Agent based coordination for Smart Grid and Microgrid – Distributed Techniques in Power System Operation

Iasonas Kouveliotis-Lysikatos
Despina Koukoula
Nikos Hatziargyriou
Sensor networks

- A sensor network (smart meters ?) is comprised of a large number of “cheap” devices, in place for measuring, monitoring and event detecting purposes.

- Peer nodes are connected via existing ICT/power networks.
  - Advantages are not measured by the efficient resource exploitation.
  - Significant advantage the very low requirements for infrastructure investment.
Overlay ICT networks
Intelligent Software Agent

- **Agent** = “somebody or something that acts by performing a role or creates a specific result”
- **Characteristics**: autonomy, reaction, feedback, rational, mobility, communication ability
- **Multi-agent Systems**: coordination of distributed computational resources for the solution of a complex problem
Agents’ Concerns

To whom should I talk?

What kind of information should I send?

What should I do with the received data?
Load curtailment after a MV/LV substation:

- 20-30 kW
- 10 kW
- 8 kW
- 4 kW
- 14 kW

Sum: 36 kW
Avg: 9 kW

- 14 kW – 7 kW
- 13 kW – 6.5 kW

Sum: 30 kW
Avg: 7.5 kW
JADE

- Distributed agent platform.
- Graphical user interface
- Support to the execution of multiple, parallel and concurrent agent activities via the behaviour
- FIPA-compliant Agent Platform
The JADE Agent class

- Multitask computational model
- Tasks (or behaviours) are executed concurrently
- A scheduler, internal to the base Agent class and hidden to the programmer, automatically manages the scheduling of behaviours.
- Exchanges messages with other agents
Jade Behaviors
Distributed MAS architecture in JADE
DISTRIBUTED ALGORITHMS
Optimization of the Power System Operation

- Problems that can be addressed:
  - Economic Dispatch
  - Demand Side Management
  - Voltage control
  - Congestion management
  - Optimal Power Flow
- Formulation as a Resource Allocation Problem
- Distributed Optimization Techniques
Modeling of the Grid Components

- Power Grid
  - Graph representation of the network
- Controllable components of the grid
  - Controllable loads
  - Controllable DGs
  - Market-based approach
- Communication architecture
Communication Architectures

Centralized or Distributed Architecture?

- Robustness
- Reduced Communication cost
- Privacy
- Parallel computation
- Tolerance in communication delays
- Scalability - Extensibility – “Plug-and-play”
Distributed optimization techniques

• Distributed algorithms
  • Population Dynamics
  • Consensus algorithms
  • Synchronized oscillators
  • Distributed gradient
  • Additive Increase Multiplicative Decrease (AIMD)
• Interplay of control theory, distributed optimization, dynamical systems, graph theory and algebraic topology

Goal: Form a collective task or achieve a global behavior using local interactions!
Centralized Problem Solving

- Gathering of the information in a single node
- Calculation of the solution
- Communication of the solution to all other nodes
In an iteration of a distributed algorithm in most cases every node performs the following steps:

- Communication with neighbors
- Update the values of the their variables

The nodes iterate until convergence is detected.
Population Dynamics

Births and immigration add individuals to a population.

Deaths and emigration remove individuals from a population.

[http://bio1152.nicerweb.com/]
Population Dynamics

- Populations of Individuals living in different habitats
- Individuals interact with their environment
- Their welfare is modeled with a “fitness” function: 
  \[
  \begin{align*}
  \text{fitness}(x) &= \text{function} \\
  \end{align*}
  \]
Eventually, an equilibrium can be reached when the fitness of all nodes is equal to the average fitness.

Is this algorithm distributed?
Local Information Constraint

With the local replicator equation, nodes exchange only information about their nearest habitats and decisions are made over neighborhoods to yield the same payoff across the environment.
Economic Dispatch

- Minimization of quadratic costs of the generators:
  \[
  (\quad) \quad (\quad)
  \]

  Subject to the constraint:
  - the total load of the system,
  - production of generator unit \(i\), \( (\quad) \) its cost and
  - the set containing all the generators of the system

- Technical constraints:

- Economic Dispatch Criterion:
  \[
  (\quad) \quad (\quad) \quad (\quad)
  \]
Application in Economic Dispatch
Application in Economic Dispatch

\[
\begin{align*}
\text{Constraint for} & \quad [ \quad ] \\
& \quad \{ \quad \} \\
& \quad ( \quad )
\end{align*}
\]

- Constraint for \( [ \quad ] \)
- Fitness function: \( ( \quad ) \) \( ( \quad ) \)
Congestion management

Congestion management in radial distribution networks

Steps for coping with the downstream congestion:

- Detection of a congested line
- Split graph into two subgraphs
- Balance the excessive power that causes the congestion over the two subgraphs
Congestion management

Congestion management in radial distribution networks

Same steps as in the previous case for upstream congestions
Economic Dispatch in Distribution System

<table>
<thead>
<tr>
<th>Node</th>
<th>(kW)</th>
<th>(kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>1.1</td>
<td>900</td>
</tr>
<tr>
<td>3</td>
<td>1.2</td>
<td>1500</td>
</tr>
<tr>
<td>4</td>
<td>1.9</td>
<td>800</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>900</td>
</tr>
</tbody>
</table>
Economic Dispatch in Distribution System
Economic Dispatch in Distribution System
Economic Dispatch in Distribution System
Application (Population Dynamics)
Application (Population Dynamics)
Application (Population Dynamics)
Simple Consensus protocol

**Model:**

\[
\dot{x}_i(t) = \sum_{j \in N_i} a_{ij} (x_j(t) - x_i(t))
\]

\[
\begin{align*}
\lambda_i(0) &= \text{any fixed admissible value} \\
x_i(0) &= \text{any fixed admissible value} \\
y_i(0) &= \begin{cases} \\
d\frac{D}{|N_0^-|} - x_i(0) & \text{if } i \in N_0^- \\
-x_i(0) & \text{otherwise} \\
\end{cases}, \forall i \in V.
\end{align*}
\]

**Initialization:**

**Iteration:**

\[
\lambda_i(k+1) = \sum_{j \in N_i^+} p_{i,j} \lambda_j(k) + \epsilon y_i(k)
\]

\[
x_i(k+1) = \beta_i \lambda_i(k+1) + \alpha_i
\]

\[
y_i(k+1) = \sum_{j \in N_i^+} q_{i,j} y_j(k) - \left(x_i(k+1) - x_i(k)\right)
\]
Applications (Consensus)
Distributed OPF

- Minimization of active power losses (or in combination with generators cost)
- The **ADMM** (Alternating Direction Method of Multipliers) algorithm can be used in combination with a convexification technique
- The non-convex OPF problem can be convexified with:
  - Semidefinite programming (SDP) relaxation technique
  - Second-order cone (SOC) relaxation.
GOSSIP ALGORITHM
Gossiping Background

- Consensus algorithms
- Epidemic algorithms for database replication
Gossip Algorithm properties

Gossip algorithms include local processes at every grid node, so as:

- Utilize only local info from their neighborhood
- Perform at most $O(\log n)$ calculation steps within a finite time unit
- The algorithm requires $O(\text{poly}(\log n) + |F_i|)$ memory at every node assuming $|F_i|$ as the internal memory need at every node to create its own output
- There is no need for synchronization
How Gossip Algorithm Characteristics are related to distribution grids

- Large scale applications ➔ scalability
- Dispersed solution to locally caused problems, near to the trigger point of the problem ➔ no need for central coordination

- Gossip algorithms are a very concrete basis for the decentralized decision making procedure and the distributed problem solution
### Generic Gossip algorithm

<table>
<thead>
<tr>
<th>Active thread</th>
<th>Passive thread</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>do once</strong> for each $T$ time units at a random time</td>
<td><strong>do forever</strong></td>
</tr>
<tr>
<td><strong>begin</strong></td>
<td><strong>begin</strong></td>
</tr>
<tr>
<td>$p = \text{SelectPeer}()$</td>
<td><strong>receive</strong> $\text{info}_p$ from $p$</td>
</tr>
<tr>
<td><strong>send</strong> $\text{DataExchange (state)}$ to $p$</td>
<td><strong>send</strong> $\text{DataExchange (state)}$ to $q$</td>
</tr>
<tr>
<td><strong>receive</strong> $\text{info}_p$ from $p$</td>
<td>**state} = \text{DataProcesing}(\text{info}_q)$</td>
</tr>
<tr>
<td>$\text{state} = \text{DataProcesing}(\text{info}_p)$</td>
<td><strong>End</strong></td>
</tr>
</tbody>
</table>
Graph basis

- Incidence matrix $A$ ... depicted on weights’ matrix $W$
- Laplace matrix $L = D - A$

$G = (V, E)$
$N_i \subseteq V$

<table>
<thead>
<tr>
<th>Node $i$</th>
<th>Neighbours $N_i$</th>
<th>Degree $d_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{2}</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>{1,3,5}</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>{2,4,6,7}</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>{3}</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>{2}</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>{3}</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>{3,8,9}</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>{7}</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>{7}</td>
<td>1</td>
</tr>
</tbody>
</table>
Distribution Grids as Graphs overlay ICT networks
**Eigenvectors & eigenvalues**

- Let $A$ be a $n \times n$ matrix with real elements. The real or complex number $\lambda$ constitutes an eigenvalue of matrix $A$ if there is a non-zero vector $v$ with real or complex elements such that $A \cdot v = \lambda \cdot v$.
- The non-zero vector $v$ is called an eigenvector of matrix $A$ corresponding to the eigenvalue $\lambda$.
- Eigenvectors and eigenvalues are very important tools for the spectral analysis in graph theory and are related to the incidence matrix $A$ and the Laplace matrix of the graph.
- A non directional graph has a symmetric matrix $A$ and therefore the real eigenvalues are measuring the graph spectrum.
- The spectral radius of a symmetric matrix is an infinite linear operator, deriving from the comparison of the absolute values of the members of the graph spectrum. It is the upper limit of the spectrum and is denoted as $\rho(\cdot)$.
- Assuming that $\lambda_1, ..., \lambda_n$ are the eigenvalues of the matrix $A \in \mathbb{C}^{n \times n}$, $\rho(A)$ is defined as $\max\{|\lambda_1|, |\lambda_n|\}$.
Eigenvectors & eigenvalues

- Gossip - An applied transfer of a linear function over a linear mapping in a distributed manner
- The dominant eigenvector is used as a measurement of the centrality of graph edges (Google page rank -> eigenvector of A of www graph has the ranking of webpages as its elements)
- The centrality of eigenvectors are a measure of the nodal effect on the grid.
- Relevant grades are associated with the grid nodes, so as nodes with greater grades have a greater impact on the grid than nodes with lower grades
Basic idea behind Gossiping

- Let every node having a local variable $x_i$ at its disposal
- We need to calculate a linear function over the local variables of the nodes, e.g.

\[ W \in \mathcal{L}, \mathcal{L} = \{ W \in \mathbb{R}^{n \times n} \mid W_{ij} = 0, \forall j \notin \mathcal{E} \text{ and } i \neq j \} \]
Convergence acceleration

\[ \lim_{t \to \infty} W_t = \frac{1^T \cdot 1}{n} \]

\[ 1^T \cdot W = 1^T \]

\[ W \cdot 1 = 1 \]

minimize \( \rho(W - 11^T/n) \)

subject to \( W \in \mathcal{S}, \quad 1^T W = 1^T, \quad W 1 = 1 \)
Gossip algorithm for calculation of summation

<table>
<thead>
<tr>
<th>Active thread</th>
<th>Passive thread</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>do once</strong> for each T time units at a random time</td>
<td><strong>do forever</strong> begin</td>
</tr>
<tr>
<td><strong>begin</strong></td>
<td><strong>receive</strong> info(_p) from (p)</td>
</tr>
<tr>
<td>(p = \text{SelectPeer}())</td>
<td><strong>send</strong> DataExchange (state) to (q)</td>
</tr>
<tr>
<td><strong>send</strong> DataExchange (state) to (p)</td>
<td><strong>state</strong> = DataProcessing(info(_q))</td>
</tr>
<tr>
<td><strong>receive</strong> info(_p) from (p)</td>
<td><strong>End</strong></td>
</tr>
<tr>
<td><strong>state</strong> = DataProcessing(info(_p))</td>
<td></td>
</tr>
<tr>
<td><strong>end</strong></td>
<td></td>
</tr>
</tbody>
</table>

**SelectPeer()**

\[ j \in N_i \]

**DataProcessing**

\[ P_i^{t+1} = W_{ii} \cdot P_i^t + \sum_{j \in N_i} W_{ij} \cdot P_j^t \]
Gossip Algorithm for calculation of the summation

1: procedure GOSSIP($x_i, w_{ii}, w_{ij} \in W: j \in N_i$, threshold)
2:    round $r=0$
3:    $x_{AV,r}^{est,i} \leftarrow x_{i,0}$
4:    sends to all nodes $j \in N_i$ the value $x_{AV,r}^{est,i}$
5:    error$\leftarrow 1$
6:    round $r>0$
7:    while error$>$threshold
8:       receives from all nodes $j \in N_i$ their value $x_{AV,r}^{est,j}$
9:       calculates $x_{AV,r+1}^{est,i} = w_{ii} \cdot x_{AV,r}^{est,i} + \sum_{j \in N_i} w_{ij} \cdot x_{AV,r}^{est,j}$
10:      error$=|x_{AVG,r+1}^{est,i} - x_{AVG,r}^{est,i}|$
11:      sends to all nodes $j \in N_i$ the value $x_{AV,r+1}^{est,i}$
12:     end while
13:    return $x_{AV,r+1}^{est,i}$
14: end procedure
Selection of weights’ matrix for convergence acceleration

Initialization with the best constant weights based on the Laplace matrix eigenvalues $a^* = \frac{2}{\lambda_1(L) + \lambda_{n-1}(L)}$

$$a^* = \frac{2}{\lambda_1(L) + \lambda_{n-1}(L)}$$

$$W_{ij} = \begin{cases} 
\alpha^* , & \text{for } \{i,j\} \in \mathcal{E} \\
1 - d_i \cdot \alpha^* , & \text{for } i = j \\
0 & \text{else}
\end{cases}$$

Optimization via subgradient methods

Optimization techniques for semidefinite problems
Algorithm for minimum calculation

1: round \( r = 0 \)

2: \( x_{\text{MIN},r}^{\text{est},i} \leftarrow x_{i,0} \)

3: sends to all nodes \( j \in N_i \) the value \( x_{\text{MIN},r}^{\text{est},i} \)

4: round \( r > 0 \)

5: receives from all nodes \( j \in N_i \) their value \( s x_{\text{MIN},r}^{\text{est},j} \)

6: calculates \( x_{\text{MIN},r+1}^{\text{est},i} = \min \{ x_{\text{MIN},r}^{\text{est},i}, x_{\text{MIN},r}^{\text{est},j} \} \)

7: if \( x_{\text{MIN},r+1}^{\text{est},i} < x_{\text{MIN},r}^{\text{est},i} \) then

8: sends to all nodes \( j \in N_i \) the value \( x_{\text{AV},r+1}^{\text{est},i} \)

9: end if
Distributed resource allocation

minimize \quad f(x) \triangleq \sum_{i=1}^{n} f_i(x_i)

subject to \quad \sum_{i=1}^{n} x_i = c,

Strictly convex functions fi

x_i(t + 1) = x_i(t) - W_{ii} f_i'(x_i(t)) - \sum_{j \in \mathcal{N}_i} W_{ij} f_j'(x_j(t)), \quad i = 1, \ldots, n,
Applications on grid management

- Constrained Optimal power flow - Congestion
- Resource allocation – Address RES variability with flexible loads
  - Load shedding – utility function
  - Fairness index – local load balancing in computers
- Mitigation of Voltage violations via sensitivity analysis and linearization of power flow equations

\[
V_i = V_{feeder} - \sum_{j=1}^{n} z_{ij} \cdot \frac{P_j - i \cdot Q_j}{(V_j^{to})^*}
\]

\[
z_{ij} = \begin{cases} 
\sum_{l \in \text{path}(i)} z_b(l), & \text{for } i = j \\
\sum_{l \in \text{path}(i) \cap \text{path}(j)} z_b(l), & \text{for } i \neq j 
\end{cases}
\]
Example of application

Minimize \( F = \sum_{i=0}^{n} u_i(P_i) \)

subject to \( S_{TOT}^{e_1} \leq S_{max} \)

\( P_{min} \leq P_i^{e_1} \leq P_i^{e_0} \)

\[
\sum_{i=0}^{n} (S_i^{e_1})^* = S_{max}
\]

\[
\frac{d u_i}{d P_i^*} = \lambda^*
\]

<table>
<thead>
<tr>
<th>Node</th>
<th>( P_{max} ) (p.u.)</th>
<th>( P_{t=0} ) (p.u.)</th>
<th>Utility Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>0.300</td>
<td>0.100</td>
<td>3.57</td>
</tr>
<tr>
<td>3</td>
<td>0.302</td>
<td>0.300</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>0.192</td>
<td>0.030</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>0.185</td>
<td>0.150</td>
<td>3.74</td>
</tr>
<tr>
<td>6</td>
<td>0.353</td>
<td>0.020</td>
<td>3.61</td>
</tr>
<tr>
<td>7</td>
<td>0.257</td>
<td>0.000</td>
<td>3.33</td>
</tr>
<tr>
<td>8</td>
<td>0.330</td>
<td>0.300</td>
<td>3.85</td>
</tr>
<tr>
<td>9</td>
<td>0.223</td>
<td>0.100</td>
<td>3.48</td>
</tr>
</tbody>
</table>
Utility piece-wise linear function
Gossiping for summation
Resource re-allocation
Summations

Gossip Rounds

Total Apparent Power (10^{-1} p.u.)

node1, node2, node3, node4, node5, node6, node7, node8, node9

S_{t=0}, S_{\text{max}}
Initialization for $V_5$

For the calculation of $V_5$, every node $j$ prepares itself to participate in the gossiping for $V_5$ by selecting the appropriate $z_{5j}$ and calculating the $x_{j,r=0} = z_{5j} \cdot \frac{P_{j-i}Q_j}{(V_j^{t_0})^r}$, assuming $V_j^{t_0} = 1$ p. u.

$$x_{j,r=0} = 10^{-3}$$

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0.2596 + 0.1281i</td>
<td>0.6813 + 0.8015i</td>
<td>0.2902 - 0.0628i</td>
</tr>
<tr>
<td></td>
<td>0.1299 + 0.0039i</td>
<td>0</td>
<td>0.8566 + 0.6886i</td>
<td>0.2466 + 0.2546i</td>
</tr>
</tbody>
</table>
Gossiping for voltage calculation at bus 8
We are not gossiping, we are net-working!
Applications using Multi-Agent Systems

- The JADE platform uses text files to initialize and define the:
  - Network topology
  - Economical and technical parameters
Applications using Multi-Agent Systems

- Exchanges peer-to-peer messages to perform the optimizations:

- Basic GUI to present results:
Applications using Multi-Agent Systems
Application in MELTEMI Community Smart Grids pilot site

- Controllers in each house
- Peer-to-peer communication between the controllers utilizing the local LAN
- Active power curtailment by controlling the household appliances
Application in MELTEMI

- Controllers
Thank you!

Iasonas Kouveliotis-Lysikatos
Phd student
Power Systems Lab
School of Electrical and Computer Engineering
National Technical University of Athens
e-mail: [iasonask@mail.ntua.gr]

Despina Koukoula
Phd student
Power Systems Lab
School of Electrical and Computer Engineering
National Technical University of Athens
e-mail: [kdespina@power.ece.ntua.gr]

http://www.smartrue.gr/