ELSEVIER

Computers & Geosciences 30 (2004) 203–209

COMPUTERS & GEOSCIENCES

www.elsevier.com/locate/cageo

Least-cost paths in mountainous terrain☆

W.G. Rees*

Scott Polar Research Institute, University of Cambridge, Lensfield Road, Cambridge CB2 1ER, UK

Received 1 June 2003; received in revised form 6 October 2003; accepted 5 November 2003

Abstract

Footpaths in a mountainous area of Wales are modelled as least-cost paths between the start and end points. The cost function is defined on the basis of topography alone, and is defined in such a way that the cost penalty for excessively steep slopes is an adjustable parameter of the model. Least-cost paths are calculated by applying Dijkstra's algorithm to a Digital Elevation Model. Comparison of these calculated least-cost paths with existing footpaths suggests that the latter do not usually follow the least-time route, but instead optimise the metabolic cost for human locomotion. The method developed here is proposed as a means of exploring possible routes for new footpaths in mountainous areas.

Keywords: Digital elevation model; Least-cost path; Dijkstra algorithm; Mountain paths; Physiology

1. Introduction

The research described in this paper began with a practical question: why do footpaths in mountainous areas take the routes that they do? It is evident that footpaths will have evolved for a variety of reasons, and with different origins. Some may have begun as pack-animal or vehicular (e.g. cart) tracks, or may make use of these for part of their length, some may have been developed by people finding the easiest route to the next village or to a fishing lake, others may have begun life as sheep tracks, without any initial human intervention, and some may have been developed specifically for recreational walking. Nevertheless, it is reasonable to suppose that if a path has continued to be used as a means of walking from A to B, it will represent in some sense an optimum route between those points. This implies that there is some concept of 'cost' that is minimised by the actual path.

The hypothesis to be tested here is that the cost function is quantifiable, and that it is controlled by the steepness of the terrain. Testing of this hypothesis therefore requires a digital elevation model (DEM), a suitable model cost function, and a procedure for calculating the least-cost route through the DEM between two specified points. Several procedures exist, but the best known is Dijkstra's algorithm. This has a very substantial by-product: In calculating the least-cost route from A to B, it will also generate the least-cost routes from all other points in the DEM to B. Thus it is particularly suited to an investigation of the least-cost status of footpaths in a situation where there are many routes, with different starting points, leading to the same destination. This situation typically arises where the destination is the summit of a mountain. Dijkstra's algorithm has been applied recently in a somewhat similar context by Carver and Fritz (1999, 2000) and by Fritz and Carver (2000), who used it to consider 'remoteness' from access points in a wilderness area.

The aim of the work described in this paper is thus to devise a cost function that is simple but that contains at least one adjustable parameter. Dijkstra's algorithm is then applied to find the least-cost routes from a variety of starting points to a mountain summit, and the parameter(s) of the cost function are varied until the least-cost route most closely resembles a real footpath.

☆http://www.iamg.org/cgeditor/index.htm.

*Corresponding author. Tel.: +44-1223-336540; fax: +44-1223-336549.

E-mail address: wgr2@cam.ac.uk (W.G. Rees).

0098-3004/$ - see front matter
doi:10.1016/j.cageo.2003.11.001

2. The Dijkstra algorithm

The minimum cost route between two locations in the DEM is calculated using Dijkstra's (1959) algorithm. In terms of the structure of the DEM, this algorithm can be described briefly as follows. The key to the procedure is a list of 'active' cells. Each cell in this list has two attributes: the cost of getting from this cell to the target cell, and a pointer indicating the cell to which one would move from this cell in tracing the least-cost route from it to the target. Both of these attributes are provisional and can be altered during the implementation of the algorithm. Once a cell ceases to be a member of the active list, its attributes become definite.

In detail, the logical procedure for applying Dijkstra's algorithm in this context is as follows:

1. Assign a definite cost of zero to the target cell.
2. Identify all the neighbouring cells to the target cell and place them in the list of 'active' cells. For each of these cells, calculate and assign the cost of reaching the target cell, and assign a pointer that points to the target cell.
3. Find the cell in the list that has the lowest cost—call this cell **C**, and call the cost k.
4. Identify the set **S** of all the neighbouring cells of **C**. For each cell **C**' in **S**, calculate the cost l of moving to **C**.
   - 4.1. If $\mathbf{C}'$ is not yet a member of the list, add it to the list with a cost $k+l$ and a pointer that points to **C**.
   - 4.2. If $\mathbf{C}'$ is already a member of the list, compare the value of $k+l$ with the provisional cost of this cell. If $k+l$ is greater than or equal to the provisional cost, do nothing. However, if $k+l$ is less than the provisional cost, change the attributes of the cell $\mathbf{C}'$ so that its cost is now $k+l$ and its pointer now points to the cell **C**. (This procedure is termed 'relaxation'.)
5. Change the attributes of the cell **C** from provisional to definite, and remove it from the list.
6. Repeat from (3) until the list is empty.

This procedure has the great virtue that it calculates the minimum cost route to the target cell from *every* cell in the DEM. It can be shown by a simple inductive argument that the solution is correct, in the sense that there is no lower-cost route to the target from any specified cell than the one indicated by the algorithm, provided that the cost of moving from one cell to another cannot be negative (e.g. Sedgewick, 2001). However, the solution may not be unique. For example, at step (3) the algorithm may encounter two cells with equally low costs. The least-cost route from a yet unassigned cell might pass through either of these, dependent on the order in which they are processed by the algorithm.

3. Choice of cost function

An essential aspect of the algorithm outlined above is the ability to calculate the cost of moving from one cell of the DEM to another. The choice of a suitable cost function is not obvious. Here, it is modelled on recent research on walking times in mountainous areas (Rees (2003), and see the addendum), which suggests that the horizontal component of the speed v maintained on a surface of slope m is adequately described by a quadratic function

$$\frac{1}{v} = a + bm + cm^2. \tag{1}$$

('Slope' is defined here, and throughout this paper, as $\mathrm{d}h/\mathrm{d}x$ where h is the height gained and x is the horizontal distance travelled.) Rees (2003) found suitable value of the coefficients for walking time to be $a = 0.75\,\mathrm{s\,m^{-1}}$, $b = 0.09\,\mathrm{s\,m^{-1}}$, $c = 14.6\,\mathrm{s\,m^{-1}}$, although his data suggested that the value of b was not critical and could be set to zero without significant loss of accuracy. (A rather more complicated formula, with 16 coefficients, is used by the Schweizer Wanderwege SAW (1999). However, over the important range $|m| < 0.35$ it does not differ radically from the form of Eq. (1)). Thus, it is proposed that the cost of moving between two points with a horizontal separation d and a vertical separation h can be modelled as

$$k = ad + c\frac{h^2}{d}, \tag{2}$$

where the coefficients a and c do not necessarily have the same values that are appropriate to Eq. (1) (because the 'cost' to be minimised is not necessarily equal to the walking time). The form of Eq. (2) is convenient in that it models the same cost for a given path whether it is ascended or descended; thus, the calculated least-cost route from a starting point to the summit of a mountain will also be the least-cost route for the descent. Eq. (2) is also simple to investigate since it has only one effective parameter, namely, the value of c/a. Multiplying a and c by the same constant will change the absolute values of the costs, but will not change the least-cost route.

The third advantage of the cost function defined in Eq. (2) is that, unlike functions in which the cost is a linear function of the vertical separation h (for example, functions based on the well-known 'Naismith's rule' (Naismith (1892) for walking time), it will favour a zigzag ascent or descent of sufficiently steep slopes in preference to a path of maximum gradient. This behaviour, which similar to that of real paths, can be seen with reference to Fig. 1.

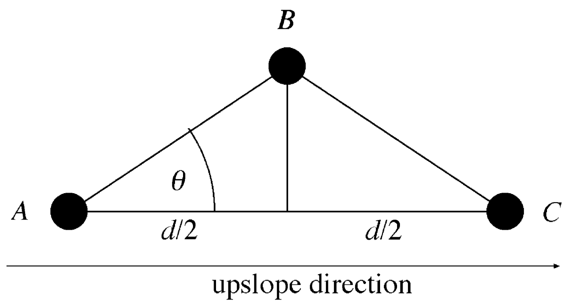

Fig. 1. Least-cost route from A to C may be direct or via the point B.

For a slope m in the direction from A to C, the cost of the direct route AC is

$$d(a + cm^2)$$

while the cost of the indirect route ABC is

$$d\left(\frac{a}{\cos\theta} + cm^2 \cos\theta\right).$$

It is thus favourable to make a zigzag in the direction θ provided that

$$m^2 > \frac{a}{c\cos\theta},$$

from which it follows that no zigzag can be favourable if the value of $|m|$ is less than the critical value

$$m_{crit} = \sqrt{\frac{a}{c}}. \quad (3)$$

These considerations apply strictly only to the situation where the topographic height is a continuous function of position, as in the real world. In the usual form of DEM, however, the height is specified only at discrete locations on a square grid of cells. A comparable analysis of this situation is considerably more complicated, but it can be noted that in the simplest cases, where the direction of the steepest slope is parallel to either of the edges of the DEM or makes an angle of 45° with them, the value of θ in the foregoing analysis can only take the values 0° or 45°. In these cases the critical value of $|m|$, above which a zigzag path is favoured, is $2^{1/4}m_{crit}$.

4. Implementation of the algorithm

Preliminary development of the algorithm was carried out using the *NIH Image* image processing software. This public-domain software (developed at the US National Institutes of Health and available on the Internet at http://rsb.info.nih.gov/nih-image/) has an interpreted macro-language, similar to Pascal, that allows fairly complex image processing tasks to be implemented, albeit relatively slowly, which fits it for algorithm development work. It also has the benefit of running on pre-OS X Apple Macintosh computers. A later implementation was made in C++. The author did not investigate the possibility of developing the algorithm within a standard GIS package, judging that programming the Dijkstra algorithm would be a more complex task.

The algorithm closely follows the description given in Section 2. It operates on a DEM consisting of N cells in width by M cells in height. A cell with coordinates (i_0, j_0) is defined to have up to eight neighbours, i.e. in the coordinate range defined by $|i - i_0| \leqslant 1$, $|j - j_0| \leqslant 1$, provided that this range is not truncated by the bounds of the DEM. (It is possible to define more than 8 neighbours, for example, by including information on the direction from which a path arrives at one of the neighbouring cells.)

Three more $N \times M$ arrays are defined: one to hold the cost value of each cell, one to hold the pointer to the next cell in the least-cost path towards the target, and one to hold a flag to indicate the status of the cell as unprocessed, active or processed (i.e. with definite values of cost and pointer). In addition, the algorithm maintains a list containing the i and j coordinates and provisional cost of every 'active' cell. The list is stored in order of provisional cost, with the 'cheapest' cell first. This avoids the need to search the list to find the lowest cost (step 3 of the algorithm), but requires the list to be re-sorted each time it is changed. In practice the list can become quite long (the number of entries reaches a maximum of the order of NM), so these list-sorting operations limit the speed of implementation of the algorithm. The running time of the algorithm is expected to be proportional to $(NM)^2\ln(NM)$. In the original implementation of the algorithm using *NIH Image*, this limited its practical applicability to DEMs of up to $O(10^4)$ cells, while the recoding into C++ meant that it could process of the order of 10^6 cells in around 30 min.

The implementation of the algorithm was tested, and its characteristics observed, by applying it to a synthetic DEM. This represented an area of 5 km × 5 km in cells of width 50 m. The topography consisted of a centrally placed cone with a slope m of 0.3. Heights were stored in this DEM as 8-bit integers with a resolution of 5 m. Fig. 1 shows the least-cost routes calculated from points distributed around the edge of this DEM to a target at the summit, for three different cost functions. These were defined by Eq. (2) with values of c/a of 10, 14 and 18, respectively. Eq. (3) shows that, for $m = 0.3$, a zigzag route is preferable to a direct (steepest ascent or descent) route for $c/a > 11.1$. Following the argument in Section 3, a zigzag path should be preferable to a direct route in the DEM, provided that the latter is either parallel to one of the edges of the DEM or makes an angle of 45° with them, for $c/a > 15.7$. These considerations dictated the choice of the three values of c/a used for this test.

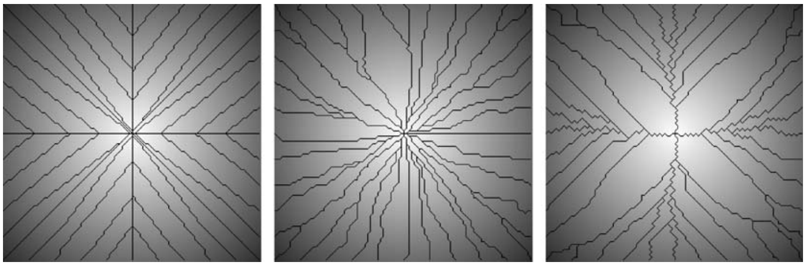

Fig. 2. Calculated least-cost paths to summit of a conical mountain with slope $m = 0.3$. Figures correspond to values of $c/a = 10$, 14 and 18, respectively. Paths have been superimposed on a greyscale representation of DEM.

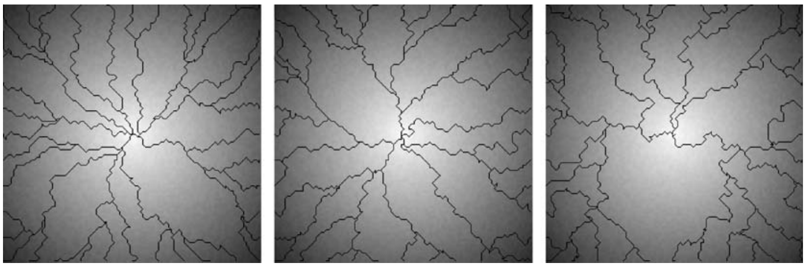

Fig. 3. As Fig. 2, but with random uncorrelated noise added to DEM.

Fig. 2 shows that, as expected, the direction of the calculated least-cost route is correct only when it is orthogonal or diagonal to the axes of the DEM. When $|m|$ is less than m_{crit} the general trend of the calculated routes is composed of orthogonal and diagonal segments. As the value of c/a is increased, the tendency of the paths to make zigzags increases as expected. The amplitude of the zigzags is in general as small as possible, i.e. of the order of 1 cell in the DEM, while in the case of a real path there is no constraint on this amplitude. The figure also illustrates the fact that, for $m > m_{crit}$, the calculated routes do not possess the same symmetry as the DEM. This can be attributed to the fact that, in this artificially symmetrical DEM, many pixels will have the same provisional cost, and the order in which they are chosen will break the symmetry.

Fig. 3 shows the results of the same procedure with random uncorrelated noise (amplitude 25 m) added to the DEM. This provides a somewhat more realistic simulation of real topography, and it is apparent that the undesirable (and incorrect) tendency of the algorithm to choose routes in directions related to the orientation of the DEM has been eliminated.

5. Simulation of real mountain paths

A DEM suitable for investigation of this algorithm was acquired from the Ordnance Survey of Great Britain (OSGB) through the EDINA Digimap service (http://edina.ac.uk/digimap/). The DEM covered an area of $9\,\mathrm{km} \times 12\,\mathrm{km}$, centred approximately on the summit of the mountain Snowdon (Welsh name: Yr Wyddfa) at 53° 04′N, 4° 04′W and 1085 m a.s.l., in Wales. This area is suitable, since there are many established walking paths to the summit, with mean slopes ranging from about 0.11 to 0.17 and standard deviations in the slope ranging from 0.08 to 0.23. The DEM has a cell size of $50\,\mathrm{m} \times 50\,\mathrm{m}$, and the data are supplied with a vertical precision of 1 m although the intrinsic accuracy is probably about 3 m. The DEM data were converted to an 8-bit representation in which the vertical interval was 5 m for simplicity of processing using the *NIH Image* software.

The summit of Snowdon was set as the target cell in the DEM, and the Dijkstra least-cost algorithm was run for different values of c/a as defined in Eq. (2). These values were 0, 2, 4, 6, 8, 10, 15 and 20. A value of zero corresponds to no cost penalty for a change of height, so that the true minimum cost route between two points is a straight line. However, as illustrated in Section 4, the nature of the DEM means that the algorithm is able to find this route only if it is orthogonal or diagonal to the axes of the DEM. The value of 20 was chosen to correspond approximately to the walking-time formula (Eq. (1)).

The least-cost routes were computed for paths with six different starting points, corresponding to well-known footpaths. The divergence between the computed least-cost route and the existing footpath was characterised by

Table 1
Values of the normalised divergence between calculated minimum cost paths and existing footpaths, as a function of the ratio c/a

		Route					
		A	B	C	D	E	F
	Distance (km)	3.9	4.1	5.8	4.4	4.7	5.3
c/a	(true straight line)	0.044	0.042	0.030	0.094	0.136	0.062
	0	0.047	0.099	0.078	0.192	0.201	0.028
	2	0.053	0.063	0.025	0.059	0.088	0.017
	4	0.022	0.068	0.053	0.059	0.093	0.082
	6	0.049	0.052	0.019	0.057	0.031	0.086
	8	0.072	0.091	0.034	0.065	0.026	0.017
	10	0.091	0.067	0.039	0.088	0.201	0.019
	15	0.045	0.205	0.079	0.169	0.144	0.025
	20	0.166	0.047	0.070	0.077	0.068	0.187

The table also shows the straight-line distance between the starting and ending points of the path, and the normalised divergence between the existing path and the straight-line route between these points.

calculating the area enclosed between the two routes, and then normalising this area by dividing it by the square of the distance between the starting point and the summit. In some cases there was more than one footpath from the starting point to the summit, in which case the footpath that gave the lowest value of the normalised divergence was selected. The results of this analysis are shown in Table 1, and the corresponding routes in Fig. 4. The calculated routes for $c/a = 0$ are not shown. As has been mentioned previously, the 'short-range' nature of the algorithm means that it is unable to generate the correct result (a straight line) in the case $c/a = 0$ unless the line is orthogonal or diagonal to the DEM axes. For this reason, a row representing the true straight line solution has been added to Table 1.

A (routes from Pen y Pass): The situation here is complicated by the presence of two lakes. Nevertheless, the results are fairly clear-cut. The calculated route with $c/a = 4$ is a remarkably close approximation to an existing and popular footpath, although all values of this parameter up to 6 are reasonably similar to existing footpaths. The least-time route is the least similar to any real routes, although interestingly, the route with $c/a = 15$, which penalises steep slopes almost as strongly as the walking-time formula, is reasonably similar to an existing path.

B (routes from Nant Gwynant): In this case, the route with the smallest divergence from an actual path is the true straight line, closely followed by the route calculated for $c/a = 20$ (shortest walking time) which mimics a real path reasonably closely. Values of c/a from 2 to 6, and 10, are also similar to real paths.

C (routes from Beddgelert): There is only one commonly used path from this point, and it is fairly direct. It is exceptionally well approximated by values of c/a of 2 and from 6 to 10.

D (routes from Rhyd Ddu): The usual path from this starting point, which is rather indirect, is reasonably well approximated by value of c/a of 2–8, although none of the calculated paths is particularly close to it.

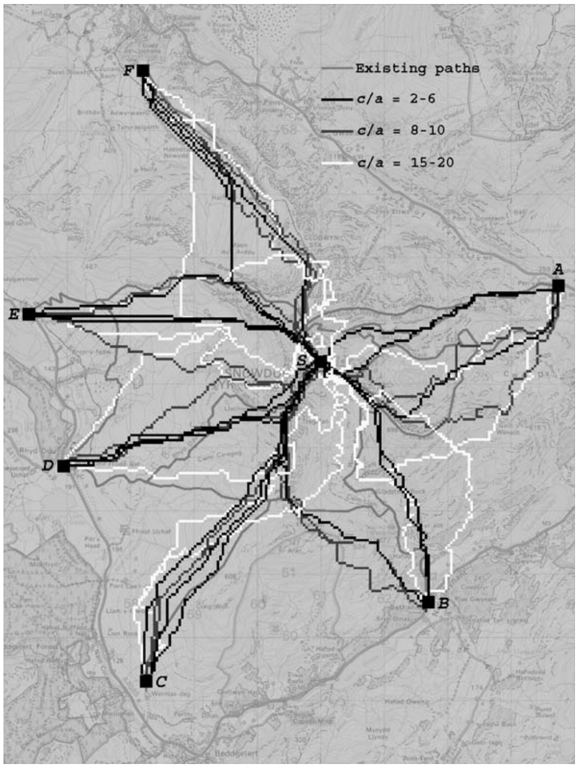

Fig. 4. Calculated least-cost routes to summit (S) of Snowdon from 6 starting points A to F. Coverage is 9 km × 12 km. Background map© Crown Copyright Ordnance Survey. An EDINA Digimap/JISC supplied service.

E (routes from Snowdon Ranger Youth Hostel): Again, there is only one commonly used path from this starting point. Unlike the path from Llanberis, it is rather indirect. It is very closely approximated by values of c/a of 6 and 8, and reasonably well by $c/a = 20$.

F (routes from Llanberis): There is only one commonly used path from this starting point, following a gently sloping spur of the mountain. The slope is sufficiently shallow that it is used by a rack-and-pinion mountain railway, and fairly straight. Consequently, it is not surprising that there is little variation in most of the calculated least-cost routes. The minimum walking time route is the least like the existing path.

These results show that most existing footpaths to the summit of Snowdon can be reasonably described as least-cost routes, although the value of c/a appropriate to the cost function varies over the full range (0–20) explored in the analysis. On the other hand, values in the range 4–8 produce plausible footpaths from every starting point, whether or not these exist as real paths. A value of $c/a = 20$, corresponding to the minimum walking time, sometimes produces realistic paths but sometimes produces paths that are very unconvincing.

6. Discussion

Analysis of the results from Snowdon suggest that most real footpaths look reasonably like least-cost paths with the cost function of Eq. (2) and values of c/a between 4 and 8. Such paths are clearly not optimal for walking time, so one may ask if there is any other criterion against which these paths could be judged to be optimal. Minetti (1995) extended the work of Margaria (1938, 1976) on the metabolic cost of walking on slopes of different gradients. He estimated the maximum slope that could be sustained for 5 h, as a function of altitude. At sea level this slope is about 0.4, declining to 0.25 at 3000 m. From Eq. (3), we observe that the range $4 \leqslant c/a \leqslant 8$ corresponds to a range of critical slopes of 0.35–0.5, which is clearly consistent with Minetti's results. This suggests that the cost function in the case of most paths corresponds to metabolic cost, which is plausible. This suggests an application of this research to the selection of walking routes in previously unexplored areas, or of new alternative paths in areas where it is desired to 'rest' existing paths in the interests of erosion control. Specifically, it is suggested that by designing such a path as a least-cost path with $c/a \approx 6$ (the value would need to be adjusted somewhat for high-altitude paths), it would be a comfortable and metabolically efficient path to walk on.

7. Conclusions and suggestions for further work

A simple application of Dijkstra's algorithm to footpaths in a mountainous area of Wales suggests that they can be approximately described as least-cost paths with a cost function given by Eq. (2). The value of the ratio c/a that defines the cost function varies from path to path. Occasionally this value is as high as 15–20, which is consistent with a minimum walking time route, but more commonly the value of c/a is in the range 4–8. This latter range is consistent with maximum metabolic efficiency of a path for human locomotion. It is thus suggested that new paths could be identified using this method with a value of $c/a \approx 6$.

The investigation described here takes no account of the varying difficulty of walking over different kinds of terrain. At the most extreme level, the algorithm as currently implemented assumes that it is possible to walk on water, but it also fails to apply cost penalties to, for example, marshy or boulder-strewn ground. These deficiencies could be met relatively straightforwardly by including a data layer containing a 'cost factor' by which the calculated cost of moving between two cells should be multiplied. The cost factor would, for example, be infinite for a significant water body.

The cost function of Eq. (2) does not distinguish between uphill and downhill slopes of the same magnitude, so that it will choose the same route for the ascent and the descent of a mountain. If this is regarded as an unrealistic restriction (it is not excessively so in the author's opinion), it could be avoided by replacing the cost function with a quadratic modelled in Eq. (1). Such a model would have two independent parameters, b/a and c/a, so it would be necessary to explore this two-dimensional parameter space in order to find the optimum combination or combinations of parameters for reproducing plausible routes.

The investigation described here is limited in scope (a single, comparatively small mountain massif in the UK), and should be extended to other areas. However, it would be easy to establish its likely applicability to other areas where there are existing paths by calculating the slope statistics of those paths and using them to estimate the most appropriate value of c/a from Eq. (3).

Acknowledgements

I am grateful to Adele Garner for a number of helpful discussions, and for producing the C++ version of the algorithm. I thank Dr. S Carver for his constructive comments on an earlier draft of the manuscript.

Appendix. A: a note on walking time

For over 100 years, estimates of the time taken to walk in hilly terrain have mostly been based on 'Naismith's Rule' (Naismith, 1892), which in its metricated form states that the time is 12 min per horizontal kilometre plus 10 s for each metre of ascent. Descents are ignored. In the terminology of Eq. (1), the rule can thus be expressed as

$$\frac{1}{v} = a' + b'm$$

if $m \geqslant 0$ and

$$\frac{1}{v} = a'$$

if $m < 0$. The coefficients in the standard Naismith rule are $a' = 0.72\,\mathrm{s\,m^{-1}}$ and $b' = 10\,\mathrm{s\,m^{-1}}$. Various refinements to the rule have been proposed, such as adjustments to the coefficients to take account of variables such as fatigue, fitness, ground conditions and so on, but most of these have ignored the common experience that steep downhill slopes cannot be covered as quickly as the corresponding distance on level terrain. Langmuir (1984) introduced a piecewise linear function to address this problem, but it is awkward to apply, and polynomial functions are generally more convenient.

Rees (2003) investigated the applicability of the simple quadratic formula given in Eq. (1), collecting data from ten separate walks in England and Wales totaling around 100 km in length. The data were collected using a GPS receiver in a manner similar to that described by Carver and Fritz (2000), then imported into a DEM to determine the variation of height and horizontal position as functions of time. The observed values of T/D (total walking time, excluding halts, divided by total horizontal distance) were modelled using Eq. (1) and also using Naismith's rule. The results showed that Eq. (1) provided a much better fit to the data than Naismith's rule, even if the coefficients of the latter were allowed to vary, and that the linear term b in Eq. (1) could be set to zero without any significant decrease in the goodness of fit. The coefficients $a = 0.75\,\mathrm{s\,m^{-1}}$, $b = 0.09\,\mathrm{s\,m^{-1}}$, $c = 14.6\,\mathrm{s\,m^{-1}}$ reported in Section 3 are 'personal' to the author and his style of walking, level of fitness and so on, but the results suggest that Eq. (1) is likely to be generally valid.

References

Carver, S., Fritz, S., 1999. Mapping remote areas using GIS. In: Usher, M.B. (Ed.), Landscape Character: Perspectives on Management and Change. Natural Heritage of Scotland Series 8 The Stationery Office, Edinburgh, pp. 213.

Carver, S., Fritz, S., 2000. Munro-bagging with a computer…? Naismith's rule and the long walk in. Scottish Mountaineering Club Journal 191, 317–322.

Dijkstra, E.W., 1959. A note on two problems in connection with graphs. Numerische Mathematik 1, 269–271.

Fritz, S., Carver, S., 2000. Modelling remoteness in roadless areas using GIS. Proceedings of the 4th International Conference and Workshop on Integrating Geographical Information Systems and Environmental Modelling. Banff, Canada. http://www.colorado.edu/research/cires/banff/pubpapers/157/ (accessed 03.10.03).

Langmuir, E., 1984. Mountaincraft and Leadership. The Scottish Sports Council, Edinburgh. p. 480.

Margaria, R., 1938. Sulla fisiologia e specialmente sul consumo energetico della marcia e della corsa a varia velocità ed inclinazione del terreno.. Atti Della Accademia Nazionale Dei Lincei Memorie 7, 299–368.

Margaria, R., 1976. Biomechanics of human locomotion. In: Biomechanics and Energetics of Muscular Exercise. Clarendon Press, Oxford, pp. 67–129.

Minetti, A.E., 1995. Optimum gradient of mountain paths. Journal of Applied Physiology 79, 1698–1703.

Naismith, W.W., 1892. Scottish Mountaineering Club Journal 2, 136.

Rees, W.G., 2003. Naismith's rule overhauled. Scottish Mountaineering Club Journal, submitted for publication.

Schweizer Wanderwege SAW, 1999. http://www.swisshiking.ch/wi/030400wimzb_index.htm (accessed 13.05.03).

Sedgewick, R., 2001. Algorithms in C, Part 5: Graph Algorithms, 3rd ed. Addison-Wesley, Boston. 512pp.