CENTRALESUPELEC
2nd year of cursus centralien
Department of Mechanics

Computer aided design of a water supply network

AI & R Lab
Laboratorio del dioporco facile del Poliminchia

Relatore: Filippo Gatti
Correlatore: Andrea Barbarulo

Authors:
calvet-litzell Pere
Cognetti Giovanni
Marzorati Edoardo
Vaccarino Marcello

Academic year 2017-2018

In remembrance of Jean-Baptiste...

Abstract

Water supply remains a major issue in several countries. When designing a water supply network optimality is a priority. The aim of this project is to find optimal network structures using automation and machine learning. The development process is divided into several stages. During the first stage, network topology has been studied. A network has been designed using our software on real-world data.

Next stages will involve adding further parameters such as water velocity or pressure to the existing model. The project has a multidisciplinary nature. Using geographical data requires a certain level of acquaintance with different formats and software such as QGis. On the other hand, mastering a programming language like Python is required to implement the different algorithms and libraries.

Acknowledgements

Ringrazio ....

Capitolo 1

Introduction

L'introduzione deve essere atomica, quindi non deve contenere nè sottosezioni nè paragrafi nè altro. Il titolo, il sommario e l'introduzione devono sembrare delle scatole cinesi, nel senso che lette in quest'ordine devono progressivamente svelare informazioni sul contenuto per incatenare l'attenzione del lettore e indurlo a leggere l'opera fino in fondo. L'introduzione deve essere tripartita, non graficamente ma logicamente:

1.1 General context

La prima parte contiene una frase che spiega l'area generale dove si svolge il lavoro; una che spiega la sottoarea più specifica dove si svolge il lavoro e la terza, che dovrebbe cominciare con le seguenti parole l̈o scopo della tesi è . . .;̈ illustra l'obbiettivo del lavoro. Poi vi devono essere una o due frasi che contengano una breve spiegazione di cosa e come è stato fatto, delle attività sperimentali, dei risultati ottenuti con una valutazione e degli sviluppi futuri. La prima parte deve essere circa una facciata e mezza o due

1.2 Motivation and Objectives

1.3 Brief description of the approach

La seconda parte deve essere una esplosione della prima e deve quindi mostrare in maniera più esplicita l'area dove si svolge il lavoro, le fonti bibliografiche più importanti su cui si fonda il lavoro in maniera sintetica (una pagina) evidenziando i lavori in letteratura che presentano attinenza con il lavoro affrontato in modo da mostrare da dove e perché è sorta la tematica di studio. Poi si mostrano esplicitamente le realizzazioni, le direttive future di ricerca, quali sono i problemi aperti e quali quelli affrontati e si ripete lo scopo della tesi. Questa parte deve essere piena (ma non grondante come la sezione due) di citazioni bibliografiche e deve essere lunga circa 4 facciate.

1.4 Structure of the reporto

La terza parte contiene la descrizione della struttura della tesi ed è organizzata nel modo seguente. "La tesi è strutturata nel modo seguente.

Nella sezione due si mostra ...

Nella sez. tre si illustra ...

Nella sez. quattro si descrive ...

Nelle conclusioni si riassumono gli scopi, le valutazioni di questi e le prospettive future ...

Nell'appendice A si riporta ... (Dopo ogni sezione o appendice ci vuole un punto)."

I titoli delle sezioni da 2 a M-1 sono indicativi, ma bisogna cercare di mantenere un significato equipollente nel caso si vogliano cambiare. Queste sezioni possono contenere eventuali sottosezioni.

Capitolo 2

State of the art

Nella seconda sezione si riporta lo stato dell'arte del settore, un inquadramento dell'area di ricerca orientato a portare il lettore all'interno della problematica affrontata. Bisogna dimostrare di conoscere le cose fatte fino ad ora in questo campo e il perché si sia reso necessario lo svolgimento di questo lavoro. Questa sezione deve essere grondante di citazioni bibliografiche [**?**].

ï»¿

Capitolo 3

Approach

3.1 Root architecture

Biologically inspired models can provide interesting insights. Organisms that have gone through several rounds of evolutionary selection seem to be able to deliver efficient and nearly-optimal solutions. The use of such models seems to have produced satisfactory results for transport networks. Reading Chloé Arsonâs presentation on bio-inspired geomechanics, we discovered the potential advantage of using root system architecture to design water lines. Prof. Arson conducted an experiment to compare the predictions of a root growth model with real water line networks. Root growth is a gene-controlled phenomenon. Therefore, different species may present different growth patterns. In addition, soil structure has also an influence on root structures. For example the presence of physical obstacles, such as boulders, alters geotropic growth. Prof. Arson also pointed out that a rocky soil would require a different model. Other characteristics like water and nutrient gradients or bacteria play a key role in root growth. Prof. Arsonâs experiment consisted in growing roots on a scale plastic model of the Georgia Tech campus. The results would allow to validate the accuracy of the mathematical model. Afterwards they could be compared to the existing water network and thus assess its efficiency. Prof. Arson also introduced leaf venation systems which bear certain resemblance to water line networks. Indeed, the growth of a leaf is governed by the presence of auxin (plant hormones) sources which can be seen as the nutrient sources of the root model. We contacted Prof. Arson who gave us a very interesting bibliography on the subject of root growth models. Prof. Pierretâs article stresses the complex relationship between soil structure and soil biological activity. Soil is a habitat for many organisms and is also responsible for the movement and transport of resources which are necessary for their survival. Through their roots, plants play a key role in many soil processes. Soil properties affect root growth which in turn affects resource acquisition and therefore the plantâs impact on its environment (soil). Interest for root systems architecture comes from the necessity in agriculture of increasing productivity and minimizing water and nutrient losses. A good understanding of soil processes seems necessary to achieve this end. Moreover, Pierret points out that whereas soil biological and chemical processes have been carefully studied, physical processes need more attention. The article examines main biological factors that influence soil processes. It underlines the complex interactions between physical and chemical-biological processes and the impossibility to treat them separately. According to Pierret, roots are essential to study this complexity. In the second part of the article, the huge diversity of root classes is examined. This implies the necessity of using specific models for each species. The last part of the article discusses how modelling can provide clearer insights on the interactions between roots and soil. Lionel Dupuyâs article describes the evolution of root growth models. The first models appeared in the early 1970s and focused mainly on root length. However since the 1990s new complex models have emerged thanks to the use of more powerful computers. phenomenon has been fostered by the Â´Â´need for predictive technologies" at different scales. Dupuy suggests a new theoretical framework which takes into account individual root developmental parameters. He introduces "equations in discretized domains that deform as a result of growth". Simulations conducted by Dupuy have revealed some patterns in what seemed a complex and heterogeneous problem. More precisely, it seems that roots develop following travelling wave patterns of meristems. V. M. Dunbabin also mentions the progress accomplished in the area of root growth modelling. The early models did not take into account the root growth in response to a heterogeneous soil environment. Nowadays, models must include soil properties and accurate descriptions of plant function. The aim of these simulations is again to provide a better understanding of the efficient acquisition of water and nutrients by plants. Resource availability has a clear impact on both the roots and the stem of the plan. For example, a low nutrient concentration diminishes shoot growth and therefore leaf and stem mass fractions as well. It has been observed that roots respond locally to soil properties. This characteristic allows the plant to forage with more precision and reduce metabolic cost. Three-dimensional models are able to seize the complexity of the problem. Previous models were rather simple and relied upon one-dimensional functions of rooting depth vs. time. One of the most interesting articles is Atsushi Teroâs "Rules for Biologically Inspired Adaptive Network Design". In order to solve the problem of transport networks efficiency, Tero created a mathematical model based on organisms that build biological networks. He explains that these biological networks have been honed by many rounds of evolutionary selection and that they can provide inspiration to design new networks. He praises their good balance between cost, transport efficiency and, above all, fault tolerance. One of such organisms is physarum polycephalum, a type of slime mold. Tero let physarum grow on a map of the Tokyo area where major cities were marked by food sources. A first network was obtained. In order to improve the results, the experiment was carried out a second time. However, illumination was used to introduce the real geographical constraints such as coastlines or mountains $illuminationreducesphysarum\hat{a}sgrowth$. The results were very satisfactory and the biological network was very similar to the existing Tokyo transport network. Tero developed a mathematical model that tried to reproduce Physarumâs behavior. The principle of the model is that tube thickness depends on the internal flow of nutrients. Thus a high rate tends thickens a tube and a low rate leads to its decay. As shown by Prof. Arsonsâ paper "Bio-inspired fluid extraction model for reservoir rocks", slime mold growth can also be used to study the flow in a porous medium. The use of Root Architecture Models was abandoned in order to investigate the use of Machine Learning, more specifically, Artificial Neural Networks.

3.2 Artificial neural networks

The growth of network usage and their increasing complexity, in particular for communication technologies application, drives towards the improvement of routing technique. One track of this research is the development of "smart" techniques for network design and management.

For our project we chose to follow this direction, combine sub-optimal AI algorithms to develop a possibly innovative solution. Lead by example, we will give an overview of the most edge braking applications in this field. This will allow us to introduce the main concepts and get down to the techniques we focused on.

AI is applied to many complex routing problems: one example is very large-scale integration (VLSI). The process of designing integrated circuits is hard due to the large number of often conflicting factors that affect the routing quality such as minimum area, wire length. Rostam Joobbani, a knowledge-based routing expert from Carnegie-Mellon University (1986), proved that an AI approach to the subject could dramatically improve performances.

A more recent example is the use of AI in Wireless Sensor Networks. WSNs are spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to other locations. Management of those networks is particularly challenging because of the dynamic environmental conditions. J. Barbancho and al. (2007) wrote a review about the use of artificial intelligence techniques for WSNs for path discovery and other purposes. The study shows the potential of Artificial Neural Networks.

Artificial Neural Networks learn to do tasks by considering examples, generally without task-specific programming. An ANN is based on a collection of connected units called artificial neurons. Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it.

N. Ahad, J. Quadir, N. Ahsan (2016) published a review focused on techniques and applications of artificial neural networks for wireless networks. The advantage of using ANN is that can make the network adaptive and able to predict user demand.

Concerning shortest path problems Michael Turcanik (2012) used a Hopfield neural network as a content-addressable memory for routing table look-up. A routing table is a database that keeps track of paths in a network. Whenever a node needs to send data to another node on a network, it must first know where to send it. If the node cannot directly connect to the destination node, it has to send it via other nodes along a proper route to the destination node. Most nodes do not try to figure out which route(s) might work; instead, a node will send the message to a gateway in the local area network, which then decides how to route the ”package” of data to the correct destination. Each gateway will need to keep track of which way to deliver various packages of data, and for this it uses a Routing Table. Turcanik replaced the table with an ANN. His study shows the performance of routing table look-up in terms of speed and adaptability.

This excursus gives an idea of the incredibly various applications of ANN in routing problems. We would like now to focus on the use of Hopfield

Neural Network, which is the most classical solution for routing problems with ANN's.

Hopfield (1984) proposed the use of his algorithm to give heuristic solutions to the travel salesman problem. TSP is a well known NP-hard minimization problem. As defined by Karl Menger the TSP is "the task to find, for finitely many points whose pairwise distances are known, the shortest route connecting the points". So having n cities, our travel salesman has to associate to each city X a position k in the tour so that:

$$\sum_{X}\sum_{Y\neq X}\sum_{j} d_{XY}y_{Xj}(y_{Y,j+1}+y_{Y,j-1})$$

is minimal, where dXY is the distance between city X and Y.

E. Wacholder and al. (1989) developed a more efficient implementation of the Hopfield NN for the travel sales-man problem. The algorithm was successfully tested on many problems with up to 30 cities and five salesmen, while a non-optimized brute-force approach would take billions of billions of years to return. In all test cases, the algorithm always converged to valid solutions.

Mustafa K. Mehmet Ali and Faouzi Kamoun (1993) considered modeling shortest path problem with Hopfield Neural Network for the first time. The researchers asserted that HNN can find shortest path effectively and sometimes it would be better to use such a network instead of classic algorithms such as Dijkstra.

We will now explain what Hopfield Neural Networks are, with particular attention to the TSP application, although the definition we will give is general. It is a recurrent ANN, as opposed to feed forward NN, which means neurons interconnections forms a directed cycle, so neurons are both input and output. Hopfield nets are sets n2 nodes where X [1, n] and k [1, n] and the state is characterized by the binary activation values y = (yXj) of the nodes. A TSP problem with n cities can be modeled as an Hopfield net of dimension n2, where yXj is 1 if the city X is in the k-position of the tour.

The input sk(t+1) of the neuron k is:

$$s_k\,(t+1) = \sum_{j\neq k} y_i\,(t)\,w_{jk} + \theta_k$$

where wjk is the weight of the connection between j and k and is the bias The forward function is applied to the node input to obtain the new activation value at time t+1:

$$y_k\,(t) = sgn(s_k(t-1))$$

The energy function is as follow so that the optimal solution will minimize it:

$$E = \frac{A}{2}\sum_X\sum_j\sum_{k\neq j} y_{Xj}y_{Xk} + \frac{B}{2}\sum_j\sum_X\sum_{X\neq Y} y_{Xj}y_{Yj} + \frac{C}{2}(\sum_X\sum_j y_{Xj} - n)^2$$
$$+\frac{D}{2}\sum_X\sum_Y\sum_j d_{XY}y_{Xj}(y_{Y,j+1} + y_{Y,j-1})$$

The first two terms are null if and only if the there is a maximum of one active neuron for each row and column respectively. The third term is null if and only if there are n active neurons. The last term takes in account the distance of the path, that should be minimized as well.

The Hebbian rule to update the weights is deduced from the energy function:

$$w_{Xj,Yk} = \; -A\delta_{XY}\,(1-\delta_{jK}) - B\delta_{jk}\,(1-\delta_{XY}) - C - Dd_{XY}(\delta_{k,j+1} + \delta_{k,j-1})$$

where kj = 1 if j = k and zero otherwise. As in the energy function the first term inhibits connection within each row, the second within columns, the third is the global inhibition and the last term takes into account the distance between the cities.

Under the hypothesis $w_{Xj,Yk} = w_{Yk,Xj}$ the method can be proved to have stable points. At each iteration the net updates his parameters according to the Hebbian rule and the evolution of the state can be proved to be monotonically nonincreasing with respect of the energy function. Performing then a gradient descent, after a certain number of repetition the state converge to a stable point that is a minima of the energy function.

Artificial Neural Networks are not the best suited tool to solve routing problems. First of all, creating an ANN is a difficult task. Therefore, another method is worth considering. In addition, it is not possible to know whether the ANN will deliver a solution that will converge and whether this solution is optimal. Thus ANNs as a poorly efficient way to solve the problem.

Nonetheless, machine learning can still be useful. Indeed, the interpretation of geographical information files can be done through clustering algorithms.

Capitolo 4

Progetto logico della soluzione del problema

4.1 Problem staterment and modelisation

The overall idea of our software solution is to connect water sources and consumers to a network of pipes in the most efficent way possible. Indeed the definition of the best net is a key problem. Factors that makes an aqueduct the optimal one are not easily modeled. We can suppose that variables such length, height, water speed and pressure, viscosity ecc should be taken into account. For our first approach, we decided to consider only the pipeline length so that we simplify acqueduct design to a classical routing problem. On this base, we will then be able to add complexity. We can now more formaly define our poblem. Being a topography a graph representing the meshed surface of a region, the problem of designing an acqueduct as the one of finding the recovering-graph on the topograpy graph connecting water consumers and sources. We will use the euclidian metric on a space with tree dimensions. ======= ======= ¿¿¿¿¿¿¿ locmodmemt

4.2 Problem statement and modelisation

=======

4.3 Problem statement and modelisation

¿¿¿¿¿¿¿ 8dd11b4d40687fdbdea76ee797dfca1bc124b104 The overall idea of our software solution is to connect water sources and consumers to a network of pipes in the most efficient way possible. Indeed the definition of the best net is a key problem. Factors that makes an aqueduct the optimal one are not easily modelled. We can suppose that variables such length, height, water speed and pressure, viscosity etc should be taken into account. For our first approach, we decided to consider only the pipeline length so that we simplify aqueduct design to a classical routing problem. On this base, we will then be able to add complexity. We can now more formally define our problem. Being a topography a graph representing the meshed surface of a region, the problem of designing an aqueduct as the one of finding the recovering-graph on the topography graph connecting water consumers and sources. We will use the euclidian metric on a space with tree dimensions.

4.4 Data reading

In **??** is shown to the flow chart of our software. The input is composed of three geographical files: mesh represent the topology of the studied region, source-sinks contains the locations for each water source and consumer (called sink from now on). The last file describes the roads network in the area as the pipes should preferably run along roads for is cheaper to place them. For more information about the geographical data format, please refer to 5.1 Reading those input two graphs can be initialised: the topography and the aqueduct describing graphs. The first is a graph where nodes represent positions and edges the possible transitions between them. Transitions on roads will be preferable. The aqueduct graph is initialised with source and sinks as nodes and no edges. An insight of the data structure we used to represent graph is given in section 5.2

4.5 Clustering

Running classical algorithms such a brute-force TSP or a minimum spanning tree to link the nodes in the sink-source graph would not be feasible for computational reasons. Thanks to the particular nature of our problem a simplification is possible: we divide the aqueduct system in two layers: adduction and distribution nets. The adduction layer brings water from the source to the inhabited areas whereas the distribution segment is in charge of the "last kilometre" distribution. This two layer solution is commonly used in aqueduct design and network design in general: internet is an example. To achieve this need to recognise group of buildings such as villages or neighbourhoods. Those sets are called sink clusters. This approach caries multiple advantages. From the computational point of view reduce the dimension of the sets on which we run the routing algorithms. On the other side, once the two layers are identified we can use different strategies to connect the nodes, as we will se in the next paragraph. Efficient implementations of clusterings algorithms are provided in scikit-learn. Scikit-learn is a well-known machine learning library for Python and it features various classification, regression and clustering algorithms. After this operation sinks are labeled as part of they respective cluster. A more detailed explanation can be found at 5.3.

4.6 Routing

We now can design the water systems connecting the sinks. Letâs first consider the distribution layer, which is to say the problem of connecting the sinks of a cluster. This operation is broken down in two tasks. First, find all the paths connecting sinks, than chose the smallest recovering graph, i.e. the smallest aqueduct satisfying the specifics. To find the path connecting the sinks an optimal approach is used on the topography graph, in our case the Dijstrak algorithm. The length of those path and the traversed nodes are saved as attributes of edges of the aqueduct graph. Please pay attention to the fact that edges on the aqueduct graph are paths on the topography graph. This operation creates a complete graph for each cluster. Note that the set of those graph is a partition of the aqueduct graph. The second stage consists in eliminating the redundant edges: to do this we run the Kruskal algorithm and calculate the minimum spanning tree. For more information read section 5.3. An other approach, favoured in classical aqueducts design would be to calculate a partially connected graph where certain nodes are connected to exactly one other node whereas others are connected to two or more other nodes. This makes possible to have some redundancy without the expense and complexity required for a connection between every node in the network. At this first stage of the project this part is left for further investigation. Considering now the adduction system a very similar approach can be used: the clusters should be interlinked and connected to a source. Each distribution network, as is a tree, has a root node. We can initialise the adduction graph with all the cluster's roots nodes and the water sources. Finally this graph is connected with the technique previously used.

Capitolo 5

Technical aspects

5.1 Geographical Data

The overall idea is to take maps and automatically trace an aqueduct on it, in order to do that, we start from the map's shapefile. Shapefile is a popular geospatial vector data format for geographic information systems software. It spatially describes geometries: points, polylines and polygons. These, for example, could represent water wells, roads or buildings. As those primitive geometrical data types come without any attributes to specify what they represent, a table of records to store attributes is provided. Websites like osm2shp or Geofabrik provide an immense database of shapefiles available for download. Moreover desktop software like Qgis provides shapefile editing tools. This way we can both download real-world maps and create our own. Then through Qgis' meshing plug-in Gmsh we can mesh the surfaces of the map and export the result in vtk format as seen in Fig. 3.1 However, shapefiles seldom have information on the elevation (that is the Z coordinate) of the objects they represent. It is therefore necessary to use another format: the Digital Elevation Model (DEM). Digital Elevation Models provide this missing piece of information that can subsequently be added to the shapefile's attribute table. DEMs can be converted into meshes thanks to software such as SAGA. Meshes saved as vtk files can easily be used in Python. Indeed, vtk files are a simple and efficient way to describe mesh-like data structures. The vtk file boils down to those two elements: points and cells. Points have 3D coordinates while cells are surfaces, expressed by the points delimiting them. Point and cell data (scalar or vector) can also be assigned. We have therefore a file representing a graph, a classical mathematical model on which many operations can be performed: routing and clustering among others. We now come to our software. Python has been chosen as easy to use, widespread programming language, good for rapid prototyping and rich in package and libraries. The problem is divided in two main tasks: modelling the data structure that represents the graph and the algorithmic part, the aqueduct design.

5.2 Data Structure

To implement the data-structure we chose to use NetworkX. NetworkX is a Python package for the creation, manipulation, and study of complex networks. The package provides classes for graph objects, generators to create standard graphs, IO routines for reading in existing datasets, algorithms to analyze the resulting networks and some drawing tools. The software takes as input two shape files: the first describes the topology, the seconds the source and sinks. The topology is either a mesh, representing the geography of the region or a polyline with just the roads net of the region. The roads are particularly important because aqueducts are built along roads for logistical reasons. The second file is a polygon file containing the buildings that should be served by the aqueduct and the water sources. From these data, a first graph is obtained. The graph has as nodes the points described in topology file plus the buildings. The coordinates of bindings-representing nodes are the average of the coordinates of also have the metadata associated. The edges are the edges described in the topology file plus the edges connecting the building to the nearest node of the network in order to obtain a connected graph.

5.3 Clustering

In this section we will explain the different routing techniques used.
Dijstrak
TSP

Capitolo 6

Presentation and validation of experimental results

Si mostra il progetto dal punto di vista sperimentale, le cose materialmente realizzate. In questa sezione si mostrano le attività sperimentali svolte, si illustra il funzionamento del sistema (a grandi linee) e si spiegano i risultati ottenuti con la loro valutazione critica. Bisogna introdurre dati sulla complessità degli algoritmi e valutare l'efficienza del sistema.

Capitolo 7

Conclusions

7.1 Summary of Thesis Achievements

Si mostrano le prospettive future di ricerca nell'area dove si è svolto il lavoro. Talvolta questa sezione può essere l'ultima sottosezione della precedente. Nelle conclusioni si deve richiamare l'area, lo scopo della tesi, cosa è stato fatto,come si valuta quello che si è fatto e si enfatizzano le prospettive future per mostrare come andare avanti nell'area di studio.

7.2 Applications

7.3 Future wok

Appendice A

Documentazione del progetto logico

Documentazione del progetto logico dove si documenta il progetto logico del sistema e se è il caso si mostra la progettazione in grande del SW e dell'HW. Quest'appendice mostra l'architettura logica implementativa (nella Sezione 4 c'era la descrizione, qui ci vanno gli schemi a blocchi e i diagrammi).

Appendice B

Documentazione della programmazione

Documentazione della programmazione in piccolo dove si mostra la struttura ed eventualmente l'albero di Jackson.

Appendice C

Listato

Il listato (o solo parti rilevanti di questo, se risulta particolarmente esteso) con l'autodocumentazione relativa.

Appendice D

Il manuale utente

Manuale utente per l'utilizzo del sistema
color

Appendice E

Use case scenario

In this section will be presented a couple of use case scenarios, from the easier to the more complex ones.

E.1 Simple 3D routing

In this example a simple 3D routing is computed and rendered. Calling the command vesuvio$_{e}xample()inthepythonshellwillexecutethefollowinginstructions$
router = Router(topo$_{f}ile = ”vtk/Vesuvio”)router.route_{v}esuvio(32729, 31991)router.write2vtk(rout$
The topography of the Vesuvio and surrounding areas is loaded from a vtk file into a networkx graph data structure of the router python class. The shortest path is computed using the Dijstrak algorithm between two points. This path is then exported as a vtk file. Finally the two vtk, the topography and the path are rendered using the vtk library. The result is shown in figure

E.2 Clustering

E.3 Travel Salesman Problem

E.4 Minimum Spanning Tree

E.5 Automatic design

Executing the command paesi$_{e}xample()willreproducethecasestudiedin4.6.Thefollowinginstruction$

Appendice F

Datasheet

Eventuali Datasheet di riferimento.