AUTONOMIC CONTROL ALGORITHM SELECTION IN DECENTRALISED POWER SYSTEMS: A VOLTAGE CONTROL CASE STUDY

James KING
Parsons Brinckerhoff – UK
james.king@pbworld.com

Philip TAYLOR
Durham University – UK
p.c.taylor@durham.ac.uk

Samuel JUPE
Parsons Brinckerhoff – UK
JupeS@pbworld.com

ABSTRACT

As power systems evolve into "Smart Grids" and beyond, they will increasingly rely on numerous sophisticated control algorithms to meet the demands of users and optimise operation. However, it is unlikely that a single algorithm can deliver the best performance for all possible network configurations, conditions and performance objectives. By judiciously selecting different algorithms better overall performance can be achieved. This paper describes an approach to selecting control algorithms for power systems; including a case study demonstrating both the need for and the effectiveness of algorithm selection for a voltage control problem on an 11 kV network.

INTRODUCTION

There are numerous drivers for adoption of "Smart Grids" [1], including allowing more renewable generation to connect, increasing reliability, improving efficiency and deferring expensive network reinforcements. An essential technical support of Smart Grid efforts is the development of sophisticated control algorithms such as those for voltage [2] or power flow control [3].

However, it is unlikely that individual control algorithms could solve all potential control problems, as their design will inevitably be limited by the assumptions and compromises made by their designers. Control algorithms tend to be designed for a particular control task, and could, therefore, lack the flexibility to be applied to other control tasks; for example, it would be likely for a voltage control algorithm to perform badly at managing thermal constraints compared with a power flow management algorithm. The complexity of a network's configuration (including its size, physical topology and mix of connected devices) and state can degrade algorithm performance; for example, a voltage control algorithm that assumes load-only networks is likely to control voltages poorly if applied to networks with significant distributed generation. Furthermore, uncertainty about the current or future network configuration and state can affect algorithm performance; for example, an algorithm that predicts future loads in order to dispatch matching levels of generation may under-dispatch if loads unexpectedly change, such as in response to an unanticipated cold spell. As algorithm performance varies, there will be conditions where different algorithms will be best at meeting the current performance objectives. Therefore, improved overall performance may be achieved by selecting algorithms appropriate to the current problem.

The problem of judiciously selecting control algorithms for power systems is being researched as part of the Autonomic Power System (APS) project [4]. The APS is a multi-disciplinary, 4.5-year, UK government-funded project being undertaken by over 40 researchers in a consortium of UK universities including Durham University, the University of Manchester and the University of Strathclyde; along with industrial partners including IBM, National Grid and Parsons Brinckerhoff. The project seeks to develop power system network operation and control for a time horizon of 2050. This is beyond the current vision of Smart Grids, with the APS aiming to meet the constantly-changing goals of the system's stakeholders by continually and appropriately deploying decentralised control algorithms in dynamic zones of control (illustrated in Fig. 1).

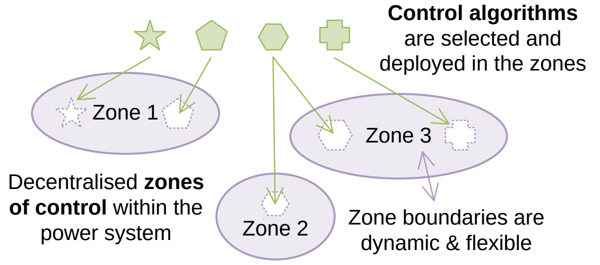


Figure 1 – Overview of the APS concept

Within each control zone of the APS, it is necessary to select control algorithms appropriate to the current objectives, network configuration and conditions. Research into methods of algorithm selection in power systems is not just limited to operational selection of control algorithms, however. The methods can be extended to develop planning tools for future networks and could also direct the development of novel control algorithms, by finding performance gaps in current algorithms.

THE ALGORITHM SELECTION PROBLEM

The challenge of selecting algorithms appropriate for solving different problems has been established in Computer Science for some time, with the so-called "Algorithm Selection Problem" being formulated by John Rice almost 40 years ago [5]. A recent increase in interest in this problem has seen successful applications in, for example: Boolean satisfiability [6], optimisation [7] and genetic algorithms [8], among others. However, to the authors' knowledge algorithm selection and its potential benefits have not yet been applied to power systems control.

Model of the Algorithm Selection Problem

Rice produced a model of the Algorithm Selection Problem, the general form of which is presented in Fig. 2.

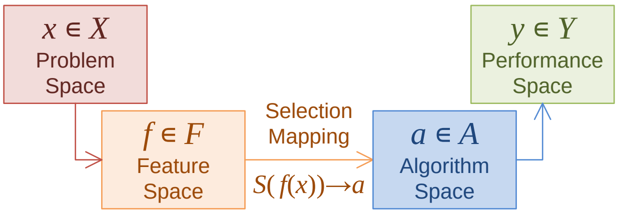


Figure 2 – Model of the Algorithm Selection Problem

Rice's model consists of four spaces:

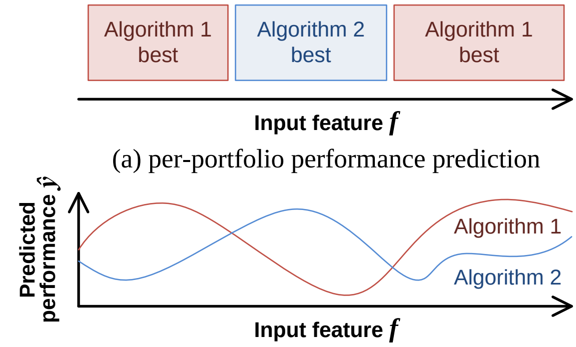
- *Problem Space*: a space X containing all the problem instances x under consideration. In power systems, this space could contain a complete description of the system's state, including bus voltages, individual generator set points and switch statuses, for example.
- *Feature Space*: a space F containing features $f(x)$ extracted from each problem instance x, which should ideally be simpler and a lower dimension than x. Where the description of each problem instance is small the problem instance can be used directly without the need for feature extraction, so in these cases: $f(x) = x$. In power systems, this space would contain some subset of the problem space, such as voltages at certain key buses and the sum of total generator output, for example.
- *Algorithm Space*: a space A containing all the algorithms, a, under consideration (sometimes referred to as the *algorithm portfolio*).
- *Performance Space*: a space Y containing the performance, $y(a, x)$, of each algorithm a applied to each problem instance x. Performance can be multi-dimensional so, taking a power system example, each algorithm's performance could include its ability to control voltages and the total losses it incurs once deployed, for example.

The issue at the heart of the Algorithm Selection Problem is, given the other items in the model above, to determine the *selection mapping* $S(f(x)) \rightarrow a$. This is the mapping from problem features in F to an algorithm in A; in other words, an *algorithm selector*. Most commonly S is chosen in order to maximise the norm of performance $\|y\|$ for all x (typically, the norm is a measure of average performance), although there are variations such as only considering a subclass of problems or a subclass of all possible mappings.

Deriving Selection Mappings (Algorithm Selectors)

Deriving the best selection mapping is, ironically, another selection problem. Rice proposed to use the tools of approximation theory to tackle this problem, although the majority of recent work on the Algorithm Selection Problem has used machine learning techniques instead. The approach is essentially to gather empirical performance data for the algorithm portfolio of interest, and apply machine learning to develop performance models of the algorithms. The performance models can be developed in one of two ways so that they can be used as algorithm selectors:

- On a *per-portfolio* basis, where a performance model for the whole portfolio of algorithms is learnt, which predicts the best performing algorithm a from features $f(x)$ (Fig. 3a).
- On a *per-algorithm* basis, where a performance model for each algorithm is learnt that predicts the performance $\hat{y}$ from features $f(x)$ (Fig. 3b) and the algorithm with the best $\hat{y}$ is selected.



(b) per-algorithm performance prediction

Figure 3 – Different algorithm performance models that can be used for algorithm selection

There are two broad types of machine learning that can be used for algorithm selection: *classification* and *regression*. Classification seeks to predict what *class* (a category) a set of input features belongs to, and is most suited to per-portfolio selection. Regression, on the other hand, seeks to predict *real-valued* outputs from input features, and is more suited to per-algorithm prediction and selection.

CASE STUDY

This case study examines the ability of a number of control algorithms to maintain voltages within limits (0.97 and 1.03 pu) in an 11 kV network under a variety of network conditions. The potential of selecting between the different algorithms is then investigated as a way of improving the voltage control performance.

Network Model

The network model for this case study came from the AuRA-NMS project [2]. The network topology is shown in Fig. 4, and consists of a grid infeed at 33 kV connected to three step-down 33/11 kV transformers, which in turn supply radial feeders, each with multiple loads. On one feeder are two Distributed Generators (DGs).

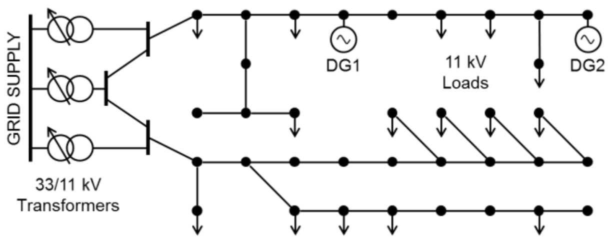


Figure 4 – Case study 11 kV network

The model was implemented in IPSA and the following features were varied within it, with each combination forming a problem instance (2625 in all) to be solved:

- Transformer(s) tap setting {1, 2, … 21}
- DG 1 output {0%, 25%, 50%, 75%, 100%} (real power modulated, with a constant power factor)
- DG 2 output {0%, 25%, 50%, 75%, 100%}
- Load levels {50%, 75%, 100%, 125%, 150%}

Control Algorithms

The following algorithms were considered in the case study:

1. *NULL*: this algorithm does not produce any control actions, and thus characterises the network's base response to the current problem instance.
2. *AVC098*: implements Automatic Voltage Control (AVC) on the 3 transformers simultaneously, with a voltage set point of 0.98 pu on the LV side.
3. *AVC102*: AVC with a voltage set point of 1.02 pu.
4. *CBR*: implements a version of the case-based reasoning voltage control approach developed for the AuRA-NMS project [2]. The case base for this algorithm comprises of previous voltage-outside-limit events and control actions (tap changer steps and/or DG output modulation) taken to mitigate these. On detecting voltages outside limits, the case base is interrogated to find a past case most similar to the current conditions, and the actions taken for the previous case are retrieved and applied to the current network.

Algorithm Performance

Each of the 4 control algorithms was applied to the full set of problem instances. Figure 5 shows that no algorithm could solve all the voltage control problem instances. The poor performance of *NULL* indicates that without using one of the other algorithms, the network will suffer from under- or over-voltage for the majority of network conditions studied. The weak performance of *CBR*, compared with *AVC098* and *AVC102*, is most likely due to the case base not being optimised to the range of problem instances considered. Also, the implementation of the *CBR* algorithm lacked the online validation step used in [2], which may also have lowered its performance.

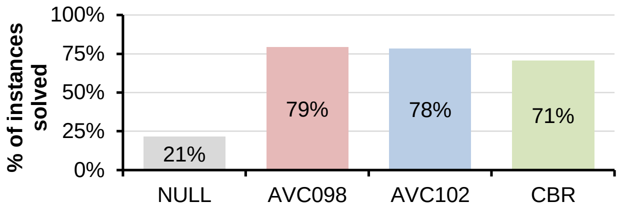


Figure 5 – Performance of each algorithm on all problem instances

However, as shown in Fig. 6, each problem instance can be solved by one or more of the algorithms. Therefore, if an appropriate algorithm is selected for each instance, there is the potential to solve every problem instance.

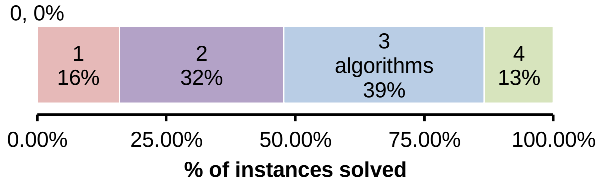


Figure 6 – Number of algorithms able to solve each problem instance

Algorithm Selector Development

As there was a potential performance gain in selecting control algorithms for the case study network, an algorithm selector was developed. This was a per-portfolio selector, which directly predicts which algorithm would perform best based on the current problem instance features.

To develop the selector, each problem instance was labelled with an algorithm that was able to solve that instance. Where more than one algorithm could solve an instance, the labelled algorithm was chosen according to the order *AVC098*, *AVC102*, *CBR* and *NULL*, as each instance could only be labelled with one algorithm. Thus each problem instance was represented by a tuple consisting of the 4 input features (e.g. load level) and 1 output feature (the labelled algorithm). This labelled instance data was randomised and then used to induce a classification tree using the C4.5 tree learning algorithm in the Weka machine learning toolkit [9].

Algorithm Selector Performance

The tree learnt using 66% of the labelled instance data is shown in Fig. 7 (overleaf). When tested using the remaining 34% of instance data, the tree was able to predict the correct algorithm to use with 100% accuracy, and thus each problem instance could be solved. The performance of this algorithm selection approach compares favourably with the performance of the individual algorithms applied to the same test dataset (as shown in Table 1), as none of these can solve all the test instances, whereas the selection approach can solve all. Furthermore, the tree structure was stable and gave the same predictive performance even when only 25% of the instance data was used for training.

Algorithm	No. of Instances Solved	% of Instances Solved
NULL	207	23.21%
AVC098	704	78.92%
AVC102	690	77.35%
CBR	631	70.74%
Using Selector	892	100.00%

Table 1 – Performance of individual algorithms and selection approach on test dataset (34% of instances)

DISCUSSION

The high accuracy of the algorithm selector is in part due to the learning algorithm used, but it also in part due to the order of algorithms used when labelling the instance data.

It was found that changing the order of the algorithms affected the predictive performance of the learnt tree. The order (*NULL*, *CBR*, *AVC102*, *AVC098*) had the lowest prediction accuracy (94.28%) when trained on 66% of the instance data; although it still solved more problem instances than any of the individual algorithms.

The reason why the order chosen affects the predictive performance is that it introduces artificial structure into the instance data. The order changes the number and location of regions of one class (labelled control algorithms) within the performance space, increasing its complexity. As the tree learning algorithm looks for these regions and attempts to define partitions to separate them, the different compositions of regions will affect the learning algorithm's ability to create partitions that effectively separate classes. Thus, misclassifications are more likely to occur and the predictive performance of the learnt tree is lower.

The size of the learnt tree indicates the complexity of the performance space being learnt. The tree in Fig. 7 is relatively small compared with the trees learnt from different algorithm orders, which increase the complexity of the performance space. For example, the order that gives the lowest predictive performance produces a tree with 159 nodes, whereas the tree of Fig. 7 has only 39.

The effect of different orderings on predictive performance is one weakness of per-portfolio algorithm selection. Selection via per-algorithm performance models may be more appropriate in some cases. Per-algorithm selection may also reduce the training overhead when new control algorithms are introduced, as only a performance model for that algorithm needs to be learnt, whereas a per-portfolio selector would have to be completely re-learned.

CONCLUSIONS & FUTURE WORK

In this paper algorithm selection has been presented as a way to improve network operational performance when no single control algorithm can provide the best performance for all conditions. Both the need for and the effectiveness of algorithm selection has been demonstrated through a case study of voltage control on an 11 kV network.

Future work will compare per-portfolio selection with per-algorithm selection, as well as analysing the appropriateness of different machine learning techniques for these selection approaches. Power flow management will be considered as a control task as well as voltage control, with control objectives including loss minimisation, conservative voltage reduction and increasing penetration of renewables. A variety of algorithms will be implemented and tested against these tasks and performance objectives. The range of test networks will be expanded to include numerous voltages, different network topologies and a diverse array of devices. Important intra- and inter-network features and interactions will be characterised so that selections can be made for arbitrary networks and zones within networks.

REFERENCES

[1] Smart Grids European Technology Platform: http://www.smartgrids.eu/

[2] T. Xu et al., 2011, "Case-based reasoning for coordinated voltage control on distribution networks," *Electr. Pow. Syst. Res.*, vol. 81, no. 12, 2088–2098

[3] S. Jupe and P. Taylor, 2009, "Strategies for the control of multiple distributed generation schemes", *20th Intl. Conf. and Exhib. on Electricity Distribution (CIRED)*

[4] S. D. J. McArthur et al., 2012, "The Autonomic Power System – Network Operation and Control Beyond Smart Grids", *3rd IEEE PES ISGT Europe*, IEEE

[5] J. R. Rice, 1976, "The Algorithm Selection Problem", *Advances in Computers*, vol. 15, 65–118

[6] L. Xu et al., 2008, "SATzilla: Portfolio-based Algorithm Selection for SAT", *Journal of Artificial Intelligence Research*, vol. 32, 565–606

[7] H. Guo and W. Hsu, 2007, "A machine learning approach to algorithm selection for NP-hard optimization problems: a case study on the MPE problem", *Ann. Oper. Res.*, vol. 156, no. 1, 61–82

[8] M. Graff and R. Poli, 2010, "Practical performance models of algorithms in evolutionary program induction and other domains", *Artificial Intelligence*, vol. 174, no. 15, 1254–1276

[9] M. Hall et al., 2009, "The WEKA Data Mining Software: An Update", *SIGKDD Explorations*, vol. 11, no. 1, 10–18

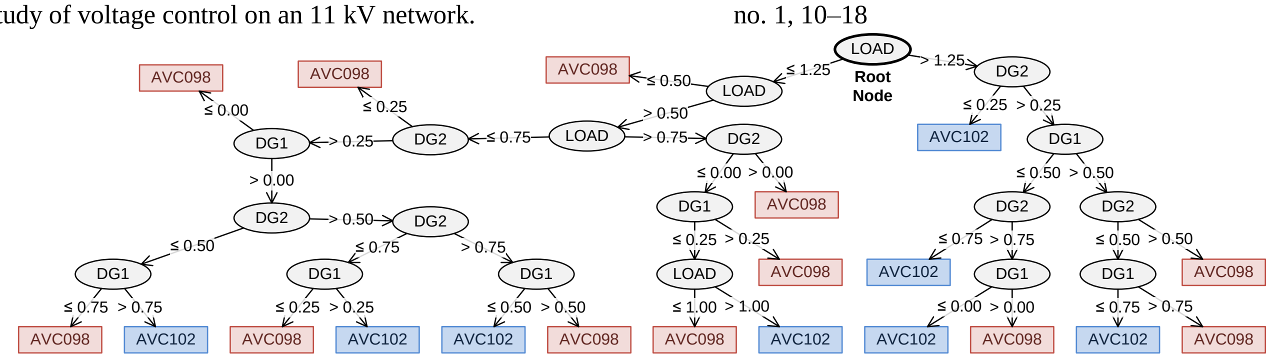


Figure 7 – Learnt classification tree for voltage control algorithm selection