Distributed routing algorithms to manage power flow in agent-based active distribution network

Citation for published version (APA):
https://doi.org/10.1109/ISGTEUROPE.2010.5638951

DOI:
10.1109/ISGTEUROPE.2010.5638951

Document status and date:
Published: 01/01/2010

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
Distributed routing algorithms to manage power flow in agent-based active distribution network

Phuong H. Nguyen, Wil L. Kling, Member, IEEE, Giorgos Georgiadis, Marina Papatriantafilou, Le Anh Tuan, Member IEEE, Lina Bertling, Senior Member, IEEE

Abstract— The current transition from passive to active electric distribution networks comes with problems and challenges on bi-directional power flow in the network and the uncertainty in the forecast of power generation from grid-connected renewable and distributed energy sources. The power flow management would need to be distributed, flexible, and intelligent in order to cope with these challenges. Considering the optimal power flow (OPF) problem as a minimum cost flow represented with the graph, this paper applies a cost-scaling push-relabel algorithm in order to solve the OPF in a distributed agent environment. The algorithm’s performance is compared with the successive shortest path algorithm developed in our previous work. The simulation is implemented for both meshed and radial networks. The simulation results show the advantages of the cost-scaling push-relabel algorithm over the shortest path algorithm in the radial networks with respect to significantly reduced number of exchanged messages on the agent platform, and thus the reduced time for calculation. This will be of great importance if the method is to be applied to a large system.

Index Terms—Smart grid, active distribution network, optimal power flow, multi-agent system, graph theory, cost-scaling, push-relabel.

I. INTRODUCTION

The European energy and climate change targets for the 2020 and beyond would require fast development and use of cost-effective low-carbon energy technologies. A future integrated European power grid will be expected to have a central role to accommodate the large-scale deployment of renewable and decentralized energy sources. The recent European Electricity Grid Initiative (EEGI) [1] proposes a nine-year European research, development and demonstration (RD&D) program to accelerate innovation and the development of the electricity networks of the future in Europe into Smart Grid. The Smart Grid will be a user-centered, market-based, interactive, reliable, flexible, and sustainable electrical network system. Under the distribution network activities of the EEGI, among many highlighted functional projects, active demand-response, metering infrastructure, smart metering data processing, system integration of distributed energy resources (DER), integration of energy storage options in the network management, infrastructure to host electric vehicles/plug-in hybrid electric vehicles, methods and system support, integrated communication solution, etc. are proposed.

The large-scale integration of distributed generation (DG) challenges distribution systems in coping with bidirectional power flows, voltage variations, fault level increases, protection selectivity, power quality and stability. Consequently, several new concepts, such as Microgrid, Autonomous Network, Active Network have to be developed to deal with those problems [2]-[3]. Although differing in approach and implementation, they share the same objective of transferring the current passive distribution networks into active networks (ADN). In the ADN, the power flow management is one of the major problems and needs to be dealt with in order to avoid overloading of components in the network [4].

Distributed methods of control are expected to be helpful for the power system health and security, as they reduce dependencies and enhance the ability of the system to remain in operation after disturbances, loss of equipment, etc. Indeed, distributed solutions that possess locality properties (i.e. where execution depends only on local information) are known to have stabilizing and self-healing properties [5]-[6]. Furthermore, it is possible to have meaningful distributed control solutions for controlling power-flows and other operational objectives not only for parallelizing the problem at a global scale, but also for distributing responsibility among the electric cells, too.

Inspired from the elegant well-known solutions for the min-cost flow problem in graphs proposed in [7]-[8], we propose such a localized distributed solution to the power flow problem that has the following advantages: (i) it enables network elements to operate in a completely autonomous way, based on demand/supply information from their immediate environment; (ii) it offers increased resilience to the network, since autonomous network elements can respond faster to local changes in the power flow.

Our study shows the potential applications of both the successive shortest path and cost-scaling push-relabel algorithms on optimal flow routing in the ADN concept. They are implemented in multi-agent system (MAS) environment which is suitable with distributed context of the future Smart Grid [9]-[10].
II. POWER ROUTING IN ACTIVE DISTRIBUTION NETWORKS

Conventional distribution networks are stable and passive with unidirectional electricity transportation. The term of Active Distribution Network (ADN) is mentioned recently since the distribution network becomes active with DER and RES units leading to bidirectional power flows [11]. It addresses a modernizing architecture of future intelligent power grids to cope with challenges from high penetration of DGs. The so called ADN concept needs to incorporate flexible and intelligent control with distributed intelligent systems [12]. This research elaborates a major capability of the ADN in handling power dispatch and bi-directional flow.

A. Problem formulation

The power flow needs to be controlled to avoid congestion in the network while minimizing the total production cost and maximizing the network security. Hence, this optimizing problem of power flow management, referred to as the optimal power flow (OPF) problem, can be formulated in a mathematical model as follows:

\[
\begin{align*}
\min & \quad A = \sum_{i \in G} \alpha_i P_{G_i} + \sum_{(i,j) \in T} \beta_{ij} P_{T_{ij}} \\
\text{s.t.} & \quad \sum_{i \in G} P_{G_i} = \sum_{(i,j) \in T} P_{T_{ij}} + \sum_{i \in L} P_{L_i} \\
& \quad P_{G_i} \leq P_{G_{max}}, \forall (i \in G) \\
& \quad P_{T_{ij}} \leq P_{T_{max}}, \forall (i,j) \in T
\end{align*}
\]

where, \\
\[A\] total cost function. \\
\[P_{G_i}, P_{T_{ij}}, P_{L_i}\] power generation, power transmission, and load demand. \\
\[P_{G_{max}}, P_{T_{max}}\] Power generation capacity, power transmission capacity. \\
\[\alpha, \beta_{ij}\] represent production cost, device availability. \\
\[G, T, L\] generation, transmission, and load component sets.

The objective function of equation (1) is the total cost for power delivery from the generation areas to the load parts. The production cost constant \((\alpha)\) is the price for selling electricity that can be defined as the nodal price of each generating cell. The transmission cost constant \((\beta_{ij})\) is defined as the charge for using transmission components that depends on the availability and capacity (rating) of the devices. Beside the power balance condition in the equality constraint, the transmitted power needs to be within the device’s thermal limits in the inequality constraint.

Depending on the scale of each cell (sub-network), the reactive power balance will be solved autonomously or globally for a larger area. In the simplified optimization model, the research assumes that all cells are large enough to deal with the reactive power balance autonomously. The voltage constraints can be guaranteed by adjusting DG’s power output and the tap changers of the transformers within cells [13]. Note that the autonomous voltage regulation does not change the power exchanged among cells.

B. Power Routers - Flexible Interfaces

A power router (PR) is a combination of an agent (software) and a power flow controller (hardware), as shown in Fig. 1. Each moderator representing a cell can obtain local area information such as the power flow on incoming (outgoing) feeders, power generation reserve, power load demand, and costs of production and load priority. Besides managing autonomous control actions, this moderator agent can route messages to communicate with the same level agents. A power flow controller (PFC) which is an application of AC/DC/AC converters or an intelligent node [14] controls the power flow for its feeders based on the set points given by the moderator.

With advance control functions based on applications of electronic devices and MAS technology, the PR is expected to create a flexible interface for the future grid. Cells, Microgrid, Autonomous Network, or others can be integrated in the ADN by Voltage Source Inverter (VSI)-based PFC of this interface. Installing PRs in critical local area networks as routers in the internet can help to control power flow actively to avoid congestion problems. Please note that it is not necessary to have PR in every cells of the ADN.

III. DISTRIBUTED OPTIMAL ROUTING ALGORITHMS

Agent-based ADN with the PR interface opens a distributed platform for more flexible and distributed control algorithms. In the graph model, the power flow optimization can be defined as a minimum cost flow problem that regards to both the shortest path (economy) and the maximum flow (capacity) [15]. The cost-scaling algorithm which can be considered as the generalization of the push-relabel algorithm is a strong solution to this problem [5]-[7].

A. Cost-scaling algorithm and distributed implementation for power flow networks

Cost-scaling belongs to polynomial-time algorithms to solve the minimum cost flow problem in complex networks. It is different from capacity scaling which is a scaled version of the successive shortest path algorithm investigated in our previous work [7]. The same example of a 5-bus system is
used to illustrate the algorithm.

The power grid, firstly, is converted to a graph \( G(V,E) \), where \( V \) presents for the set of vertices and \( E \) presents for edges. The edge length (edge cost) \( c_{ij} \) and residual (available) capacity \( r_{ij} \) associated with each edge \((i,j)\) is derived from the transmission cost \( \beta_{ij} \) and the transmission line capacity \( u_{ij} \). A virtual source node \((s)\) is added to connect with cell generation by a source edge \((s,i)\) with residual capacity \( r_{si} \) (cell generation available) and cost \( c_{si} \) (cell production cost \( a_{i} \)). Each cell \( i \) is associated with a load demand \( d_{i} \), node potential \( \pi_{i} \), and excess flow into node \( \epsilon_{i} \).

The excess flow is defined as:
\[
\epsilon_{i} = \sum_{(i,j) \in E} f(i,j) - d(i) \geq 0 \tag{2}
\]
while \( f(i,j) \) is pre-flow that satisfies the flow bound constraint. A node \( i \) with \( \epsilon_{i} > 0 \) is called active node. A branch \((i, j)\) is admissible if \(-\epsilon/2 \leq c_{ij}^{\pi} < 0\).

The algorithm starts with scaling factor \( \varepsilon = \max \{\alpha \beta, \beta_{ij} \} \), and \( \pi_{i} = 0; \forall i \in V; \forall (i,j) \in E \). For a given node potential \( \pi_{i} \), the reduced cost of an arc \((i, j)\) is
\[
c_{ij}^{\pi} = c_{ij} - \pi_{i} + \pi_{j} \tag{3}
\]
Fig. 2 shows represented direct graph of the electrical test network with its parameters. Each active node \( i \) can locally detect and perform on an admissible arc \((i, j)\) a push operation:
\[
\beta_{ij} = \min \{\epsilon_{i}, r_{ij} \} \tag{4}
\]
When the active node \( i \) contains no admissible arc, the algorithm applies a relabel operation to update the node potential by
\[
\pi_{i} = \pi_{i} + \varepsilon/2 \tag{5}
\]
Note that the relabel operation at node \( i \) will increase \( \varepsilon/2 \) units on incoming arcs and decrease \( \varepsilon/2 \) units on outgoing arcs of the node due to the reduced cost condition (3). Consequently, it creates new admissible arcs for push operation. When there is no possibility to push flow forward, node \( i \) can push flow backward to source node \( s \).

In the example, as active node \( s \) has no admissible branch at this point, it performs relabel operation to update node potential as \( \pi_{s} = 3.5 \), as shown in Fig. 3. The operation yields two admissible arcs \((s, 2)\) and \((s, 3)\). Consequently, the push operation is applied on these arcs and makes them saturated. As \( \epsilon_{3} = 8 \), node \( 3 \) is active and added in the list \( S \). The active node list \( S \) is built in first-in-first-out format.

The algorithm repeats until there is no active node in the list. The pre-flow has been converted to \( \varepsilon \)-optimal flow completely. By decreasing \( \varepsilon/2 \) value of \( \varepsilon \) and saturating every arc with negative reduced cost, the \( \varepsilon \)-optimal flow is converted.
The graph is represented by a principle agent distributed agent environment. Each normal node has corresponding action.

Moreover, due to its locality, the algorithm has self-stabilizing and self-healing properties (in response to transient errors or changes in demand/supply, cost or topology), following the analysis in [6]. It is reasonable to assume that nodes will be able to adapt locally to small changes in these parameters (via push-relabel operations), leading to the fast stabilization and recovery. We conjecture that more extensive changes, such as a cascade failure effect, will need more time to recover from but this time will be significantly less than other, centralized min-cost flow solutions.

Concerning the convergence time, as there is no global schedule on the order in which the admissible branch operations are activated, the worst case bound depends on the size of the network. However, in the average case the convergence time is expected to be significantly smaller and the analysis of this property is a significant part for the continued work on this problem.

Fig. 4. Radial configuration of the 5-bus test network.

Fig. 5. Variation of power generation in cases of the radial network.

**TABLE II**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Pseudo-code for ( a_i ) Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Switch</strong> Mode ( i )</td>
<td>Case 1: ( a_i ) initialization()</td>
</tr>
</tbody>
</table>
| | \[
| \begin{align*}
| P_{\text{load}} & , P_{\text{gen}} , \alpha_i , P_{\pi}, \beta_i ; (i,j) \in E \bigg] \leftarrow \text{Grid} \\
| \text{excess} & \leftarrow P_{\text{gen}} - P_{\text{load}} \\
| \text{if} (\text{type} = \text{source}) & \text{then} \\
| & \text{send}_i \text{inform} \left\{ P_{\text{gen}}, \alpha_i, \text{excess} \right\} \rightarrow a_j \\
| \text{else} & \text{end} \\
| S & \leftarrow S - a_j \\
| \text{if} S \neq \emptyset & \text{then send}_i \text{push} \left\{ S, \varepsilon \right\} \rightarrow a_{[j]} \\
| \text{else} & \text{stop} \\
| \text{end} \\
| \text{end} \\
| Case 2: \text{receive push}_i \text{request} \left\{ S, \varepsilon \right\} \rightarrow a_j | |
| | while (excess > 0) do |
| | \begin{align*}
| & \text{begin} \\
| & \text{if contain admissible}(i,j) \in E \text{ then} \\
| & \delta = \min (\text{excess}, r_i) \\
| & \text{send}_i \text{push request} \left\{ \delta, S, \varepsilon \right\} \rightarrow a_j \\
| & \text{else} \\
| & \pi = \pi_i + \varepsilon / 2 \\
| & \varepsilon = \varepsilon_i - r_i + \pi; (i,j) \in E \\
| & \text{end} \\
| & S \leftarrow S - a_j \\
| & \text{if } \varepsilon > 1/n \text{ then} \\
| & \varepsilon \leftarrow \varepsilon / 2 \\
| & \text{send}_i \text{update_e} \{ \varepsilon \} \rightarrow a_j \\
| & \text{else stop} \\
| & \text{end} \\
| \text{end} \\
| Case 3: \text{receive push}_i \text{request} \left\{ S, \varepsilon \right\} \rightarrow a_j | |
| | excess = excess + \delta \\
| | if (excess > 0) then \\
| | S \leftarrow a_j \\
| | \text{send}_i \text{update}_j \{ S \} \rightarrow a_i \\
| | \} |

to \( \varepsilon/2 \)-optimal pre-flow and a new iteration starts. When \( \varepsilon < 1/n \), the algorithm terminates.

In this work, the cost-scaling algorithm is implemented in a distributed agent environment. Each normal node \( i \) of the graph is represented by a principle agent \( a_i \) with its pseudo-code as shown in Table I. A socket proxy agent \( sp_i \) is associated with \( a_i \) to establish a communication with the electrical grid. The virtual source node is represented by a principle agent \( a_j \) with its pseudo-code as shown in Table II. Since each node needs only knowledge from its immediate neighborhood to execute the algorithm, it suffices that nodes exchange the corresponding information with their neighbors each time that there is a change. Thus each node knows when a branch incident to itself is admissible and can take the corresponding action.

**B. Properties of the algorithm**

The proposed method’s convergence properties follow from the analysis of the min-cost flow algorithm in [7]. Moreover, due to its locality, the algorithm has self-stabilizing and self-healing properties (in response to transient errors or changes in demand/supply, cost or topology), following the analysis in [6]. It is reasonable to assume that nodes will be able to adapt locally to small changes in these parameters (via push-relabel operations), leading to the fast stabilization and recovery. We conjecture that more extensive changes, such as

**TABLE III**

<table>
<thead>
<tr>
<th>From Cell</th>
<th>To Cell</th>
<th>Power flow variation</th>
<th>CS-PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>6.846</td>
<td>2.009</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1.842</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>-3.033</td>
<td>-3.000</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4.866</td>
<td>5</td>
</tr>
<tr>
<td>No. of messages</td>
<td>137</td>
<td>154</td>
<td></td>
</tr>
</tbody>
</table>

**OPTIMAL OPERATION**

<table>
<thead>
<tr>
<th>From Cell</th>
<th>To Cell</th>
<th>Power flow, MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>6.846</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1.842</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>-3.033</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4.866</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power flow, MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.009</td>
</tr>
<tr>
<td>0.000</td>
</tr>
<tr>
<td>-3.000</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power flow, MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.096</td>
</tr>
</tbody>
</table>

**OPTIMAL OPERATION**

<table>
<thead>
<tr>
<th>From Cell</th>
<th>To Cell</th>
<th>Power flow, MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>6.846</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1.842</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>-3.033</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4.866</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power flow, MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.009</td>
</tr>
<tr>
<td>0.000</td>
</tr>
<tr>
<td>-3.000</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power flow, MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.096</td>
</tr>
</tbody>
</table>

**OPTIMAL OPERATION**

<table>
<thead>
<tr>
<th>From Cell</th>
<th>To Cell</th>
<th>Power flow, MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>6.846</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1.842</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>-3.033</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4.866</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power flow, MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.009</td>
</tr>
<tr>
<td>0.000</td>
</tr>
<tr>
<td>-3.000</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power flow, MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.096</td>
</tr>
</tbody>
</table>
with the push-relabel operation of the CS-PR.

explained by the augmenting characteristic of SSP compared
scenarios but slightly different in power flows. It can be
of both SSP and CS-PR. They have the same power generation
algorithm (SSP). Table III shows power generation and flow
relabel (CS-PR) algorithm with the successive shortest path
receive data from the socket proxy agents of JADE.

TCP/UDP/IP Toolbox, each “Embedded Matlab Function” in
agent in JADE is used as a server socket. By using the
communication between Matlab/Simulink and JADE is based
Matlab/Simulink. MAS is created under the Java Agent
Fig. 6. Pre-flows after performing relabel and push operation.

As the algorithms are applied in the meshed network, SSP
has less number of messages exchanged in MAS platform
has less number of messages exchanged in MAS platform
(137) than CS-PR has (154).

B. Radial network
Based on the push-relabel principle, the CS-PS algorithm is
expected to spend much less computation efforts in the radial
networks. We investigate this advantage of the algorithm on a
radial configuration of the 5-bus test network, as shown in Fig.
Two simulation cases, i.e., base case and extreme case, are
examined.

In the base case, the production costs \( \alpha \) of three bus
generation are remained as in the previous case. The
transmission costs \( \beta \) of all branches are assumed equal 1
p.u. Fig. 5 shows power generation during the simulation time.
Initial state of the system is generated by SimPowerSystem
toolbox with \( P_{g1}=10.446 \text{MW}, P_{g2}=9.351 \text{MW}, \) and \( P_{g3}=16.308 \). At \( t=10\text{sec} \), the optimal routing algorithms is started. In this
simulation case, the CS-PR algorithm has the same power
generation \( (P_{g1}=5.668 \text{MW}, P_{g2}=10 \text{MW}, \) and \( P_{g3}=18 \text{MW}) \) and
flow as SSP with total cost of 126.88 p.u. However, CS-PR
has exchanged significant less messages (76) comparing to
SSP’s (115). As can be seen from the results, the push-relabel
operation of CS-PR is very effective in the radial
configuration. It omits significant unnecessary loops which the
augmenting path algorithm has in the same network.

To see more in detail the CS-PR’s advantage, the
simulation investigates an extreme case with power generation
injected only in bus 1. Production costs of three bus generation
are 1, 10, and 10, respectively. It is assumed that there are
enough generation and line capacities. Fig. 6(a) and 6(b) show

<table>
<thead>
<tr>
<th>TABLE IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERFORMANCES OF THE ALGORITHMS ON THE RADIAL NETWORK</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Base case</td>
</tr>
<tr>
<td>Extreme case</td>
</tr>
</tbody>
</table>

IV. SIMULATION AND RESULTS

The above example of the 5-bus system is simulated using
Matlab/Simulink. MAS is created under the Java Agent
communication between Matlab/Simulink and JADE is based
on client/server socket communication. The socket proxy
agent in JADE is used as a server socket. By using the
TCP/UDP/IP Toolbox, each “Embedded Matlab Function” in
Matlab/Simulink can create a client socket to send data to and
receive data from the socket proxy agents of JADE.

A. Meshed network
In this case, we use the 5-bus meshed test network
presented in [7] to compare results of the cost-scaling push-
relabel (CS-PR) algorithm with the successive shortest path
algorithm (SSP). Table III shows power generation and flow
of both SSP and CS-PR. They have the same power generation
scenarios but slightly different in power flows. It can be
explained by the augmenting characteristic of SSP compared
with the push-relabel operation of the CS-PR.

As the algorithms are applied in the meshed network, SSP

Fig. 6. Pre-flows after performing reliable and push operation.
represented direct graphs of the test network for CS-PR and SSP algorithms. Fig. 6(d) shows the main drawback of the SSP method in the final iteration loop. The algorithm discovers the final shortest path in bold lines which goes through all nodes of the network. The algorithm continues to send flow on the same arcs with previous augmenting path before it fills full the last arc \((5, r)\). On the contrary, CS-PR pushes flows along the individual arcs. Excess of 35 units is pushed from \(s\) straightforward through each arc to lower nodes. Fig. 6(c) shows the final stage of push operation when bus 4 realizes admissible arc \((4, 5)\) and sends 5 units to bus 5. It is quite straightforward and there is no repetitive computation.

By pushing as much power as possible along the radial feeder, CS-PR takes only 52 messages to converge while SSP needs nearly two times that number in messages (100). In this case, both algorithms have also the same power generation and flow result with total cost of 104.13 p.u.

V. CONCLUSIONS AND FUTURE WORK

Following the previous work on the application of the successive shortest path algorithm [7], this paper presents a distributed implementation of the cost-scaling push-relabel algorithm to manage the power flow in the active distribution network. Due to its locality, the algorithm presented in this paper has self-stabilizing and self-healing properties in response to transient errors or changes in the demand/supply, cost or topology. Performances of the two algorithms are compared in both meshed and radial networks. In the meshed network, there is no significant difference between two methods. The advantage of the cost-scaling push-relabel solution is realized in the radial test network. Number of messages exchanged among multi-agent system in this algorithm is significantly smaller than in the successive shortest path method.

By using a global schedule on the order of the active nodes, the algorithm includes a subtle centralized characteristic. To have a fully distributed algorithm, we plan to remove this piece as part of future work. A large-scale test network, including simulations of the Swedish power grid in connection to the real-time simulator system ARISTO [12], will also be investigated in the future.

VI. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the Swedish Civil Contingencies Agency (MSB) and from European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n\^o 257007.

VII. REFERENCES


VIII. BIOGRAPHIES

Phuong H. Nguyen was born in Hanoi, Vietnam in 1980. He received his M.Eng. in Electrical Engineering from the Asian Institute of Technology, Thailand in 2004. From 2004 to 2006 he worked as a researcher at the Power Engineering Consulting Company No. 1, Electricity of Vietnam. In the end of 2006 he joined the Electrical Power System Research group at Eindhoven University of Technology, the Netherlands as a Ph.D student. He is working under the framework of the “Electrical Infrastructure of the Future” project.

Wil L. Kling (M'95) was born in Heesch, The Netherlands in 1950. He received the M.Sc. degree in electrical engineering from the Eindhoven University of Technology, The Netherlands, in 1978. From 1978 to 1983 he worked with Kema and from 1983 to 1998 with Sep. Since then he is with TenneT, the Dutch Transmission System Operator, as senior engineer for network planning and network strategy. Since 1993 he is a part-time Professor at the Delft University of Technology and since 2000 he is also a part-time Professor in the Electric Power Systems Group at the Eindhoven University of Technology, The
Netherlands. From December 2008 he is appointed as a full-time professor and a chair of EPS group at the Eindhoven University of Technology. He is leading research programs on distributed generation, integration of wind power, network concepts and reliability.

Mr. Kling is involved in scientific organizations such as Cigre and IEEE. He is the Dutch Representative in the Cigre Study Committee C6 Distribution Systems and Dispersed Generation.

Giorgos Georgiadis was born in Drama, Greece in 1981. He received his M.Sc. in Computer Science and Engineering from the Department of Computer Engineering and Informatics, University of Patras, Greece in 2006. At 2007 he joined the Distributed Computing and Systems group at the Department of Computer Science and Engineering, Chalmers University of Technology, Sweden as a PhD student. His research interests include distributed algorithms and overlay networks.

Marina Papatriantafilou is Associate Professor at the Department of Computer Science and Engineering, Chalmers University of Technology, Sweden. She received the PhD degree from the Department of Computer Engineering and Informatics, University of Patras, Greece in 1996. She has also worked at the National Research Institute for Mathematics and Computer Science in the Netherlands (CWI), Amsterdam and at the Max-Planck Institute for Computer Science (MPI) Saarbruecken, Germany. Her research is on distributed and multiprocessor computing, including synchronization, communication/coordination, with emphasis in robustness, fault-tolerance and dynamic aspects.

Le Anh Tuan (S’01, M’09) received his Ph.D. in 2004 in Power Systems from Chalmers University of Technology, Sweden, and his M.Sc. degree in 1997 in Energy Economics from Asian Institute of Technology, Thailand. Currently he is a senior lecturer at the Division of Electric Power Engineering, Department of Energy and Environment, Chalmers University of Technology, Sweden. His research interests include power system operation and planning, power market and deregulation issues, grid integration of renewable energy and plug-in electric vehicles.

Lina Bertling (S’98-M’02-SM’08) was born in Huddinge, Sweden, in 1973. She has a Professor Chair in Sustainable Electric Power Systems and is Head of the Division of Electric Power Engineering, at the Department of Energy and Environment, at Chalmers University of Technology, in Gothenburg, Sweden. She has been with Svenska Kraftnät, the Swedish Transmission System Operator during 2007-2009, and from June 2008 as head of the R&D. She has been with KTH School of Electrical Engineering, in Stockholm, during 1997-2009 where she finalized her Docent degree, Associate Professor, in 2008, and the Ph.D. in 2002, both in Electric Power Systems. Her research interests are in transmission and distribution systems including high voltage equipment and HVDC, and wind power systems with applications for reliability assessment and modeling, and maintenance planning.

Dr. Bertling is a senior member of IEEE and a member of Cigré, Cired, World Energy Council, and the Royal Swedish Academy of Engineering Sciences. She was the general chair of the 9th International conference on probabilistic methods applied to power systems (PMAPS) in Stockholm, in 2006 and is the chair of the first IEEE PES Conference on Innovative Smart Grid Technologies Europe 2010 in Gothenburg in 2010.