Optimal Path Calculation in the Countryside

A Raster-attribute-based Model Approach

Philipp Le[1], Gerald Eichler[2]

[1] Otto-von-Guericke-University Magdeburg
Faculty of Electrical Engineering and Information Technology
Universitätsplatz 2, 39106 Magdeburg, Germany
philipp.le@st.ovgu.de, dl6ple@darc.de
[2] Deutsche Telekom AG
Technology Innovation: Portfolio, Strategy and Data
Deutsche-Telekom-Allee 7, 64295 Darmstadt, Germany
gerald.eichler@telekom.de, dl1dsr@darc.de

Abstract. Using a landscape raster-based model for optimal route navigation in orienteering sports, velocity related attributes are mapped onto single squares. Both, accelerators and dampeners as well as barriers are taken into consideration. The optimization calculation between points is based on A* while the entire route is calculated by asymmetric traveller salesman problem solutions.

Keywords: Asymmetric Traveller Salesman Problem (ATSP), Orienteering Map, Digital Elevation Model (DEM), Amateur Radio Direction Finding (ARDF), Navigation, Route Optimization, Raster Graph Transformation.

1 Foot Orienteering and Amateur Radio Direction Finding

Orienteering is an example, where routing is not limited to predefines paths. Foot orienteering is considered the mother of all derivatives in orienteering sports. Primarily, the competitor aims to find the shortest path between two control points (CP) in terms of time consumption. This leads into the question: Where is the right place to leave a given foot path network into vegetation covered countryside?

A detailed map (Fig. 1) and compass assist the competitor. Except for restricted areas, any way through rural terrain is valid. Amateur Radio Direction Finding (ARDF), where CPs are hidden transmitters called foxes, extends the competition goal by two dimensions, using a portable direction selective radio receiver:

1. **Fox location:** The location of CPs within the countryside is not known from the very beginning. However, after five minute running, a competitor has a rough idea about their locations by means of his radio.
2. **Fox order:** The order in which competitors search for and discover the foxes is entirely at their discretion except for the finish beacon, which shall be registered as the last one of the transmitters.

This contribution focuses on (2) and therefore, it is better applicable to Foxoring. The artificial term is constructed out of “FOXhunting” and “OrienteeRING”. To discover the optimal path, beside the nature of countryside, there is also a set of secondary criteria like personal maximum speed and continuation or abilities in navigation in complex environments. Furthermore, weather and time of the year influence the competition terrain. Differences between the three sports are listed in Table . While Orienteering is essentially targeting navigation and object discovery capabilities, the key of Foxoring is the discovery of the optimal order of control points. ARDF requires even more strategic decisions during the entire competition, due to limited air time of each fox with one out of five minutes.

Table 1. Comparison of Orienteering, Foxoring and classical ARDF.

	Orienteering	Foxoring	ARDF
CP sequence	Fix	Flexible	Flexible
CP position	Exactly known	Roughly known	Not known
CP indication	Kite 30 by 30 cm	Log station only	Kite 30 by 30 cm
Radio transmitter	n/a	Constantly	1 min interval
Transmission pwr.	n/a	Low power (mW)	High power (W)
Number of CPs	10 to 30	7 to 20 + beacon	5 + beacon

For all three disciplines, the sportsman is given an orienteering map (Fig. 1). In the approach of this contribution, the map will be overlaid by a regular raster (Section 2). Incorporated information is based on public geo-data. Section 3 will elaborate on maps and a Digital Elevation Model (DEM) as data sources.

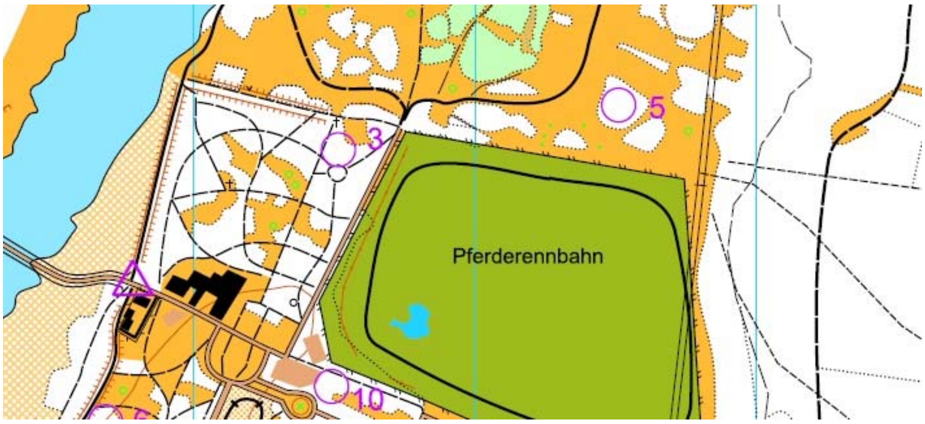


Fig. 1. Section of orienteering map of “Magdeburg Herrenkrug”, Germany [1].

The model approach (Section 4) will overlay a raster with equal regular shapes of a defined granularity [2]. To apply a classical Traveller Salesman Problem (TSP) solution, after A* reduction, the final routing is based on vectored graphs (Section 5).

2 Regular Raster Shapes

Natively, vectored paths are perfect for route optimization problems. However, orienteering offers the entire landscape as competition area. Therefore, a “navigation on areas” approach [2] will be targeted. As ARDF competitions mostly not exceed a race length of 12 km bee line, the relevant part of the map is typically not larger than 3 km by 3 km. At a scale of 1:10.000 this will fit an A3 sheet of paper.

2.1 Options for Regular Shape Selection

The types of convex polygons with equal edge length which are sufficient to raster the projection of a surface into regular shapes is limited to triangle, square and hexagon (Fig. 2). For reasons of given data sources, the rectangle will be evaluated as an additional candidate right here.

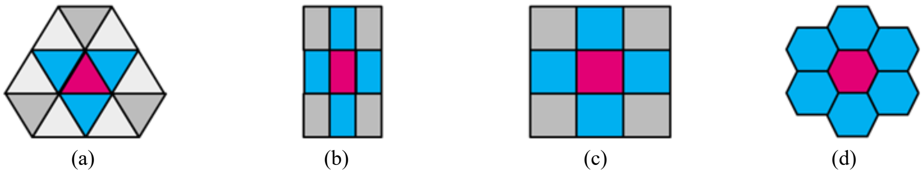


Fig. 2. Regular shape raster decomposition (a) triangle, (b) rectangle, (c) square, (d) hexagon.

Having path mapping in mind, the neighbourhood relations of raster fields are important. Starting from a single shape, two types of direct neighbours can be identified:

1. Line neighbour (L): Two adjacent shapes have exactly two corners in common.
2. Corner neighbour (C): Two adjacent shapes have exactly one corner in common.

By design, each raster field has the same number of neighbours. Neighbours with the distance one, sharing either one or two corners, should be called first (1^{st}) order neighbours. In other words, L neighbours share one edge.

Table 2. Neighbourhood relationship of shapes of 1^{st} order.

	Triangle	Rectangle	Square	Hexagon
Number of L-neighbours	3	4	4	6
Number of C-neighbours	9	4	4	0
Different L-distances	1	2	1	1
Different C-distances	2	1	1	n/a
Different L-directions	3	4	4	6
Different C-directions	9	4	4	6

Table 2 summarizes the absolute number and the number of different distances and directions of first order neighbours. As the comparison shows, triangles have a rather complex neighbourhood, which can be simplified by using hexagons as a super-shape of triangles, i.e. each hexagon can be split into six triangles. In small map scale, a rectangle would perfectly fit to geo-coordinates, as 1 arc sec in Central Europe equals about a size of 20 m by 30 m. However, a square better represents the symmetric resolution for the given problem in latitude and longitude. Therefore, **squares** are taken for this raster approach model.

2.2 Multi-order Neighbourhood of Squares

More general the order of neighbourhood is defined as the number of circle-free and outwards first order steps between two selected raster fields. Outwards indicates an increasing distance by each step. Distances are always measured between the centres of squares; see white arrows in Fig. 3.

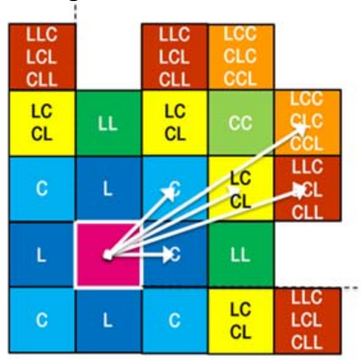


Fig. 3. Multi-order neighbourhood specification of squares.

As distance reference the edge length of a single square is to be considered as *a*.

Table 3. Neighbourhood relationship of shapes of 1st order.

Neighbour type	Order	Neighbours	Distance	New directions	
L	1	4	a	4	8
C	1	4	$\sqrt{2}*a$	4	
LL	*2*	*4*	$\sqrt{4}*a = 2*a$	*0*	1*8
LC/CL	2	8	$\sqrt{5}*a$	8	
CC	*2*	*4*	$\sqrt{8}*a = \sqrt{2}*a$	*0*	
LLC/LCL/CLL	3	8	$\sqrt{10}*a$	8	2*8
LCC/CLC/CCL	3	8	$\sqrt{13}*a$	8	

Two findings characterize the multi-order model for squares:

1. Line (L) and corner (C) direction extension towards higher order (O) is always commutative regarding distance and direction.
2. Only 8*(O-1) new directions in higher order are potential of interest, as others can be modelled iteratively by lower order without any routing gain.

To simplify the methodology for the upcoming sections, only the **first order relationship** will be applied within this contribution.

3 Data Sources for Attribution

Orienteering typically takes place in difficult rural areas. There are two factors that heavily influence the speed of the competitor: height differences and the surface coverage e.g., plants or swamp. This information is given to the competitor by a detailed "orienteering map". For the envisaged approach the information needs to be transferred into the raster model (Section 4).

3.1 SRTM Height Data

Although height information is given by contour lines within the orienteering map, there is a better source to meet the raster approach. The Shuttle Radar Topography Mission (SRTM) has collected earth surface height data by Spacebone Imaging Radar measurements (SIR-C). The data is public available all over the world between 60 degree latitude south and 60 degree latitude north. There are data sets of two resolutions: 1 arc sec and 3 arc sec. The latter is generated by calculating the average of nine SRTMGL1 values[1]. The access to the SRTMGL1 resolution requires a personal registration. The other sets of Table 4 are subject to instant download.

The sources claim an accuracy of about 6 (DLR) to 12 meters (NASA) [3]. However, height differences of neighbour fields are more important, which imply better results. The data is provided as file of 16-bit-integer numbers in big endian (BE) notation without a header as a bulk for one degree latitude by one degree longitude. Positive values represent heights in meter above mean sea level. Values are arranged from West to East and then North to South. The value "-32768" is reserved for not known height. SRTM Void Filled altitude data are the result of additional processing to address areas of missing data or voids in the SRTM Non-Void Filled set.

Table 4. Available SRTM data sets [4].

File set	Resolution	Number of values	DTED accuracy	Rectangle in Central Europe
SRTMGL1	1 arc sec	3601*3601	Level 2	20 m * 30 m
SRTMGL3	3 arc sec	1201*1201	Level 1	60 m * 90 m
SRTMGL30	30 arc sec	121*121	Level 0	600 m * 900 m

There is a unique file naming convention. The tiles are distributed as zip files, containing HGT files labelled with the coordinate of the southwest cell. As an example, looking for the data covering Magdeburg, the file *N52E011.hgt* is required.

[1] Official data source URL: https://lta.cr.usgs.gov/SRTM1Arc

3.2 SRTM Mapping from Rectangle onto Squares

A field of one arc sec by one arc sec equals roughly to a rectangle of 20 m by 30 m in Northern and Central Europe. To symmetrize the map raster, it is proposed to make three squares out of two vertical rectangles. An absolute metric accuracy of the altitude is not required to solve the given problem sufficiently.

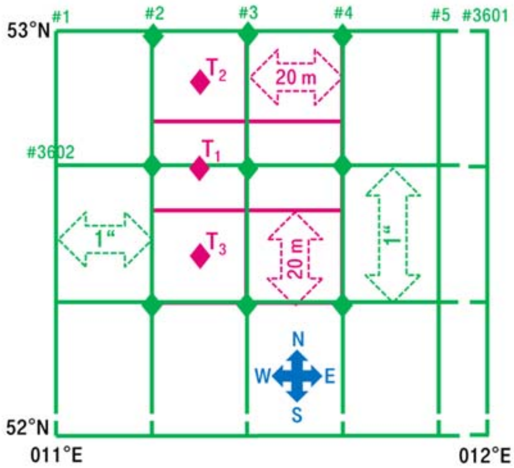


Fig. 4. Sequence of height values in the sample file *N52E011.hgt* (green). Three types T of rectangle to square height transformation (pink).

There are three types of rectangle R to square S height h transformation as shown in Fig. 4. For T_1 the average of the west (W) and east (E) is calculated (1). For T_2 and T_3 a weighted average is calculated out of the northwest (NW), northeast (NE), southwest (SW) and southeast (SE) grid corner (2, 3).

$$h_S = 1/2*(h_{R(W)}+h_{R(E)}) \quad (1)$$

$$h_S = 1/3*(h_{R(NW)}+h_{R(NE)}) + 1/6*(h_{R(SW)}+h_{R(SE)}) \quad (2)$$

$$h_S = 1/6*(h_{R(NW)}+h_{R(NE)}) + 1/3*(h_{R(SW)}+h_{R(SE)}) \quad (3)$$

The transformation also shifts the single height value from the given WGS84 grid to the centre of the squares.

3.3 Orienteering Map Objects

An orienteering map is a collection of vectored and carefully adjusted objects, either of punctual (0D) type (big single tree, landmark, single rock, …), linear (1D) type (foot path, street, creek, …) or areal (3D) type (meadow, forest, lake, …). A natural area is given a colour and optional pattern attribute which is an indication for the percentage of maximum speed, a competitor typical reaches on ideal paths. All symbols, colours and patterns are standardized by the International Orienteering Federation (IOF) [5].

As punctual objects are mainly required for orientation, they can simply be ignored for later raster attribution, as they do never influence the speed of a competitor. 1D and 2D objects can indicate both, passable and impassable character. While passable objects modify the average speed of a competitor, impassable objects are a rather strong restriction. Linear objects which raise neither one nor other e.g., a high voltage power line are out of interest for raster attribution.

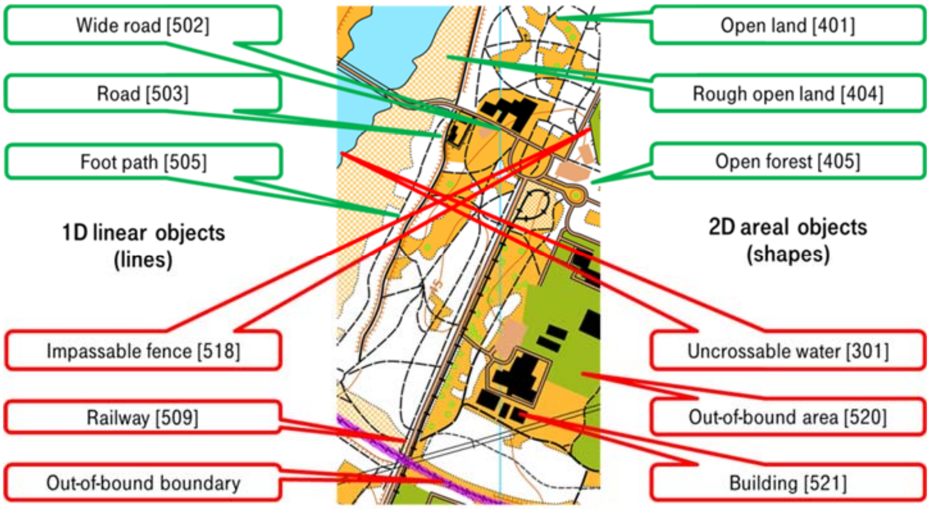


Fig. 5. Section of orienteering map "Magdeburg Herrenkrug" with samples of passable (green) and impassable (red) 1D and 2D objects. In brackets [ ] the ISOM-ID.

All symbols are standardized in shape, size, colour and pattern. The International Specification for Orienteering Map (ISOM) applies a basic colour scheme, where spot colour printing should be processed in the following order:

- {White}: Open forest without ground vegetation
- Yellow: Open areas and culture land
- Green: Natural ground vegetation
- Grey: Canopy
- Brown: Landform relief and contour lines
- Blue: Water and marsh
- Black: Path network, rocks and manmade features
- Purple: Race specific add-on information

For raster attribution a subset of the ISOM standard is selected that influence clearly the speed of the competitor. Both, passable objects (Table 5) and impassable objects (Table 6) are required to be taken into consideration. Section 4 will introduce the overall modelling and provide application examples.

4 Raster Model Building

Since classical path search algorithms rely on graphs, the given map has to be transformed to such. Using regular shapes (Section 2), the map gets rastered into a network of nodes. Each of them represents a single part area that the map has been decomposed into. A suitable shape is a square of 20 x 20 m that easily allows the later import of altitude data (Section 3). Using these squares, information will be extracted from the map in four steps:

1. Each square obtains a numerical value that is associated with the underlying areal objects' properties.
2. Linear objects modify these values.
3. Correct insertion order of objects has to be guaranteed so that dominant objects win over non-dominant ones.
4. Altitude data will be used for the final transformation into a bi-directional graph.

4.1 Areal Objects

In an orienteering map, properties of areal objects are represented by their colour and colour pattern, respectively (Fig. 6a). Their appearance and meaning are standardised [5]. They often already represent a normalised speed value. An open forest (ISOM-ID 405) is the reference that will be set to a normalized, unitless speed value of 100. Most symbols describe the vegetation and soil condition that have the biggest impact on the runner's speed.

Table 5. Visualisation of crossable areas and usable paths at the map [5].

Object	ISOM-ID	Speed adjust	Object colour	Object pattern
Indistinct marsh	310	40	Blue 26 %	Horizontal dashed line
Open land	401	90	Yellow 100 %	Solid area
Rough open land	403	80	Yellow 50 %	Solid area
Open forest	405	100	White (no colour)	Solid area
Vegetation, slow running	406	70	Green 20 %	Solid area
Vegetation, walk	408	40	Green 50 %	Solid area
Vegetation, fight	410	10	Green 100 %	Solid area
Paved area, Wide road, Road	501 502 503	130	Brown 50%	Black border line
Vehicle track Footpath	504 505	120	Black 100 %	Dashed line
Small footpath	506	110	Black 100 %	Dashed line
Less distinct small path Narrow ride or linear trace	507 508	100	Black 100 %	Double dash Long dashes

A direct connection between areal object's properties and real speed is a very complex and hard to describe, because they depend on physical and cognitive performance as well [6]. Thus, the normalised speed value is an approximation that is admissible to build the model. Using a lookup table (Table 5), each colour pattern has a fixed mapping to a normalised speed value v(n) where n identifies the square (Fig. 6c).

In the basic case, the speed value is the real speed projected on the horizontal plane. Otherwise the altitude change between two squares must be considered as well, when calculating the distance. With this simplification, the two-dimensional one is admissible.

Special areal objects are out-of-bound areas (Table 6). Some of them are explicitly designated like private estates and may result in a penalty. Others implicitly restrict the accessibility of an area like buildings or deep lakes. They have a fixed value of $v(n) = 0$.

Table 6. Visualisation of uncrossable areas and barriers at the map [5].

Object	ISOM-ID	Speed value	Object colour	Object pattern
Cliff	201	0	Black	Solid line with dashes
Uncrossable body of water	301	0	Blue Black	Solid area Borderline
Vegetation impassable	411	0	Green 50 % Black 50 %	Solid area
Cultivated land	412	0	Yellow 100 %	Solid area with black dots
Impassable wall Impassable fence	516 518	0	Black 100 %	Solid line with double dots, dashes resp.
Area that shall not be entered	520	0	Yellow 100 % Green 50 %	Solid area
Building	521	0	Black 100 % or 65 %	Solid area with black frame
Out-of-bound area	709	0	Purple 29 %	Line grid

After the map has been sectioned into regular shapes, many of them will cover a borderline of two or more different areal objects. In this case, the dominant colour of the shape will determine the normalised speed value (Fig. 6b). A colour is considered dominant when it has the highest part in the square compared to other colours.

Finer raster resolutions than 20 m by 20 m would provide a more detailed and more exact model, if a precise map was available. But on the other side, they would increase the computing time as well. Since orienteering maps only give a generalised representation of the landscape, they lack in details in smaller scales. Thus, finer resolutions are not expected to improve the model.

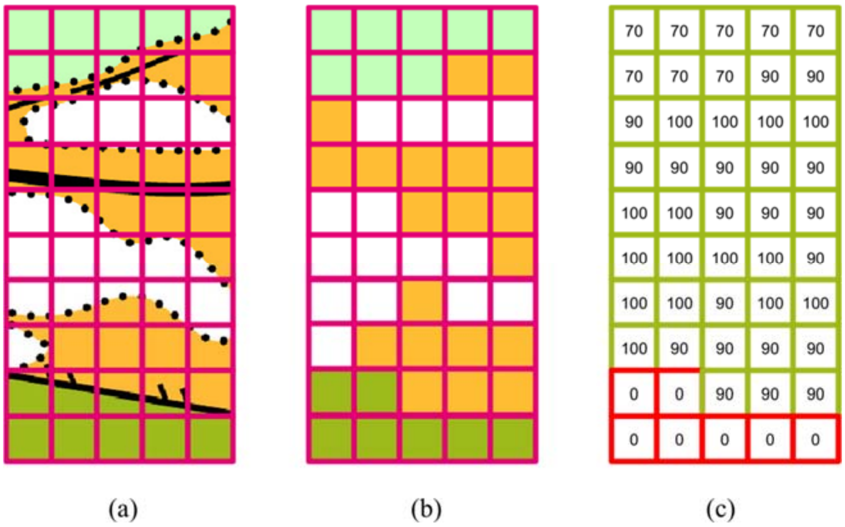


Fig. 6. Extraction of areal objects' properties from a map. (a) Part of the map rastered into squares, (b) Dominant colour of each square, (c) Normalised speed values assigned to each square.

4.2 Linear Objects

Following the areal object extraction, linear objects must be transferred into the model. Linear objects can be classified into:

1. Objects with distinct normalised speed value (Table 5).
2. Decelerators that scale the areal object's normalised speed value down (Table 7).
3. Barriers that are impassable (Table 6).

Roads and wide paths take course through the landscape as individual objects. They have greater normalised speed values v_{lin} than neighbouring areal objects. Thus, linear objects reassign new values $v'(n)$ to the squares they pass (4). The values are fixed but never smaller than the underlying areal object $v(n)$.

$$v'(n) = max(v(n), v_{lin}) \tag{4}$$

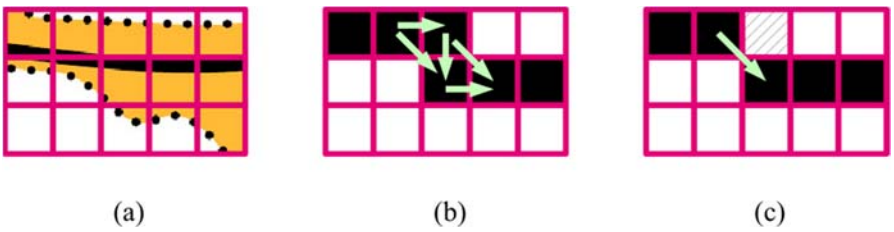


Fig. 7. The path split problem. (a) Part of the map, (b) Multiple paths extracted from the map, (c) Removing the second path.

Care has to be taken when inserting these objects. If they go along the border of two squares (Fig. 7a) they could be assigned to both of them. Especially when they hit a corner of four squares, this assignment will be inaccurate. Here three squares sharing a common corner would get assigned the linear object. Considering the neighbourhood relationships of these squares, the path will be split into three possible ways which is an illegal misrepresentation.

Table 7. Visualisation of decelerating objects at the map [5].

Object	ISOM-ID	Speed scaling factor	Object colour	Object pattern
Earth wall	105	0,7	Brown	Solid line with dots
Small knoll	109	0,5	Brown	Dot
Small depression	111	0,5	Brown	Half-circle
Pit	112	0,3	Brown	V-shape
Trench	215	0,6	Black	Border line
Crossable watercourse	304	0,6	Solid line	Solid line
Minor water channel	306	0,8	Blue	Dashed line
Narrow marsh	309	0,9	Blue	Dotted line

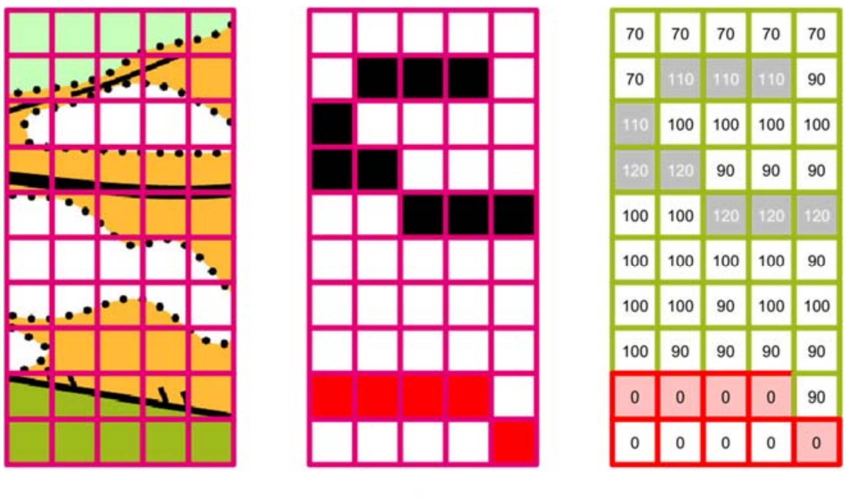


Fig. 8. Extraction of linear objects' properties from the map. (a) Part of the map rastered into squares, (b) Linear objects found on the map, (c) Re-assignment of speed values.

Thus, the insertion algorithm has to ensure that a single linear object has no more than two direct neighbour squares. There is an approach to regularly insert it into all squares it touches (Fig. 7b). If the map data is a bit-mapped graphic the linear objects must be transformed to a set of vectors prior to this in order to avoid ambiguity. Then all squares that cause a split of the path have to be removed in a manner that the square with the shortest part of the linear object will lose it (Fig. 7c).

This procedure is repeated for the other linear object classes as well (Fig. 8b). They differ in the assignment of speed values (Fig. 8c). Decelerators like ramparts or ditches are special features of the landscape. They have no fixed speed value, but a scaling factor k that reduces the speed value v' (5) of the underlying areal object.

$$v' = k * v\ (n) \tag{5}$$

4.3 Order of Insertion

Like restricted areas, linear barriers (fences, railways etc.) assign the square that they pass a value of zero. At this point, the order of including the different linear objects becomes important. Barriers should always be dominant over other objects. Else a possibly dominant path could remove a neighbouring barrier in the same square and enable a passage that is not present in the original environment. On the other hand a bridge under a railway or a street through a restricted area explicitly provides a valid passage. Thus following order in insertion is suggested:

1. Areal objects' dominant colour patterns are mapped to the squares.
2. Decelerating linear object scale the speed values.
3. Objects with distinct speed values replace the previously assigned ones if they are greater.
4. Linear barriers are inserted.

Every square that has both barriers and paths with a distinct value that do not cross each other needs a special treatment. The barrier is moved to the neighbour square so that neither the barrier nor the path gets deleted.

The result is a matrix of normalised speed values that will be transformed into a unidirectional graph in the next step (Fig. 9b). Each node of that graph represents a square. Nodes are linked with their direct neighbours. The weight $w\ (n, m)$ of the edges is the mean speed of the two square n and m that they are connected to (6). A special case is squares with a value of zero. Their links to their neighbours are deleted which completely disconnects them from the network.

$$w\ (n, m) = \tfrac{1}{2} * (v'\ (n) + v'\ (m)) \tag{6}$$

Furthermore, the map has to be re-scanned to passages under barriers or restricted areas and bridges. Those objects have been ignored in the previous step of insertion. They provide additional links in the network. An example is a street crossing another street on a bridge. Although each of them solely is passable they have no direct connection. The balustrade of the bridge will be shown as a barrier on the map that would cut the underlying street. This step ensures that the connection is re-established.

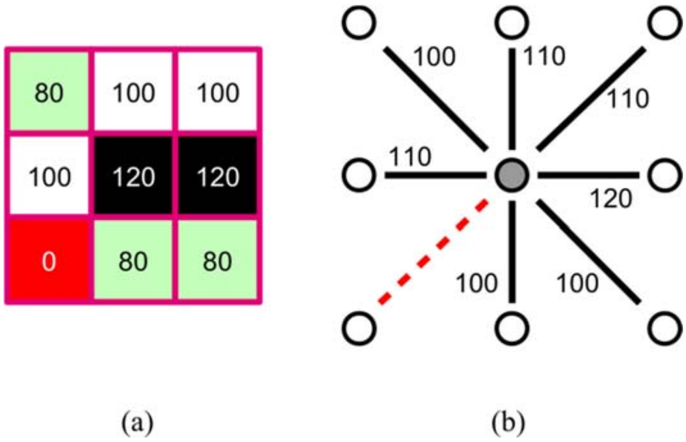


Fig. 9. Building a graph from normalised speed values. (a) Matrix of normalised speed values, (b) Graph with average speed values on its edges.

4.4 Applying Altitude Data

Finally, the SRTM data set is incorporated (Fig. 11a). Each single square will be a dedicated absolute altitude value in meters *e (n)* assigned to. An increase or decrease in the elevation profile influences the runner's speed. Steeper ascents' impacts are bigger.

An increase would usually lower the normalised speed value in contrast to a decrease that speeds the runner up to a certain degree. Too steep decreases however have a negative effect on the speed as well. Based on experiments, a lookup table (LUT) is utilised to resemble this fact (Table 8). It delivers a value that scales the edge's weight. The lookup table relies on empirical data and is might possibly not be defined by an analytical function (Fig. 10).

Table 8. Lookup table to transform altitude differences to speed correction values for a distance *d (n, m)* = 20 m.

Altitude difference $\lvert \Delta e\ (n, m) \rvert$	Steepness $\lvert \Delta e'\ (n, m) \rvert$	Increase adjust *LUT (e' (n, m))*	Decrease adjust *LUT (e' (n, m))*
0 m	0.00	1.00	1.00
1 m	0.05	0.98	1.02
5 m	0.25	0.80	1.10
10 m	0.50	0.55	1.08
15 m	0.75	0.40	0.85
20 m	1.00	0.25	0.25
40 m	2.00	0.00	0.00
60 m	3.00	0.00	0.00

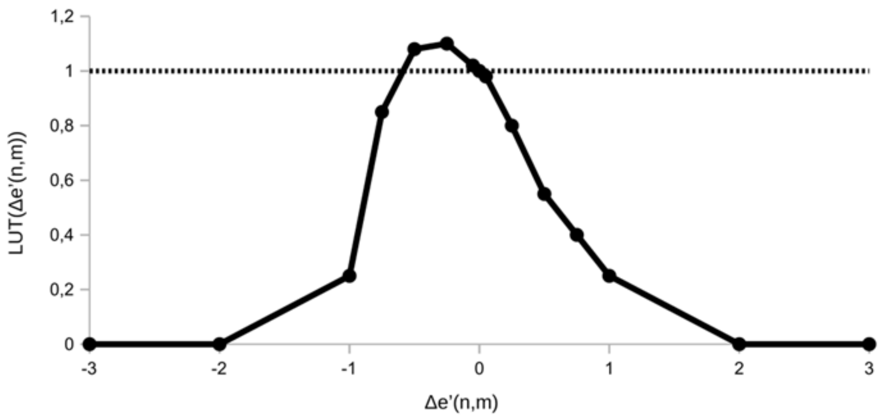


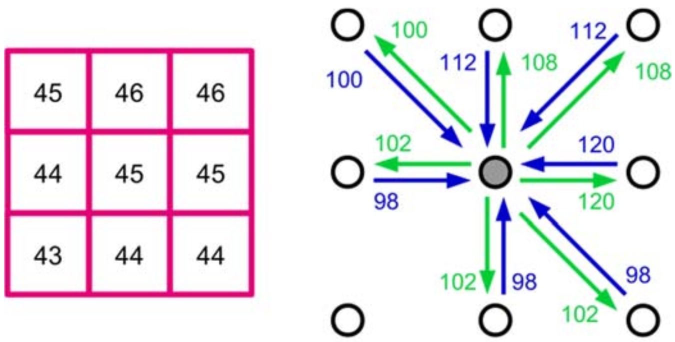
Fig. 10. Interpolated empirical values form the complete altitude correction lookup table.

The steepness is correlated to the difference of altitude of two squares *n* and *m* and their altitude difference $\Delta e\ (n,\ m) = e\ (m) - e\ (n)$ in meters (7). The steepness is usually normalised on the horizontal distance between the centres of the squares *d (n, m)*.

$$\Delta e'(n,m) = \frac{\Delta e\,(n,m)}{d\,(n,m)} = \frac{e\,(m) - e\,(n)}{d\,(n,m)} \tag{7}$$

These values are used to modify the weights *w (n, m)* of the graph's edges. The sign of the slope between two squares depends on the direction. Thus is necessary to build a bi-directional graph where an increase or decrease of altitude is considered separately (Fig. 11b). The value delivered from the LUT scales the speed value (8).

$$w'\ (n,\ m) = w\ (n,\ m) * LUT\ (\Delta e'\ (n,\ m)) \tag{8}$$



(a) (b)

Fig. 11. Bidirectional graph including altitude data. (a) Altitude data, (b) Graph built from speed values and altitude data.

The following assumptions are made on the non-linear function representing the lookup table:

1. There is no effect if the altitude does not change between neighbours.
2. Very high differences in altitude (cliffs for example) reduce the speed to zero.

Actually, individual data have to be found for both the mapping of objects to a speed value and for modifiers of the speed (like decelerators and altitude) since the physical conditions of every sportsman is different. In the usual case, a general model of an average sportsman is sufficient. But it should be noted that other models might have impact found route [7]. They might for example punish off-road parts in favour of streets.

5 Route Calculation

Classical graph search algorithms can be employed to find an optimal route on the given map after it has been transferred into a graph. The problem is visiting all control points and then returning to the start position which is known as the travelling salesman problem (TSP). Since a bidirectional graph is given it is an asymmetric TSP (ATSP) [8].

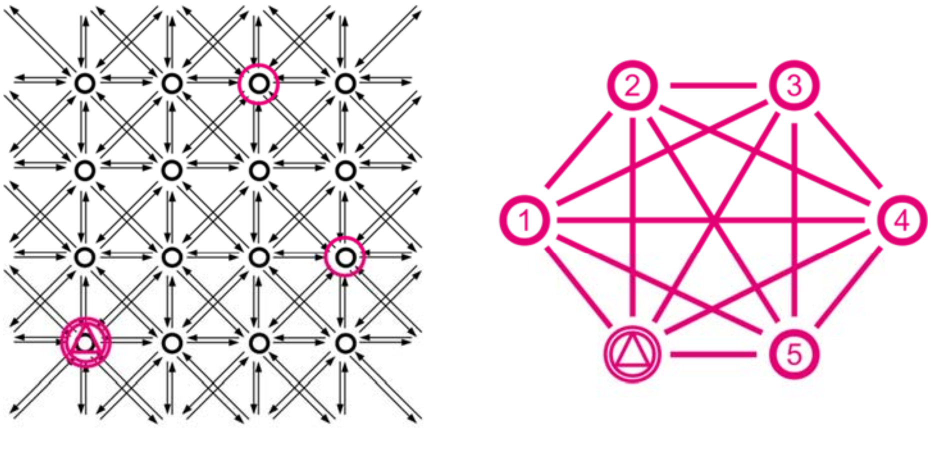
The graph obtained by extracting the map's objects (Fig. 12a) is not eligible for directly solving the TSP. Using the original graph, the solution would yield a route visiting all squares, which is not feasible.

5.1 Graph Reduction

Prior to applying the TSP, it is necessary to reduce the graph until its nodes solely are the control points including start and finish (Fig. 12b). This can be reached by pairwise searching for the shortest path between each control point. The A* algorithm [9] is proposed therefore.

The A* algorithm finds the path in the graph that has the smallest cost. Smallest cost refers to minimum time in this case. It is an iterative algorithm. In each step, it expands its solution towards the goal, using an estimate of the cost $f(n)$ required going to that final node. The estimate is calculated from the real cost $g(n)$ needed to go from the start node to n and a heuristic $h(n)$ that estimates the cost from n to the goal without knowing the real costs (9). An admissible heuristic never overestimates the real costs.

$$f(n) = g(n) + h(n) \tag{9}$$



(a) (b)

Fig. 12. Graph reduction by pairwise searching the local optimal path. (a) A part the original graph showing the square nodes as black circles and some control points marked purple, (b) Reduced graph with the full set of control points.

Given a start and end point, it finds always the path with the smallest cost. The original path represents the speed on its edges. But the criterion of optimisation is the time needed to complete the competition. Thus the speed needs to be converted into a time representing value. The time *t (n, m)* required to go from a square *n* to *m* and speed *w'(n, m)* between the squares are reciprocally connected by the distance between the centres of them *d (n, m)* (10). Attention has to be paid to the different distances of direct edge neighbours and diagonal neighbours (Section 2).

$$t\left(n,m\right)=\frac{d\left(n,m\right)}{w'\left(n,m\right)} \tag{10}$$

An essential part of the A* algorithm is its heuristic. If it is zero the A* resembles the Dijkstra algorithm. But a better heuristic will greatly improve the performance of the A* algorithm. In this application, a heuristic based on the beeline is proposed. The direct distance from the node recently visited by the algorithm and the goal is measured as if there were no obstacles. It is then converted to a time value. Because the heuristic must not overestimate the time, a reasonable speed value for conversion is required. It shows that the maximum speed on the whole map meets this requirement. In worst case, the time will always be underestimated, which may decrease the algorithm's performance, but never gives wrong results. A proof can be shown, using the triangle inequality. At the end of its search, the A* algorithm estimated time to the goal $f\ (n_{goal})$ equals the real one $g\ (n_{goal})$ (11).

$$f\ (n_{goal}) = g\ (n_{goal}) \tag{11}$$

If the heuristic *h (n)* underestimates the time and is added to the currently found time *g (n)* it will always be lower than the real time required (12).

$$g\ (n_{goal}) \leq g\ (n) + h\ (n) \tag{12}$$

The pairwise application of the A* algorithm on every control points yields a full matrix of minimum cost in respect of time needed to travel to them. It is further interpreted as an adjacent matrix, forming a second, reduced, bi-directional graph that only has the control points as its nodes. The result of the local optimisation gives the minimum time required for going from one to the other on the edges of the graph.

5.2 Solving the Travelling Salesman Problem

Now, the ATSP can be applied to the new graph. Several algorithms exist to solve it [10]. The solution gives a path that connects every control point and is the global optimum (Fig. 13). All edges of the optimal path get re-expanded, using the previously found local routes on the original graph. The so obtained route can easily be mapped onto the orienteering map where each square that it has been decomposed to is a node.

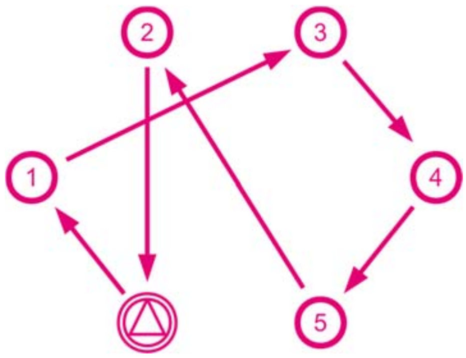


Fig. 13. Solving the ATSP on the reduced graph.

The final result can be exported to a common file format like the GPS exchange format (GPX)[2]. A mapping to WGS 84[3] conform geo-coordinates is required in this case. That allows the visualisation to be done in any software capable of reading this file format (Fig. 14).

[2] GPX 1.1: the GPS Exchange Format. URL: http://www.topografix.com/gpx.asp

[3] The World Geodetic System 1984. URL: http://gisgeography.com/wgs84-world-geodetic-system/

Fig. 14. Calculated route, real route and beeline on the map "Magdeburg Herrenkrug".

6 Review of the ARDF Problem Solution

6.1 Summary of the Two-step Route Optimization Approach

Publicly available DEM e.g., SRTM data, are native raster data. The available resolution of 1 arc sec equals to a rectangle of approx. 20 by 30 meters in Central Europe. For the model building, see Fig. 15, rectangles will be substituted by squares. Each square will be assigned a set of independent attributes, where, besides altitude, the type of vegetation is the most important one. Both, linear and areal objects are taken into consideration. They can be extracted from an orienteering map as majority decision for the texture pattern of each given square.

Afterwards, areal and linear vector objects are mapped onto the neighbourhood relations by a speed adaptation factor or by cutting the neighbourhood in case of barriers. As sometimes closed areas offer pathways, food paths and roads are calculated last as overlay.

Altitude differences influence the speed of the competitor. As experimental investigations show, the adaptation is individually. Therefore, a LUT is recommended to represent the correction factor to be applied to the entire model. Depending on the direction different corrections must be applied, which results in a bi-graph. This methodology is also applicable for other personal characteristics.

In the following steps, known algorithms of asymmetric path optimization are applied e.g., A* to calculate an optimal route as a sequence of first order raster fields by solving the ATSP.

Fig. 15 illustrates the entire chain of model building and route calculation in eight sequential steps.

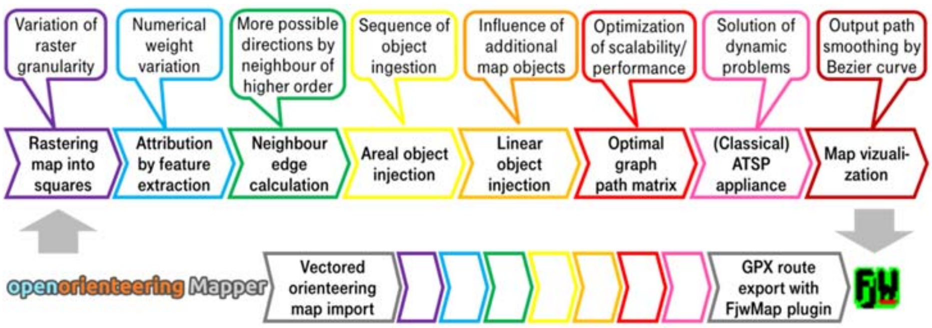


Fig. 15. Process steps of modelling/calculation plus further ideas per process step.

The selected square resolution of about 20 m by 20 m fits to cover a typical orienteering map of up to 16 square kilometres.

6.2 Further Work

Orienteering maps are often created with the *OpenOrienteering Mapper*. The open source project [11] offers a rich function set, already exceeding comparable commercial tools. An exporter could be defined to transfer all relevant linear and areal objects into the raster approach.

With *FjwMap* a well-established route planning and visualization tool is freely available [12]. Applying GPX, a common import format is already supported. An import plug-in for the raster approach results is proposed.

For each step of the processing queue shown in the upper boxes of Fig. 15, a key improvement has been already identified. By extending the variety and number of barrier objects, the verification of the right order of map feature extraction required, not to miss potential passages.

The proposed solution does not yet consider classical objective ARDF specifics, namely the uncertain CP locations as well as transmission to silence ratio of 1 to 4. Those should be called dynamic problems.

In a next step, an individual parametrisation will be needed. LUT for speed adjusting should be personalized and verified by further experiments and analysing publicly available competition results[4].

Depending on the estimated distance to the upcoming CP, the competitor adjusts the speed and movement direction often in a special manner to succeed logging the CP even out of the transmission time. Furthermore, the influence of increasing tiredness during a competition should become part of the personal model.

References

1. Le, P.: Open Orienteering Map: Magdeburg Herrenkrug – scale 1:15.000, March 2016.
2. Roth, J.: Navigation durch Flächen. In: Proceedings of the 13th Workshop on Location Based Applications and Services (LBAS), Jena (Germany), September 2016.
3. U. S. Department of Defence: Performance Specification: Digital Terrain Elevation Data (DTED), Standard MIL-PRF-89020B. URL: https://dds.cr.usgs.gov/srtm/version2_1/ Documentation/MIL-PDF-89020B.pdf, last accessed October 2017.
4. U.S. Geological Survey (USGS): SRTM Topography. URL: https://dds.cr.usgs.gov/srtm/version2_1/Documentation/SRTM_Topo.pdf, last accessed October 2017.
5. International Orienteering Federation: ISOM 2017 – International Specification for Orienteering Maps. ISBN: 978-91-639-3394-3, March 2017.
6. Kolb, H.; Sobotka, R.; Werner, R.: A Model of Performance-Determining Components in Orienteering. In: Scientific Journal of Orienteering 3(2), pp.71-81, 1987.
7. Myrvoldt, B. O.: Is it Possible to Find a "Best" Route? A Look at Accuracy and Significance in Route Choice Comparison. In: Scientific Journal of Orienteering 12(1), pp.19-36, 1996.
8. Schmidt, L.; Wolff, D.: Varianten des Travelling Salesman Problem (TSP). URL: http://ls11-www.cs.tu-dortmund.de/people/chimani/GA09/ue/kv57.pdf. Presentation at Technical University Dortmund, January 2010.
9. Hart, P. E.; Nilsson, N. J.; Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. In: IEEE Transactions on Systems Science and Cybernetics SSC4. 4 (2), pp. 100–107, 1968.
10. Rego, C.; Gamboa, D.; Glover, F.; Osterman, C.: Traveling salesman problem heuristics: Leading methods, implementations and latest advances. European Journal of Operational Research, 211 (3): 427–441, 2011.
11. Pastor, K.: OpenOrienteering – Mapper cross-platform, open source, release 0.7.0. URL: http://www.openorienteering.org/ June 2017.
12. Schade, K.-H.: FjwMap. URL: http://www.dl7vdb.de/index_eng.htm#fjwmap, last accessed October 2017.

[4] German ARDF result URL: http://ardf.darc.de/contest/