Demand Allocation in Water Distribution Network Modelling – A GIS-Based Approach Using Voronoi Diagrams with Constraints

Nicolai Guth and Philipp Klingel

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/50014

1. Introduction

In water distribution network modelling the topology of the network is commonly mapped as a (directed) graph consisting of nodes and links (Walski et al., 2001). Pipe sections with constant parameters are modelled as links. Intersections, water inlets, points of water withdrawal and locations of pipe parameter changes are modelled as nodes. Coordinates related to the nodes define the spatial location. To calculate the hydraulic system state (pressures and flows) the water demand of the consumers is assigned to the relevant nodes as the driving parameter (demand driven simulation). Figure 1 schematically shows the abstraction of a water distribution network.

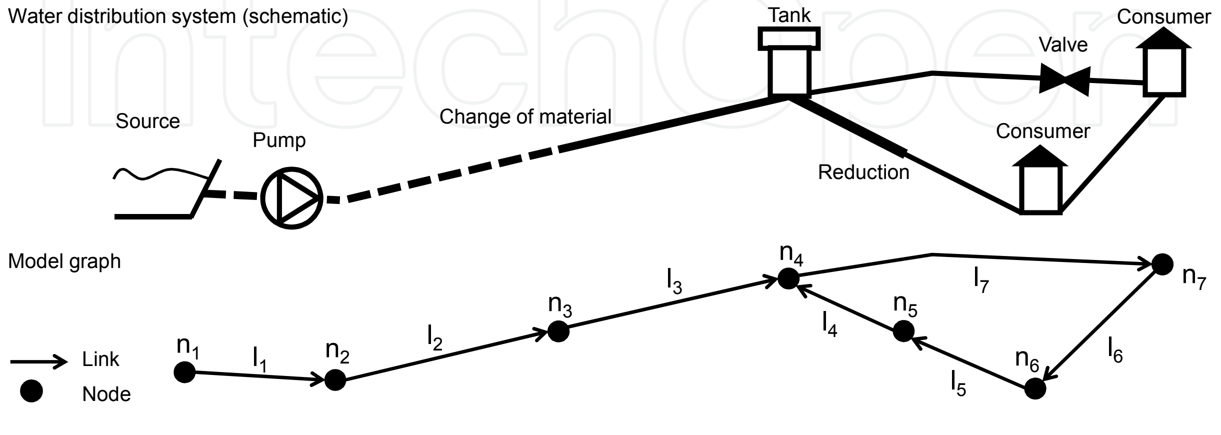
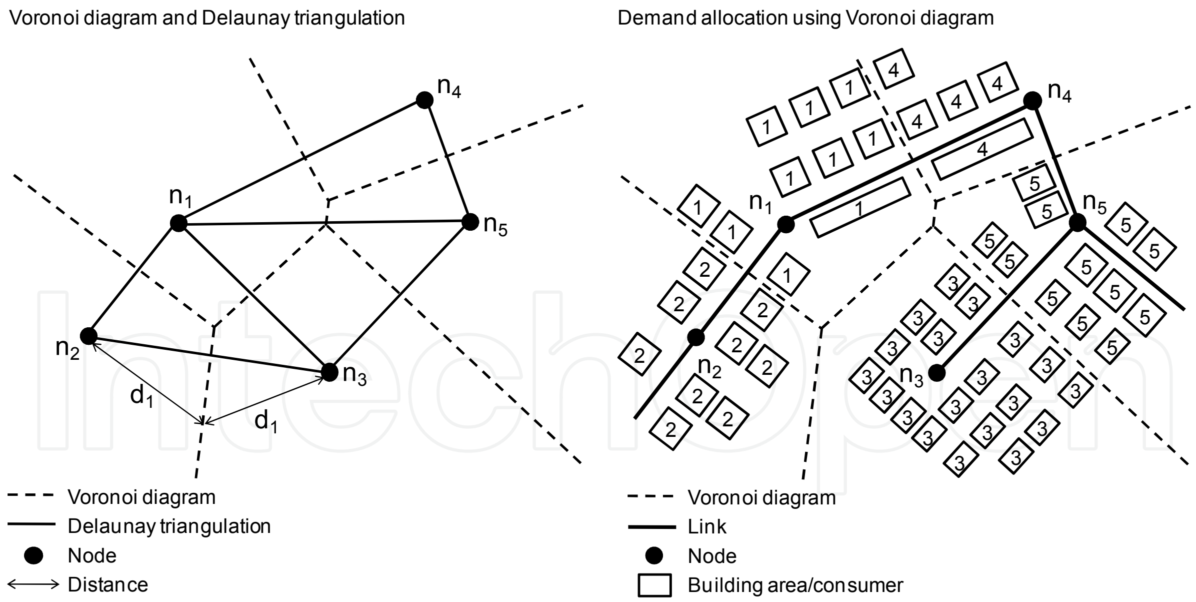


Figure 1. Illustration of a water distribution system and the corresponding model graph (Klingel, 2010)

To reduce the model size typically not every known demand (e.g. demand at a house connection) is modelled as a node. Besides, location and demand of consumers are often not known, especially in developing countries.

To allocate demands to the nodes Voronoi diagrams might be used. The Voronoi diagram is also known as Thiessen polygon and dual to the Delaunay triangulation (c.f. fig. 2, left sketch). The Voronoi diagram is based on a set of nodes and defines one area for each node with every point within the area being closer to the originating node of the area than to any other node. In the following these areas are referred to as Voronoi regions. They represent the catchment areas of the nodes. The borders of the Voronoi regions constitute of lines called bi-sectors. Thus, known demands (value and location) can be allocated to the closest nodes according to the Euclidean distance. This approach is also commonly applied to determine, for example, the rainfall catchment areas of inlets of sewer networks.

If demands and their location are not known, they can be determined based on the size of the Voronoi regions, the population distribution and density and a consumption per capita value. Alternatively, building areas can be allocated to the nearest nodes based on the Voronoi diagrams if the correspondent data is available. In this case the actual demand is calculated by using a coefficient relating the population to a building area unit and a consumption per capita value. The sketch on the right in figure 2 schematically shows the allocation of demands (in this case building areas) to nodes using a Voronoi diagram.


Voronoi diagram and Delaunay triangulation

Demand allocation using Voronoi diagram

Figure 2. Illustration of a Voronoi diagram and a Delaunay triangulation and demand allocation using a Voroinoi diagram

Pipe network data, population distribution and building data are commonly stored and managed in geographic information systems (GIS). Thus, a GIS-application to allocate water demand values to the network model nodes is self-evident and commonly applied.

The calculation of common Voronoi diagrams is based on a set of nodes and the Euclidean distance between the nodes and the points of the areas. Various well known and often used algorithms to generate such Voronoi diagrams exist, for example the sweep-algorithm (Fortune, 1986; Preparata and Shamos, 1985) and the divide-and-conquer-approach (Shamos, 1975).

The approach to allocate demands using such common Voronoi diagrams follows the assumption that a consumer is connected to the closest existing pipe and to the closest existing node respectively. Boundaries and obstacles are not considered as constraints. In this context, areas with different population densities, for example, constitute boundaries and railway trails and lakes within the supply area, for example, constitute obstacles which cannot be passed by distribution pipes. Thus, with common Voronoi diagrams consumers, areas or building areas are possibly allocated to nodes which are closest according to the Euclidean distance but are not closest if boundaries or obstacles are considered. This constitutes a significant limitation of the application of Voronoi diagrams based on the Euclidean distance for the allocation of water demand.

If boundaries are convex, no obstacles are located in between two originating nodes of a bi-sector. The Voronoi diagram can still be calculated based on the Euclidean distance. The convex boundaries can be considered easily by intersecting the Voronoi diagram with the convex boundary. Conversely, sections of non-convex boundaries or obstacles might be located in between two originating nodes of a bi-sector. In these cases the determination of the bi-sectors based on the Euclidean distance is not sufficient any more. The shortest way to bypass the non-convex section or obstacle respectively has to be considered instead.

Okabe et al. (2000) describe this problem as "Voronoi diagrams with obstacles" and introduce the "shortest path Voronoi diagram". Thereby, obstacles are supposed to be considered by calculating Voronoi diagrams based on the shortest path instead of the Euclidean distance. A satisfactory solution of this approach would result in an increased model accuracy and therefore higher reliability of the simulation calculations.

Figure 3 illustrates the difference between a Voronoi diagram, which is determined based on the Euclidean distance (left sketch) and based on the shortest path (right sketch). Not considering (convex sections of) the boundaries, for example, results in an allocation of areas to node n_1 which can only be part of the Voronoi regions of the nodes n_4 and n_5 if the boundaries given in the example are respected. Not considering the shortest path to bypass non-convex sections of the boundaries leads, for example, to an allocation of an area to node n_2 instead of node n_1. Accordingly, an area is allocated to node n_3 instead of node n_2 if the shortest path to bypass the obstacle given in the example is ignored and the Voronoi diagram is determined based on the Euclidean distance.

Several authors have tackled this problem. The approach of Aronov (1987) considers non-convex boundaries but not obstacles within the Voronoi regions. The approach of Papadopoulou and Lee (1998) does consider obstacles as constraints but only if specific conditions are given. None of the known approaches solves the problem satisfactorily for an application in the given context. The consideration of both, non-convex boundaries and obstacles would be desirable.

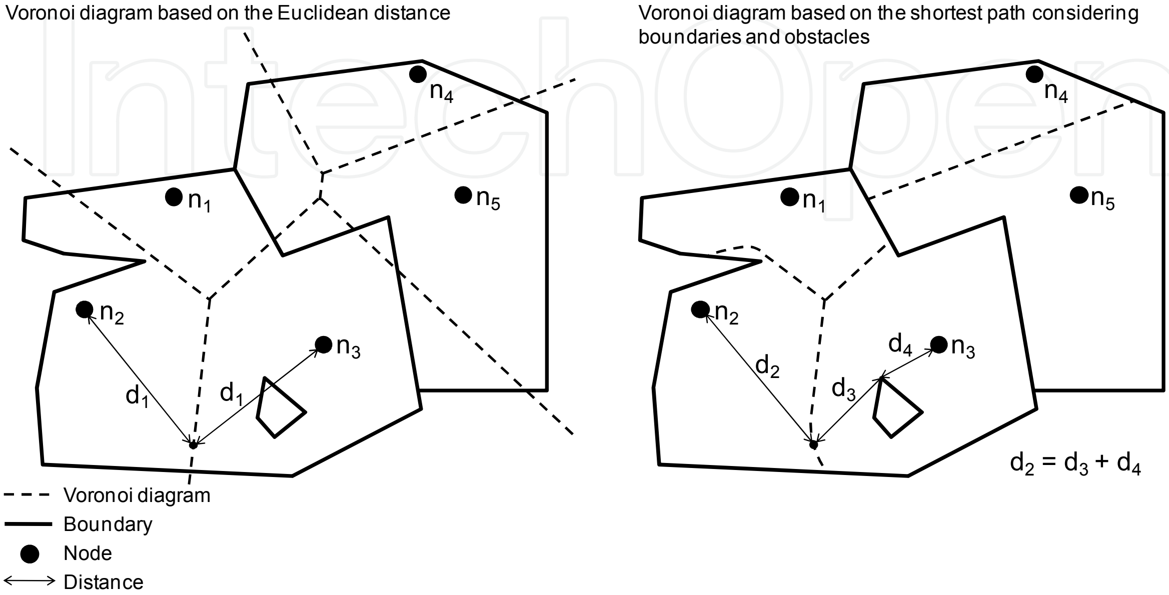

Voronoi diagram based on the Euclidean distance
Voronoi diagram based on the shortest path considering boundaries and obstacles
n_1
n_2
n_3
n_4
n_5
d_1
d_1
d_2
d_3
d_4
$d_2 = d_3 + d_4$
Voronoi diagram
Boundary
Node
Distance

Figure 3. Comparison of a Voronoi diagram based on the Euclidean distance and the shortest path (Klingel, 2010)

In this context, the additively weighted Voronoi diagram is of particular interest. Johnson and Mehl (1939) invented the additive weights of originating nodes of Voronoi diagrams to model the development of crystals, which grow with different speed into different directions. Every node is weighted with a positive or negative constant (additive constant) which is subtracted from the distance function when calculating the Voronoi diagram. Thus, contrary to a common Voronoi diagram based on two nodes, the bi-sector is not a straight line in the geometric centre between the nodes if the two nodes are weighted with different constants. The bi-sector constitutes a hyperbola being closer to the node with the smaller weight (Aurenhammer, 1991).

This chapter presents a novel approach to calculate Voronoi diagrams based on the shortest path instead of the Euclidean distance considering non-convex boundaries and obstacles, which Guth (2009) has developed in his Bachelor thesis. At critical points of boundaries and obstacles, the so called subordinated nodes are temporarily defined and weighted with an additive constant to consider the shortest path as distance function for the calculation of the Voronoi diagram. The approach is presented in section 2 of this chapter. The resultant algorithm is expounded in section 3. The testing of the algorithm and the implementation of the algorithm as a GIS-tool based on the software package ESRI ArcGIS (Ormsby et al., 2004) are demonstrated in sections 4 and 5.

2. Approach

The basis of the novel approach described herein is a geometric and iterative method to calculate Voronoi diagrams. Thereby, the pairs of originating nodes of a bi-sector (the immediate neighbours) have to be determined first. Each node of a pair is defined to be the origin (centre) of concentric circles. Circles with the same radius are intersected. The intersection points constitute points of the bi-sector. Thus, the Voronoi diagram based on the Euclidean distance can be calculated.

As discussed in section 1, to consider non-convex boundaries and obstacles, the determination of the Voronoi diagram has to be based on the shortest path. Thus, the shortest paths from every point of a bi-sector to each of the originating nodes of the bi-sector have the same length. This leads to a linear form of the bi-sector if the points of the bi-sector are 'visible' from both originating nodes (the length of the shortest path is equal to the length of the Euclidean distance) and to a hyperbolic form if a non-convex boundary section or an obstacle leads to a shortest path which is longer than the Euclidean distance (c.f. fig. 3).

When considering boundaries and obstacles they are assumed to be represented by polygons with vertices, which is the case in the context of the development of a GIS-solution. The critical points of bounding polygons are the extreme vertices of non-convex sections. Accordingly, the critical points of obstacles are the extreme vertices of convex sections. These critical points are identified and added to the set of given nodes (primary nodes) as subordinated nodes. A subordinated node is always related to the closest primary node. An additive constant which represents the shortest path length to that primary node is assigned to each subordinated node. In figure 4 an example of three primary nodes (n_1, n_2 and n_3), a bounding polygon and three obstacles is given. Subordinated nodes are defined at all critical points of the boundary and the obstacles. For example, the three subordinated nodes n_{11}, n_{12} and n_{13} are related to node n_1. The additive constant of the subordinated node n_{12} is also shown exemplarily in figure 4.

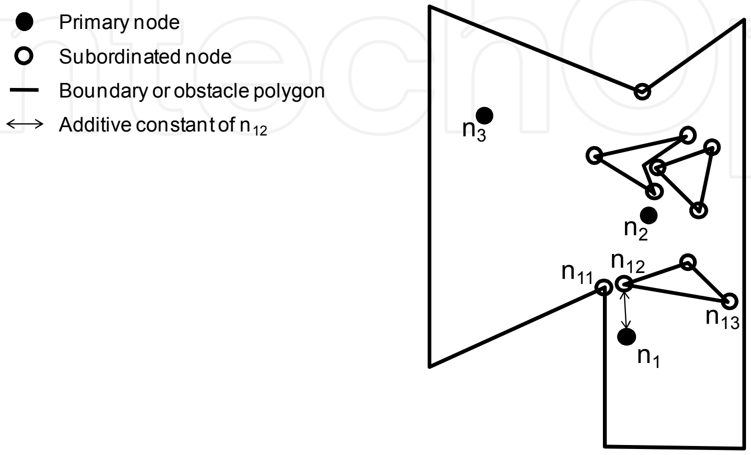


Figure 4. Illustration of subordinated nodes of a given configuration in a bounding polygon with three primary nodes and three obstacles

Even though the principle idea of using additively weighted nodes was invented by Johnson and Mehl (1939), their approach to calculate the Voronoi diagram cannot be applied successfully to solve the given problem. Transferred to the given context, the approach does only solve configurations where the additive constant of a subordinated node is shorter than the distance between the primary node and the related subordinated node (Okabe et al., 2000). In the given case the additive constant is always equal to the distance between primary and subordinated node. Thus, the development of a novel solution is needed.

Like the computation of Voronoi diagrams based on a set of primary nodes by intersecting circles, the determination based on primary and subordinated nodes can be regarded as the projection of the intersection of cones. Each node (located in the x-y-plane) represents the tip of a cone which is opened upwards. All cones have the same cone angle. The cones at the subordinated nodes are moved vertically up (z-direction) by the value of their additive constant. At a certain height (z-value), the intersection of the cones can be represented by the intersection of circles with the radius of the cones at that specific height (z-value). Cones related to primary nodes have the same radii at the same height. Thus, the projection (to the x-y-plane) of the intersection of cones related to primary nodes results in a linear shaped section of the according bi-sector. Accordingly, the intersection of the cones related to a primary and a subordinated node results in a hyperbolic shape due to the different radii at a certain height of the cones.

To simplify the calculation, the projected cones are treated (discretised) as concentric circles. The height above the plane is reflected by the circle's radius. Circles with the same radii are intersected to calculate bi-sector points. For the subordinated nodes' circles, the radii are decreased by the respective additive constant. Resultant radii below zero are not admissible.

For the calculation of the Voronoi diagram the radii of the circles are increased successively. For each radius (iteration), the circles which are related to the primary nodes or their subordinated nodes are intersected in several calculation steps. In one calculation step the circles of two nodes are intersected. If there is an intersection point which is visible from both circles' centres (originating nodes) and is not located within another circle, the point is stored as bi-sector point. Visibility may be limited by the bounding polygon or an obstacle. The difference between the radii of two sequent iterations constitutes the discretisation of the iterative approach.

The result of the approach is a Voronoi diagram consisting of polygons with discretely determined vertices. As GIS do not store mathematical functions, this is not a drawback. The resulting polygons can be directly stored without any further processing.

Figure 5 shows a schematic illustration of the approach. The given configuration comprehends two primary nodes (n_1 and n_2). A boundary polygon constitutes a critical point. The pertinent subordinated node n_{21} is related to the closest primary node, node n_2. In the area visible from both primary nodes (right of the 'visibility boundary' of node n_2) the bi-sector between the nodes is linear. The bi-sector vertices (black squares in the figure) are determined by intersecting concentric circles with the same radii. In the area not visible from node n_2 (left of the 'visibility boundary' of node n_2), the bi-sector has a hyperbolic shape. The vertices (black rhombi in the figure) are determined by intersecting the concentric circles of the primary node n_1 and the subordinated node n_{21}. The radii of the circles with the subordinated node n_{21} as origin are always decreased by the additive constant, which is the length of the shortest path between the nodes n_2 and n_{21}. In the figure, circles which are intersected, are plotted in the same colour.

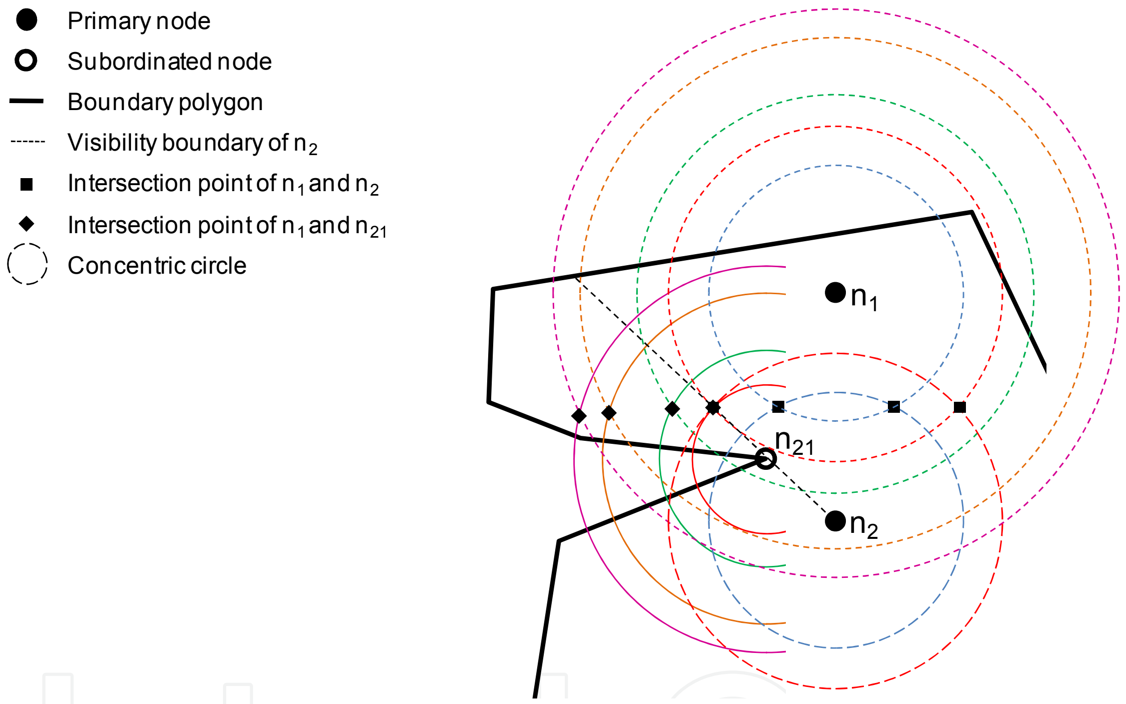

● Primary node
○ Subordinated node
— Boundary polygon
------ Visibility boundary of n_2
■ Intersection point of n_1 and n_2
◆ Intersection point of n_1 and n_{21}
◌ Concentric circle
n_1
n_{21}
n_2

Figure 5. Schematic illustration of the approach

3. Algorithm

3.1. Overview

For the realisation of the approach described in the previous section, an algorithm was developed, tested (c.f. section 4) and finally implemented in a GIS-tool (c.f. section 5). An overview of the individual steps of the algorithm is given in figure 6.

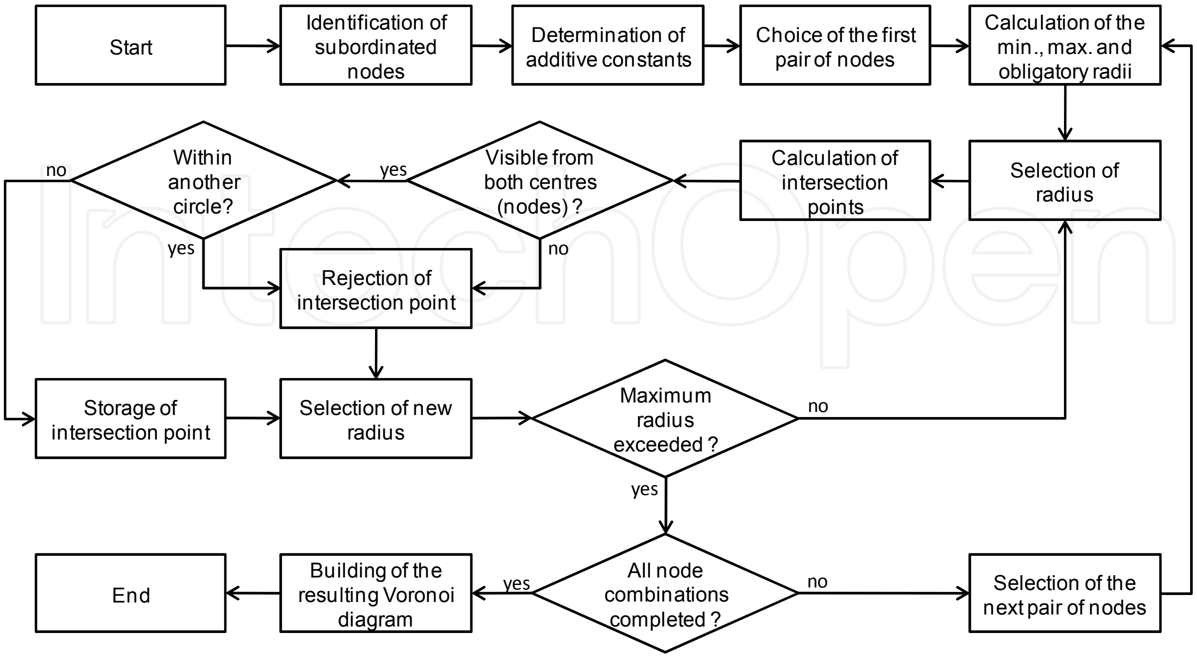

Start
Identification of subordinated nodes
Determination of additive constants
Choice of the first pair of nodes
Calculation of the min., max. and obligatory radii
Selection of radius
Calculation of intersection points
Visible from both centres (nodes) ?
yes
no
Within another circle?
no
yes
Rejection of intersection point
Storage of intersection point
Selection of new radius
Maximum radius exceeded ?
no
yes
All node combinations completed ?
yes
no
Selection of the next pair of nodes
Building of the resulting Voronoi diagram
End

Figure 6. Flowchart of the algorithm

First, the subordinated nodes of the given configuration are identified and weighted with an additive constant. Next, points of the bi-sectors are calculated by intersecting circles with corresponding radii and the nodes as origin. Finally, based on the calculated set of bi-sector points, the Voronoi diagram is determined. Thus, the algorithm can be divided into three principle parts: initialization, computation and creation of the resulting Voronoi diagram. Each of the following subsections describes one of the three parts.

3.2. Initialization

Identifying the subordinated nodes constitutes the first step of the initialization. Therefore, the bounding polygon and the polygons which describe the obstacles are analysed regarding the exterior angle at each vertex. For the bounding polygon this is facilitated by calculating the azimuths of the current vertex to the predecessor (t_1 in equation 1) and the successor (t_2 in equation 1) vertices. The difference between the azimuths is subtracted from the full circle. Thus, the outer angle β is calculated by equation 1:

$$\beta = 2\pi - (t_1 - t_2) \qquad (1)$$

The result is normalised to $\beta \in [0;2\pi)$ by successive subtraction of 2π where required. Assuming that the polygon is defined clockwise, the vertex is marked as a subordinated node if it holds that $\beta < \pi$. The angles β of all vertices are added ($\sum\beta$) and compared with the theoretical angular sum W. W is computed by equation 2, where n is the number of vertices:

$$W = (n + 2)\cdot \pi \qquad (2)$$

If $\sum\beta = W$, the polygon is defined clockwise and the marks are correct. Otherwise, the order of the vertices is switched and the algorithm is executed again. For the obstacle polygons, the same algorithm is used but the marks are switched in the end. Thus, only the vertices which stick out of the obstacle (convex sections) are considered as subordinated nodes. The identified subordinated nodes are added to the list of originating nodes.

In the succesive step, with the algorithm of Dijkstra (1959), the relation of subordinated nodes and primary nodes is determined and the additive constant for each subordinate node is calculated. The Dijkstra algorithm analyses the graph based on all subordinated and primary nodes with links between pairs of nodes which are visible to each other considering the given boundaries and obstacles. From each subordinated node the possible paths to all primary nodes are calculated. Thus, the shortest path determines the primary node related to the subordinated node. Furthermore, the length of the shortest path constitutes the additive constant.

3.3. Computation of bi-sector points

As mentioned before, the bi-sectors between the Voronoi regions are made up of intersection points of circle intersections. Node pair by node pair, circles of all nodes are intersected with each other to determine the intersection points. The process starts with a randomly chosen pair of nodes.

For the circles to be intersected, the minimum and maximum radius and obligatory radii are determined first. The minimum radius is where both circles touch. To identify the maximum radius, the maximal possible distances from each originating node (centres of the circles) to a visible point within the boundary are calculated. The shorter distance is used as maximum radius. The obligatory radii for each pair of nodes are those where the intersection points lie on a segment of the boundary or obstacle polygon. When the circles of a primary and a subordinated node are intersected, the obligatory radii have to be determined iteratively.

The iteration method is described here below and illustrated in figure 7. B_1 is the starting vertex of a segment, IP the unknown intersection point of circles with the centres (originating nodes) n_1 and n_2. The vector $\underline{v}$ points into the direction of the segment and is normalised to (length) 1. The unknown vectors $\underline{u}$ and $\underline{w}$ point from the intersection point to the respective centres.

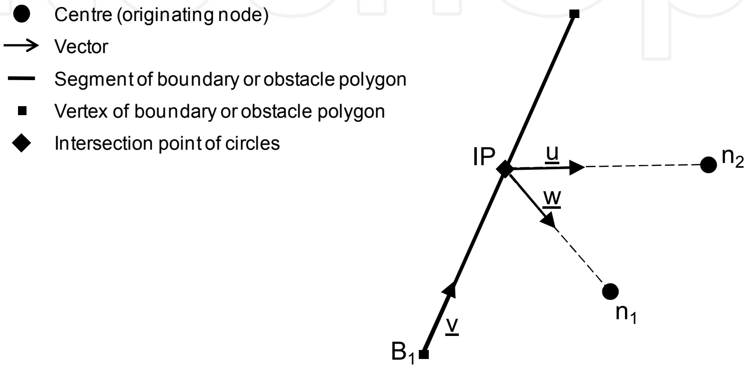

● Centre (originating node)
→ Vector
— Segment of boundary or obstacle polygon
■ Vertex of boundary or obstacle polygon
◆ Intersection point of circles
IP
u
n2
w
n1
v
B1

Figure 7. Setup for the iterative calculation of intersection points on polygon segments

To compute the unknown vectors, the following constraints (equations 3 to 7) are used with u_x, u_y, w_x, w_y, v_x, v_y as x- and y-components of the respective vectors, n_{1x}, n_{1y}, n_{2x}, n_{2y}, B_{1x}, B_{1y} as x- and y-coordinates of the respective points and m as length multiplication factor of vector $\underline{v}$, which expresses the distance from B_1 to IP:

$$u_x + n_{1x} - n_{2x} - w_x = 0 \tag{3}$$

$$u_y + n_{1y} - n_{2y} - w_y = 0 \tag{4}$$

$$m \cdot v_x + w_x + B_{1x} - n_{1x} = 0 \tag{5}$$

$$m \cdot v_y + w_y + B_{1y} - n_{1y} = 0 \tag{6}$$

$$\sqrt{w_x^2 + w_y^2} = \sqrt{u_x^2 + u_y^2} \tag{7}$$

Equation 3 and 4 express a closed vector-path from IP via n_1 and n_2 back to IP. Equation 5 and 6 represent the closed path from B_1 via IP and n_1 back to B_1. Equation 7 guarantees, that the vectors $\underline{u}$ and $\underline{w}$ have an equal length. Taking the constraints into account, m can be computed using equation 8:

$$m = \frac{n_{2x}^2 - 2 \cdot n_{2x} \cdot B_{1x} + n_{2y}^2 - 2 \cdot n_{2y} \cdot B_{1y} + 2 \cdot n_{1x} \cdot B_{1x} - n_{1x}^2 + 2 \cdot n_{1y} \cdot B_{1y} - n_{1y}^2}{2 \cdot (n_{2x} \cdot v_x + n_{2y} \cdot v_y - n_{1x} \cdot v_x - n_{1y} \cdot v_y)} \tag{8}$$

Finally, IP can be calculated with equation 9:

$$\underline{IP} = B_1 + m \cdot \underline{v} \tag{9}$$

In the first iteration, the centres are moved in normal direction away from the segment by the value of the additive constant. The normal direction is used as temporary vector for $\underline{u}$ and $\underline{w}$ and IP is computed. In the next iterations, the centres are moved in direction of the last $\underline{u}$ and $\underline{w}$ respectively. After the changes of $\underline{u}$ and $\underline{w}$ fall below a given bound, the final radii are calculated by adding the distance from IP to the centre and the additive constant.

After the minimum, maximum and obligatory radii are identified the intersection points (bi-sector points) related to the pair of nodes are calculated. Beginning with the minimum radii, the circles of the pair of nodes are intersected. For each intersection point it is checked if the point is visible from both centres and if no other centre is closer to the point. If one of the constraints is not fulfilled, the point is dismissed. Otherwise, the point is stored in a list, individually created for each pair of nodes. After the first intersections with the minimum radius, the radius is increased. The new radius is determined with a function, which takes the preceding intersection results into account. For small bends of the resulting bi-sectors the increment is larger than for intersections which form a strong bend. Increments may also be negative if necessary. When the maximum radius is reached, the obligatory radii for the points on the boundary and the obstacles are processed and the next pair of nodes is chosen.

3.4. Determination of Voronoi diagram

As the intersection points are stored in lists without order, the resulting polygons of the bisectors cannot be created directly. For each list a polyline is created first. This is done by picking the first point in the list and searching for the point with the greatest distance to it. This point is defined as the starting point of the polyline. Next, the closest point to the starting node is searched for, defined as the next point and marked. The process is repeated until no unmarked points are left on the list. To form the resulting Voronoi diagram, the individual polylines are merged by comparing the coordinates of the polylines' starting points and endpoints and assembling the nearest ones.

4. Testing

The developed approach and algorithm respectively are tested in order to evaluate the functionality and performance. Several configurations of given sets of nodes and boundaries are analysed by comparing the Voronoi diagram based on the Euclidean distance and the Voronoi diagram calculated with the novel approach. The comparison is mainly based on the size of the Voronoi regions. Two of the tested cases are discussed in the following.

The first case is a synthetic example configuration solely created to test the functionality of the developed algorithm. It consists of only three originating nodes (n_1, n_2, n_3), a non-convex boundary and three obstacles located within the boundary. Figure 8 shows the case and the calculated Voronoi diagrams based on the Euclidean distance and the shortest path. The individual sizes of the areas and the differences are shown in table 1. The differences do not sum up to zero as the approach is approximate and small holes between the individual Voronoi regions may occur. Furthermore, the values in the table are rounded to two decimal places.

Node	Voronoi region based on the Euclidean distance [length²]	Voronoi region based on the shortest path [length²]	Difference [length²]	Difference [%]
n_1	2,067.60	1,889.24	-178.36	-8.6
n_2	2,210.80	2,213.44	2.63	0.1
n_3	2,120.66	2,296.30	175.65	8.3

Table 1. Comparison of the Voronoi region sizes of case 1

The Voronoi region of node n_1 decreases significantly in size (-8.6%) with the new approach compared to the Euclidean Voronoi diagram due to the large area on the left. Considering the bounding polygon, this area is not visible from the primary node n_1. Thus, within the shortest path Voronoi diagram, the specific area is allocated to the Voronoi regions of the nodes n_2 and n_3 instead of n_1. The Voronoi region of node n_2 remains almost the same (0.1%): it gains area from the region of node n_1, but nearly the same amount is allocated to the region of node n_3 in the upper part of the image, where the bisector is bent to the right around the obstacle. Coming closer to the boundary the same bi-sector is bending a little to the left again. This is due to the change of the shortest path to bypass the left obstacle from the way around the obstacle on the left hand side to the way around on the right hand side

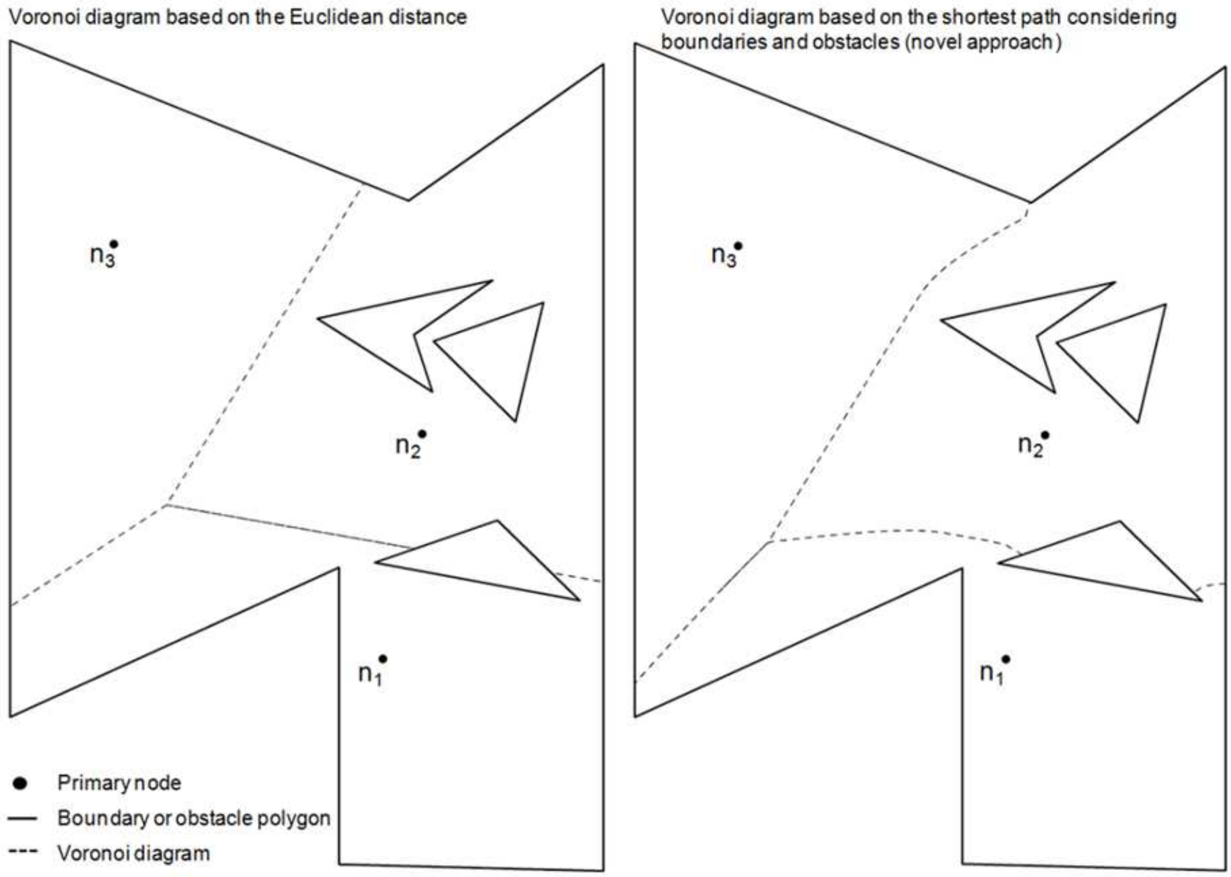

Voronoi diagram based on the Euclidean distance
Voronoi diagram based on the shortest path considering boundaries and obstacles (novel approach)
n3
n2
n1
n3
n2
n1
● Primary node
— Boundary or obstacle polygon
--- Voronoi diagram

Figure 8. Test case 1 with calculated Voronoi diagrams based on the Euclidean distance and the shortest path (novel approach)

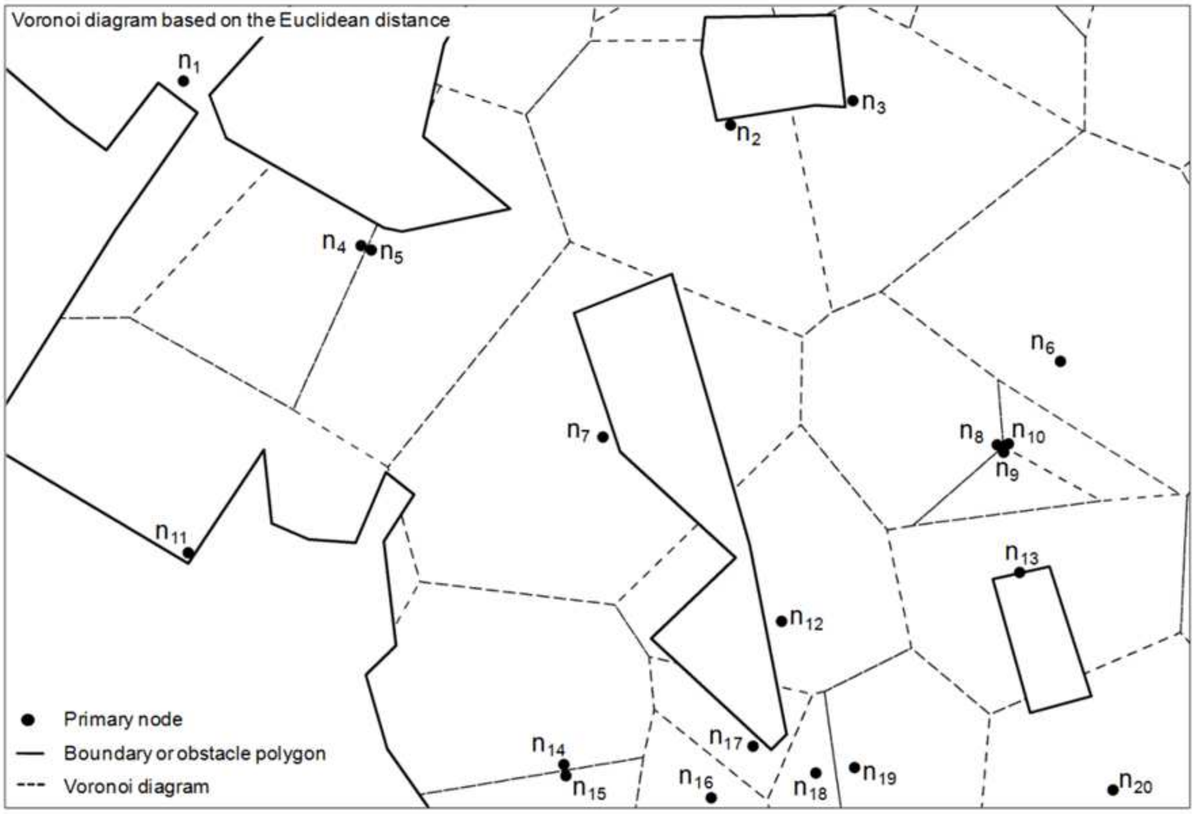
(between the two obstacles). Within the Euclidean Voronoi region of node n_3 no obstacles or non-convex sections of the bounding polygon are located which might have an influence on the bi-sectors if the Voronoi diagram is calculated based on the shortest path. Thus, it gains area of the Voronoi regions of both nodes n_1 and n_2 (8.3%).

The second case is a detail of an existing city in Algeria. It consists of 20 originating nodes (n_1, n_2, …, n_{20}), a non-convex boundary and several obstacles located within the boundary. Figure 9 shows the case and the calculated Voronoi diagrams based on the Euclidean distance and the shortest path. The individual sizes of the Voronoi regions and the differences are shown in table 2. In addition to the explanation given already for table 1, the area differences do not sum up to zero as only a small detail of the data set is shown.

Case 2 clearly shows, that the Voronoi diagram based on the Euclidean distance relates areas to nodes, which are closer to other nodes, if boundaries or obstacles are respected. This is the case, for example, for the nodes n_7, n_{11} and n_{12}. To node n_{11} the area to the right is assigned, which can only be reached by passing through the Voronoi region of node n_7. To the two nodes close to the obstacle in the centre, n_7 and n_{12}, areas on the opposite side of the obstacle are allocated. Contrarily, all Voronoi regions of the shortest path Voronoi diagram have a direct connection to the related originating node. As ascertained in table 2, the differences in the area sizes are up to 13.3% gain and -12,7% loss respectively.

Node	Voronoi region based on the Euclidean distance [length²]	Voronoi region based on the shortest path [length²]	Difference [length²]	Difference [%]
n_{1}	33,820.06	38,722.92	-4,902.86	-12.70
n_{2}	28,519.46	25,179.86	3,339.60	13.30
n_{3}	20,510.38	20,517.97	-7.59	0.00
n_{4}	14,185.69	13,991.04	194.65	1.40
n_{5}	19,486.64	20,731.79	-1,245.15	-6.00
n_{6}	36,329.44	35,075.23	1,254.21	3.60
n_{7}	t, the 51	33,492.01	-3,970.49	-11.90
n_{8}	15,188.57	14,350.52	838.05	5.80
n_{9}	3,656.39	3,656.57	-0.18	0.00
n_{10}	2,809.11	2,809.19	-0.08	0.00
n_{11}	23,676.38	25,289.44	-1,613.06	-6.40
n_{12}	16,706.28	16,185.26	521.03	3.20
n_{13}	16,916.03	17,686.10	-770.07	-4.40
n_{14}	21,487.65	20,796.47	691.18	3.30
n_{15}	7,085.54	7,085.79	-0.25	0.00
n_{16}	6,603.06	6,603.53	-0.46	0.00
n_{17}	3,779.35	3,831.44	-52.09	-1.40
n_{18}	3,156.40	2,834.48	321.92	11.40
n_{19}	11,720.82	11,494.35	226.47	2.00
n_{20}	23,474.75	22,949.54	525.21	2.30

Table 2. Comparison of the Voronoi region sizes of case 2



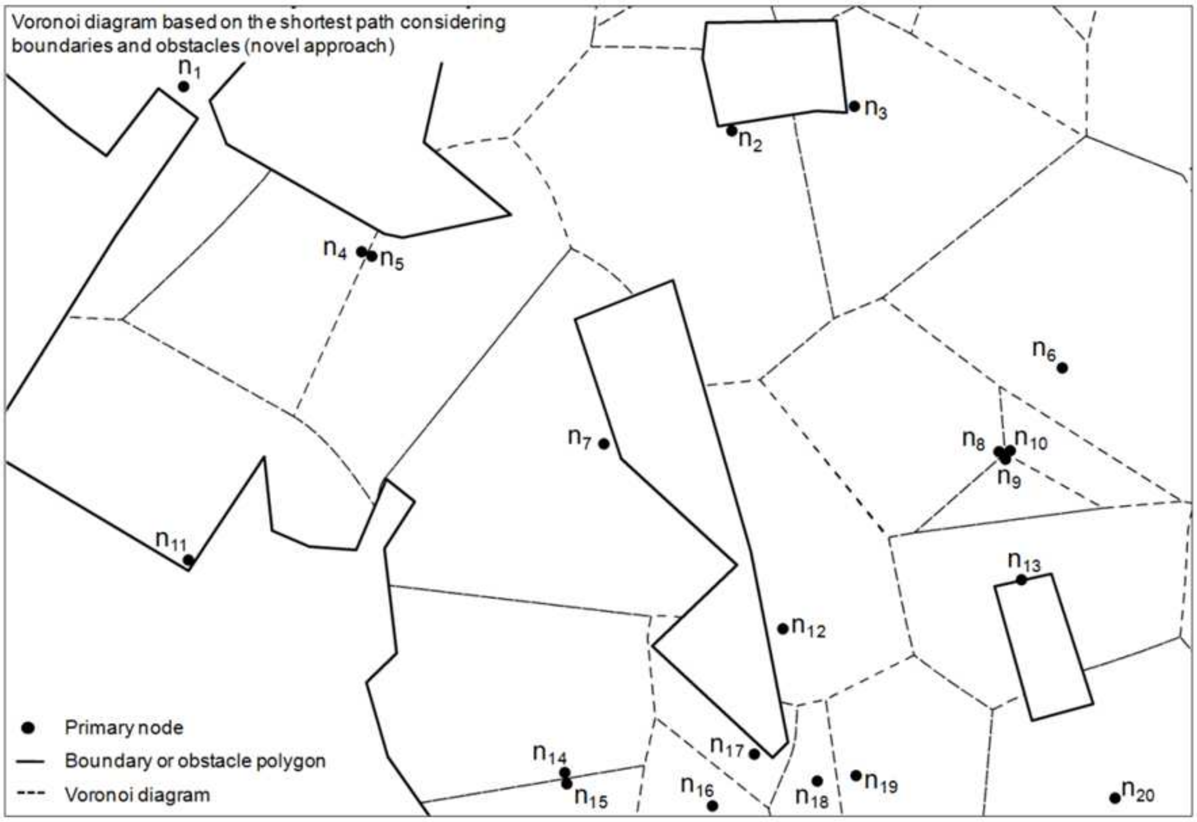

Voronoi diagram based on the shortest path considering boundaries and obstacles (novel approach)
n1
n2
n3
n4
n5
n6
n7
n8
n9
n10
n11
n12
n13
n14
n15
n16
n17
n18
n19
n20
● Primary node
— Boundary or obstacle polygon
--- Voronoi diagram

Figure 9. Test case 2 with calculated Voronoi diagrams based on the Euclidean distance and the shortest path (novel approach)

5. GIS-tool

Due to the spatial reference of the water distribution network data and the basic data for demand determination such like population density, population distribution, building areas etc. a GIS-application to determine and allocate water demand values to the network model nodes is advantageous. Following is a description of a GIS-tool implemented as an independent library of functions and integrated into ESRI ArcGIS (version 9).

ArcGIS, a product of the company ESRI, constitutes of several interlinked GIS products. The desktop application *ArcMap* supports visualising, analysing and editing of spatial data and linked attributes. The desktop application *ArcCatalog* enables data management. The data is commonly stored in relational data bases (*geodatabases*) but using single files (e.g. shape files) is possible too. Geometric objects (*features*) are modelled as vector data through defining shape and location of the object. Three principle *feature types* exist: *Point features* (punctual objects), *line features* (linear objects with start point, end point and vertices) and *polygon features* (areas defined by a polygon where the start and end points have the same location). Attributes (without spatial reference) can be linked to *features*. *Features* of the same type and with identical characteristics are stored in object classes (*feature classes*). All ArcGIS products are based on the library *ArcObjects*, developed as Microsoft Component Object Model (COM). *ArcObjects* offers numerous functionalities for the extension of ArcGIS products or the integration of applications into ArcGIS products. For further details on ArcGIS it is referred to the technical literature, c.f. e.g. Ormsby et al. (2004). The application for demand allocation described in this section is based on the ArcGIS data concept and the *ArcObjects* functionalities. The application was implemented as a dynamically linked library (DLL) which is integrated into *ArcMap* as a toolbar with an own user interface.

The demand allocation consists of two principle steps: (1) The calculation of the catchment areas of the nodes (Voronoi regions related to the originating nodes) and (2) the determination of the water demand within the catchment area and its allocation to the corresponding node.

For the calculation of the catchment areas (first step), input parameters are the demand nodes of the network model graph, the boundaries of the populated areas and the obstacles within the populated areas. The nodes have to be provided as *point features* in a *point feature class*, the boundaries and obstacles as *polygon features* in *polygon feature classes*. With the approach described in the previous sections the Voronoi diagram is calculated based on the nodes and stored in another *polygon feature class*. The references between Voronoi regions (catchment areas) and originating nodes are saved as an attribute of the nodes.

For the calculation of the demand per catchment area (second step) three options are developed and implemented considering the commonly available data basis: (A) Values and locations of the demands are given as *point features*, (B) areas with numbers of the population living there are given as *polygon features* or (C) building areas are given as *polygon features* and areas with the population are also given as *polygon features*. Moreover, option B and C need a constant demand per capita value as input parameter

Option A) The demand values [volume/time] are accumulated per catchment area by intersecting the catchment areas (*polygon features*) and the demands (*point features*) according to the spatial references. Finally, the accumulated demands per catchment area are allocated to the originating nodes.

Option B) The population per catchment area [inhabitants] is determined by intersecting the areas with constant values of the population (*polygon features*) and the catchment areas (*polygon features*) according to the spatial reference. To calculate the demand per node [volume/time], the population values [inhabitants] of the catchment area are multiplied with the demand per capita [volume/inhabitant/time].

Option C) The population per building area [inhabitants] is determined by intersecting the areas with constant values of the population (*polygon features*) and the building areas (*polygon features*) according to the spatial reference. The building areas, and thus the population, are accumulated per catchment area by intersecting the building areas (*polygon features*) and the catchment areas (*polygon features*) according to the spatial references. Finally, the demand per node [volume/time] is calculated by multiplying the population values of the catchment areas [inhabitants] and the demand per capita [volume/inhabitant/time].

Figure 10 shows an exemplary detail of the visualised result of a demand allocation with the GIS-tool. In the example, the nodes of the network model (black dots), a building area data set (dark green and orange areas) and a population density data set (transparent blue and transparent green areas) are given (as *point feature class* and *polygon feature classes*). Moreover, a satellite picture of the setting is used as basic layer. Based on the set of nodes and the bounding population density polygons, the Voronoi diagram was calculated applying the novel approach (*polygon feature class*). The Voronoi regions (*polygon features*) constitute the catchment areas of the nodes (red lines). Considering the given data, the demand is determined based on the population density and the building distribution (option C).

For each population density area (*polygon feature*) the number of inhabitants is calculated by multiplying the respective density and area. Next, the determined population is related to the building areas. Therefore, the total building area per population density area is determined by intersecting the two *polygon feature classes* according to their spatial reference. With the number of inhabitants and the total area of buildings per population density area the inhabitants per building area unit are calculated for each population density area. After accumulating the building areas per node according to the node catchment areas the number of population per node can be calculated. Finally, with the specific per capita demand, the demand per node is determined. Non domestic consumers with given demands (orange areas in the figure), such like industries or administrations, are excluded from the process described and allocated to the nodes directly.

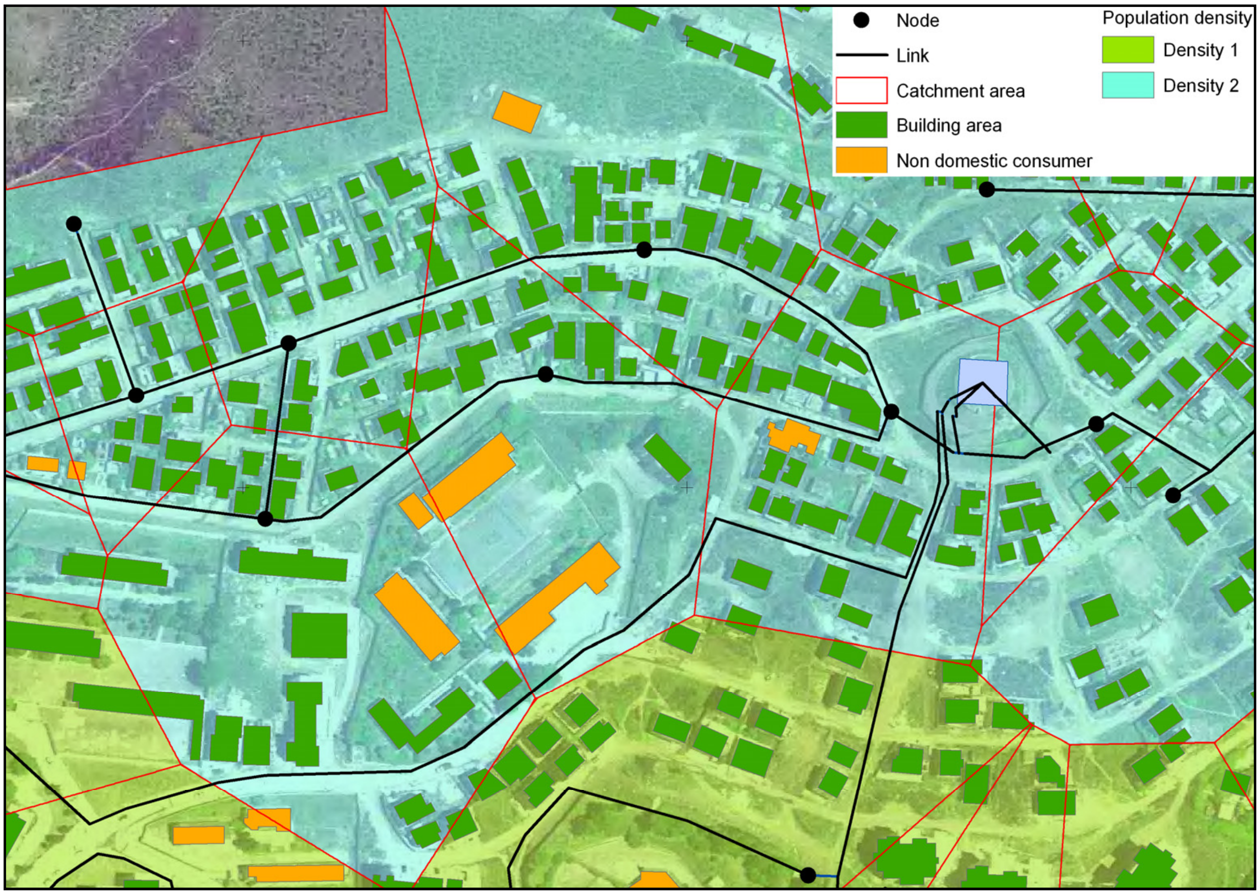


Figure 10. Exemplary solution of demand allocation with the developed GIS-tool (option c)

6. Conclusion

The need for a more precise demand allocation in order to increase the accuracy of water distribution network models led to the development of a novel approach to calculate Voronoi diagrams with constraints. The constraints considered are convex and non-convex boundary and obstacle polygons. Thus, the approach basically solves the problem of generating Voronoi diagrams in arbitrary environments.

By means of a geometric approach, namely successive circle intersection with discrete radii, the bi-sectors which form the Voronoi diagram are determined. Thus, the result is not a mathematically exact solution but an iterative approximation. Hereby, the accuracy can be controlled by the distances between the individual radii. For GIS software, the discretisation does not present a drawback as only polygons with discrete vertices can be stored.

At the moment, the algorithm is not comparable with other algorithms determining Voronoi diagrams based on the Euclidean distance (sweep-algorithm, divide-and-conquer-approach) regarding the computation speed. But the advantages of increased accuracy of the results of an application for demand allocation in water distribution network modelling (and possibly in other applications) outbalance the speed issue. Furthermore, with an enhanced radius controlling mechanism and a reduction of the circle combinations for which intersections are computed, there is still potential for improvements. Moreover, the overall speed could be increased by using highly parallel computations, since each circle combination can be worked at separately. This could be facilitated by means of GPU or multi CPU usage.

Another method to compute the shortest path Voronoi diagram following the described approach is the analytic representation of the bi-sectors. A bi-sector between just two nodes can be defined as a mathematical function. The major challenge is to identify those sections of the functions, which make up the bi-sectors of the total Voronoi diagram based on more than two nodes by intersecting the functions. Due to the mathematical properties the function intersections cannot be computed directly but only iteratively. Thus, in a mathematical sense, the approach is not exact as well. A benefit of the analytically described bi-sectors is the fact that only two points per function have to be calculated. To compute a Voronoi diagram polygon, the relevant vertices can be simply calculated with a free choice of density using the determined functions. Thus, the computation has no impact on the time needed for the actual determination of the mathematical description of the Voronoi diagram. An analytic solution is currently being developed by the authors.

Furthermore, a GIS-tool to allocate the water demand was developed and presented in section 5. The GIS-tool consists of two principle calculation steps. First, the catchment areas of the nodes are computed using the developed approach to calculate Voronoi diagrams. The second step consists of three options to determine the demand. The most precise result is achieved by applying option A due to the superior input data. In this case, values and location of demands are known and directly allocated to the nodes according to the Voronoi diagram. If such input data is not available, the distribution of water demand has to be determined first. Option B offers an approach to determine the water demand based on the population distribution, which is considered to be constant for defined areas. Option C follows the same approach but provides a higher accuracy by relating the population (and thus, the demand) to the building density instead of considering a constant population distribution for specific areas. The developed GIS-tool has been already successfully applied in several network modelling projects of the Institute for Water and River Basin Management of the Karlsruhe Institute of Technology and proofed to be a useful solution for demand allocation.

Finally, it is worth mentioning that an application of the presented approach is not limited to demand allocation in water distribution network modelling. A broad field of applications is conceivable, such as determination of rainfall catchment areas, modelling of growth processes or stochastic analysis.

Author details

Nicolai Guth and Philipp Klingel
Karlsruhe Institute of Technology (KIT), Institute for Water and River Basin Management, Germany

7. References

Aronov, B. (1987). On the geodesic Voronoi diagram of point sites in a simple polygon. *Proceedings of the third annual symposium on computational geometry, SCG '87*, pp. 39-49, Association for Computing Machinery, ISBN 0-89791-231-4, Waterloo, Ontario, Canada, 1987

Aurenhammer , F. (1991). Voronoi diagrams – A Survey of a Fundamental Geometric Data Structure. *ACM Computing Survey*, Vol.23, No.3, (September 1991), pp. 345-405, ISSN 0360-0300

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, Vol.1, No.1, (1959), pp. 269-271, ISSN 0029-599X

Fortune, S. (1986). A Sweepline Algorithm for Voronoi Diagrams. *Proceedings of the second annual symposium on computational geometry, SCG '86*, pp. 313-322, Association for Computing Machinery, ISBN 0-89791-194-6, Yorktown Heights, New York, USA, 1986

Guth, N. (2009). *Ermittlung der Einzugsflächen von Knoten mittels Voronoi-Diagrammen unter Berücksichtigung von Nebenbedingungen*, Bachelor thesis (unpublished), Hochschule Karlsruhe Technik und Wirtschaft, Karlsruhe, Germany

Johnson, W. A. & Mehl, R. F. (1939). Reaction kinetics in processes of nucleation growth. *Transactions of American Institute of Mining and Metallurgical and Petroleum Engineers*, Vol.135, (1939), pp. 416-458

Klingel, P. (2010). *Von intermittierender zu kontinuierlicher Wasserverteilung in Entwicklungsländern*, PhD thesis, Karlsruhe Institute of Technology, Department of Civil Engineering Geo and Environmental Sciences, Karlsruhe, Germany, Retrieved from <http://digbib.ubka.uni-karlsruhe.de/volltexte/1000019357>

Okabe, A.; Boots, B.; Sugihara, K. & Nok Chiu, S. (2000). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams* (2. edition), John Wiley & Sons, Ltd, ISBN 0-471-98635-6, Chichester, England

Ormsby, T.; Napoleon, E.; Burke, R.; Groess, C. & Feaster, L. (2004). *Getting to know ArcGIS Desktop: Basics of ArcView, ArcEditor and ArcInfo* (2. Edition), ESRI Press, ISBN 978-1589482104, Redlands, USA

Papadopoulou, E. & Lee, D. T. (1998). A new approach for the geodesic Voronoi diagram of points and simple polygon and other restricted polygonal domains. *Algothimica*, Vol.20, Nr.4, (1998), pp. 319-352, ISSN 0178-4617

Preparata, F. P. & Shamos, M. I. (1985). *Computational geometry: an introduction* (1. edition), Springer, ISBN 0-387-96131-3, New York, USA

Shamos, M. I. (1975). Geometric complexity. *Proceedings of seventh annual ACM symposium on Theory of computing, STOC '75*, pp. 224-233, Association of Computing Machinery, Albuquerque, New Mexico, USA, 1975

Walski, T. M.; Chase, D. V.; Savic, D. A.; Grayman, W.; Beckwith, S. & Koelle, E. (2003). *Advanced Water Distribution Modeling and Management*. 1, Haestad Press, ISBN 0-9714141-2-2, Waterbury, USA