ELECTRIC POWER RESEARCH INSTITUTE

OpenDSS Load Allocation and State Estimation Algorithm

Roger Dugan
Sr. Technical Executive
EPRI
942 Corridor Park Blvd.
Knoxville, TN 37932

October 16, 2008

This white paper describes the load allocation and state estimation algorithms presently programmed into the OpenDSS program. Of course, the user may write another algorithm in an external program and drive the OpenDSS accordingly.

EnergyMeters and Sensors

Load allocation and the load allocation portion of the Estimation process is controlled by the Energymeter object. Thus, the best results are achieved by placing an EnergyMeter object at the head of each feeder in the problem. The user then specifies the **Peakcurrent** property as an Array of 3 current magnitudes (for a typical 3-phase meter), nominally representing the "drag hand" readings at the substation for each of the three phases. That is, the peak current readings over a monitoring interval. For example:

```
! Define the Energy Meter
New Energymeter.Feeder1 Element=Line.Line1  Terminal=1

(… other statements …)

! Script for allocating loads
Energymeter.Feeder1.peakcurrent=[394, 301, 403]
Allocateloads
Show Currents Elements    ! see how well we did


Set ShowExport=Yes       ! so Export automatically appears
Export Estimation        ! View Estimation errors in CSV file
```

For annual simulations, this is usually the peak current for the year, not counting temporary conditions in which other feeders are attached to the feeder under study during a contingency.

When the AllocateLoad function of the EnergyMeter object is invoked, it runs down the feeder adjusting Load object AllocationFactor and Cfactor properties in the direction needed to better match the Peakcurrent values.

With an EnergyMeter object at the head of the feeder, Sensor objects may be scattered over the feeder.

A Sensor is a benign circuit element that represents a device on the feeder or load that can sense and report some combination of voltage, current, kwatts, and kvars. For the load allocation and estimation process, a Sensor takes responsibility for setting AllocationFactors and Cfactors for any Load objects down stream from its location to better match the values in its measurement registers. The measurement registers are loaded using some combination of the V, I, P, and Q properties. These are array values representing the number of phases monitored by the physical sensor. Before loading the registers they should be cleared using the Clear property.

A simple example:

```
! Define a Sensor
New Sensor.MySensor Element=Line.SomeLine Terminal=1 kVBase=11


(… other statements …)

! Script for preparing for allocating loads by current value
Sensor.MySensor.Clear=Yes   Currents=[50, 45,39] ! set current registers
```

The sensor "MySensor" picks up its connection to the phase currents in the Line object called "SomeLine" when it is attached to terminal 1 of the line in the New statement. This defines the number of phases for the sensor and also defines which node voltages correspond to the terminal. The Sensor copies the node references from the Line.

During the Allocation or Estimation process, the Sensor compares the values in its registers to values calculated by the DSS. Then it computes correction factors for the loads under its part of the circuit tree.

As the EnergyMeter at the head of the feeder builds its circuit tree zone, it assigns each Load object to either itself or its closest upline Sensor. This establishes the connection from the Load to its correction factors during the process. Thus, all the EnergyMeter has to do during the process is ask each load to check its controlling sensor to obtain the appropriate correction for the next step in the process.

Defining Load Objects

There are presently 5 different ways to define the base load values of kW and kvar in a Load object:

1. kW, PF
2. kW, kvar
3. kVA, PF
4. XFKVA * Allocationfactor, PF
5. kWh/(kWhdays*24) * Cfactor, PF

The first three define a fixed load whose kW and kvar values are assumed to be known. They may be subsequently modified by the load model, LOADSHAPE objects, and GROWTHSHAPE object, but that is not to be confused with modification by the Allocation or Estimation process.

The Allocation and Estimation processes are intended to find and set the kW and kvar values from either connected kVA or metered kWh values. You must define some (ideally, most) of the loads with one of the last two (4 or 5) methods for the Allocation and Estimation algorithms to have the greatest success. If all the loads were known and fixed, there would be no need for Allocation or Estimation.

Load objects that are subject to allocation are defined using either the **xfkVA** or **kWh** properties and related properties.

Loads that are known are defined using a combination of the kW , PF, and kVA properties.

Once the Allocation or Estimation has concluded, the kW and kvar values determined by the algorithm as assumed to be the known values and are subsequently modified by load shapes and growthshapes if desired.

Note that the **PF** property should be set by some means for the load allocation algorithms to work. There is a reasonable default value for many types of composite loads. The property may be explicitly set when each load is defined. Or you may set the kW and kvar properties to a typical value, which has the side effect of establishing the nominal power factor. Then you would override that specification by setting either the xfkVA or kWh property.

The “allocationfactor” property of each load defined by XFKVA defaults to 0.5. Thus, the nominal kW of the load is defined by the present values of allocationfactor, xfkVA, and PF.

3-phase and 1-phase Loads

The load allocation algorithm handles 3-phase and 1-phase loads differently. In the OpenDSS, 2- or 3-phase loads are assumed to be balanced over the phases present. If the unbalance is significant define the loads with 2 or 3 single-phase Load objects. This gives the load allocation process maximum flexibility for matching the measured values, because it can alter each phase independently.

2- and 3-phase loads are modified by the average of the three individual phase correction factors.

If a particular load is a mix of balanced 3-phase loads such as motors and 1-phase loads such as computers or lighting, it might be good to define a 3-phase load at the site along with three single-phase loads. You may theoretically have any number of loads connected to the same bus, although it is possible that some combinations will not have good convergence characteristics.

XFKVA Property

This is the kVA rating of the transformer supplying the load. In the case where several loads are lumped together, use the total kVA rating. Obviously, you could use an artificial value for this property simply to skew the allocation.

AllocationFactor Property

The AllocationFactor is multiplied times the XFKVA property to determine the estimate of the total load kVA. This is then split into kW and kvar using the present value of the PF property. The default value is 0.5, which is a common average value for many feeders. You may set this to a different value. The load allocation algorithm will then start from the value you set and ratio it up or down to gain a more satisfactory match to the objective.

kWh, kWhDays Property

These properties are used to represent the load by the metered kWh over some period. The average kW usage is determined from the kWh property divided by the total number of hours (24 * kWhDays) in the metered period. Default value for kWhDays is 30.

CFactor Property

This factor converts the average kW usage to peak kW, kvar at the specified PF. This factor is adjusted for loads defined by the kWh property instead of the AllocationFactor property. The default value is 4. Typical values run in the 1-4 range.

Load Allocation Algorithm

The basic load allocation algorithm is as follows:

A snapshot power flow is performed using the known load values and the nominal guess at the loads to be allocated. The resulting currents are compared to the specified Peakcurrent array in the Energymeter object at the head of the feeder. Two allocation factors are then computed:

1. One for single-phase loads based on the ratio difference between the peakcurrents for each phase and the computed estimate.
2. The average of the 1-phase allocation factors computed in step 1 for allocating 2-phase and 3-phase loads.

The AllocationFactor or Cfactor properties on all loads in the Energymeter's zone are modified by one of these factors depending on the number of phases and the way the Load is defined. The Load object knows how it is defined and chooses the proper one.

Similar correction factors are computed for each Sensor object in the EnergyMeter's Zone.

This process is presently repeated the number of times specified by the **NumAllocIterations** option to improve the guess. In the past, the program used other techniques for improving the guess, such as running a limited number of monte carlo cases (20 to 50) randomly varying the allocationfactor properties around a target and capturing the case that gave the best match to the Peakcurrent array.

In general, there was not sufficient improvement in the estimation to warrant doing more than execute the simple algorithm twice. However, if you think the allocation could be improved, simply issue the “Allocateloads” or “Estimate” command again and it will repeat the process starting from where the last solution left off.

```
Set NumAllocIterations=4   ! Default is 2
AllocateLoads              ! Does 4 iterations on the load allocations
AllocateLoads              ! Does 4 more
```

Summary

In summary, load allocation is carried out by

1. Building the circuit model
    a. Define Loads with XFKVA or kWH property
    b. Introduce sufficient 1-phase loads to allow the allocation algorithm to adjust for unbalance
2. Defining an EnergyMeter object at the head of each feeder of interest.
3. Defining Sensor objects at other places if desired.
4. **Solve** in snapshot model, adjusting allocation factors to converge if necessary
5. Issue **AllocateLoads** command or **Estimate** command
6. Check power flow results to check the allocation.
7. **Export Estimate** will produce a CSV file with a summary of how well each EnergyMeter object and Sensor object register values match the present solution
8. **Dump Allocationfactors** will produce a script that can be used to define the allocation factors in future simulations directly without having to perform the allocation function. For example (a snippet from a dump file):

```
Load.s50c.AllocationFactor=0.57336
Load.s56b.AllocationFactor=0.58478
Load.s58b.AllocationFactor=0.58478
Load.s59b.AllocationFactor=0.58478
Load.s60a.CFactor=4.5404
Load.s62c.AllocationFactor=0.57336
Load.s63a.AllocationFactor=0.56755
Load.s64b.AllocationFactor=0.58478
Load.s69a.AllocationFactor=0.56944
Load.s70a.AllocationFactor=0.56944
Load.s71a.AllocationFactor=0.56944
Load.s73c.CFactor=4.468
Load.s74c.AllocationFactor=0.5585
Load.s75c.AllocationFactor=0.5585
Load.s76a.AllocationFactor=0.56944
Load.s76b.AllocationFactor=0.56743
Load.s76c.AllocationFactor=0.5585
```