

A Computationally Improved Heuristic Algorithm for Transmission Switching Using Line Flow Thresholds for Load Shed Reduction

Tanveer Hussain, Siddharth Suryanarayanan, and Timothy M. Hansen
South Dakota State University
Brookings, SD, USA
{tanveer.hussain; sid.suryana; timothy.hansen}@sdstate.edu

S M Shafiul Alam
Idaho National Laboratory
Idaho Falls, ID, USA
SMShafiul.Alam@inl.gov

Abstract—We present a computationally improved heuristic algorithm for transmission switching (TS) to recover load shed. Research from the past showed that changing power system topology may control power flows and remove line congestion. Hence, TS may reduce the required load shed. One of the main challenges is to find a potential TS candidate in a suitable time. Here, we propose a novel heuristic method that is capable of finding the potential TS candidate faster than existing algorithms in literature. The proposed method is compatible with both the AC and DC optimal power flows (OPF). Three metrics are used to compare the proposed algorithm with the state-of-the-art from literature to show the speedup and accuracy achieved. The proposed method is implemented on the IEEE 30-bus system, PEGASE 89-bus system, IEEE 118-bus system, and Polish 2383-bus system. The results on the large-scale Polish 2383-bus system shows that the proposed algorithm is scalable to large real-world systems. Parallel computing is implemented to further improve the computational performance of the proposed algorithm.

Index Terms—Contingency analysis, load-shed recovery, transmission switching

I. INTRODUCTION

During emergency situations in the electric power system, operators may be forced to perform load shedding to maintain system security and stability. In the case of large generation loss, load may be shed to balance the insufficient generation and to maintain the system frequency within the specified threshold of operation. In case of line contingencies, the faulted line is isolated and power is rerouted on un-faulted lines; sometimes, this may lead to line congestion when the new route has lines operating at or near their ratings. If re-dispatching generation and other corrective actions fail to avoid such post-contingency violations, operators may resort to load shedding. In all these cases, a lack of proper load shedding may lead to cascading failures and blackouts. Researchers in

the past have developed algorithms to minimize the amount of load shed while also preventing cascading failures [1], [2].

Conventionally, power system operators could control the power flows only through re-dispatching generators. In the 1980s, a new concept—now known as transmission switching (TS)—was suggested where power flows can be controlled by altering the electricity network configuration [3], [4]. TS is a planned line outage to minimize the impact of an unplanned contingency in the electric power system. Topology changes can provide power flow control in meshed networks because some topologies may favor generation re-dispatch, thus, potentially reducing the need for shedding load. Research has shown TS as an effective means of reducing line overloads [5], [6], reducing line losses [7], [8], correcting voltage violations [9], [10], and load shed reduction (LSR) [11]–[14]. Despite all these advantages, TS is not widely employed by the industry [15]. One of the main reasons for the low (or slow) adoption of TS by the industry is the complexity (i.e., computational expense) of the algorithms proposed thus far to find a potential TS candidate for larger systems [16].

One method to find the potential TS candidate is by switching lines one at a time, known as complete enumeration (CE) [6], or exhaustive search method (ESM) [12]. An advantage of CE/ESM is the guarantee of finding a TS candidate for LSR should one exist, thus, making it the standard base case for comparing newer algorithms [6], [12]. CE/ESM performs well for small systems, but for larger systems with a multitude of lines CE/ESM becomes computationally prohibitive and is intractable.

In [11], a mixed integer program (MIP)-based DC optimal load shed recovery with transmission switching model (DCOLSR-TS) is presented. DCOLSR-TS is an NP-hard problem and faced computational issues. For these reasons, [11] presented a computationally less expensive MIP heuristic algorithm (MIP-H) that allows only one TS per iteration to reduce the computational complexity of solving DCOLSR-TS. Reference [12] argues that DCOPF is a crude approximation of ACOPF—therefore DCOPF based TS models produce inaccurate results for the potential TS candidates—and presents an MIP model for AC power flows (MIPAC) by modifying

Work supported through the INL Laboratory Directed Research & Development (LDRD) Program under DOE Idaho Operations Office Contract DE-AC07-05ID14517 and subcontract 501523 to South Dakota State University (SDSU). This research made use of the “Roaring Thunder” cluster at SDSU funded under NSF grant number CNS-1726946.

three computationally costly constraints of an existing mixed-integer linear optimization model called linear-programming approximation of AC power flows (LPAC) [14], [17]. When seeking the best single switching action for IEEE 118-bus system, replacing CE/ESM by MIPAC results in an average speedup of approximately 2.3 times.

We developed a method that is more computationally efficient than CE/ESM and is compatible with both DCOPF and ACOPF. Note that no new formulation for DCOPF or ACOPF is presented in this manuscript, rather the goal is to use existing tools for finding potential TS candidates. Simulations are performed using a popular environment for optimal power flow (OPF) studies [18]. The structure of the OPF given in [18] is extensible and our work takes advantage of one of its internal extensions that models dispatchable loads as “negative generation” with negative cost and negative output corresponding to the load values. The fixed loads are converted to dispatchable loads and the default value of lost load (VOLL) in [18] is used as the curtailment price. The interested reader is pointed to Section IV of [18] for details on the dispatchable loads and formulations of the DCOPF and ACOPF problems.

Our work assumes that the power system is N-1 compliant. Hence, we considered N-2 contingencies in this manuscript. Instead of switching multiple transmission lines, authors in [6] restricted their work to a single TS based on industrial feedback. Similar to [6], our work also focuses on finding a single TS candidate after N-2 contingencies for LSR. Finding multiple TS candidates after N-k (where $k \geq 2$) contingencies using this algorithm is left for future work. Moreover, in our study, the CE/ESM is used as the base case. The DCOPF based base case (CE/ESM) is used to compare the results from the DCOPF based proposed algorithm and the ACOPF based base case (CE/ESM) is used to evaluate the results from the ACOPF based proposed algorithm.

The rest of the paper is organized as follows. Section II introduces and formulates the proposed method. The experimental setup is described in Section III. In Section IV, the effectiveness of the proposed algorithm is validated by comparing the results of the proposed method with the base case solution from CE/ESM. Section V concludes.

II. PROPOSED METHOD: THE LIMIT BRANCHES TS (LBTS) ALGORITHM

As discussed earlier, one of the main reasons for shedding load after a contingency is line congestion. Both DCOPF and ACOPF consider thermal and loading constraints of lines and are useful tools to assess load shedding. Our proposed heuristic algorithm is based on the assumption that line congestion leads to load shedding after contingencies. The effect of this assumption on the proposed algorithm is given in Section IV. Fig. 1 shows the flowchart of the proposed algorithm using both DCOPF and ACOPF formulations as shown in the shaded color (orange) blocks. If the DCOPF/ACOPF converges, our proposed algorithm will make a list of the lines operating at or above a predetermined *threshold* (e.g., 99% of the emergency line ratings). This represents a mathematical sieve named *limit*

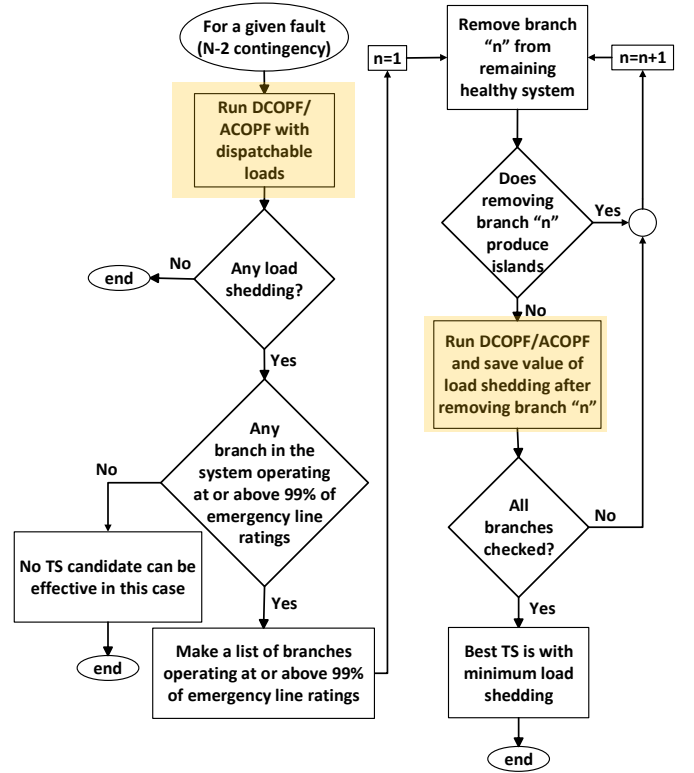


Fig. 1. Flowchart for the proposed algorithm. The blocks shown in shaded color (orange) represent the choice of either the DCOPF or ACOPF algorithm.

branches for sifting the congested lines; thus, the proposed method is titled limit branches with TS algorithm (LBTS). Similar to the CE/ESM algorithm, the proposed algorithm will remove the lines one at a time to find the potential TS candidate. The difference between the CE/ESM and the LBTS is that the sieve greatly reduces the number of test cases for determining the potential TS candidates. The proposed algorithm’s sensitivity to the chosen threshold is analyzed in Subsection IV-C.

The main contribution of the LBTS algorithm is three-fold and shown in Section IV:

- computationally less expensive,
- scalable to large-scale power systems, and
- compatible with both DCOPF and ACOPF formulations.

III. EXPERIMENTAL SETUP

We describe the choice of the test systems, the computation environment, and the contingency types studied here.

TABLE I
FOUR TEST SYSTEMS OF INCREASING ORDER OF NETWORK SIZE

Test systems	# of buses	# of generators	# of branches
IEEE 30-bus	30	6	41
PEGASE 89-bus	89	12	210
IEEE 118-bus	118	54	186
Polish 2383-bus	2383	327	2896

TABLE II
NON-TRIVIAL CONTINGENCY LIST FOR THE FOUR TEST SYSTEMS IN INCREASING ORDER OF MAGNITUDE

Test systems	DCOPF/ACOPF	CL-ALL	G1 & G2	L1 & L2	G1 & L1
IEEE 30-bus system	DCOPF	NONE	NONE	NONE	NONE
	ACOPF	148	15	NONE	133
PEGASE 89-bus system	DCOPF	420	NONE	388	32
	ACOPF	1059	8	972	79
IEEE 118-bus system	DCOPF	801	39	368	394
	ACOPF	3808	69	2770	969
Polish 2383-bus system	DCOPF	120	40	40	40
	ACOPF	120	40	40	40

A. Computational environment and test systems

Table I shows four test systems of increasing orders of network size, namely, i) the IEEE 30-bus system [19]; ii) the Pan European Grid Advanced Simulation and State Estimation (PEGASE) 89-bus system [20]; iii) the IEEE 118-bus system [19], [21]; and iv) the Polish 2383-bus system [18], are considered to test the scalability and accuracy of the proposed LBTS algorithm. For the first three test systems, simulations are performed on a 3.89 GHz desktop computer with 16 GB RAM and without utilizing parallel processing. The Polish 2383-bus is a large scale system with 2896 branches, for which the simulations are performed on compute nodes of South Dakota State University's "Roaring Thunder" Cluster [22]. Each compute node has 40 cores and 192 GB RAM. All simulations are performed using the dedicated OPF simulation environment from [18].

B. Contingency types

As mentioned above, for this paper, we focused on N-2 contingencies with an assumption that systems are N-1 compliant. We developed a contingency list (CL) based on DCOPF and ACOPF. CL is a list of contingencies that will overload any remaining component of the power system. Prior to creating the N-2 CL, the test systems were checked for compliance with the N-1 contingency requirements. In [23], for the IEEE 118-bus system, the radial transmission elements are excluded from the CL as they are not subject to reliability standards defined by Federal Energy Regulatory Commission (FERC) and the system cannot be N-1 compliant without removing the radial transmission elements from the N-1 CL. Hence, Table II shows the N-2 CL without including the radial transmission lines for the same reason mentioned in [23].

CL-ALL includes three types of N-2 contingencies, namely, i) two generator failures (G1 & G2); ii) two non-radial line failures (L1 & L2); and iii) mixed generator and non-radial line failures (G1 & L1). Further, we classified each contingency as trivial or non-trivial; the former and latter are subsets of CL that result in zero or non-zero load shed, respectively, after a generation re-dispatch [11]. Only non-trivial contingencies are of interest and Table II shows the number of non-trivial entries in the CL for the first three test systems. For the Polish 2383-bus system, LBTS is tested on a restricted test set of 120 N-2 contingency events, divided equally among the three types of contingencies. This stems from the computational intractability of CE/ESM on large test systems [11].

IV. CASE STUDY

Three metrics, namely, the percentage load shed reduction (%LSR), the worst speedup (WS), and average speedup (AVG) are used to compare LSR performance and computational efficiency of the LBTS algorithm with the base case.

A. Metrics

As discussed earlier, if there exists the best TS candidate, the base case (i.e., CE/ESM) will find it. Hence, we assume that the maximum LSR (i.e., 100%) is achieved by the base case; therefore we normalize %LSR by a factor η , given in (1),

$$\eta = \frac{\sum LS_{without\ TS}}{\sum LS_{base\ case}} \quad (1)$$

where, LS represents the load shed in MW . The %LSR is given by (2),

$$\%LSR = \eta \times \frac{(\sum LS_{without\ TS} - \sum LS_{LBTS})}{\sum LS_{without\ TS}} \times 100 \quad (2)$$

After running simulations for each contingency event from the CL in Table II, the worst speedup is calculated by (3) after finding the maximum time taken by a contingency in CL for the base case ($T_{max. base\ case}$) and the LBTS algorithm ($T_{max. LBTS}$),

$$WS = \frac{T_{max. base\ case}}{T_{max. LBTS}} \quad (3)$$

Similarly, the average speedup is calculated by (4) after finding the average time taken by the CL for the base case ($T_{avg. base\ case}$) and the LBTS algorithm ($T_{avg. LBTS}$),

$$AVG = \frac{T_{avg. base\ case}}{T_{avg. LBTS}} \quad (4)$$

B. Simulation results

The "NONE" entries in Table II represent the lack of non-trivial CL for those cases. Table III shows a comparison of the LBTS algorithm to the base case for both the DCOPF and ACOPF based formulations. In this case study, the LBTS algorithm considered the list of branches operating $\geq 99\%$ of the emergency line ratings. CL-ALL is considering all contingency types to show overall results. Note that %LSR for CL-ALL in Table III is not an average of %LSR of three contingency types. It is calculated by using (2) and by summing total LS of three contingency types with and without TS. Similarly, the WS shown in Table III is not an average

TABLE III

COMPARISON OF THE BASE CASE WITH THE LBTS ALGORITHM (CONSIDERING LIST OF BRANCHES OPERATING $\geq 99\%$ OF EMERGENCY LINE RATINGS)

		DCOPF based LBTS			ACOPF based LBTS			
		PEGASE 89-bus	IEEE 118-bus	Polish 2383-bus	IEEE 30-bus	PEGASE 89-bus	IEEE 118-bus	Polish 2383-bus
CL-ALL	%LSR	63.3	72.6	53	72.3	59	53.3	33
	Speedup	WS	65.5	51.5	219	9.3	47	112.4
		AVG	175.5	90.1	230	15.6	117	190.3
G1 & G2	%LSR	-	88.6	19	52.6	32.3	85.4	38
	Speedup	WS	-	60.2	91.4	12.2	114.6	52.8
		AVG	-	82.3	116.6	18	111	66.6
G1 & L1	%LSR	73	89.2	61	81.2	64.2	80.2	18
	Speedup	WS	71.2	42.6	339	7.5	119	39
		AVG	176	80	398	13.2	132	64.1
L1 & L2	%LSR	62.5	5.2	60	-	58.8	15.4	50
	Speedup	WS	63	62	144	-	46.9	34.9
		AVG	175	108	174	-	109	70.7

TABLE IV

COMPARISON OF COMPUTATIONAL PERFORMANCE OF THE LBTS WITH THE BASE CASE FOR THE POLISH 2383-BUS SYSTEM

Contingency type	DCOPF results				ACOPF results			
	Base case time (s)		LBTS time (s)		Base case time (min)		LBTS time (min)	
	AVG	σ	AVG	σ	AVG	σ	AVG	σ
G1 & G2	805	81.3	6.9	1.4	540	1.5	3.3	1.4
G1 & L1	2148	48	5.4	0.5	780	0.5	3.7	2.7
L1 & L2	974	254	5.6	1.2	984	0.5	5	2.2

of WS of three contingency types. Rather, it is calculated by using (3) and by finding the maximum time taken by the three contingency types for the base case and the LBTS algorithm.

In Table III, the DCOPF based CE/ESM is used to evaluate the results from the DCOPF based LBTS algorithm and the ACOPF based CE/ESM is used to evaluate the results from the ACOPF based LBTS algorithm. The results for CL-ALL in Table III shows that more than 50% LSR is achieved by considering the list of branches operating $\geq 99\%$ of the emergency line ratings, except for the Polish 2383-bus system. %LSR may be further improved by decreasing the threshold (i.e., $\geq 99\%$ of the emergency line ratings) as shown in Subsection IV-C. Considering individual contingency types, maximum %LSR achieved is 89.2% for IEEE 118-bus G1 & L1 case.

The WS and AVG improved from the smaller (i.e., the IEEE 30-bus system) to the larger test systems as shown in Table III. The reason is that the number of transmission lines increases for larger test system, which makes the base case computationally expensive. Contrarily, the list of branches operating $\geq 99\%$ of emergency line ratings in the LBTS does not increase significantly for larger systems and hence WS increases as we move from smaller to larger test cases. This feature of scalability helps to implement LBTS in real world larger systems, as shown in Table IV. Table III shows that the AVG gained by the LBTS is larger than the WS, which means most of the contingencies in CL take less time than the time taken by the worst contingency.

Table IV shows the scalability and computational performance of the LBTS on the Polish 2383-bus system. Table IV compares the time taken to solve single contingency by the LBTS algorithm with the time taken to solve single contin-

gency by the base case (CE/ESM). Note that the units of time are seconds and minutes for the DCOPF and the ACOPF results, respectively. For the Polish 2383-bus system, the LBTS reduced the time for finding the potential TS candidate from several minutes to less than ten seconds in the DCOPF based algorithm; and, it reduced the time for the same system from several hours to less than ten minutes for the ACOPF based algorithm. The reason for this speedup is that base case is checking 2894 branches, whereas the LBTS algorithm is checking 14 branches to find the potential TS candidate. Note that for the Polish 2383-bus system, the number of branches operating at or above 99% of emergency line ratings does not exceed 14 branches for any contingency type. Thus, the LBTS algorithm addresses a maximum of 14 branches. The results in Table IV can be further improved by using high-performance computing (HPC) as shown in Subsection IV-D and Table VI as each contingency can be independently calculated.

C. Sensitivity analysis

Results shown in Table III are based on the list of branches operating $\geq 99\%$ of emergency line ratings. Here, we want to show the effect of choosing a different threshold on the simulation results. Table V shows a sensitivity analysis for cases where the list of branches operating from $\geq 60\%$ to $\geq 99\%$ of emergency line ratings. This parametric sweep from $\geq 60\%$ to $\geq 99\%$, in increments of 10%, is performed for each non-trivial contingency type shown in Table III. For all cases shown in Table V, the WS increased as we move from $\geq 60\%$ to $\geq 99\%$. On the other hand, %LSR decreases as we move from $\geq 60\%$ to $\geq 99\%$. This makes sense because as we decrease the threshold from $\geq 99\%$ of emergency line ratings to $\geq 60\%$ of emergency line ratings, the list of branches will increase; hence, the WS will decrease as expected. Similarly, checking more branches will increase the %LSR. The value of %LSR of G1 & G2 for the Polish 2383-bus system in the DCOPF based LBTS algorithm is improved from 19% to 98% when the threshold for the emergency line rating is decreased from $\geq 99\%$ to $\geq 60\%$, as shown in Table V. Hence, higher %LSR may be achieved at the cost of lower speedup. In conclusion, choosing a lower threshold, i.e., $\geq 60\%$ versus choosing a higher threshold, i.e., $\geq 99\%$, is a trade-off between the speedup and accuracy.

TABLE V
PARAMETRIC SWEEP FROM $\geq 60\%$ TO $\geq 99\%$ OF THE OVERLOADING THRESHOLD FOR THE LBTS ALGORITHM

Test system	Contingency type	DCOPF-based LBTS					ACOPF-based LBTS					
		60%	70%	80%	90%	99%	60%	70%	80%	90%	99%	
IEEE 30-bus	G1 & G2	-	-	-	-	-	95.7	95.7	53	53	52.6	%LSR
		-	-	-	-	-	6	7.1	7.1	9.7	12.2	WS
	G1 & L1	-	-	-	-	-	90	90	82	82	81.2	%LSR
		-	-	-	-	-	4.3	4.8	5.6	6.2	7.5	WS
	L1 & L2	-	-	-	-	-	-	-	-	-	-	%LSR
		-	-	-	-	-	-	-	-	-	-	WS
PEGASE 89-bus	G1 & G2	-	-	-	-	-	39	39	36	32.3	32.3	%LSR
		-	-	-	-	-	17	22	23	94	114.6	WS
	G1 & L1	73.2	73	73	73	73	65	65	64.2	64.2	64.2	%LSR
		18.6	25.2	33	63	71.2	16	21	24	60	119	WS
	L1 & L2	63.3	62.6	62.6	62.5	62.5	60	59	59	58.8	58.8	%LSR
		18	20.6	36	57	63	17	21	31	35	46.9	WS
IEEE 118-bus	G1 & G2	97	94	90	90	88.6	92	92	91	85.4	85.4	%LSR
		21	28	27	46.6	60.2	17	20	25.5	29	52.8	WS
	G1 & L1	91	90	90	89.2	89.2	82.2	82	81.5	80.8	80.2	%LSR
		20	27	28	35.4	42.6	15.2	18.7	22	28	39	WS
	L1 & L2	23	23	20.2	5.2	5.2	23	22	18.8	15.8	15.4	%LSR
		21.3	33	34	44	62	14	19	24	28	34.9	WS
Polish 2383-bus	G1 & G2	98	92	89	52	19	74	74	39	38	38	%LSR
		14	20	41	65	91.4	18	24	35	90	238.3	WS
	G1 & L1	65	63	63	62	61	40	32	22	22	18	%LSR
		34	77	93	228	339	12	16	26	28	75	WS
	L1 & L2	65	63	63	62	60	92	77	63	60	50	%LSR
		20	39	40	96	144	13	20	33	68	220	WS

D. LBTS parallelization

From Fig. 1, we can observe that checking each branch from the list of branches is independent, and thus can be run in parallel on an independent processing unit. Using HPC can improve the computational efficiency of the LBTS algorithm by breaking the problem into independent sub-problems/iterations and then solving the sub-problems simultaneously on multiple cores. In this study, we showed the efficacy of HPC using the largest test case, i.e., the Polish 2383-bus system. The computational time for the DCOPF based LBTS is viable even without parallelization, as shown in Table IV. A popular numerical software for engineering applications' parallel computing toolbox is used to solve the ACOPF based LBTS algorithm on multiple cores [24]. Simulations for parallel experiments are performed on the compute nodes of South Dakota State University's "Roaring Thunder" Cluster. The operating system and software used is the same as used for results shown in Table IV.

Fig. 2 shows the computational time for the parallelized ACOPF based LBTS for a contingency in the form of average speedup (S_p) achieved. S_p is a ratio of sequential execution time (T_1) to parallel execution time (T_p), i.e., $S_p = \frac{T_1}{T_p}$ where p is the number of cores. For sequential execution, the number of cores, p , is one. The values for the sequential ACOPF based LBTS algorithm (T_1) is shown in Table IV. Table VI shows the time taken to solve single contingency by parallelizing the ACOPF based LBTS algorithm on 14 cores. Comparing the results from Table IV with Table VI shows that the parallelized ACOPF based LBTS reduces the computational time from several minutes to approximately one minute.

As discussed above, the LBTS algorithm checks 14 branches to find a potential TS candidate and by the nature

TABLE VI
COMPUTATIONAL TIMES FOR THE PARALLELIZED ACOPF BASED LBTS
USING 14 CORES ON THE HPC ENVIRONMENT

Time (s)	AVG σ	G1 & G2	G1 & L1	L1 & L2
		42.1	46.3	60.1
		16	30	24

of the LBTS algorithm, each branch iteration is independent. Hence, if we use 14 cores to parallelize the LBTS algorithm, each core will be assigned the computation to check for only one branch. Increasing the number of cores to more than 14 will not improve the speed any further because the number of cores is more than the number of branches. This is shown in Fig. 2 as the speedup becomes constant for $p \geq 14$. The two speedups that are decreased (i.e., for $p=11$ and $p=12$) might arise from an unequal work load distribution among the cores—this is a conjecture.

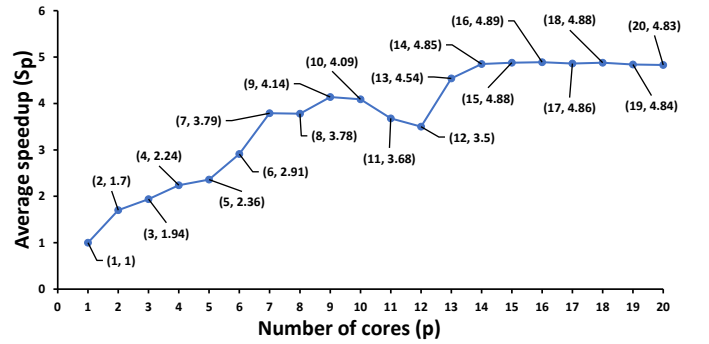


Fig. 2. The average speedup curve for the simulation of the Polish 2383-bus system using the parallelized ACOPF based LBTS. The x-axis indicates the number of cores used for the parallelization and the y-axis depicts the respective value of S_p .

V. CONCLUSIONS

Load shedding is performed under emergency conditions to maintain system security and stability. Minimized load shedding and TS is a viable solution evidenced by past research, but finding the potential TS candidate in a suitable time remains a challenge for system operators. This paper presented a computationally improved heuristic algorithm named LBTS to recover load shed after N-2 contingencies. Three metrics were used to compare the LBTS algorithm with existing algorithm in the literature. The comparisons showed that the LBTS can find the potential TS candidate faster than the well-known existing algorithm, i.e., the CE/ESM in literature. LBTS is compatible with both DCOPF and ACOPF formulation and results showed that the LBTS performed equally with both formulations. For CL-ALL, LBTS recovered more than 50% of LS with notable high speed. The authors of this paper are exploring solutions to improve %LSR of the LBTS. The WS and AVG of the LBTS algorithm also improved as we moved from smaller to larger test systems. This feature of scalability helped to implement the LBTS to real world larger system, i.e., the Polish 2383-bus system. For large scale Polish 2383-bus system, DCOPF based LBTS algorithm found the potential TS candidate in less than 10 seconds. Parallelization of code on an HPC environment improved the computational efficiency of the ACOPF based LBTS algorithm. Parallelized ACOPF based LBTS found the the potential TS candidate in a minute for the large scale Polish 2383-bus system. We also presented a sensitivity analysis for choosing a threshold other than $\geq 99\%$ of emergency line ratings. The results show that choosing a lower threshold value than $\geq 99\%$ will increase the %LSR and decrease the WS.

REFERENCES

- [1] C. Zhai, H. Zhang, G. Xiao, and T.-C. Pan, "A model predictive approach to preventing cascading failures of power systems," *arXiv preprint arXiv:1710.05184*, 2017.
- [2] M. Sinha, M. Panwar, R. Kadavil, T. Hussain, S. Suryanarayanan, and M. Papic, "Optimal load shedding for mitigation of cascading failures in power grids," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, 2019, pp. 416–418.
- [3] R. Van Amerongen and H. Van Meeteren, "Security control by real power rescheduling network switching and load shedding," in *Proc. 1980 CTGRE Report*, vol. 32, no. 2, 1980.
- [4] H.-J. Koglin and H. Muller, "Overload reduction through corrective switching actions," in *Int. Conf. Power Sys. Monitor. & Contr.*, vol. 24, no. 26, 1980.
- [5] S. A. Sadat and M. Sahraei-Ardakani, "Reducing the risk of cascading failures via transmission switching," *arXiv preprint arXiv:1810.00651*, 2018.
- [6] X. Li, P. Balasubramanian, M. Sahraei-Ardakani, M. Abdi-Khorsand, K. W. Hedman, and R. Podmore, "Real-time contingency analysis with corrective transmission switching," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2604–2617, 2017.
- [7] R. Bacher and H. Glavitsch, "Loss reduction by network switching," *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 447–454, 1988.
- [8] S. Fliscounakis, F. Zouai, G. Simeant, and R. Gonzalez, "Topology influence on loss reduction as a mixed integer linear programming problem," in *2007 IEEE Lausanne Power Tech*, 2007, pp. 1987–1990.
- [9] Wei Shao and V. Vittal, "Corrective switching algorithm for relieving overloads and voltage violations," *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1877–1885, 2005.
- [10] W. Shao and V. Vittal, "Bip-based opf for line and bus-bar switching to relieve overloads and voltage violations," in *2006 IEEE PES Power Systems Conference and Exposition*, 2006, pp. 2090–2095.
- [11] A. R. Escobedo, E. Moreno-Centeno, and K. W. Hedman, "Topology control for load shed recovery," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 908–916, 2014.
- [12] W. E. Brown and E. Moreno-Centeno, "Transmission-line switching for load shed prevention via an accelerated linear programming approximation of ac power flows," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2575–2585, 2020.
- [13] P. Dehghanian, Y. Wang, G. Gurralla, E. Moreno-Centeno, and M. Kezunovic, "Flexible implementation of power system corrective topology control," *Electric Power Systems Research*, vol. 128, pp. 79–89, 2015.
- [14] C. Coffrin and P. Van Hentenryck, "Transmission system restoration with co-optimization of repairs, load pickups, and generation dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 72, pp. 144–154, 2015.
- [15] K. W. Hedman, S. S. Oren, and R. P. O'Neill, "A review of transmission switching and network topology optimization," in *2011 IEEE Power and Energy Society General Meeting*, 2011, pp. 1–7.
- [16] M. Sahraei-Ardakani, X. Li, P. Balasubramanian, K. W. Hedman, and M. Abdi-Khorsand, "Real-time contingency analysis with transmission switching on real power system data," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 2501–2502, 2016.
- [17] C. Coffrin and P. Van Hentenryck, "A linear-programming approximation of ac power flows," *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 718–734, 2014.
- [18] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.
- [19] R. Christie, "Power systems test case archive," *Electrical Engineering dept., University of Washington*, 2000.
- [20] C. Jozs, S. Fliscounakis, J. Maeght, and P. Panciatici, "Ac power flow data in matpower and qcqp format: itesla, rte snapshots, and pegase," *arXiv preprint arXiv:1603.01533*, 2016.
- [21] S. Blumsack, *Network topologies and transmission investment under electric-industry restructuring*. Carnegie Mellon University, 2006.
- [22] South Dakota State University, "Research Computing," accessed: Oct. 5, 2020. [Online]. Available: <https://tinyurl.com/y5xto7ng>
- [23] K. W. Hedman, R. P. O'Neill, E. B. Fisher, and S. S. Oren, "Optimal transmission switching with contingency analysis," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1577–1586, 2009.
- [24] MathWorks, "Parallel Computing Toolbox," accessed: Oct. 5, 2020. [Online]. Available: <https://tinyurl.com/yjyrtq9uo>