Model and heuristics for the shortest road layout problem

L. Mandow and **J.L. Pérez de la Cruz** [1]

Abstract. The road layout problem consists of finding a valid road layout between two locations described by their azimut and tridimensional coordinates in a topographical map. Valid layouts should comply with certain road design regulations that normally depend on design parameters such as road speed and type (e.g. highway, or conventional road). The paper presents a discrete state space for this problem that approximates real layouts, and discusses heuristic estimates that allow searching for the shortest approximation in such space using the A* heuristic search algorithm. These ideas have been implemented and tested in a road layout editor.

1 INTRODUCTION

Designing a road layout is a complex task that involves many conflicting criteria. These include the topography where the road will be inserted, economic factors (e.g. road length, land movements), and geological, hydrological, or environmental concerns, to name a few. Road design regulations impose their own constraints on valid layouts. These regulations normally aim at achieving homogeneous geometric features that should induce driving without excesive variations in speed, and in comfortable and safe conditions.

This paper considers the generation of valid road layouts focusing mainly on the minimization of road length. The model and heuristics described in this paper have been integrated in a road layout editor and deployed and tested in a civil engineering firm.

Road layout design is a common civil engineering practice with economic importance. Given the a priori similarities of this task with well known AI techniques (i.e., path finding subject to certain constraints) it is somewhat surprising that the problem has reached only limited attention as a potential area for the application of AI. Some work has been done in order to conceptualize and solve the problem from a global point of view, identifying ”obstacles” and searching for a path that avoids them [3]. However, to the author’s knowledge, no previous attempt has been carried out to integrate a realistic road layout generator in a CAD tool.

The rest of the paper is organized as follows. Section 2 describes a discrete state space that approximates road layouts. Section 3 presents admissible heuristic functions for the shortest road layout using the well known A* algorithm [1]. Section 4 presents some experimental results. Section 5 discusses how the ideas presented in the paper have been put in practice. Finally some conclusions are briefly outlined.

2 STATE SPACE DEFINITION

In this paper we approach the road layout problem from a discrete point of view. This section shows how the generation of a geometrically correct layout between two points in a tridimensional space can be approximated by a search in a state space. Formally, the *road layout problem* can be modeled as a configuration problem: given a set of predefined components, find an assembly of components that satisfies the requirements and obeys the constraints. Requirements are typically translated into either constraints (hard requirements) or preferences (soft requirements) [2]. Here, the components are straight lines, arcs, and clotoids. The first two are parametrized by their length, and angle and radius respectively. Clotoids are transition curves with variable radius. These are inserted at each arc or straight line transition to ensure smooth driving. The selection of each component imposes certain constraints on the components that may follow, e.g. the radius of an arc constrains the radii and direction of the next arc.

These geometric constraints may vary depending on the *legal definition* of the road (e.g. highway, or conventional road) and *project speed* (e.g. 40, 60, 80, 100, or 120 Km/h). The term C-80 refers to a conventional road with project speed of 80 Km/h.

One of the main difficulties in a discretization of the road layout problem is the need to keep a sufficiently representative but finite number of possibilities. The state space definition that follows considers only arc sections. Straight lines can be approximated with large radii arcs. It is a widespread practice to ignore clotoids in preliminary road designs. It suffices to consider minimum arc lengths to ensure that clotoids can be properly accomodated at later stages.

In the proposed state space, each state stores detailed information on the road’s plan and elevation.

2.1 Plan layout

While human designers use topographical maps to explore different plan layouts, our model uses *digital terrain models* (heights grids). This introduces a first discretization of the problem.

Basically, our model uses grid points (e.g. with a 100 meter resolution) to approximate road layouts. More precisely, each layout will be an 8-neighbour path over this grid. However, not every 8-neighbour path will map onto a valid road layout. Therefore certain geometric constraints are imposed on succesor generation to ensure that each path approximates a real road layout.

Arc sections are approximated with two different kinds of states: those that represent the transition or union between two arcs, and those that approximate inner points of the arc.

Currently, each state stores the following data regarding plan layout:

- Grid coordinates $x\ y$ of the current state.
- Grid coordinates $x_s\ y_s$ of the arc’s starting point.
- Azimut α_s of the arc at its starting point.
- Range of arc radii approximated for the current arc.

[1] Dpto. Lenguajes y Ciencias de la Computación. Universidad de Málaga. Apartado 4114 - 29080 Málaga (Spain). Email: lawrence@lcc.uma.es perez@lcc.uma.es

- Range of arc radii allowed by road regulations, depending on the previous arc.

Successor states are generated to approximate a given set of arcs. Each arc starts and ends in a grid point. However, intermediate grid points are selected as the arc is generated. Figure 1 shows a point A, which represents the start of a new arc, and a succesor point B, that approximates the range of arcs at a distance d or closer. This distance can be calculated from point B and the centers c_1, c_2 of the arcs. Figure 2 shows a sequence of two arcs (one from point A to point B, and another form point B to point C), and their approximation on the grid.

This model presents several interesting properties

- Choosing a small value for the distance parameter d results in a better approximation, though it also reduces the set of alternatives that can be generated. Our experience has shown that $d = 50m.$ is often a good approximation for grid resolutions of 100 m.
- The model can be easily reversed, i.e. given an arc between two grid points B and C, the same approximation can be found searching from B towards C, and from C towards B.
- At most, the approximated curves will be d meters away from their approximation path grid points.

This turns out to be quite precise in human terms. Preliminary road design stages usually involve topographic maps that span 8 x 14 Km. in a standard sheet of paper. At this scale a thick pen easily yields marks that are about 100 m. wide.

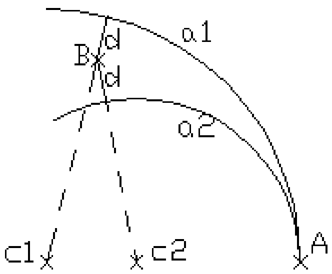

Figure 1. Grid point B, successor of A, approximates arcs with radii ranging from a1 to a2.

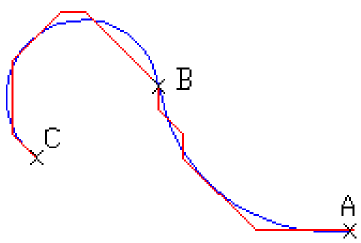

Figure 2. Arc sequence with its approximation on the grid.

2.2 Elevation layout

The elevation of preliminary layouts is usually considered only in very approximate terms. Human designers draw curves in topographical maps according to their experience and the information provided by contour lines. Once the curve sequence of the preliminary layout is outlined in the map, the elevation can be optimized to minimize land movements using more detailed topographical measures. In other words, the optimization of plan an elevation criteria is usually too complex in human terms to be carried out simultaneously.

Layout elevation has to take into account at least two factors. First of all, it should adapt to the underlaying terrain, avoiding excesive separations. Finally, the sequence of slopes and hill brows should comply with road regulations.

Basically, the model considers two different kinds of states according to elevation information: those that represent hill brows, and those that define the slopes between them.

Currently, each state stores the following data regarding layout elevation:

- Layout height z corresponding to the current coordinates $x\,y$, whenever these represent a hill brow.
- Coordinates $x_b\,y_b\,z_b$ of the last hill brow.
- Minimum slope length according to road regulations, depending on the previous slope.
- Length traversed from the last hill brow.
- Range of slopes reaching the current state from the last hill brow. This is calculated incrementally taking into account the range of slopes allowed by the regulation, the maximum terrain separation, and the traversed terrain profile.

The *maximum terrain separation* is currently a configurable parameter that measures how much the road layout might separate vertically from the terrain. Figure 3 shows a possible terrain elevation profile, and a set of valid slope ranges emanating form point A, presumably a hill brow. Figure 4 shows how the maximum terrain separation e constrains this slope range when point B is considered as a successor of point A. It can be easily seen that point C is a possible continuation of the path. However, in such case C should be itself a new hill brow. Each new hill brow state sets its current height as close as possible to the terrain. According to good road design practice, hill brows are introduced only at the start of new road sections.

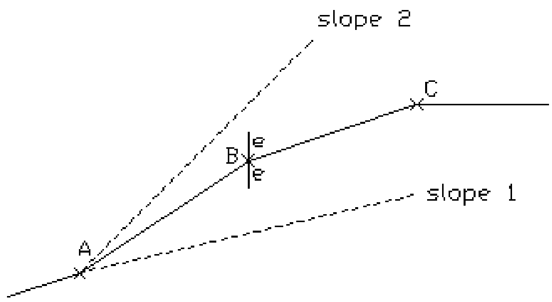

Figure 3. Terrain elevation profile (solid line) and a range of slopes emanating from point A

Terrain separation turns out to have a major impact on the set of possible layouts. Figure 5 shows the shortest layout generated between two locations allowing a maximum separation of 40 m. Figure

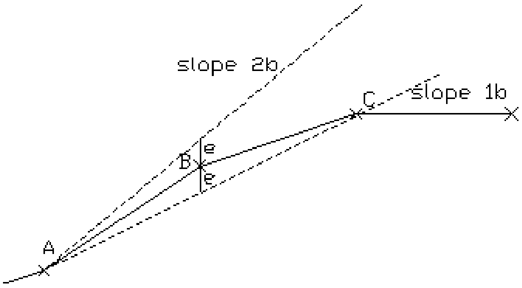

Figure 4. Terrain elevation profile (solid line) and a range of slopes emanating from point A and constrained to a maximum separation of e from point B.

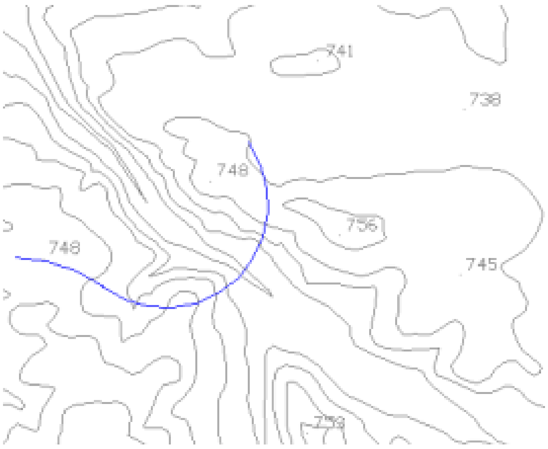

Figure 5. Shortest road layout between two locations (maximum terrain separation of 40 m.)

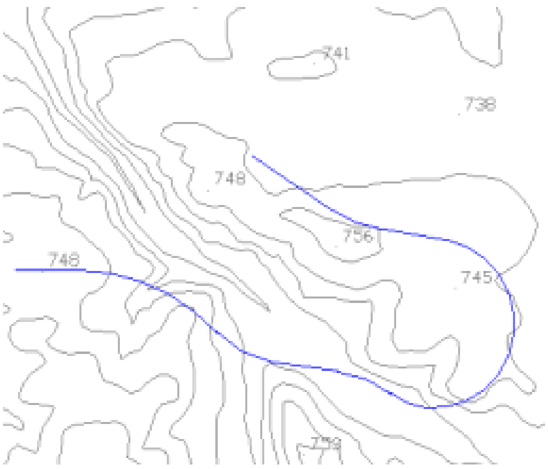

Figure 6. Shortest road layout between two locations (maximum terrain separation of 10 m.)

6 shows the shortest layout for the same problem when the maximum separation is reduced to 10 m. Contour lines are 10 meters appart.

Better terrain adaptation often calls for longer layouts. The model ensures that every generated layout has at least an elevation profile that complies with road regulations.

2.3 PROBLEM DEFINITION

Given the previous state space definition, an *instance of the road layout problem* is defined by an initial state, and a goal point(γ), characterized by its coordinates x_γ, y_γ, z_γ, and the desired goal angle or azimut of the layout at that point (α_γ).

A *solution* will be every path or state sequence that starts at the initial state and ends in a state with coordinates x_γ, y_γ, z_γ that corresponds to an arc tangent to the orientation α_γ at that point.

The *shortest road layout problem* aims at finding the shortest solution.

In principle, conventional optimization graph search algorithms can be used to search for solutions in the proposed state space. The well-known A* algorithm is guaranteed to find the optimal solution given an admissible (optimistic) cost estimate for the problem. Particularly, the algorithm's efficiency can be substantially improved using better and better cost estimates.

3 HEURISTICS

In this section we present simple, but effective, heuristic functions for the shortest road layout problem. Let us denote by e a state, and by Δx, Δy, Δz the distance in each coordinate between γ and e. All examples and graphics shown in this paper refer to C-80 roads with a resolution of $100m$. Similar results can be obtained for other resolutions and road types.

3.1 Grid distance

The 8-neighbour distance between every grid point and the goal yields an optimistic cost estimate for this problem.

$$h_1(e) = \left\{ \begin{array}{ll} (|\Delta x| - |\Delta y|) + |\Delta y|\sqrt{2} & \text{if } |\Delta x| \geq |\Delta y| \\ (|\Delta y| - |\Delta x|) + |\Delta x|\sqrt{2} & \text{if } |\Delta x| < |\Delta y| \end{array} \right\} \tag{1}$$

3.2 Taking into account the goal's azimut

While grid distance is a good medium-range distance estimate, it can be quite poor in the goal's close neighbourhood. Due to the constraints imposed by road regulations on minimum arc radius and the radii of consecutive arcs, two points apparently close in the plane might need long road sections unless they are properly aligned. This is illustrated in figure 7 where, given an initial point and orientation A, a longer road is needed to reach an apparently closer point (C) than a farther one (B). To estimate how close a grid point is to a given goal, a backwards search was carried out from a goal point with $\alpha_\gamma = 180°$. The cost of the optimal layout to every neighbour grid point was recorded and is shown in figure 8. This results are consistent with a minimum arc radius of 265 m for C-80 roads. This search is carried out assuming flat terrain (i.e. elevation information is ignored).

All data calculated backward searching from a point P with starting orientation of $0°$, can be used as optimistic cost estimates for roads that, starting from any given grid point, wish to reach point P with an orientation of $180°$.

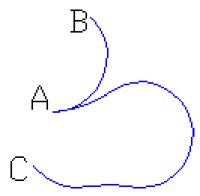

Figure 7. Minimum road lengths from A, with orientation of $0°$, to apparently close destinations in the plane B,C.

These data have been calculated for different grid distances and stored in a lookup table. Since cost estimates depend on the goal's azimut, a different table needs to be calculated for every possible azimut. Our table stores azimut in radians with a precision of two decimals, yielding 628 possible orientations. A complete 11 x 11 heuristic table with 628 orientations should need 75988 data. Using simetries this can be simplified to 79 orientations (9559 data) corresponding to the $0,\pi/4$ radians range. This heuristic is inspired in the *pattern database* idea described in [4].

$$h_2(e) = table(\Delta x, \Delta y, \alpha_{goal}). \quad (2)$$

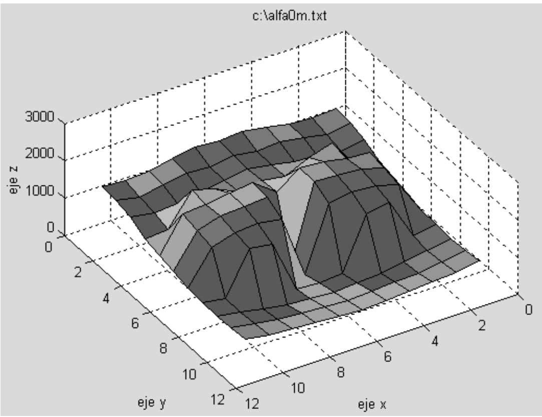

Figure 8. Minimum length for C-80 roads emmanating from the grid's center with $\alpha = 0°$ to near points. Grid resolution is $100m$

4 EXPERIMENTAL TESTS

Table 1 shows the evolution of the mean heuristic estimate of h_1 and h_2 for points in square grids of increasing size. This gives an idea of the savings in search depth achieved by each heuristic. It can be easily seen that h_2 is very effective in the goals close neighbourhood.

To test the heuristics, 30 random layouts were generated with depth 10, 30 with depth 15, and 30 with depth 20 over a given topographical map. Their starting and ending points were taken as sample problems. Then, the A* search algorithm was used to generate all optimal solutions to these problems using $h_1(e)$, and $h_3(e) = max\{h_1(e), h_2(e)\}$ with a 11×11 heuristic lookup table.

Table 1. Mean heuristic values for precompiled lookup tables of increasing size.

table	$h_1(e)$	$h_2(e)$
1×1	0	0
3×3	107.30	1266.29
5×5	193.14	1334.61
7×7	275.91	1288.61
9×9	357.66	1199.95
11×11	438.95	1132.09

The results summarized in tables 2 and 3 show that using h_3 the number of iterations reduces on average to 55.7%, the number of nodes in the search tree to 53.1%, and the running time to 44%. The optimal cost of the solutions ranges from 948 to 2462 meters, with an average of 1656 meters.

Table 2. Summary of experimental results (iterations and nodes).

	Number de iterations		Num.of generated nodes	
	h_1	h_3	h_1	h_3
Total sum	453392.00	252557.00	685190.00	363967.00
Mean	5094.29	2837.62	7698.76	4089.52
Standard dev.	10786.25	6483.86	16638.28	8189.37
Minimum	11.00	11.00	23.00	23.00
Maximum	58462.00	56370.00	105930.00	56934.00

Table 3. Summary of experimental results (time).

	Time (sec.)	
	h_1	h_3
Total sum	4743.27	2093.89
Mean	53.30	23.53
Standard dev.	147.89	68.80
Minimum	0.05	0.05
Maximum	1072.94	597.15

It seems likely, however, that gains in efficiency are not uniform for all problems, and that they should vary according to the difficulty of each particular instance. One possible difficulty measure can be the number of iterations needed by h_1 (the reference heuristic) to solve each problem [4]. Tables 4 and 5 show efficiency gains for three different problem subsets. In the first subset, which includes problems that needed up to 10000 iterations with h_1, the use of h_3 saved 30.3% of the search effort. In the second subset, between 10000 and 30000 iterations, h_3 saved 42.31% of the search effort. Finally, in the last subset, with the most complex problems,h_3 achieved savings up to 54.3%. Therefore, it seems likely that h_3 achieves better savings with more complex problems.

The cost of solutions in each subset shown in tables 4 and 5 show clearly that the solution's optimal cost is not necessarily a good predictor of problem complexity. The instance with higher cost (2462 m.) needs less than 2400 iterations in both cases.

5 PRACTICAL APPLICATIONS

The model and heuristics described in previous sections have been implemented and integrated in a road layout editor. This tool allows human designers to visualize topographical maps and draw preliminary road layouts using simple drag and click operations. The editor interactively warns the user of valid parameter ranges for each new

Table 4. Number of iterations according to problem complexity using h_1. The number of problems in each group is shown between parenthesis.

	[0, 10000] it. (76)		[10001, 30000] it. (9)	
	h_1	h_3	h_1	h_3
Total num it.	106622.00	74358.00	163924.00	94578.00
Mean	1402.92	978.39	18213.78	10508.67
Standard dev.(σ)	2050.56	1529.94	5764.15	4748.47
Minimum	11.00	11.00	11237.00	5208.00
Maximum	9630.00	6800.00	26730.00	19431.00
Cost ($\bar{c}$, σ)	1553.49 406.89		2160.10 56.18	
Cost (min, max)	948.53 2462.74		2089.95 2272.79	

Table 5. Number of iterations according to problem complexity using h_1. The number of problems in each group is shown between parenthesis.(II)

	[30001, 60000] it (4)	
	h_1	h_3
Total num it.	182486.00	83621.00
Mean	45711.50	20905.25
Standard dev.(σ)	11749.87	20553.10
Minimum	32990.00	6593.00
Maximum	58462.00	56370.00
Cost ($\bar{c}$, σ)	2271.02 144.96	
Cost (min, max)	2024.26 2397.06	

section, and highlights those sections that do not comply with road regulations. Road layouts can be combined into different road alternatives, evaluated according to different cost measures, and saved in exchange drawing formats used by other commercial CAD and elevation profile optimization software.

The editor includes an *automatic layout generator* that uses the A* search algorithm and the state space previously discussed. While the model and heuristics are not powerful enough yet to generate large road sections in an interactive fashion, they suffice to quickly solve smaller problems. These include joining unconnected layout sections with valid sections or posing detailed alternatives, tasks that otherwise would require certain effort from the human designer.

For example, figure 9 shows a topographical map where the user has drawn two unconnected road layouts. The automatic generator eases the task of joining both layouts providing the solution shown in figure 10.

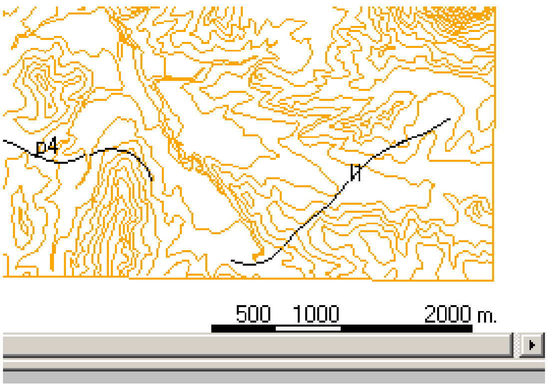

Figure 9. Topographical map with two unconnected road layouts.

Practical experience with the editor, and experimental tests like those presented in the previous section, show that layouts up to $2-3km$ long can be routinely generated with fast (interactive) response times. It is fair to say that much longer layouts can also be easily generated depending on the topography given enough time.

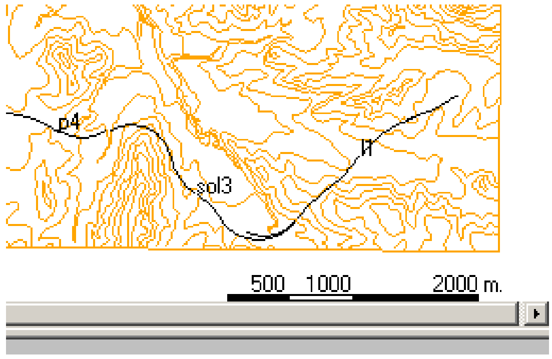

Figure 10. Unconnected layouts are connected by a new automatically generated layout.

6 CONCLUSIONS

Preliminary road layout is an important subtask in every road design process. A practical state space has been defined for this problem. The model considers both plan and elevation features. Generated layouts approximate with desired precision real layouts that comply with current road regulations while adapting to the terrain under consideration.

Simple heuristics have been tested to generate shortest road layouts using the A* algorithm. The experiments suggest that taking into account the goal's azimut can greatly improve efficiency when combined with simple distance estimations. This heuristic also seems to behave better when the problem gets harder.

The ideas described in this paper have been implemented and tested into an interactive road layout editor.

ACKNOWLEDGEMENTS

The work described in this paper was partially supported by European Union and Spanish CICYT funds, grant FEDER 1FD1997-1922.

REFERENCES

[1] Hart P.E., Nilsson N.J., Raphael, B. (1968): A formal basis for the heuristic determination of minimum cost paths *IEEE Transactions on System Science and Cybernetics SSC-4* pp. 100-107.

[2] Schreiber G., Akkermans A., de Hoog R., Shadbolt N, Van de Velde W., Wielinga B. (2000): *Knowledge Engineering and Management* The MIT Press.

[3] Pérez de la Cruz J.L., Conejo-Muoz R., Morales-Bueno R. and Puy-Huarte J. (1995): Highway design by constraint specification. *Artificial Intelligence in Engineering 9* pp. 127-139.

[4] Culberson J.C., Schaeffer J. (1996): Searching with pattern databases. *Advances in Artificial Intelligence, Gordon McCalla (Ed.)* Springer.