of cut and fill, we need the cross sectional area and length of the road. Since the road is defined as a piecewise circular-linear curve and the terrain is modeled as a piecewise-linear function, these calculations need to be split up into several sections. Hence we have
\[
    V_c = \sum_{i \in CR} \int_{a_i}^{b_i} \text{area}_c(s) ds,
\]
where \( \text{area}_c(s) \) is the cross-sectional area of the road that needs to be cut at \( s \) and \( i \in CR \) if \( [a_i, b_i] \) brackets a cut region of the road, i.e., one where the ground needs to be cut. Similarly, we have
\[
    V_f = \sum_{i \in FR} \int_{a_i}^{b_i} \text{area}_f(s) ds,
\]
where \( \text{area}_f(s) \) is the cross-sectional area of the road that needs to be filled at \( s \) and \( i \in FR \) if \( [a_i, b_i] \) brackets a fill region of the road.

The cross-sectional area of the road that needs to be cut is calculated as a trapezoid defined by the two angles \( \theta_1 \) and \( \theta_2 \), the width of the road, \( w \),
and the distance of the road from the ground, $h_c(s)$, as in Figure 2.10. We denote $z_r(s)$ as the height of the road at $s$ and $z_g(s) = M(x(s), y(s))$ as the height of the ground at $(x(s), y(s))$. We can find the distance of the road from the ground as $h_c(s) = z_g(s) - z_r(s)$. Then

$$area_c = h_c(s)(w + \frac{1}{2} \kappa h_c(s)),$$

where $\kappa = \cot(\theta_1) + \cot(\theta_2)$.

Symmetrically we let $h_f(s) = z_r(s) - z_g(s)$, and calculate the cross-sectional area of road that needs to be filled as

$$area_f = h_f(s)(w + \frac{1}{2} \kappa h_f(s)).$$

**Paving Costs**

The paving costs are simply calculated using the length and width of the road

$$Cost_{paving} = \int wC_p ds,$$

where $C_p$ is the cost of paving the road per meter squared and $w$ is the width of the road.
Chapter 3

Implementing the Surrogate

An initial implementation of the Surrogate as described in Chapter 2.2 was written by Hirpa [Hir14] in MATLAB [Inc]. This code was used as a proof of concept for the road design surrogate function. Beginning with this prototype implementation as a reference, a new surrogate function was developed for this thesis.

3.1 Improving the Surrogate

In order to use the surrogate function it needed to be converted from MATLAB to C code. This change was required to reduce the running time of the algorithm for use in the full algorithm. This process was done using a MATLAB to C code conversion tool. In order to use this tool significant refactoring of the original code was required.

During refactoring, some implementation and design problems were uncovered and fixed. In particular there were two errors in how the calculations for the volume of cutting and filling were performed that were discovered. In order to ensure the quality of the final surrogate twenty-four unit tests were implemented. During the refactoring and testing some undesirable behaviours of the solutions returned by the surrogate were observed. Modifications to the surrogate were made to remove these defects and improve the quality of solutions found with the surrogate. We denote the original surrogate design by Hirpa-2014 and the modified one with Pushak-2015.

3.1.1 The Jacobian

The original coordinates used to define the road alignment model are standard Euclidean coordinates. As described in Chapter 2.2.3, these coordinates are transformed to parametric form to calculate the cost of the surrogate alignment. Recall the parametric form

\[ r(s) = (x(s), y(s), z(s)), \]
3.1. Improving the Surrogate

and the calculation for the volume of cut

\[ V_c = \sum_{i \in CR} \int_{a_i}^{b_i} \text{area}_c(s)ds. \]

Since this integral is computed using the parameter \( r(s) \) in place of \((x, y, z)\), the conversion from \( dx dy dz \) to \( ds \) requires the use of a Jacobian, which was omitted in the original design. In this case, the Jacobian is computed as

\[ |r'(s)|, \]

where \(|\cdot|\) is the Euclidean norm, so

\[ dx dy dz = |r'(s)|ds = |(x'(s), y'(s), z'(s))|ds. \]

To simplify the implementation, we vary the parameter \( s \) from 0 to 1 for each continuous region \([a_i, b_i]\) that can be computed with a single integral. The parametric form of the equations for the tangent sections connecting points \((x_j, y_j, z_j)\) and \((x_{j+1}, y_{j+1}, z_{j+1})\) can be written as

\[
\begin{align*}
x(s) &= x_j + (x_{j+1} - x_j)s, \\
y(s) &= y_j + (y_{j+1} - y_j)s, \\
z(s) &= z_j + (z_{j+1} - z_j)s.
\end{align*}
\]

So the Jacobian is

\[
|r'(s)| = |((x_{j+1} - x_j), (y_{j+1} - y_j), (z_{j+1} - z_j))| = |(x_{j+1}, y_{j+1}, z_{j+1}) - (x_j, y_j, z_j)|,
\]

which is simply the distance between the two points.

For the circular sections of the road, the parametric equation is written as

\[
\begin{align*}
x(s) &= C_j + r_j \cos(\theta(s)), \\
y(s) &= C_j + r_j \sin(\theta(s)), \\
z(s) &= z_j = z_{j+1},
\end{align*}
\]

where \(C_j\) is the center of the circle, \(r_j\) is the radius of the circle and \(\theta(s) = \theta_j + (\theta_{j+1} - \theta_j)s\), such that \(r(0) = (x_j, y_j, z_j)\) and \(r(1) = (x_{j+1}, y_{j+1}, z_{j+1})\).
The Jacobian is then
\[
||r'(s)|| = ||(-r_j \sin(\theta(s))\theta'(s), r_j \cos(\theta(s))\theta'(s), 0)||
\]
\[
= \sqrt{(r_j \theta'(s) \sin(\theta(s)))^2 + (r_j \theta'(s) \cos(\theta(s)))^2}
\]
\[
= \sqrt{(r_j \theta'(s))^2 (\sin^2(\theta(s)) + \cos^2(\theta(s)))}
\]
\[
= \sqrt{(r_j \theta'(s))^2}
\]
\[
= |r_j \theta'(s)|
\]
\[
= |r_j(\theta_{j+1} - \theta_j)|.
\]

### 3.1.2 Computing the Integral

A closed form version of the volume of cut and fill integrals was used in the prototype implementation of the surrogate function. These integrals were originally solved by hand and typed into the code. This process requires careful attention to notation and is prone to human error.

As an example, an easy error that was made stated that the volume of fill was the negative of the volume of cut. While intuitively correct, the direct implementation of the integral could not simply be negated. Recall from Chapter 2.2.3

\[
\text{area}_c = h_c(s)(w + \frac{1}{2}\kappa h_c(s)),
\]
\[
h_c(s) = z_g(s) - z_r(s)
\]

and

\[
\text{area}_f = h_f(s)(w + \frac{1}{2}\kappa h_f(s)),
\]
\[
h_f(s) = z_r(s) - z_g(s).
\]

The correct relationship is

\[
h_f(s) = -h_c(s),
\]

which yields

\[
\text{area}_f = -h_c(s)(w - \frac{1}{2}\kappa h_c(s)) \neq -h_c(s)(w + \frac{1}{2}\kappa h_c(s)) = \text{area}_c.
\]

To avoid human error, the symbolic MATLAB code in Listing 3.1 was used to compute the closed form integrals that were used in the surrogate.
3.1. Improving the Surrogate

Listing 3.1: Source code written to calculate the closed form integrals for the volume of cutting and filling of tangent sections in the surrogate function Hirpa-2014.

Similarly, the closed form integral for volume of cut and fill integrals for the circular sections of the road can be calculated using the symbolic MATLAB code in Listing 3.2.

Listing 3.2: Source code written to calculate the closed form integrals for the volume of cutting and filling of tangent sections in the surrogate function Hirpa-2014.
3.1. Improving the Surrogate

\[ ac = w \cdot h + \frac{1}{2} \cdot \kappa \cdot h^2; \]
\[ Vc = \int (ac \cdot \text{Jacobian}, s, s1, s2); \]
\%
\text{Calculate the volume of fill for curves.}
\[ hf = zr - zg; \]
\[ af = ah + \frac{1}{2} \cdot \kappa \cdot h^2; \]
\[ Vf = \int (af \cdot \text{Jacobian}, s, s1, s2); \]

Listing 3.2: Source code written to calculate the closed form integrals for the volume of cutting and filling of circular sections in the surrogate function Hirpa-2014.

3.1.3 Relaxing the Constant Curve Elevation Constraint

Preliminary tests using the new surrogate were performed by using NO-MAD [AAC$^+$], a derivative-free optimization solver. An example of one of the graphical solutions is displayed in Figure 3.1, which highlights an undesirable behaviour of the circular road sections. We see that in the original surrogate model the constraint requiring the elevation to be constant over the curve located on a steep hill causes the road alignment to extend far underneath of the level of the ground. Figure 3.2 shows the volume of earthwork that needs to be done along the road alignment in Figure 3.1. The two large bumps occur during the circular sections of the road, adding significant cost to the optimal solutions. The second curve in this image corresponds to the one noted in the first figure, while the other one is not visible in the other image due to the angle.

This behaviour of requiring increased earthwork around the circular sections of the road was hypothesized to be dominating the other costs and hence was detracting from the quality of the solutions found with the surrogate. With the earthwork volume integrals being calculated symbolically there was no longer a need for a simplified model and so the constant elevation constraint was relaxed.

For the circular sections of the road, the new parametric equation is written as

\[ x(s) = C_j + r_j \cos(\theta(s)), \]
\[ y(s) = C_j + r_j \sin(\theta(s)), \]
\[ z(s) = z_j + (z_{j+1} - z_j)s, \]

where $C_j$ is the center of the circle, $r_j$ is the radius of the circle and $\theta(s) = \theta_j + (\theta_{j+1} - \theta_j)s$, such that $r(0) = (x_j, y_j, z_j)$ and $r(1) = (x_{j+1}, y_{j+1}, z_{j+1})$. 

19
3.1. Improving the Surrogate

Figure 3.1: A sample of an optimal solution found using the surrogate function with Nomad [AAC\textsuperscript{+}]. The test was run with and without the constraint which imposes a constant elevation over the circular sections of the road. The black line is the original surrogate with the constraint and the white line is the relaxed one. The $x$ and $y$ axes are in meters, the $z$ axis is in decameters.

The Jacobian is then
\[
||r'(s)|| = ||(-r_j \sin(\theta(s)) \theta'(s), r_j \cos(\theta(s)) \theta'(s), z_{j+1} - z_j)||
\]
\[
= \sqrt{(r_j \theta'(s) \sin(\theta(s)))^2 + (r_j \theta'(s) \cos(\theta(s)))^2 + (z_{j+1} - z_j)^2}
\]
\[
= \sqrt{(r_j(\theta_{j+1} - \theta_j))^2 + (z_{j+1} - z_j)^2}.
\]

Unlike the tangent sections of the road, the curves were only given an elevation at their beginning and end, defining a constant slope over the curve. The results seen in figures 3.1 and 3.2 show that the single elevation change was still able to yield solutions where the road was much better able to match the elevation of the ground. Recalling that the road alignments are restricted within corridors, we see in Figure 3.1 that the new road alignment found has moved substantially to the left.

The updated Jacobian and parametric equations are solved with the symbolic MATLAB code in Listing 3.3 to produce the final, closed form integrals used in the new version of the surrogate function.

```matlab
% Set up the symbolic code.
syms x1 x2 y1 y2 z1 z2 r A B C kappa w j m s s1 s2 theta1 theta2
```
3.1. Improving the Surrogate

Figure 3.2: The volume of earthwork that needs to be done along the length of the roads in Figure 3.1. The original surrogate is in black and the version with the relaxed curve elevation constraint is in gray. A value above zero indicates a region of cut and a value below zero indicates a region of fill.

Listing 3.3: Source code written to calculate the closed form integrals for the volume of cutting and filling of curve sections in the surrogate function Pushak-2015.
3.1. Improving the Surrogate

3.1.4 Mass-Haul

In some roads the earth hauling costs comprise roughly half of the total earthwork costs [JSJ06]. Given this fact, we hypothesized that assuming the earth hauling costs are negligible will significantly affect the solutions returned by the surrogate. A Mass-Haul algorithm was used to quickly approximate the cost of the optimal earth hauling solution. The Mass-Haul algorithm was chosen for its relatively low runtime complexity, despite providing a quick cost approximation.

Algorithm 1: The Mass Haul Algorithm.

\begin{verbatim}
input : An array of volumes, \( vol \), and an array of distances, \( dist \), each of size \( n \).
output: An estimate of the cost of the optimal earthwork solution.

/* Preprocess so that the earthwork balances out in the end. */
if totalVolumeCut > totalVolumeFill then
  Find largest \( vol(i) > 0 \);  
  Move excess mass from \( vol(i) \) to nearest waste pit;
else
  Find largest \( vol(i) < 0 \);  
  Move extra mass from nearest waste pit to \( vol(i) \);
end

/* The main loop for the Mass-Haul Algorithm. */
j = 1;
for \( i = 1 \) to \( n \) do
  while \( vol(i) > 0 \) do
    while \( vol(j) <= 0 \) do
      \( j++ \);
    end
    /* Guaranteed to find \( j > 0 \) because of preprocessing. */
    Move \( vol(i) \) to \( vol(j) \);
  end
end
\end{verbatim}

The Mass-Haul algorithm takes as input two arrays of length \( n \). The first stores the volume of earthwork needing to be done along different segments of the road and the second stores the length of each segment of the road.
The cost of moving material depends on the volume of material and the distance it needs to be moved. Earth that is moved a longer distance can be moved with different kinds of trucks, while short movements can simply be done with a bulldozer. Preprocessing can be done to determine the optimal type of truck for a given distance. Additional input parameters are used in the Mass-Haul algorithm to model these costs.

An initial pass is done by the algorithm to determine whether or not there is a greater volume of cutting or filling that needs to be done along the road. In the case where there is more cutting, the segment of the road with the most volume of cut is found and sufficient material is moved to the nearest waste pit to ensure that the rest of the road’s earthwork will be balanced, that is, the road can be made without using another waste pit. Symmetrically, when there is more filling required, the segment with the largest volume of fill is found and extra material is brought to the road.

A second pass is then done, which scans along the road until it finds the first segment which requires earthwork. If cutting is required, then the nearest available fill segment is used to move the earth. The filling segments are handled symmetrically.

The time complexity of the Mass-Haul algorithm is found by observing that the both the preprocessing and main body of the algorithm require a constant number of passes over the road. The preprocessing performs a single pass to determine the position with the most mass and whether or not the total volume of cutting or filling is greater. The main body of the algorithm, while it appears to have multiple nested loops, is in fact also done in linear time. This is because it uses two variables, \(i\) and \(j\) which each pass from 1 to \(n\), tracking the positions of cut and fill respectively. This yields a time complexity of \(O(n)\), where \(n\) is the number of segments along the road.

Pseudo-code for the mass-haul algorithm is included in Algorithm 1. The full source code available in appendix A.1.

3.2 Converting between the Full Model and the Surrogate

Using the surrogate in combination with the full model requires defining a mapping between the full model’s road alignment and terrain data. While the full optimization process has not yet been adapted to use this surrogate, this section describes a two hypothetical conversion techniques that could be used. The sketch of the conversion techniques and their relative complexity are discussed in Chapter 3.2.1. In Chapter 3.2.2, we examine
3.2. Converting between the Full Model and the Surrogate

the challenges that arise from the two methods when converting the road alignments’ coordinates between the two models and present methods for the conversion.

3.2.1 Converting the Terrain

This section summarizes two different methods for converting the initial terrain corridor data provided by engineers into the piecewise linear step function used by the surrogate.

Option A

The first approach is the simplest method for converting the terrain. In the initial input data, the distance between each cross-section is always a constant size. Additionally, the points are always equally spaced along the cross sections. Instead of using the proper Euclidean coordinates, we can simply use the distance along the road as the $x$ coordinate in the surrogate, and the offset from the initial alignment as the $y$ coordinate. We denote this system using $(d,o)$, where $d$ is the distance and $o$ is the offset. The visual interpretation for this coordinate system is analogous to simply stretching the original coordinate system as seen in Figure 3.3.

![Figure 3.3: The visual representation of using the distance and offset coordinates of the terrain coordinate as the input to the surrogate function.](image)

One possible flaw with this model, is that it may not provide accurate solutions for the curved portions of the corridor. Numerical testing needs to be done to determine whether or not this simplification will significantly affect the quality of the solutions found by the surrogate.

Option B

In this section, we describe the sketch of an algorithm for another approach to the terrain conversion problem. Here, we use the euclidean coord-
3.2. Converting between the Full Model and the Surrogate

Coordinates provided by the initial corridor instead of the distance and offset. Since the surrogate function still requires the evenly spaced step function, this could be generated using interpolation to find the planes of best fit.

We begin by drawing a Euclidean rectangle that contains the entire corridor. This rectangle is filled by a two-dimensional array, which will be used to form the new terrain model. A first pass is done, scanning each of the original terrain points and putting them into the positions of the two-dimensional array that correspond to the nearest Euclidean coordinates. Next, each position in the two dimensional array is scanned and the plane of best-fit is found to form the piecewise-linear function. If there are no data points near to one of the array entries, an elevation level of infinity can be used to indicate that this is outside of the corridor, and hence an infeasible region of the map.

While this method does require more time to set up the terrain, it should provide a more accurate representation of the terrain than the one presented in Option A.

3.2.2 Converting the Road Alignment

The methods for converting the road alignment coordinates between the terrain models are described in Chapter 3.2.1. In this section we see the simple method in Option A above requires additional work when converting the road alignments’ coordinates between the two models. In comparison, Option B yields a trivial conversion between the two coordinate systems.

**Option A**

To convert a horizontal alignment in the full model to the surrogate, we construct a distance and offset coordinate system in the full model, which translates to \(x\) and \(y\) coordinates in the surrogate. Since the full model uses \(x\) and \(y\) coordinates, rather than distance and offset, it is non-trivial to convert a Euclidean \((x, y)\) coordinate into a distance and offset \((d, o)\) coordinate system.

We begin by forming the set of boxes with smallest area that contain each pair of adjacent, cross-sections for the initial intersection points in the full model. To convert a given point from the full model, we find the boxes which contain the point. This narrows down the distance of the point to being on at most two tangent sections of your road. Using a similar strategy, we can determine which pair of cross-sections (within the box) the point is between. The point can then be projected onto the two nearest cross-sections. We
Interpolate the projected point’s $d$ and $o$ coordinates on each cross section by using the $(d,o)$ coordinates of the two nearest data points on each of the cross sections. Using this information, we can interpolate the final distance and offset coordinates used by the surrogate.

Converting a horizontal alignment in the surrogate model to the full model is comparatively simple. For each plane in the surrogate’s terrain, we keep track of the corresponding point in the original model. We round the $(d,o)$ coordinates of the road to the nearest terrain data points and interpolate.

**Option B**

In this model, the coordinate systems used in the surrogate and the full model are both Euclidean Coordinates. Hence, we do not require any additional techniques to convert the road alignments’ coordinates.
Chapter 4

Numerical Tests

Sample numerical tests were run and are presented in this Chapter. Due to time constraints comparing the solutions returned by the surrogate with the full algorithm is left as future work.

All numerical tests were done using input parameters for the construction costs provided by Softree Technical Systems Inc. The terrain data was acquired from the USGS National Map Viewer [USG] and constructed using a digital elevation model from California. Numerical tests were run on a 64-bit desktop computer running Windows 7 with 32 GB of RAM. The processor used was an Intel(R) Xeon(R) CPU E5-1620 v2 running at 3.70 GHz. The surrogate was prototyped and tested in MATLAB R2014a, using the Optimization Toolbox.

4.1 Comparing the Mass-Haul Algorithm

The results from a sample solution found with two versions of the surrogate give indication for an improvement in the quality of solutions found with the surrogate using the Mass-Haul algorithm. Two versions of the surrogate were compared using Nomad [AAC\(^+\)], a derivative-free optimization solver to find the optimal solutions. The first used the Mass-Haul algorithm, while the other used the original waste cost calculations instead. Both used the relaxed constraint for the circular sections of the road.

The two solutions found with the surrogate are presented in Figure 4.1. Recall that the intersection points of the road are restricted within boxes to capture the corridor-like constraint of the full algorithm. In this case, the boxes are squares with side lengths of 100 meters. From this observation we can see that the optimal road alignments found using the two version of the surrogate function are spatially very different.

In Figure 4.2 we see the plot of earthwork that needs to be done along the road alignments in Figure 4.1. In particular we have labeled three points of note along the road alignments. In the first and third, we see that there are two downward spikes, corresponding to regions of fill in the solution returned by the original surrogate. In the original alignment, we note that
there is nowhere nearby that needs to be cut and hence the material needed to fill Points 1 and 3 must be brought from relatively far away. In the second alignment, which uses the Mass-Haul algorithm, such alignments are penalized. Indeed, we see these downward spikes are no longer present for the new alignment. On the other hand, we note that at Point 2 we see in both alignments a spikes going both up and down. In this case, while the total volume of earthwork required is larger than at Points 1 or 3, we see that the same general trend is still present in the new alignment. The Mass-Haul algorithm does not penalize such alignments as heavily, since the earth in these regions only needs to be moved a short distance to level out the road.

4.2 Pareto Front

Building upon the original surrogate function Hirpa et al. [HHLT14] determined the pareto front between the user costs and the earthwork costs for building new roads. The pareto front is defined to be the set of non-dominated points of a multi-objective function. That is, given a set of candidate road alignments the surrogate function will map these alignment
4.2. Pareto Front

Figure 4.2: The volume of earthwork that needs to be done along the road alignments in Figure 4.1. A positive value indicates a region of cut, while a negative value indicates the region of fill. The original optimal surrogate road alignment is in gray, the alignment found with the surrogate function using the Mass-Haul algorithm is in black.

A point belongs to the set of pareto-optimal points if it is a non-dominated point, i.e., there does not exist an alignment whose cost vector has each component strictly less than that of the non-dominated point.

The original version of the surrogate function returned a vector of two costs, the earthwork costs and the paving costs. Which this was used to determine the pareto front, the paving costs were later relabeled as the user costs. The user costs are determined by the amount of time required to travel along a road and hence, similar to the paving costs, are length dependent.

Using a similar strategy, an updated version of the pareto front has been found using the new surrogate function. Numerical results were done with three different solvers. The first was using a Multi-Objective Genetic Algorithm (MOGA), which is available as a solver in the MATLAB software. The second used a derivative-free solver known as a Direct-Multisearch [CMVV11], or DMS. Finally, a weighted-sum method (WS) was used with the fmincon solver available in MATLAB. Since the fmincon solver optimizes a single-valued objective function, a parameter \( \theta \) was varied from 0 to 1 at intervals of 0.01. The objective value used was then \( cost = \theta cost_{earthwork} + (1 - \theta) cost_{user} \). The MOGA solver used a stopping criterion based on the toler-
4.2. Pareto Front

ance for the changes in the objective function. The DMS and WS methods used as a stopping criterion the maximum number of function evaluations.

Two choices of parameters were chosen for the MOGA and DMS solvers in finding the pareto front. The results are summarized in table 4.1. For more information on the parameter selection the reader is referred to [HHLT14].

Table 4.1: A summary of the parameter choices and results for the solvers used to find the pareto front.

<table>
<thead>
<tr>
<th>Solver</th>
<th>Tolerance</th>
<th>function evaluations</th>
<th>time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOGA</td>
<td>$10^{-4}$</td>
<td>21,833</td>
<td>386</td>
</tr>
<tr>
<td>MOGA</td>
<td>$10^{-8}$</td>
<td>960,115</td>
<td>16,821</td>
</tr>
<tr>
<td>DMS</td>
<td>N/A</td>
<td>24,000</td>
<td>444</td>
</tr>
<tr>
<td>DMS</td>
<td>N/A</td>
<td>800,000</td>
<td>54,306</td>
</tr>
<tr>
<td>WS (100 runs)</td>
<td>N/A</td>
<td>24,000 (each)</td>
<td>8463 (total)</td>
</tr>
</tbody>
</table>

Figure 4.3: The plot of all solutions found by the weighted sum method. The boxed points are the ones that belong to the pareto front. Note the vertical scale does not start at zero and all costs are within 10%.

In Figure 4.3 we present the plot of all of the solutions along with the pareto front found using the weighted sum method. Figure 4.4 contains the pareto fronts found by all three algorithms, including the two parameter
choices for the MOGA and DMS algorithm. We see that MOGA with a tolerance of $10^{-8}$, DMS with 800,000 function evaluations and WS return a relatively similar pareto front, indicating that the methods are converging to the same solution.

![Pareto Fronts Graph]

Figure 4.4: The plot of the pareto fronts found by the various algorithms with their respective parameters.

### 4.3 Cost Parameter Stability

A similar approach to the Weighted Sum method in Chapter 4.2 was used to find the optimal solutions for roads with five different weights to the cost parameters. These solutions were found using NOMAD with the surrogate formulation Pushak-2015. The objective function used for the solutions was similar to the weighted sum method, however only five different values for $\theta$ were used, i.e., 0, 0.25, 0.5, 0.75 and 1.

We see that when only the user costs are considered, the road alignment takes the straightest possible path within the corridor constraints, as expected. The other four alignments show a variety of different solutions, indicating that the exact choice of cost parameters significantly affects the optimal road alignment found.
4.3. Cost Parameter Stability

Figure 4.5: Five different optimal road alignments found with different weights for the cost parameters. The white line represents a value of $\theta = 0$ (all user costs) and the black line represents a value of $\theta = 1$ (all earthwork costs). The three shades of the grey lines represents the values 0.25, 0.5 and 0.75. The $x$ and $y$ axes are in meters, the $z$ axis is in decameters.
Chapter 5

Conclusion

The next step for the surrogate function is more rigorous testing to compare the quality of its optimal solutions to those of the full three-dimensional model. To isolate the components of the surrogate function for comparison, a first test would compare only the vertical alignment solution quality. This could be done by running the full model’s vertical alignment optimization problem, and comparing it to the surrogate’s optimal solution using a fixed horizontal alignment. Such testing would help to reveal the impact of the Mass-Haul algorithm compared to the original design of the surrogate function.

Provided that the surrogate function’s vertical solutions are promising, a second set of tests need to be done comparing the solutions for the full three-dimensional road alignment. The results of which, while perhaps more interesting for the final intent of the surrogate, may prove less telling than the vertical alignment tests. The added complexity and increased number of variables in the full, three-dimensional problem is likely to cause the surrogate function and the full problem to converge to different solutions. For proof of the fact that the function can converge to many different local minima, we only need to consider the example in Chapter 4.3. With this example in mind, it is a likely assumption that the surrogate and full problem will find different local minima. A metric that may prove more telling, is to compare the price of the optimal solutions found with the surrogate and the full algorithm, when both are evaluated using the full model.

We propose an alternative test which may help to determine whether the surrogate captures similar local minima to the full algorithm. The surrogate function could be given the optimal solution found by the full algorithm as its initial alignment. If this alignment were to be significantly altered when optimized by the surrogate function, it would indicate that the two models do not contain the same argminima. If this were to be determined to be the case, the design of the surrogate model would need to be re-examined and improved before integration with the full model.

Assuming further testing yields positive results, there are a few possible directions for the surrogate. Future plans for the full model include
developing a three-dimensional model which simultaneously optimizes the horizontal and vertical alignment. Should this be done, the surrogate function may prove easy to integrate. However, since the surrogate model is able to optimize both the horizontal and vertical components, while the current full model performs these steps separately, combining the two is not as simple. A simple, but likely inefficient option, would be to separate the two problems with the surrogate similar to the full model. First taking a step in the horizontal direction, and then quickly approximating the optimal vertical solution with the surrogate. While this method would prove easiest in implementation, it does not take advantage of the potential available through the surrogate function to simultaneously optimize both the vertical and horizontal alignments.

Alternatively, the reverse type of approach could be done, by simultaneously optimizing the vertical and horizontal alignments with the surrogate function. When the full model is then used as the objective function, the vertical component of the alignment could either be simply ignored, or used as a warm start in solving the vertical problem. In either case, the current iterate’s vertical alignment would not be the one at which the full model’s objective function is evaluated. With this method, should the vertical alignments produce by the surrogate be far from the optimal vertical alignments, then the step with the full model should indicate that the surrogate is not making progress, alerting the algorithm that a drastic step is necessary.

Finally, the precision of the surrogate function could be further improved by evaluating earthwork using a piecewise quadratic vertical alignment instead of a piecewise linear one. While the computation is more complicated, symbolic computation should help and result in a comparable computation time.
Bibliography


Bibliography


Appendix
Appendix A

Source Code

A.1 Mass Haul Algorithm

```matlab
function [cost, volume] = massHaul(volume, distance, clf, clo, cle, chf, cho, che, lf, lo)
%Mass haul function created to better approximate the cost of the earthwork for the surrogate function.

%Author: Yasha Pushak
%Last Edited: April 26, 2015

%volume – An array containing the volume of earthwork being moved at various points along the road. The array is padded with zeros at the end for speed.
%distance – The length of each section corresponding to the volume array. The array is padded with zeros at the end for speed.
%clf – The cost of loading for a Free Haul
%clo – The cost of loading for an Over Haul
%cle – The cost of loading for an End Haul
%chf – The cost of hauling for a Free Haul
%cho – The cost of hauling for an Over Haul
%che – The cost of hauling for and End Haul
%lf – The maximum length for which a Free Haul is cheapest.
%lo – The maximum length for which an Over Haul is cheaper than an End Haul.

%Preprocessing

%Find out how many non-zero entries are in the arrays.
n = sum(distance~=0);

%CumulativeDistance is used in the preprocessing for speed.
cumulativeDistance = zeros(size(distance));
cumulativeDistance(1) = distance(1);
for i = 2:numel(distance)
    cumulativeDistance(i) = cumulativeDistance(i-1) + distance(i);
end

%Preprocess to remove any excess mass. Move the excess mass from the point with the most mass to the nearest pit (either the start of the end).
```
A.1. Mass Haul Algorithm

29 %Find how much we have left over.
excess = sum(volume);
31 %Find where most of the material is coming from.
if(excess > 0)
   maxv = -1;
   argmaxv = -1;
   for i = 1:n
      if(maxv < volume(i))
         maxv = volume(i);
         argmaxv = i;
      end
   end
   argmost = argmaxv;
elseif(excess < 0)
   minv = 1;
   argminv = -1;
   for i = 1:n
      if(minv > volume(i))
         minv = volume(i);
         argminv = i;
      end
   end
   argmost = argminv;
end
volume(argmost) = volume(argmost) - excess;
% We assume we are moving the earth from the middle of the bucket.
haulDist = distance(argmost)/2;
if(cumulativeDistance(argmost) < cumulativeDistance(n) - cumulativeDistance(argmost))
   % The closest pit is as the start.
   for i = 1:argmost-1
      % This loop could be replaced using the information in cumulativeDistance, but since this code is intended to be used later, I have left it this way for readability.
      haulDist = haulDist + distance(i);
   end
else
   % The closest pit is at the end.
   for i = argmost+1:n
      haulDist = haulDist + distance(i);
   end
end
% Find the cost of hauling the earth to the waste pit.
deposit = abs(excess);
cost = haulCost(deposit, haulDist, clf, clo, cle, chf, cho, che, If, lo);
A.1. Mass Haul Algorithm

%The main body of the algorithm.
j = 1;
done = 0;
for i = 1:n
  if (done)
    break;
  end
  while (volume(i) > 0)
    while (volume(j) >= 0)
      j = j + 1;
      if (j == n+1)
        %The volume arrays are padded with extra zeros for
        %efficiency, we want to stop when we have reached the end.
        done = 1;
        break;
      end
    end
  end
  %If j reached the end of the array we are done.
  if (done)
    break;
  end

%Assume we are moving the mass from the center of each
section
haulDist = distance(i)/2 + distance(j)/2;
if (j > i)
  haulDist = haulDist + cumulativeDistance(j-1) -
    cumulativeDistance(i);
else
  haulDist = haulDist + cumulativeDistance(i-1) -
    cumulativeDistance(j);
end
%we can’t deposit more than there is space for, nor can we
deposit more than we are hauling.
deposit = min(abs(volume(j)), abs(volume(i)));
%Move the earth.
volume(i) = volume(i) - deposit;
volume(j) = volume(j) + deposit;
%Find the cost of the movement.
cost = cost + haulCost(deposit, haulDist, clf, clo, cle, chf, cho,
che, If, lo);
end
end

Listing A.1: An Implementation of the Mass Haul algorithm. An emphasis
was made on code readability and documentation for future use.
A.2 Mass Haul Costs

function [cost] = haulCost(deposit, haulDist, clf, clo, cle, chf, cho, che, lf, lo)
%Helper function to calculate the hauling cost given the amount being
%hauling and the different costs for hauling, loading, and their
%corresponding lengths.
%Author: Yasha Pushak
%Last Edited: April 5, 2015
%deposit – The volume of earthwork being moved
%haulDist – The distance the earth is being moved
%clf – The cost of loading for a Free Haul
%clo – The cost of loading for an Over Haul
%cle – The cost of loading for an End Haul
%chf – The cost of hauling for a Free Haul
%cho – The cost of hauling for an Over Haul
%che – The cost of hauling for an End Haul
%lf – The maximum length for which a Free Haul is cheapest.
%lo – The maximum length for which an Over Haul is cheaper than an End Haul.
if (haulDist < lf)
    %Free Haul
    cost = clf*deposit + chf*deposit*haulDist;
elseif (haulDist < lo)
    %Over Haul
    cost = clo*deposit + cho*deposit*haulDist;
else
    %End Haul
    cost = cle*deposit + che*deposit*haulDist;
end
end

Listing A.2: A helper function used by the Mass Haul Algorithm in listing A.1 to determine the costs of hauling some volume of earth over a set distance.