

## FEASIBILITY-BASED DESIGN MODEL FOR ROAD VERTICAL ALIGNMENT

HONGZHI YANG<sup>1\*</sup>, XULIANG GUO<sup>2</sup>,  
ZHENFENG WANG<sup>3</sup>, SHANSHAN HU<sup>4</sup>

<sup>1</sup>*School of Highway, Chang'an University, Xi'an, China*

<sup>2</sup>*School of Highway, Chang'an University, Xi'an, China*

<sup>3</sup>*Highway&Municipal Engineering Division, China Railway Design Corporation,  
Tianjin, China*

<sup>4</sup>*Tianjin urban construction design institute, Tianjin, China*

Received 24 September 2020; accepted 29 June 2021

**Abstract.** Road vertical alignment design is a multi-objective design problem that needs to consider multiple constraints. Intelligent design based on optimization algorithms cannot wholly solve problems, such as multi-objective, uncertainty, and constraint dynamics. The article proposes a model of dynamically transforming design constraints into feasible regions as the design develops, to provide decision information before design actions rather than performing constraint evaluation after the design that reduces the empirical estimation. The design actions are divided into new design actions and modifying design actions, and corresponding feasible regions derived from constraints of design specifications and control elevations are established, respectively. Geometrical equations and program algorithms of feasible regions are described in the graphic environment, which is applied to the vertical alignment design to improve the design efficiency and decision-making level.

**Keywords:** Road design, vertical alignment, feasibility-based design model, design constraints, feasible region.

\* Corresponding author. E-mail: [yhz@chd.edu.cn](mailto:yhz@chd.edu.cn)

Hongzhi YANG (ORCID ID 0000-0001-6152-8739)

Xuliang GUO (ORCID ID 0000-0002-6450-1571)

Copyright © 2021 The Author(s). Published by RTU Press

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

## Introduction

The traditional road vertical alignment design uses a heuristic design method supported by the CAD system. Based on their own design experience, the designer tentatively proposes a design solution, evaluates whether the design solution meets the design constraints, and then revises the design to complete the design task in repeated iterations. In this process, the role of the CAD system is for geometric calculations and the drawing of design solutions (Jha, Schonfeld & Jong, 2006; M. Singh, P. Singh & P. Singh, 2019).

Traditional design methods describe design problems based on qualitative descriptions of design experience. The design process needs to iterate repeatedly, which affects the design efficiency. Studies such as Jha and Schonfeld (2000), Beiranvand et al. (2017), Li et al. (2017) indicated that using optimization models could considerably speed up the design process and provide preliminary design solutions for alignment design. From the 1960s to the present, the research on vertical alignment design has mainly focused on the study of alignment optimization models.

Vertical alignment optimization is a process of finding the optimal solution through an optimization algorithm based on design specifications and constraints, intending to minimize road construction cost. Easa (1988), Moreb (1996) proposed a vertical alignment optimization model using linear programming with the goal of minimization of earthworks. A mixed-integer linear programming model considering blocks and side-slopes accurately estimates the optimal earthwork costs and reduces the calculation time (Hare, Lucet & Rahman, 2015; Hare, Koch & Lucet, 2011). Recently the genetic algorithms (GAs) have been used to solve vertical alignment optimization (Goktepe, Lav & Altun, 2009; Jha, Schonfeld & Jong, 2006). Enumeration, dynamic programming, and numerical search have been used to solve the vertical alignment optimization problem, which is summarised in detail by Jha, Schonfeld and Jong (2006). The discrete dynamic programming was integrated with the Weighted Ground Line Method (WGLM) to achieve the best vertical alignment of highways in terms of earthwork optimization (Goktepe, Altun & Ahmedzade, 2010).

The alignment optimization model mathematically solves the problem of the optimal solution under certain constraints, but it lacks application in practical design. Hare et al. (2014) indicated that their optimization model was used by their industry partner and was developed as interactive road design software. Most of the papers did not mention software development and application.

The highway geometric design is ill-defined or there are ill-structured problems requiring some creativity for their solution (Jonassen, 1997). First, the unclear description of road geometric design problems includes uncertainty about concepts, rules, and principles. There is uncertainty between geometric design indicators and design goals, such as the impact of the combination of horizontal and vertical alignment on safety and aesthetics. Second, the definition of a design problem contains inconsistencies. Lamm, Psarianos and Mailaender (1999) indicated that highway geometric design was measured according to six conflicting goals. Designers need to balance conflicting design goals. Finally, the designer should propose multiple solutions to find an acceptable compromise solution in the uncertain design space. The result of optimization is often used as an initial reference for interactive design. Due to the uncertainty of the design space, any change in the equilibrium state during the design process will make the optimization solution unfeasible. For example, due to traffic safety considerations, it is necessary to increase the amount of engineering on a specific section of the road, which leads to the abandonment of the optimal solution to save the most engineering costs.

As the mathematical optimization model can hardly solve problems of vertical alignment design, the article proposes a model of vertical alignment design process based on the feasible region. In the design process, design constraints are transformed into mathematical models and graphics. By dynamically calculating the boundaries of the uncertain design space, it provides designers with more decision-making information support. Feasible region models based on design specification constraints and control elevation constraints under the states of design element creation and design element modification are established, which improves design efficiency.

## 1. Design process model based on feasible regions

Just as a design is a product of the designer's approach, it is also a reflection of a particular pattern of constraints that make up the problem (Lawson, 2006). The design of vertical alignment can be described as the problem of selecting grades and vertical curves and combining them reasonably under the constraints of design specifications and external environment to meet the design goals, such as function, safety, economy, and aesthetics. Design constraints are reflections of design goals.

In the process of design, from the initial proposal to the final design that meets the design goals, there are many intermediate design states.

The formation process of the intermediate design state is functionally represented as a series of design activities. The follow-up activities are the development and evolution of the previous activity, forming a design process composed of multiple design activities. In this process, design constraints are a mapping of the space of the design goals. Multiple design constraints tailor the design space to form a feasible region of the design feature space. The feasible region can be defined as follows:

**Definition 1.** Feasible region: It refers to a design space which can meet various design constraints (including design specification constraints and elevation control constraints) in the process of vertical alignment design. As long as the VPI is set in this design space, the design solution will be feasible.

The design process model  $M$  can be described as follows:

$$M = \{I, P, Q, O, C, \varphi\}, \quad (1)$$

where  $I/O$  is the input and output set;  $P$  is the design process set,  $P = \{P_1, P_2, \dots, P_n\}$ ;  $Q$  is the state set,  $Q = \{Q_1, Q_2, \dots, Q_n\}$ ;  $C$  is the constraint set,  $C = \{C_1, C_2, \dots, C_n\}$ ; the state  $Q_i | C_i$  that satisfies a specific constraint corresponds to  $P_i$ ;  $\varphi$  is a mapping set,  $\varphi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ , where  $\varphi_i = Q_i | C_i \rightarrow Q_{i+1} | C_{i+1}$ .

Design states that satisfy certain constraints are transformed by mapping, which is the design sub-processes:

$$P_{i+1} = Q_i \left| \begin{array}{c} \varphi_i \\ C_i \rightarrow Q_{i+1} \\ C_{i+1} \end{array} \right. \quad (2)$$

The design input gets the design output through the design process:  
 $I \times P \rightarrow O$ .

For the vertical alignment design process model (VADPM), the design process model input includes horizontal alignment, ground line of the alignment, etc. Design sub-processes  $P_i$  refers to the design of the grade line and vertical curve, modification of vertical points of intersection (VPI), and vertical curve (VC). Design constraint set includes controls of design criteria, controls of elevation.

The vertical alignment design is generally developed gradually from the starting point to the endpoint. Design state of vertical alignment:

$$Q_i = \{Stn_i, Elv_i, K_i, Stn_{i+1}, Elv_{i+1}, K_{i+1}, \dots, Stn_n, Elv_n, K_n\}, \quad (3)$$

where

$Stn_i$  – the station of VPI<sub>*i*</sub>;

$Elv_i$  – the design elevation of VPI<sub>*i*</sub>;

$K_i$  – the rate of vertical curvature at VPI<sub>*i*</sub>.

Satisfying design constraints is one of the core issues of the model. In interaction design, the designer, with the support of the CAD software,

obtains a specific design state through a series of design actions. The software built-in Design Check module and the designer's experience determine whether the design state satisfies design constraints. If it is not satisfied, the design should be modified, and this is a design-evaluation-modification iterative design method. In this method, the requirements of the design constraint  $C_{i+1}$  are met through empirical estimation of constraints in the design process  $P_{i+1}$  and design evaluation after reaching the design state  $Q_{i+1}$ . For  $P_{i+1}$  and  $Q_{i+1}$ , the design constraint  $C_{i+1}$  quantitatively acts after the completion of the stage design.

Figure 1 shows that the design states  $Q_i$  and  $Q_{i+1}$  are inherited.

$$Q_{i+1} = Q_i \cup \{Stn_{i+1}, Elv_{i+1}, K_{i+1}\}. \quad (4)$$

During the development of  $Q_i$  to  $Q_{i+1}$ , the design space under constraint tailoring constitutes the feasible region  $FD_i$  of  $Q_{i+1}$ . If  $\{Stn_{i+1}, Elv_{i+1}, K_{i+1}\}$  determined by  $P_{i+1}$  is within the feasible region, the design constraints  $C_{i+1}$  are satisfied. The feasible region model

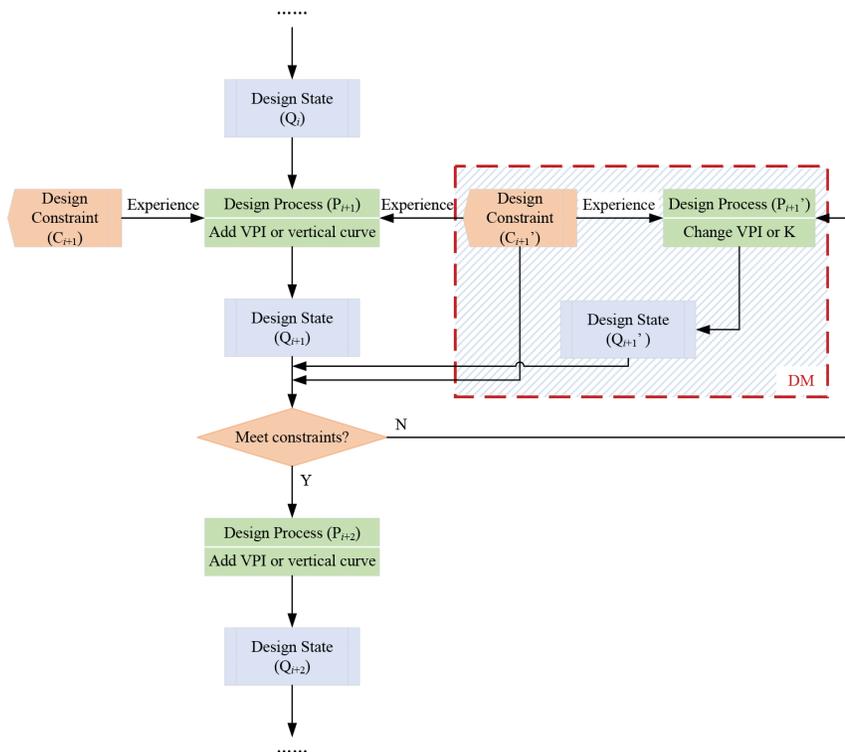
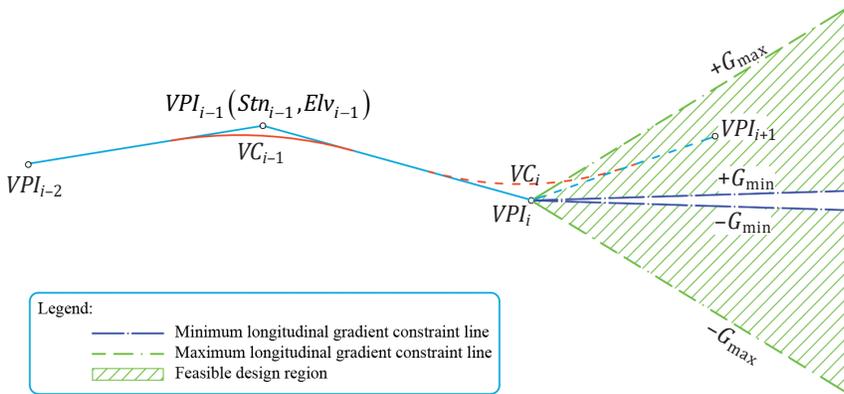


Figure 1. Design-evaluation-modification design method

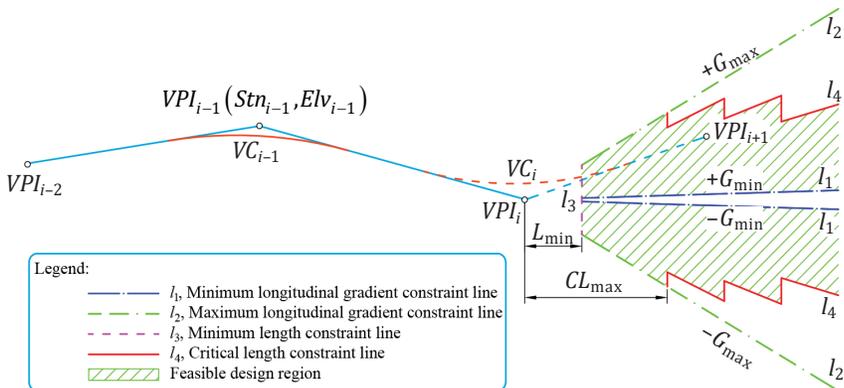
transforms the evaluation and modification after the design process into the evaluation and design under the limitation of the feasible region during the design process.

Figure 2 shows that when the transition takes place from the design state  $Q_i$  to  $Q_{i+1}$ , i.e., when determining the coordinate of  $VPI_{i+1}$ , the grade constraints include (AASHTO, 2011; Fanning et al., 2016; Wang, Guo & Luo, 2016):

1. For the reason of drainage, the minimum gradient should generally be higher than 0.3%, so the line  $l_1$  in Fig. 2a is drawn;
2. Considering safety, economy, and efficiency, the maximum gradient should be controlled and the line  $l_2$  in Fig. 2b should be drawn.



**Figure 2a.** Feasible regions during the design process: feasible region satisfies the constraints of the gradient



**Figure 2b.** Feasible regions during the design process: feasible region satisfies the constraints of the gradient and length of grade

The line  $l_1$  and  $l_2$  form the feasible region in Fig. 2a. During the design, as long as  $VPI_{i+1}$  is guaranteed in the feasible region, grade control should be satisfied. Similarly, the feasible region is superimposed on the minimum grade length ( $l_3$ ) and the critical length of grade ( $l_4$ ), then the new feasible region that satisfies the constraints of the gradient and length of grade is formed. This is the idea of the design process model (FRDPM) based on the feasible region.

$$FRDPM = \{I, C, f, FD, P, \theta, Q, O\}, \quad (5)$$

where

$FD$  – the design feasible region set;

$f$  – mapping of design constraints set to design feasible region set, where

$f_{i+1} : C_{i+1} \mid Q_i \rightarrow FD_{i+1}$ ;

$\theta$  – mapping of design states, where  $\theta_i : Q_i \xrightarrow{P_{i+1} \mid FD_{i+1}} Q_{i+1}$ , which means that the mapping from the design state  $Q_i$  to  $Q_{i+1}$  is done by design process elements within the  $FD_{i+1}$ .

The vertical alignment design flow based on FRDPM is shown in Fig. 3. It can be seen that FRDPM transforms the constraint evaluation

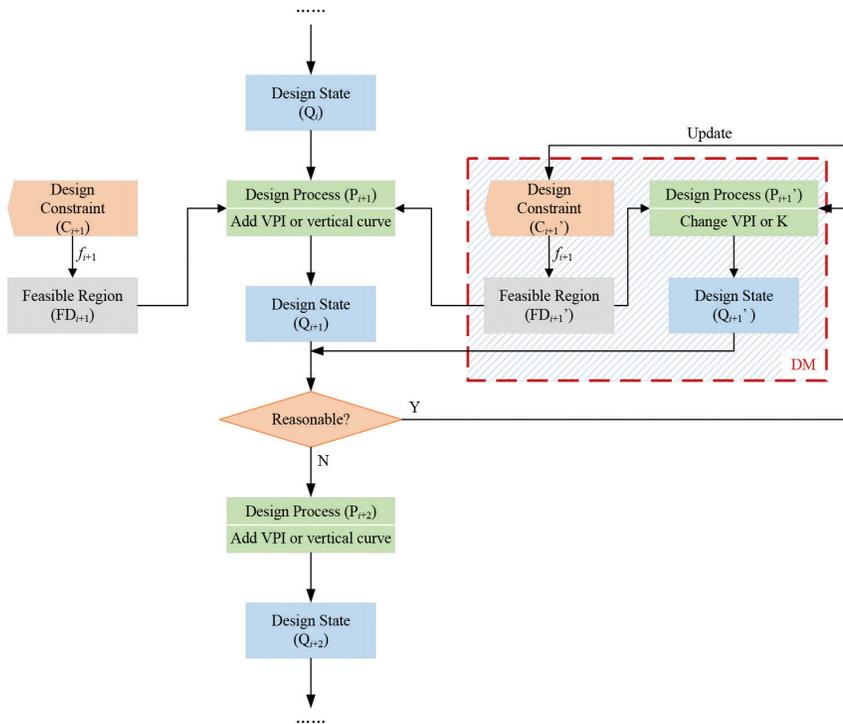


Figure 3. Vertical alignment design based on FRDPM

in the design process into a design within the feasible region. Compared with the traditional vertical alignment design approach, FRDPM could provide real-time graphical constraint information before design actions rather than constraint evaluation after design, which will reduce the work of repeated modifications and improve design efficiency.

## 2. Design constraints of vertical alignment

There are many constraints on vertical alignment design. This paper mainly discusses design specification constraints and control elevation constraints.

The vertical alignment comprises grades and the vertical curves between them, and design specification constraints include grade constraints and vertical curve constraints.

$$C_S = \{C_g, C_c\}, \quad (6)$$

where

$C_S$  – design specification constraints;

$C_g$  – grade constraints;

$C_c$  – vertical curve constraints.

$$C_g = \{G_{\min} \leq G \leq G_{\max}, L_{\min} \leq L \leq L_{\max}(G_i)\}, \quad (7)$$

where

$G$  – gradient between two VPI, %;

$G_{\max}$  – maximum gradient for specified design speed, %;

$G_{\min}$  – minimum gradient according to design specification, %;

$L$  – length of grade between two VPI, m;

$L_{\min}$  – minimum length of grade between two VPI, m;

$L_{\max}(G_i)$  – critical length of steep grade between two VPI for specified grade and design speed.

The paper calculates the feasible region based on the minimum and maximum grade lengths in Chinese specifications (Wang, Guo & Luo, 2016).

$$C_c = \{K \geq K_{\min}, L_c \geq L_{c\min}\}, \quad (8)$$

where

$K$  – rate of vertical curvature;

$K_{\min}$  – minimum rate of the vertical curve for specified design speed;

$L_c$  – length of the vertical curve, m;

$L_{c\min}$  – minimum length of the vertical curve for specified design speed.

Control elevation constraints include the following: control elevations for overpasses or underpasses to ensure vertical clearances; control

elevations for bridges, culverts, or subgrades that consider flood levels; control elevations for tunnels based on geological conditions; and control elevations for intersection considering the intersecting road (Fwa, Chan & Sim, 2002; Kim et al., 2007). These constraints generate many tie points, defined as levels above which, through which or below which the profile must pass (Wolhuter, 2019).

$$C_E = \{C_a, C_b, C_t\}, \quad (9)$$

where

$C_E$  – control elevation constraints;

$C_a$  – constraints of tie points above which the vertical alignment must pass;

$C_b$  – constraints of tie points below which the vertical alignment must pass;

$C_t$  – constraints of tie points through which the vertical alignment must pass.

### 3. VPI feasible region model for new design

Vertical alignment design can be divided into creating design objects and modifying design objects according to design actions. The present paper discusses the feasible region model for the new design and the feasible region model for design modifications.

#### 3.1. Feasible region of $C_g$

Equation (7) shows that  $C_g$  is composed of gradient constraint  $C_{g1}$  and grade length constraint  $C_{g2}$ :

$$C_{g1} = \{G_{\min} \leq G \leq G_{\max}\}; \quad (10)$$

$$C_{g2} = \{L_{\min} \leq L \leq L_{\max}(G_i)\}; \quad (11)$$

$$C_g = C_{g1} \cup C_{g2}. \quad (12)$$

Therefore,

$$FD_{A1} = FD_{g1} \cap FD_{g2}, \quad (13)$$

where

$FD_{A1}$  – feasible region of adding a new VPI under the constraint of  $C_g$ ;

$FD_{g1}$  – feasible region of adding a new VPI under the constraint of  $C_{g1}$ ;

$FD_{g2}$  – feasible region of adding a new VPI under the constraint of  $C_{g2}$ .

Once the coordinate of  $VPI_i(Stn_i, Elv_i)$  is determined,  $FD_{A1}$  of  $VPI_{i+1}(Stn_{i+1}, Elv_{i+1})$  can be defined as follows:

**Definition 2.**  $FD_{g1}$ : if  $(Stn_{i+1}, Elv_{i+1}) \in FD_{g1}$ , then  $(Stn_{i+1}, Elv_{i+1})$  should satisfy:

$$(1) Stn_{i+1} \in (Stn_i, +\infty);$$

$$(2) \left| \frac{Elv_{i+1} - Elv_i}{Stn_{i+1} - Stn_i} \right| \in [G_{\min}, G_{\max}].$$

$FD_{g1}$  is shown in Fig. 4.

**Definition 3.** Function of the critical length of steep grade:

$$L_{\max}(G) = \begin{cases} L_1 & \text{if } G_0 < G \leq G_1 \\ L_2 & \text{if } G_1 < \leq G \leq G_2 \\ \dots & \dots \\ L_n & \text{if } G_{n-1} < G \leq G_n \end{cases}. \quad (14)$$

Taking the Chinese specification as an example, when the design speed is 80 km/h, the critical lengths of steep grade are shown in Table 1.

Table 1. Critical length of steep grade ( $V_d=80$  km/h)

G	3%	4%	5%	6%
$L_{\max}(G)$	1100 m	900 m	700 m	500 m

As shown in Eq. (14),  $\{L_1, L_2, L_3, L_4\} = \{1100\text{m}, 900\text{m}, 700\text{m}, 500\text{m}\}$ ,  
 $\{G_0, G_1, G_2, G_3, G_4\} = \{0.03, 0.04, 0.05, 0.06, 0.07\}$ .

$FD_{g2}$  and  $FD_{A1}$  of  $VPI_{i+1}(Stn_{i+1}, Elv_{i+1})$  can be defined as follows:

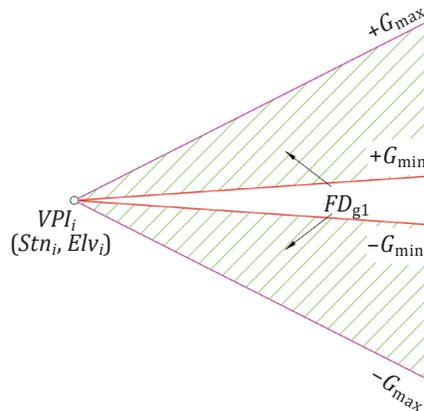


Figure 4. Feasible region of  $FD_{g1}$

**Definition 4.**  $FD_{g2}$ : if  $(Stn_{i+1}, Elv_{i+1}) \in FD_{g2}$ , then  $(Stn_{i+1}, Elv_{i+1})$  should satisfy:

$$(1) Stn_{i+1} \in [Stn_i + L_{\min}, +\infty);$$

$$(2) \text{ If } \left| \frac{Elv_{i+1} - Elv_i}{Stn_{i+1} - Stn_i} \right| \in (G_j, G_{j+1}], \text{ then } Stn_{i+1} \in [Stn_i + L_{j+1}, +\infty),$$

where  $G_j, G_{j+1} \in \{G_0, G_1, \dots, G_n\}, L_{j+1} = L_{\max}(G_{j+1})$ .

**Definition 5.**  $FD_{A1}$ : if  $(Stn_{i+1}, Elv_{i+1}) \in FD_{A1}$ , then  $(Stn_{i+1}, Elv_{i+1})$  should satisfy:

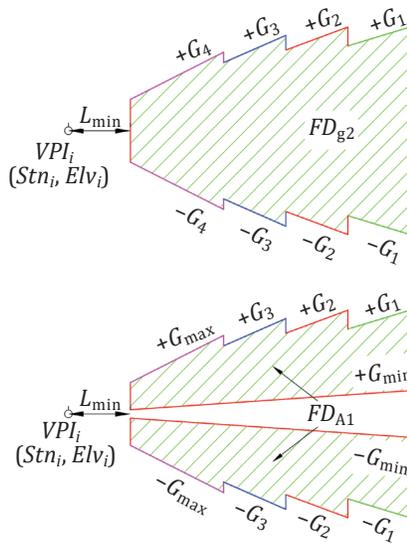
$$(1) (Stn_{i+1}, Elv_{i+1}) \in FD_{g1};$$

$$(2) (Stn_{i+1}, Elv_{i+1}) \in FD_{g2};$$

The corresponding feasible regions are presented in Fig. 5.

### 3.2. Feasible region of $C_g$ and $C_c$

Once the  $VPI_{i+1}$  falls within the feasible region  $FD_{A1}$ , the design constraints on  $C_g$  will be satisfied. However, whether vertical curves that satisfy design constraints can be set is still unknown. Therefore, the influence of  $C_c$  on feasible regions should be considered. Vertical curves take the form of symmetrical parabolas, so the vertical curves have equal lengths on either side of VPI and can be designed using Eq. (15) (Higuera de Frutos & Castro, 2017).



**Figure 5.** Feasible regions of  $FD_{g2}$  and  $FD_{A1}$

$$L_{c(i)} = K_i A_i, i \in N^*, \quad (15)$$

where  $A_i$  is the algebraic difference of gradient.

$$A_i = |G_{i+1} - G_i|, \quad (16)$$

where  $G_{i+1}, G_i$  are tangent grades in percent.

As shown in Fig. 6 as long as  $VPI_{i+1}$  is not the EP (End Point), the necessary constraint for  $L_i$ :

$$L_i \geq \frac{L_{c(i)}}{2} + \frac{L_{c(i+1)}}{2} \geq L_{cmin}. \quad (17)$$

Meanwhile, the necessary constraint for  $VC_i$ :

$$\begin{cases} K_i \geq K_{imin} \\ \frac{L_{c(i)}}{2} \leq \text{Min} \left( L_{i-1} - \frac{L_{c(i-1)}}{2}, L_i - \frac{L_{cmin}}{2} \right) \end{cases} \quad (18)$$

Let  $\alpha = \text{Min} \left( 2L_{i-1} - L_{c(i-1)}, 2L_i - L_{cmin} \right)$ . Combining Eq. (16) and Eq. (18) gives:

$$G_i - \frac{\alpha}{K_{imin}} \leq G_{i+1} \leq G_i + \frac{\alpha}{K_{imin}}. \quad (19)$$

As shown in Eqs. (17) and (19) that can be regarded as the potential constraint of  $C_c$  on  $VPI_{i+1}$ , the following equation can be obtained by mapping rules:

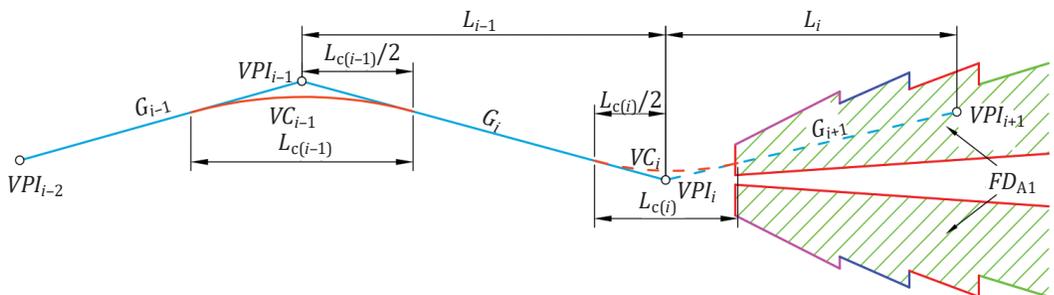
$$FD_{A2} = FD_{c1} \cap FD_{A1}, \quad (20)$$

where

$FD_{A1}$  – feasible region of adding a new VPI under the constraints of  $C_g$  and  $C_c$ ;

$FD_{c1}$  – feasible region of adding a new VPI under the constraint of  $C_c$ .

$FD_{c1}$  and  $FD_{A1}$  of  $VPI_{i+1} (Stn_{i+1}, Elv_{i+1})$  can be defined as follows:



**Figure 6.** Feasible region of  $FD_{A1}$  in the design process

**Definition 6.**  $FD_{c1}$ : if  $(Stn_{i+1}, Elv_{i+1}) \in FD_{c1}$ , then  $(Stn_{i+1}, Elv_{i+1})$  should satisfy the condition of

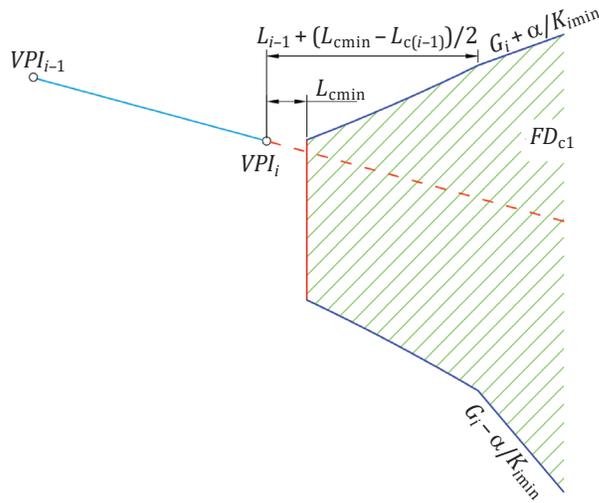
$$\frac{Elv_{i+1} - Elv_i}{Stn_{i+1} - Stn_i} \in \left[ G_i - \frac{\alpha}{K_{imin}}, G_i + \frac{\alpha}{K_{imin}} \right].$$

**Definition 7.**  $FD_{A2}$ : if  $(Stn_{i+1}, Elv_{i+1}) \in FD_{A2}$ , then  $(Stn_{i+1}, Elv_{i+1})$  should satisfy:

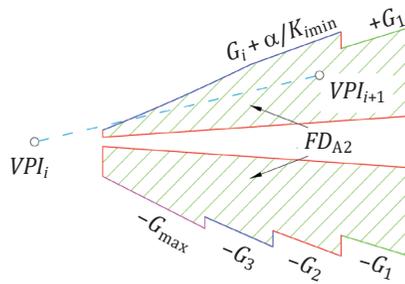
(1)  $(Stn_{i+1}, Elv_{i+1}) \in FD_{c1}$ ;

(2)  $(Stn_{i+1}, Elv_{i+1}) \in FD_{A1}$ .

$FD_{c1}$  and  $FD_{A2}$  are shown in Figs. 7 and 8, respectively.



**Figure 7.** Feasible region of  $FD_{c1}$



**Figure 8.** Feasible region of  $FD_{A2}$

### 3.3. Feasible region of $C_S$ and $C_E$

Once there are tie points, not only grade constraints  $C_g$  and vertical curve constraints  $C_c$ , but also control elevation constraints  $C_E$  should be considered when adding a new VPI. Limited to space, only  $C_a$  in  $C_E$  is discussed. As shown in Fig. 9, point E is a tie point, and the vertical alignment should pass over it.

According to the coordinate of  $VPI_i$  and point E, the feasible region  $FD_{A2}$  can be divided into three parts, as shown in Fig. 10.

The significance of each region:

$FD_1$ : If  $VPI_i$  is in this region, the constraint  $C_a$  can be satisfied.

$FD_2$ : If  $VPI_i$  is in this region, the constraint  $C_a$  may be satisfied.

$FD_3$ : If  $VPI_i$  is in this region, the constraint  $C_a$  cannot be satisfied.

$FD_3$  is the impossible region and should be deleted.  $FD_1$  and  $FD_2$  should be retained and set in different styles for users' reference. Thus,

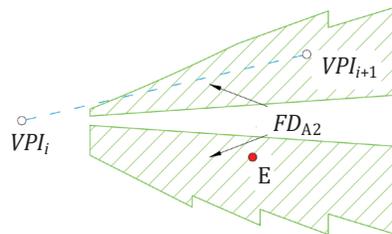
$$FD_{A3} = FD_{A2} - FD_3, \quad (21)$$

where

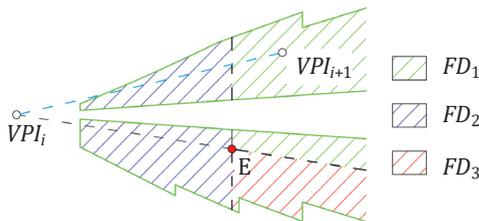
$FD_{A3}$  – feasible region of adding a new VPI under the constraints of  $C_S$  and  $C_E$ ;

$FD_3$  – impossible region of adding a new VPI under the constraints of  $C_S$  and  $C_E$ .

Based on the coordinate of point  $E(Stn_E, Elv_E)$ ,  $FD_1$ ,  $FD_2$ ,  $FD_3$  and  $FD_{A3}$  can be defined as follows:



**Figure 9.** A tie point when adding a new VPI



**Figure 10.** Division of  $FD_{A2}$

**Definition 8.**  $FD_1$ : if  $(Stn_{i+1}, Elv_{i+1}) \in FD_1$ , then  $(Stn_{i+1}, Elv_{i+1})$  should satisfy:

- (1)  $(Stn_{i+1}, Elv_{i+1}) \in FD_{A2}$ ;
- (2)  $Stn_{i+1} \in (Stn_E, +\infty]$ ;
- (3)  $\frac{Elv_{i+1} - Elv_i}{Stn_{i+1} - Stn_i} \in \left[ \frac{Elv_E - Elv_i}{Stn_E - Stn_i}, +\infty \right)$ .

**Definition 9.**  $FD_2$ : if  $(Stn_{i+1}, Elv_{i+1}) \in FD_2$ , then  $(Stn_{i+1}, Elv_{i+1})$  should satisfy:

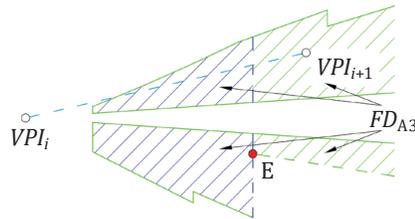
- (1)  $(Stn_{i+1}, Elv_{i+1}) \in FD_{A2}$ ;
- (2)  $Stn_{i+1} \in (-\infty, Stn_E]$ .

**Definition 10.**  $FD_3$ : if  $(Stn_{i+1}, Elv_{i+1}) \in FD_3$ , then  $(Stn_{i+1}, Elv_{i+1})$  should satisfy:

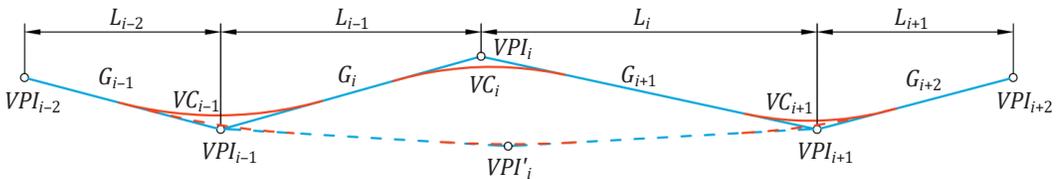
- (1)  $(Stn_{i+1}, Elv_{i+1}) \in FD_{A2}$ ;
- (2)  $Stn_{i+1} \in [x_E, +\infty)$ ;
- (3)  $\frac{Elv_{i+1} - Elv_i}{Stn_{i+1} - Stn_i} \in \left( -\infty, \frac{Elv_E - Elv_i}{Stn_E - Stn_i} \right]$ .

**Definition 11.**  $FD_{A3}$ : if  $(Stn_{i+1}, Elv_{i+1}) \in FD_{A3}$ , then  $(Stn_{i+1}, Elv_{i+1})$  should satisfy:

- (1)  $(Stn_{i+1}, Elv_{i+1}) \in FD_{A2}$ ;
- (2)  $(Stn_{i+1}, Elv_{i+1}) \notin FD_3$ .



**Figure 11.** Feasible region of  $FD_{A3}$



**Figure 12.** Schematic diagram of moving a middle VPI

$FD_{A3}$  is shown in Fig. 11. Then the algorithm of determining VPI feasible region for adding a new VPI can be summarised as Algorithm 1 in the APPENDIX.

## 4. VPI feasible region model for design modifications

The modification of VPI includes inserting, deleting, and moving the VPI. The paper only discusses moving a middle VPI when the front VPI and the next VPI are fixed. As shown in Fig. 12,  $VPI_{i-1}$  and  $VPI_{i+1}$  are fixed. The movement of  $VPI_i$  causes the front gradient, the rear gradient, and vertical curves to change. In this process, there is also a feasible region for  $VPI_i$ .

### 4.1. Feasible region of $C_g$

As shown in Fig. 12, when moving a middle VPI, its constraints mainly come from two parts: the front design constraint  $C_f$  and the rear design constraint  $C_r$ . Then

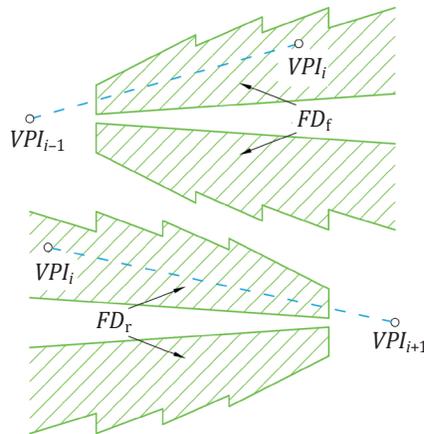
$$FD_{M1} = FD_f \cap FD_r, \quad (22)$$

where

$FD_{M1}$  – feasible region of moving a middle VPI under the constraint of  $C_g$ ;

$FD_f$  – feasible region of moving a middle VPI under the constraint of  $C_f$ ;

$FD_r$  – feasible region of moving a middle VPI under the constraint of  $C_r$ .



**Figure 13.** Feasible regions of  $FD_f$  and  $FD_r$

The situation is the same for  $FD_f$  and  $FD_{A1}$ , and  $FD_r$  can be regarded as  $FD_{A1}$  in the opposite direction. Therefore,  $FD_f$  and  $FD_r$  are presented in Fig. 13, and  $FD_{M1}$  is shown in Fig. 14.

#### 4.2. Feasible region of $C_g$ and $C_c$

As shown in Fig. 13, moving a middle VPI will also be constrained by three vertical curves:  $VC_{i-1}$ ,  $VC_i$ , and  $VC_{i+1}$ . The following equations can be obtained by mapping rules:

$$FD_{c2} = FD_{cf} \cap FD_{cm} \cap FD_{cr}, \quad (23)$$

where

$FD_{c2}$  – feasible region of moving a middle VPI under the constraint of  $C_c$ ;

$FD_{cf}$  – feasible region of moving a middle VPI under the constraint of  $VC_{i-1}$ ;

$FD_{cm}$  – feasible region of moving a middle VPI under the constraint of  $VC_i$ ;

$FD_{cr}$  – feasible region of moving a middle VPI under the constraint of  $VC_{i+1}$ .

And

$$FD_{M2} = FD_{c2} \cap FD_{M1}, \quad (24)$$

where

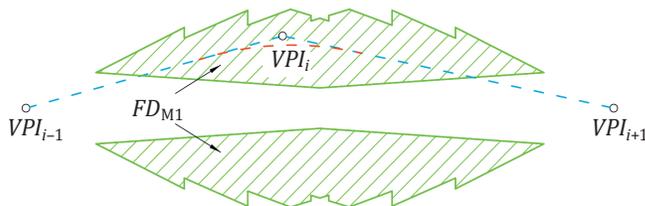
$FD_{M2}$  – feasible region of moving a middle VPI under the constraint of  $C_g$  and  $C_c$ .

Similarly, the situation is the same for  $FD_{cf}$  and  $FD_{c1}$ , and  $FD_{cr}$  can also be regarded as  $FD_{c1}$  in the opposite direction. Therefore,  $FD_{cf}$  and  $FD_{cr}$  are presented in Fig. 15 and Fig. 16, respectively.

Both the front and rear gradients are not constant, so  $FD_{cm}$  is different from  $FD_{cf}$  and  $FD_{cr}$ . The constraint on  $VC_i$  can be expressed by Eq. (25):

$$\begin{cases} K_i \geq K_{imin} \\ \frac{L_{c(i)}}{2} \leq \text{Min} \left( L_{i-1} - \frac{L_{cmin}}{2}, L_i - \frac{L_{cmin}}{2} \right) \end{cases} \quad (25)$$

Let  $\beta = \text{Min}(2L_{i-1} - L_{cmin}, 2L_i - L_{cmin})$ . Then

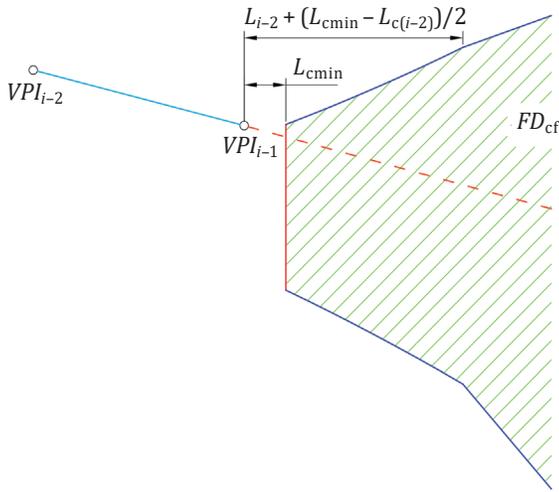


**Figure 14.** Feasible region of  $FD_{M1}$

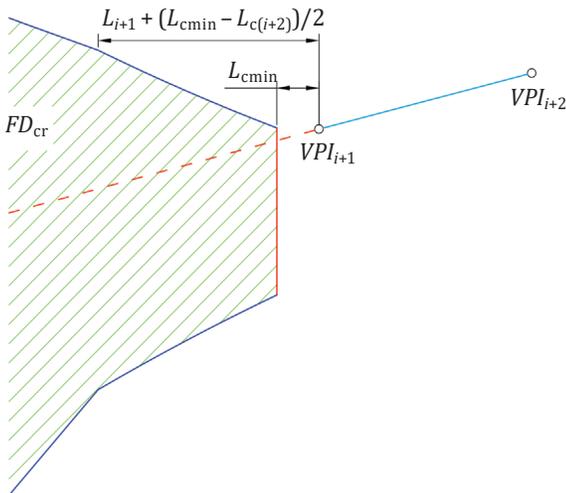
$$|G_{i+1} - G_i| \leq \frac{\beta}{K_{i\min}}. \quad (26)$$

Besides, the necessary constraint for  $L_{i-1}$  and  $L_i$ :

$$\begin{cases} L_{i-1} \geq \frac{L_{c(i-1)}}{2} + \frac{L_{c(i)}}{2} \geq L_{c\min} \\ L_i \geq \frac{L_{c(i)}}{2} + \frac{L_{c(i+1)}}{2} \geq L_{c\min} \end{cases}. \quad (27)$$



**Figure 15.** Feasible region of  $FD_{cf}$



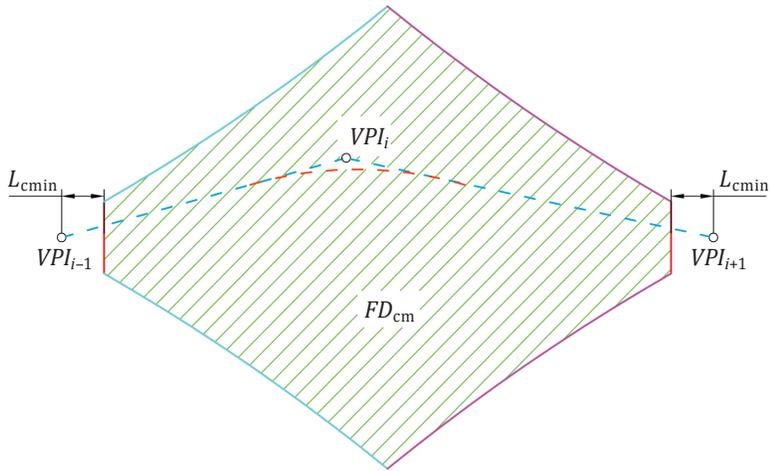
**Figure 16.** Feasible region of  $FD_{cr}$

Similar to  $FD_{c1}$ ,  $FD_{cm}$  can be defined as follows:

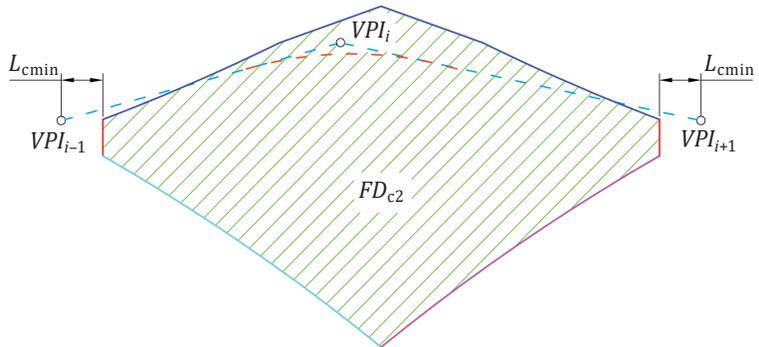
**Definition 12.**  $FD_{cm}$ : if  $(Stn_i, Elv_i) \in FD_{cm}$ , then  $(Stn_i, Elv_i)$  should satisfy:

- (1)  $Stn_i \in [Stn_{i-1} + L_{cmin}, Stn_{i+1} - L_{cmin}]$
- (2)  $\left| \frac{Elv_{i+1} - Elv_i}{Stn_{i+1} - Stn_i} - \frac{Elv_i - Elv_{i-1}}{Stn_i - Stn_{i-1}} \right| \in \left[ 0, \frac{\beta}{K_{imin}} \right]$ .

$FD_{cm}$  is shown in Fig. 17. From Eq. (23),  $FD_{c2}$  is presented in Fig. 18. Similarly,  $FD_{M2}$  is presented in Fig. 19 based on Eq. (24).



**Figure 17.** Feasible region of  $FD_{cm}$



**Figure 18.** Feasible region of  $FD_{c2}$

### 4.3. Feasible region of $C_S$ and $C_E$

As shown in Fig. 20, if there is a tie point, control elevation constraints  $C_E$  should also be considered when moving the  $VPI_i$ . Limited to space, only  $C_a$  in  $C_E$  is discussed, which means the vertical alignment should pass over point E.

Similar to  $FD_{A2}$ ,  $FD_{M2}$  can be divided into two parts, as shown in Fig. 21.

The significance of each region:

$FD_4$ : If  $VPI_i$  is in this region, the constraint  $C_a$  can be satisfied.

$FD_5$ : If  $VPI_i$  is in this region, the constraint  $C_a$  cannot be satisfied.

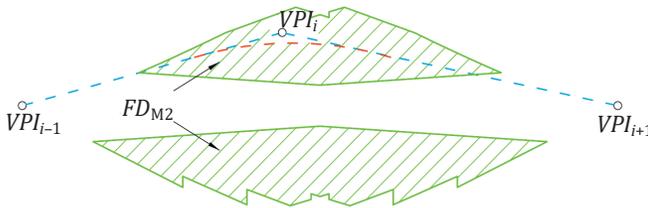
Thus,

$$FD_{M3} = FD_{M2} - FD_5, \quad (28)$$

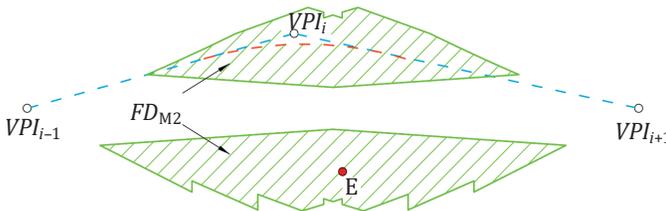
where

$FD_{A3}$  – feasible region of moving a middle VPI under the constraints of  $C_S$  and  $C_E$ ;

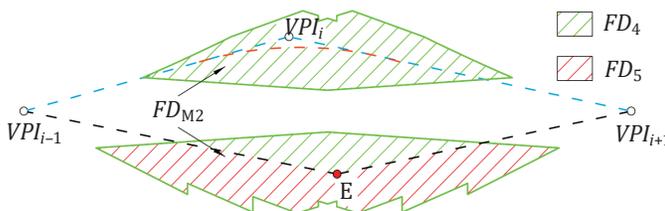
$FD_5$  – impossible region of moving VPI under the constraints of  $C_S$  and  $C_E$ .



**Figure 19.** Feasible region of  $FD_{M2}$



**Figure 20.** A tie point when moving a middle VPI



**Figure 21.** Division of  $FD_{M2}$

Based on the coordinate of point E ( $Stn_E, Elv_E$ ),  $FD_5$  and  $FD_{M3}$  can be defined as follows:

**Definition 13.**  $FD_5$ : if  $(Stn_i, Elv_i) \in FD_5$ , then  $(Stn_i, Elv_i)$  should satisfy:

$$(1) (Stn_i, Elv_i) \in FD_{M2};$$

$$(2) \frac{Elv_i - Elv_{i-1}}{Stn_i - Stn_{i-1}} \in \left[ -\infty, \frac{Elv_E - Elv_{i-1}}{Stn_E - Stn_{i-1}} \right];$$

$$(3) \frac{Elv_{i+1} - Elv_i}{Stn_{i+1} - Stn_i} \in \left[ \frac{Elv_{i+1} - Elv_E}{Stn_{i+1} - Stn_E}, +\infty \right].$$

**Definition 14.**  $FD_{M3}$ : if  $(Stn_i, Elv_i) \in FD_{M3}$ , then  $(Stn_i, Elv_i)$  should satisfy:

$$(1) (Stn_i, Elv_i) \in FD_{M2};$$

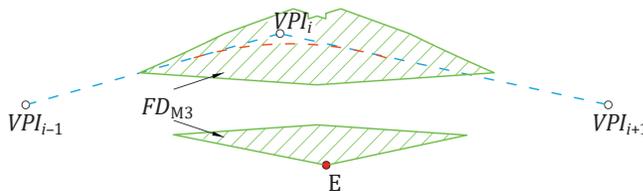
$$(2) (Stn_i, Elv_i) \notin FD_5.$$

$D_{M3}$  is shown in Fig. 22. Then the algorithm of determining VPI feasible region for moving a middle VPI can be summarised as Algorithm 2 in the APPENDIX.

## Conclusion

The paper has proposed a design process model based on feasible regions to provide decision information support for vertical alignment design. The main idea of the design process model is to convert constraints into feasible regions and provide designers with information support before design actions instead of design evaluation after design actions.

We discuss design specification constraints and control elevation constraints in vertical alignment design. Design specification constraints include grade, grade length,  $K$ -value, and length of vertical curve constraints. Control elevation constraints include elevation constraints for bridges, tunnels, intersections, overpasses, and underpasses based on flood levels or vertical clearances. We convert the above constraints into feasible region models for new and modified designs, respectively. Using mathematical and graphical methods, we give graphical numerical solutions and algorithm descriptions of feasible



**Figure 22.** Feasible region of  $FD_{M3}$

regions. Therefore, the feasible regions of the interactive design process are dynamically drawn to satisfy constraint information needs.

In addition to design specifications and control elevations, earthwork is a critical consideration in vertical alignment design. The design indices corresponding to each point in the feasible region is different, and the corresponding earthwork is also different. Combined with the cross-section design and earthwork calculation, the earthwork at each point in the feasible region can be theoretically calculated. Therefore, in the interactive design process, the earthwork in the feasible region can provide the designer with decision support for the cost goal, which is one of the future works.

Selecting reasonable points within the feasible region to determine the grade and vertical curve still requires some empirical judgment. There is a need for in-depth research on local optimization algorithms based on feasible regions, and dynamic optimization calculations in the interactive design process to improve design efficiency.

## Acknowledgements

The authors would like to express their sincere thanks for the support from the National Key Research and Development Program of China.

## Funding

The research has been supported by the National Key Research and Development Program of China [No. 2020YFC1512003].

## REFERENCES

- AASHTO. (2011). A Policy on Geometric Design of Highways and Streets, 2011: American Association of State Highway and Transportation Officials, Washington, D.C.
- Beiranvand, V., Hare, W., Lucet, Y., & Hossain, S. (2017). Multi-haul quasi network flow model for vertical alignment optimization. *Engineering Optimization*, 49(10), 1777–1795. <https://doi.org/10.1080/0305215X.2016.1271880>
- Easa, S. M. (1988). Selection of roadway grades that minimize earthwork cost using linear programming. *Transportation Research Part A: General*, 22(2), 121–136. [https://doi.org/10.1016/0191-2607\(88\)90024-6](https://doi.org/10.1016/0191-2607(88)90024-6)
- Fanning, R., Veith, G., Whitehead, M., & Aumann, P. (2016). Guide to road design part 3: geometric design.

- Fwa, T., Chan, W., & Sim, Y. (2002). Optimal vertical alignment analysis for highway design. *Journal of Transportation Engineering*, 128(5), 395–402. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2002\)128:5\(395\)](https://doi.org/10.1061/(ASCE)0733-947X(2002)128:5(395))
- Goktepe, A. B., Altun, S., & Ahmedzade, P. (2010). Optimization of vertical alignment of highways utilizing discrete dynamic programming and weighted ground line. *Turkish Journal of Engineering and Environmental Sciences*, 33(2), 105–116. <https://doi.org/10.3906/muh-0904-1>
- Goktepe, A. B., Lav, A. H., & Altun, S. (2009). Method for optimal vertical alignment of highways. Paper presented at the Proceedings of the Institution of Civil Engineers-Transport.
- Hare, W., Hossain, S., Lucet, Y., & Rahman, F. (2014). Models and strategies for efficiently determining an optimal vertical alignment of roads. *Computers & Operations Research*, 44, 161–173. <https://doi.org/10.1016/j.cor.2013.11.005>
- Hare, W., Lucet, Y., & Rahman, F. (2015). A mixed-integer linear programming model to optimize the vertical alignment considering blocks and side-slopes in road construction. *European Journal of Operational Research*, 241(3), 631–641. <https://doi.org/10.1016/j.ejor.2014.08.035>
- Hare, W. L., Koch, V. R., & Lucet, Y. (2011). Models and algorithms to improve earthwork operations in road design using mixed integer linear programming. *European Journal of Operational Research*, 215(2), 470–480. <https://doi.org/10.1016/j.ejor.2011.06.011>
- Higuera de Frutos, S., & Castro, M. (2017). A method to identify and classify the vertical alignment of existing roads. *Computer-Aided Civil and Infrastructure Engineering*, 32(11), 952–963. <https://doi.org/10.1111/mice.12302>
- Jha, M. K., & Schonfeld, P. (2000). Integrating genetic algorithms and geographic information system to optimize highway alignments. *Transportation Research Record*, 1719(1), 233–240. <https://doi.org/10.3141/1719-31>
- Jha, M. K., Schonfeld, P., & Jong, J.-C. (2006). Intelligent road design (Vol. 19). WIT press.
- Jonassen, D. H. (1997). Instructional design models for well-structured and III-structured problem-solving learning outcomes. *Educational Technology Research and Development*, 45(1), 65–94. <https://doi.org/10.1007/BF02299613>
- Kim, E., Jha, M. K., Schonfeld, P., & Kim, H. S. (2007). Highway alignment optimization incorporating bridges and tunnels. *Journal of Transportation Engineering*, 133(2), 71–81. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2007\)133:2\(71\)](https://doi.org/10.1061/(ASCE)0733-947X(2007)133:2(71))
- Lamm, R., Psarianos, B., & Mailaender, T. (1999). Highway design and traffic safety engineering handbook.
- Lawson, B. (2006). How designers think: The design process demystified. Routledge.
- Li, W., Pu, H., Schonfeld, P., Yang, J., Zhang, H., Wang, L., & Xiong, J. (2017). Mountain railway alignment optimization with bidirectional distance transform and genetic algorithm. *Computer-Aided Civil and Infrastructure Engineering*, 32(8), 691–709. <https://doi.org/10.1111/mice.12280>
- M. Singh, M. P., P. Singh, P., & P. Singh, P. (2019). Fuzzy AHP-based multi-criteria decision-making analysis for route alignment planning using geographic

- information system (GIS). *Journal of Geographical Systems*, 21(3), 395–432. <https://doi.org/10.1007/s10109-019-00296-0>
- Moreb, A. A. (1996). Linear programming model for finding optimal roadway grades that minimize earthwork cost. *European Journal of Operational Research*, 93(1), 148–154. [https://doi.org/10.1016/0377-2217\(95\)00095-X](https://doi.org/10.1016/0377-2217(95)00095-X)
- Wang, S., Guo, T., & Luo, M. (2016). Design Specification for Highway Alignment (Jtg D20-2017). People's Transportation Press, Beijing.
- Wolhuter, K. (2019). Geometric design of roads handbook. CRC Press.

## Appendix: Algorithms

---

### Algorithm 1. Determination of a VPI feasible region for adding a new VPI

---

**Input:**  $L_{i-1}$ ,  $L_{c(i-1)}$ ,  $VPI_i(Stn_i, Elv_i)$ ,  $E(Stn_E, Elv_E)$ ,  $G_{min}$ ,  $G_{max}$ ,  $L_{min}$ ,  $L_{max}(G_i)$ ,  $L_{cmin}$ , and  $K_{min}$ .

**Output:**  $FD_{A3}$

- 1:  $FD_{g1} \Leftarrow$  Create a new region object based on Definition 1 with inputs  $VPI_i(Stn_i, Elv_i)$ ,  $G_{min}$ , and  $G_{max}$
  - 2:  $FD_{g2} \Leftarrow$  Create a new region object based on Definition 3 with inputs  $VPI_i(Stn_i, Elv_i)$ ,  $L_{min}$ , and  $L_{max}(G_i)$
  - 3:  $FD_{A1} \Leftarrow FD_{g1} \cap FD_{g2}$
  - 4:  $FD_{c1} \Leftarrow$  Create a new region object based on Definition 5 with inputs  $L_{i-1}$ ,  $L_{c(i-1)}$ ,  $L_{cmin}$ , and  $K_{min}$
  - 5:  $FD_{A2} \Leftarrow FD_{c1} \cap FD_{A1}$
  - 6: If  $E(Stn_E, Elv_E)$  exists, then
  - 7:  $FD_3 \Leftarrow$  Create a new region object based on Definition 9 with inputs  $E(Stn_E, Elv_E)$  and  $FD_{A2}$
  - 8: else
  - 9:  $FD_3 \Leftarrow \emptyset$
  - 10:  $FD_{A3} \Leftarrow FD_{A2} - FD_3$
- 

### Algorithm 2. Determination of a VPI feasible region for moving the middle VPI

---

**Input:**  $L_{i-2}$ ,  $L_{i+1}$ ,  $L_{c(i-2)}$ ,  $L_{c(i+2)}$ ,  $VPI_{i-1}(Stn_{i-1}, Elv_{i-1})$ ,  $VPI_{i+1}(Stn_{i+1}, Elv_{i+1})$ ,  $E(Stn_E, Elv_E)$ ,  $G_{min}$ ,  $G_{max}$ ,  $L_{min}$ ,  $L_{max}(G_i)$ ,  $L_{cmin}$ , and  $K_{min}$ .

**Output:**  $FD_{M3}$

- 1:  $FD_f \Leftarrow$  Create a new region object according to the creation process of  $FD_{A1}$  with inputs  $VPI_{i-1}(Stn_{i-1}, Elv_{i-1})$ ,  $G_{min}$ ,  $G_{max}$ ,  $L_{min}$ , and  $L_{max}(G_i)$
  - 2:  $FD_r \Leftarrow$  Create a new region object according to the creation process of  $FD_{A1}$  with inputs  $VPI_{i+1}(Stn_{i+1}, Elv_{i+1})$ ,  $G_{min}$ ,  $G_{max}$ ,  $L_{min}$ , and  $L_{max}(G_i)$
  - 3:  $FD_{M1} \Leftarrow FD_f \cap FD_r$
  - 4:  $FD_{cf} \Leftarrow$  Create a new region object according to the creation process of  $FD_{c1}$  with inputs  $VPI_{i-1}(Stn_{i-1}, Elv_{i-1})$ ,  $L_{i-2}$ ,  $L_{c(i-2)}$ ,  $L_{cmin}$ , and  $K_{min}$
  - 5:  $FD_{cm} \Leftarrow$  Create a new region object based on Definition 9 with inputs  $VPI_{i-1}(Stn_{i-1}, Elv_{i-1})$ ,  $VPI_{i+1}(Stn_{i+1}, Elv_{i+1})$ ,  $L_{cmin}$ , and  $K_{min}$
  - 6:  $FD_{cr} \Leftarrow$  Create a new region object according to the creation process of  $FD_{c1}$  with inputs  $VPI_{i+1}(Stn_{i+1}, Elv_{i+1})$ ,  $L_{i+1}$ ,  $L_{c(i+2)}$ ,  $L_{cmin}$ , and  $K_{min}$
  - 7:  $FD_{c2} \Leftarrow FD_{cf} \cap FD_{cm} \cap FD_{cr}$
  - 8:  $FD_{M2} \Leftarrow FD_{c2} \cap FD_{M1}$
-

- 
- 9: If  $E(Stn_E, Elv_E)$  exists, then
  - 10:  $FD_5 \Leftarrow$  Create a new region object based on Definition 12 with inputs  $E(Stn_E, Elv_E)$  and  $FD_{M2}$
  - 11: else
  - 12:  $FD_5 \Leftarrow \emptyset$
  - 13:  $FD_{M3} \Leftarrow FD_{M2} - FD_5$
- 

## Notations

- $A$  – Algebraic difference in gradient;  
 $A_i$  – Algebraic difference between  $G_i$  and  $G_{i+1}$ ;  
 $C$  – Constraint set;  
 $C_a$  – Constraint of tie points above which the vertical alignment must pass;  
 $C_b$  – Constraint of tie points below which the vertical alignment must pass;  
 $C_c$  – Vertical curve constraint;  
 $C_E$  – Control-elevation constraint;  
 $C_t$  – Constraint of tie points through which the vertical alignment must pass;  
 $C_g$  – Grade constraint;  
 $C_{g1}$  – Gradient constraint;  
 $C_{g2}$  – Grade-length constraint;  
 $C_S$  – Design-specification constraint;  
 $Elv_i$  – Design elevation of  $VPI_i$ ;  
 $f$  – Mapping of design-constraint set to design the feasible-region set;  
 $FD$  – Design feasible-region set;  
 $FD_1$  – Definite feasible region derived from  $FD_{A2}$ ;  
 $FD_2$  – Possible region derived from  $FD_{A2}$ ;  
 $FD_3$  – Impossible region derived from  $FD_{A2}$ ;  
 $FD_4$  – Definite feasible region derived from  $FD_{M2}$ ;  
 $FD_5$  – Impossible region derived from  $FD_{M2}$ ;  
 $FD_{A1}$  – Feasible region of adding a new VPI under constraint  $C_g$ ;  
 $FD_{A2}$  – Feasible region of adding a new VPI under constraints  $C_g$  and  $C_c$ ;  
 $FD_{A3}$  – Feasible region of adding a new VPI under constraints  $C_S$  and  $C_E$ ;  
 $FD_{c1}$  – Feasible region of adding a new VPI under constraint  $C_c$ ;  
 $FD_{c2}$  – Feasible region of moving a middle VPI under constraint  $C_c$ ;  
 $FD_{cf}$  – Feasible region of moving a middle VPI under constraint  $VC_{i-1}$ ;  
 $FD_{cm}$  – Feasible region of moving a middle VPI under constraint  $VC_i$ ;  
 $FD_{cr}$  – Feasible region of moving a middle VPI under constraint  $VC_{i+1}$ ;  
 $FD_f$  – Feasible region of moving a middle VPI under constraint  $C_f$ ;

$FD_{g1}$  – Feasible region of adding a new VPI under constraint  $C_{g1}$ ;  
 $FD_{g2}$  – Feasible region of adding a new VPI under constraint  $C_{g2}$ ;  
 $FD_{M1}$  – Feasible region of moving a middle VPI under constraint  $C_g$ ;  
 $FD_{M2}$  – Feasible region of moving a middle VPI under constraints  $C_g$  and  $C_c$ ;  
 $FD_{M3}$  – Feasible region of moving a middle VPI under constraints  $C_S$  and  $C_E$ ;  
 $FD_r$  – Feasible region of moving a middle VPI under constraint  $C_r$ ;  
 $G$  – Gradient between two VPIs;  
 $G_i$  – Gradient between  $VPI_{i-1}$  and  $VPI_i$ ;  
 $G_{max}$  – Maximum gradient according to the design specification;  
 $G_{min}$  – Minimum gradient according to the design specification;  
 $I$  – Input set;  
 $K$  – Rate of vertical curvature;  
 $K_i$  – Rate of vertical curvature at  $VPI_i$ ;  
 $K_{imin}$  – Minimum rate of vertical curvature at  $VPI_i$ ;  
 $l_1$  – Minimum longitudinal gradient constraint line;  
 $l_2$  – Maximum longitudinal gradient constraint line;  
 $l_3$  – Minimum length of the constraint line;  
 $l_4$  – Critical length of the constraint line;  
 $L$  – Length of the grade between two VPIs;  
 $L_c$  – Length of vertical curve;  
 $L_{c(i)}$  – Length of VC at  $VPI_i$ ;  
 $L_{cmin}$  – Minimum length of VC for a specified design speed;  
 $L_i$  – Length of the grade between  $VPI_i$  and  $VPI_{i+1}$ ;  
 $L_{max}(G_i)$  – Critical length of the steep grade between two VPIs for specified grade and design speed;  
 $L_{min}$  – Minimum length of the grade between two VPIs;  
 $O$  – Output set;  
 $P$  – Design-process set;  
 $Q$  – State set;  
 $Stn_i$  – Design station of  $VPI_i$ ;  
 $VC$  – Vertical curve;  
 $VC_i$  – Vertical curve at  $VPI_i$ ;  
 $VPI$  – Vertical points of intersection;  
 $VPI_i$  –  $i^{th}$  vertical points of intersection;  
 $\theta$  – Mapping of design states;  
 $\Phi$  – Mapping set.