Line Selection and Algorithm Selection for Transmission Switching by Machine Learning Methods

*

Zhu Yang
Industrial Engineering & Operations Research
University of California, Berkeley
Berkeley, USA
zhuyang@berkeley.edu

Shmuel S. Oren
Industrial Engineering & Operations Research
University of California, Berkeley
Berkeley, USA
oren@ieor.berkeley.edu

***Abstract*—Since the initial proposal of the Optimal Transmission Switching problem, a mixed integer program and different heuristics have been presented to achieve considerable cost reduction within a practical time frame. This paper proposes two machine learning based methods to further reduce the computation time as well as cutting down the generation cost. The first method is to apply machine learning algorithms to prioritize the possible line switching actions. The second method is to use machine learning to develop effective algorithm selectors among transmission switching algorithms suggested in the literature. The proposed methods are tested on IEEE 118-bus test case and FERC 13867-bus test case. The results demonstrated that both line selection and algorithm selection offer performance benefits over using the single transmission switching algorithm in the previous literature.**

***Index Terms*—Transmission switching, machine learning, algorithm selection**

NOMENCLATURE

Sets and Indices:

n	Bus.
N	Set of all buses.
g	Generator.
G_n	Set of generators at bus n.
k	Transmission line.
K	Set of all lines.
$\hat{K}$	Set of transmission lines in service.
$\bar{K}$	Set of transmission lines out of service.
K_n^{to}	Subset of lines with n as ”to” bus.
K_n^{fr}	Subset of lines with n as ”from” bus.
n_k^{to}	”To” bus for linke k.
n_k^{fr}	”From” bus for linke k.

Parameters:

B_k	Susceptance of line k.
c_g	Unit cost of power from generator g.
P_g^M	Maximum power from generator g.
P_g^m	Minimum power from generator g.
P_k^M	Maximum power flow on line k.
P_k^m	Minimum power flow on line k.
θ^M	Maximum voltage angle difference.
θ^m	Minimum voltage angle difference.
P_n^{dem}	Demand load at bus n.

Decision Variables:

P_g	Power from generator g.
P_g	Power flow on line k.
θ^n	Voltage angle at bus n.
z_k	Decision to have line k on or off. Binary: 0 (out of service) /1 (in service).
s_k	Decision to switch line k. Binary: 0 (no switch) /1 (switch).

I. INTRODUCTIONS

Power networks, composed of the generators, loads and transmission lines, are typically large and complexly interconnected systems. Power dispatched from generators flow to loads to satisfy their respective demand, according to laws of physics and satisfying the power flow limits of transmission lines. The power flow on an individual transmission line is therefore, influenced by the topology of the network, the properties of its components as well as the loading and generation pattern. This makes cost reduction by the transmission switching, which changes the topology of the network, possible.

Though the transmission network is traditionally viewed as a static component of the power system, system operators have been using transmission switching as a way to tackle the voltage control problem. Fisher et al. [1] first proposed a formal treatment of transmission switching as an optimization problem aiming at reducing the dispatching cost over the entire network. The proposed DCOPF based Optimal Transmission Switching (OTS) is a mixed integer program. It results in more than 25% cost savings on IEEE 118 bus test case, but also take a very long runtime for a good solution, which prohibits it from practical application. Several heuristics have been investigated to tackle the computational challenge caused by the inherent curse of dimensionality of the MIP formulation. There are two mainstream heuristics. The first is to consider possible line switches one at a time and solve a series of DCOPF problems, which could also be time consuming if considering all the possible line switches. The authors in [2] show that high performance computing can be used to parallelize the computation and improve performance. A priority list can be established by using a sensitivity factor developed from the dual problem to find the desirable line switches faster [3]. Alternatively, authors in [4] use the power transfer distribution factors (PTDFs) and "flow canceling transactions", which eliminates the need to solve DCOPFs. This makes the solution more scalable in network size and faster. However, computing the flow cancelling transactions is practical only for a limited number of line switches.

From a broader perspective, selecting the best algorithm to solve a given problem has been the subject of many studies. The algorithm selection problem, originally described by Rice [5] , has attracted a great deal of attention over the last decades. With the development of various algorithms for a single application, researchers have come to realize it is very difficult to find one best algorithm for the application with different properties or data inputs. The algorithm selection problem can be viewed as a learning problem: the aim is to learn a model that captures the relationship between the properties of the datasets and the algorithms, in particular their performance. This model can then be used to predict the most suitable algorithm for a given new dataset. Machine learning has a rich history in algorithm selection for various applications. It is a classification problem which are trained by supervised learning using a number of training examples consisting of observations labelled with the correct class. The selector is trained to pick the best algorithm with inputs being parameters of the system's status. Many machine learning techniques for producing classifiers have been used to create algorithm selectors. A meta-learning inspired framework for analyzing the performance of heuristics for optimization problems by neural networks is proposed by [7]. The effectiveness of an integrated algorithm selection method is demonstrated in simulation systems with decision trees when users have limited knowledge of the underlying algorithms and their implementations in [8]. The authors of [9] show the performance of support vector machine based automatic tuning system for computational kernels. The authors of [10] present network state based algorithm selectors for the power flow management and show performance benefit based on IEEE 14- and 57-bus network and a real network.

This paper is structured as follows: Section II provides background on algorithm selection and machine learning. Section III illustrates the test case networks. Section IV presents the existing transmission switching heuristics based on DCOPF and a new machine learning based line selection algorithm and compares their performances. Section V describes the algorithm selector developed with machine learning for transmission switching problems and shows the performance benefits of such selectors. Section VI discusses the performance improvement in terms of both computation time and cost reduction brought by these two machine learning based methods and Section VII gives concluding remarks.

II. Algorithm Selection

A. Algorithm Selection Problem

Following Rice [5] and Vanschoren [6], the Algorithm Selection Problem can be characterized by the following four elements:

- The Problem Space (P), characterized by all the inputs x in the dataset used for the study. In this paper each x represents a different network state.
- The Feature Space (F), characterized by the key characteristics produced by a feature extraction process $f(x)$, that can be used to represent the problem. In transmission switching features it could be the line status or the loading conditions of each network state.
- The algorithm Space (A), containing the set of algorithms from which we can obtain a solution to a given problem. In this paper it contains heuristics proposed to solve the transmission switching problem.
- The performance measures space (Y), containing ranges of measures that characterize the behavior of an algorithm on a given problem. In this paper the performance is measured by the cost reduction from the line switches.

Solving the algorithm selection problem can be stated as follows:

For a given problem instance $x \in P$, with features $f(x) \in F$, find the selection mapping S(f(x)) into the algorithm space A, such that the selected algorithm $\alpha \in A$ maximize the performance mapping $y(\alpha \in A) \in Y$.

B. Algorithm Selection Methodologies

In an ideal world, we would know enough information about the algorithms and dataset to choose the most suited algorithm based on certain characteristics of a problem to solve. However in reality, the systems like power grid are too vast and complex for such analysis. Due to the fact that a large amount of data is involved in the various applications including power flow problems, machine learning is a natural choice to derive selectors. Several most prevailing machine learning methodologies are discussed here.

1) Case-based reasoning: As first introduced in [11], case-based reasoning chooses algorithms for the existing problem with knowledge of past problems. Instead of trying to learn what characteristics affect the performance, it just used the performances of past known problems to infer performance on new problems. The most intuitive and commonly used case-based reasoning algorithm is nearest neighbor classifier. WEKA IBk nearest neighbor classifier with 10 nearest neighbors is used in this paper.

2) Classification: Intuitively, algorithm selection is a simple classification problem - label each problem instance with the algorithm from the algorithm space that should be used to solve it. We can solve this classification problem by a classifier that discriminates between the algorithms based on the characteristics of the problem.

Many of the numerous machine learning techniques for producing classifiers have been used to create algorithm selectors, most popular two being artificial neural networks and decision trees. In this paper we use "MultilayerPerceptron" ANN implementation and "J48" implementation of classification tree from Weka machine learning software.

III. Case Study Networks

A. IEEE 118-bus Network

The first case study network is a modified version of the IEEE 118-bus system. It consists of 118 buses, 19 generators, and 186 lines. Ratings have been assigned to a number of the network branches so that several lines are fully loaded, or congested, thus creating conditions where the transmission switching can be applied to alleviate the overloads. To train and test the algorithms and selectors, different network states were randomly created covering a range of credible conditions. In each network state, the total load and each of the generators' output were scaled by random variables drawn from independent uniform distributions. 30000 network states were generated for training and another 10000 were generated for performance evaluation.

B. FERC 13867-bus Network

The second test case is FERC 13867-bus Network derived from the data set we obtained from PJM Regional Transmission Organization. The system consists of 13,867 buses, 1,011 generators and 18,824 branches. Compared to the IEEE 118-bus system, this network is much more realistic and representative of a industrial scale power network. We also obtained the loading condition of a typical summer day which was used to produce 4000 network states drawn from independent uniform distributions for the purpose of training and evaluation. Note that for this network we used fewer data for training and evaluation due to computational power limit.

IV. Transmission Switching Heuristics

A. Model Description

Based on the Optimal Transmission Switching model in [1], we present the following single period economic dispatch model based on DCOPF. It is a mixed integer program where z_k represents the switching decision. $z_k = 1$ means the line is on and $z_k = 0$ means the line is off. We partition the set of lines $K = \hat{K} \cup \bar{K}$ in the network into two sets of lines: $\hat{K}$ representing the set of lines in service and $\bar{K}$ representing the set of lines out of service. The Kirchhoff's voltage and current laws are linearized and losses and reactive power flows are ignored. The objective is to minimize the generation cost. Voltage angle limits are imposed by Eq. (1b) and the capacity limits on generating unites are imposed by Eq. (1c). Eq. (1d) ensures the power balance for each bus. For lines originally in service, Eq. (1e) makes sure the flow respects the line flow limits if it stays on and the flow is zero if it is to be switched off. M in Eqs. (1f) and (1g) is a very large number that makes sure Kirchhoff's law holds when the line stays on. When the line is out of service at the beginning, the flow on it must be zero, as Eqs. (1h) and (1i) states.

$$\min_{\theta_n, P_g, P_k} \sum_{n \in N} \sum_{g \in G_n} c_g P_g$$

s.t.

$$\theta_n^m \le \theta \le \theta_n^M, k \in K \tag{1a}$$
$$P_g^m \le P_g \le P_g^M, n \in N \tag{1b}$$
$$\sum_{k \in K_n^{to}} P_k - \sum_{k \in K_n^{fr}} P_k + \sum_{g \in G_n} P_g = P_n^{dem}, n \in N \tag{1c}$$
$$P_k^m z_k \le P_k \le P_k^M z_k, k \in \hat{K} \tag{1d}$$
$$-P_k + z_k B_k(\theta_{n_k^{fr}} - \theta_{n_k^{to}}) + (1 - z_k)M \ge 0, k \in \hat{K} \tag{1e}$$
$$-P_k + z_k B_k(\theta_{n_k^{fr}} - \theta_{n_k^{to}}) - (1 - z_k)M \le 0, k \in \hat{K} \tag{1f}$$
$$0 \le P_k \le 0, k \in \bar{K} \tag{1g}$$
$$P_k = 0, k \in \bar{K} \tag{1h}$$
$$z_k \in \{0, 1\} \tag{1i}$$

The above model, being a mixed integer program, faces practical computational challenges. Even for the IEEE 118-bus test case it takes more than half an hour to solve within 9e-6 optimality gap on a four processor laptop. It prompts a greedy approach which only considers one line switch at a time. The following modified DCOPF, a linear program, can be solved fast and it will give the new cost after the line switch if we move the line in consideration from set $\hat{K}$ to $\bar{K}$.

$$\min_{\theta_n, P_g, P_k} \sum_{n \in N} \sum_{g \in G_n} c_g P_g$$

s.t.

$$\theta_n^m \le \theta \le \theta_n^M, k \in K \tag{2a}$$
$$P_g^m \le P_g \le P_g^M, n \in N \tag{2b}$$
$$\sum_{k \in K_n^{to}} P_k - \sum_{k \in K_n^{fr}} P_k + \sum_{g \in G_n} P_g = P_n^{dem}, n \in N \tag{2c}$$
$$P_k^m \le P_k \le P_k^M, k \in \hat{K} \tag{2d}$$
$$P_k = B_k(\theta_{n_k^{fr}} - \theta_{n_k^{to}}), k \in \hat{K} \tag{2e}$$
$$0 \le P_k \le 0, k \in \bar{K} \tag{2f}$$
$$P_k = 0, k \in \bar{K} \tag{2g}$$

Following the idea of Fuller [3], in order to come up with a reasonable criterion for selecting which lines to switch, we would like to know the sensitivity of the optimal cost on the switching action. For this reason, we express the above DCOPF equivalently as the following nonlinear program:

$$\min_{\theta_n, P_g, P_k} \sum_{n \in N} \sum_{g \in G_n} c_g P_g \tag{3a}$$

s.t.

(2b), 2(c) and

$$\sum_{k \in K_n^{to}} P_k - \sum_{k \in K_n^{fr}} P_k + \sum_{g \in G_n} P_g = P_n^{dem}, n \in N, [\rho_n] \tag{3b}$$

$$P_k^m(1 - s_k) \leq P_k \leq P_k^M(1 - s_k), k \in \hat{K}, [\lambda_k^-, \lambda_k^+] \tag{3c}$$

$$P_k^m s_k \leq P_k \leq P_k^M s_k, k \in \bar{K}, [\lambda_k^-, \lambda_k^+] \tag{3d}$$

$$P_k = B_k(1 - s_k)(\theta_{n_k^{fr}} - \theta_{n_k^{to}}), k \in \hat{K}, [\psi_k] \tag{3e}$$

$$P_k = B_k s_k(\theta_{n_k^{fr}} - \theta_{n_k^{to}}), k \in \bar{K}, [\psi_k] \tag{3f}$$

$$s_k = 0, k \in K, [\gamma_k] \tag{3g}$$

The above model 3 is mathematically equivalent to model 1, but more complicated by introducing a new variable s_k which represents the switching decision. For a line in service ($k \in \hat{K}$), $s = 1$ means it switches the line off and $s = 0$ means the line stays on, and vice versa. The dual variable γ indicates the rate of change of the objective function with respect to a small increase in the right hand side of Eq. (3g). Therefore it can be used as an indicator of possible cost reduction resulting from a line switch. A priority list can be produced by ranking the lines by ordering their respective γ from smallest to the largest (more negative γ indicates larger possible decrease in objective function, i.e. higher cost reduction). From KKT conditions we can derive the following fomular:

$$\gamma_k = \quad P_k^M \lambda_k^+ + P_k^m \lambda_k^- + B_k(\theta_{n_k^{fr}} - \theta_{n_k^{to}})\psi_k, k \in \bar{K} \tag{4a}$$

$$\gamma_k = \quad P_k(\rho_{n_k^{fr}} - \rho_{n_k^{to}})\psi_k, k \in \hat{K} \tag{4b}$$

We can see that all the variables on the right hand side are either parameters of the problem or the optimal primal/dual variables from model 2. Calculating γ doesn't necessarily require solving model 3, a nonlinear program.It can be obtained by just solving a linear program: model 2. This drastically reduces the computation time and makes producing a priority list operationally feasible.

B. Transmission Switching Algorithms

1) Enumeration of All Lines: The first algorithm is a direct enumeration of all the lines. For every possible line switch, a DCOPF with the switched line is run to compare the cost after the line switch with the original cost without the line switch, We select the line switching action which results in the greatest improvement. We iterate until we can find no improving switching action.

2) Line Selection with Priority Listing: The second algorithm involves ranking the lines according to the sensitivity factors computed in Eqs. (4a), (4b). In this algorithm we run DCOPF with a single switched line according the the priority list and implement the first switching action that results in an improvement. We first evaluates the first k lines in the priority list. If a cost reduction is found, we implement the line switch with the most cost reduction and stop. Otherwise we move on to next k lines until the list is exhausted.

3) Line Selection with Machine Learning: Due to the nature of the power network as a large and complex system and the fact that most properties of the network remain unchanged after a line switch, machine learning seems a natural choice for the line selection. 30000 test cases are created for IEEE network and 3000 test cases are created for FERC network, for training, tuning and validation. First we run the line enumeration algorithm on all the test cases so we have the complete information on the performances of every line switch which we can label now. Then another 10000 test cases of IEEE network and 1000 test cases of FERC network are used to evaluate the performance of the machine learning algorithm.

We used three established machine learning algorithms which are 10 nearest neighbor, artificial neural network and decision tree, as introduced in Section II.B . These are the most used classification methods that can take in the the parameters and loading conditions of the power network and produce a list of high priority line switches. The process of this algorithm is very similar to the line selection with priority listing. The only difference is that here we use machine learning to produce the list of lines which are worth evaluating than others.

C. Results

The average cost reductions from the three transmission switching algorithms described in the previous section are listed in Table I. Due to running time constraints we only perform 10 line switches at maximum for each case. After 10 switches the cost reduction gets very close to best know optimal for the 118-bus IEEE network, but not so for FERC network, understandably due to the size of the FERC model. We can see that for non-machine learning algorithms line enumeration always performs better than line selection with priority listing. This is expected since line enumeration is a generally more robust algorithm and at every step it examines every single possible line switch. However they both perform worse than 10 nearest neighbor and artificial neural network based line selection. The best performer is neural network with little surprise for its consistent impressive performance with continuous-valued inputs, high tolerance to noisy data and ability to classify untrained patterns. The idea behind K nearest neighbor is simple but here it well captures the property of the power network that the switching action of a similar network state can be highly relevant and suggestive. The decision tree approach performs worse than others. The possible explanation is that it is most suitable for linear separable classes which is not the case for power systems. And data for transmission switching problem contains a lot of noise and outlier which decision tree approach is sensative to.

A significant advantage of the machine learning based line selection algorithm, besides its superior performance in cost reduction, is that once it completes the training, the runtime to select a line switch with cost reduction is negligible. In the previous algorithms, solving DCOPF and line selection are done online, which means for a practical power network

TABLE I
AVERAGE % COST REDUCTIONS OF TRANSMISSION SWITCHING ALGORITHMS (10 SWITCHES)

Algorithm	IEEE Case	FERC Case
Line Enumeration	22.34%	1.94%
Line Selection with Priority Listing	21.86%	1.35%
10 Nearest Neighbor	22.50%	2.50%
ANN	23.79%	2.95%
DT	17.80%	1.21%

such as FERC network, it can take hours on a commercial laptop. It is especially critical for line enumeration algorithm, where a DCOPF optimization problem has to be solved at each step and for every branch. However if we use the machine learning approach, at each step we only need to examine a few switches that the algorithm suggests, which takes a few seconds, therefore saving computation power and time. It makes the real time transmission switching practical within the computational power of a system operator.

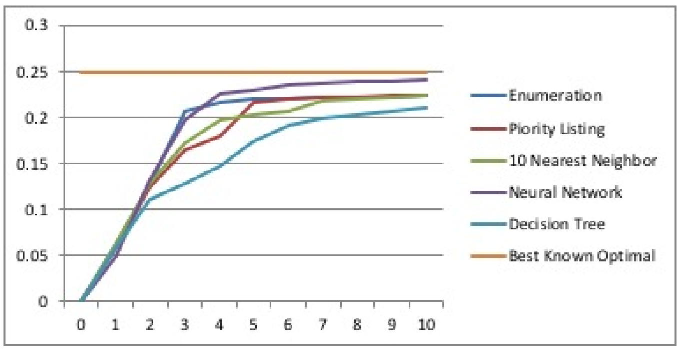

Fig. 1. Average % Cost Reduction for First 10 Switches(IEEE 118 bus Case)

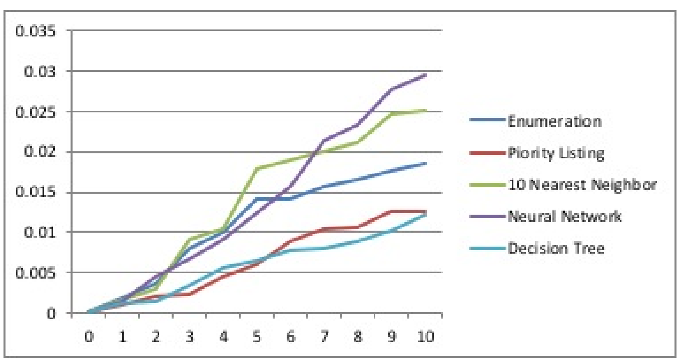

Fig. 2. Average % Cost Reduction for First 10 Switches(FERC 13867 bus Case)

V. ALGORITHM SELECTION FOR TRANSMISSION SWITCHING

A. Generation of Algorithm Selector

As discussed in Section I, there are two mainstream transmission switching formulations: one based on DCOPF and the other with PTDF and flow canceling transactions. This section presents a novel method for creating algorithm selectors for transmission switching. In the algorithm space, three algorithms are considered: line enumeration, line selection with priority listing and PTDF based method. The first two algorithms are discussed in Section IV.A. We use the PTDF based algorithm as illustrated in [4]. The selectors are in the form of classifiers that take measurements of the network as input and provide an algorithm selection as output. To create an algorithm selector, the following five steps are followed:

Step 1: Generate algorithm performance dataset to be used for building selectors. For IEEE 118 Bus network, a performance dataset is created by testing each algorithm on 30000 test cases generated for each network. The performance datasets used to create the selectors, and the states within it are separate from the 10000 test cases that are used for testing the performance of the selectors and algorithms. For FERC network, the procedure is similar except for the fact that only 3000 test cases are used for training and 1000 are used for evaluation.

Step 2: Split the algorithm performance dataset into equal parts for training, tuning and validation.

Step 3: Iterate over possible selection sets and create selectors. In this step first we train a selector with training and tuning parts of the dataset then evaluate the selectors with the validation dataset.

Step 4: Re-split algorithm performance dataset into equal parts for training and tuning.

Step 5: Take the most effective selector built in Step 3 and re-train it.

We still use the same three machine learning algorithms: k nearest neighbor, artificial neural network and decision tree, described in Section II.B.

TABLE II
AVERAGE % COST REDUCTIONS OF TRANSMISSION SWITCHING ALGORITHMS AND SELECTORS (10 SWITCHES)

Algorithm	IEEE Case	FERC Case
Line Enumeration	22.34%	1.94%
Line Selection with Piority Listing	21.86%	1.35%
PTDF Method	20.98%	1.40%
10 Nearest Neighbor Based Selector	22.87%	1.87%
ANN Based Selector	23.60%	2.93%
DT Based Selector	22.44%	2.56%
Oracle Selections	24.60%	3.24%

B. Results

Table II shows the percentage cost reduction of individual transmission switching algorithms and the algorithm selectors developed in Section V.A, for both IEEE 118-bus network and FERC network respectively. The last row of the tables shows the cost reduction achieved based on an "oracle" that has perfect a priori knowledge of which algorithm will be most effective. In other words, this is the performance achieved with optimal algorithm selection decisions made for each line switch. Note that it does not mean the line selection is the optimal.

For IEEE 118-bus test case, all three selectors show performance improvement from the best performer of individual transmission switching algorithm - Line Enumeration. The cost reduction improvement is small, less than 2% in the best case. This is expected since the individual algorithm's solution gets very close to the best known optimal: 24.88%. Even with optimal algorithm selection at every step the cost reduction is 24.60%, only one percent higher than our best selector performer - ANN based selector.

The selectors shows more performance improvement for FERC test case. Even though the nearest neighbor based selector performs worst than the line selection, the best performer - ANN based selector results in around 50% more cost reduction than the best performer in the individual algorithm, and doubles the cost reduction of the other two. It also gets very close to the oracle selection, which means that the selector wisely chooses the algorithms so that it captures most of the benefits brought by the ability to select algorithms. DT based selector shows considerable performance improvement from the individual algorithms as well. In terms of computation time, the selector itself adds negligible time to the computation since it just performs a linear calculation to select the algorithm for use.

VI. Discussion

Machine learning has been shown to be capable of improving the performance of transmission switching algorithms both by line selection and algorithm selection. When applied to line selection, the appropriate machine learning algorithm brings the benefits in both cost reduction and computation time. It is notable that even though two of the machine learning algorithms tested show more cost reduction, the third performs worse than all the non-machine learning algorithms. If a single objective of reducing cost is to be considered only, the use of machine learning technique has to be carefully considered.

All three machine learning based selectors prove effective in terms of cost reduction for both test cases. It shows considerable improvement especially for the realistic FERC test case. Due to the high complexity of the FERC network, it is expected that no algorithm will always be the most effective, therefore rendering algorithm selector useful. Even though with the machine learning algorithm we tested, the selector gets very close to oracle selection, it isn't necessarily the best selector there is. Although we tested it on two networks that are drastically different in their scales and characteristics, it doesn't guarantee the performance improvement for all power networks.

Line selection with machine learning can cut down the computation time needed to achieve the cost reduction required. However, both the machine learning based line selection and algorithm selection require computational power and time to train the model. The offline training time can be substantial, especially for the algorithm selection. Without proper parallelization techniques for the machine learning methods, it takes more than 10 hours to train the selector for FERC case. The effective parallelization of the machine learning methods and the increasing computational power within power systems are the two critical factors that will speed up the process of creating selectors.

VII. Conclusion

In this paper we demonstrated that machine learning can be used to improve the performance of transmission switching algorithms, by both line selection and algorithm selection. We tested three machine learning algorithms for line selection and two of them result in more cost reduction than the best performer among individual algorithms. All machine learning algorithms can reduce the computation time in selecting an effective line switch. The three machine learning based algorithm selector all outperform the individual switching algorithms. With the artificial neural network based selector the cost reduction gets very close to the oracle selection. The performance improvement is especially significant for the FERC case, showing great benefit potential of algorithm selectors for real life power networks.

The good performance of neural network and decision tree methods on IEEE 118-bus and FERC test case doesn't guarantee that they can improve the performance for other networks. Due to the complexity of the power network, the machine learning algorithm used has to be carefully selected. In the future, other power system applications should be investigated to see if they would also benefit from algorithm selection based on machine learning.

References

[1] Fisher, Emily B., Richard P. O'Neill, and Michael C. Ferris. "Optimal transmission switching." IEEE Transactions on Power Systems 23.3 (2008): 1346-1355.

[2] Papavasiliou A, Oren S S, Yang Z, et al. An application of high performance computing to transmission switching[C]//IREP bulk power system dynamics and control symposium, Rethymnon, Greece. 2013.

[3] J. D. Fuller, R. Ramasra, and A. Cha, ?Fast heuristics for transmission-line switching,? IEEE Transactions on Power Systems, vol. 27, no. 3, pp. 1377?1386, August 2012.

[4] Ruiz, P. A., Rudkevich, A., Caramanis, M. C., Goldis, E., Ntakou, E., & Philbrick, C. R. (2012, October). Reduced MIP formulation for transmission topology control. In Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on (pp. 1073-1079). IEEE.

[5] Rice, John R. "The algorithm selection problem." Advances in computers. Vol. 15. Elsevier, 1976. 65-118.

[6] Vanschoren J. Understanding machine learning performance with experiment databases[J]. lirias. kuleuven. be, no. May, 2010.

[7] Smith-Miles, Kate A. "Towards insightful algorithm selection for optimisation using meta-learning concepts." Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on. IEEE, 2008.

[8] Ewald, Roland, Jan Himmelspach, and Adelinde M. Uhrmacher. "An algorithm selection approach for simulation systems." Principles of Advanced and Distributed Simulation, 2008. PADS'08. 22nd Workshop on. IEEE, 2008.

[9] Vuduc, Richard, James W. Demmel, and Jeff A. Bilmes. "Statistical models for empirical search-based performance tuning." The International Journal of High Performance Computing Applications 18.1 (2004): 65-94.

[10] King, James E., Samuel CE Jupe, and Philip C. Taylor. "Network state-based algorithm selection for power flow management using machine learning." IEEE Transactions on Power Systems 30.5 (2015): 2657-2664.

[11] Riesbeck, Christopher K., and Roger C. Schank. Inside case-based reasoning. Psychology Press, 2013.