Running GO package on a Linux Cluster via Open Mpi

Yuyang Chen
Agenda

• Introduction

• Live demonstration of running GO package

• GO package vs MATPOWER OPF result comparison.

• GO package code related discussions

• Conclusion
Introduction

• According to the paper “Efficient AC Optimal Power Flow and Global Optimizer Solutions” a software package have been written that utilizes parallelism on a Linux cluster to speed up calculations.

• We will see the result of running this package today
Live demonstration of running GO package
GO package vs MATPOWER (D&C vs MIPS) OPF result comparison.

• Cases that show near identical Objective function values: \textit{case14, case 30}
• Cases that show significantly better Objective function values: \textit{case6ww, case9}.
• Case that shows a hard time of converging to a global optimum: \textit{case 3}
Cases that show near identical Objective function values: **case14** (case14 runs slightly better in GO package), case 30

```
>> runpf('case14.m')

(using built-in linear solver)
Converged!
Converged in 1.98 seconds
Objective Function Value = 8081.53 $/hr
```

```
>> runpf('case30.m')

(using built-in linear solver)
Converged!
Converged in 0.34 seconds
Objective Function Value = 576.89 $/hr
```

---

<table>
<thead>
<tr>
<th>Feasible solution for OFF is found,</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact Objective Function Value : 8081.53 $/hr</td>
</tr>
<tr>
<td>For the best solution: 9001.523924</td>
</tr>
</tbody>
</table>

---

| So far the best solution: 9001.523924 |
|-------------------------------|------------------|
| Lower Bound : 1980000000 |
| Dep : -122.7390385 |
| Minimum Delta.EVD : 4.245934538e-05 |
| Number of Nodes in list : 6 |
| total time cost : 0.636003 sec |

Finished

---

| So far the best solution: 9001.523924 |
|-------------------------------|------------------|
| Lower Bound : 1980000000 |
| Dep : -122.7390385 |
| Minimum Delta.EVD : 4.245934538e-05 |
| Number of Nodes in list : 6 |
| total time cost : 0.636003 sec |

Finished

---
Cases that show better Objective function values: case6ww, case9.

<table>
<thead>
<tr>
<th>Bus</th>
<th>[Sf]</th>
<th>mu</th>
<th>[Sf]</th>
<th>[Smax]</th>
<th>[St]</th>
<th>[St]</th>
<th>mu</th>
<th>Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
<td>7.874</td>
<td>60.00</td>
<td>60.00</td>
<td>57.52</td>
<td>-</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

```
>> runopf('case6ww.m')
```

(using built-in linear solver)
Converged!
Converged in 0.06 seconds
Objective Function Value = 3143.97 $/hr

```
>> runopf('case9.m')
```

(using built-in linear solver)
Converged!
Converged in 0.07 seconds
Objective Function Value = 5296.69 $/hr

```
flag率(601,549),(634,582)
```

```matlab
Primal value greater than SFTB
Gap : 4211.74385
Minimum Delta, EVD : 4.24389150e-314
Number of Nodes in list : 1
total time cost : 101.398354 sec
Finished
```
Case that shows a hard time of converging to a global optimum: case 3

```
>> runopf('case3.m')

(using built-in linear solver)
Converged!
Converged in 0.11 seconds
Objective Function Value = 5812.60 $/hr
```

Primal value greater than SFTB
So far the best solution: 5842.812814
lower Bound: 5790.151936
Gap: 0.009012990672
Minimum Delta_EVD: 4.804262799e-05
Number of Nodes in list: 24521
total time cost: 2382.272032 sec
GO package code related discussions

• C++ to MEX file conversion (Single source file vs multiple source files)

    ```
    mex -v -compatibleArrayDims yprime.c
    
    The output displays information specific to your platform and compi
    ```

    ```
    mex -largeArrayDims fulltosparse.F loadsparse.F
    ```

• The problem is compiling of c files in this package is not generic. It involves MPI PICC compiler options.

• More tests needed for conversion.

• Also currently the code package is setup to run on a linux cluster that supports MPI, only.
Conclusion

• GO package shows significant improvement in some cases and sometimes it shows difficulties to converge. The implementation is not yet robust.

• Generally GO package takes longer than MIPS in MATPOWER, but it finds better solution that MATPOWER misses