AN EFFICIENT ALGORITHM FOR RECONSTRUCTING ANISOTROPIC SPREAD COST SURFACES AFTER MINIMAL CHANGE TO UNIT COST STRUCTURES

Jose Valdez
Doctoral Candidate
Geomatics Program
Department of Forest Sciences
Colorado State University

Denis J. Dean
Associate Professor
Geomatics Program
Department of Forest Sciences
Colorado State University

ABSTRACT

Anisotropic cost spreading operations are frequently used to find optimal routes across a landscape. These optimal routes often represent minimum cost paths for new roads, trails, or utility lines, but they can also be used to represent the paths followed by a growing wildfire, the routes followed by wildlife moving through an area, and so on. Recently, researchers and other GIS users have developed sophisticated analysis procedures that make use of repeated anisotropic cost spreading operations. Frequently, these multiple cost spreading operations only differ from one another by relatively minor changes to the unit cost maps that comprise the majority of the inputs used by each anisotropic cost spreading operation.

Anisotropic cost spreading procedures are highly computer intensive and when applied to large databases, can require a considerable amount of time to produce solutions. However, if a particular cost spreading problem is only a minor variation of a previously solved problem, it stands to reason that a solution to the new problem can be derived relatively easily by modifying the existing solution to the old problem. The purpose of this study is to develop a cost spreading solution algorithm that can be applied to problems that are slight modifications of previously solved problems, and test and evaluate this new procedure relative to standard anisotropic cost spreading procedures under a variety of circumstances.

INTRODUCTION

Anisotropic cost spreading operations are used to find minimum cost paths across a landscape. Paths defined using this technique can be used to represent wildlife migration routes, least cost road construction paths to connect two or more points of interest, wildfire spread paths, etc. (Dean, 1997; Heimiller and Dean, 1998).

The idea of building a spread cost surface as part of a process intended to define least-cost paths is not new; Warntz (1965) introduced the relevant concepts 35 years ago. However, the computer hardware and software available in the mid 1960s precluded practical implementation of cost spreading, so Warntz's ideas remained largely theoretic for more than three decades. Fortunately, the speed and capability of the current generation of computers makes it possible to develop, and in at least some cases implement, spread cost concepts. The principles and concepts of practical anisotropic cost spreading are described by Huriot, *et al.* (1989) and Smith (1989).

Some recent applications of anisotropic cost spreading in natural resources management have been highly repetitive in nature. These applications require hundreds or thousands of cost spreading analyses to be conducted during the course of solving larger models designed to find optimal locations for wildfire mitigation activities, impacts of road developments on wildlife migration patterns, and so on (Pool and Dean, 1996; Heimiller and Dean, 1998). Unfortunately, current cost spreading algorithms are highly computer intensive and time consuming, and fall into the NP-complete class of computer algorithms (e.g., the amount of time needed to complete a spread cost analysis increases exponentially as the size of the data sets used in the analysis increases). As a result, repeated cost spreading operations involving large data sets quickly become prohibitively time consuming.

However, in certain situations it seems possible to efficiently solve multiple cost spreading operations. In particular, applications such as the one reported by Heimiller and Dean (1998) require repeated construction of anisotropic spread cost surfaces that are only minor variations of an initial surface. It seems plausible to hypothesize that updating an existing spread cost surface to create a new surface that differs only slightly from the original should require only a fraction of the time needed to build the original spread cost surface from scratch.

The purpose of this study was to develop and test a cost spreading solution algorithm that can be used to construct new anisotropic cost spreading surfaces that are slight modifications of previously built surfaces. The algorithm was tested and evaluated by comparing its results to those produced by standard anisotropic cost spreading procedures.

GLOSSARY

For the sake of clarity, before describing the new algorithm developed in this study, certain terms and background material will be presented. A *unit cost map* is a raster spatial data layer where each raster cell contains a value representing the cost of traversing (moving) from the center of the cell in question to the center of a particular adjacent cell. Thus, a cell in an “up and left” unit cost map containing the value *X* implies that it will cost *X* units to move from the center of the current cell to the center of the cell above and to the left of the current cell. Note that the values in a unit cost map can represent any type of costs, whether they be economic, ecological, temporal, etc. The anisotropic cost spreading algorithm used in this study requires eight unit cost maps as inputs, one representing the cost of movement in each of the eight possible directions of movement from one cell center to an adjacent cell center (Figure 1).

A *source map* identifies a source cell or cells that will be used as the starting point(s) for the anisotropic cost spreading analysis. A source map and the eight unit costs maps mentioned previously represent the nine inputs required by the anisotropic cost spreading operation.

A *spread cost map* is the first of two outputs produced by the anisotropic cost spreading operation. The spread cost map records the total cost of traversing from each cell in the map back to the least-costly-to-reach source cell. Thus, a cell in a spread cost map containing the value *X* indicates that the total cost of traveling from the cell in question back to the least-costly-to-access source cell is *X* units.

Finally, a *backlink map* is the second (and last) output of the anisotropic cost spreading algorithm. Each cell in a backlink map contains a code value that identifies which of the cell’s neighbors is on the minimum cost path running from the cell back to the least-costly-to-access source cell. By repeatedly tracing backward from cell to neighboring cell through the backlink map, it is possible to find the minimum cost route from any given cell back to the least-costly-to-access source cell.

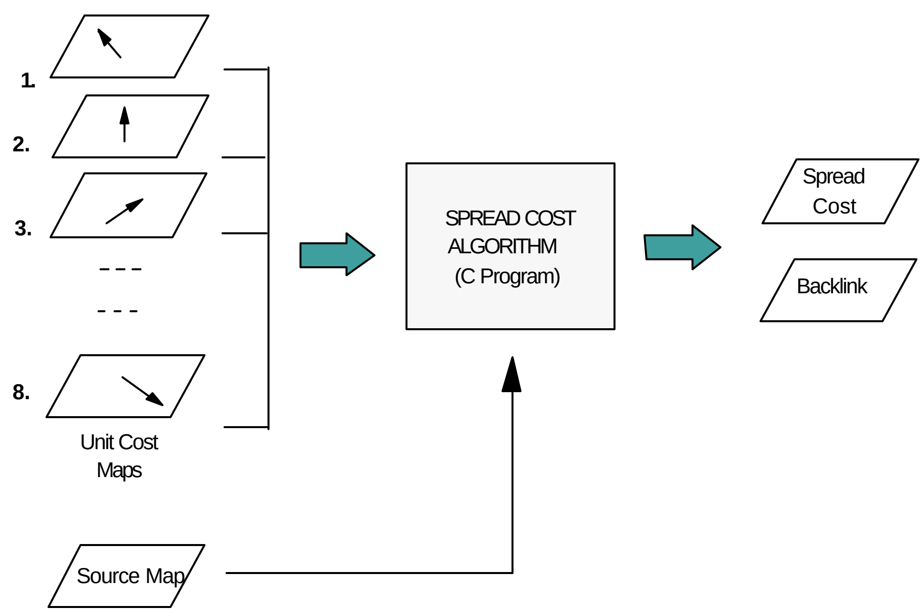

Figure 1. Anisotropic cost spreading flowchart process as defined by Dean (1996).

STANDARD ANISOTROPIC COST SPREADING

A variety of algorithms have been used to implement anisotropic cost spreading; the only real difference between these algorithms is how they deal with the direction-specific nature of anisotropic unit costs. Some algorithms use a single unit cost surface which gives each raster cell a single base unit cost. This base cost is then modified via some user-specified mathematical function to reflect cost differences due to direction (e.g., unit cost = base cost ? azimuth) (Eastman, 1995; Arc/Info, 1992). An alternative method for handling unit costs has been presented by Dean (1996). As described previously, this method uses multiple direction-specific unit cost maps to reflect costs of movement in various directions. The first technique has the advantage of being able to consider movement in an infinite number of directions, but has the disadvantage of forcing the same cost modification function on all cells throughout the entire map. In contrast, Dean (1996)'s technique can only consider costs in a finite number of directions, but allows the cost modification function to vary infinitely across the map. Dean (1996)'s method was used as the starting point for this study.

Regardless of how the directionality of unit costs is handled, all anisotropic cost spreading algorithms build their outputs by iteratively searching for least-costly routes from cells whose spread cost values are known to neighboring cells whose spread cost values are unknown (Dean, 1996; 1997). In the first iteration of the algorithm, all source cells are assigned a spread cost of zero, and source cells in the backlink map are assigned a code value indicating that they mark the ending point of any minimum cost paths originating at non-source cells. The second and all subsequent iterations evaluate all cells that were either (i) assigned spread cost values in any previous iteration, or (ii) adjacent to cells that were assigned spread costs value in previous iterations. For any given cell ?, this evaluation involves testing each cell that both neighbors ? and already has a spread cost computed in a previous iteration. Call such a neighbor ?. The cost of traveling from the center of ? to the center of ? can be obtained from the unit cost maps, and the total cost of reaching ? from the least-costly-to-access spread source cell can be obtained from the growing spread cost map. Thus, the total cost of traveling from the least-costly-to-access spread source to cell ? through cell ? can be computed by simply adding ?'s spread cost to unit cost values extracted from the input unit cost surfaces. The anisotropic spread cost algorithm functions by carrying out evaluations of this sort for all possible ?'s for each cell ?, and then recording the lowest value discovered for each cell ? as the new spread cost for ? (and recording the direction of movement from ? to ? in the backlink map for cell ?). This process stops only when an iteration occurs where no cells are assigned new spread cost values. Note that since this process continuously reevaluates spread costs computed in previous iterations, initial suboptimal solutions for individual cells are corrected in later iterations. This ensures that the final solution produced by this algorithm is truly optimal.

A MODIFIED SPREAD COST ALGORITHM FOR USE IN SPECIAL CASES

Applications such as the one proposed by Heimiller and Dean (1998) require repeated anisotropic cost spreading operations. In the case of Heimiller and Dean (1998), spread cost problems are solved iteratively. In general, the only difference between the spread cost problem encountered in iteration $n+1$ and the problem solved in iteration n is a change in the unit costs in one or a tiny handful of raster cells. Additionally, the unit cost changes are unidirectional: All unit cost changes involve replacing lower unit cost from iteration n with higher unit cost in iteration $n+1$.

A modified spread cost algorithm was developed to solve the type of spread cost problems encountered by Heimiller and Dean (1998). The modified algorithm is based on the knowledge that the only raster cells whose spread cost and backlink values can change from iteration n to iteration $n+1$ will be those cells whose backlink paths pass through the cells whose unit costs changed between iterations n and $n+1$. This is a result of the fact that in Heimiller and Dean (1998)'s model, unit costs can only increase from one iteration to the next. Consider a cell from iteration $n+1$. If the minimum cost path from this cell to a spread source (i.e., the path recorded in the backlink map) does not pass through a cell whose unit cost value increased, the cell will not be impacted by the unit cost increase. Conversely, a cell from iteration $n+1$ whose backlink path does pass through a cell whose unit costs have increased will likely suffer increased spread costs due to the unit cost increase. These increased spread costs may also cause the minimum cost path from the cell back to a spread source to shift, which will alter the cell's backlink characteristics. Note that it is possible that a cell whose backlink path passes through a cell whose unit cost have increased may suffer no increase in spread cost if an alternative route to a spread source is available at the same total cost as the original route. In this case, a cell's spread cost value will not change, but its backlink characteristics will.

The modified spread cost algorithm is shown in Figure 2. As shown in this figure, the algorithm's principles are simple, however, its implementation as a computer program can be tedious if the backlink map is not properly built. Fortunately, Dean's (1996) algorithm outputs a well structured backlink map, so implementing the new algorithm was fairly straightforward.

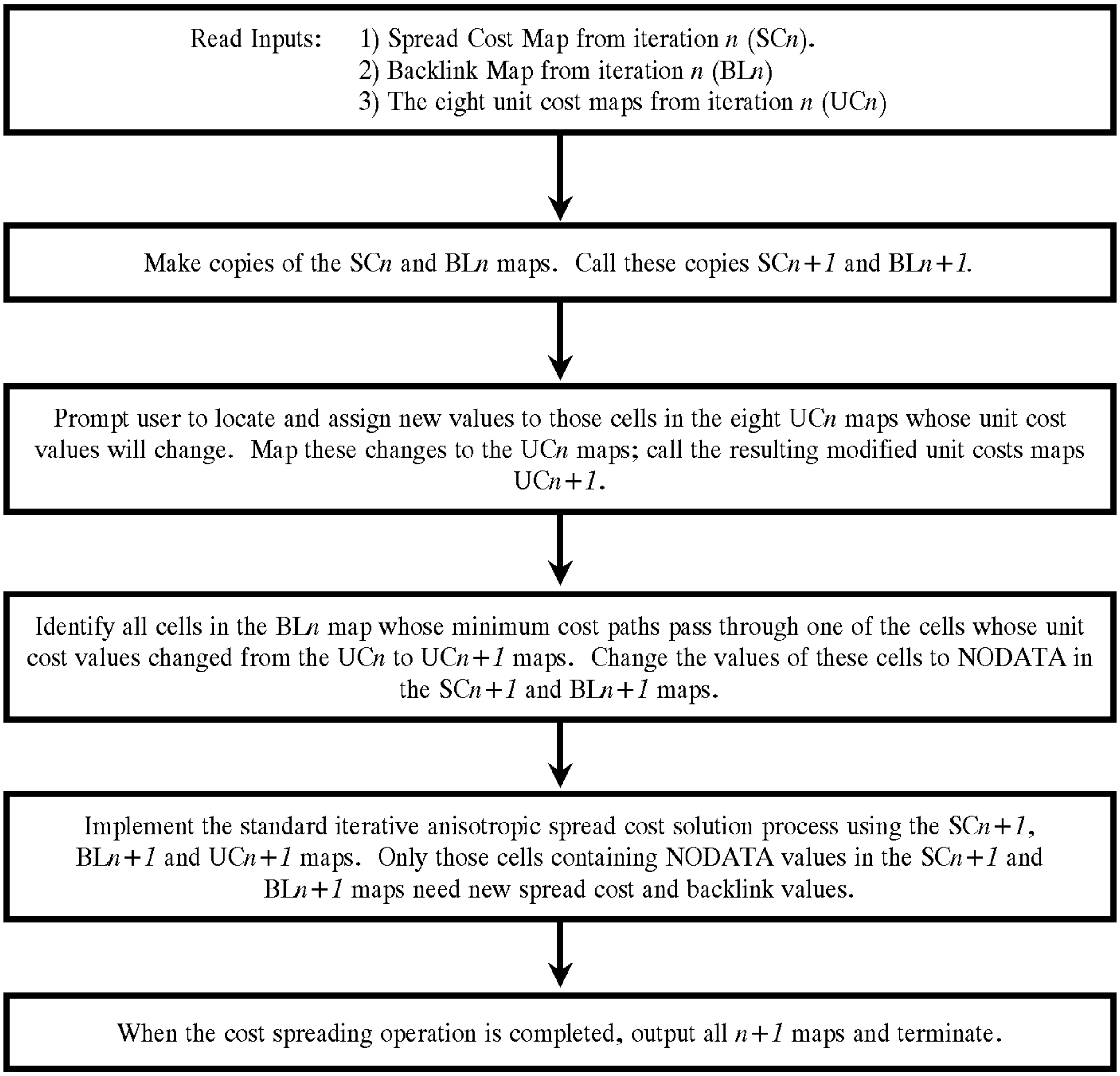

Figure 2. Flow chart showing the main steps to reconstruct an anisotropic spread cost map.

Algorithm implementation and evaluation

Both the standard anisotropic cost spreading operation as described by Dean (1997) and the modified spread cost algorithm just described were implemented using the C programming language. Unit cost maps were derived through an ARC/INFO AML (Arc Macro Language) program by computing, in a cell-by-cell basis, wildfire spread rates as defined by Rothermel (1972). Actual forest fuel inventory and terrain data obtained from the U.S. Forest Service, the U.S. Park Service and the U.S. Geological Survey were used for this purpose. These unit cost maps were of a variety of sizes:

100 cells x 100 cells, 150x150, 200x200, 250x250, 300x300, and 450x300. In addition to these unit cost maps, spread source maps of the same dimensions were created in ARC/INFO's GRID module by arbitrarily defining a single source point within each source map.

Once the unit cost and source maps were built, they were used as inputs in the standard cost spreading algorithm. This produced what we termed the *original spread cost* and *original backlink* maps. Next, a single cell was selected from the unit cost maps; the cost in this cell was arbitrarily changed to twice its original value. This resulted in a new set of unit cost maps that was then used as inputs into both the standard anisotropic spread cost system (along with the original spread source map) and the modified system (along with both the original spread source, original backlink and original spread cost maps). The amount of time needed by both of these algorithms to produce final results was recorded. This process was repeated to obtain multiple examples of various unit cost change scenarios.

Table 1 reports the number of repetitions evaluated, stratified by a number of criteria. Rows in the table correspond to the six sizes of raster maps evaluated. The second column of the table records the number of repetitions used in evaluating the traditional spread cost algorithm as defined by Dean (1997). Columns three through twelve record the number of repetitions used to evaluate the modified algorithm. These repetitions are broken down into ten categories based on the percentage of the total number of raster cells available that will be effected by the unit cost change. Our goal was to have ten repetitions for each entry in this table, but some table entries represented extremely unusual situations. For example, it was highly unusual to find a situation where changing the unit cost in one cell caused changes in 80% or more of the cells in the spread cost and/or backlink maps. As a result, some table entries show that less than ten repetitions were available for analysis.

Table No. 1. Number of evaluations (repetitions) carried out to obtain average computer processing time required to build from scratch and to rebuild anisotropic spread cost surfaces.

		Percentage of total number of cells requiring re-calculation									
Grid Size	**From Scratch**	**0-10**	**10–20**	**20-30**	**30–40**	**40-50**	**50-60**	**60–70**	**70-80**	**80-90**	**90–100**
100x100	10	10	10	4	2	10	10	1	2	10	1
150x150	10	10	10	10	1	9	10	2	0	10	10
200x200	10	10	10	10	10	10	1	10	6	0	4
250x250	10	10	10	10	10	4	10	4	10	0	2
300x300	10	10	10	10	10	0	10	0	10	10	10
450x300	10	10	10	10	10	10	10	10	10	0	1

RESULTS AND DISCUSSION

Table 2 reports the results of the execution time comparison. Clearly, the time gains produced by the new algorithm are most dramatic when only a small percentage of the total number of cells require recomputing, but measurable gains can still be seen when up to 70-80% of the map requires recomputing.

Table 3 shows the results of an analyses where every cell in a series of unit cost maps was evaluated. This evaluation involved determining how many spread cost cells would be effected by changing the unit cost of the single cell being evaluated. Thus, the 97.60% value in the upper left entry of Table 3 indicates that for 97.60% of the cells in the map, changing a single cell's unit costs results in changes to between 0 and 10% of the cells in a spread cost map.

Table 3 indicates that in over 99% of the cases investigated, changing the unit costs in a single cell resulted in 20% or less of the cells requiring recomputation. Table 2 indicates that in the 0 to 20% range of required recalculation, the new algorithm produced over 94% reductions in the amount of time needed to produce a final solution. Taken together, these results indicate that in the vast majority of cases, the new algorithm produces enormous time savings.

Table No. 2. Computer time (seconds) required to build from scratch and to rebuild a spread cost map from an existing one. Time needed to identify cells that will be changed is included in these figures.

		Percentage of total number of cells requiring re-calculation										
Grid Size	**From Scratch**	**0 - 10**	**10 - 20**	**20 - 30**	**30 – 40**	**40 - 50**	**50 - 60**	**60 - 70**	**70 – 80**	**80 - 90**	**90 - 100**	**Row Ave. sec. (%)**
100x100	3.8	0.0[‡]	0.1	0.75	1.0	1.2	1.5	2.0	2.5	3.3	3.8	1.6 (42[¶])
150x150	13.2	0.0	0.3	0.1	1.0	3.0	4.4	6.5	---	12.4	13.6	4.5 (34)
200x200	31.6	0.0	0.5	1.0	5.2	6.2	11.0	14.2	16.7	---	31.5	9.6 (30)
250x250	61.4	0.0	1.9	8.0	16.9	14.0	26.5	28.25	36.9	---	61.0	14.7 (24)
300x300	106.0	0.0	2.6	9.9	19.4	---[§]	39.5	---	84.3	102.0	106.0	45.4 (42)
450x300	186.0	0.0	16.1	10.0	36.9	78.5	87.4	115.8	119.4	---	186.0	72.2 (38)
Col. Ave.												(35)

[‡] Zero values does not imply that the operation is instantaneous, but that the elapsed time was so short that the computer could not accurately register it.

[§] No cases were present in this range for the grid analyzed.

[¶] Percentage of time reported in column two.

Table No. 3. Percentage and number of cells to be re-calculated when building a spread cost map from an existing map.

	Percentage of total number of cells requiring to be re-calculated (Number of cells)									
Grid Size	**0 – 10**	**10 - 20**	**20 - 30**	**30 – 40**	**40 - 50**	**50 - 60**	**60 - 70**	**70 - 80**	**80 - 90**	**90 - 100**
100x100	97.60	1.52	0.04	0.02	0.23	0.24	0.01	0.02	0.31	0.01
(10 000)	(9 760)	(152)	(4)	(2)	(23)	(24)	(1)	(2)	(31)	(1)
150x150	98.06	1.01	0.66	0.003	0.04	0.05	0.003	0.00	0.07	0.07
(22 500)	(22 064)	(229)	(150)	(1)	(9)	(12)	(2)	(0)	(17)	(16)
200x200	98.56	0.68	0.38	0.20	0.03	0.002	0.107	0.01	0.00	0.01
(40 000)	(39 426)	(273)	(152)	(80)	(14)	(1)	(43)	(6)	(0)	(4)
250x250	99.09	0.38	0.13	0.14	0.006	0.07	0.006	0.10	0.00	0.003
(62 500)	(61 935)	(239)	(84)	(92)	(4)	(45)	(4)	(94)	(0)	(2)
300x300	99.34	0.16	0.09	0.12	0.00	0.13	0.00	0.08	0.04	0.01
(90 000)	(89 407)	(150)	(85)	(111)	(0)	(123)	(0)	(74)	(38)	(12)
450x300	99.28	0.25	0.09	0.08	0.02	0.09	0.05	0.10	0.00	0.007
(135 000)	(134 031)	(347)	(131)	(116)	(33)	(122)	(71)	(148)	(0)	(1)

CONCLUSIONS

In the vast majority of cases, the modified spread cost algorithm developed here clearly produced results much more quickly than did the conventional algorithm. However, it must be noted that the modified algorithm is only applicable in a small number of cases where spread cost operations could being used. Future research will be aimed at (1) expanding the range of applicability of the modified algorithm, and (2) testing the algorithms performance under a larger variety of situations.

LITERATURE CITED

ARC/INFO. 1992. Grid command references. Environmental Systems Research Institute, Inc.

Dean, D. J. 1996. Finding minimum-cost routes for access roads to single and multiple harvest sites. *In:* Proceedings of the 1996 Southern Forestry and GIS Conference. (Greg J. Arthaud and William C. Hubbard, editors). The University of Georgia, Athens, GA. 416 p.

Dean, J. D. 1997. Finding optimal routes for networks of harvest site access roads using GIS-based techniques. Can. J. For. Res., 27(1): 11-22.

Dean, D. J. and D.B. Pool. 1996. Optimizing wildfire mitigation efforts using risk based GIS. *In:* Proceedings of the 1996 Southern Forestry and GIS Conference. (Greg J. Arthaud and William C. Hubbard, editors). The University of Georgia, Athens, GA. 416 p.

Eastman, J. R. 1995. Anisotropic cost analysis. *In:* Idrisi for Windows: User's guide version 1.0. Clark Labs for Cartographic Technology and Geographic Analysis. Clark University, Worcester, Mass.

Heimiller, D. and D. J. Dean. 1998. Optimizing placement of prescribed burns to maximize wildfire burn times to points of value in the wildland/urban interface zone. *In:* Proceedings of the 2nd Southern Forestry GIS Conference. (Whiffen, J. H. and W. C. Hubbard, editors). The University of Georgia, Athens, GA. 339 p.

Huriot, J. M., T. E. Smith, and J. F. Thisse. 1989. Minimum-cost distances in spatial analysis. Geographical Analysis, 21(2): 294-315.

Rothermel, R. C. 1972. A mathematical model for predicting fire spread in wildland fuels. US, For. Serv. Res. Pap. INT-115. Intermountain Forest and Range Experiment Station, Odgen, UT. 40 p.

Smith, T. E. 1989. Shortest-path distances: an axiomatic approach. Geographical Analysis, 21(1): 1-31.

Warntz, W. 1965. A note on surfaces and paths and applications to geographical problems, Ann Arbor: Michigan Inter-University Community of Mathematical Geographers.