Optimization Strategies in Water Resource Management

A. Candelieri, E. Messina, F. Archetti

Department of Computer Science, Systems and Communications, University of Milano-Bicocca, Italy
Outline

Sectorization of a Water Distribution Network (WDN) as IP problem

Simulation and Clustering for analytical localization of leaks (in Sectorized WDN)

Improving localization through regression techniques
Sectorization of a Water Distribution Network

Consists of defining “virtually” independent sections of the network, usually known as District Metering Areas (DMA), easier to be monitored and controlled.

Enables leakage assessment and detection through monitoring of hydraulic measures (e.g., Minimum Night Flow analysis)

Can be defined as a graph clustering task.

A water distribution network can be represented through a graph $G=\langle V, E \rangle$, where $V$ and $E$ are the nodes and pipelines sets respectively.

Each node $v_i$ have three properties:
- Piezometric level ($L_i$)
- Customer Demand ($D_i$)
- (Geographical) Position ($P_i$), with $P_i = (x_i, y_i)$

The $E$ set can be divided in two: $E_V$ and $E_N$, pipelines with and without a valve, respectively.

$$E = E_V \cup E_N \quad \text{and} \quad E_V \cap E_N = \emptyset$$
**Sectorization – Problem Definition**

**Problem**

Identifying sectors (clusters of nodes), given the available set of control valves installed, which

☑ minimize the number of connections (with valve) among different sectors and
☑ maximize internal homogeneity of sectors, in terms of level (elevation), demand and geographical distance among nodes.

We define the following integer decision variable

\[
x_{il} = \begin{cases} 
1 & \text{if } v_i \in C_l \\
0 & \text{if } v_i \notin C_l 
\end{cases}
\]

where \( l = 1, \ldots, k \) and \( C_l \) is the \( l \)-th cluster (sector)
Objective Function:
Maximizing internal \textit{homogeneity} while minimizing \textit{connectivity}
Each node must belong to one and only one cluster (sector)
Sectorization as an Integer Programming Problem

\[
\begin{align*}
\min f &= w_H \sum_{l=1}^{k} H_{etl} + w_C \sum_{l=1}^{k} C_{onn_l} \\
\sum_{l=1}^{k} x_{il} &= 1 \quad \forall i = 1, \ldots, n \\
\sum_{i=1}^{n} x_{il} &\geq 1 \quad \forall l = 1, \ldots, k
\end{align*}
\]

Each cluster must have at the least one node
Each cluster has to be connected to another one by at least one pipeline with valve
Sectorization as an Integer Programming Problem

\[
\begin{align*}
\min f &= w_H \sum_{l=1}^{k} H_{i_l} + w_C \sum_{l=1}^{k} C_{i_l} \\
\sum_{l=1}^{k} x_{i_l} &= 1 \quad \forall i = 1, \ldots, n \\
\sum_{i=1}^{n} x_{i_l} &\geq 1 \quad \forall l = 1, \ldots, k \\
\sum_{v_i \in C_l} \sum_{z=1}^{n} \sum_{v_j \in C_z} a_{ij}^V &\geq 1 \quad \forall C_l, z \neq l \\
\sum_{v_i \in C_l} \sum_{z=1}^{n} \sum_{v_j \in C_z} a_{ij}^N &= 0 \quad \forall C_l, z \neq l
\end{align*}
\]

Only pipelines with valve can connect nodes belonging to different clusters.
**Sectorization as an Integer Programming Problem**

\[
\begin{align*}
\min f &= w_H \sum_{l=1}^{k} \text{Het}_l + w_C \sum_{l=1}^{k} \text{Conn}_l \\
\sum_{l=1}^{k} x_{il} &= 1 \quad \forall i = 1, \ldots, n \\
\sum_{i=1}^{n} x_{il} &\geq 1 \quad \forall l = 1, \ldots, k \\
\sum_{v_i \in C_l} \sum_{z=1}^{k} \sum_{v_j \in C_z} a_{i,j}^V &\geq 1 \quad \forall C_l, z \neq l \\
\sum_{v_i \in C_l} \sum_{z=1}^{k} \sum_{v_j \in C_z} a_{i,j}^N &= 0 \quad \forall C_l, z \neq l \\
\forall C_l, \forall v_i, v_j \in C_l &\exists (v_i, u_1), (u_1, u_2), \ldots, (u_m, v_j) \text{ with } u_1, u_2, \ldots, u_m \in C_l,
\end{align*}
\]

The sub-graph induced by each cluster has to be connected.
Finally, parameters to be set by the user are:

- number of desired districts ($k>1$),
- relative relevance of homogeneity ($w_H$) and connectivity ($w_C$),
- relative relevance of level ($w_L$), demand ($w_D$) and geographical distance ($w_P$)
**Preliminary Simplification**

\[
\sum_{v_i \in C_l} \sum_{z=1}^k \sum_{v_j \in C_z} a_{ij}^N = 0 \quad \forall C_l, z \neq l
\]

This constraint can be removed!

**Aggregation in “natural” districts**

Nodes connected by a pipeline without valve **HAVE TO** belong to the same district.

Aggregating these nodes, a first “natural” sectorization can be identified ...

...however, number of the “natural” districts identified is generally higher then desired.
Solving Sectorization through Java and CPLEX (MIP)

∀C_l, ∀v_i, v_j ∈ C_l \exists (v_i, u_1), (u_1, u_2), \ldots, (u_m, v_j) \text{ with } u_1, u_2, \ldots, u_m ∈ C_l,

The **incumbent callback** in *Concert Technology* or the *Callable Library* can be used when the need to enforce more complex constraints on solutions exists (*incumbent callback as a filter*).

In our study we have used the incumbent callback to verify that each cluster induces a connected sub-graph.
Performance Evaluation

System properties:

- CPU: Intel Core i7-2820QM, 2.30 GHz
- RAM: 8 GB
- OS: Windows 7 (64 bit)
- CPLEX 12.4
- JVM: 1.6.0_26
- Eclipse: Indigo Release (20110615-0604)

Validation Schema:

- Number of nodes: 10, 11, 12, 13 and 14
- Three different levels of coverage for each network: 0.3, 0.6 and 0.9
- Networks creation: three different initialization for each pair (nodes; coverage)
- Performance index: computational time
- Internal sector homogeneity is ignored, analysis was aimed at estimating the impact of the network structure (number of nodes and coverage) on the performances
Real World Case Study

A water distribution network in a little town in Lombardia, Northern Italy (H2OLEak project, www.h2oleak.it)

- covered area 13 km²
- 6300 citizens (2600 users)
- level ranging from 107 m to 118.9 m
- 45 km of pipelines (931 pipelines)
- 200 valves
- **146** nodes after preliminary network simplification

Impossible to provide a sectorization in practical time!!!
(Estimated Time: more than 13500 years!!!!)
Constrained Agglomerative Clustering

A heuristic approach based on agglomerative clustering...

At each step the pair of adjacent clusters (linked by with valve) providing the best objective function value is aggregated *

On the same system, computational time is lower than 2 minutes for a single run!!!

Due to the heuristic nature of the approach, multiple configurations of parameters should be set and tested


Compared to the meta-heuristic approach recently proposed by Izquiero et al (2011)**, and based on multi-agents, our approach does not dependent on the number of sources into the network

An example of sectorization...

The real world case study: 4 districts, relevance is 0.5 both for homogeneity and connectivity.
Analytical Localization of leaks
Analytical Localization of Leaks

1) Build a dataset through a leakage scenarios simulation (EPANET)
2) Group scenarios according to the “effect” of each simulated leak and save centroids
3) When a possible leakage is detected, compare real situation with centroids
4) Plan physical check on pipelines belonging to the clusters of the most similar centroid

Algorithms: Farthest First, Agglomerative, Induced Bisecting, In-deep Bisecting

Clustering fitness evaluation:

- **Highly** Localizing clusters - having less than 25% of pipelines of the network;
- **Average** Localizing clusters - having between 25% and 50% of pipelines;
- **Poorly** Localizing clusters - having between 50% and 75% of pipelines;
- **No** Localizing clusters - having more than 75% of pipelines.

AIRO 2012, Vietri, Italy
Results

Highly Localizing Clusters

No Localizing Clusters

Average Localizing Clusters

Poorly Localizing Clusters

AIRO 2012, Vietri, Italy
**Improving localization through Regression**

Furthermore, a Regression Model to estimate discharge coefficient of leaks, in order to improve the efficiency of consequent physical localization

The Regression Model works in parallel to clustering - on the same features set. After the most representative centroid is identified, pipelines belonging to correspondent cluster are ranked/selected by comparing simulated and regressed discharge coefficients to identify the most probably affected pipeline(s)

**Validated through 10-fold cross validation**
- correlation coefficient = 0.9997
- Mean Absolute Error = $10^{-4}$
- Root Mean Squared Error = $2*10^{-4}$
- Relative-Mean Absolute Error = 0.8764%
- Root Relative Mean Squared Error = 2.5368%.
Conclusions

Sectorization of a Water Distribution Network (WDN) as IP problem

→ **Proposed a heuristic approach for supporting administrators of WDN**

Simulation and Clustering for analytical localization of leaks (in Sectorized WDN)

→ **Developed an approach to reduce time and cost for physical check and rehabilitation**

Improving localization through regression techniques

→ **Identified a regression method to improve physical check efficiency**
Thanks for your attention
Solving Sectorization through Java and CPLEX (MIP)

The **incumbent callback** in *Concert Technology* or the *Callable Library* can be used when the need to enforce more complex constraints on solutions exists (*incumbent callback as a filter*).

During the populate procedure, the incumbent callback is called each time a new solution is found, even if the new solution does not improve the objective value of the incumbent.

The incumbent callback allows applications to accept or reject the new solution based on specific criteria.

The incumbent callback has been used to verify the connectivity constraint (4) for each identified sector (i.e., node cluster).
When an integer solution is found: the incumbent
After CPLEX® finds an integer solution, it does the following:
♦ It makes that integer solution the incumbent solution and that node the *incumbent node*.
♦ It makes the value of the objective function at that node (modified by the objective difference parameter) the new cutoff value.
♦ It prunes from the tree all subproblems for which the value of the objective function is no better than the incumbent.

What are filters of the solution pool?
*Filtering allows you to control properties of the solutions generated and stored in the solution pool.* CPLEX® provides two predefined ways to filter solutions.
♦ If you want to filter solutions based on their difference as compared to a reference solution, use a diversity filter, as explained in Diversity filters.
♦ If you want to filter solutions based on their validity in an additional linear constraint, use a range filter, as explained in Range filters.
Those two ways are practical for most purposes. However, if you require finer control of which solutions to keep and which to eliminate, use an incumbent callback, as explained in Incumbent callback as a filter.

Incumbent callback as a filter
If you need to enforce more complex constraints on solutions (if you need to enforce nonlinear constraints, for example), you can use the incumbent callback in Concert Technology or the Callable Library. During the populate procedure, the incumbent callback is called each time a new solution is found, even if the new solution does not improve the objective value of the incumbent. The incumbent callback allows your application to accept or reject the new solution based on your own criteria.
Number of pipelines with valve linking different districts, depending on the relevance of homogeneity for a 4 and 10 districts sectorization, separately
Induced vs In-deep Bisecting

Clusters generation in Induced Bisecting

Clusters generation in In-deep Bisecting
Clustering Results

$k=50$

$k=70$

$k=100$

$k=150$
Least Median Squared Linear Regression proved to be effective enough to predict discharge coefficient of the leak depending on pressure and flow variations.

**Regression Model**

\[ dc = -0.0123*ΔP1 + 0.0025*ΔP2 - 0.0026*ΔP3 + 0.0023*ΔP4 + 0.008*ΔP5 - 0.081*ΔF \]

where \( ΔPi \) is the pressure variation at the \( i-th \) monitoring node and \( ΔF \) is the flow variation at the pumping system, with respect to the faultless network.

**Validated through 10-fold cross validation**

- correlation coefficient = 0.9997
- Mean Absolute Error = \( 10^{-4} \)
- Root Mean Squared Error = \( 2*10^{-4} \)
- Relative-Mean Absolute Error = 0.8764%
- Root Relative Mean Squared Error = 2.5368%.