Models and strategies for efficiently determining an optimal vertical alignment of roads

Warren Hare, Shahadat Hossain, Yves Lucet, Faisal Rahman

April 8, 2013

Abstract

Selecting an 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 the vertical alignment, so a good vertical alignment can have a profound impact on final construction costs. In this research, we improve the performance of a previous mixed-integer linear programming model, and we propose a new quasi network flow model. Both models use a piecewise quadratic curve to compute the minimum cost vertical alignment and take earthwork operations into account. The models consider several features such as side-slopes, and physical blocks in the terrain. In addition to improving the precision, we propose several techniques that speed up the search for a solution, so that it is possible to make interactive design tools. We report numerical tests that validate the accuracy of the models, and reduce the calculation time.

1 Introduction

Road design refers to the problem of connecting two 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 subproblem in forest management [CMW98], and in the design of efficient highway networks [ACGRSV13].

The problem of designing roads can be split into three interrelated sub-problems [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 paper we propose two new models for the vertical alignment problem (incorporating the earthwork problem) that can be solved by deterministic optimization algorithms.

Nondeterministic methods to compute a vertical alignment were considered for example by Lee and Cheng [LC01] who proposed a three layered heuristic method for vertical alignment optimization, and by Fwa et al. [FCS02] who presented a genetic algorithm formulation of the vertical alignment problem. In [GLA09], Goktepe et al. proposed a vertical alignment
model using genetic algorithms. 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 trial and error method by enumerating all possible combinations of feasible grades for vertical alignment considering earthwork cost. Goh et al. [GCF88] used dynamic programming to compute an optimal vertical alignment. Moreb [Mor96] proposed a model combining vertical alignment and earthwork operations in a single linear program that outputs the alignment as a piecewise-linear function. Moreb and Aljohani [MA04] improved 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. Koch and Lucet [KL10] improved Moreb’s model [Mor09] by reducing the error in the slope constraint. Hare, Koch, and Lucet [HKL11] were the first to propose a mixed integer linear programming model for earthwork optimization that handles blocks. In [HLR13], Hare, Lucet, and Rahman proposed a mixed integer linear programming model for vertical alignment considering side slopes and blocks. Ahuja et al. [AMO93, p. 12-13] give a network flow model for earthwork optimization without blocks.

In this research, we extend the previous model [HLR13] and propose a new quasi network flow model based on [AMO93, p. 12-13] for the vertical alignment optimization problem considering earthwork costs while satisfying industry standards and rules (supplied by our industry partner Softree Technical Systems Ltd.). This quasi network flow model incorporates side slopes, physical blocks and other design constraints of the mixed integer linear program proposed by Hare, Lucet, and Rahman [HLR13]. Using a test set of 60 roads, we numerically examine several techniques to reduce computation time. Our tests clearly demonstrate the speed advantage of the new quasi network flow model presented in this work. Since the quasi network flow model puts restrictions on the cost function, we also report on several techniques to speed up the search for a solution of the mixed-integer linear programming model.

In this research, we extend the previous model [HLR13] and propose a new quasi network flow model based on [AMO93, p. 12-13] for the vertical alignment optimization problem considering earthwork costs while satisfying industry standards and rules (supplied by our industry partner Softree Technical Systems Ltd.). This quasi network flow model incorporates side slopes, physical blocks and other design constraints of the mixed integer linear program proposed by Hare, Lucet, and Rahman [HLR13]. Using a test set of 60 roads, we numerically examine several techniques to reduce computation time. Our tests clearly demonstrate the speed advantage of the new quasi network flow model presented in this work. Since the quasi network flow model puts restrictions on the cost function, we also report on several techniques to speed up the search for a solution of the mixed-integer linear programming model.

In Section 2, we define the variables for both models. Then, in Subsection 2.1, we describe and improve the previously designed mixed integer linear programming model for vertical alignment optimization. In Subsection 2.2, we introduce the new quasi network flow model for vertical alignment optimization. We describe several techniques to improve the solution time in Section 3. In Section 4, we show numerical results and deduce the most efficient techniques. In Section 4.3, we discuss the results and the drawbacks of the models considered. Section 5 concludes the paper with directions for future research.

2 Model description

In this section we describe two models that solve the vertical alignment problem for a known horizontal alignment. The variables, parameters, and sets will be defined as they appear.

In both models, 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 1, a ground profile and a vertical alignment of a sample road are shown.

In order to model the vertical alignment, we split the road into several sections. Let $S = \{1, 2, ..., n\}$ be the index set for the sections. Every section $i (\in S)$ has a station number $s_i$ and

http://www.softree.com
Figure 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. Cross-section areas are given at each station \( s_i \).

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 used to approximate the volume of materials by multiplying with the cross section area. 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 named \( h_i \). 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 problem size becomes large.

The entire road profile is represented as a quadratic spline (Figure 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 \( G = \{1, 2, \ldots, \bar{g}\} \) be the index set of the segments and \( S_g = \{1, 2, \ldots, n_g\} \) be the index set of \( n_g \) sections in spline segment \( g \). The function \( \delta : G \times S_g \to S \) 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 \( g \in G, j \in S_g, i \in S \).

Every two consecutive segments have one common station between them (Figure 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 \( g \in G \) 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

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

In general, we can represent the spline function as follows

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

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 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., $s - s_{\delta(g,1)} \forall g \in G$. 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 3 and 4 the cross-section of a road with and without side-slopes for a cut and a fill are shown respectively.
To model the side slopes, the approximation can be made with trapezoid shaped cross-sections with several rectangles (Figure 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 the input data for the model. These volumes are approximated for each offset level from the ground profile. Suppose for each section $i \in S$, $L_i = \{1, 2, ..., n_l\}$ 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 $l \in L_i$ at section $i$. Sample input for a section is shown in Table 1. Let the parameter $R^+_{i,l}$ represent the given cut volume and $R^-_{i,l}$ represent the given fill volume of material respectively for each section $i \in S$ and each level $l \in L_i$.

The relation between cut and fill volume with the height of the offset can be understood clearly from Figure 6. 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 $V^+_i$ ($i \in S$), and $V^-_i$ ($i \in S$) 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 $B = \{1, 2, ..., n_\beta\}$ and $W = \{1, 2, ..., n_\omega\}$ are the two sets of indices that are used to index borrow and waste pits respectively. The function $\vartheta : B \rightarrow S$ maps a borrow pit index to the section index to which it is attached, and the function $\varphi : W \rightarrow S$ 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.

Figure 3: The cross-section of a road with and without side-slopes (for cut).

Figure 4: The cross-section of a road with and without side-slopes (for fill).

Figure 5: Approximation of side-slopes (for a cut).
Table 1: Cut and fill volume of material

<table>
<thead>
<tr>
<th>Level Offset</th>
<th>Cut Volume $R_{i,l}^+$</th>
<th>Fill Volume $R_{i,l}^-$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 -4</td>
<td>262.06</td>
<td>0</td>
</tr>
<tr>
<td>2 -2</td>
<td>168.44</td>
<td>0</td>
</tr>
<tr>
<td>3 0</td>
<td>82.82</td>
<td>0</td>
</tr>
<tr>
<td>4 2</td>
<td>5.28</td>
<td>0.08</td>
</tr>
<tr>
<td>5 4</td>
<td>0</td>
<td>70.06</td>
</tr>
</tbody>
</table>

Figure 6: Approximation of side-slopes (for a cut).

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 the 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 $t + 1$.

2.1 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 variable is $O(n^2)$, where $n$ is the number of sections. This model is an extension of the mixed integer linear programming model proposed by Hare, Lucet, Rahman [HLR13] considering blocks.
and side slopes.

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 \(i \in N\), the index set \(N^i\) which consists of all indices \(j\) such that \(x_{ij}\) is a permitted move,

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

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

\[
N^i = \left\{ j : \begin{array}{ll}
j \in S \cup B, & \text{if } i \in S \\
j \in S & \text{if } i \in B \\
j & \text{if } i \in W \end{array} \right\}.
\]

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

\[
N^2 = \{(i, j) : j \in N^i\}
\]

Note that \(N^2 \neq N \times N\), as \(N^2\) 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 \(\mathcal{T}\) be the set of all time steps. The objective function can be represented as follows, 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,

\[
\min \left( \sum_{i \in S \cup B} p_i V_i^+ + \sum_{i \in S \cup W} q_i V_i^- + \sum_{(i, j) \in N^2, \in \mathcal{T}} r_{ij} x_{ijt} \right),
\]

where \(V_i^+\) (for \(i \in S \cup B\)), and \(V_i^-\) (for \(i \in S \cup W\)) 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 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 below). We call these the balance constraints:

\[
\sum_{j \in N_{i}^{-}\in T} x_{ijt} = V_{i}^{+}, \quad \forall i \in S \cup B. \tag{1}
\]

\[
\sum_{j \in N_{i}^{+}\in T} x_{ijt} = V_{i}^{-}, \quad \forall i \in S \cup W. \tag{2}
\]

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 3).

\[
P(s_{i}) - h_{i} = u_{i}, \quad \forall i \in S \tag{3}
\]

There could be a cut and fill of the same material at the same section according to the input volume table (see Table 1). Usually the cut volume decreases as the height of the offset level \( L_{i,l} \) increases (see Table 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 4,5).

\[
V_{i}^{+} \geq R_{i,l}^{+} + \frac{R_{i,l+1}^{+} - R_{i,l}^{+}}{L_{i,l+1} - L_{i,l}}(u_{i} - L_{i,l}), \quad \forall i \in S, l \in L_{i} \setminus \{n_{i}\}, \tag{4}
\]

\[
V_{i}^{-} \geq R_{i,l}^{-} + \frac{R_{i,l+1}^{-} - R_{i,l}^{-}}{L_{i,l+1} - L_{i,l}}(u_{i} - L_{i,l}), \quad \forall i \in S, l \in L_{i} \setminus \{n_{i}\}. \tag{5}
\]

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 1). These extra volume of materials would cause stockpiling on the road.

For example, in Figure 7, 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 7, 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 4 and 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 \( i \in S \) (Constraints (6),(7)). The binary variable
\( u_i = 0 \)
\( V_i^+ \geq 82.82 \) 
\( V_i^- \geq 0 \)

\( u_i = 2 \)
\( V_i^+ \geq 5.28 \) 
\( V_i^- \geq 0 \)

\( u_i = 0 \)
\( V_i^+ \geq 82.82 \) 
\( V_i^- \geq 0 \)

\( u_i = 0 \)
\( V_i^+ \geq 82.82 \) 
\( V_i^- \geq 0 \)

Figure 7: Stockpiling scenario when constraints 6, 7 are not considered.

\( \nu_{i,l} (\forall i \in S, l \in L_i \setminus \{n_i\}) \) 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 (8)).

\[
V_i^+ \leq R_i^+ + \frac{R_i^+ - R_i^{+l}}{L_i,l+1 - L_{i,l}} (u_i - L_{i,l}) + M(1 - \nu_{i,l}), \quad \forall i \in S, l \in L_i \setminus \{n_i\},
\]

(6)

\[
V_i^- \leq R_i^- + \frac{R_i^- - R_i^{-l}}{L_i,l+1 - L_{i,l}} (u_i - L_{i,l}) + M(1 - \nu_{i,l}), \quad \forall i \in S, l \in L_i \setminus \{n_i\},
\]

(7)

\[
\sum_{l \in L_i \setminus \{n_i\}} \nu_{i,l} = 1, \quad \forall i \in S.
\]

(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 (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 (10)). Similarly, material movement is not possible if there is no access road after a block (Constraint (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 (12)).

\[
x_{ij,t} \leq My_{k_1,t-1}, \quad \forall t \in T \setminus \{0\}, k \in T_{i,j} \in N^k_b,
\]

(9)

\[
x_{ij,t} \leq My_{k_1,t-1}, \quad \forall t \in T \setminus \{0\}, k \in T_{i,j} \in N^k_{\to b},
\]

(10)

\[
x_{ij,t} \leq My_{k_1,t-1}, \quad \forall t \in T \setminus \{0\}, k \in T_{i,j} \in N^k_{\to b},
\]

(11)

\[
x_{ij,t,m} \leq My_{k_1,t-1} + My_{k_2,t-1}, \quad \forall t \in T \setminus \{0\}, (k_1,k_2) \in \mathcal{P}_2, i,j \in N^{k_1,k_2}_b.
\]

(12)

The block removal indicator constraints (13, 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.
At each time step $t \in T$, we want to remove at least one block. This is guaranteed with block removal enforcement constraint (15). The monotonicity constraint (16), states that once a block is removed it remains removed.

\begin{align*}
\sum_{t=0}^{u} \sum_{j \in \mathcal{N}^{\mathcal{I}}} x_{\xi(k)jt} + M(1 - y_{ku}) & \geq V _{\xi(k)}^+, \quad \forall k \in \mathcal{I}, u \in T, \\
\sum_{t=0}^{v} \sum_{j \in \mathcal{N}^{\mathcal{I}}} x_{j\xi(k)t} + M(1 - y_{ku}) & \geq V _{\xi(k)}^-, \quad \forall k \in \mathcal{I}, u \in T.
\end{align*}  

The smoothness constraints (17) and (18) ensure that the transitions from one spline segment to the next are smooth, i.e., same grade and elevation (see Figure 2).

\begin{align*}
P_g - 1(s_{\delta(g,1)}) = P_g(s_{\delta(g,1)}), & \quad \forall g \in \mathcal{G} \setminus \{1\}, \\
P'_{g-1}(s_{\delta(g,1)}) = P'(s_{\delta(g,1)}), & \quad \forall g \in \mathcal{G} \setminus \{1\}.
\end{align*}  

The maximum allowable speed depends on the grade of the road profile. The grade constraint (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.

\begin{align*}
G_L \leq P_g'(s_{\delta(g,1)}) \leq G_U, & \quad \forall g \in \mathcal{G} \setminus \{1\}.
\end{align*}  

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 constraint (20, 21, 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 $i \in \mathcal{H}$.

\begin{align*}
P(s_1) &= y_A, & P(s_n) &= y_B, \\
P'(s_1) &= \bar{y}_A, & P'(s_n) &= \bar{y}_B, \\
P(s_i) &= \mathcal{H}_i, & \forall i \in \mathcal{H}.
\end{align*}  

Finally, (23, 24, 25) are bound constraints. The parameter $M_i^+$, and $M_i^-$ are the largest volume possible to be cut and filled from a section.

\begin{align*}
x_{ijt} & \geq 0, & \forall (i,j) & \in \mathcal{N}^2, t \in T, \\
0 & \leq V_i^+ \leq M_i^+, & \forall i & \in \mathcal{S} \cup \mathcal{B}, \\
0 & \leq V_i^- \leq M_i^-, & \forall i & \in \mathcal{S} \cup \mathcal{W}.
\end{align*}
2.2 quasi network flow model

In this section the quasi network flow model is described for computing an optimal vertical alignment. Although we are calling it the quasi network flow model, it is a modification of the traditional network flow model [AM093, 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 $O(n)$, where $n$ is the number of sections.

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 8).

![Figure 8: A typical section $i$ at time-step $t$ (where $\vartheta(j) = \varphi(k) = i$).](image)

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 nodes for both directions (Figure 8). 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 $f_{i-1,i,t}$ and $f_{i+1,i,t}$ ($\forall i \in S, t \in T$) 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 $f_{i+1,i,t}$ and $f_{i-1,i,t}$ ($\forall i \in S, t \in T$) represent the load (fill) flows of materials to the East and West transit nodes, and within the same section respectively. Similarly, the variables $f_{i+1,i,t}$ and $f_{i-1,i,t}$ ($\forall i \in S, t \in T$) 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 $f_{j,\vartheta(j)-1,t}$ and $f_{j,\vartheta(j)+1,t}$ ($\forall j \in B, t \in T$) 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 $f_{\varphi(j)-1,j,t}$ and $f_{\varphi(j)+1,j,t}$ ($\forall j \in W, t \in T$) be the flows of materials from the
East and West transit nodes to waste pit respectively.

The cost of hauling materials from section \( i \) to section \( i-1 \) (respectively \( i+1 \)) is represented by \( c_{i,i-1} = cd_{i,i-1} \) (respectively \( c_{i,i+1} = cd_{i,i+1} \)), 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,

\[
\min \sum_{i \in S \cup B} p_i V_i^+ + \sum_{i \in S \cup W} q_i V_i^- + \sum_{i \in S} \left( c_{i,i-1} f_{i,i-1,t}^r + c_{i,i+1} f_{i,i+1,t}^r \right) \\
+ \sum_{j \in B, t \in T} c d_j \left( f_{j,\vartheta(j)-1,t}^b + f_{j,\vartheta(j)+1,t}^b \right) \\
+ \sum_{j \in W, t \in T} c d_j \left( f_{\varphi(j)-1,j,t}^w + f_{\varphi(j)+1,j,t}^w \right) .
\]

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 (29)), and for the West transit node (Constraint (30)) ensure that the supply and demand at both transit nodes are equal.

\[
f_{i-1,i,t}^r + f_{i,i+1,t}^u + \sum_{j \in B, \vartheta(j) = i} f_{j,i+1,t}^b = f_{i,i+1,t}^r + f_{i-1,i,t}^l + \sum_{j \in W, \vartheta(j) = i} f_{i+1,j,t}^w \\
\text{for all } t \in T, i \in S, \tag{29}
\]

\[
f_{i+1,i,t}^r + f_{i,i-1,t}^u + \sum_{j \in B, \vartheta(j) = i} f_{j,i-1,t}^b = f_{i,i-1,t}^r + f_{i+1,i,t}^l + \sum_{j \in W, \vartheta(j) = i} f_{i-1,j,t}^w \\
\text{for all } t \in T, i \in S. \tag{30}
\]

The balance constraints ensure that the total unload flows from a section is equal to the cut volume of that section (Constraint (31)), and the total load flows to a section is equal to the fill volume (Constraint (32)). Borrow (Constraint (33)) and waste (Constraint (34)) pits constraints are written similarly.

\[
\sum_{t \in T} f_{i,i-1,t}^u + f_{i,i+1,t}^u + f_{i,i,t}^u = V_i^+, \quad \forall i \in S, \tag{31}
\]

\[
\sum_{t \in T} f_{i-1,i,t}^l + f_{i+1,i,t}^l + f_{i,i,t}^l = V_i^-, \quad \forall i \in S, \tag{32}
\]

\[
\sum_{t \in T} f_{j,\vartheta(j)-1,t}^b + f_{j,\vartheta(j)+1,t}^b = V_j^+, \quad \forall j \in B, \tag{33}
\]

\[
\sum_{t \in T} f_{\varphi(j)-1,j,t}^w + f_{\varphi(j)+1,j,t}^w = V_j^-, \quad \forall j \in W. \tag{34}
\]
The volume constraints, smoothness constraints, grade constraints, and fixed point constraints are the same for this quasi network flow model as described in the quadratic model.

If a section \( i \) is a block, then for the East transit node equations \( f^e_{i-1,i,t} = f^e_{i,i,t} \) and \( f^e_{i,i+1,t} = f^e_{i+1,i,t} \) (\( \forall t \in T \)) ensure that no material can transfer across Section \( i \) to the East direction. Because \( f^e_{i-1,i,t} = f^e_{i-1,i,t} \) implies that when materials are transferred from Section \( i - 1 \) to section \( i \), all the materials are loaded (filled) to section \( i \). And \( f^e_{i,i+1,t} = f^e_{i+1,i,t} \) 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, \( f^w_{i+1,i,t} = f^w_{i+1,i,t} \) and \( f^w_{i-1,i,t} = f^w_{i-1,i,t} \) 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 35, 36, 37, and 38. Similarly, for the West direction, constraints 39, 40, 41, 42 ensure no movement over a block.

\[
-(f^e_{i-1,k-1,}\zeta(k),t) - f^e_{i-1,k-1,}\zeta(k),t) \leq M y_{k,t-1}, \quad \forall k \in I, t \in T, \quad (35)
\]

\[
f^e_{i-1,k-1,}\zeta(k),t) - f^e_{i-1,k-1,}\zeta(k),t) \leq M y_{k,t-1}, \quad \forall k \in I, t \in T, \quad (36)
\]

\[
-(f^e_{i,k-1,\zeta(k)+1,t} - f^e_{i,k-1,\zeta(k)+1,t}) \leq M y_{k,t-1}, \quad \forall k \in I, t \in T, \quad (37)
\]

\[
f^e_{i,k-1,\zeta(k)+1,t} - f^e_{i,k-1,\zeta(k)+1,t} \leq M y_{k,t-1}, \quad \forall k \in I, t \in T. \quad (38)
\]

\[
-(f^e_{i+1,k-1,\zeta(k),t} - f^e_{i+1,k-1,\zeta(k),t}) \leq M y_{k,t-1}, \quad \forall k \in I, t \in T, \quad (39)
\]

\[
f^e_{i+1,k-1,\zeta(k),t} - f^e_{i+1,k-1,\zeta(k),t} \leq M y_{k,t-1}, \quad \forall k \in I, t \in T, \quad (40)
\]

\[
-(f^e_{i,k-1,\zeta(k)-1,t} - f^e_{i,k-1,\zeta(k)-1,t}) \leq M y_{k,t-1}, \quad \forall k \in I, t \in T, \quad (41)
\]

\[
f^e_{i,k-1,\zeta(k)-1,t} - f^e_{i,k-1,\zeta(k)-1,t} \leq M y_{k,t-1}, \quad \forall k \in I, t \in T. \quad (42)
\]

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 (43), (44)), borrow flow (Constraints (45), (46)), or waste flow (Constraints (47), (48)) between two blocks with no access roads until one of the blocks is removed.

\[
f^e_{i,i+1,t} \leq M(y_{k,t-1} + y_{k+1,t-1}), \quad \forall t \in T, (k_1, k_2) \in \mathbb{I}^2, \quad \zeta(k_1) \leq i, i+1 \leq \zeta(k_2), \quad (43)
\]

\[
f^e_{i+1,i,t} \leq M(y_{k,t-1} + y_{k+1,t-1}), \quad \forall t \in T, (k_1, k_2) \in \mathbb{I}^2, \quad \zeta(k_1) \leq i, i+1 \leq \zeta(k_2), \quad (44)
\]

\[
f^b_{i,j,\theta(j)+1,t} \leq M(y_{k,t-1} + y_{k+1,t-1}), \quad \forall t \in T, (k_1, k_2) \in \mathbb{I}^2, \quad \zeta(k_1) \leq \theta(j), \theta(j) + 1 \leq \zeta(k_2), \quad (45)
\]

\[
f^b_{i,j,\theta(j)-1,t} \leq M(y_{k,t-1} + y_{k+1,t-1}), \quad \forall t \in T, (k_1, k_2) \in \mathbb{I}^2, \quad \zeta(k_1) \leq \theta(j), \theta(j) + 1 \leq \zeta(k_2), \quad (46)
\]

\[
f^w_{i,j,\varphi(j)+1,t} \leq M(y_{k,t-1} + y_{k+1,t-1}), \quad \forall t \in T, (k_1, k_2) \in \mathbb{I}^2, \quad \zeta(k_1) \leq \varphi(j), \varphi(j) + 1 \leq \zeta(k_2), \quad (47)
\]

\[
f^w_{i,j,\varphi(j)-1,t} \leq M(y_{k,t-1} + y_{k+1,t-1}), \quad \forall t \in T, (k_1, k_2) \in \mathbb{I}^2, \quad \zeta(k_1) \leq \varphi(j), \varphi(j) + 1 \leq \zeta(k_2). \quad (48)
\]
There will also be no transit flow (49, 50), borrow flow (51, 52), or waste flow (53, 54) after the last block with no access road until the block is removed.

\[
\begin{align*}
    f_{i+1, i}^{r} & \leq M y_{k_{1}, t-1}, & \forall t \in T, c(k) \leq i, i + 1 \leq n, k \in \bar{I}, \\
    f_{i, i}^{r} & \leq M y_{k_{1}, t-1}, & \forall t \in T, c(k) \leq i, i + 1 \leq n, k \in \bar{I}, \\
    f_{j, \vartheta(j)+1, t}^{b} & \leq M y_{k_{1}, t-1}, & \forall t \in T, c(k) \leq \vartheta(j) - 1, \vartheta(j), \vartheta(j) + 1 \leq n, k \in \bar{I}, \\
    f_{j, \vartheta(j)-1, t}^{b} & \leq M y_{k_{1}, t-1}, & \forall t \in T, c(k) \leq \vartheta(j) - 1, \vartheta(j), \vartheta(j) + 1 \leq n, k \in \bar{I}, \\
    f_{\varphi(j)+1, j, t}^{w} & \leq M y_{k_{1}, t-1}, & \forall t \in T, c(k) \leq \varphi(j) - 1, \varphi(j), \varphi(j) + 1 \leq n, k \in \bar{I}, \\
    f_{\varphi(j)-1, j, t}^{w} & \leq M y_{k_{1}, t-1}, & \forall t \in T, c(k) \leq \varphi(j) - 1, \varphi(j), \varphi(j) + 1 \leq n, k \in \bar{I}.
\end{align*}
\]

Similarly, there will also be no transit flow (55, 56), borrow flow (57, 58), or waste flow (59, 60) before the first block with no access road until the block is removed.

\[
\begin{align*}
    f_{i, i+1}^{r} & \leq M y_{k_{1}, t-1}, & \forall t \in T, 1 \leq i, i + 1 \leq c(k), k \in \bar{I}, \\
    f_{i, i}^{r} & \leq M y_{k_{1}, t-1}, & \forall t \in T, 1 \leq i, i + 1 \leq c(k), k \in \bar{I}, \\
    f_{j, \vartheta(j)+1, t}^{b} & \leq M y_{k_{1}, t-1}, & \forall t \in T, 1 \leq \vartheta(j) - 1, \vartheta(j), \vartheta(j) + 1 \leq c(k), k \in \bar{I}, \\
    f_{j, \vartheta(j)-1, t}^{b} & \leq M y_{k_{1}, t-1}, & \forall t \in T, 1 \leq \vartheta(j) - 1, \vartheta(j), \vartheta(j) + 1 \leq c(k), k \in \bar{I}, \\
    f_{\varphi(j)+1, j, t}^{w} & \leq M y_{k_{1}, t-1}, & \forall t \in T, 1 \leq \varphi(j) - 1, \varphi(j), \varphi(j) + 1 \leq c(k), k \in \bar{I}, \\
    f_{\varphi(j)-1, j, t}^{w} & \leq M y_{k_{1}, t-1}, & \forall t \in T, 1 \leq \varphi(j) - 1, \varphi(j), \varphi(j) + 1 \leq c(k), k \in \bar{I}.
\end{align*}
\]

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 (constraint (61) and (62)).

\[
\begin{align*}
    \sum_{t=0}^{u} \left( f_{c(k), c(k)-1, i, t}^{w} + f_{c(k), c(k)+1, i, t}^{w} \right) + M(1 - y_{ku}) & \geq V_{c(k)}^{+}, & \forall k \in I, u \in T, \\
    \sum_{t=0}^{u} \left( f_{c(k)-1, c(k), i, t}^{w} + f_{c(k)+1, c(k), i, t}^{w} \right) + M(1 - y_{ku}) & \geq V_{c(k)}^{-}, & \forall k \in I, u \in T.
\end{align*}
\]

The block removal enforcement constraint (15) and monotonicity constraint (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.

\[
\begin{align*}
    f_{i,i-1,t}^r &\geq 0, & f_{i,i+1,t}^r &\geq 0, & \forall i \in \mathcal{S}, t \in \mathcal{T}, \\
    f_{i,i-1,t}^l &\geq 0, & f_{i,i+1,t}^l &\geq 0, & \forall i \in \mathcal{S}, t \in \mathcal{T}, \\
    f_{i,i-1,t}^u &\geq 0, & f_{i,i+1,t}^u &\geq 0, & \forall i \in \mathcal{S}, t \in \mathcal{T}, \\
    f_{i,i-1,t}^b &\geq 0, & f_{i,i+1,t}^b &\geq 0, & \forall i \in \mathcal{S}, t \in \mathcal{T},
\end{align*}
\]

(63) (64) (65) (66) (67)

3 Accelerating the search for a solution

Our motivation in this paper is to determine methods to solve the vertical alignment problem in the most efficient manner. In this section we describe four strategies that we used to decrease the time to find a solution.

3.1 Reducing binary variables for offset levels

In the volume constraints (4, 5, 6, 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. In this section we examine a method to reduce the number of binary variables at the cost of adding some continuous variables. We numerically test this new model in Section 4.

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 \(L = \{0, 1, ..., n_l-2\}\) 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:

\[
l = \sum_{k=0}^{\lceil \log_2(n_l-2) \rceil} b_{lk} \cdot 2^k,
\]

where \(b_{lk} \in \{0,1\}\) for each \(l \in \mathcal{L}\). Let \(B^1_l = \{l \in \mathcal{L} : b_{lk} = 1\}\) and \(B^0_l = \{l \in \mathcal{L} : b_{lk} = 0\}\).

Define the set \(\mathcal{L}' = \{0, 1, 2, ..., \lceil \log_2(n_l-2) \rceil\}\) 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 (6, 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 (68) and (69) ensure that only one of \(\Gamma_{i,l}\) (\(\forall l \in \mathcal{L}\)) can equal 1 for each section \(i \in \mathcal{S}\).

\[
\sum_{l \in B^1_k} \Gamma_{i,l} \leq \alpha_{ik}, \quad \forall i \in \mathcal{S}, k \in \mathcal{L}', \tag{68}
\]

\[
\sum_{l \in B^0_k} \Gamma_{i,l} \leq (1 - \alpha_{ik}), \quad \forall i \in \mathcal{S}, k \in \mathcal{L}' \tag{69}
\]

In applying the above, we also note that Constraint (8) is no longer required.
Example Consider the case of 5 input offset levels. Then we have 4 intervals and the set \( L' = \{0, 1\} \). Thus for each section \( i \in S \), we get the following inequalities:

\[
\Gamma_{i,1} + \Gamma_{i,3} \leq \alpha_{i0}, \\
\Gamma_{i,0} + \Gamma_{i,2} \leq 1 - \alpha_{i0}, \\
\Gamma_{i,2} + \Gamma_{i,3} \leq \alpha_{i1}, \\
\Gamma_{i,0} + \Gamma_{i,1} \leq 1 - \alpha_{i1}.
\]

Therefore, if we choose \((\alpha_{i1}, \alpha_{i0}) = (1, 0)\) that represents interval 2 of offset levels, then only \( \Gamma_{i,2} \) can be 1.

Now, if we replace the binary variable \( \nu_{i,l} \) with the continuous indicator variable \( \Gamma_{i,l} \) in the volume constraints (6, 7), and add new constraints (68) and (69), 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 (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. In Section 4, we numerically compare these two modeling approaches.

### 3.2 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].

Let \( \lambda_{i,l} (\forall l \in L_i) \) be the set of SOS2 variables for each section \( i \in S \). These variables denote non-negative weights for all levels such that their sum is 1 (Constraint (70)). The offset variable \( u_i \) can be any point in between two input offset levels \( (l \text{ and } l + 1) \) that can be calculated by the weighted sum of the two adjacent input offset heights (Constraint (71)). Similarly, the cut and fill volumes can be calculated using the weighted sum of input volumes of two adjacent offset levels (constraints (72) and (73)). In this formulation, we do not have to calculate slopes for each interval. As with all methods, this formulation is numerically tested in Section 4.

\[
\sum_{l \in L_i} \lambda_{i,l} = 1, \quad \forall i \in S, \tag{70}
\]

\[
\sum_{l \in L_i} \lambda_{i,l} L_{i,l} = u_i, \quad \forall i \in S, \tag{71}
\]

\[
\sum_{l \in L_i} \lambda_{i,l} R_{i,l}^+ = V_i^+ , \quad \forall i \in S, \tag{72}
\]

\[
\sum_{l \in L_i} \lambda_{i,l} R_{i,l}^- = V_i^- , \quad \forall i \in S. \tag{73}
\]

### 3.3 Block constraints with SOS1 variables

In the basic quadratic and quasi network flow models, the block constraints have monotonicity constraint (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 $k \in I$, $t \in \mathcal{T}$, uptil 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 MIP solvers.

For each block $k \in I$, we introduce a set of SOS1 variables $\gamma_{kt}$ 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 $\gamma_{kt}$ instead of just one $y_{kt}$ per constraint. In particular, the block constraints (9), (10), (11),(12), (13), (14), and (15) of the quadratic model are rewritten using SOS1 variables as constraints (74), (75), (76), (77), (76), and (77).

$$x_{ijt} \leq M \sum_{w=0}^{t-1} \gamma_{kw}, \quad \forall t \in \mathcal{T} \setminus \{0\}, k \in I, (i,j) \in \mathcal{N}_b^k,$$  \hspace{1cm} (74)

$$x_{ijt} \leq M \sum_{w=0}^{t-1} \gamma_{kw}, \quad \forall t \in \mathcal{T} \setminus \{0\}, k \in \bar{I}_-, i, j \in \mathcal{N}_b^{r-k},$$  \hspace{1cm} (75)

$$x_{ijt} \leq M \sum_{w=0}^{t-1} \gamma_{kw}, \quad \forall t \in \mathcal{T} \setminus \{0\}, k \in \bar{I}_+, i, j \in \mathcal{N}_b^{k+},$$  \hspace{1cm} (76)

$$x_{ijt,m} \leq M \sum_{w=0}^{t-1} (\gamma_{k_1,w} + \gamma_{k_2,w}), \quad \forall t \in \mathcal{T} \setminus \{0\}, (k_1,k_2) \in \bar{I}_2^k,$$

$$i, j \in \mathcal{N}_b^{k_1,k_2}.$$  \hspace{1cm} (77)

$$\sum_{t=0}^{u} \sum_{j \in \mathcal{N}_b^{\varsigma(k)}} x_{j\varsigma(k)} t + M \left(1 - \sum_{w=0}^{u} \gamma_{kw}\right) \geq V_{\varsigma(k)}^+ k \in I, u \in T,$$  \hspace{1cm} (78)

$$\sum_{t=0}^{u} \sum_{j \in \mathcal{N}_b^{\varsigma(k)}} x_{j\varsigma(k)} t + M \left(1 - \sum_{w=0}^{u} \gamma_{kw}\right) \geq V_{\varsigma(k)}^- k \in I, u \in T.$$  \hspace{1cm} (79)

The block constraints for the basic quasi network flow model are rewritten in an analogous manner. We test this formulation numerically in Section 4.

### 3.4 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 modelling method uses essentially the same techniques as those discussed in Subsection 3.1 for offset levels. We can represent each time step $t \in \mathcal{T}$ in binary format as follows:

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

where $b_{ij} \in \{0,1\}$.

We then introduce extra continuous variables $\beta_{kt} \in [0,1]$ (for all $t \in \mathcal{T}, k \in I$) to indicate time step $t$ when the block $k$ is removed. The new constraints (80) and (81) ensure that there will always be consecutive ones in $\beta_{kt} (\forall t \in \mathcal{T})$ for each $k \in I$ upto the last time step $t$. This represents that once a block is removed it remains removed.
\[ \sum_{u=0}^{t-1} \beta_{ku} \leq t \sum_{j \in B_i^t} (1 - v_{kj}), \quad \forall k \in \mathcal{I}, t \in \mathcal{T} \setminus \{0, n_b\}, \]  
\[ \sum_{u=0}^{n_b} \beta_{ku} \geq (n_b + 1) - \sum_{j=0}^{\lfloor \log_2 n_b \rfloor} 2^j \cdot v_{kj}, \quad \forall k \in \mathcal{I}. \]  

Therefore, the monotonicity constraint (16) is not needed any more.

**Remark 2.** We no longer require the monotonicity constraint \( \beta_{k(t+1)} \geq \beta_{kt} \) for each \( t \in \mathcal{T} \cap \{1, 2, \ldots, n_b\} \) under this formulation.

This approach reduces the number of binary variables from \( n_b(n_b + 1) \) to \( n_b \lfloor \log_2 n_b \rfloor \), but increases the number of continuous variable by \( n_b(n_b + 1) \) and constraints by \( n_b \). As with all techniques, we examine the impacts of this technique numerically in Section 4.

## 4 Experimental results

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 an industry partner.

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 MIP 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 MIP 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 Subsections 2.1 and 2.2). We have changed both models with some strategies to speedup the solution time. For offset levels, in addition to the basic method, we use a reduced binary variable technique and a SOS2 variable technique (see Subsections 3.1 and 3.2). For the blocks, in addition to the basic way of dealing with blocks, we use a SOS1 variable technique and a reduced \( n \log n \) technique (see Subsections 3.3 and 3.4). In other words, 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). In total we considered 18 different strategies and two solvers. These 18 strategies are shown in Table 2. All the test problems are solved with a 1% MIP gap tolerance and a maximum 4 minute CPU time.

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 by

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

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

\[ \rho_s(\tau) = \frac{|\{p \in \mathcal{P} : \log_2(r_{p,s}) \leq \tau\}|}{|\mathcal{P}|} \times 100\%. \]
Table 2: List of strategies with blocks

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Model</th>
<th>Level Technique</th>
<th>Block Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>QBB</td>
<td>Quadratic</td>
<td>Basic</td>
<td>Basic</td>
</tr>
<tr>
<td>QBN</td>
<td>Quadratic</td>
<td>Basic</td>
<td>n log n</td>
</tr>
<tr>
<td>QBS</td>
<td>Quadratic</td>
<td>Basic</td>
<td>SOS1</td>
</tr>
<tr>
<td>QNB</td>
<td>Quadratic</td>
<td>Binary reduction</td>
<td>Basic</td>
</tr>
<tr>
<td>QNN</td>
<td>Quadratic</td>
<td>Binary reduction</td>
<td>n log n</td>
</tr>
<tr>
<td>QNS</td>
<td>Quadratic</td>
<td>Binary reduction</td>
<td>SOS1</td>
</tr>
<tr>
<td>QSB</td>
<td>Quadratic</td>
<td>SOS2</td>
<td>Basic</td>
</tr>
<tr>
<td>QSN</td>
<td>Quadratic</td>
<td>SOS2</td>
<td>n log n</td>
</tr>
<tr>
<td>QSS</td>
<td>Quadratic</td>
<td>SOS2</td>
<td>SOS1</td>
</tr>
<tr>
<td>NBB</td>
<td>quasi network flow</td>
<td>Basic</td>
<td>Basic</td>
</tr>
<tr>
<td>NBN</td>
<td>quasi network flow</td>
<td>Basic</td>
<td>n log n</td>
</tr>
<tr>
<td>NBS</td>
<td>quasi network flow</td>
<td>Basic</td>
<td>SOS1</td>
</tr>
<tr>
<td>NNB</td>
<td>quasi network flow</td>
<td>Binary reduction</td>
<td>Basic</td>
</tr>
<tr>
<td>NNN</td>
<td>quasi network flow</td>
<td>Binary reduction</td>
<td>n log n</td>
</tr>
<tr>
<td>NNS</td>
<td>quasi network flow</td>
<td>Binary reduction</td>
<td>SOS1</td>
</tr>
<tr>
<td>NSB</td>
<td>quasi network flow</td>
<td>SOS2</td>
<td>Basic</td>
</tr>
<tr>
<td>NSN</td>
<td>quasi network flow</td>
<td>SOS2</td>
<td>n log n</td>
</tr>
<tr>
<td>NSS</td>
<td>quasi network flow</td>
<td>SOS2</td>
<td>SOS1</td>
</tr>
</tbody>
</table>

The percentage $\rho_s$ of all problems solved by strategy $s$ at $\tau$ gives us an overall assessment of the performance of strategy $s$. High values of $\rho_s$ near $\tau = 1$ represents fast solve times. High values of $\rho_s$ when $\tau$ 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].

Out of the 60 test problems, 46 test problems contain blocks and 14 test problems are setup without blocks. In Table 6 and 7 the solution times required by solvers CPLEX and CBC respectively are listed for 18 different strategies for the 46 test problems with blocks. In table 8 and 9 the solution times are listed for the 14 test problems with no blocks for 6 different techniques (see Table 3) for both MIP solvers.

4.1 Results for problems with blocks

The performance profiles of 18 different strategies with solver CPLEX are shown in Figure 9. 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.

Similarly, in the case of CBC, the performance profiles of 18 different strategies are shown in Figure 10. 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 takes a little bit more time.
Figure 9: Performance profile for solver CPLEX (with blocks)

Figure 10: Performance profile for solver CBC (with blocks)
From Figures 9 and 10, 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 3.3 and 3.4), we see that the SOS1 technique works better than the other techniques for CBC, whereas the basic technique works well for CPLEX.

Figure 11: Performance profile of different strategies of the quadratic model for solver CPLEX (with blocks)

Figure 12: Performance profile of different strategies of the quadratic model for solver CBC (with blocks)

In Figures 11, 12, 13, and 14 the performance profiles are shown with various techniques separately for the quadratic and quasi network flow models for each MIP solver. For CBC (Figure 12, 14), the SOS2 level technique and the SOS1 block technique work better than the other strategies by a slight margin. However, for CPLEX (Figure 11, 13), the SOS2 level technique combined with the basic block technique works better than other strategies.
Figure 13: Performance profile of different strategies of the quasi network flow model for solver CPLEX (with blocks)

Figure 14: Performance profile of different strategies of the quasi network flow model for solver CBC (with blocks)
4.2 Results for problems with no blocks

For problems without blocks, the list of strategies is shown in Table 3. The run times for these 6 strategies in the case of both solvers are listed in Table 8 and 9. The performance profiles for CPLEX and CBC for these 6 strategies are shown in Figure 15 and 16 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>
<thead>
<tr>
<th>Strategy</th>
<th>Model</th>
<th>Level Trick</th>
</tr>
</thead>
<tbody>
<tr>
<td>QB</td>
<td>Quadratic</td>
<td>Basic</td>
</tr>
<tr>
<td>QN</td>
<td>Quadratic</td>
<td>Binary reduction</td>
</tr>
<tr>
<td>QS</td>
<td>Quadratic</td>
<td>SOS2</td>
</tr>
<tr>
<td>NB</td>
<td>quasi network flow</td>
<td>Basic</td>
</tr>
<tr>
<td>NN</td>
<td>quasi network flow</td>
<td>Binary reduction</td>
</tr>
<tr>
<td>NS</td>
<td>quasi network flow</td>
<td>SOS2</td>
</tr>
</tbody>
</table>

4.3 Summary of experimental result

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.

- 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 with SOS2 levels 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 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 4, 5 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 4: 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>quasi network flow</th>
<th>Quadratic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level Technique</strong></td>
<td>SOS2 Basic Binary Reduction</td>
<td>SOS2 Basic Binary Reduction</td>
</tr>
<tr>
<td><strong>Block Technique</strong></td>
<td>Basic SOS1 n log n</td>
<td>Basic SOS1 n log n</td>
</tr>
</tbody>
</table>

Table 5: 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>quasi network flow</th>
<th>Quadratic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level Technique</strong></td>
<td>SOS2 Basic Binary Reduction</td>
<td>SOS2 Basic Binary Reduction</td>
</tr>
<tr>
<td><strong>Block Technique</strong></td>
<td>SOS1 Basic n log n</td>
<td>SOS1 Basic n log n</td>
</tr>
</tbody>
</table>

5 Conclusion

In this paper, we develop two models for optimal design of the vertical alignment considering the earthwork cost. The first, quadratic, model is a minor enhancement of work from Hare, Lucet, and Rahman [HLR13]. The second, quasi network flow, model reworks the model to create a linear number of continuous variables, at the cost of a more restrictive objective function. Both models incorporate features such as sides slopes and physical blocks.

We further introduce six potential techniques to improve the solution speed that are applicable for both models. Numerical tests shows that using the best combination of techniques
can consistently generate major improvements in solution time, regardless of which LP solver is used. For example, the quasi network flow model using the best speed techniques improves solution speed from the basic quadratic model by a factor of 173.

The techniques developed in this paper have made it possible to solve a vertical alignment design for a very long road in seconds. Using these improvement techniques, our industry partner (Softree Technical Systems Ltd.) has developed an interactive road design software.
<table>
<thead>
<tr>
<th>Test #</th>
<th>QBB</th>
<th>QNB</th>
<th>QSB</th>
<th>QNN</th>
<th>QNS</th>
<th>QSN</th>
<th>NSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.45</td>
<td>3.14</td>
<td>2.45</td>
<td>11.69</td>
<td>16.11</td>
<td>10.95</td>
<td>0.27</td>
</tr>
<tr>
<td>2</td>
<td>3.03</td>
<td>3.46</td>
<td>3.57</td>
<td>70.85</td>
<td>133.41</td>
<td>123.73</td>
<td>0.40</td>
</tr>
<tr>
<td>3</td>
<td>9.55</td>
<td>26.27</td>
<td>24.89</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>1.46</td>
</tr>
<tr>
<td>4</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>20.88</td>
</tr>
<tr>
<td>5</td>
<td>43.44</td>
<td>23.92</td>
<td>49.00</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>2.53</td>
</tr>
<tr>
<td>6</td>
<td>54.05</td>
<td>43.39</td>
<td>127.68</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>4.05</td>
</tr>
<tr>
<td>7</td>
<td>89.68</td>
<td>175.70</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>14.19</td>
</tr>
<tr>
<td>8</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>28.58</td>
</tr>
<tr>
<td>9</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>50.46</td>
</tr>
<tr>
<td>10</td>
<td>111.67</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>16.52</td>
</tr>
<tr>
<td>11</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>112.71</td>
</tr>
<tr>
<td>12</td>
<td>57.78</td>
<td>56.80</td>
<td>92.90</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>25.07</td>
</tr>
<tr>
<td>13</td>
<td>43.98</td>
<td>92.03</td>
<td>51.95</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>14.51</td>
</tr>
<tr>
<td>14</td>
<td>60.40</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>14.59</td>
</tr>
<tr>
<td>15</td>
<td>56.89</td>
<td>32.18</td>
<td>79.72</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>13.65</td>
</tr>
<tr>
<td>16</td>
<td>70.76</td>
<td>41.68</td>
<td>99.71</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>15.25</td>
</tr>
<tr>
<td>17</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>14.62</td>
</tr>
<tr>
<td>18</td>
<td>150.65</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>109.03</td>
</tr>
<tr>
<td>19</td>
<td>27.61</td>
<td>56.59</td>
<td>110.77</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>10.33</td>
</tr>
<tr>
<td>20</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>109.58</td>
</tr>
<tr>
<td>21</td>
<td>76.63</td>
<td>76.14</td>
<td>71.28</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>6.57</td>
</tr>
<tr>
<td>22</td>
<td>30.89</td>
<td>127.81</td>
<td>70.25</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>10.53</td>
</tr>
<tr>
<td>23</td>
<td>43.44</td>
<td>23.92</td>
<td>49.00</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>25.07</td>
</tr>
<tr>
<td>24</td>
<td>54.05</td>
<td>43.39</td>
<td>127.68</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>25.07</td>
</tr>
<tr>
<td>25</td>
<td>609.88</td>
<td>175.70</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>14.19</td>
</tr>
<tr>
<td>26</td>
<td>276.12</td>
<td>127.81</td>
<td>70.25</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>10.33</td>
</tr>
<tr>
<td>27</td>
<td>76.63</td>
<td>76.14</td>
<td>71.28</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>6.57</td>
</tr>
<tr>
<td>28</td>
<td>30.89</td>
<td>127.81</td>
<td>70.25</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>10.53</td>
</tr>
<tr>
<td>29</td>
<td>43.44</td>
<td>23.92</td>
<td>49.00</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>25.07</td>
</tr>
<tr>
<td>30</td>
<td>54.05</td>
<td>43.39</td>
<td>127.68</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>25.07</td>
</tr>
<tr>
<td>31</td>
<td>609.88</td>
<td>175.70</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>14.19</td>
</tr>
<tr>
<td>32</td>
<td>276.12</td>
<td>127.81</td>
<td>70.25</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>10.33</td>
</tr>
<tr>
<td>33</td>
<td>76.63</td>
<td>76.14</td>
<td>71.28</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>6.57</td>
</tr>
<tr>
<td>34</td>
<td>30.89</td>
<td>127.81</td>
<td>70.25</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>10.53</td>
</tr>
<tr>
<td>35</td>
<td>43.44</td>
<td>23.92</td>
<td>49.00</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>25.07</td>
</tr>
<tr>
<td>36</td>
<td>54.05</td>
<td>43.39</td>
<td>127.68</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>25.07</td>
</tr>
<tr>
<td>37</td>
<td>609.88</td>
<td>175.70</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>14.19</td>
</tr>
<tr>
<td>38</td>
<td>276.12</td>
<td>127.81</td>
<td>70.25</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>10.33</td>
</tr>
<tr>
<td>39</td>
<td>76.63</td>
<td>76.14</td>
<td>71.28</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>6.57</td>
</tr>
<tr>
<td>40</td>
<td>30.89</td>
<td>127.81</td>
<td>70.25</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>10.53</td>
</tr>
<tr>
<td>41</td>
<td>43.44</td>
<td>23.92</td>
<td>49.00</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>25.07</td>
</tr>
<tr>
<td>42</td>
<td>54.05</td>
<td>43.39</td>
<td>127.68</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>25.07</td>
</tr>
<tr>
<td>43</td>
<td>609.88</td>
<td>175.70</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>14.19</td>
</tr>
<tr>
<td>44</td>
<td>276.12</td>
<td>127.81</td>
<td>70.25</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>10.33</td>
</tr>
<tr>
<td>45</td>
<td>76.63</td>
<td>76.14</td>
<td>71.28</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>6.57</td>
</tr>
<tr>
<td>46</td>
<td>30.89</td>
<td>127.81</td>
<td>70.25</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>10.53</td>
</tr>
<tr>
<td>Test #</td>
<td>QBB</td>
<td>QBN</td>
<td>QBS</td>
<td>QNB</td>
<td>QNN</td>
<td>QNS</td>
<td>QSB</td>
</tr>
<tr>
<td>-------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>1</td>
<td>7.94</td>
<td>14.92</td>
<td>12.48</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>1.78</td>
</tr>
<tr>
<td>2</td>
<td>84.62</td>
<td>29.96</td>
<td>30.63</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>0.49</td>
</tr>
<tr>
<td>3</td>
<td>106.41</td>
<td>89.66</td>
<td>108.05</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>25.03</td>
</tr>
<tr>
<td>4</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>199.37</td>
</tr>
<tr>
<td>5</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>162.16</td>
</tr>
<tr>
<td>6</td>
<td>86.93</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>5.75</td>
</tr>
<tr>
<td>7</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>8</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>9</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>10</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>89.99</td>
</tr>
<tr>
<td>11</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>12.29</td>
</tr>
<tr>
<td>12</td>
<td>31.92</td>
<td>59.55</td>
<td>30.01</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>1.33</td>
</tr>
<tr>
<td>13</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>14</td>
<td>144.72</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>4.49</td>
</tr>
<tr>
<td>15</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>16</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>32.30</td>
</tr>
<tr>
<td>17</td>
<td>125.53</td>
<td>129.79</td>
<td>114.93</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>4.34</td>
</tr>
<tr>
<td>18</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>175.26</td>
</tr>
<tr>
<td>19</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>25.78</td>
</tr>
<tr>
<td>20</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>17.16</td>
</tr>
<tr>
<td>21</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>32.30</td>
</tr>
<tr>
<td>22</td>
<td>59.80</td>
<td>94.50</td>
<td>81.45</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>1.78</td>
</tr>
<tr>
<td>23</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>5.34</td>
</tr>
<tr>
<td>24</td>
<td>101.28</td>
<td>193.48</td>
<td>91.70</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>14.29</td>
</tr>
<tr>
<td>25</td>
<td>38</td>
<td>59.80</td>
<td>94.50</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>5.34</td>
</tr>
<tr>
<td>26</td>
<td>38</td>
<td>59.80</td>
<td>94.50</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>5.34</td>
</tr>
<tr>
<td>27</td>
<td>38</td>
<td>59.80</td>
<td>94.50</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>5.34</td>
</tr>
<tr>
<td>28</td>
<td>38</td>
<td>59.80</td>
<td>94.50</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>5.34</td>
</tr>
</tbody>
</table>
Table 8: Solution times for CPLEX with 6 strategies for 14 test problems without blocks

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

Table 9: Solution times for CBC with 6 strategies for 14 test problems without blocks

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


