Original scientific paper

Tabu search optimization of forest road alignments combined with shortest paths and cubic splines

Kazuhiro Aruga, Toshiaki Tasaka, John Sessions, Edwin S. Miyata

Abstract – Nacrtak

This paper describes a program for optimizing forest road alignments using the Dijkstra shortest path method and a cubic spline function. We previously developed a method for optimizing forest road alignments once a series of intersection points (IPs) were selected manually using Tabu Search. The application of the program to a part of Capitol State Forest in Washington State, USA indicated that the program successfully found better alignments than manually selected initial alignments. In order to find initial solutions without manually initialized solutions, the Dijkstra method and a cubic spline function were combined with our optimization program. The Dijkstra method connected some segments between two end points and the spline function generated smooth vertical alignments between two end points based on the horizontal alignments. In order to adapt the new method for forest road design to our existing method, the program converted the spline curves to straight and parabolic sections. The solution using a spline function was 10% poorer than the solution without a spline function, but computing time significantly reduced from 73 hours to 19 hours using the spline function. Furthermore, the program generated smooth vertical alignments automatically. This study reports our initial effort to use the spline function in optimal road design. Additional investigation could improve solution quality using the Dijkstra method and cubic splines.

Keywords: *forest road alignment, Tabu Search, cubic spline curve, Dijkstra method, solution quality*

1. Introduction – *Uvod*

We developed the program to simultaneously optimize horizontal and vertical alignments of forest roads based on manually selected initial alignments (Aruga et al. 2005c). The program precisely generates the road prism using a high resolution Digital Elevation Model (DEM) and accurately calculates earthwork volume with the Pappus-based model (Easa 2003). The program successfully optimized the horizontal and vertical alignments simultaneously (Aruga et al. 2005c). In the study, we used Tabu Search to optimize a forest road alignment because a previous comparison of Tabu Search and the Genetic Algorithm applied to a forest road profile (vertical alignment optimization with fixed horizontal alignment) indicated that Tabu Search found a good solution in less time than the Genetic Algorithm (Aruga et al. 2005a). However, the Genetic Algorithm found slightly better solutions than Tabu Search when optimizing a forest road profile, even though it took longer. Therefore, we continued to examine the relative efficiencies of Tabu Search compared to the Genetic Algorithm for solution quality and computing time for the simultaneous solution of horizontal and vertical road alignment (Aruga et al. 2005d). Although coding skill, fine-tuning of algorithms, and testing of parameters were different between the Genetic Algorithm and Tabu Search, the study indicated that our Tabu Search procedure was more suitable for forest road design than our Genetic Algorithm.

In highway design, Chew et al. (1989) developed a program to optimize a 3-dimensional alignment simultaneously using cubic spline functions. The model transformed the constraints into one-dimensional constraints by the method of constraint transcription used in the optimal control theory. Then, the model becomes a constrained nonlinear program structure with the coefficient vectors of spline functions as its decision variables. As the objective function including integrals are not easy to compute, a numerical integration, the quasi Newton descent algorithm, is used for computation during search. Although this model optimizes 3-dimensional alignment smoothly, the model simplified road costs and did not make a detailed earthwork volume estimate for each of alternative road locations. Furthermore, different initial solutions with human judgment needed to be input in order to run the model. Jong and Schonfeld (2003) developed a model for simultaneously optimizing three dimensional highway alignments with a Genetic Algorithm. Their model generated initial solutions automatically and, while considering various cost components and constraints, found reasonably good solutions efficiently.

Our previous procedure used a Tabu Search with short term memory followed by an intensification procedure. Diversification relied on the user to specify the number and location of intersection points for the horizontal alignment and the number of grade change points for the vertical alignment. To assist the user, the number and location of intersection points for the horizontal alignment and the number of grade change points for the vertical alignment could be generated by the model. To avoid manually initializing intersection and grade change points, the Dijkstra method (Smith 1982) and a spline function (Schumaker 1993) were combined with our optimization program. First, the Dijkstra method connects some segments between two end points automatically. Then, the spline function generates smooth vertical alignments between two end points on the horizontal alignments. In order to adapt the new method for forest road design to our existing method, our program converts the spline curves to straight and parabolic sections (Tasaka et al. 1996). This study first describes our forest road design optimization program combined with the Dijkstra method and the spline function. Then, we apply the model to the high-resolution DEM from LiDAR data of Capitol State Forest in Washington State, USA. Finally, solution quality and computing time of the model with the Dijkstra method and the spline function are discussed.

2. Working methods and materials – *Metode rada i materijal*

2.1 Computer Model – *Računalni model*

Before using the model, a designer determines the placement of intersection points (IPs) on Geographic Information System (GIS) or Computer-aided design (CAD) which shows contour maps, orthophotographs, land slide risk map, and endangered species habitat to help a designer select a good route location (Figure 1). Since the program was ex-

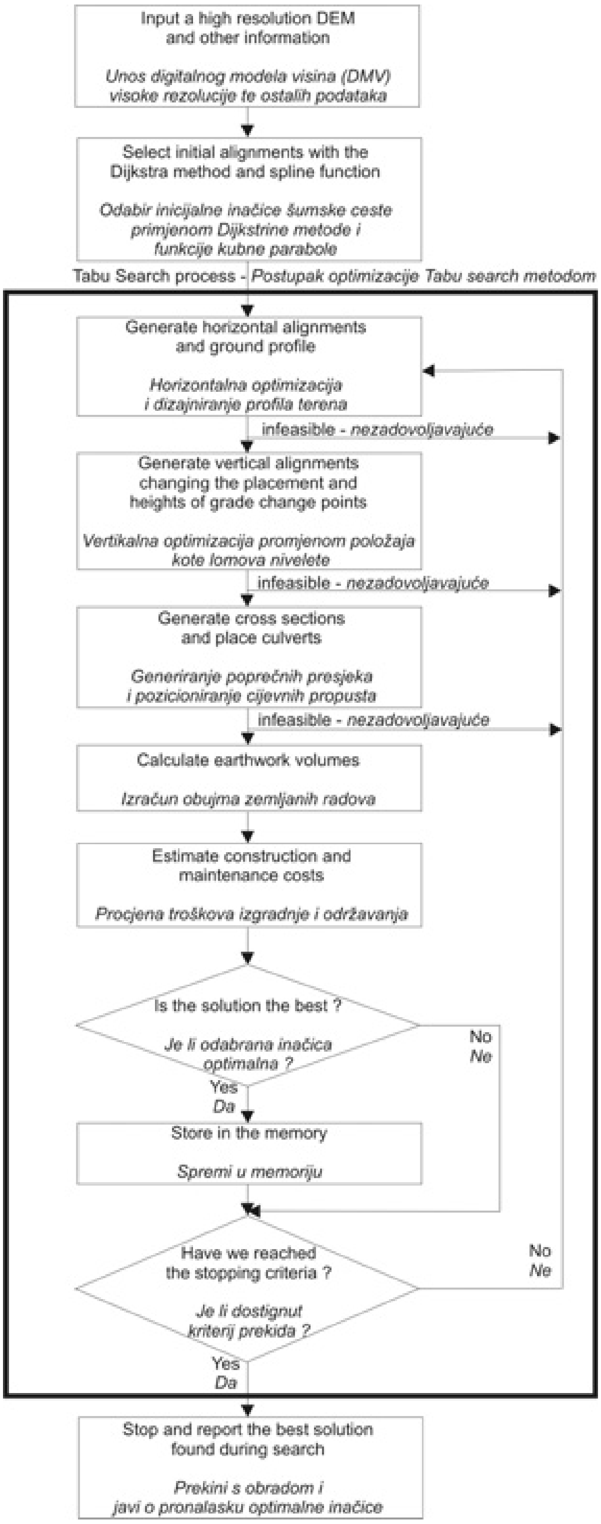

Figure 1 Flowchart of the model

Slika 1. *Dijagram toka podataka modela optimizacije trase šumske ceste* tended to generate initial horizontal alignments using the Dijkstra method, a designer does not have to determine the placement of IPs. Second, the spline function is used in order to generate smooth vertical alignments. Because our existing program uses tangents and curves, our procedure converted the spline curves to straight and parabolic sections. If it is not necessary to use our existing method, the spline curves could be used as vertical alignments directly due to their smoothness. Third, the program generates cross sections, calculates earthwork volumes, and estimates construction and maintenance costs. Finally, it optimizes forest road alignments using Tabu Search with the objective of minimizing total cost. In the following sections, we describe the method to generate initial alignments using the Dijkstra method and the spline function. Cost calculations have been documented in Aruga et al. (2005a), earthwork volume calculations have been documented in Aruga et al. (2005b), and geometric specifications and optimization techniques have been documented in Aruga et al. (2005c).

2.2 Dijkstra method – *Dijkstra metoda*

The general heading for these methods will be »shortest path problems«, although it is not always necessary to be concerned with measuring the length of a path. The same methods can be used for paths with costs associated with them, in which case one might search for the cheapest path. However, for simplicity, the term shortest path is used in this explanation. In this method, the first step is to ensure that there is a distance associated with every pair of nodes in the network. Dijkstra's method assigns a label to every node in the network. This label is the distance to that node from the start (s) along the shortest path found thus far. The label can be in one of two states; it may be a permanent label, in which case the distance found is along the shortest of all paths, or it may be temporary, corresponding to some uncertainty as to whether the path found is the shortest of all. The method gradually changes temporary labels into permanent ones. Given a set of nodes with temporary labels, the aim is to try and make these labels smaller by finding paths to these nodes using the shortest paths to permanently labeled nodes, followed by an arc from a node with a permanent label. Once this has been done, the node with the smallest temporary label is selected, and its label made permanent. This process is repeated until the terminus (t) has been assigned a permanent label, which must happen eventually, since every time the algorithm is used, one less temporary label is left, and so the number of nodes with temporary labels decreases to zero.

The Dijkstra method is widely used for automatic grid cell-based road route location procedures (Heinimann et al. 2003, Suzuki et al. 1998). By combining the Dijkstra method, initial horizontal alignments would not need to be prespecified before running the program. In order to apply the Dijkstra method, the program divides some sections between Beginning of Point (BP) and End of Point (EP) and some sections between the line from BP to EP and the farthest point in the planning area. Similar to Tabu Search optimization, the program calculates earthwork volumes using 6-meter distances between cross sections (Aruga et al 2005b). Then, the Dijkstra method searches for initial solutions with minimum earthwork volumes from BP to EP (Figure 2).

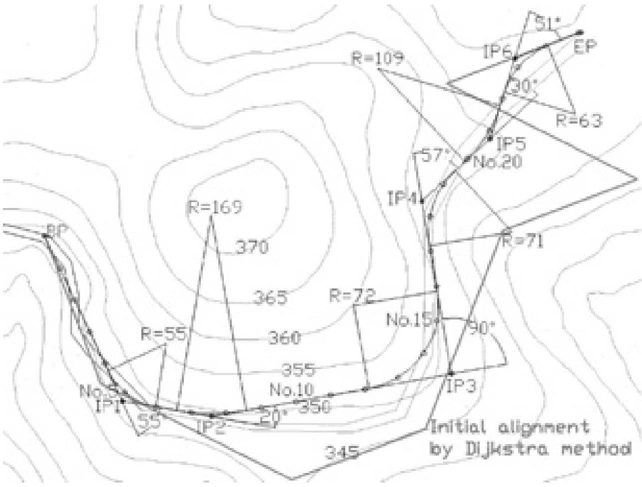

Figure 2 Forest road plane using the initial solution found by the Dijkstra method showing contour intervals, horizontal alignment, and 30-m road

Slika 2. *Prikaz osi inicijalne inačice optimalne trase šumske ceste određene Dijkstrinom metodom na slojničkom planu*

2.3 Cubic spline function – *Matematička funkcija kubne parabole*

The cubic spline function is the following equation (Schumaker 1993):

$$z(t) = A + Bt + Ct^2 + Dt^3 \quad (t_i \leq t \leq t_{i+1}) \tag{1}$$

In this study, $z(t)$ is the vertical coordinate of a point on the spline function where A, B, C, and D are expressed using the coordinates z_i, z_{i+1} and first derivatives z'_i, z'_{i+1} of two end points on the spline function.

$$A = z_i \tag{2}$$

$$B = z'_i \tag{3}$$

$$C=\frac{3(z_{i+1}-z_i)}{t_{i+1}^2}-\frac{2z'_t}{t_{i+1}}-\frac{z'_{i+1}}{t_{i+1}} \qquad (4)$$

$$D=\frac{2(z_i-z_{i+1})}{t_{i+1}^3}-\frac{z'_t}{t_{i+1}^2}-\frac{z'_{i+1}}{t_{i+1}^2} \qquad (5)$$

As second derivatives of the spline function are continuous, the following equation is obtained.

$$t_{i+2}z'_{i+2}(t_{i+1}+t_{i+2})z'_{i+1}+t_{i+1}z'_{i+2}=$$

$$=\frac{3}{t_{i+1}t_i}\left\{t_{i+1}^2(z_{i+2}-z_{i+1})+t_{i+2}^2(z_{i+1}-z_i)\right\} \qquad (6)$$

We assume that curvatures of the start and end points of a forest road are zero. The following two equations can be obtained.

$$z'_1+\frac{z'_2}{2}=\frac{3(z_2-z_1)}{2t_2} \qquad (7)$$

$$z'_{n+1}+2z'_n=\frac{3(z_n-z_{n-1})}{t_n} \qquad (8)$$

Spline curves can be determined using equations 6, 7, and 8. In order to convert spline curves to straight and parabolic sections, the program first finds the points of minimum curvatures on segments. Curvatures $f\varkappa(t)$ and radii $r(t)$ are:

$$\varkappa(t)=\frac{z''(t)}{\sqrt[3/2]{1+z'(t)^2}} \qquad (9)$$

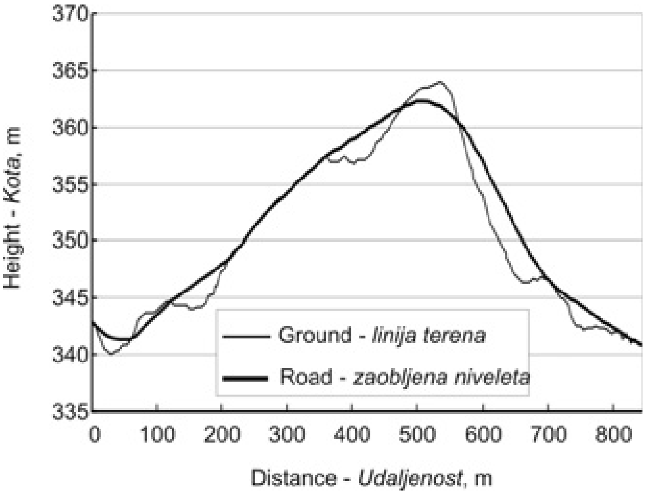

Figure 3 Forest road profile using spline function on the initial horizontal alignment.

Slika 3. *Crtani uzdužni profil inicijalne inačice optimalne trase šumske ceste određen Schumakerovim matematičkim izrazom kubne parabole*

$$\mathrm{r(t)}=\frac{1}{\varkappa(t)} \qquad (10)$$

Then, the model identifies stations near the points of minimum curvatures as grade change points. Grade change points are selected from stations on straight roadways. The model does not select from stations on curved roadways because it is assumed that the road grade remains constant along horizontal curves. Converted straight and parabolic sections from spline curves are shown in Figure 3.

2.4 Optimization algorithm – *Algoritam optimiziranja*

After the initial solution is determined using the Dijkstra method and spline function, the simultaneous optimization process is conducted using Tabu Search (Figure 1). There are a number of parameters which must be determined as to whether they will be fixed or variable. For each iteration, the model examines the locations of IPs and radii of horizontal curves. After examining horizontal alignments, the model examines the placement of grade change points. Then, the model examines the heights of grade change points. The model conducts this process until a specified number of iterations.

After the best solution is found using the above procedure, the program intensifies the search in the region of the best solution. Tabu Search is again applied to examine vertical heights of grade change points around the best alignment in the first search process. This second search process is done while the placement of grade change points and IPs, and radii of horizontal curves are fixed. This second search process refines the best solution found from the first process and provides a better solution for the final best alignment. This second search process stops after another specified number of iterations.

3. Research results and discussions – *Rezultati istraživanja s raspravom*

3.1 Computer model application – *Primjena računalnoga modela*

The study site was a part of Capitol State Forest in Washington State, USA. The site was covered by 70-year-old coniferous forests such as Douglas-fir, western hemlock, and redcedar. Dominant tree height was approximately 50 m. This site was measured by a small footprint LiDAR system in the spring of 1999 and the LiDAR data was converted into a 1.52 m x 1.52 m grid DEM. After the study site was measured

Table 1. Total costs with different numbers of sections using the Dijkstra method

Tablica 1. *Ukupan trošak i potrebno vrijeme obrade pojedine inačice trase šumske ceste za različit broj sekcija i lomnih točaka nivelete uz primjenu Dijkstrine metode*

Section – Sekcija	4 x 2			6 x 3		
No. grade change points - *Broj lomova nivelete*	Computing time - *Vrijeme obrade podataka*	Length - Duljina (m)	Total Cost - *Ukupan trošak* ($)	Computing time - *Vrijeme obrade podataka*	Length - *Duljina* (m)	Total Cost - *Ukupan trošak* ($)
1	13 h 44 min	763	39 815			
2	10 h 22 min	753	39 105	10 h 03 min	760	40 085
3	6 h 00 min	753	39 004	5 h 17 min	724	38 612
4	10 h 16 min	754	38 527	22 h 00 min	717	35 438
5	11 h 17 min	751	37 259	13 h 00 min	716	35 323
6	10 h 26 min	751	37 183	11 h 37 min	715	34 726
7	3 h 47 min	751	37 053	11 h 24 min	670	32 439
Total time – *Ukupno vrijeme*	65 h 52 min			73 h 21 min		

by LiDAR in 1999, the forest road was extended. We examined the alignments of this extended road.

In order to apply the Dijkstra method, the program divided four sections between the BP and EP and two sections between the line from BP to EP and the farthest point in the planning area. Then, the Dijkstra method was used to search for initial solutions with minimum earthwork volumes from the BP to EP (Figure 2). The initial solution found by the Dijkstra method was quite different from the extended road. The road length was 952 m. Average and total cost of the initial solution were $282,33/m and $269 027, respectively. This total cost was much larger than the initial solution of manually initialized alignments, $71 680. In spite of this poorer initial alignment, subsequent Tabu Search, which examined grade change points sequentially without the spline function with 1 000 iterations for the first search process and 100 iterations for the second search process, found good horizontal and vertical alignments. Road length was 751 m (Table 1). Average and total costs of solution with seven grade change points were $49,27/m and $37 053, respectively.

Then, we applied the spline function to generate vertical alignments. The spline function generated smooth vertical alignments with seven grade change points (Figure 3). Then, Tabu Search optimized horizontal and vertical alignments based on the initial alignments. Road length of the solution found by subsequent Tabu Search was 810 m (Table 2). Total cost of the solution was $37 224. Therefore, the solution was a little poorer than that without spline function. However, computing time was significantly reduced from 65 hours (Table 1) to 11 hours (Table 2).

Table 2 Total costs with different numbers of sections using the Dijkstra method and the spline function (* is calculated by Tabu Search with 200 iterations without any improvements on the objective function of the best solution found)

Tablica 2. *Ukupan trošak i potrebno vrijeme obrade pojedine inačice trase šumske ceste za različit broj sekcija uz primjenu Dijkstrine metode i Schumakerove matematičke funkcije kubne parabole*

Section – *Sekcija*	No. grade change points - *Broj lomova nivelete*	Iteration – *Broj ponavljanja*	Computing time- *Vrijeme obrade podataka*	Length- *Duljina* (m)	Total Cost - *Ukupan trošak* ($)
4 x 2	7	1 000	11 h 38 min	810	37 224
	7	*661	18 h 34 min	825	37 729
6 x 3	12	1 000	19 h 24 min	765	35 971
	12	*489	8 h 08 min	765	35 971

3.2 The number of sections – *Broj sekcija*

We examined effects of the number of sections because the Dijkstra method with more sections is expected to generate smoother initial horizontal alignments. The model with 6x3 sections could find a little smoother initial horizontal alignment. However, the road length, 1 050 m, was longer, and average and total costs of initial solution, $453,34/m and $476 235, were higher than those with 4x2 sections. Furthermore, it could not find a feasible vertical alignment with one grade change point (Table 1). The best solution with seven grade change points, $32 439, was better than that on 4x2 sections, $37 053, and slightly better than that started from the manually initialized horizontal alignment, $32 701 (Table 3). The road length was 670 m. Moreover, the amount of material borrowed from or disposed to out of the road construction area with 6x3 sections, 4,20 m^3, was smaller than 16,52 m^3 with 4x2 sections while the amount of material for the manually initialized horizontal alignment was the smallest, 1,75 m^3. The model with 8x4 sections could not find a feasible horizontal alignment because the feasible horizontal curves could not be generated between two straight roadway sections due to shorter intervals.

Similarly, we applied the spline function to generate vertical alignments on the initial horizontal alignment found by the Dijkstra method with 6x3 sections. The spline function generated smooth vertical alignments with twelve grade change points (Table 2). Then, Tabu Search optimized horizontal and vertical alignments based on the initial alignments. Road length of the solution found by subsequent Tabu Search was 765 m (Table 2). Total cost of the solution was $35 971. It was 10% poorer than that without the spline function. However, computing time was significantly reduced from 73 hours to 19 hours. Furthermore, the program generated smooth vertical alignments automatically. Our initial experience suggests the spline function could be a potentially important tool for automated optimization of forest road alignments, but it needs more investigations. For the range of sections examined, the solution optimized from the initial solution by the Dijkstra method with 6x3 sections was better than that from the initial solution found by the Dijkstra method with 4x2 sections. The Dijkstra method with more sections generated smoother initial horizontal alignments and the spline function generated smoother initial vertical alignments based on the initial horizontal alignments. Then, Tabu Search found better forest road alignments.

3.3 Stopping criteria – *Kriterij prekida*

We previously examined the effect of stopping criteria on the results of Tabu Search which searched from a manually initialized horizontal alignment (Aruga et al 2005c). We used 1 000 iterations for the first search process and 100 iterations for the second search process. Since Tabu Search obtained a big gain within 100 iterations, we have examined the effect of reducing the number of the first search process iterations, which was 100 instead of 1 000, on the results (Aruga et al. 2005c). The result showed that the difference between the results during 100 iterations and 1 000 iterations with seven grade change points was 5% (Table 3). Although Tabu Search found slightly poorer solutions during 100 iterations

Table 3 Total costs with different numbers of iterations (gray areas indicate the best solution)

Tablica 3. *Ukupan trošak pojedine inačice trase šumske ceste s različitim brojem ponavljanja i različitim brojem lomnih točaka nivelete (sivom je bojom obilježena optimalna inačica)*

Iteration – *Broj ponavljanja*	1 000		100		200		300	
No. grade change points – *Broj lomova nivelete*	Length – *Duljina* (m)	Total Cost – *Ukupan trošak* ($)	Length – *Duljina* (m)	Total Cost – *Ukupan trošak* ($)	Length – *Duljina* (m)	Total Cost – *Ukupan trošak* ($)	Length – *Duljina* (m)	Total Cost – *Ukupan trošak* ($)
1	730	37 761	760	39 573	760	39 573	760	39 573
2	712	36 472	740	37 989	742	36 662	705	35 674
3	716	35 473	740	37 929	742	36 587	709	35 528
4	716	35 168	740	37 762	742	36 709	694	34 631
5	670	32 946	738	36 239	734	37 292	701	34 012
6	670	32 766	738	35 824	734	37 117	694	33 248
7	670	32 701	699	34 312	720	35 766	684	32 599

than those during 1 000 iterations, computing time was significantly reduced from 44 hours to 7 hours (Table 4). We continued to examine the effect of stopping criteria on solution quality and computing time for the simultaneous solution of horizontal and vertical road alignment. We examined 200 and 300 iterations for the first process (Table 3). Although Tabu search with more iterations were expected to find better solutions, our Tabu search with 200 iterations could not find better solutions than that with 100 iterations (Table 3). Tabu search with 300 iterations found better solutions than that with 100 iterations as expected. Moreover, Tabu search with 300 iterations found better solution than that with 1 000 iterations.

The stopping rule could also have been a certain number of iterations »without any improvement on the objective function of the best solution found«. Therefore, we examined 100, 200, and 300 iterations without any improvement on the objective function of the best solution found (Table 5). Tabu search with 200 iterations found the best solution. Furthermore, Tabu search with 200 iterations took shortest time in Tabu searches which found the best solutions (Table 4, 6). Therefore, it might be a better option to use 200 iterations without any improvement on the objective function of the best solution found for less computing time in this problem.

We also examined the effect of stopping criteria on the results with the Dijkstra method and the spline function (Table 2). Tabu Search with 6x3 sections using the spline function with 200 iterations without any improvement on the objective function of the best solution found, found the same solution with that during 1 000 iterations and computing time with 200 iterations without any improvement

Table 4 Computing time with different numbers of iterations (gray areas indicate the best solution)

Tablica 4. *Vrijeme računalne obrade podataka za pojedinu inačicu trase šumske ceste s različitim brojem ponavljanja i različitim brojem lomnih točaka nivelete (sivom je bojom obilježena optimalna inačica)*

Iteration – *Broj ponavljanja*	1 000	100	200	300
1	6 h 02 min	1 h 37 min	2 h 03 min	2 h 38 min
2	6 h 30 min	1 h 42 min	3 h 27 min	4 h 10 min
3	6 h 22 min	1 h 06 min	1 h 29 min	2 h 04 min
4	7 h 48 min	0 h 53 min	2 h 20 min	2 h 49 min
5	7 h 21 min	0 h 59 min	1 h 39 min	2 h 24 min
6	6 h 12 min	0 h 49 min	1 h 47 min	2 h 26 min
7	4 h 28 min	0 h 44 min	2 h 09 min	3 h 05 min
Total time – *Ukupno vrijeme*	44 h 43 min	7 h 50 min	14 h 54 min	19 h 36 min

Table 5 Total costs with different numbers of iterations without any improvements on the objective function of the best solution found (gray areas indicate the best solution)

Tablica 5. *Ukupni trošak pojedine inačice trase šumske ceste uz korištenje različitih ograničenja tzv. »kontrolnih« ponavljanja nakon pronalaženja optimalnoga rješenja i uz različit broj lomnih točaka nivelete (sivom je bojom obilježena optimalna inačica)*

No. grade change points – *Broj lomova nivelete*	Iteration – *Broj ponavljanja*	Length – *Duljina*	Total Cost – *Ukupan trošak*	Iteration – *Broj ponavljanja*	Length – *Duljina*	Total Cost – *Ukupan trošak*	Iteration – *Broj ponavljanja*	Length – *Duljina*	Total Cost – *Ukupan trošak*
	100	(m)	($)	200	(m)	($)	300	(m)	($)
1	175	760	39 573	275	760	39 573	375	760	39 573
2	171	742	36 662	418	705	35 674	518	705	35 674
3	100	742	36 587	202	709	35 528	302	709	35 528
4	100	742	36 592	321	694	34 631	421	694	34 631
5	100	742	36 017	204	701	34 012	304	701	34 012
6	100	742	36 111	202	694	33 248	302	694	33 223
7	101	740	35 695	238	684	32 599	338	684	32 599

Table 6 Computing time with different numbers of iterations without any improvements on the objective function of the best solution found (gray areas indicate the best solution)

Tablica 6. *Vrijeme računalne obrade podataka za pojedinu inačicu trase šumske ceste uz korištenje različitih ograničenja tzv. »kontrolnih« ponavljanja nakon pronalaženja optimalnoga rješenja i uz različit broj lomnih točaka nivelete (sivom je bojom obilježena optimalna inačica)*

Iteration – *Broj ponavljanja*	100	200	300
1	1 h 34 min	1 h 41 min	2 h 23 min
2	1 h 06 min	1 h 55 min	2 h 44 min
3	1 h 09 min	1 h 53 min	2 h 03 min
4	1 h 20 min	1 h 36 min	2 h 17 min
5	1 h 25 min	1 h 32 min	2 h 18 min
6	1 h 44 min	1 h 23 min	2 h 20 min
7	1 h 33 min	0 h 20 min	2 h 41 min
Total time – *Ukupno vrijeme*	9 h 51 min	10 h 20 min	16 h 46 min

on the objective function of the best solution found was less than half of that with 1,000 iterations (Table 2). Therefore, we could say that 200 iterations without any improvement on the objective function of the best solution found would be a better option for better solution and less computing time in this problem.

4. Concluding Remarks – *Zaključna razmatranja*

This paper introduced the use of the Dijkstra method and the spline function to our optimization program for forest road alignment to find initial solutions without requiring manually initialized solutions. After the Dijkstra method connected some segments between two end points automatically, the spline function generated smooth vertical alignments. Then, the program converted the spline curves to straight and parabolic sections. Because the spline function generates smooth alignments, it is possible to use the spline function as forest road alignments.

Although the program should be further tested and verified, it could optimize forest road alignments automatically. We have another LiDAR dataset of the Utsunomiya University Forest. Further tests will be conducted using the dataset and comparison between computer designed roads and actual constructed roads would be analyzed. The test site in this study was stable and the ground slope was not steep. However, in Japan where many forests are on deep soil and rainfall is high, the forests are subject to slope failures. Future work should distinguish stable places and locate forest roads in these places. Additional road construction options should be considered including the use of wooden or concrete walls to create smaller cut slope heights (Oohashi 2001). In steep and unstable terrain, constraints on cut slope height could be included in the mathematical formulation either as a hard constraint or as a penalty in the objective function.

Acknowledgement – *Zahvala*

LiDAR data were provided by the USDA Forest Service, PNW Research Station, Resource Management and Productivity Program.

5. References – *Literatura*

Aruga, K., Sessions, J., Akay, A.E., 2005a: Heuristic techniques applied to forest road profile. J. Forest Research 10(2), p. 83–92.

Aruga, K., Sessions, J., Akay, A.E., 2005b: Application of an airborne laser scanner to forest road design with accurate earthwork volumes. J. Forest Research 10(2), p. 113–123.

Aruga, K., Sessions, J., Akay, A.E., Chung, W., 2005c: Simultaneous optimization of horizontal and vertical alignments of forest roads using Tabu Search. International J. Forest Engineering 16(2), p. 137–151.

Aruga, K., Sessions, J., Miyata, E.S., 2005d: Comparison of Tabu Search and the Genetic Algorithm for forest road alignment optimization with soil sediment evaluation. In: Proceedings of COFE 2005: Soil, Water and Timber Management: Forest Engineering Solutions in Response to Forest Regulation, Fortuna, California 2005, CD-ROM.

Chew, E.P., Goh, C.J., Fwa, T.F., 1989: Simultaneous optimization of horizontal and vertical alignments for highways. Transportation Research B 23, p. 315–329.

Easa, S.M., 2003: Estimating earthwork volumes of curved roadways: simulation model. J. Surveying Engineering 129(1), p. 19–27.

Glover, F., 1989: Tabu search-Part I. ORSA J. Comput. 1, p. 190–206.

Heinimann, H.R., Stuchelberger, J., Chung, W., 2003: Improving automatic grid cell based road route location procedures. In: Proceedings of Austro2003: High Tech Forest Operations for Mountainous Terrain, Schlaegl, Austria 2003, CD-ROM.

Jong, J.C., Schonfeld, P., 2003: An evolutionary model for simultaneously optimizing three-dimensional highway alignments. Transportation Research B 37, p. 107–128.

Oohashi, K., 2001: Michi zukuri no subete (*Forest road planning and construction*). National Forestry Extension Association in Japan, Tokyo 2001.

Reeves, C., 1993: Genetic Algorithms. In: Reeves, C., (eds) Modern heuristic techniques for combinatorial problems. John Wiley & Sons, Inc. p. 151–196, New York 1993.

Schumaker, L.L., 1993: Spline functions: Basic Theory. Krieger Publishing Company. Melbourne, Florida, 553pp 1993.

Smith, D.K., 1982: Network optimization practice: a computational guide. Ellis Horwood Limited. p. 55–65, West Sussex 1982.

Suzuki, H., Ichihara, K., Noda, I., 1998: Road planning in forest for recreation. J. Jpn. Forest Engineering Society 13(3), p.151–160.

Tasaka, T., Ochi, S., and Matsuo, T., 1996: Planning method of terrain acceptable forest-roads applying with spline function. Bull. Utsunomiya Univ. Forest 32, p. 17–26.

Sažetak

Računalna optimizacija trase šumske ceste kombinirana s najkraćom inačicom i kubnom parabolom

U radu je opisan način rada druge, razvijene (dorađene) inačice računalnoga programa za odabir najpovoljnije trase šumske ceste (tzv. »Tabu search« optimizacija), u horizontalnom i vertikalnom smislu, koji je kombiniran s Dijkstrinom metodom najkraćega (najjeftinijega) puta i Schumakerovom funkcijom kubne parabole.

Bazna inačica računalnoga programa za odabir najpovoljnijega položaja trase šumske ceste (Aruga i dr. 2005c) temeljena je na ručno (od projektanta na odgovarajućim GIS podlogama) odabranim poligonskim točkama (tjemena). Kao podloga služe slojnički zemljovidi, digitalni model visina (DMV) visoke rezolucije, ortofoto snimci te zemljovidi erodibilnih i zaštićenih područja. Za pojedine se inačice trasa predmetne šumske ceste vrlo točno izračunava obujam zemljanih radova tzv. »Pappus« modelom (Easa 2003), ali prethodno projektant treba definirati i lomne točke nezaobljene nivelete. Prva je inačica računalnoga programa testirana na dijelu državnih šuma države Washington u Sjedinjenim Američkim Državama i pokazala je vrlo dobre rezultate.

U daljnjoj se fazi razvoja računalnoga programa željelo u što većoj mjeri automatizirati cjelokupan postupak optimizacije te je u tu svrhu iskorištena Dijkstrina metoda horizontalne (položajne) optimizacije trase šumske ceste. On funkcionira na principu traženja najkraće dionice (u ovom slučaju to znači najjeftinije dionice) pojedinoga segmenta trase i omogućuje horizontalnu optimizaciju trase šumske ceste bez prethodnoga projektantskoga ručnoga unošenja osovinskoga poligona na odgovarajuće GIS podloge. Dijkstrina je metoda često upotrebljavana metoda pri postupku raščlambe i odabiru najpovoljnijih trasa šumske ceste (Heinimann i dr. 2003, Suzuki i dr. 1998). Poprečni se presjeci nalaze na međusobnoj udaljenosti od 6,00 m (Aruga i dr. 2005b), a temeljem minimalnih zemljanih radova određuje se inicijalna (početna) optimalna inačica trase šumske ceste.

Za svaku se inačicu horizontalno optimizirane trase Schumakerovom funkcijom (matematički izraz 1) kubne parabole obavlja optimizacija zaobljene nivelete (pri čemu se koriste matematički izrazi od 2 do 8). Zbog načina se rada računalnoga programa na ovaj način formirana zaobljena niveleta transformira u zaobljenu niveletu sastavljenu od ravnih pravaca vertikalnih krivina (položaj verikalne krivine i njezin radijus određuju se matematičkim izrazima 9 i 10). Lomne se točke nivelete odabiru isključivo na ravnim pravcima trase šumske ceste u horizontalnoj projekciji, budući da se pretpostavlja kako uzdužni nagib mora biti stalan čitavim horizontalnim kružnim lukom.

Nakon na opisani način odabrane bazne (početne) optimalne inačice trase šumske ceste složeni se postupak optimizacije odvija sukladno dijagramu toka podataka prikazanom na slici 1. Detaljna se (fina) raščlamba obavlja u bližem području inicijalne inačice trase šumske ceste i stalnim se ponavljanjem (čiji se

broj može definirati) postupka optimizacije dolazi do sve bolje i bolje trase šumske ceste.

Računalni program dizajnira poprečne presjeke, izračunava zemljane radove te daje procjenu ukupnih troškova izgradnje i kasnijega održavanja pojedine inačice šumske ceste. Konačno se minimiziranjem ukupnih troškova (troškova izgradnje i održavanja) definira najpovoljnija inačica trase šumske ceste.

Istraživanje je provedeno u sedamdesetogodišnjim sastojinama četinjača (duglazija, čuga i crveni cedar) s prosječnom visinom dominantnih stabala od 50 m. Područje je istraživanja snimljeno sustavom LiDAR u proljeće 1999. godine te je iz prikupljenih podataka izrađen DMV formata 1,52 × 1,52 m. Potom je na terenu projektirana šumska cesta koja je poslužila za usporedbu s optimalnom inačicom određenom računalnim programom. Dijkstrinom je metodom određena bazna inačica optimalne trase šumske ceste (na principu minimalnih količina zemljanih radova) i ona se prilično razlikovala od projektirane šumske ceste (slika 2). Ukupna je duljina ove bazne inačice iznosila 952 m, ukupan trošak gradnje 269 027,00 $, a prosječan trošak gradnje 282,33 $/m. Trošak je izgradnje projektirane šumske ceste ukupno iznosio 71 680 $. Za traženje smo optimalne inačice trase šumske ceste koristili 1000 ponavljanja kod »grube« optimizacije i 100 ponavljanja kod »fine« optimizacije. Optimalna je konačna inačica trase šumske ceste određena računalnim programom duljine 751 m, ukupnoga troška izvedbe 37 053 $ i prosječnoga troška izvedbe 49,27 $/m. Tada je primijenjena matematička funkcija kubne parabole kojom smo formirali zaobljenu niveletu sa sedam lomnih točaka (slika 3). Na taj je način optimizirana trasa šumske ceste imala duljinu od 810 m i ukupni trošak izgradnje od 37,224 $, što je neznatno slabiji rezultat nego bez matematičke funkcije kubne parabole, ali je vrijeme utrošeno na obradu podataka smanjeno sa 65 sati na 11 sati.

U radu je, za Dijkstrinu metodu optimizacije, istraživan i broj poligonskih točaka jer smo pretpostavili kako bi veći broj tjemena trebao značiti i bolje uklapanje trase šumske ceste u teren, ali i broj lomnih točaka nivelete (od jedne do sedam). Model s više poligonskih točaka dao je terenu prilagođeniju trasu šumske ceste koja je zbog svoje veće duljine u konačnici bila skuplja od trase šumske ceste s manje tjemena. Najbolje je rješenje postignuto kod sedam lomnih točaka nivelete i za formulu sekcija 6 × 3. Ti su rezultati prikazani u tablici 1.

Na formuli smo sekcija 6 × 3, koja se pokazala kvalitetnijom od formule 4 × 2, dok formulu 8 × 4 u danim uvjetima nije bilo moguće primijeniti, proveli daljnji postupak optimizacije Schumakerovom funkcijom kubne parabole te smo dobili zaobljenu niveletu s dvanaest lomnih točaka. Ta »fina« podinačica »Dijkstrine« inačice trase šumske ceste ima duljinu od 765 m i ukupni trošak izgradnje od 35 971 $, što je za oko 10 % lošije u usporedbi bez uporabe Schumackerove funkcije, ali je vrijeme obrade podataka smanjeno sa 73 na 19 sati, uz napomenu kako je zaobljena niveleta dizajnirana automatski (bez pomoći projektanta). Takav rezultat sugerira kako je kubna parabola potencijalno vrlo kvalitetan alat pri automatiziranju postupka optimizacije šumskih cesta, ali je u tom smjeru potrebno provesti dodatna istraživanja.

Pri istraživanju je prve inačice dizajniranoga računalnoga programa (s ručno unesenim poligonskim točkama u prvoj fazi optimizacije – horizontalnoj projekciji), na samom početku istraživanja, za prvu (»grubu«) optimizaciju korišteno 1000, a za drugu (»finu«) optimizaciju 100 ponavljanja (Aruga i dr. 2005c); rezultati su prikazani u tablici 3. Za tu smo dorađenu inačicu računalnoga programa smanjili broj ponavljanja i za horizontalnu i za vertikalnu optimizaciju na 100, 200 i 300 te raščlanili postignute rezultate i potrebno vrijeme računalne obrade. Broj je lomnih točaka nivelete šumske ceste od jedne do sedam, a rezultati su također u tablici 3. Iako smo očekivali kako će veći broj ponavljanja dati bolje rezultate, pri 100 smo ponavljanja dobili bolje rezultate nego kod 200 ponavljanja, dok smo kod 300 ponavljanja dobili najbolje rezultate optimizacije (bolje nego kod 1000 ponavljanja). Vrijeme je računalne obrade podataka za određeni broj ponavljanja i različit broj lomnih točaka nivelete dano u tablici 4.

Umjesto unaprijed definiranoga fiksnoga broja ponavljanja u postupku optimizacije možemo kao kriterij za zaustavljanje daljnje optimizacije (stopping criteria) definirati broj ponavljanja, nakon jednom pronađenoga optimalnoga rješenje, pri kojima se ne događaju nikakvi kvalitativni pomaci. Drugim riječima, optimizacija se dalje ne može odvijati jer je pronađeno rješenje najbolje i cjelokupan postupak optimizacije se zaustavlja. Rezultati raščlambe tzv. »testiranja« najbolje trase šumske ceste i vremena potrebnoga za računalnu obradu podataka prikazani su u tablicama 5 i 6.

Iako dizajnirani računalni program u budućnosti moramo još testirati i potvrditi njegovu učinkovitost, on nesumnjivo omogućuje optimizaciju trase šumske ceste automatski. Budući da posjedujemo bazu podataka prikupljenu snimanjem sustavom LiNDAR za šumu Sveučilišta Utsunomiya, sljedeća će istraživanja biti usmjerena usporedbi i raščlambi postojećih trasa šumskih cesta u sveučilišnoj šumi i računalnim

programom utvrđenih optimalnih trasa šumskih cesta. Također će se u obzir uzeti različita područja glede nagiba terena i opasnosti pojave erozijskih procesa. Treba razviti i u računalni program uključiti dodatne mogućnosti dizajniranja poprečnih presjeka (obložni i potporni zidovi) na strmim i nestabilnim terenima, čime će se osigurati stabilnost objekta i smanjiti troškovi održavanja.

Ključne riječi: *položaj šumske ceste, Tabu Search, kubna parabola, Dijkstrina metoda, kvaliteta rješenja*

Authors address – Adresa autora:

Assoc. Prof. Kazuhiro Aruga, Ph.D.
Department of Forest Science
Faculty of Agriculture
Utsunomiya University, Mine,
Utsunomiya 321-8505
JAPAN
E-mail: aruga@cc.utsunomiya-u.ac.jp

Prof. Toshiaki Tasaka, Ph.D.
Department of Forest Science
Faculty of Agriculture
Utsunomiya University, Mine,
Utsunomiya 321-8505
JAPAN
E-mail: tasaka@cc.utsunomiya-u.ac.jp

Prof. John Sessions, Ph.D.
Department of Forest Engineering
College of Forestry, Oregon State University
Corvallis Oregon 97331
USA
E-mail: john.sessions@oregonstate.edu

Prof. Edwin S. Miyata, Ph.D.
Department of Industrial Engineering
College of Engineering and Forest Resources
University of Washington
Seattle Washington 98195
USA
E-mail: esm@u.washington.edu

Received (*Primljeno*): December 15, 2005
Accepted (*Prihvaćeno*): May 11, 2006