Models and strategies for efficiently optimizing the vertical alignment of roads for multimaterial

by

Shahadat Hossain

B.Sc., Bangladesh University of Engineering and Technology, 2009

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 2013

Abstract

Selecting optimal vertical alignment while satisfying safety and design constraints is an important task during road construction. The amount of earthwork operations depends on the design of a vertical alignment, so a good vertical alignment can have a profound impact on construction costs. In “Optimizing the vertical alignment under earthwork block removal constraints in road construction” M.Sc. thesis 2012, F. Rahman proposed two models for vertical alignment optimization. However, the models were limited to a single material and the volume of material needed to be a convex function of the distance from the ground profile. Consequently, these models cannot be used in commercial road design softwares. In this research, we remove the previous limitations so the models can now be used for road design software. Both models use a piecewise quadratic curve to minimize the cost for earthwork operations. The models consider several new features as well as all the features of the previous models that are described in previous research by F. Rahman. Moreover, we improve the solution time, so that it is possible to make interactive design tools. We provide numerical tests that validate our results.

Preface

Models and strategies for efficiently optimizing the vertical alignment of roads for single material part of this thesis has been adapted from a manuscript co-authored with Dr. Warren Hare, Dr. Yves Lucet, and Faisal Rahman at the University of British Columbia, Okanagan campus. The manuscript has been submitted for publication to the Journal of Computers and Operations Research [HHLR13]. The paper included models and techniques for single material only. A US provisional patent was also filed based on the results contained in the present thesis.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my sincere gratitude to my two supervisors, Dr. Yves Lucet and Dr. Warren Hare, for their continuous help and support in all stages of this research. I would like to thank them for being open and encouraging to new ideas, listening to my ideas, and helping me to organize those in proper way. Their knowledge, guidance, and patience were essential for this thesis. It was a great privilege to have such friendly supervisors that helped me to overcome many challenges and enjoy my graduate life.

A very special thanks goes to our industrial partner (Softree Technical Systems Inc.) for providing us all the help that we need. In particular David Mills, Craig Speirs, and Alexis Guigue were always very supportive and helpful to give technical information for this thesis. The week that I worked with them in Vancouver helped me a lot to organize my thesis.

I would like to thank everyone who directly or indirectly helped me in my research, course works, and in other activities during my graduate life.

I owe my deepest gratitude to my parents, and family members for their prayers and support throughout my life. Without their selfless and continuous help I would not be able to come here.

I also like to thank all of my friends who helped me to pass some good times in Kelowna, took me to camping, and gave me rides to several places.

This research is partially funded by an Collaborative Research and Development (CRD) Grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada, MITACS, and Softree Technical Systems Inc.

Dedication

To Rusdania Bushra Asha.

Chapter 1

Introduction

In this chapter, we describe our motivation for research in road design, define the road design problem, and discuss some previous works in road design.

1.1 Motivation

The state of the road network system is one of the strongest indicator of a country's level of development. So every year all countries spend a significant amount of their budget on construction and maintenance of roads and highways. For example, the Canadian government in 2011 allocated $32,000,000,000 for roadway transportation [Min12a]. A significant amount of this budget is spent on building new roads and maintaining existing roads. The cost of constructing a road without considering the cost of engineering design or property acquisition, can be in the range of $2,100,000 to $5,000,000 per kilometer for two lanes [Min12b] . As a result, the government and the companies who build and maintain roads and highways are interested to find a way to minimize the cost of road construction and maintenance.

While constructing roads and highways, cost minimization is not the only issue the road and highway engineers have to deal with. Every year thousands of road collisions and casualties happen. These accidents cost a lot, not only in terms of money but also in terms of life. So it is very important to consider safety requirements, rules, and regulations while designing roads to minimize accidents. Furthermore, road engineers also need to consider the environmental impacts [Jha03] and the socioeconomic effects [DJ09].

Designing roads or highways to minimize costs during the construction process while satisfying safety requirements, rules, and regulations considering all socioeconomic and environmental impacts is a very difficult task. Thus automating the road design process to the optimal design could save a lot of money. Unfortunately there is no road design software in existence today that has an automated and interactive way for computing optimal design of roads or highways considering multiple materials and all practical design issues. In this research we work to address these issues and take a step towards building a software tool for computing a mathematically proven optimal design of a road.

1.2 Road design problem

Road design refers to the problem of connecting some given end points by selecting an economical alignment while satisfying various design specifications, safety constraints, and considering environmental and socioeconomic impacts [JS03], [JSJ06]. Minimizing the cost of road construction spans a wide range of road types: from highways to mountain roads. It is a sub-problem in forest management [CMW98], and in the design of efficient highway networks [ACRV13].

The problem of designing roads can be split into three interrelated subproblems [AAS04]. First, the horizontal alignment provides the road trajectory from a satellite's eye view considering political and social issues [KSJ07]. Then, the vertical alignment is a modification of the ground profile that minimizes the cost of construction while satisfying safety and regulation constraints [Mor09]. Finally, a solution to the earthwork problem describes how material is rearranged optimally to build the desired vertical alignment at minimal cost [HKL11]. For computational efficiency, many researchers combine the last two stages together to design optimal vertical alignments while minimizing earthwork costs.

Since two given points can be connected in numerous possible ways, selecting the best alignment quickly from many potential alignments is a difficult task. Manual processes of designing roads take hours of skilled labor and there is no guarantee on the optimality of the final design. In this research we propose models and strategies to calculate the vertical alignment (incorporating the earthwork problem) that can be solved by deterministic optimization algorithms.

1.2.1 Horizontal alignment

A horizontal alignment is a route from source to destination; it follows a topographical map and satisfies various design code requirements [AAS04]. It should also consider several socioeconomic and political issues [KSJ07] , e.g., passing through specific town and staying away from hospitals and schools. The cost of designing a horizontal alignment depends on several factors such as land cost, travel cost, and on the vertical alignment cost discussed next. A sample horizontal alignment is shown in Figure 1.1. Horizontal alignment optimization is usually done using one of the three techniques: calculus of variation, dynamic programming, and network optimization.

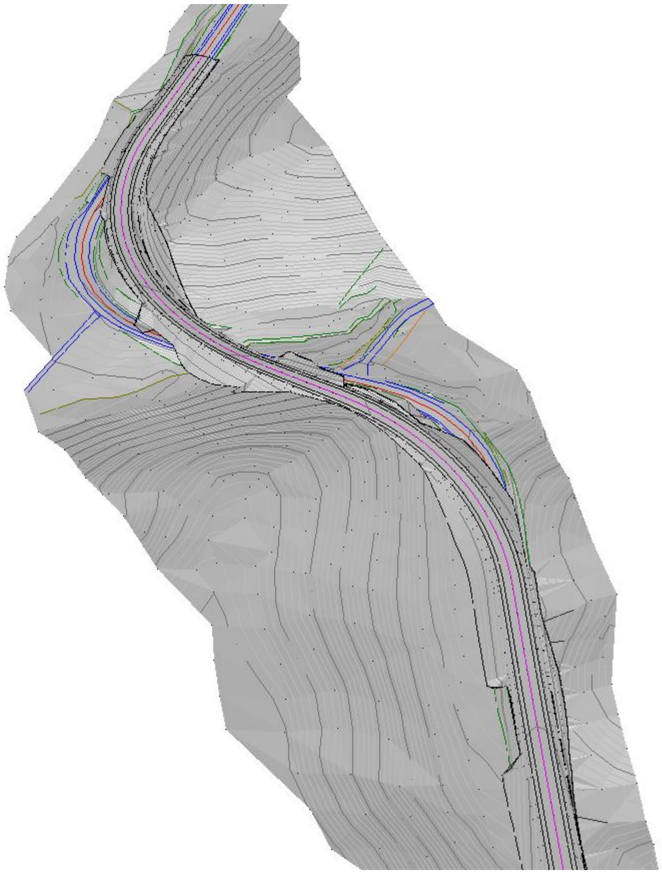


Figure 1.1: Horizontal alignment as visualized in *RoadEng* road design software.

In calculus of variation, one tries to find a curve connecting two end points in space which minimizes the integral of a cost function ([JSJ06], [Wan95]). The horizontal alignment problem is similar to the typical problem in calculus of variation to some extent. This method is used to determine horizontal alignment in [SH81], [SH82], and [HBS68]. This methods needs to determine continuous local cost function. Since the cost depends on various factors, some approximation and assumptions are required to calculate the cost function. However, this method produces a continuous road, and a global optimum is guaranteed.

Dynamic programming is a well known method for optimization. Mostly this technique is used for discrete problems. Few studies ([OEC73], [Tri87b], [JSJ06]) showed the optimization of a horizontal alignment using a dynamic programming method. The problem with this approach is that it takes a long time to solve large problems and the output alignment is non-smooth.

Network optimization method formulate the horizontal alignment problem as a network problem such as a shortest path problem, a min cost flow problem, etc. Some early papers [TM70], [Tur78], [Tri87a] used network optimization methods to optimize a horizontal alignment. This method is simple and easy to understand, but formulating the cost function for each edge of the network considering the cost of vertical alignment is hard and requires significant approximations.

1.2.2 Vertical alignment

After computing a horizontal alignment, one can draw a curve between the two given points and get the vertical ground profile along this curve ([HKL11]). An example of a vertical alignment is shown in Figure 1.2. In vertical alignment optimization, a road profile is fitted to the ground profile minimizing the cost of construction while satisfying safety and regulation constraints. It determines which hill to cut or ravine to fill to construct a road. The costs associated with the vertical alignment are the cut and fill costs of materials. Once a vertical alignment is determined, the earthwork optimization minimizes the cost of moving materials to build the desired vertical alignment.

Vertical alignment optimization [Eas88], [GCF88], [Mor96], [LC01], [FCS02], [MA04], [GLA05], [GAA09], [GLA09], [Mor09], [KL10], [HKL11] of roads has been modeled by several researchers considering many constraints. As the earthwork cost depends on the design of the vertical alignment, it is common to combine the two stages together to design a vertical alignment while considering the earthwork.

Nondeterministic methods to compute a vertical alignment were considered. Lee and Cheng [LC01] proposed a three layered heuristic method for vertical alignment optimization. They used several layers to convert a nonlinear mixed integer linear program to a mixed integer linear program. Their model was nonlinear at first because of the use of crest and sag curve equations to write the constraints on slopes limit. Fwa et al. [FCS02] and Goktepe et al. [GLA09] presented genetic algorithms formulation of the ver-

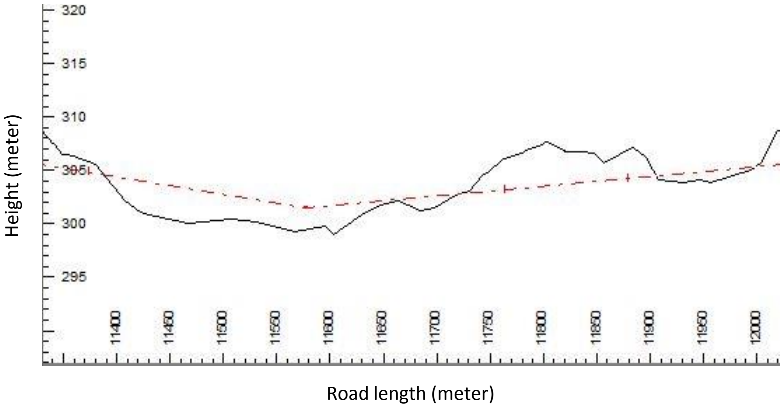


Figure 1.2: Vertical alignment as visualized in *RoadEng* road design software. The solid line is the ground profile and the dotted line is the road profile.

tical alignment problem. A detailed model solved with a genetic algorithm can be found in [JSJ06].

Several deterministic algorithms were proposed that take different constraints into account. Easa [Eas88] proposed a model for vertical alignment considering earthwork costs. It enumerates all possible combinations of feasible grades, and then for each grade it calculates the cut and fill using a linear program to minimize earthwork operations. It is a trial and error procedure. Goh et al. [GCF88] modeled the optimal vertical alignment using dynamic programming considering several costs such as cut, fill, pavement, and land acquisition costs.

Moreb [Mor96] proposed a model combining vertical alignment and earthwork operation together in a single linear program. However, it gives a piecewise linear segments for the vertical alignment, which is not desirable. Moreb and Aljohani [MA04] improves the previous work of Moreb [Mor96] by representing the road profile as a quadratic spline. Goktepe et al. [GLA05] proposed a dynamic programming model to solve the vertical alignment problem and later in [GAA09] they extended the previous model by combining it with the Weighted Ground Line Method for cut-fill balancing and earthwork minimization. Moreb ([Mor09]) also proposed an improved linear program that solved the problem of sharp connectivity of piecewise linear segments by adding more constraints. This model generalized the technique with a piecewise polynomial of any degree. Koch and Lucet ([KL10]) improved Moreb's model ([Mor09]) by changing two constraints to avoid unnecessary errors. They also showed that up to a quadratic spline it is possible to maintain the linear structure of the model. But they did not consider side slopes and blocks into the model.

Hare, Koch, and Lucet ([HKL11]) proposed a mixed integer linear programming model for earthwork optimization considering blocks. They were the first to introduce the block concept during earthwork optimization and determined an optimal schedule for material movement. A block is an obstacle such as mountains, rivers, etc. that needs to be dealt with before materials can be moved over the location of the block. In [HLR13], Hare, Lucet, and Rahman proposed a mixed integer linear programming model for vertical alignment considering side slopes and blocks. Side slopes refer to the gradual increase or decrease of the height from the road profile to the ground profile. When blocks are not considered, Ahuja et al. [AMO93, p. 12-13] gave a network flow model for earthwork optimization.

Rahman [Rah12] proposed two models for vertical alignment while minimizing the earthwork cost. He extended the work of Moreb [Mor09], Koch and Lucet [KL10], and Hare et al. [HKL11]. The two models are called the quadratic model and the quasi network flow model. The quadratic model is a mixed integer linear program, and the quasi network flow model is a modification of a minimum cost network flow problem that turns it into another mixed integer linear program with significantly less variables. These two models are the most recent and constitute the state-of-the-art for solving vertical alignment problems using deterministic optimization algorithms. Sides slopes and blocks are considered in both models but only when a single material is considered, which is not realistic and prevents the model from being able to calculate a good approximation of the real construction cost.

1.2.3 3D alignment

In the ideal case, designing a three dimensional alignment of roads by considering the above mentioned horizontal, vertical alignment, and earthwork sub problems together would be the most efficient strategy. However, calculating 3D alignment of a road, considering all of these simultaneously, is a very difficult problem. Most of the researchers use heuristic based algorithms to calculate 3D alignments.

Chew et al. [CGF89] proposed a three dimensional highway alignment optimization model that determines the horizontal and vertical alignments simultaneously. They applied the concept of optimal control theory and used cubic spline polynomial interpolation to build a smooth vertical alignment. The model is a general constrained non-linear optimization problem and cannot guarantee global optimality. Tat and Tao [TT03] developed a model for 3D alignment using genetic algorithms. It considers horizontal and vertical curvature, and gradient constraints simultaneously using GIS data. It also uses a cost model which depends on location, volume, length, etc. Akay [Aka06] proposed a method to design 3D alignments for forest roads using a combination of simulated annealing algorithm and linear programming. This model considers various soil types along the road and takes into account the safety of the drivers. But it manually selects an initial trial route. Cheng and Lee [CL06] proposed an iterative process splitting the whole process into three stages. In each iteration, a new horizontal alignment is obtained by slightly adjusting the current alignment. A corresponding vertical alignment is determined by solving a series of mixed integer linear program. Aruga [Aru05] uses a tabu search method using a high-resolution digital elevation model and later Aruga et al. [ATSM06] improved the model using Dijkstra's shortest path algorithm and a cubic spline function.

Models of the 3D alignment of highways are subject to nonlinear constraints. They have to deal with multiple design alternatives and various design parameters [GLA05]. As a result, most researchers first determine horizontal alignment and then compute a more precise vertical alignment.

1.3 Our research focus

Our research is focused on vertical alignment optimization for road design. We have already discussed above some research works for solving the vertical alignment problem. Several research works emphasized different issues and tried to model them accordingly. We worked closely with our industry partner Softree Technical Systems Inc.[1] who develops road design software. We have consulted with professional road engineers and learned what needs to be considered for a practical vertical alignment problem. The following requirements were found most relevant for modeling the vertical alignment problem:

- Multi-material: A typical road consists of several materials. The cost of road construction depends on these various types of materials. So we need to build a model which can consider more than one material.

[1] http://www.softree.com

Multi-material can be defined by both different quality of the same material and different materials.

- Side slopes: Side-slopes needs to be considered to obtain an acceptable material volume approximation. Side slopes refer to the gradual increase or decrease of the height from the road profile to the ground profile. Discarding them could introduces an unacceptable error in cost calculation.
- Cut and fill at the same section: Cut and fill refers to the excavation and embankment of materials to make a smooth surface for the road construction. As the earth is uneven in shape, there could be cut and fill at the same section of the same material. Section refers to the smallest unit of a road, and there is no cost for moving materials within it.
- Blocks: There could be blocks such as rivers, rocky mountains that needs to be considered, because it is not possible to move materials across or in between those blocks before we deal with them.

Therefore to make a practical design of a vertical alignment, one has to take all of these constraints along with other common design code constraints [AAS04] into account. In addition, the challenge is to make an interactive software tool for designing optimal vertical alignment. So the model needs to give an accurate solution with acceptable computational time.

Moreb's model ([Mor09]) can be easily extended to accomodate multi-materials. But it does not consider side slopes and blocks. Hare et al. ([HKL11]) introduces blocks for earthwork optimization only. Rahman's models ([Rah12]) incorporate blocks, side slopes. But it is only solvable for single material. It needs the input volume to be convex with respect to the height from the ground level. It also cannot consider cut and fill at the same section: it needs strictly either a cut or a fill at a given section.

In our research, we have improved and combined the works done by Moreb ([Mor09]), Koch and Lucet ([KL10]), Hare et al. ([HKL11]) and Rahman ([Rah12]). We propose two models similar to Rahman's models. Both models incorporates multimaterial, side slopes, cut and fill, and blocks. Our models can also consider convex, and nonconvex input volumes. It can handle cut and fill of the same material at the same section. So we fulfill all requirements for computing a practical vertical alignment. Moreover, we devised some techniques to reduce the solution time in order to work towards the target of developing an interactive design tool.

In Chapter 2, we define the variables for both models, and then we describe and improve the previously designed mixed integer linear programming model for vertical alignment optimization. In Chapter 3, we modify the quasi network flow model for vertical alignment optimization. We describe several techniques to improve the solution time in Chapter 4. In Chapter 5, we test the techniques from Chapter 4, show numerical results and deduce the most efficient techniques. Chapter 6 concludes with directions for future research.

Chapter 2

The quadratic model for vertical alignment

In this chapter we describe the variables, parameters, and sets required to model the vertical alignment of roads. Then we describe the *quadratic model* for vertical alignment optimization for a known horizontal alignment.

2.1 Model variables and parameters

The road profile is designed as a quadratic spline. The idea of approximating the road profile by a spline was described in [MA04] and [Mor09]. Later in [KL10], it was shown that up to a quadratic spline, the linearity of the model can be maintained. In Figure 2.1, a ground profile and a vertical alignment of a sample road are shown.

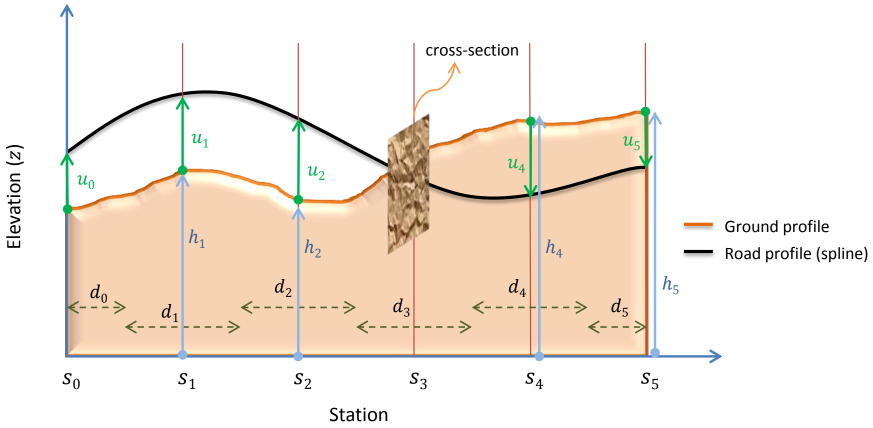


Figure 2.1: Side view of ground profile and road profile. Here, d_i is the effective length of a section, h_i is the ground elevation, u_i is the offset from the ground profile towards the road profile. Cross-section areas are given at each station s_i.

In order to model the vertical alignment, we split the road into several sections. Let $\mathcal{S} = \{1, 2, ..., n\}$ be the index set for the sections. Every section i ($\in \mathcal{S}$) has a station number s_i and an effective length d_i. The difference between a station and a section is that a station is a point, whereas a section is a piece of ground stretched by an associated effective length. At every station, a cross section area of the ground profile is taken. So we can determine the cross section area of each material at each station point. The effective length is multiplied with the cross section area to approximate the volume of materials. A section is associated with the volume of materials, whereas a station is associated with the cross section areas at a particular point.

The height of the ground profile of Section i is defined as h_i. The decision variable u_i is the vertical offset from the ground profile at Section i. We assume that throughout the entire length of the section, the road has a fixed height and a fixed cross section. In reality the heights and cross sections vary throughout the section, so this provides only an approximate value. The approximation error can be reduced by decreasing the section length, i.e., increasing the number of sections. However, if section lengths are small, the number of section increases and the problem size becomes large.

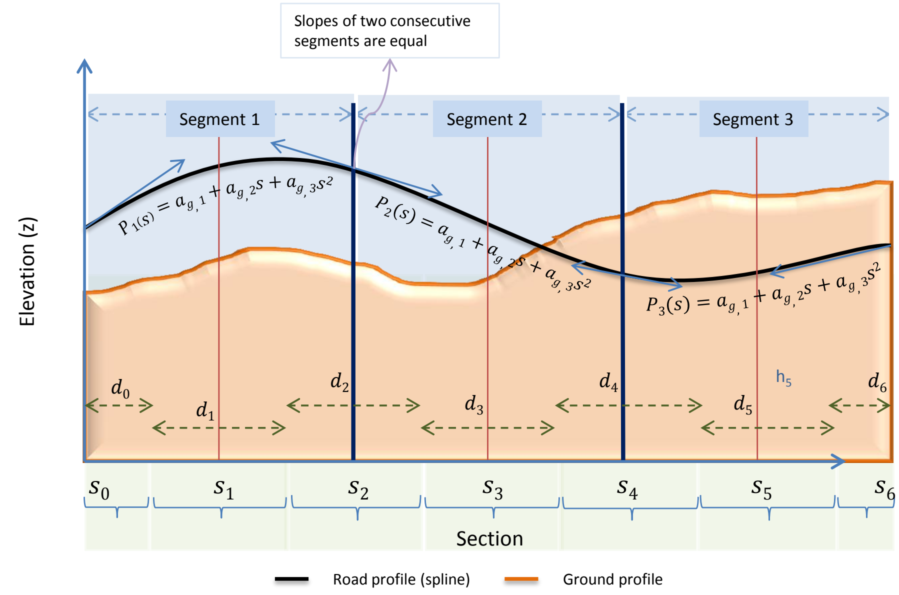
The entire road profile is represented as a quadratic spline (Figure 2.2). The sections are grouped into segments. In every segment there may be several sections, and each segment is associated with a quadratic function. Let $\mathcal{G} = \{1, 2, \ldots, \bar{g}\}$ be the index set of the segments and $\mathcal{S}_g = \{1, 2, ..., n_g\}$ be the index set of n_g sections in spline segment g. The function $\delta : \mathcal{G} \times \mathcal{S}_g \rightarrow \mathcal{S}$ maps the index in the spline segment to the actual section index, i.e., if $\delta(g, j) = i$, then $s_i = s_{\delta(g,j)}$, for all $g \in \mathcal{G}, j \in \mathcal{S}_g$, $i \in S$.

Every two consecutive segments have one common station between them (Figure 2.2). Therefore, the last station of a spline segment is equal to the first station of the next spline segment, i.e., $s_{\delta(g,n_g)} = s_{\delta(g+1,1)}, \forall g \in \{1, \ldots, \bar{g} - 1\}$. The quadratic function for each segment $g \in \mathcal{G}$ is defined as

$$P_g(s) = a_{g,1} + a_{g,2}s + a_{g,3}s^2,$$

where s is the distance from the current station to the beginning of the road along the alignment. The derivative of $P_g(s)$ is written

$$P_g'(s) = a_{g,2} + 2a_{g,3}s.$$



Slopes of two consecutive segments are equal
Segment 1
Segment 2
Segment 3
Elevation (z)
$P_1(s) = a_{g,1} + a_{g,2}s + a_{g,3}s^2$
$P_2(s) = a_{g,1} + a_{g,2}s + a_{g,3}s^2$
$P_3(s) = a_{g,1} + a_{g,2}s + a_{g,3}s^2$
d_0
d_1
d_2
d_3
d_4
d_5
h_5
d_6
s_0
s_1
s_2
s_3
s_4
s_5
s_6
Section
Road profile (spline)
Ground profile

Figure 2.2: Spline segment of vertical alignment.

In general, we can represent the spline function as follows

$$P(s) = \begin{cases} P_1(s) & \text{if } s_{\delta(1,1)} \leq s \leq s_{\delta(1,n_1)}, \\ P_2(s) & \text{if } s_{\delta(2,1)} \leq s \leq s_{\delta(2,n_2)}, \\ \vdots & \\ P_{\bar{g}}(s) & \text{if } s_{\delta(\bar{g},1)} \leq s \leq s_{\delta(1,n_{\bar{g}})}. \end{cases}$$

Similarly, we will use the notation $P'(s)$ to represent the derivative of $P(s)$. Since every two adjacent sections have one common section between them, to get a smooth curve, the height and slope of the beginning section of one segment should be equal to the height and slope of the ending section of the previous segment.

Remark 2.1. For a long road, the value of s in $P_g(s)$ could be large. Since $P_g(s)$ has a square term of s, numerical errors can happen for large values of s. To eliminate these numerical errors from the implementation of the models, each value of s is calculated with respect to the beginning of its segment, i.e., $s - s_{\delta(g,1)} \forall g \in \mathcal{G}$. Since this does not change the models, for the ease of describing the models, we use s instead of $s - s_{\delta(g,1)}$ in $P_g(s)$.

Side-slope refers to the gradual decrease and increase in height from the road profile to the ground profile of a cut and fill section, respectively. Side-slopes are very important from the civil engineering point of view because they help to make the road durable. Therefore, to model the vertical alignment problem accurately, we must take side-slopes into consideration. In Figures 2.3 and 2.4 the cross-section of a road with and without side-slopes for a cut and a fill are shown, respectively.

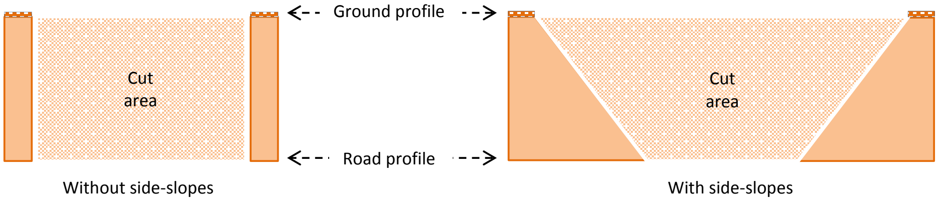


Figure 2.3: The cross-section of a road with and without side-slopes (for cut).

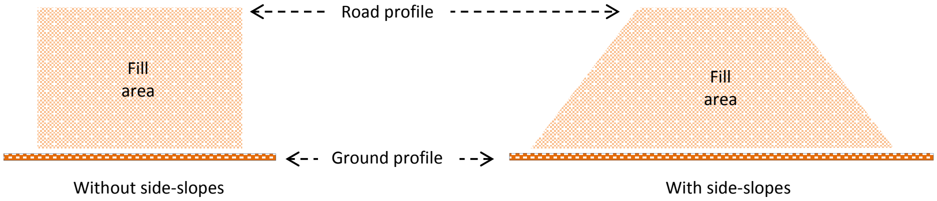


Figure 2.4: The cross-section of a road with and without side-slopes (for fill).

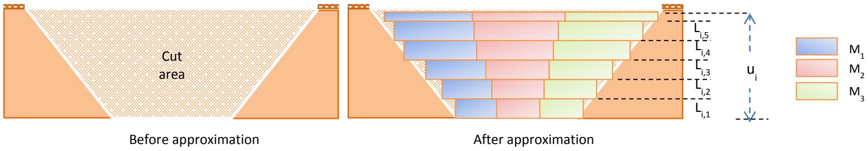


Figure 2.5: Approximation of side-slopes (for a cut). Here, M_1, M_2, M_3 are three kinds of arbitrary materials shown for simple illustration.

To model the side slopes, the approximation can be made with trapezoid shaped cross-sections with several rectangles (Figure 2.5). We make an approximation of the volumes of material at different offset levels from the ground profile. The volumes of material for cut and fill at every section are

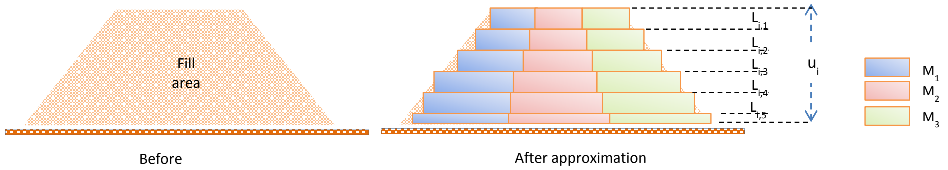


Figure 2.6: Approximation of side-slopes (for a fill).

the input data for the model. These volumes are approximated for each offset level from the ground profile. Suppose for each section $i \in \mathcal{S}$, $\mathcal{L}_i = \{1, 2, ..., n_l\}$ is the index set for the different offset levels. The parameter $L_{i,l} \in \mathbb{R}$ represents the altitude of cut or fill for level $l \in \mathcal{L}_i$ at section i. Let $\mathcal{K} = \{1, 2, ..., m\}$ be the index set of the material types. Sample input for a section is shown in Table 2.1. Let the parameter $R^+_{i,l,m}$ represent the given cut volume and $R^-_{i,l,m}$ represent the given fill volume of material, respectively, for each section $i \in \mathcal{S}$, each level $l \in \mathcal{L}_i$, and for each material type $m \in \mathcal{K}$.

Table 2.1: Cut and fill volume of material

Level l	Offset $L_{i,l}$	Cut Volume $R^+_{i,l,m}$	Fill Volume $R^-_{i,l,m}$
1	-4	262.06	0
2	-2	168.44	0
3	0	82.82	0
4	2	5.28	0.08
5	4	0	70.06

The relation between cut and fill volume with the height of the offset can be understood clearly from Figure 2.7. For the vertical alignment problem, we need to find out the optimal offset (u_i) from the ground profile and the corresponding cut and fill volume for that offset. Let the variable $V^+_{i,m}$ ($i \in \mathcal{S}$, $m \in \mathcal{K}$), and $V^-_{i,m}$ ($i \in \mathcal{S}$, $m \in \mathcal{K}$) represent the total cut and fill volume of material, respectively.

During the construction of a road, some materials may need to be dumped, and some materials may need to be brought in. Borrow pits are the external sections from which materials can be borrowed, and waste pits are the external sections in which materials can be dumped. Suppose $\mathcal{B} = \{1, 2, ..., n_\beta\}$ and $\mathcal{W} = \{1, 2, ..., n_\omega\}$ are the two sets of indices that are

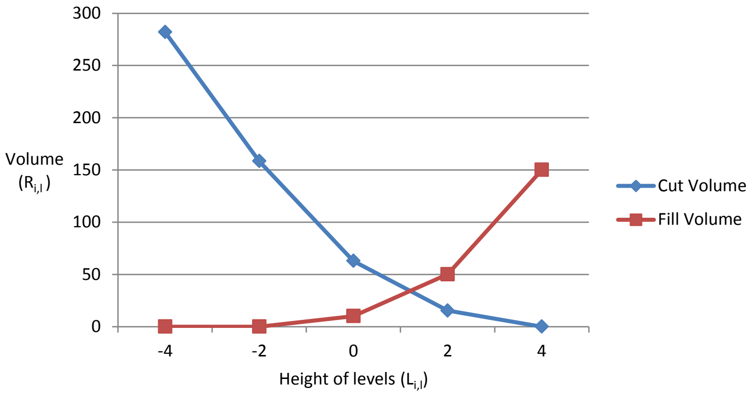


Figure 2.7: Volume approximation of side-slope

used to index borrow and waste pits, respectively. The function $\vartheta : \mathcal{B} \rightarrow \mathcal{S}$ maps a borrow pit index to the section index to which it is attached, and the function $\varphi : \mathcal{W} \rightarrow \mathcal{S}$ maps the same section for a waste pit. There are also access roads from which the construction of the road can be initiated. These access roads are also attached to a section and act as a gateway to the road. Let $\mathcal{R} = \{1, 2, ..., n_r\}$ be the set of indices that is used to index the access roads. We assume that there is a borrow and a waste pit attached to each access road with infinite capacity for dumping and borrowing materials. Suppose the function $\varrho : \mathcal{R} \rightarrow \mathcal{S}$ maps an access road index to a section index to which it is attached, and the set $\mathcal{N} = \{\mathcal{S} \cup \mathcal{B} \cup \mathcal{W}\}$ is used to represent sections, borrow pits, and waste pits collectively.

In practice, it happens that some sections of land cannot be accessed before removing the obstacles. In [HKL11], the block concept was first introduced for earthwork operation in road design. We use this concept for vertical alignment modeling of road design. The block scenario is similar to the situation where engineers have to build a bridge or tunnel before they can move across a river or through a mountain, respectively. Suppose $\mathcal{I} = \{1, 2, ..., n_b\}$ is the set of indices that is used to index blocks. We define time steps t to indicate when blocks are removed. It can be shown that at most $(n_b + 1)$ number of time steps are needed to remove all the blocks [HKL11]. Let $\mathcal{T} = \{0, 1, 2, ..., n_b\}$ be the set to represent $n_b + 1$ number of time steps needed to remove n_b blocks, and let the function $\varsigma : \mathcal{I} \rightarrow \mathcal{S}$ map a block index to a section index. To keep track of when a block is removed, we introduce binary variables y_{kt} for each block $k \in \mathcal{I}$ and $t \in \mathcal{T}$. If a block is removed at time step t, then movement across the block is possible from the next time step $t+1$.

2.2 The quadratic model

In this section the *quadratic model* is described for obtaining an optimal vertical alignment. This model is referred to as the quadratic model because the required number of material movement variables is $\mathcal{O}(n^2)$, where n is the number of sections.

2.2.1 The quadratic model for single material

In [Rah12], the quadratic model for single material was described. That model had several features, for example, it incorporated blocks and sides slopes into the vertical alignment optimization. However, it had several limitations. That model was not capable to incorporate more than one material. The input volumes needed to be convex with respect to the height from the ground level. It could not consider cut and fill at the same section.

The reason of not being able to incorporate all of the above features into the model was the way it defined the volume constraints. It required that volume for cut and fill must be zero at the ground level, which is not the case in practice because of the uneven shape of the earth. Therefore, [Rah12] approximated the input to apply their models.

In this research, we change the volume constraint to accommodate all the features into a new modified model which is described in the next section.

2.2.2 The quadratic model for multimaterial

In this section we describe the quadratic model for multimaterial by modifying the volume constraint. We also incorporate all the features that are required. For ease of explanation, first we introduce variables for material movement without considering blocks. Later we will introduce movement variables considering blocks. To track material movement, we define variable x_{ij}, which represents the amount of material moved from section i to section j. For each $i \in \mathcal{N}$, the index set $\mathcal{N}^i_{\rightarrow}$ consists of all indices j such that x_{ij} is a permitted move,

$$\mathcal{N}^i_{\rightarrow} = \left\{ j \ : \ \begin{array}{ll} j \in S \cup \mathcal{W}, & \text{if } i \in \mathcal{S} \\ j \in \mathcal{S} & \text{if } i \in \mathcal{B} \\ \emptyset & \text{if } i \in \mathcal{W} \end{array} \right\}.$$

Similarly, for each $j \in \mathcal{N}$ the index set $\mathcal{N}^j_{\leftarrow}$ which consists of all indices i such that x_{ij} is a permitted move,

$$\mathcal{N}^j_{\leftarrow} = \left\{ i \;:\; \begin{array}{ll} i \in \mathcal{S} \cup \mathcal{B}, & \text{if } j \in \mathcal{S} \\ \emptyset & \text{if } j \in \mathcal{B} \\ i \in \mathcal{S} & \text{if } j \in \mathcal{W} \end{array} \right\}.$$

Note that for any $i, j \in \mathcal{N}$, $j \in \mathcal{N}^i_{\rightarrow}$ if, and only if, $i \in \mathcal{N}^j_{\leftarrow}$. Finally, $\mathcal{N}^2$ consists of all index pairs (i, j) such that x_{ij} is a permitted move,

$$\mathcal{N}^2 = \{(i, j) \;:\; j \in \mathcal{N}^i_{\rightarrow}\}$$

Note that $\mathcal{N}^2 \neq \mathcal{N} \times \mathcal{N}$, as $\mathcal{N}^2$ omits index pairs (i, j) that correspond to illogical moves, such as movement from a road section into a borrow pit.

Now we introduce variables for material movements considering blocks. For tracking material movement at every time step, we define the variable x_{ijt}, which represents the amount of material moved from section i to section j at time step t. Let $\bar{\mathcal{I}}^2 = \{(k_1, k_2) \in \mathcal{I} \times \mathcal{I} : k_1 < k_2$ and k_1, k_2 are two consecutive blocks with no access road in between$\}$, $\bar{\mathcal{I}}_{\leftarrow} = \{k \in \mathcal{I}$: the set of blocks k before which there is no access road$\}$, and $\bar{\mathcal{I}}_{\rightarrow} = \{k \in \mathcal{I}$: the set of blocks k after which there is no access road$\}$.

For all $k \in \mathcal{I}$, the set $\mathcal{N}^k_b$ consists of all pairs of sections $(i, j) \in \mathcal{N}^2$, such that movement between section i and j requires the removal of block k. The set $\mathcal{N}^{k1,k2}_b$, for all $(k_1, k_2) \in \bar{\mathcal{I}}^2$, consists of all pairs of sections $(i, j) \in \mathcal{N}^2$, such that sections i and j are between blocks k_1 and k_2. For all $k \in \bar{\mathcal{I}}_{\leftarrow}$, we set $\mathcal{N}^{\leftarrow,k}_b = \{(i, j) \in \mathcal{N}^2$: section i and j are before block $k\}$, and for all $k \in \bar{\mathcal{I}}_{\rightarrow}$, we define $\mathcal{N}^{k,\rightarrow}_b = \{(i, j) \in \mathcal{N}^2$: section i and j are after block $k\}$.

The cost of a vertical alignment depends on the embankment, excavation and hauling of materials from section to section. Let p_i represent the excavation cost per unit of volume and q_i represent the embankment cost per unit of volume for section $i (\in \mathcal{N})$. Let r_{ij} be the hauling cost per unit of volume from section i to section j.

The objective of the model is to minimize the total excavation costs, embankment costs, and hauling costs for moving materials between sections, borrow pits, and waste pits over all time steps. The objective function can be represented as follows,

$$\min \left(\sum_{\substack{i \in \mathcal{S} \cup \mathcal{B} \\ m \in \mathcal{K}}} p_i V^+_{i,m} + \sum_{\substack{i \in \mathcal{S} \cup \mathcal{W} \\ m \in \mathcal{K}}} q_i V^-_{i,m} + \sum_{\substack{(i,j) \in \mathcal{N}^2 \\ t \in \mathcal{T}}} r_{ij} x_{ijt} \right),$$

where $V^{+}_{i,m}$ (for $i \in \mathcal{S} \cup \mathcal{B}$, $m \in \mathcal{K}$), and $V^{-}_{i,m}$ (for $i \in \mathcal{S} \cup \mathcal{W}$, $m \in \mathcal{K}$) represent the total cut and fill volume of material respectively.

The total volume of material moved from a section to other sections at all time steps must be equal to the cut volume of that section (Constraint (2.1) below). Similarly, the total volume of material moved to a section from other sections at all time steps must be equal to the fill volume of that section (Constraint (2.2) below). We call these the balance constraints:

$$\sum_{\substack{j \in \mathcal{N}^{i}_{\rightarrow} \\ t \in \mathcal{T} \\ m \in \mathcal{K}}} x_{ijt,m} = V^{+}_{i,m}, \qquad \forall\, i \in \mathcal{S} \cup \mathcal{B}, m \in \mathcal{K}. \tag{2.1}$$

$$\sum_{\substack{j \in \mathcal{N}^{i}_{\leftarrow} \\ t \in \mathcal{T} \\ m \in \mathcal{K}}} x_{jit} = V^{-}_{i,m}, \qquad \forall\, i \in \mathcal{S} \cup \mathcal{W}, m \in \mathcal{K}. \tag{2.2}$$

The offset value u_i is the difference between the road profile, which is represented by the quadratic spline $P(s_i)$, and the given ground profile (h_i) (Constraint (2.3)).

$$P(s_i) - h_i = u_i, \qquad \forall i \in \mathcal{S} \tag{2.3}$$

There could be a cut and fill of the same material at the same section according to the input volume table (see Table 2.1). Usually the cut volume decreases as the height of the offset level $L_{i,l}$ increases (see Table 2.1); and the fill volume increases along with the height of the offset level. That means the relationship between volume and offset level is convex. In general, if we have only a convex relationship for piecewise approximation in linear programming and we need minimization, then we can model that relationship without using binary variables (Constraints (2.4) and (2.5)).

$$V^{+}_{i,m} \geq R^{+}_{i,l,m} + \frac{R^{+}_{i,l+1} - R^{+}_{i,l,m}}{L_{i,l+1} - L_{i,l}}(u_i - L_{i,l}), \quad \forall i \in \mathcal{S}, l \in \mathcal{L}_i \setminus \{n_l\}, m \in \mathcal{K} \tag{2.4}$$

$$V^{-}_{i,m} \geq R^{-}_{i,l,m} + \frac{R^{-}_{i,l+1} - R^{-}_{i,l,m}}{L_{i,l+1} - L_{i,l}}(u_i - L_{i,l}), \quad \forall i \in \mathcal{S}, l \in \mathcal{L}_i \setminus \{n_l\}, m \in \mathcal{K}. \tag{2.5}$$

However, in the vertical alignment model, we are not minimizing only the cut and fill cost of material, but also minimizing the movement cost of material. As a result, if the cost of movement becomes larger than the cost of cut and fill, a stockpiling pattern will appear. Therefore, at certain offset level u_i, the cut and fill volume could be larger than what we expect from the input volume table (Table 2.1). These extra volume of materials would cause stockpiling on the road.

For example, in Figure 2.8, a stockpiling scenario is shown. There is a waste pit attached with Road Section 3. The dead haul distance of the waste pit from Road Section 3 is $10m$. Suppose the offset heights of the Road Section 1, Section 2, and Section 3 are 0, 2 and 0 m respectively. The cut volumes of these three sections are 82.82 m^3, 5.28 m^3, and 82.82 m^3, and the fill volumes are 0 m^3, 0.08 m^3 and 0 m^3 respectively. In this instance, Road Section 2 needs a small amount of fill. So all the cut volumes must be hauled and dumped to the waste pit. However, if the fill cost is less than the hauling cost of material to the waste pit, then instead of carrying the material to the waste pit, the material will be dumped to a nearby station. In Figure 2.8, materials are dumped from Road Section 1 to Road Section 2, and from Road Section 2 to Road Section 1, causing the fill volume to get large. These dumpings of material to nearby road sections are valid according to Constraints 2.4 and 2.5, because the constraints only ensure the lower bound of cut and fill.

Therefore, we need to introduce binary variables to ensure that we get the correct cut and fill volumes at optimal offset u_i for each section $i \in \mathcal{S}$ (Constraints (2.6) and (2.7)). The binary variable $\nu_{i,l}$ ($\forall i \in \mathcal{S}, l \in \mathcal{L}_i \setminus n_l$) is used to indicate which interval (between $L_{i,l}$ and $L_{i,l+1}$) of the offset levels is selected. The parameter M_i is the largest volume possible to be cut or filled at Section i. But for simplicity, we use M everywhere instead of M_i. At any section, only one of the interval offsets can be selected (Constraint (2.8)).

$$V^+_{i,m} \leq R^+_{i,l,m} + \frac{R^+_{i,l+1} - R^+_{i,l,m}}{L_{i,l+1} - L_{i,l}}(u_i - L_{i,l}) + M(1 - \nu_{i,l}), \qquad \forall i \in \mathcal{S}, l \in \mathcal{L}_i \setminus \{n_l\}, m \in \mathcal{K}, \quad (2.6)$$

$$V^-_{i,m} \leq R^-_{i,l,m} + \frac{R^-_{i,l+1} - R^-_{i,l,m}}{L_{i,l+1} - L_{i,l}}(u_i - L_{i,l}) + M(1 - \nu_{i,l}), \qquad \forall i \in \mathcal{S}, l \in \mathcal{L}_i \setminus \{n_l\}, , m \in \mathcal{K}, \quad (2.7)$$

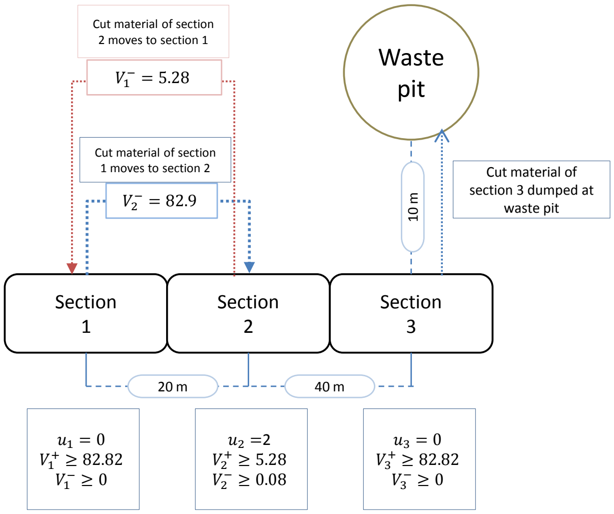


Figure 2.8: Stockpiling scenario when Constraints (2.6) and (2.7) are not considered.

$$\sum_{l \in \mathcal{L}_i \setminus \{n_l\}} \nu_{i,l} = 1, \qquad \forall i \in \mathcal{S}. \tag{2.8}$$

If there is a block between two sections, then material movement is not possible between those two sections until the block is removed (Constraint (2.9)). If there is no access road before a block, then material cannot be moved among sections before that block until the block is removed (Constraint (2.10)). Similarly, material movement is not possible if there is no access road after a block (Constraint (2.11)). Furthermore, if there is no access road between two blocks, then material movement is not possible in between those two blocks until one of them is removed (Constraint (2.12)).

$$\begin{aligned}
x_{ijt,m} &\leq M y_{k,(t-1)}, && \forall t \in \mathcal{T} \setminus \{0\}, k \in \mathcal{I}, \\
& && (i,j) \in \mathcal{N}_b^k, m \in \mathcal{K}, && (2.9) \\
x_{ijt,m} &\leq M y_{k,(t-1)}, && \forall t \in \mathcal{T} \setminus \{0\}, k \in \bar{\mathcal{I}}_{\leftarrow}, \\
& && i, j \in \mathcal{N}_b^{\leftarrow,k}, m \in \mathcal{K}, && (2.10) \\
x_{ijt,m} &\leq M y_{k,(t-1)}, && \forall t \in \mathcal{T} \setminus \{0\}, k \in \bar{\mathcal{I}}_{\rightarrow}, \\
& && i, j \in \mathcal{N}_b^{k,\rightarrow}, m \in \mathcal{K}, && (2.11)
\end{aligned}$$

$$x_{ijt,m} \leq My_{k_1,(t-1)} + My_{k_2,(t-1)}, \qquad \forall t \in \mathcal{T} \setminus \{0\}, (k_1, k_2) \in \bar{\mathcal{I}}^2,$$
$$i, j \in \mathcal{N}_b^{k1,k2}, m \in \mathcal{K}. \tag{2.12}$$

The block removal indicator constraints (2.13, 2.14) ensure that when the required amount of materials are cut and filled for any section containing a block, then the block is considered removed from that section.

$$\sum_{t=0}^{u} \sum_{j \in \mathcal{N}_{\rightarrow}^{\varsigma(k)}} x_{\varsigma(k)jt,m} + M(1 - y_{ku}) \geq V^{+}_{\varsigma(k),m}, \qquad \forall k \in \mathcal{I}, u \in T, m \in \mathcal{K}, \tag{2.13}$$

$$\sum_{t=0}^{u} \sum_{j \in \mathcal{N}_{\leftarrow}^{\varsigma(k)}} x_{j\varsigma(k)t,m} + M(1 - y_{ku}) \geq V^{-}_{\varsigma(k),m}, \qquad \forall k \in \mathcal{I}, u \in T, m \in \mathcal{K}. \tag{2.14}$$

By the end of each time step $t \in \mathcal{T}$, we want to remove at least the number of blocks equal to the number of time steps, so that we do not need more than $n_b + 1$ time steps. This is guaranteed with block removal enforcement constraint (2.15). The monotonicity constraint (2.16), states that once a block is removed it remains removed.

$$\sum_{t=0}^{u} \sum_{k \in \mathcal{I}} y_{kt} \geq t + 1, \qquad \forall u \in \mathcal{T} \setminus \{n_b\}, \tag{2.15}$$

$$y_{k,t} \geq y_{k,t-1}, \qquad \forall k \in \mathcal{I}, t \in \mathcal{T} \setminus \{0\}. \tag{2.16}$$

The smoothness constraints (2.17) and (2.18) ensure that the transitions from one spline segment to the next are smooth, i.e., same grade and elevation (see Figure 2.2).

$$P_{g-1}(s_{\delta(g,1)}) = P_g(s_{\delta(g,1)}), \qquad \forall g \in \mathcal{G} \setminus \{1\}, \tag{2.17}$$

$$P'_{g-1}(s_{\delta(g,1)}) = P'_g(s_{\delta(g,1)}) \qquad \forall g \in \mathcal{G} \setminus \{1\}. \tag{2.18}$$

The maximum allowable speed depends on the grade of the road profile. The grade constraint (2.19) makes sure that the grade of the spline segments are within the allowed maximum (G_U) and the minimum (G_L) value, so that it can maintain the safety parameters.

$$G_L \leq P'_g(s_{\delta(g,1)}) \leq G_U, \qquad \forall g \in \mathcal{G} \setminus \{1\}. \tag{2.19}$$

Usually, the elevation and the grade of the starting point and the ending point of a road are defined explicitly. In addition to those points, the road designer can set the elevation and grade value for some fixed control points on the road (for example at an intersection). Let $\mathcal{H}$ be the index set of all control points. The fixed point constraints (2.20), (2.21), (2.22) set the grade and elevation on specific control points including the starting and the ending point. The parameters y_A, y_B are the elevations of the starting and the ending stations. Similarly, $\bar{y}_A, \bar{y}_B$ are the slopes of the starting and the ending spline segments. The parameters $\mathcal{H}_i$ is the elevation of the station $i \in \mathcal{H}$.

$$P(s_1) = y_A, \qquad P(s_n) = y_B, \tag{2.20}$$

$$P'(s_1) = \bar{y}_A, \qquad P'(s_n) = \bar{y}_B, \tag{2.21}$$

$$P(s_i) = \mathcal{H}_i, \qquad \forall i \in \mathcal{H}. \tag{2.22}$$

Finally, Constraints (2.23), (2.24), and (2.25) are bound constraints. The parameter M_i^+, and M_i^- are the largest volume possible to be cut and filled from a section.

$$x_{ijt,m} \geq 0, \qquad \forall (i,j) \in \mathcal{N}^2, t \in \mathcal{T}, m \in \mathcal{K} \tag{2.23}$$

$$0 \leq V_{i,m}^+ \leq M_i^+, \qquad \forall i \in \mathcal{S} \cup \mathcal{B}, m \in \mathcal{K}, \tag{2.24}$$

$$0 \leq V_{i,m}^- \leq M_i^-, \qquad \forall i \in \mathcal{S} \cup \mathcal{W}, m \in \mathcal{K}. \tag{2.25}$$

In this chapter we have discussed the quadratic model for calculating optimal vertical alignment of roads. We have considered several features such as multimaterials, cut and fill at the same section of the same material, blocks, slide slopes, etc. However, the problem size increases quadratically with the number of sections. Consequently it takes a long time to solve the model for long roads. Another limitation is that we still need convex relationship between volumes of materials and the offset value. In Chapter 3, we discuss the quasi network flow model that reduces the size of the problem, and the computation time.

Chapter 3

Quasi network flow model for vertical alignment

In this chapter the *quasi network flow* model is described for computing an optimal vertical alignment. This model is a modification of the traditional network flow model [AMO93, p. 12-13]. It combines the ideas of network flow and scheduling models. This model schedules the movement of materials among different sections in discrete time steps because of the presence of blocks in roads. In this model, the required number of material movement variable is $\mathcal{O}(n)$, where n is the number of sections.

3.1 Quasi network flow for single material

In [AMO93, p. 12-13], a network flow model is described for earthwork optimization. Rahman [Rah12] modeled the vertical alignment problem by modifying the network flow model and considering blocks and side slopes. Same as the quadratic model described in the previous chapter, this quasi network flow model has the same limitation. However, by modifying the volume constraint and without altering the input volume of materials we are able to use it for multimaterial.

3.2 Quasi network flow for multimaterial

In the quasi network flow model, sections and pits are considered as nodes, and the feasible moves are represented as arcs. This quasi network flow model consists of a series of nodes, which are connected to their adjacent nodes (Figure 3.1).

The materials can flow from section to section towards the East and West directions. In every section we have to decide whether we want to fill the section with materials or transfer it to the next section, or cut materials to add more to the flow. To represent these flows towards the East and West directions, at each section virtual nodes are introduced called transit

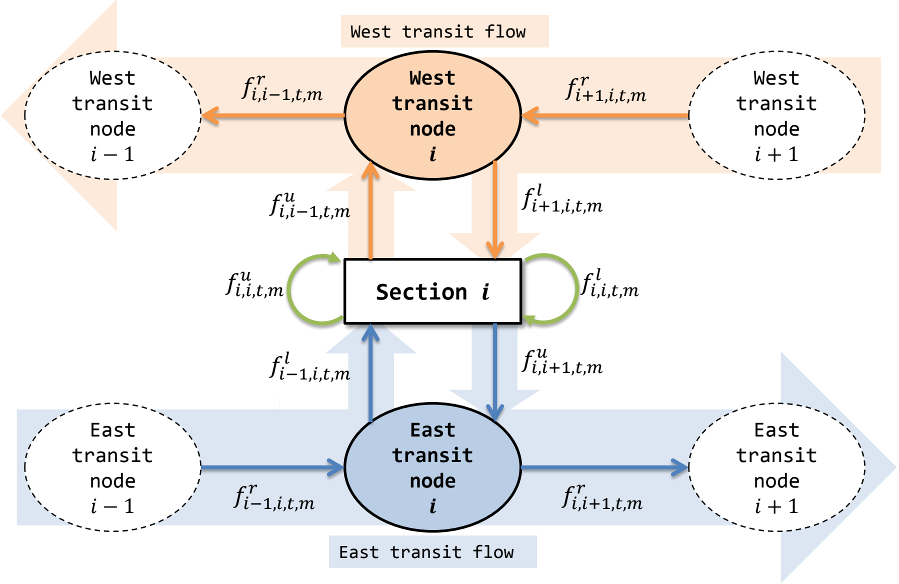


Figure 3.1: A typical section i at time-step t.

nodes for both directions (Figure 3.1). If materials moved from one section to another towards the *West* direction, then the material goes through the West transit nodes, and this flow of material is referred as West transit flow. Similarly, the flow towards the *East* direction goes through East transit nodes, and is referred as East transit flow. Let the variables $f^r_{i,i-1,t,m}$ and $f^r_{i,i+1,t,m}$ $(\forall i \in \mathcal{S}, t \in \mathcal{T}, m \in \mathcal{K})$ be the flow of materials from one transit node to the next in the West and East directions, respectively.

To represent the cut and fill volume of materials at each section, unload and load flows are introduced for each section. Note that, cut and fill can happen within the same section. The cut materials can be transferred to the East and West transit nodes, and within the same section. Similarly, the fill materials can be brought from the East and West transit nodes, and within the same section. Let the variables $f^l_{i-1,i,t,m}$, $f^l_{i+1,i,t,m}$, and $f^l_{i,i,t,m}$ $(\forall i \in \mathcal{S}, t \in \mathcal{T}, m \in \mathcal{K})$ represent the load (fill) flows of materials to the East and West transit nodes, and within the same section, respectively. Similarly, the variables $f^u_{i,i-1,t,m}$, $f^u_{i,i+1,t,m}$, and $f^u_{i,i,t,m}$ $(\forall i \in \mathcal{S}, t \in \mathcal{T}, m \in \mathcal{K})$ represent the unload (cut) flows of materials from the West and East transit nodes, and within the same section, respectively.

The borrow and waste pits are attached to their associated sections.

Each borrow pit can only have unload flows to both transit nodes. These unload flows represent the cut volume from the borrow pit. The flows from borrow pits are called borrow flows. Let the variables $f^b_{j,\vartheta(j)-1,t}$ and $f^b_{j,\vartheta(j)+1,t,m}$ $(\forall j \in \mathcal{B}, t \in \mathcal{T}, m \in \mathcal{K})$ be the flows of materials to the West and East transit nodes, respectively, from the borrow pit. Similarly, for each waste pit there are load flows from both directions. These load flows represent the fill volume of the waste pit and are called waste flows. Let the variables $f^w_{\varphi(j)-1,j,t,m}$ and $f^w_{\varphi(j)+1,j,t,m}$ $(\forall j \in \mathcal{W}, t \in \mathcal{T}, m \in \mathcal{K})$ be the flows of materials from the East and West transit nodes to waste pit, respectively.

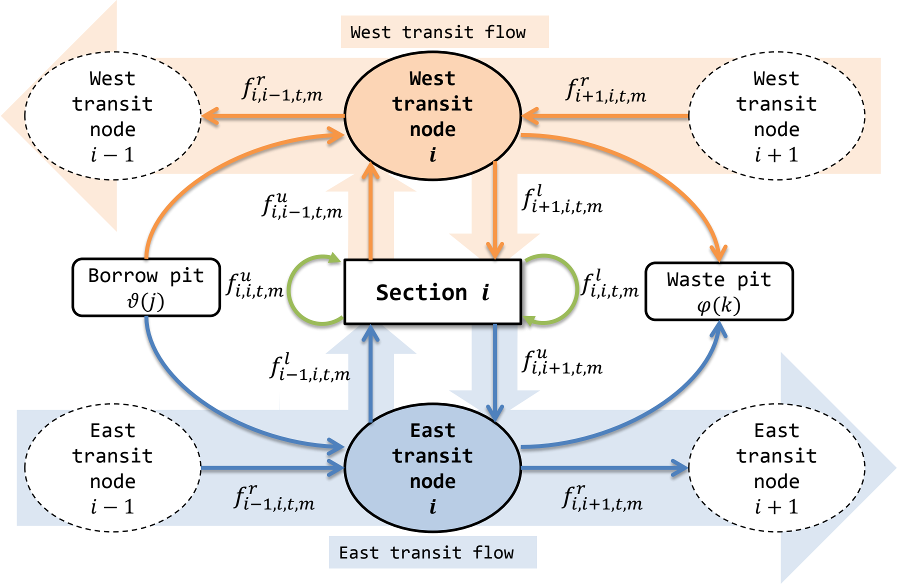


Figure 3.2: A typical section i at time-step t with borrow and waste pit associated with it (where $\vartheta(j) = \varphi(k) = i$). The function $\vartheta : \mathcal{B} \to \mathcal{S}$ maps a borrow pit index to the section index to which it is attached, and the function $\varphi : \mathcal{W} \to \mathcal{S}$ maps the same section for a waste pit.

The cost of hauling materials from Section i to Section $i-1$ (respectively $i+1$) is represented by $c^r_{i,i-1} = cd_{i,i-1}$ (respectively $c^r_{i,i+1} = cd_{i,i+1}$), where c represents the cost of moving one unit volume of material per unit distance, and d_{ij} is the distance between section i and j.

The objective function of the quasi network flow model is also to minimize the total cost in excavation, embankment and hauling materials, which can be represented as follows,

$$\begin{aligned}\min \sum_{\substack{i\in\mathcal{S}\cup\mathcal{B}\\ m\in\mathcal{K}}} p_{i,m}V^{+}_{i,m} + \sum_{\substack{i\in\mathcal{S}\cup\mathcal{W}\\ m\in\mathcal{K}}} q_{i,m}V^{-}_{i,m} &+ \sum_{\substack{i\in\mathcal{S}\\ t\in\mathcal{T}\\ m\in\mathcal{K}}} \left(c^{r}_{i,i-1}f^{r}_{i,i-1,t,m} + c^{r}_{i,i+1}f^{r}_{i,i+1,t,m}\right)\\ &+ \sum_{\substack{j\in\mathcal{B}\\ t\in\mathcal{T}\\ m\in\mathcal{K}}} c\bar{d}_j\left(f^{b}_{j,\vartheta(j)-1,t,m} + f^{b}_{j,\vartheta(j)+1,t,m}\right)\\ &+ \sum_{\substack{j\in\mathcal{W}\\ t\in\mathcal{T}\\ m\in\mathcal{K}}} c\bar{d}_j\left(f_{\varphi(j)-1,j,t,m} + f^{w}_{\varphi(j)+1,j,t,m}\right).\end{aligned}$$

Notice that the objective function in this model is more strictly defined than in the quadratic model described earlier. In particular, the hauling cost of material must be a linear function of the distance traveled. This is unrealistic for long distances, but reasonably accurate for short hauls.

The transit nodes are virtual nodes that provide transits for all associated flows. Therefore, the sum of the flows (demand) coming to a transit node must be equal to the sum of the flows leaving (supply) the node. The flow constraints for the East transit node (Constraint (3.1)), and for the West transit node (Constraint (3.2)) ensure that the supply and demand at both transit nodes are equal.

$$f^{r}_{i-1,i,t,m} + f^{u}_{i,i+1,t,m} + \sum_{\substack{j\in\mathcal{B}\\ \vartheta(j)=i}} f^{b}_{j,i+1,t} = f^{r}_{i,i+1,t,m} + f^{l}_{i-1,i,t,m} + \sum_{\substack{j\in\mathcal{W}\\ \vartheta(j)=i}} f^{w}_{i-1,j,t,m},$$
$$\forall t\in\mathcal{T}, i\in\mathcal{S}, m\in\mathcal{K}, \quad (3.1)$$

$$f^{r}_{i+1,i,t,m} + f^{u}_{i,i-1,t,m} + \sum_{\substack{j\in\mathcal{B}\\ \vartheta(j)=i}} f^{b}_{j,i-1,t,m} = f^{r}_{i,i-1,t,m} + f^{l}_{i+1,i,t,m} + \sum_{\substack{j\in\mathcal{W}\\ \vartheta(j)=i}} f^{w}_{i+1,j,t,m},$$
$$\forall t\in\mathcal{T}, i\in\mathcal{S}, m\in\mathcal{K}. \quad (3.2)$$

The balance constraints ensure that the total unload flows from a section is equal to the cut volume of that section (Constraint (3.3)), and the total load flows to a section is equal to the fill volume (Constraint (3.4)). Borrow

(Constraint (3.5)) and waste (Constraint (3.6)) pits constraints are written similarly.

$$\sum_{t\in\mathcal{T}} f^u_{i,i-1,t,m} + f^u_{i,i+1,t,m} + f^u_{i,i,t,m} = V^+_{i,m}, \qquad \forall i \in \mathcal{S}, m \in \mathcal{K}, \tag{3.3}$$

$$\sum_{t\in\mathcal{T}} f^l_{i-1,i,t,m} + f^l_{i+1,i,t,m} + f^l_{i,i,t,m} = V^-_{i,m}, \qquad \forall i \in \mathcal{S}, m \in \mathcal{K}, \tag{3.4}$$

$$\sum_{t\in\mathcal{T}} f^b_{j,\vartheta(j)-1,t,m} + f^b_{j,\vartheta(j)+1,t,m} = V^+_{j,m}, \qquad \forall j \in \mathcal{B}, m \in \mathcal{K}, \tag{3.5}$$

$$\sum_{t\in\mathcal{T}} f_{\varphi(j)-1,j,t} + f^w_{\varphi(j)+1,j,t,m} = V^-_{j,m}, \qquad \forall j \in \mathcal{W}, m \in \mathcal{K}. \tag{3.6}$$

The volume constraints (2.4), (2.5), (2.6), (2.7), (2.8), smoothness constraints (2.17), (2.18), grade constraints (2.19) and fixed point constraints (2.20), (2.21), (2.22) are the same for this quasi network flow model as described in the quadratic model.

If section i is a block, then for the East transit node equations $f^r_{i-1,i,t,m} = f^l_{i-1,i,t,m}$ and $f^r_{i,i+1,t,m} = f^u_{i,i+1,t,m}$ ($\forall t \in \mathcal{T}, m \in \mathcal{K}$) ensure that no material can transfer across section i to the East direction. Because $f^r_{i-1,i,t,m} = f^l_{i-1,i,t,m}$ implies that when materials are transferred from section $i-1$ to section i, all the materials are loaded (filled) to section i. And $f^r_{i,i+1,t,m} = f^u_{i,i+1,t,m}$ implies that when materials are transferred from section i to section $i+1$, all the materials are unloaded (cut) from Section i to Section $i+1$. Similarly, in the case of West transit nodes, $f^r_{i+1,i,t,m} = f^l_{i+1,i,t}$ and $f^r_{i,i-1,t,m} = f^u_{i,i-1,t,m}$ ensure that no material can transfer across Section i to the West direction.

From the above reasoning, we can ensure that no material movement can occur over a block to the East direction, until that block is removed by Constraints (3.7), (3.8), (3.9), and (3.10). Similarly, for the West direction, Constraints (3.11), (3.12), (3.13), (3.14) ensure no movement over a block.

$$-(f^r_{\varsigma(k)-1,\varsigma(k),t,m} - f^l_{\varsigma(k)-1,\varsigma(k),t,m}) \leq My_{k,t-1}, \quad \forall k \in \mathcal{I}, t \in \mathcal{T}, m \in \mathcal{K}, \tag{3.7}$$

$$f^r_{\varsigma(k)-1,\varsigma(k),t,m} - f^l_{\varsigma(k)-1,\varsigma(k),t,m} \leq My_{k,t-1}, \quad \forall k \in \mathcal{I}, t \in \mathcal{T}, m \in \mathcal{K}, \tag{3.8}$$

$$-(f^r_{\varsigma(k),\varsigma(k)+1,t,m} - f^u_{\varsigma(k),\varsigma(k+1),t,m}) \leq My_{k,t-1}, \quad \forall k \in \mathcal{I}, t \in \mathcal{T}, m \in \mathcal{K}, \tag{3.9}$$

$$f^r_{\varsigma(k),\varsigma(k)+1,t,m} - f^u_{\varsigma(k),\varsigma(k+1),t,m} \leq My_{k,t-1}, \quad \forall k \in \mathcal{I}, t \in \mathcal{T}, m \in \mathcal{K}. \tag{3.10}$$

$$-(f^r_{\varsigma(k)+1,\varsigma(k),t,m} - f^l_{\varsigma(k)+1,\varsigma(k),t,m}) \leq My_{k,t-1}, \quad \forall k \in \mathcal{I}, t \in \mathcal{T}, m \in \mathcal{K}, \tag{3.11}$$

$$f^r_{\varsigma(k)+1,\varsigma(k),t,m} - f^l_{\varsigma(k)+1,\varsigma(k),t,m} \leq My_{k,t-1}, \quad \forall k \in \mathcal{I}, t \in \mathcal{T}, m \in \mathcal{K}, \tag{3.12}$$

$$-(f^r_{\varsigma(k),\varsigma(k)-1,t,m} - f^u_{\varsigma(k),\varsigma(k)-1,t,m}) \leq My_{k,t-1}, \quad \forall k \in \mathcal{I}, t \in \mathcal{T}, m \in \mathcal{K}, \tag{3.13}$$

$$f^r_{\varsigma(k),\varsigma(k)-1,t,m} - f^u_{\varsigma(k),\varsigma(k)-1,t,m} \leq My_{k,t-1}, \quad \forall k \in \mathcal{I}, t \in \mathcal{T}, m \in \mathcal{K}. \tag{3.14}$$

The block constraints must also ensure that no earth movement can occur among sections, borrow pits, and waste pits: in between two blocks, before the first block, and after the last block with no access roads, until blocks are removed. Therefore, there is no transit flow (Constraints (3.15), (3.16)), borrow flow (Constraints (3.17), (3.18)), or waste flow (Constraints (3.19), (3.20)) between two blocks with no access roads until one of the blocks is removed.

$$f^r_{i,i+1,t,m} \leq M(y_{k_1,t-1} + y_{k_2,t-1}), \quad \forall t \in \mathcal{T}, (k_1,k_2) \in \bar{\mathcal{I}}^2, m \in \mathcal{K}, \; \varsigma(k_1) \leq i, i+1 \leq \varsigma(k_2), \tag{3.15}$$

$$f^r_{i+1,i,t,m} \leq M(y_{k_1,t-1} + y_{k_2,t-1}), \quad \forall t \in \mathcal{T}, (k_1,k_2) \in \bar{\mathcal{I}}^2, m \in \mathcal{K}, \; \varsigma(k_1) \leq i, i+1 \leq \varsigma(k_2), \tag{3.16}$$

$$f^b_{j,\vartheta(j)+1,t,m} \leq M(y_{k_1,t-1} + y_{k_2,t-1}), \quad \forall t \in \mathcal{T}, (k_1,k_2) \in \bar{\mathcal{I}}^2, m \in \mathcal{K}, \; \varsigma(k_1) \leq \vartheta(j)-1, \vartheta(j), \; \vartheta(j)+1 \leq \varsigma(k_2), \tag{3.17}$$

$$f^b_{j,\vartheta(j)-1,t,m} \leq M(y_{k_1,t-1} + y_{k_2,t-1}), \quad \forall t \in \mathcal{T}, (k_1,k_2) \in \bar{\mathcal{I}}^2, m \in \mathcal{K}, \; \varsigma(k_1) \leq \vartheta(j)-1, \vartheta(j), \; \vartheta(j)+1 \leq \varsigma(k_2), \tag{3.18}$$

$$f^w_{\varphi(j)+1,j,t,m} \leq M(y_{k_1,t-1} + y_{k_2,t-1}), \quad \forall t \in \mathcal{T}, (k_1,k_2) \in \bar{\mathcal{I}}^2, m \in \mathcal{K}, \; \varsigma(k_1) \leq \varphi(j)-1, \varphi(j), \; \varphi(j)+1 \leq \varsigma(k_2), \tag{3.19}$$

$$f^w_{\varphi(j)-1,j,t,m} \leq M(y_{k_1,t-1} + y_{k_2,t-1}), \quad \forall t \in \mathcal{T}, (k_1,k_2) \in \bar{\mathcal{I}}^2, m \in \mathcal{K}, \; \varsigma(k_1) \leq \varphi(j)-1, \varphi(j), \; \varphi(j)+1 \leq \varsigma(k_2). \tag{3.20}$$

There will also be no transit flow (Constraints (3.21) and (3.22)), borrow flow (Constraints 3.23 and 3.24), or waste flow (Constraints 3.25 and 3.26) after the last block with no access road until the block is removed.

$$f^{r}_{i,i+1,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, \varsigma(k) \leq i, i+1 \leq n, k \in \bar{\mathcal{I}}_{\rightarrow}, m \in \mathcal{K}, \tag{3.21}$$

$$f^{r}_{i+1,i,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, \varsigma(k) \leq i, i+1 \leq n, k \in \bar{\mathcal{I}}_{\rightarrow}, m \in \mathcal{K} \tag{3.22}$$

$$f^{b}_{j,\vartheta(j)+1,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, \varsigma(k) \leq \vartheta(j)-1, \vartheta(j), \vartheta(j)+1 \leq n, \; k \in \bar{\mathcal{I}}_{\rightarrow}, m \in \mathcal{K}, \tag{3.23}$$

$$f^{b}_{j,\vartheta(j)-1,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, \varsigma(k) \leq \vartheta(j)-1, \vartheta(j), \vartheta(j)+1 \leq n, \; k \in \bar{\mathcal{I}}_{\rightarrow}, m \in \mathcal{K} \tag{3.24}$$

$$f^{w}_{\varphi(j)+1,j,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, \varsigma(k) \leq \varphi(j)-1, \varphi(j), \varphi(j)+1 \leq n, \; k \in \bar{\mathcal{I}}_{\rightarrow}, m \in \mathcal{K}, \tag{3.25}$$

$$f^{w}_{\varphi(j)-1,j,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, \varsigma(k) \leq \varphi(j)-1, \varphi(j), \varphi(j)+1 \leq n, \; k \in \bar{\mathcal{I}}_{\rightarrow}, m \in \mathcal{K}. \tag{3.26}$$

Similarly, there will also be no transit flow (Constraints (3.27) and (3.28)), borrow flow (Constraints (3.29) and (3.30)), or waste flow (Constraints (3.31) and (3.32)) before the first block with no access road until the block is removed.

$$f^{r}_{i,i+1,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, 1 \leq i, i+1 \leq \varsigma(k), k \in \bar{\mathcal{I}}_{\leftarrow}, m \in \mathcal{K}, \tag{3.27}$$

$$f^{r}_{i+1,i,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, 1 \leq i, i+1 \leq \varsigma(k), k \in \bar{\mathcal{I}}_{\leftarrow}, m \in \mathcal{K}, \tag{3.28}$$

$$f^{b}_{j,\vartheta(j)+1,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, 1 \leq \vartheta(j)-1, \vartheta(j), \vartheta(j)+1 \leq \varsigma(k), \; k \in \bar{\mathcal{I}}_{\leftarrow}, m \in \mathcal{K}, \tag{3.29}$$

$$f^{b}_{j,\vartheta(j)-1,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, 1 \leq \vartheta(j)-1, \vartheta(j), \vartheta(j)+1 \leq \varsigma(k), \; k \in \bar{\mathcal{I}}_{\leftarrow}, m \in \mathcal{K}, \tag{3.30}$$

$$f^{w}_{\varphi(j)+1,j,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, 1 \leq \varphi(j)-1, \varphi(j), \varphi(j)+1 \leq \varsigma(k), \; k \in \bar{\mathcal{I}}_{\leftarrow}, m \in \mathcal{K}, \tag{3.31}$$

$$f^{w}_{\varphi(j)-1,j,t,m} \leq M y_{k_1,t-1}, \qquad \forall t \in \mathcal{T}, 1 \leq \varphi(j)-1, \varphi(j), \varphi(j)+1 \leq \varsigma(k), \; k \in \bar{\mathcal{I}}_{\leftarrow}, m \in \mathcal{K}. \tag{3.32}$$

The block removal indicator constraints ensure that when the required amount of earth is excavated from or embanked to a section with a block, then the block is considered removed (Constraints (3.33) and (3.34)).

$$\sum_{t=0}^{u} \left(f^u_{\varsigma(k),\varsigma(k)-1,t,m} + f^u_{\varsigma(k),\varsigma(k)+1,t,m} \right) + M(1 - y_{ku}) \geq V^+_{\varsigma(k),m}, \quad \forall k \in \mathcal{I}, u \in T, m \in \mathcal{K}, \tag{3.33}$$

$$\sum_{t=0}^{u} \left(f^l_{\varsigma(k)-1,\varsigma(k),t,m} + f^l_{\varsigma(k)+1,\varsigma(k),t,m} \right) + M(1 - y_{ku}) \geq V^-_{\varsigma(k),m}, \quad \forall k \in \mathcal{I}, u \in T, m \in \mathcal{K}. \tag{3.34}$$

The block removal enforcement Constraint (2.15) and monotonicity Constraint (2.16) are the same for the quasi network flow model as described for the quadratic model. The extra bound constraints set the lower bounds of the variables for this model.

$$f^r_{i,i-1,t,m} \geq 0, \qquad f^r_{i,i+1,t,m} \geq 0, \qquad \forall i \in \mathcal{S}, t \in \mathcal{T}, m \in \mathcal{K}, \tag{3.35}$$

$$f^l_{i-1,i,t,m} \geq 0, \qquad f^l_{i+1,i,t} \geq 0, \qquad \forall i \in \mathcal{S}, t \in \mathcal{T}, m \in \mathcal{K}, \tag{3.36}$$

$$f^u_{i,i-1,t,m} \geq 0, \qquad f^u_{i,i+1,t} \geq 0, \qquad \forall i \in \mathcal{S}, t \in \mathcal{T}, m \in \mathcal{K}, \tag{3.37}$$

$$f^b_{j,\vartheta(j)-1,t,m} \geq 0, \qquad f^b_{j,\vartheta(j)+1,t,m} \geq 0, \qquad \forall j \in \mathcal{B}, t \in \mathcal{T}, m \in \mathcal{K}, \tag{3.38}$$

$$f^w_{\varphi(j)-1,j,t,m} \geq 0, \qquad f^w_{\varphi(j)+1,j,t,m} \geq 0, \qquad \forall j \in \mathcal{W}, t \in \mathcal{T}, m \in \mathcal{K}. \tag{3.39}$$

In this chapter we have discussed the quasi network flow model for the vertical alignment of roads. This model reduces the size of the problem from the quadratic model (see Chapter 2) from quadratic to linear with respect to the number of sections. However, it has one drawback: the cost function must be linear with respect to the hauling distance of materials. And we still need a convex relationship between volumes of materials and the offset value. In Chapter 4, we discuss some techniques to improve the solution time for both models and remove the dependency on the convex relationship between volume and offset levels.

Chapter 4

Speedup techniques

One of our goals in this research is to determine methods to solve the vertical alignment problem in the most efficient manner. The models we described in the previous two chapters can solve problems according to the full specification except one limitation, i.e., the volume of each material must be convex with respect to the height of the offset level. In this chapter we also remove that limitation of not able to handle non-convexity. To make an interactive design tools it is necessary to compute the optimal solution quickly. In this chapter we describe four strategies that we used to decrease the time to find a solution. In Chapter 5 we numerically test and compare these methods.

4.1 Techniques with offset levels

As input for both models (the quadratic and the quasi network flow), we get volume information at different offset levels. We select an optimal height among those input offset levels by approximating the volumes of materials as a piecewise linear function (see Constriants (2.4), (2.5), (2.6), (2.7), and (2.8)). We have binary variables for each offset levels for all sections. In total we need $n(n_l - 1)$ binary variables, where n is the number of sections and n_l is the number of offset levels. This large number of binary variables increases the solution time significantly. In the next two subsection we describe two techniques to accelerate the computation by modifying the constraints involving with offset levels. These techniques are applicable for both models. However, the second technique will remove the limitation of non-convexity discussed above.

4.1.1 Reducing binary variables for offset levels

In the volume constraints (Constraints (2.4), (2.5), (2.6), and (2.7)) of the quadratic and quasi network flow models, there are binary variables for all the intervals of the input offsets ($L_{i,l}$) for each section $i \in \mathcal{S}$. This requires $n(n_l - 1)$ binary variables, where n is the number of sections and $(n_l - 1)$ is the number of intervals of the input offsets.

For simplicity, we now assume that every section $i \in \mathcal{S}$ has the same number of input offsets $(L_{i,l})$ numbered from 0 to $n_l - 1$. Let $\mathcal{L} = \{0, 1, ..., n_l - 2\}$ be the index set for the intervals of the input offset levels of all sections. A number from 0 to $n_l - 2$ can be represented using $\lceil \log_2(n_l - 2) \rceil$ binary variables as follows:

$$l = \sum_{k=0}^{\lfloor \log_2(n_l - 2) \rfloor} b_{lk} \cdot 2^k,$$

where $b_{lk} \in \{0, 1\}$ for each $l \in \mathcal{L}$. Let $B_k^1 = \{l \in \mathcal{L} : b_{lk} = 1\}$ and $B_k^0 = \{l \in \mathcal{L} : b_{lk} = 0\}$.

Define the set $\mathcal{L}' = \{0, 1, 2, ..., \lfloor \log_2(n_l - 2) \rfloor\}$ to help representing $(n_l - 1)$ number of intervals of offsets in binary format. Let $\alpha_{i,k} (\forall k \in \mathcal{L}', i \in \mathcal{S})$ be the binary variable which represents the selected interval of offsets for Section i.

Note that indicator variables are still needed to write the volume constraints (2.6, 2.7). Therefore, we introduce extra continuous variables $\Gamma_{i,l} \in [0, 1]$ (for all $i \in \mathcal{S}, l \in \mathcal{L}$) to indicate which interval of the offset levels is selected. The new constraints are written in such a way that $\Gamma_{i,l}$ can only take the value 0 or 1. The variable $\Gamma_{i,l}$ plays the role of the binary variable $\nu_{i,l}$ of the previous models. The new constraints (4.1) and (4.2) ensure that only one of $\Gamma_{i,l}$ $(\forall l \in \mathcal{L})$ can equal 1 for each section $i \in \mathcal{S}$.

$$\sum_{l \in B_k^1} \Gamma_{i,l} \leq \alpha_{ik}, \qquad \forall i \in \mathcal{S}, k \in \mathcal{L}', \tag{4.1}$$

$$\sum_{l \in B_k^0} \Gamma_{i,l} \leq (1 - \alpha_{ik}), \qquad \forall i \in \mathcal{S}, k \in \mathcal{L}'. \tag{4.2}$$

In applying the above, we also note that Constraint (2.8) is no longer required because for a given combination of $\alpha_{ik} (\forall k \in \mathcal{L}')$, for each section $i \in \mathcal{S}$, only one of the $\Gamma_{i,l} (\forall l \in \mathcal{L})$ can be one.

Example 4.1. Consider the case of 5 input offset levels. Then we have 4 intervals and the set $\mathcal{L}' = \{0, 1\}$. Thus for each section $i \in \mathcal{S}$, we get the following inequalities:

$$\begin{aligned}
\Gamma_{i,1} + \Gamma_{i,3} &\leq \alpha_{i0}, \\
\Gamma_{i,0} + \Gamma_{i,2} &\leq 1 - \alpha_{i0}, \\
\Gamma_{i,2} + \Gamma_{i,3} &\leq \alpha_{i1}, \\
\Gamma_{i,0} + \Gamma_{i,1} &\leq 1 - \alpha_{i1}.
\end{aligned}$$

Therefore, if we choose $(\alpha_{i1}, \alpha_{i0}) = (1, 0)$ that represents interval 2 of offset levels, then we get the following concrete inequalities from where it is easily noticeable that only $\Gamma_{i,2}$ equals to 1.

$$\begin{aligned}
\Gamma_{i,1} + \Gamma_{i,3} &\leq 0, \\
\Gamma_{i,0} + \Gamma_{i,2} &\leq 1, \\
\Gamma_{i,2} + \Gamma_{i,3} &\leq 1, \\
\Gamma_{i,0} + \Gamma_{i,1} &\leq 0.
\end{aligned}$$

Now, if we replace the binary variable $\nu_{i,l}$ with the continuous indicator variable $\Gamma_{i,l}$ in the volume constraints (2.6, 2.7), and add new constraints (4.1) and (4.2), then we will get the same result as the original model, but the binary variables are reduced from $n(n_l - 1)$ to $n\lceil\log_2(n_l - 1)\rceil$ and n constraints (2.8) are discarded. However, the number of continuous variables is increased by $n(n_l - 1)$, and $2n\lceil\log_2(n_l - 1)\rceil$ new constraints are added.

4.1.2 Modeling volume constraints with SOS2 variables

The volume constraints in the quadratic and quasi network flow models work only when the input volume maintains the convex relationship with the height. But if the convexity does not hold, as often occurs in practice, then we need to select two adjacent binary variables to indicate an interval. However, instead of writing complicated volume constraints and binary variable selection constraints, we can use SOS2 variables to model this piecewise relationship between volume and offset height. Recall that SOS2 variables, or special ordered set variables of type 2, are variables with the property that at most two variables can be nonzero and they must be adjacent to each other [KdFJN04],[KdFN06],[MMM06],[Tom81]. Using this technique we are able to handle non-convex relation of volumes with respect to the heights of offset levels.

Let $\lambda_{i,l} (\forall l \in \mathcal{L}_i)$ be the set of SOS2 variables for each section $i \in \mathcal{S}$. These variables denote non-negative weights for all levels such that their sum is 1 (Constraint (4.3)). The offset variable u_i can be any point in between two input offset levels (l and $l+1$) that can be calculated by the weighted sum of the two adjacent input offset heights (Constraint (4.4)). Similarly, the cut and fill volumes can be calculated using the weighted sum of input volumes of two adjacent offset levels (Constraints (4.5) and (4.6)). In this formulation, we do not have to calculate slopes for each interval.

$$\sum_{l \in \mathcal{L}_i} \lambda_{i,l} = 1, \qquad \forall i \in \mathcal{S}, \tag{4.3}$$

$$\sum_{l \in \mathcal{L}_i} \lambda_{i,l} L_{i,l} = u_i, \qquad \forall i \in \mathcal{S}, \tag{4.4}$$

$$\sum_{l \in \mathcal{L}_i} \lambda_{i,l} R^+_{i,l} = V^+_i, \qquad \forall i \in \mathcal{S}, \tag{4.5}$$

$$\sum_{l \in \mathcal{L}_i} \lambda_{i,l} R^-_{i,l} = V^-_i, \qquad \forall i \in \mathcal{S}. \tag{4.6}$$

4.2 Techniques with blocks

Considering blocks during earthwork operation is a great idea. However, blocks make the problem harder to solve. We need to find an optimal schedule for material movement during vertical alignment optimization. To indicate the removal of blocks we define binary variables in Chapter 2. We need $n_b(n_b+1)$ binary variables for blocks in both models (Quadratic and Quasi network flow), where n_b is the number of blocks. Moreover, it increases the number of movement (quadratic) or flow (quasi network flow) variables by a factor of (n_b+1) to associate time steps into them.

4.2.1 Block constraints with SOS1 variables

In the basic quadratic and quasi network flow models, the block constraints have monotonicity constraint (2.16). The monotonicity constraint ensures that once a block is removed it remains removed by setting 1 to the binary variables y_{kt}, for all $k \in \mathcal{I}, t \in \mathcal{T}$, upto the last time step. However, the block k is removed when 1 is set in y_{kt} for the first time. From this reasoning, it follows that we can model the block constraints with SOS1 variable. Recall that SOS1 variables have the property that at most one variable can be 1 from that set. The SOS1 variables can be treated very efficiently by most Mixed Integer Linear Programming (MILP) solvers.

For each block $k \in \mathcal{I}$, we introduce a set of SOS1 variables γ_{kt} for all $t \in \mathcal{T}$. As a result, the monotonicity constraints are no longer needed.

However we have to rewrite all the block constraints for both basic models as a summation of γ_{kt} instead of just one y_{kt} per constraint. In particular, the block Constraints (2.9), (2.10), (2.11),(2.12), (2.13), (2.14), and (2.15) of the quadratic model are rewritten using SOS1 variables as Constraints (4.7), (4.8), (4.9), (4.10), (4.9), and (4.10).

$$\begin{aligned}
x_{ijt,m} &\leq M \sum_{w=0}^{t-1} \gamma_{kw}, && \forall t \in \mathcal{T} \setminus \{0\}, k \in \mathcal{I}, \\
& && (i,j) \in \mathcal{N}_b^k, m \in \mathcal{K}, && (4.7) \\
x_{ijt,m} &\leq M \sum_{w=0}^{t-1} \gamma_{kw}, && \forall t \in \mathcal{T} \setminus \{0\}, k \in \bar{\mathcal{I}}_{\leftarrow}, \\
& && i,j \in \mathcal{N}_b^{\leftarrow,k}, m \in \mathcal{K}, && (4.8) \\
x_{ijt,m} &\leq M \sum_{w=0}^{t-1} \gamma_{kw}, && \forall t \in \mathcal{T} \setminus \{0\}, k \in \bar{\mathcal{I}}_{\rightarrow}, \\
& && i,j \in \mathcal{N}_b^{k,\rightarrow}, m \in \mathcal{K}, && (4.9) \\
x_{ijt,m} &\leq M \sum_{w=0}^{t-1} \left(\gamma_{k_1,w} + \gamma_{k_2,w}\right), && \forall t \in \mathcal{T} \setminus \{0\}, (k_1,k_2) \in \bar{\mathcal{I}}^2, \\
& && i,j \in \mathcal{N}_b^{k1,k2}, m \in \mathcal{K}, && (4.10)
\end{aligned}$$

$$\sum_{t=0}^{u} \sum_{\substack{j \in \mathcal{N}_{\rightarrow}^{\varsigma(k)} \\ m \in \mathcal{K}}} x_{\varsigma(k)jt,m} + M\left(1 - \sum_{w=0}^{u} \gamma_{kw}\right) \geq V_{\varsigma(k)}^{+}, \quad \forall k \in \mathcal{I}, u \in T, m \in \mathcal{K}, \tag{4.11}$$

$$\sum_{t=0}^{u} \sum_{\substack{j \in \mathcal{N}_{\leftarrow}^{\varsigma(k)} \\ m \in \mathcal{K}}} x_{j\varsigma(k)t} + M\left(1 - \sum_{w=0}^{u} \gamma_{kw}\right) \geq V_{\varsigma(k)}^{-}, \qquad \forall k \in \mathcal{I}, u \in T, m \in \mathcal{K}. \tag{4.12}$$

The block constraints for the basic quasi network flow model are rewritten in an analogous manner.

4.2.2 Reducing the number of binary variables for blocks

An alternative to SOS1 constraints for working with blocks, is to rewrite the block variables using $\log n$ binary variables. This modeling method uses similar techniques as those discussed in Subsection 4.1.1 for offset levels. We can represent each time step $t \in \mathcal{T}$ in binary format as follows:

$$t = \sum_{j=0}^{\lfloor \log_2 n_b \rfloor} b_{tj} \cdot 2^j,$$

where $b_{tj} \in \{0, 1\}$. Define $B_t^1 = \{j : b_{tj} = 1\}$ and $B_t^0 = \{j : b_{tj} = 0\}$, for each $t \in T$. Let $\mathcal{T}' = \{0, 1, 2, ..., \lfloor \log_2 n_b \rfloor\}$ be a set that helps to represent $(n_b + 1)$ number of time steps in binary format. Let v_{kj} $(\forall j \in \mathcal{T}', k \in \mathcal{I})$ be the set of binary variables which represent the time step at which the block k is removed.

We then introduce extra continuous variables $\beta_{kt} \in [0, 1]$ (for all $t \in \mathcal{T}, k \in \mathcal{I}$) to indicate time step t when the block k is removed. The new constraints (4.13) and (4.14) ensure that there will always be consecutive ones in $\beta_{kt}(\forall t \in \mathcal{T})$ for each $k \in \mathcal{I}$ upto the last time step t. This represents that once a block is removed it remains removed.

$$\sum_{u=0}^{t-1} \beta_{ku} \leq t \sum_{j \in B_t^1} (1 - v_{kj}), \qquad \forall k \in \mathcal{I}, t \in \mathcal{T} \setminus \{0, n_b\}, \tag{4.13}$$

$$\sum_{u=0}^{n_b} \beta_{ku} \geq (n_b + 1) - \sum_{j=0}^{\lfloor \log_2 n_b \rfloor} 2^j \cdot v_{kj}, \qquad \forall k \in \mathcal{I}. \tag{4.14}$$

Therefore, the monotonicity constraint (2.16) is not needed any more. The other block constraints for both (quadratic and quasi network flow) models remain the same. However, one needs to replace the binary variables y_{kt} with the continuous variables β_{kt}.

Example 4.2. If we have $\mathcal{I} = \{1, 2 \ldots, 7\}$ and $\mathcal{T} = \{0, 1, 2 \ldots, 7\}$ then for each $k \in \mathcal{I}$ we get the following inequalities from the Constraint (4.13):

$$\begin{aligned}
&\beta_{k0} && \leq [(1 - v_{k0})] \cdot 1, \\
&\beta_{k0} + \beta_{k1} && \leq [(1 - v_{k1})] \cdot 2, \\
&\beta_{k0} + \beta_{k1} + \beta_{k2} && \leq [(1 - v_{k1}) + (1 - v_{k0})] \cdot 3, \\
&\beta_{k0} + \beta_{k1} + \beta_{k2} + \beta_{k3} && \leq [(1 - v_{k2})] \cdot 4, \\
&\beta_{k0} + \beta_{k1} + \beta_{k2} + \beta_{k3} + \beta_{k4} && \leq [(1 - v_{k2}) + (1 - v_{k0})] \cdot 5, \\
&\beta_{k0} + \beta_{k1} + \beta_{k2} + \beta_{k3} + \beta_{k4} + \beta_{k5} && \leq [(1 - v_{k2}) + (1 - v_{k1})] \cdot 6, \\
&\beta_{k0} + \beta_{k1} + \beta_{k2} + \beta_{k3} + \beta_{k4} + \beta_{k5} + \beta_{k6} && \leq [(1 - v_{k2}) + (1 - v_{k1}) + \\
& && \quad (1 - v_{k0})] \cdot 7.
\end{aligned}$$

From Constraint (4.14), for each $k \in \mathcal{I}$ we get the following inequality:

$$\beta_{k0} + \beta_{k1} + \beta_{k2} + \beta_{k3} + \beta_{k4} + \beta_{k5} + \beta_{k6} + \beta_{k7} \geq 8 - [2^2 v_{k2} + 2v_{k1} + v_{k0}].$$

Lemma 4.3. *Assume constraints* (4.13) *and* (4.14) *hold. If block $k \in \mathcal{I}$ is removed at time step $t \in \mathcal{T}$, then*

$$i.\ \beta_{k0} = \beta_{k1} = ... = \beta_{k(t-1)} = 0,$$

$$ii.\ \beta_{kt} = \beta_{k(t+1)} = ... = \beta_{kn_b} = 1.$$

Proof. (Part *i*): We know, the binary representation of the t^{th} time step can be given as,

$$t = \sum_{j=0}^{\lfloor \log_2 n_b \rfloor} b_{tj} \cdot 2^j.$$

If the t^{th} time step is selected for a block $k \in \mathcal{I}$, then we can write $t = \sum_{j=0}^{\lfloor \log_2 n_b \rfloor} v_{kj} \cdot 2^j$. So, $b_{tj} = v_{kj}$ for all index j for the t^{th} time step of block k. Therefore, $B_t^1 = \{j : b_{tj} = v_{kj} = 1\}$. Thus, in Constraint (4.13), the $\sum_{j \in B_t^1}(1 - v_{kj})$ term becomes zero. As a result, $\sum_{i=0}^{t-1} \beta_{ki} = 0$. This proves the first part.

(Part *ii*): If we have n_b blocks, then we know we have $n_b + 1$ time steps for each block $k \in \mathcal{I}$. Also notice, time step starts from 0 and ends at n_b. If the t^{th} time step is selected for block k, then $t = \sum_{j=0}^{\lfloor \log_2 n_b \rfloor} v_{kj} \cdot 2^j$ for that block. In the first part of this theorem, we prove that, the first t continuous indicator variables $(\beta_{k0}, \beta_{k1}, ..., \beta_{k(t-1)})$ are zero. From Constraint (4.14), we know, the sum of all the continuous variables should be at least $(n_b + 1) - t$. The continuous indicator variable can have value at most 1. So, to make the sum $(n_b + 1) - t$, we need at least $(n_b + 1) - t$ continuous indicator variables to be 1. If we rewrite Constraint (4.14), then we get the following expression,

$$\begin{aligned}
&\text{Sum of first } t \text{ continuous indicator variables} + \text{Sum of rest of the} \\
&\quad \text{continuous indicator variables} \geq (n_b + 1) - t \\
&\Rightarrow [\beta_{k0} + \beta_{k1} + ... + \beta_{k(t-1)}] + [\beta_{kt} + \beta_{k(t+1)} + ... + \beta_{kn_b}] \geq (n_b + 1) - t \\
&\Rightarrow [0] + [\text{sum of rest of the } (n_b + 1 - t) \text{ continuous indicator variables}] \\
&\qquad \geq (n_b + 1) - t.
\end{aligned}$$

To satisfy the last line, the continuous indicator variables from β_{kt} to β_{kn_b} should be one. □

Remark 4.4. (Feasibility) If block $k \in \mathcal{I}$ is removed at time step $t \in \mathcal{T}$, then the last $(n_b - t)$ inequalities of the first constraint (4.13) will always have at least $t' \in \mathcal{T}$, where $t' > t$, on the right hand side. Also, on the left hand side, there are t' continuous indicator variables, which makes the solution feasible to always get consecutive ones up to the end.

Remark 4.5. We no longer require the monotonicity constraint $\beta_{k(t+1)} \geq \beta_{kt}$ for each $t \in T \cap \{1, 2, ..., n_b\}$ under this formulation, because Lemma 4.3 proves that these properties will be satisfied for any selection of t using binary variables v_{kj}, where $j \in T'$.

Theorem 4.6. *The solution of the quadratic model is* (x_{ijtm}, y_{kt})*, where* y_{kt} *is binary variable if and only if* (x_{ijtm}, β_{kt}) *is the solution to the current model, where* β_{kt} *is continuous variable.*

Proof. Let (x_{ijtm}, y_{kt}) be the solution for the quadratic model. In the current model, we have the same constraint sets as the quadratic model except for the monotonicity constraints (2.16), which is not required anymore. Additionally, in the current model we have constraints (4.13) and (4.14) that ensure the monotonicity of β_{kt} (Lemma 4.3). So, if we have monotonicity in the quadratic model, then the solution should also satisfy the extra two constraints (4.13) and (4.14) of the current model. Therefore (x_{ijtm}, β_{kt}) is the solution to the current model.

Let (x_{ijtm}, β_{kt}) be the solution to the current model. In the quadratic model, y_{kt} is a binary variable with the same constraint satisfying Constraints (2.16), but without the extra two constraints (4.13) and (4.14) of the current model. These two extra constraints (4.13) and (4.14) ensure monotonicity in the current model. So in the solution β_{kt} will have consecutive ones till the last time step. Therefore, it should satisfy the monotonicity constraint (2.16) in the quadratic model. Hence, (x_{ijtm}, y_{kt}) is the solution to the quadratic model. □

Proposition 4.7. *Representing* $n_b + 1$ *time steps for each block* $k \in \mathcal{I}$ *in binary format, can be done using* $n_b(\lfloor \log_2 n_b \rfloor + 1)$ *binary variables.*

Proof. Let, t be an integer number. We know, t can be represented in binary format using the following formulation,

$$t = \sum_{j=0}^{\lfloor \log_2 t \rfloor} b \cdot 2^j,$$

where $b \in \{0, 1\}$. So, to represent time step n_b in binary format, we can use $\lfloor \log_2 n_b \rfloor + 1$ binary variables. For each block $k \in \mathcal{I}$, we need to have a time step which will indicate when block k is removed. There are n_b number of blocks that need to be removed, and time steps are ranged from 0 to n_b. Therefore, one can use $n_b(\lfloor \log_2 n_b \rfloor + 1)$ binary variables. □

This approach reduces the number of binary variables from $n_b(n_b + 1)$ to $n_b(\lfloor \log_2 n_b \rfloor + 1)$, but increases the number of continuous variable by $n_b(n_b + 1)$ and constraints by n_b.

Chapter 5

Numerical results

In this chapter we discuss the performance of the models and techniques that are described in earlier chapters.

5.1 Experimental setup

The experiments were setup with 60 different test problems with various section lengths, numbers of sections, blocks, access roads, and offset levels. These 60 test problems were created with help from our industry partner. We have used 7 roads that are listed in Table A.7. The configuration of all 60 test problems is listed in Tables A.8 and A.9. A sample input file for multi-material case is shown in Table A.10.

All experiments were performed on a Dell workstation with an Intel(R) Xeon(R) 3.20 GHz (4 cores) processors, a 24 GB of RAM and a 64-bit Windows 7 Enterprise operating system. We used two different MILP solvers to solve all the test problems. First, an academic edition of the IBM ILOG CPLEX Optimizer v12.4 (`http://www.cplex.com`) and second, an open source MILP solver CBC v2.7.7 (`https://projects.coin-or.org/Cbc`). The models were programmed in C++ using MS Visual Studio 2010 Professional edition.

We have two basic models: a quadratic model and a quasi network flow model (see Chapters 2 and 3). In Chapter 4, we outlined several strategies to modify both models to speedup the solution time. For offset levels, in addition to the basic method, we use a reduced binary variable technique and SOS2 variable technique (see Subsections 4.1.1 and 4.1.2). For the cases of blocks, in addition to the basic way of dealing with blocks, we use SOS1 variable technique and a reduced $n \log n$ technique (see Subsections 4.2.1 and 4.2.2). In sum, we have 2 models (quadratic, quasi network flow) with 3 level modeling techniques (basic, reduced binary, SOS2) and 3 block modeling techniques (basic, $n \log n$, SOS1), creating 18 different strategies and two solvers. These 18 strategies are shown in Table 5.1. All the test problems are solved with a 1% MIP gap tolerance and a maximum 4 minutes of CPU time.

Table 5.1: List of strategies for single material with blocks

Strategy	*Model*	*Level Technique*	*Block Technique*
QBB	Quadratic	Basic	Basic
QBN	Quadratic	Basic	$n \log n$
QBS	Quadratic	Basic	SOS1
QNB	Quadratic	Binary reduction	Basic
QNN	Quadratic	Binary reduction	$n \log n$
QNS	Quadratic	Binary reduction	SOS1
QSB	Quadratic	SOS2	Basic
QSN	Quadratic	SOS2	$n \log n$
QSS	Quadratic	SOS2	SOS1
NBB	Quasi network flow	Basic	Basic
NBN	Quasi network flow	Basic	$n \log n$
NBS	Quasi network flow	Basic	SOS1
NNB	Quasi network flow	Binary reduction	Basic
NNN	Quasi network flow	Binary reduction	$n \log n$
NNS	Quasi network flow	Binary reduction	SOS1
NSB	Quasi network flow	SOS2	Basic
NSN	Quasi network flow	SOS2	$n \log n$
NSS	Quasi network flow	SOS2	SOS1

5.2 Performance measurement

In order to compare the performance of different strategies, we use performance profiles as developed in [DM02]. Performance profiles are designed to graphically compare both speed and robustness of strategies across a test set. This is done by plotting, for each strategy, the percentage of problems that are solved within a factor of the best solve time. Let $\mathcal{P}$ be the set of all problems, and S be the set of strategies. We first compute the performance ratio

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}},$$

where $p \in \mathcal{P}$ and $t_{p,s}$ is the solving time of strategy s for problem p. The percentage of problems that are solved within a factor $\tau \in R$ are computed in logarithmic scale by

$$\rho_s(\tau) = \frac{|\{p \in \mathcal{P} : \log_2(r_{p,s}) \leq \tau\}|}{|\mathcal{P}|} \times 100\%.$$

The percentage ρ_s of all problems solved by strategy s at τ gives us an overall assessment of the performance of strategy s. High values of ρ_s near $\tau = 1$ represents fast solve times. High values of ρ_s when τ is large represents high success rates. Therefore, for a good algorithm, the plot of $\rho(\tau)$ should be found above the other algorithms. For a more detailed description of performance profiles we refer the reader to [DM02].

5.3 Experimental results for single material

Out of the 60 test problems, 46 test problems contain blocks and 14 test problems are setup without blocks. Some of the test problems are created for stress testing. For example, Test 8 (see Table A.9) is created with 40 blocks in just 50 sections that cannot be solved by any strategies within time limits. In reality, almost all problems have at most 5 and extremely rarely 10 blocks. But we include harder problems with extreme conditions to learn the limit of the models.

In Tables A.3 and A.4 the solution times required by solvers CPLEX and CBC, respectively, are listed for 18 different strategies for the 46 test problems with blocks. In Tables A.1 and A.2 the solution times are listed for the 14 test problems with no blocks for 6 different techniques (see Table 5.2) for both MIP solvers.

5.3.1 Results for problems with blocks

The performance profiles of 18 different strategies with solver CPLEX are shown in Figure 5.1. We have used MATLAB to draw the performance profiles. Note that when MATLAB plots vertical lines the value of the function ρ_s is equal to the largest values of the graph at that point. We see that the performances of Strategies NSB, NSN and NSS are very close to each other. These 3 strategies are all quasi network flow models with the SOS2 level technique and three different block techniques. These three strategies solved almost all of the problems in the problem set. Although Strategy NSB is better than the other two strategies, strategy NSN, which is the quasi network flow model with the SOS2 level technique and the $n \log n$ block technique, is very close. However, strategy NSN solves one problem less than strategy NSB.

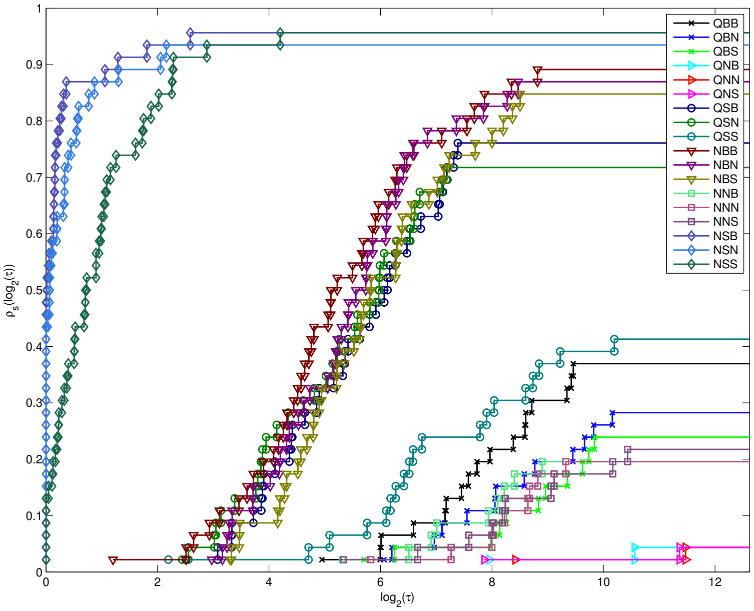


Figure 5.1: Performance profile for solver CPLEX (with blocks)

Similarly, in the case of CBC, the performance profiles of 18 different strategies are shown in Figure 5.2. Strategy NSS, which is a quasi network flow model with the SOS2 level technique and the SOS1 block technique, outperforms all the other strategies. It solves almost 90% of the problem set. Strategy NSB comes next: it can also solve 90% of the problem set but

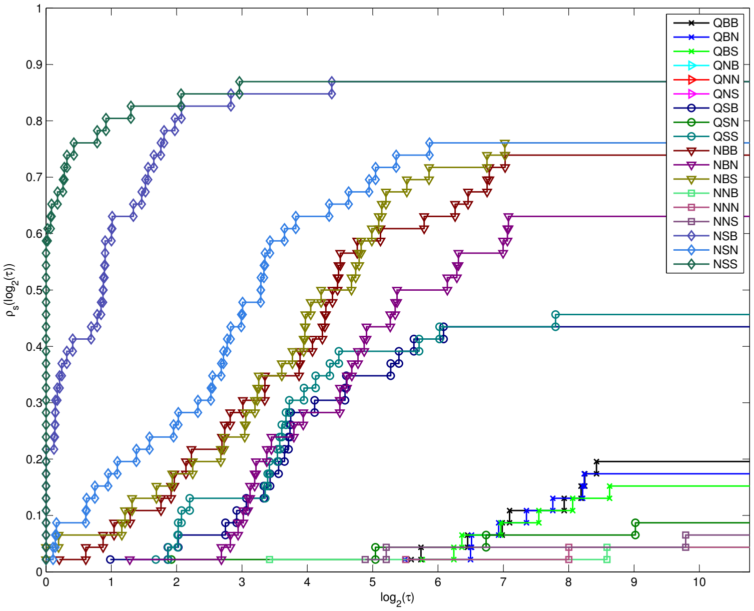


Figure 5.2: Performance profile for solver CBC (with blocks)

takes a little bit more time.

From Figures 5.1 and 5.2, we observe that the quasi network flow model performs better than the quadratic model for both solvers for solving problems with blocks. However, for the block technique (Section 4.2.1 and 4.2.2), we see that the SOS1 technique works better than the other techniques for CBC, whereas the basic technique works well for CPLEX.

In Figures 5.3, 5.4, 5.5, and 5.6 the performance profiles are shown with various techniques separately for the quadratic and quasi network flow models for each MIP solver. For CBC (Figures 5.4, 5.6), the SOS2 level technique and the SOS1 block technique work better than the other strategies by a slight margin. However, for CPLEX (Figures 5.3, 5.5), the SOS2 level technique combined with the basic block technique works better than other strategies.

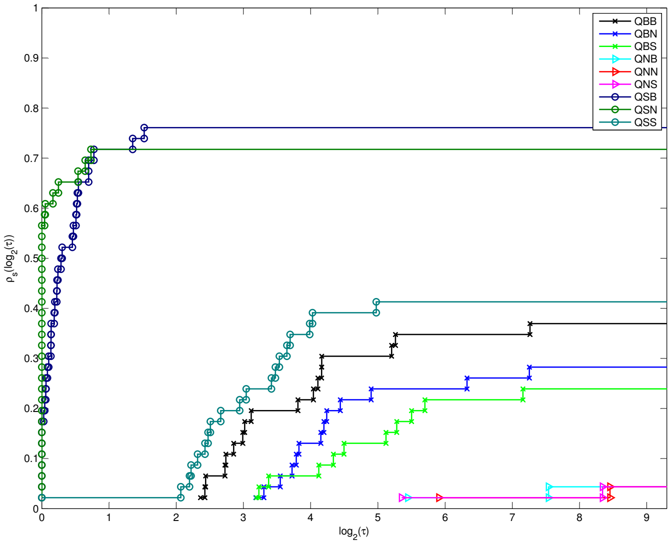


Figure 5.3: Performance profile of different strategies of the quadratic model for solver CPLEX (with blocks)

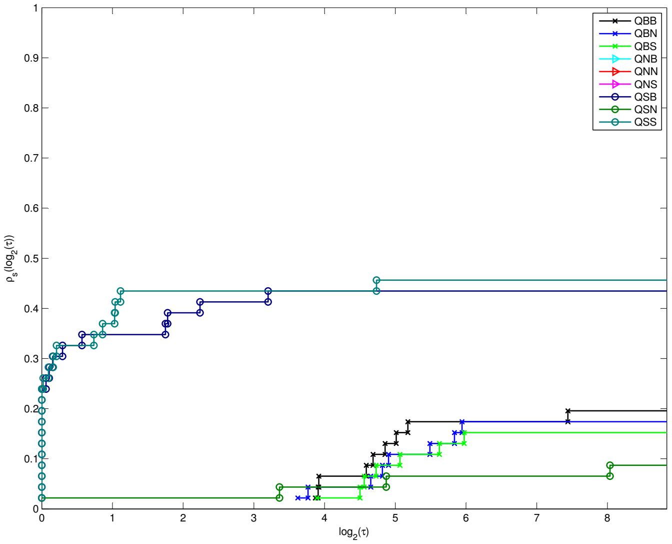


Figure 5.4: Performance profile of different strategies of the quadratic model for solver CBC (with blocks)

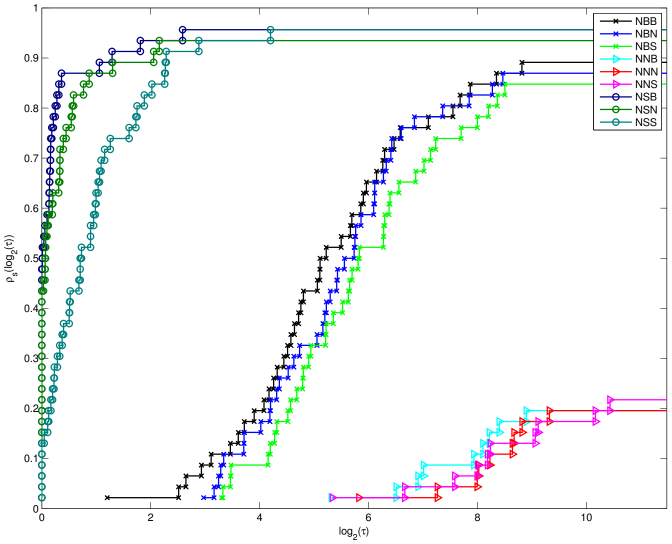


Figure 5.5: Performance profile of different strategies of the quasi network flow model for solver CPLEX (with blocks)

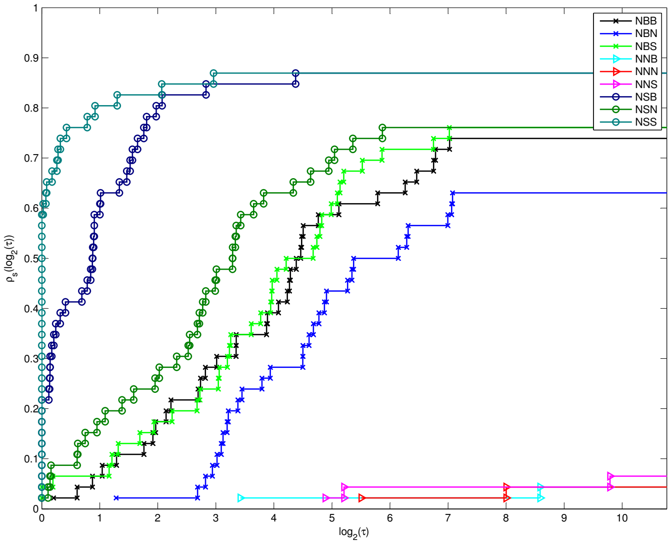


Figure 5.6: Performance profile of different strategies of the quasi network flow model for solver CBC (with blocks)

5.3.2 Results for problems with no blocks

For problems without blocks, the list of strategies is shown in Table 5.2. The run times for these 6 strategies in the case of both solvers are listed in Tables A.1 and A.2. The performance profiles for CPLEX and CBC for these 6 strategies are shown in Figures 5.7 and 5.8, respectively. From both performance profiles, we see that the SOS2 level technique makes a significant difference. Both quasi network flow and quadratic models with SOS2 can solve all of the problems, and as before, the quasi network flow model outperforms the quadratic model.

Table 5.2: List of strategies for single material without blocks

Strategy	*Model*	*Level Trick*
QB	Quadratic	Basic
QN	Quadratic	Binary reduction
QS	Quadratic	SOS2
NB	Quasi network flow	Basic
NN	Quasi network flow	Binary reduction
NS	Quasi network flow	SOS2

5.3.3 Summary of experimental result for single material

The basic quadratic model (QBB) with the basic level and block techniques can solve 38% of the problem set with blocks using CPLEX and 20% using CBC. Using the quasi network flow model (NSB), with the SOS2 level technique and the basic block technique, we improve this to 95% for CPLEX and 88% for CBC.

To compare the improvement in solution time, we consider the common problems that can be solved by all strategies. The improvement in solution time for the problems with blocks are listed below.

- For the solver CPLEX, we found the following results.
  - The quadratic model with the basic level and block technique can solve 17 problems out of 46 test problems taking 1197 seconds. Whereas the quadratic model with the SOS2 level technique and the basic block technique can solve those 17 problems in 151 seconds. The improvement in speed is a factor of 8.

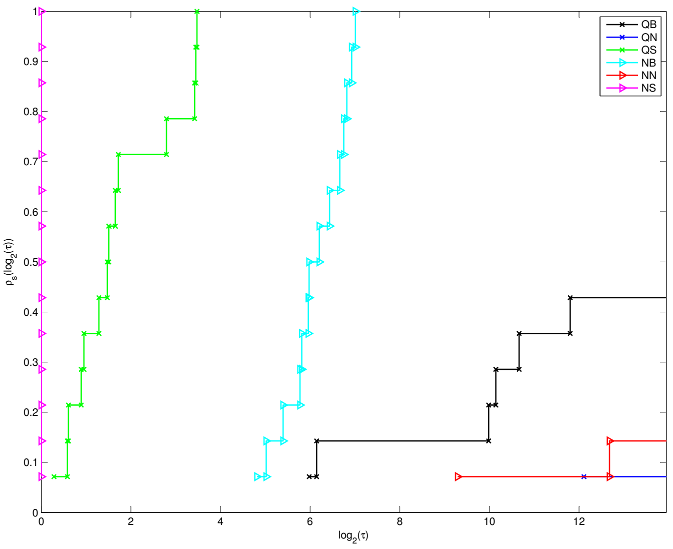


Figure 5.7: Performance profile for solver CPLEX (without blocks)

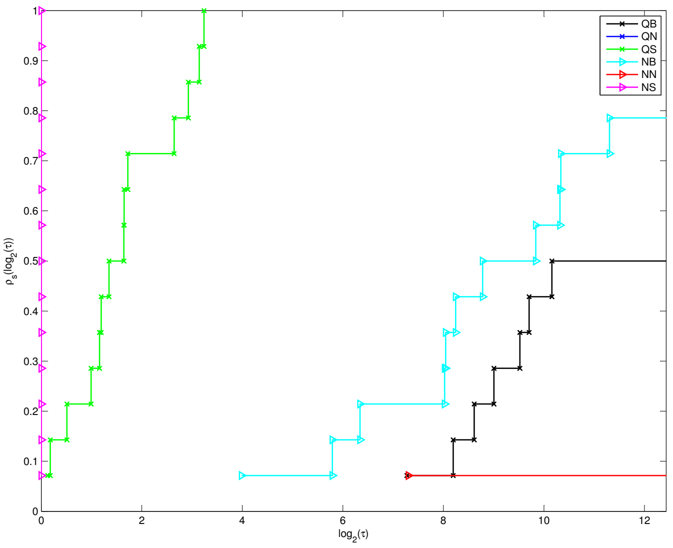


Figure 5.8: Performance profile for solver CBC (without blocks)

  - The quasi network flow model with the basic level and block technique can solve 41 test problems taking 1755 seconds. Whereas the quasi network flow model with the SOS2 level technique and the basic block technique takes 175 seconds to solve those problems. In this case the improvement in speed is a factor of 10.
  - The improvement from the basic quadratic model to quasi network flow model introduced in [Rah12] with SOS2 level technique is a factor of 173.

- For the solver CBC, we found the following results.
  - The basic strategy QBB (quadratic model with basic level and block technique) takes 785 seconds to solve 9 problems out of 46 problems. Whereas the Strategy QSB takes 50 seconds and Strategy QSS takes 205 seconds to solve those 9 problems. The performance improvement in solution time from QBB to QSB is a factor 16.
  - On the other hand, for the quasi network flow model, the basic strategy NBB takes 2128 seconds to solve 36 problems out of 46. Whereas the strategy NSS (quasi network flow with SOS2 level and SOS1 block technique) takes 623 seconds. The improvement is a factor of 3. The performance improvement from QBB to NSS is a factor of 142.

For no blocks, the basic quadratic model (QB) with basic level technique can solve 42% of the problem set using CPLEX and 50% using CBC. Using both the quadratic (QS) and the quasi network flow model(NS) with SOS2 level technique, we improve this to 100% for both solvers. For problems without blocks, the performance improvement in solution time for different strategies are listed below.

- The solution time is improved from basic quadratic model (QB) to quadratic model with SOS2 level technique (QS) by a factor of 244.2.
- The solution time is improved from basic quasi network flow model (NB) to quasi network flow model with SOS2 level technique (NS) by a factor of 94.1.
- The solution time is improved from basic quadratic model (QB) to quasi network flow model with SOS2 level technique (NS) by a factor of 1498.3. Similar results can be found for CBC.

The precedence of all techniques are shown in Tables 5.3, 5.4 for CPLEX and CBC respectively in terms of number of problems solved and solution speed.

Although the performance of the quasi network flow model is better than the quadratic model, the quasi network flow model cannot have a non linear cost function with respect to hauling distance. The quadratic model is more flexible as it can handle a nonlinear cost function.

Table 5.3: Precedence of techniques for CPLEX in terms of number of problems solved and solution speed. Precedence is decreasing from left to right in each row.

CPLEX						
Model	Quasi network flow			Quadratic		
Level Technique	SOS2	Basic	Binary Reduction	SOS2	Basic	Binary Reduction
Block Technique	Basic	SOS1	$n \log n$	Basic	SOS1	$n \log n$

Table 5.4: Precedence of techniques for CBC in terms of number of problems solved and solution speed. Precedence is decreasing from left to right in each row.

CBC						
Model	Quasi network flow			Quadratic		
Level Technique	SOS2	Basic	Binary Reduction	SOS2	Basic	Binary Reduction
Block Technique	SOS1	Basic	$n \log n$	SOS1	Basic	$n \log n$

5.4 Experimental results for multimaterial

In this section, we show the experimental results for the multimaterial vertical alignment optimization. We use the same 60 test problems that we have used for single material, but this time we consider four materials instead of one single material, and give ten minutes CPU time to solve a problem. We also use the same configuration for the computer and software (see Section 5.1).

In multimaterial test cases, the volumes of materials are mostly non-convex in nature with respect to the height of the offset levels. So we do not use the basic and binary reduction techniques for offset levels since they requires that the volumes are convex in nature. We apply only SOS2 techniques which can handle non-convex volumes with respect to heights.

5.4.1 Results for problems with blocks

The strategies that we have used for numerical testing multimaterial models with blocks are listed in Table 5.5. In Table A.5 the solution times required by CPLEX are listed for the 46 test problems with blocks. CBC cannot solve any of the test problem of mulimaterial.

Table 5.5: List of strategies without blocks for multimaterial test cases

Strategy	*Model*	*Level Technique*	*Block Technique*
QSB	Quadratic	SOS2	Basic
QSN	Quadratic	SOS2	$n \log n$
QSS	Quadratic	SOS2	SOS1
NSB	Quasi network flow	SOS2	Basic
NSN	Quasi network flow	SOS2	$n \log n$
NSS	Quasi network flow	SOS2	SOS1

The performance profiles of these six strategies is shown in Figure 5.9. In this case we observe the same behavior that we have seen for single material. NSB and NSS solve almost all the problems from the test set, although NSB has better solution time. NSN solves 88 percent problem. Among the three quasi network flow strategies, NSB is the best in terms of solution speed and robustness.

In the case of strategies with the quadratic model, the performance of the QSN and QSB are very close and significantly better than the other strategy QSS. Among the three strategy QSN is the best method in terms of number of problem solved. The reason for this kind of performance profile for the quadratic model is that the problems size grows very fast with the number of materials and blocks. Since QSN reduces the number of binary variables, the size of the problem is less than the other two strategies.

Overall, the performance profile with blocks for multimaterial test is similar to the performance of single material.

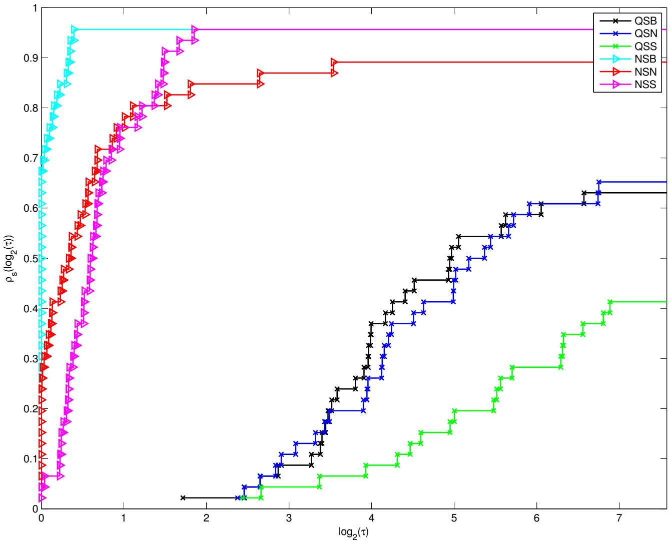


Figure 5.9: Performance profile for solver CPLEX for multi-material (with blocks)

5.4.2 Results for problems without blocks

For the case of without blocks, we now have only two strategies that are listed in Table 5.2. The solution times required by CPLEX solver are listed in Table A.6 for the 14 test cases without blocks. The performance profile is shown in Figure 5.10. In this case, it is obvious that the quasi network flow strategy with SOS2 level technique is better than the quadratic strategy with SOS2.

Table 5.6: List of strategies with blocks for multimaterial test cases

Strategy	*Model*	*Level Technique*
QS	Quadratic	SOS2
NS	Quasi network flow	SOS2

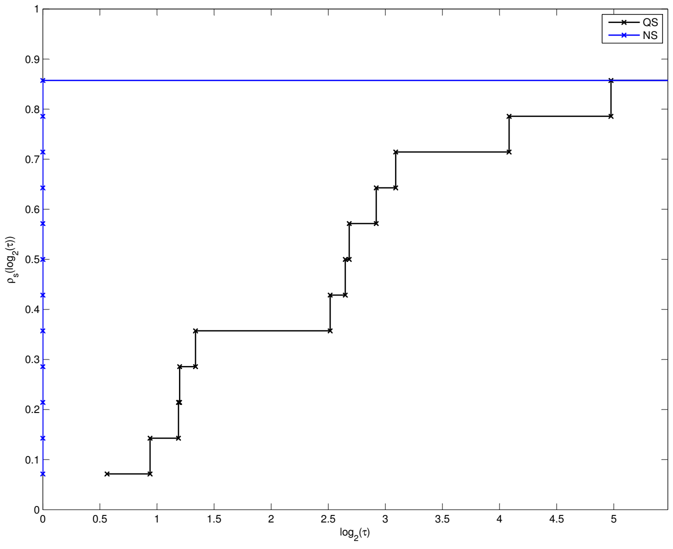


Figure 5.10: Performance profile for solver CPLEX for multi-material (without blocks)

5.4.3 Summary of results for mutlimaterial

We observe almost the same behavior in performance for the multimaterial case as we have seen in single material in terms of number of problem solved. For the mutimaterial case, the problem size increases as the number of material increases. Solver CPLEX can cope with the increase of problem size, but solver CBC is not able to solve multimaterial problems.

In the case of multimaterial, we have to consider nonconvexity in volume with respect to the height offset level. In general, all strategies with quasi network flow is significantly better than the strategies with quadratic model. In order to measure the performance in solution time explicitly, we consider the problems that are solved by both strategies. QSB can solve 27 problems with blocks out of 46 problems taking 3675 seconds in total. NSB takes 148 seconds for the same problem set. So performance increase in solution time is a factor of 25. Without block problems, the performance increase in solution time is 11 from QS to NS. This performance increase is caused mostly by the quasi network flow model, while SOS2 techniques help to increase performance for both models and accommodate the nonconvexity relation between volume and offset level.

Chapter 6

Conclusion

We started this thesis with the practical requirements that need to be considered for vertical alignment design in Chapter 1. We also discussed the shortcomings of previous research works. We developed two models based on the previous works of [Mor09], [HKL11], [Rah12]. The quadratic and quasi network flow models are an enhancement of the previous work done in [Rah12] and are described in Chapters 2 and 3. These two models are capable of considering all requirements for optimal design of the vertical alignment. Both models incorporate features such as multimaterial, sides slopes and physical blocks.

We further introduced six potential techniques to improve the solution speed that are applicable for both models in Chapter 4. Numerical tests shows that using the best combination of techniques can consistantly generate major improvements in solution time, regardless of which LP solver is used. For example, the quasi network flow model introduced in [Rah12] and further improved by the best speed techniques discussed in this research improves solution speed from the basic quadratic model by a factor of 173.

The techniques developed in this research have made it possible to solve a vertical alignment design for a very long road within a reasonable amount of time. Using these improvement techniques, our industry partner (Softree Technical Systems Inc.) has developed an interactive road design software.

6.1 Future research

6.1.1 Improvement in the quasi network flow model

The quasi network flow model has a limitation in the structure of its cost function. The hauling cost is linear with distance traveled by the volume of materials. In practice, for hauling materials, it may be enough to consider only two kinds of haul such as short haul and long haul. Short haul refers to moving material in short distance. Long haul refers to moving materials in long distance using big trucks. There could be a fixed cost for loading materials in long haul which is much higher than hauling materials in short distance. However, once the materials are put in a large truck, a long traveling distance does not increase the cost significantly. To accommodate these two hauls in our quasi network flow model, we can define another pair of East and West transit flow. Then with appropriate cost values set for the short and long haul it is probably possible to have a nonlinear cost function.

6.1.2 More options for selecting material

In our existing models, we cut and fill various sections of roads with the materials according to the given input volume tables (Table 2.1). We assume that materials are not replaceable by any other type of materials. But it can be modeled and implemented to have the alternate material options. However, it will increase the size of the problem, as we have to define all possible movement variables as x_{ijt,m_1m_2}, where m_1 and m_2 are the alternate materials pair such that low quality material m_1 is replaceable with high quality material m_2.

6.1.3 Horizontal alignment

The ultimate goal of automating the road design process is to calculate the horizontal alignment while considering the sub problem of vertical alignment. First, road engineers select a corridor for building a road using the topographical analysis of the ground profile. After fixing a corridor, road engineers select a route for the horizontal alignment within that corridor minimizing the cost while satisfying the safety, other design constraints, and considering other socioeconomic impacts. While computing the horizontal alignment, the cost is heavily dependent on the cost of the vertical alignment. Certainly, during the horizontal alignment optimization, our research will help to find the optimal solution.

Bibliography

[AAS04] AASHTO, editor. *A policy on geometric design of highways and streets*. AASHTO, 5 edition, 2004. → pages 2, 8

[ACRV13] E. Angulo, W. Castillo, R. G. Rdenas, and J. S. Vizcano. A continuous bi-level model for the expansion of highway networks. *Computers & Operations Research*, 2013. → pages 2

[Aka06] A. Akay. Minimizing total costs of forest roads with computer-aided design model. *Sadhana*, 31:621–633, 2006. → pages 7

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993. → pages 6, 23

[Aru05] K. Aruga. Tabu search optimization of horizontal and vertical alignments of forest roads. *Journal of Forest Research*, 10:275–284, 2005. → pages 7

[ATSM06] K. Aruga, T. Tasaka, J. Sessions, and S. Miyata. Tabu search optimization of forest road alignments combined with shortest paths and cubic splines. *Croatian Journal of Forest Engineering*, 27:37–47, 2006. → pages 7

[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 6

[CL06] J. F. Cheng and Y. Lee. Model for three-dimensional highway alignment. *Journal of Transportation Engineering*, 132(12):913–920, 2006. → pages 7

[CMW98] R. L. Church, A. T. Murray, and A. Weintraub. Locational issues in forest management. *Location Science*, 6(14):137 – 153, 1998. → pages 2

[DJ09] C. Davis and M. K. Jha. Modeling the effects of socioeconomic factors in highway construction and expansion. *Journal of Transportation Engineering*, 135(12):990–998, 2009. → pages 1

[DM02] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002. → pages 42

[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 4, 5

[FCS02] T. F. Fwa, W. T. Chan, and Y. P. Sim. Optimal vertical alignment analysis for highway design. *Journal of Transportation Engineering*, 128(5):395–402, 2002. → pages 4

[GAA09] A. B. Göktepe, S. Altun, and P. Ahmedzade. Optimization of vertical alignment of highways utilizing discrete dynamic programming and weighted ground line. *Turkish Journal Of Engineering & Environmental Sciences*, 10:105–116, 2009. → pages 4, 5

[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 4, 5

[GLA05] A. B. Goktepe, A. H. Lav, and S. Altun. Dynamic optimization algorithm for vertical alignment of highways. *Mathematical and Computational Applications*, 10(3):341–350, 2005. → pages 4, 5, 7

[GLA09] A. B. Goktepe, A. H. Lav, and S. Altun. Method for optimal vertical alignment of highways. *Proceedings of The Institution of Civil Engineers-transport*, 162(4):177–188, December 2009. → pages 4

[HBS68] B. E. Howard, Z. Bramnick, and J. F. Shaw. Optimum curvature principle in highway routing. *Journal of the Highway Division*, 94:61–82, 1968. → pages 3

[HHLR13] W. L. Hare, S. Hossain, Y. Lucet, and F. Rahman. Models and strategies for efficiently determining an optimal vertical alignment of roads. Technical report, Under submission, 2013. → pages iii

[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 2, 4, 6, 8, 15, 55

[HLR13] W. Hare, Y. Lucet, and F. Rahman. A mixed-integer linear programming model to optimize the vertical alignment considering blocks and side-slopes in road construction. under submission. Technical report, Under submission, 2013. → pages 6

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

[JS03] J. C. Jong and P. Schonfeld. An evolutionary model for simultaneously optimizing three-dimensional highway alignments. *Transportation Research Part B: Methodological*, 37(2):107 – 128, 2003. → pages 2

[JSJ06] M. K. Jha, P. Schonfeld, and J. C. Jong. *Intelligent Road Design*, volume 19. WIT Press, 2006. → pages 2, 3, 4, 5

[KdFJN04] A. B. Keha, I. R. de Farias Jr., and G. L. Nemhauser. Models for representing piecewise linear cost functions. *Operations Research Letters*, 32(1):44 – 48, 2004. → pages 33

[KdFN06] A. B. Keha, I. R. de Farias, and G. L. Nemhauser. A branch-and-cut algorithm without binary variables for nonconvex piecewise linear optimization. *Operations Research*, 54(5):847–858, September 2006. → pages 33

[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, May 2010. → pages 4, 6, 8, 10

[KSJ07] M. W. Kang, P. Schonfeld, and J. C. Jong. Highway alignment optimization through feasible gates. *Journal of Advanced Transportation*, 41(2):115–144, 2007. → pages 2

[LC01] Y. S. Lee and J. F. Cheng. . *Journal of Transportation Engineering-ASCE*, 127(4):303–310, Jul-Aug 2001. → pages 4

[LV92] M.G. Lay and J.E. Vance. *Ways of the World: A History of the World's Roads and of the Vehicles That Used Them*. RUTGERS University Press, 1992. → pages

[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, April 2004. → pages 4, 5, 10

[Min12a] Minister of Public Works and Government Services. Transportation in canada 2011 - comprehensive review. *Transport Canada*, Cat. No. T1-23A/2011E-PDF:15, 2012. → pages 1

[Min12b] Ministry of Transportation and Infrastructure. Construction and rehabilitation cost guide, July 2012. → pages 1

[MMM06] A. Martin, M. Mller, and S. Moritz. Mixed integer models for the stationary case of gas network optimization. *Mathematical Programming*, 105:563–582, 2006. → pages 33

[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 4, 5

[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 2, 4, 5, 6, 8, 10, 55

[OEC73] OECD. Optimisation of road alignment by the use of computers. Technical report, Organization of Economic Co-operation and Development, Paris, 1973. → pages 4

[Rah12] F. Rahman. Optimizing the vertical alignment under earthwork block removal constraints in road construction. Master's thesis, UBC, 2012. → pages 6, 8, 16, 23, 49, 55

[SH81] J. F. B. Shaw and B. E. Howard. Comparison of two integration methods in transportation routing. *Transportation Research Record*, 806:8–13, 1981. → pages 3

[SH82] J. F. B. Shaw and B. E. Howard. Expressway route optimization by ocp. *Transportation Engineering Journal*, 108:227–243, 1982. → pages 3

[TM70] A.K. Turner and R. D. Miles. The GCARS system: a computer-assisted method of regional route location. Technical Report FHWA/IN/JHRP-70/27, Purdue University, Joint Highway Research Project, 1970. → pages 4

[Tom81] J. A. Tomlin. A suggested extension of special ordered sets to non-separable non-convex programming problems. *Studies on Graphs and Discrete Programming*, 11:359–370, 1981. → pages 33

[Tri87a] D. Trietsch. Comprehensive design of highway networks. *Transportation Science*, 21(1):26–35, 1987. → pages 4

[Tri87b] D. Trietsch. A family of methods for preliminary highway alignment. *Transportation Science*, 21(1):17–25, February 1987. → pages 4

[TT03] C. W. Tat and F. Tao. Using gis and genetic algorithm in highway alignment optimization. In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, volume 2, pages 1563 – 1567 vol.2, oct. 2003. → pages 7

[Tur78] A. K. Turner. A decade of experience in computer aided route selection. *Photogrammetric Engineering and Remote Sensing*, 44:1561–1576, 1978. → pages 4

[Wan95] F. Y. M. Wan. *Introduction to the calculus of variations and its applications.* Chapman & Hall, 1995. → pages 3

Appendix

Appendix A

Tables

Table A.1: Solution times for CPLEX with 6 strategies for 14 test problems without blocks

Test #	QB	QN	QS	NB	NN	NS
1	1.485	NaN	0.032	1.181	NaN	0.021
2	1.134	NaN	0.027	1.119	NaN	0.018
3	NaN	NaN	0.028	0.647	NaN	0.023
4	31.377	136.647	0.06	1.005	203.027	0.031
5	NaN	NaN	0.065	1.474	NaN	0.035
6	74.593	NaN	0.128	2.514	NaN	0.046
7	NaN	NaN	0.122	3.144	NaN	0.05
8	NaN	NaN	0.125	4.97	NaN	0.044
9	199.839	NaN	0.176	5.671	NaN	0.056
10	NaN	NaN	0.155	4.062	NaN	0.047
11	NaN	NaN	1.605	18.932	NaN	0.147
12	NaN	NaN	1.312	22.998	NaN	0.189
13	NaN	NaN	1.322	12.813	NaN	0.119
14	156.044	NaN	1.479	10.17	86.527	0.138

Table A.2: Solution times for CBC with 6 strategies for 14 test problems without blocks

Test #	**QB**	**QN**	**QS**	**NB**	**NN**	**NS**
1	45.716	NaN	0.057	101.07	NaN	0.04
2	15.417	NaN	0.034	38.33	NaN	0.03
3	31.591	NaN	0.047	55.593	NaN	0.043
4	12.985	NaN	0.214	36.97	NaN	0.084
5	35.614	NaN	0.181	23.706	NaN	0.091
6	NaN	NaN	0.359	NaN	NaN	0.115
7	NaN	NaN	0.281	112.62	NaN	0.123
8	NaN	NaN	0.333	45.162	NaN	0.149
9	160.272	NaN	0.602	15.58	NaN	0.192
10	48.029	NaN	0.541	9.091	NaN	0.164
11	NaN	NaN	5.05	NaN	NaN	0.664
12	NaN	NaN	4.999	NaN	NaN	0.8
13	NaN	NaN	5.392	161.062	NaN	0.61
14	NaN	NaN	5.32	8.943	89.347	0.564

Table A.3: Solution time for CPLEX with 18 strategies for 48 test problems with blocks

Test #	QBB	QBN	QBS	QNB	QNN	QNS	QSB	QSN	QSS	NBB	NBN	NBS	NNB	NNN	NNS	NSB	NSN	NSS
1	1.45	3.14	2.45	11.69	16.11	10.95	0.27	0.28	1.23	0.70	0.97	0.94	4.28	12.23	4.77	0.05	0.05	0.08
2	3.03	3.46	3.57	70.85	133.41	123.73	0.40	0.38	1.60	1.62	1.56	1.40	11.50	19.18	9.02	0.05	0.05	0.06
3	9.55	26.27	24.89	NaN	NaN	NaN	1.46	1.43	9.06	2.26	2.28	1.92	33.45	63.37	52.82	0.11	0.10	0.16
4	NaN	NaN	NaN	NaN	NaN	NaN	20.88	34.66	223.09	8.91	14.00	52.68	102.79	NaN	207.15	0.37	0.43	0.75
5	43.44	23.92	49.00	NaN	NaN	NaN	2.53	2.42	18.64	3.48	2.28	4.13	83.02	78.67	240.34	0.18	0.17	0.42
6	54.05	43.39	127.68	NaN	NaN	NaN	4.05	3.29	42.56	4.53	8.71	4.98	48.24	64.71	41.54	0.16	0.30	0.36
7	89.68	175.70	NaN	NaN	NaN	NaN	14.19	12.39	102.07	17.15	25.66	52.08	NaN	NaN	NaN	1.41	6.28	10.40
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	NaN	NaN	NaN	NaN	NaN	NaN	28.58	20.61	114.49	25.56	17.55	49.82	NaN	NaN	NaN	1.43	1.34	2.74
10	111.67	NaN	NaN	NaN	NaN	NaN	16.52	6.47	75.03	39.76	15.58	21.86	NaN	NaN	NaN	0.95	0.78	0.90
11	NaN	NaN	NaN	NaN	NaN	NaN	50.46	78.99	NaN	40.62	46.36	27.03	NaN	NaN	NaN	1.34	1.18	1.18
12	NaN	NaN	NaN	NaN	NaN	NaN	78.59	114.20	NaN	77.60	58.30	164.51	NaN	NaN	NaN	3.84	4.55	1.10
13	57.78	56.80	92.90	NaN	NaN	NaN	4.54	4.12	23.42	7.96	7.49	13.95	39.06	NaN	91.41	0.30	0.38	0.36
14	43.98	92.03	51.95	NaN	NaN	NaN	1.15	1.19	6.17	15.62	13.07	14.79	NaN	34.16	130.92	0.28	0.11	0.12
15	60.40	NaN	NaN	NaN	NaN	NaN	12.46	11.75	191.66	11.28	22.65	23.36	NaN	NaN	NaN	0.52	0.42	0.81
16	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
17	NaN	NaN	NaN	NaN	NaN	NaN	33.28	28.50	132.78	34.84	18.33	34.44	NaN	NaN	NaN	2.12	1.93	2.17
18	NaN	NaN	NaN	NaN	NaN	NaN	115.21	101.36	NaN	16.40	22.56	28.85	NaN	NaN	NaN	2.87	4.89	5.68
19	231.87	NaN	NaN	NaN	NaN	NaN	47.39	43.10	NaN	14.62	24.40	13.07	NaN	203.81	NaN	1.36	1.33	1.37
20	NaN	NaN	NaN	NaN	NaN	NaN	122.71	89.45	NaN	21.00	16.15	65.18	NaN	NaN	NaN	1.60	1.70	6.48
21	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	70.54	100.87	206.28	NaN	NaN	NaN	23.45	14.16	11.26
22	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	210.73	NaN	NaN	NaN	NaN	NaN	91.53	225.44	131.60
23	150.65	NaN	NaN	NaN	NaN	NaN	25.07	22.78	NaN	7.82	30.89	18.62	NaN	NaN	NaN	0.36	0.38	1.09
24	27.61	56.59	110.77	NaN	NaN	NaN	3.76	3.18	50.44	2.20	3.46	3.60	15.62	33.19	37.86	0.13	0.13	0.13
25	NaN	NaN	NaN	NaN	NaN	NaN	14.62	10.13	NaN	13.57	15.09	45.34	NaN	NaN	NaN	0.22	0.22	0.41
26	NaN	NaN	NaN	NaN	NaN	NaN	109.03	NaN	NaN	58.77	231.60	220.83	NaN	NaN	NaN	0.75	0.77	2.48
27	109.58	182.50	NaN	NaN	NaN	NaN	6.57	6.12	68.05	29.95	26.31	40.93	NaN	NaN	NaN	0.16	0.24	0.20
28	NaN	NaN	NaN	NaN	NaN	NaN	10.33	6.37	200.25	39.88	60.48	62.07	NaN	NaN	NaN	0.20	0.17	0.19
29	70.76	41.68	99.71	NaN	NaN	NaN	2.25	1.92	24.03	35.54	25.09	35.96	NaN	NaN	NaN	0.11	0.11	0.20
30	NaN	NaN	NaN	NaN	NaN	NaN	40.41	27.70	NaN	198.18	20.70	11.30	NaN	NaN	NaN	0.44	0.64	0.62
31	54.89	127.81	70.25	NaN	NaN	NaN	6.99	6.77	33.75	28.95	11.31	11.08	NaN	NaN	NaN	0.16	0.14	0.17
32	NaN	NaN	NaN	NaN	NaN	NaN	54.91	45.04	NaN	19.65	19.08	27.24	NaN	NaN	NaN	0.33	0.34	1.10
33	NaN	NaN	NaN	NaN	NaN	NaN	36.86	21.53	NaN	21.19	26.75	26.01	NaN	NaN	NaN	0.28	0.29	0.45
34	NaN	NaN	NaN	NaN	NaN	NaN	51.57	36.25	NaN	33.64	21.29	44.98	NaN	NaN	NaN	0.64	0.58	2.80
35	NaN	NaN	NaN	NaN	NaN	NaN	65.12	40.16	NaN	15.32	17.66	20.64	NaN	NaN	NaN	0.48	0.41	0.58
36	NaN	NaN	NaN	NaN	NaN	NaN	186.31	209.07	NaN	136.10	50.84	165.77	NaN	NaN	NaN	1.42	1.54	5.24
37	NaN	NaN	NaN	NaN	NaN	NaN	212.62	NaN	NaN	50.16	155.81	22.39	NaN	NaN	NaN	2.59	2.50	2.01
38	76.63	76.14	71.28	NaN	NaN	NaN	1.44	0.59	0.50	9.66	9.27	9.16	NaN	NaN	NaN	0.66	0.11	0.53
39	NaN	NaN	NaN	NaN	NaN	NaN	49.95	34.84	NaN	18.57	24.80	27.78	NaN	NaN	NaN	0.36	0.45	0.76
40	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	7.39	NaN	135.91
41	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	170.95	NaN	NaN	NaN	NaN	NaN	24.57	22.40	107.02
42	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	216.44	NaN	NaN	NaN	NaN	5.66	7.47	5.07
43	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	133.79	65.11	228.36	NaN	NaN	NaN	4.01	5.28	4.68
44	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.75	3.74	5.73
45	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	95.16	144.01	104.32	NaN	NaN	NaN	2.36	2.11	2.73
46	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	10.64	12.03	13.62	49.16	69.87	49.84	1.23	1.42	2.01

Table A.4: Solution time for CBC with 18 strategies for 48 test problems with blocks

Test #	QBB	QBN	QBS	QNB	QNN	QNS	QSB	QSN	QSS	NBB	NBN	NBS	NNB	NNN	NNS	NSB	NSN	NSS
1	7.94	14.92	12.48	NaN	NaN	NaN	1.78	5.44	0.53	3.22	4.95	4.57	NaN	NaN	NaN	0.47	1.61	0.17
2	84.62	29.96	30.63	NaN	NaN	NaN	0.49	127.72	1.00	21.67	6.74	3.85	NaN	NaN	217.71	0.29	10.10	0.25
3	106.41	89.66	108.05	NaN	NaN	NaN	25.03	NaN	7.28	45.20	58.22	4.96	NaN	NaN	NaN	3.70	19.47	1.99
4	NaN	NaN	NaN	NaN	NaN	NaN	199.37	NaN	191.11	102.06	207.69	40.13	NaN	NaN	NaN	20.91	NaN	2.94
5	NaN	NaN	NaN	NaN	NaN	NaN	162.16	NaN	17.63	27.07	86.68	59.04	NaN	NaN	NaN	7.68	27.18	3.84
6	86.93	NaN	NaN	NaN	NaN	NaN	5.75	NaN	153.05	15.15	91.81	19.47	NaN	NaN	NaN	1.26	21.18	0.69
7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	74.79	NaN	50.99	NaN	NaN	NaN	22.06	24.67	38.00
10	NaN	NaN	NaN	NaN	NaN	NaN	89.99	NaN	NaN	44.77	21.55	28.32	NaN	NaN	NaN	2.32	18.69	9.74
11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	100.17	184.09	149.99	NaN	NaN	NaN	36.60	59.15	22.68
12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	55.86	231.40	87.56	NaN	NaN	NaN	80.08	NaN	27.12
13	NaN	167.28	NaN	NaN	NaN	NaN	12.29	NaN	22.33	14.79	11.78	15.24	NaN	NaN	NaN	2.28	9.18	1.83
14	31.92	59.55	30.01	NaN	NaN	NaN	1.33	38.83	1.53	40.16	8.86	47.30	NaN	NaN	NaN	0.74	0.70	0.36
15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	92.78	172.95	19.83	NaN	NaN	NaN	26.32	30.48	24.78
16	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
17	NaN	NaN	NaN	NaN	NaN	NaN	17.16	NaN	11.57	107.00	78.94	106.88	NaN	NaN	NaN	1.71	32.75	0.99
18	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	113.88	NaN	NaN	NaN	NaN	NaN	62.15	94.80	153.02
19	NaN	NaN	NaN	NaN	NaN	NaN	25.78	NaN	55.81	139.14	144.08	36.36	NaN	NaN	NaN	3.61	62.36	1.07
20	NaN	NaN	NaN	NaN	NaN	NaN	175.26	NaN	NaN	NaN	NaN	121.82	NaN	NaN	NaN	7.34	58.31	57.15
21	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	222.24	NaN	10.72
22	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
23	NaN	NaN	NaN	NaN	NaN	NaN	32.30	NaN	26.36	46.26	192.51	86.85	NaN	NaN	NaN	2.78	49.74	2.46
24	144.72	NaN	NaN	NaN	NaN	NaN	4.49	NaN	9.16	8.78	5.43	27.27	228.01	151.99	NaN	0.59	6.39	0.67
25	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	112.01	56.22	76.39	52.62	NaN	NaN	NaN	17.29	37.68	5.51
26	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	102.68	NaN	89.12	NaN	NaN	NaN	97.88	96.15	93.68
27	NaN	NaN	NaN	NaN	NaN	NaN	24.09	NaN	21.63	37.81	13.90	33.54	NaN	NaN	NaN	1.81	11.62	1.84
28	NaN	NaN	NaN	NaN	NaN	NaN	41.81	NaN	8.86	18.80	107.69	22.53	NaN	NaN	NaN	0.84	10.58	0.89
29	125.53	129.79	114.93	NaN	NaN	NaN	4.34	NaN	4.40	23.74	17.82	25.01	NaN	NaN	NaN	0.47	1.66	0.43
30	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	114.71	117.30	25.19	NaN	NaN	NaN	28.48	45.83	11.26
31	137.33	134.24	178.72	NaN	NaN	NaN	5.34	NaN	5.97	34.70	17.44	16.66	NaN	NaN	NaN	0.50	0.50	0.45
32	NaN	NaN	NaN	NaN	NaN	NaN	35.37	NaN	58.96	44.59	67.66	41.11	NaN	NaN	NaN	2.64	65.39	3.54
33	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	99.76	29.52	19.02	37.11	NaN	NaN	NaN	14.04	23.46	7.82
34	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	216.00	160.99	135.16	NaN	NaN	NaN	14.73	86.14	27.83
35	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	65.16	149.84	80.59	NaN	NaN	NaN	27.01	56.27	9.78
36	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	122.57	NaN	NaN	NaN	NaN	NaN	80.26	134.80	96.22
37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	147.59	NaN	230.79	NaN	NaN	NaN	110.57	NaN	60.43
38	59.80	94.50	81.45	NaN	NaN	NaN	1.78	1.65	1.77	11.87	17.77	13.85	NaN	NaN	NaN	1.53	0.44	0.53
39	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	93.20	20.81	120.71	50.20	NaN	NaN	157.72	20.95	35.08	5.34
40	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
41	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
42	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	127.43	NaN	NaN	NaN	208.73	NaN	111.51
43	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	101.28	195.48	91.70
44	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	92.31	221.35	21.92
45	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	211.04	NaN	NaN	NaN	15.44	116.83	8.25
46	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	28.65	31.34	28.30	47.49	201.13	163.63	5.12	45.18	4.43

Table A.5: Solution time for CPLEX with 6 strategies for 48 test problems with blocks for multimaterial

Test #	**QSB**	**QSN**	**QSS**	**NSB**	**NSN**	**NSS**
1	1.611	2.551	2.658	0.54	0.491	0.641
2	1.733	1.729	3.252	0.315	0.37	0.476
3	11.101	12.924	27.801	1.166	1.15	1.82
4	144.51	187.665	NaN	7.579	21.74	10.832
5	15.376	15.218	43.646	1.449	1.412	2.73
6	NaN	175.466	NaN	7.028	5.507	6.925
7	NaN	NaN	NaN	42.762	NaN	113.876
8	NaN	NaN	NaN	125.299	NaN	241.943
9	32.174	98.9	135.887	3.771	3.05	11.003
10	16.287	16.69	86.277	1.269	1.085	1.674
11	208.916	214.335	NaN	4.24	6.295	4.961
12	509.851	577.067	NaN	5.358	10.068	6.342
13	10.091	9.078	29.145	0.909	0.993	1.637
14	8.547	2.881	8.911	0.506	0.403	0.62
15	77.049	47.665	238.032	2.523	3.176	3.601
16	NaN	NaN	NaN	19.82	202.195	17.395
17	21.439	31.844	143.385	1.988	1.79	2.249
18	318.855	370.849	NaN	20.114	29.77	45.046
19	NaN	430.451	NaN	4.022	6.429	5.086
20	265.369	330.761	NaN	18.988	25.7	60.412
21	NaN	NaN	NaN	28.216	40.785	45.003
22	NaN	NaN	NaN	26.515	92.717	74.644
23	197.132	220.926	596.321	12.628	19.769	32.604
24	8.105	6.591	61.082	0.992	0.855	0.779
25	90.885	126.161	469.719	3.972	4.225	6.601
26	NaN	NaN	NaN	23.249	46.813	37.171
27	37.006	52.907	106.244	3.054	2.324	3.749
28	38.6	88.74	240.214	2.301	2.148	2.583
29	13.463	13.805	36.571	1.842	1.853	2.149
30	223.456	282.791	NaN	4.706	6.533	5.817
31	80.338	108.861	106.847	7.014	7.347	10.067
32	168.785	162.139	565.35	10.863	11.766	18.709
33	237.192	278.53	NaN	7.697	9.289	11.594
34	NaN	NaN	NaN	20.627	19.715	54.811

Continued on next page

Table A.5 – continued from previous page

Test #	QSB	QSN	QSS	NSB	NSN	NSS
35	248.307	418.303	NaN	7.933	9.46	9.372
36	NaN	NaN	NaN	34.481	44.471	96.462
37	NaN	NaN	NaN	25.11	54.156	33.075
38	17.273	17.247	17.38	2.749	2.817	2.827
39	100.066	130.871	NaN	3.013	4.826	7.016
40	NaN	NaN	NaN	NaN	NaN	NaN
41	NaN	NaN	NaN	304.881	NaN	511.828
42	NaN	NaN	NaN	62.56	391.583	84.689
43	NaN	NaN	NaN	74.586	135.362	100.502
44	NaN	NaN	NaN	NaN	NaN	NaN
45	NaN	NaN	NaN	28.381	36.46	35.489
46	572.097	NaN	NaN	8.617	8.713	12.912

Table A.6: Solution time for CPLEX with two strategies for 16 test problems without blocks for multimaterial

Test #	QS	NS
1	0.307	0.16
2	0.685	0.271
3	9.62	4.192
4	0.686	0.301
5	2.63	1.78
6	9.117	1.454
7	25.994	3.433
8	17.785	3.108
9	65.313	7.668
10	16.316	2.54
11	NaN	NaN
12	NaN	NaN
13	270.647	15.958
14	44.071	1.401

Table A.7: Test road configuration

Road	**Section Length**	**Section Count**	**Size in kilometre**
A	20	50	1
B	100	50	5
C	20	100	2
D	20	150	3
E	100	150	15
F	100	200	20
G	20	450	9

Table A.8: Test configuration of 60 test problems in terms of number of access roads, blocks, and offset levels

Test #	**Road**	**Access Rd**	**Blocks**	**Offset Levels**
1	A	1	3	10
2	A	2	5	10
3	A	5	10	10
4	A	20	25	10
5	A	4	12	10
6	B	10	15	10
7	B	4	20	10
8	B	8	40	10
9	C	2	7	10
10	C	3	8	10
11	C	13	19	10
12	C	17	22	10
13	C	1	5	10
14	C	2	3	10
15	C	4	12	10
16	C	20	30	10
17	C	5	10	16
18	D	3	12	10
19	D	9	10	10
20	D	2	7	10
21	D	25	30	10
22	D	5	20	10
23	D	3	8	10
24	D	4	5	6
25	E	7	9	10
26	E	10	21	10
27	E	2	5	10
28	E	6	8	10
29	E	1	3	10
30	E	10	15	10
31	F	1	2	10
32	F	2	5	10
33	F	4	8	10
34	F	2	10	10

Continued on next page

Table A.8 – continued from previous page

Test #	Road	Access Rd	Blocks	Offset Levels
35	F	8	12	10
36	F	5	15	10
37	F	10	20	10
38	F	1	1	10
39	F	4	7	6
40	G	25	40	20
41	G	15	30	10
42	G	20	22	10
43	G	10	15	10
44	G	5	10	20
45	G	7	9	10
46	G	3	4	4
47	A	10	0	40
48	A	1	0	30
49	B	1	0	40
50	C	1	0	16
51	C	1	0	24
52	D	1	0	16
53	D	1	0	22
54	E	1	0	24
55	F	1	0	16
56	F	1	0	10
57	G	1	0	16
58	G	1	0	30
59	G	1	0	10
60	G	1	0	4

Table A.9: Test configuration for 60 test problems with patterns of blocks and access roads associated with sections

Test #	Road	Offset Levels	Access Roads	Blocks
1	A	10	1	10;25;37
2	A	10	15;40	10;20;30;43;45
3	A	10	1;10;19;30;50	2;3;7;13;17;24;25;40;41;45
4	A	10	1;3;5;7;9;11;15;17;20;23;26;29;33;35;38;41;43;46;48;50	2;4;6;8;10;12;13;14;16;18;19;21;22;24;27;28;30;31;32;36;40;42;44;47;49
5	A	10	15;30;40;50	4;5;9;10;20;21;22;23;24;35;45;46
6	B	10	1;5;10;15;20;25;30;35;40;50	2;3;7;13;18;23;28;33;34;38;43;44;45;46;49
7	B	10	1;25;30;50	4;5;6;9;12;14;15;16;19;20;21;26;27;33;34;38;40;45;46;47
8	B	10	4;10;15;21;28;36;42;48	1;2;3;5;6;7;8;9;11;12;13;14;16;17;18;19;20;22;23;24;25;26;27;29;30;31;32;33;34;35;37;39;40;41;43;44;45;46;47;49
9	C	10	1;100	20;30;40;41;55;60;70
10	C	10	30;50;80	20;25;40;60;61;70;85;90;
11	C	10	1;10;20;25;30;40;45;50;60;70;80;90;100	5;6;14;15;22;27;33;34;35;43;48;55;56;65;68;73;75;85;96
12	C	10	1;7;14;20;25;30;35;44;50;55;60;66;70;75;80;90;100	4;5;10;11;16;22;23;27;33;38;40;41;48;53;58;64;68;73;78;81;82;94
13	C	10	50	10;20;30;60;70
14	C	10	20;50	10;30;60
15	C	10	10;30;50;80	5;6;17;25;35;40;45;55;60;85;90;95
16	C	10	1;5;10;15;20;25;30;35;45;50;55;60;65;70;75;80;85;90;95;100	2;3;7;8;12;17;23;28;33;36;40;41;42;48;53;54;58;62;63;67;72;73;77;78;83;84;88;92;97;98
17	C	16	5;25;45;65;80	2;10;15;20;30;31;35;50;70;90
18	D	10	20;80;130	10;30;35;40;50;60;70;90;100;110;140;145
19	D	10	1;10;25;35;45;55;89;97;123	7;15;30;39;51;60;70;93;109;120
20	D	10	1;150	20;40;50;55;80;100;120
21	D	10	1;10;15;20;25;30;35;40;50;55;60;70;75;80;85;90;100;105;110;120;125;130;135;140;150	5;6;7;12;17;23;27;34;42;48;53;58;66;67;68;73;78;83;88;93;94;102;107;113;114;123;127;133;137;145
22	D	10	20;40;70;100;120	5;6;10;15;25;30;34;36;45;55;60;80;85;90;95;110;115;125;130;140
23	D	10	1;80;150	10;15;20;30;50;90;100;130
24	D	6	10;40;90;130	5;20;60;100;140
25	E	10	1;20;30;50;70;90;120	10;25;40;41;80;100;130;131;140
26	E	10	1;25;35;45;55;65;75;95;125;135	5;10;15;30;40;43;50;60;70;71;80;81;85;90;100;105;115;130;131;140;141
27	E	10	30;90	10;40;50;70;120
28	E	10	20;50;90;120;130;150	10;25;26;27;70;100;125;140
29	E	10	50	20;40;100
30	E	10	1;30;45;60;70;88;101;123;130;150	10;11;20;40;51;65;78;92;109;110;128;140;141;142;143
31	F	10	1	50;100
32	F	10	1;200	10;20;40;60;100
33	F	10	1;50;100;150	10;20;70;80;120;129;140;160
34	F	10	50;100	10;20;21;22;23;30;31;60;90;130
35	F	10	1;24;44;58;77;98;120;150	9;10;11;30;32;50;60;80;100;130;140;170
36	F	10	1;50;90;140;200	10;20;30;60;70;88;101;120;121;122;130;150;170;189;182
37	F	10	10;20;40;60;80;100;120;140;160;180	5;15;30;31;32;50;55;70;71;90;110;130;131;149;150;170;172;189;190;191
38	F	10	1	100
39	F	6	1;50;130;200	30;70;90;100;120;150;180
40	G	20	1;13;20;33;40;55;66;77;90;100;120;130;150;160;190;220;240;260;280;290;300;330;340;350;360	7;8;9;16;25;35;50;60;61;70;84;88;95;96;105;110;115;125;135;140;144;155;165;170;180;185;200;210;230;250;270;285;295;310;320;335;345;355;370;380
41	G	10	1;30;70;100;130;150;170;200;230;250;270;290;330;350;370	10;15;20;40;45;55;60;61;62;80;110;120;140;160;180;185;190;198;211;220;225;245;265;280;300;310;320;340;360;380
42	G	10	10;25;40;55;70;85;100;115;140;155;165;175;190;220;250;280;300;330;360;380	4;5;15;30;50;60;65;75;90;105;125;150;160;170;200;210;240;270;290;310;340;370
43	G	10	1;30;60;90;130;160;190;250;300;350	10;15;20;50;80;110;150;170;220;230;240;280;290;330;380
44	G	20	50;100;150;250;330	10;20;70;120;130;170;200;300;310;350
45	G	10	1;50;90;130;160;260;350	10;30;70;100;120;150;200;290;380
46	G	4	1;100;300	50;120;150;200;330
47	A	40	1;5;10;15;20;30;35;40;45;50	X
48	A	30	1	X
49	B	40	1	X
50	C	16	1	X
51	C	24	1	X
52	D	16	1	X
53	D	22	1	X
54	E	24	1	X
55	F	16	1	X
56	F	10	1	X
57	G	16	1	X
58	G	30	1	X
59	G	10	1	X
60	G	4	1	X

Table A.10: A sample input table for multi-material

Station	X	Y	h	Offset	M1-Cut	M1-Fill	M1-Type	M2-Cut	M2-Fill	M2-Type	M3-Cut	M3-Fill	M3-Type	M4-Cut	M4-Fill	M4-Type
0	392201.6	6983092	311.88	-10	11.48	0	GR	32.93	0	HP	137.6	0	SR	0	0	OR
0	392201.6	6983092	311.88	-8	10.48	0	GR	29.93	0	HP	97.2	0	SR	0	0	OR
0	392201.6	6983092	311.88	-6	9.48	0	GR	26.93	0	HP	60.8	0	SR	0	0	OR
0	392201.6	6983092	311.88	-4	8.48	0	GR	23.93	0	HP	28.4	0	SR	0	0	OR
0	392201.6	6983092	311.88	-2	7.48	0	GR	20.93	0	HP	0	0	SR	0	0	OR
0	392201.6	6983092	311.88	0	0	0	GR	0	0	HP	0	0	SR	0	0	OR
0	392201.6	6983092	311.88	2	0	-32.4	GR	0	0	HP	0	0	SR	0	0	OR
0	392201.6	6983092	311.88	4	0	-76.8	GR	0	0	HP	0	0	SR	0	0	OR
0	392201.6	6983092	311.88	6	0	-133.2	GR	0	0	HP	0	0	SR	0	0	OR
0	392201.6	6983092	311.88	8	0	-201.6	GR	0	0	HP	0	0	SR	0	0	OR
0	392201.6	6983092	311.88	10	0	-282	GR	0	0	HP	0	0	SR	0	0	OR
20	392203.3	6983112	313.24	-10	11.48	0	GR	32.93	0	HP	137.6	0	SR	0	0	OR
20	392203.3	6983112	313.24	-8	10.48	0	GR	29.93	0	HP	97.2	0	SR	0	0	OR
20	392203.3	6983112	313.24	-6	9.48	0	GR	26.93	0	HP	60.8	0	SR	0	0	OR
20	392203.3	6983112	313.24	-4	8.48	0	GR	23.93	0	HP	28.4	0	SR	0	0	OR
20	392203.3	6983112	313.24	-2	7.48	0	GR	20.93	0	HP	0	0	SR	0	0	OR
20	392203.3	6983112	313.24	0	0	0	GR	0	0	HP	0	0	SR	0	0	OR
20	392203.3	6983112	313.24	2	0	-32.4	GR	0	0	HP	0	0	SR	0	0	OR
20	392203.3	6983112	313.24	4	0	-76.8	GR	0	0	HP	0	0	SR	0	0	OR
20	392203.3	6983112	313.24	6	0	-133.2	GR	0	0	HP	0	0	SR	0	0	OR
20	392203.3	6983112	313.24	8	0	-201.6	GR	0	0	HP	0	0	SR	0	0	OR
20	392203.3	6983112	313.24	10	0	-282	GR	0	0	HP	0	0	SR	0	0	OR
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.