

Load Flow Calculation For Electrical Power System Based On Run Length Encoding Algorithm

Ahmed Al Ameri, Cristian Nichita and Brayima Dakyo

Groupe de Recherche en Electrotechnique et Automatique
GREAH Lab., University of Le Havre
Le Havre, France

Emails: ahamedsaad@yahoo.com, nichita@univ-lehavre.fr

Abstract

This paper presents Data Compression Simulated algorithm for load flow calculation in electrical power systems. Real time monitor of grids required less computation time in calculation of power system analysis. Load flow problem is heart of this analysis and it basically required calculate active and reactive power flow in lines connected between buses in networks. Many topology and structures for Transmission and Distribution Systems has been proposed to reduce CPU times and memory. The proposed algorithm used Data compression technique tested different systems and results shows it is efficiency. More accuracy for large systems will need more iterations calculation which mean increasing time consumption, while Run Length Encoding (RLE) algorithm is fitness to optimized calculation numbers to exact number cause it has no zero values included. Network structure was represented as one dimension vector instead of 2D Matrix and it is effectiveness results was valid, by avoid exponential increased, by utilized this algorithm. Matlab results obtained by applied this algorithm match theoretical results.

Keywords:

Power system Analysis, Load Flow, Data Compression, Real Time Monitor, Transmission and Distribution Networks.

1. Introduction

Power system analysis plays a key role in the planning of additions or expansions to transmission and generation facilities. In fact, power flow forms the core of power system analysis and it is often the starting point for many other types of power system analyses. In addition, power flow analysis is at the heart of contingency analysis and the implementation of real-time monitoring systems. The real-time load-flow solution is much faster and requires less computer storage, is used as a technique for real-time monitoring of power systems in the presence of metering inaccuracies and data-acquisition failures [1].

In view of the topological specialty of transmission and distribution networks, many researchers has proposed

several special load flow techniques [2-4]. However, an acceptable load flow method should meet the requirements such as high speed and low storage requirements, highly reliable, and accepted versatility and simplicity.

In recent papers, networks are represents by graph theory which exploits the properties of tree with efficient use of data structure. Prasad suggested a simple algorithm for load flow analysis [5]. A new efficient load flow algorithm be modeled as graph and finally adjacency list is used for the efficient representation of Radial Distribution Network (RDN) in computer memory[6]. The distribution load flow solves directly using the single dimension vectors. An effective data structure is proposed to identify lines and number of lines available beyond the particular line [7].

2. Load Flow Calculation

Power flow analysis is an importance tool involving numerical analysis applied to a power system. In this analysis, iterative techniques are used due to there no known analytical method to solve the problem. This resulted nonlinear set of equations or called load flow equations are generated. To finish this analysis there are methods of mathematical calculations which consist plenty of step depend on the size of system. This process is difficult and takes much time to be performed.

The Newton Raphson and fast decoupled load flow solution technique and a host of their derivatives have efficiently solved the behaved power systems for a long time[8-9]. Some of ill-conditional power systems is developed when proposed method is very simple, has no mathematical approximations, and requires almost no additional storage and computation time [10].

Development of material and communication technology required reduce computation times and less memory which lead authors to using different mathematical models to find optimum solution. Partial Jacobian Updates (PJU) are included in the quasi-Newton power flow using LU

factorization method updates and/or the Matrix Modification Lemma to obtain substantial computing savings [11].

Systems is assumed that the three-phase power system is balanced and thus can be represented by its one line diagram. Load flow solution requires the construction of admittance (n x n) matrix, where n is number of bus system. The diagonal elements of the admittance matrix represent the self-admittance of the bus and the off diagonal represent the mutual admittance between buses.

$$Y = \begin{bmatrix} Y_{11} & \dots & Y_{jk} \\ \vdots & \ddots & \vdots \\ Y_{kj} & \dots & Y_{kk} \end{bmatrix} \dots\dots\dots(1)$$

The real and reactive power calculated at specific bus using initial guess and specified voltage magnitude and angle. The iteration methods were used to solve this non linear equations like (Gauss Sidle and Newton Raphson). Real and reactive power basically calculated as follows[12]:

$$P_k^{calc(x)} = \sum_{j=1}^n |V_k| |V_j| |Y_{kj}| \cos (\Theta_{kj} - \delta_k + \delta_j) \dots\dots(2)$$

$$Q_k^{calc(x)} = - \sum_{j=1}^n |V_k| |V_j| |Y_{kj}| \sin (\Theta_{kj} - \delta_k + \delta_j) \dots(3)$$

Finalized the iteration will depend on tolerance of the power mismatch which calculated by their formulae :

$$\Delta P_k^{(x)} = P_k^{sch} - P_k^{calc(x)} \dots\dots\dots(4)$$

$$\Delta Q_k^{(x)} = Q_k^{sch} - Q_k^{calc(x)} \dots\dots\dots(5)$$

Note that buses classified as three types of buses (Slack (V& δ constants), Generated buses (P&V constants) and Load buses (P&Q constants). That mean, there are two iterations to solve this problem. Firstly, it need internal iterations to calculated values of real and reactive power (2,3) when variables assumed (at beginning of solving method) or calculated (at process to achieved solution). Secondly, External iterations required to find power mismatch at tolerance (4,5), which mean increase precision will increased number of iterations. In this paper, First iteration will be called iterations for calculation while second iteration will be called iterations for accuracy.

Most of researchers concerned their attentions to reduce number of iterations for accuracy while a lot of time computation lost at iterations for calculation. No lines connected between buses will be represented as zeros elements in matrices which is not easy to avoid at calculations. Zero values in matrices spent much time and high memory through calculations and in large systems, lost time for large number of iterations of calculation will

be required. In this research, Data compression simulated algorithm proposed to find exact number of iterations for calculation which lead to minimize CPU execute time and memory space.

3. RLE (Run Length Encoding)

This section discusses simple, intuitive Data compression method that achieved by reducing redundancy. The idea behind this approach to data compression is this : If a data item d occurs n consecutive times in the input stream, replace n occurrences with the single pair nd. The n consecutive occurrence of a data item are called a run length of n, and this approach to data compression is called run length encoding (RLE)[13]. The method has been applied to text compression then to image compression.

A. In text compression:

RLE represented as simple compression algorithm used to compress containing subsequent repetitions of the same character. Run Length Coding alone is used to compress ASCII character set by a pair of RUNS and LEVELS, where RUNS shows the number of consecutive LVELES, the ASCII character in the ASCII character set [14].Code can be obtained by compressing a particular sequence, like aaaaa can be coded as #a5 (where # represents the repetition mark) while 5 is a counter for number of repetition.

The data compression ratio is a term used to quantify the reduction in data-representation size produced by a data compression algorithm. This ratio produced by RLE for text compression can be calculated by:

$$\frac{N}{N - M(L - 3)}$$

Where N is string of characters contains M repetitions of average length L each. Example: N=1000, M=10, L=3 yield a compression ratio of 1000/[1000-10(4-3)]=1.01, A better result is obtained in the case N=1000, M=50, L=10, where the ratio is 1.538.

Fig.1 is the graphical representation of the RLE algorithm [15] applying on temperature readings. The simple idea behind this algorithm is this: If a data item d occurs n consecutive times in the input stream, we replace the n occurrences with the single pair nd.

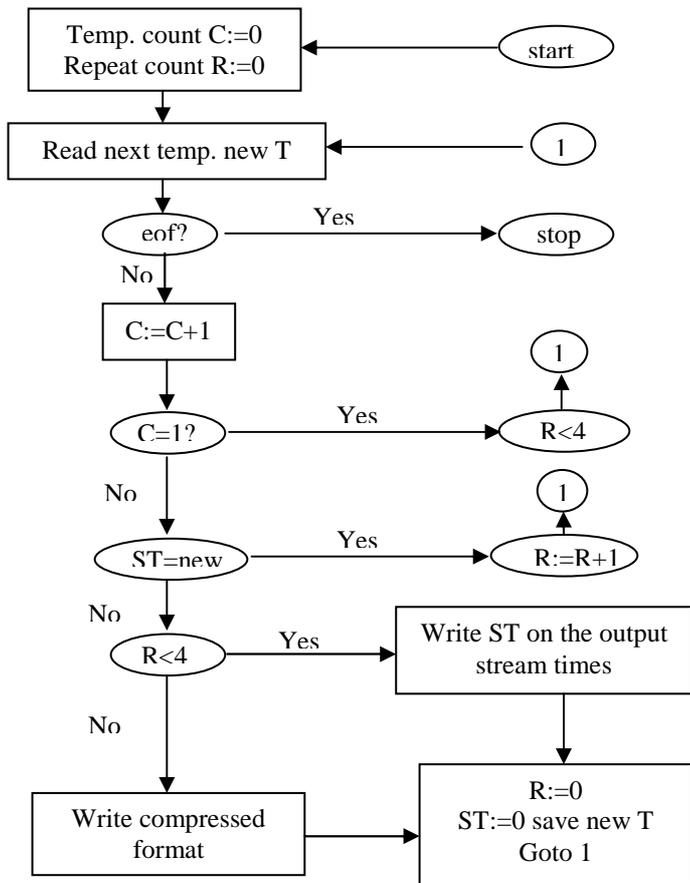


Fig.1 Flow chart for text compression

B. Image compression

A digital image consists of small dots (Pixels). Each pixel can be either one bit (white or black) or several bits (one of several colors or shades of grey). Compression an image using RLE is based on assumption that if random pixel is selected, there is a good chance that its neighbors will have the same color. The size of the compressed stream depends on the complexity of the image.

The compression ratio of a uniform area equal :

$$\frac{\text{Half the length of the perimeter}}{\text{total number of pixels in the area}}$$

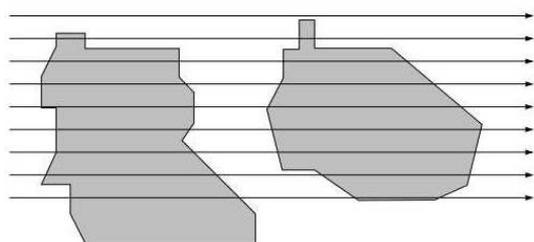


Fig. 2 Uniform area and scan lines

Best compression can be obtained when the number of scan lines traversing a better direction depends on uniform area. It can be show different direction as bellow:

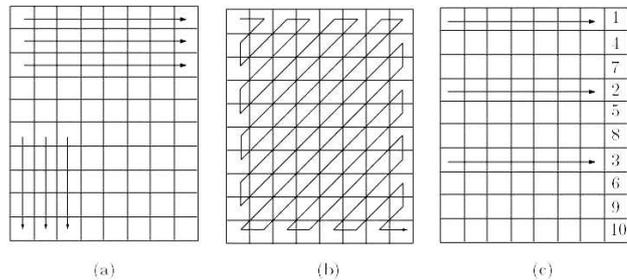


Fig.3 RLE Scanning

This algorithm is easy to implement and does not require much CPU power. RLE compression is efficient with data that contain lots of repetitive data. These can be text if they contain lots of spaces for indenting but line-art images that contain large white or black areas are far more suitable. The algorithm of the RLE creates a dictionary of the phrases that occur in the input data. When they encounter a phrase already present in the dictionary, they just output the index number of the phrase in the dictionary. This is explained in the diagram below:

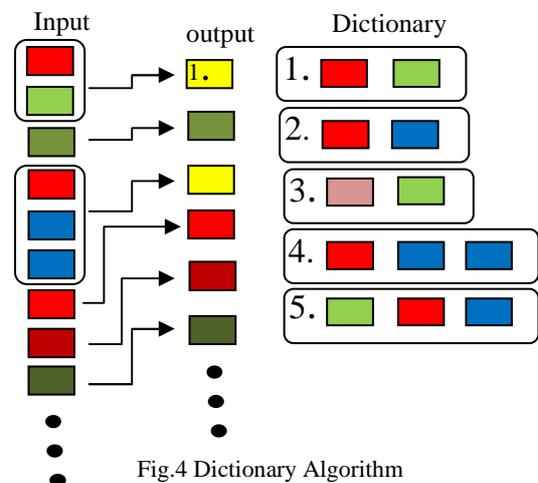


Fig.4 Dictionary Algorithm

4. RLE Algorithm

Transmission and Distribution Power system networks have different structures but all these structures has similar topology in connection view. Each bus connect to one or more other buses through one or more lines which mean number (or name) of bus will be repeated. RLE algorithm has applied to reduce this repetition and represented these networks as configuration with less iteration required in load flow calculation.

The algorithm was built as following procedure:

- 1) read input data (sending and receiving buses) for one time only.
- 2) Built matrix consists of rows that represent connection tree for each bus.
- 3) Each bus has counter to count number of lines connect with it.
- 4) Index of rows will represent index of buses.
- 5) Delete all zeros in rows and re-arrange them as one vector contains all rows of matrix.

Flow chart of RLE algorithm can be show bellow:

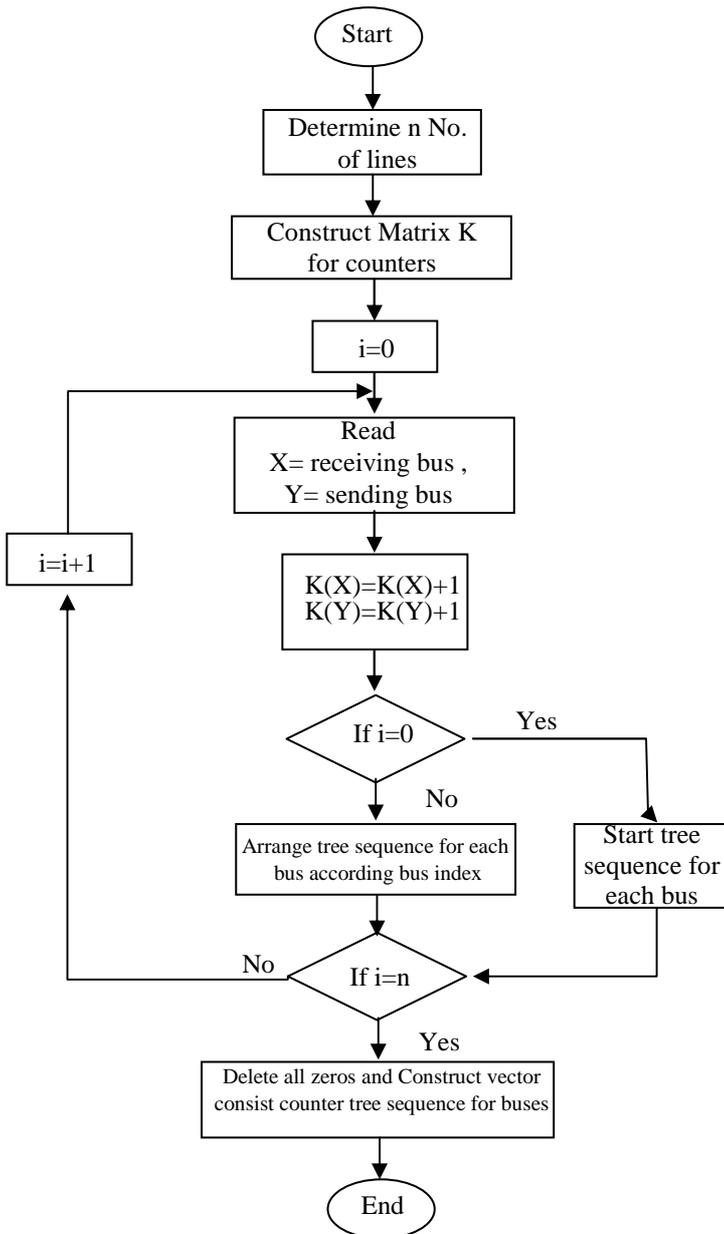


Fig.5 Flow chart for RLE Algorithm

5. Results

The effectiveness of RLE algorithm has demonstrated by selected three different numbers of buses (Standard IEEE 5, 14 and 30 buses)[16]. Differences in iteration numbers between proposed method and normal method depended mainly on number of lines, number of buses and accuracy required. It noted that differences increased exponentially when number of buses increased. That mean, high reduction in calculated time will less iteration number to calculated active and reactive power flow.

For example: for system has 5 buses and 8 lines as follows:

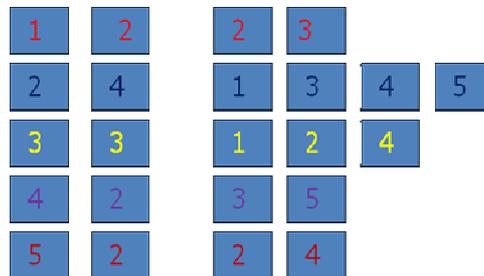
Table.1 IEEE Standard 5 Bus

Sending Bus	Receiving Bus	R	X	No. of Bus	Type of Bus
1	2	0.01938	0.05917	1	Slack
1	2	0.02000	0.06000	2	PV Bus
1	3	0.08000	0.24000	3	PQ Bus
2	3	0.06000	0.18000	4	PQ Bus
2	4	0.06000	0.18000	5	PQ Bus
2	5	0.04000	0.12000		
3	4	0.01000	0.03000		
4	5	0.08000	0.24000		

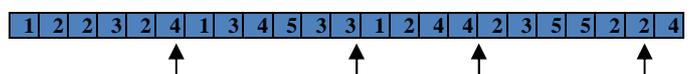
Which can be represented by two dimension matrix with 5*5 contains 1 for connection and for no connection as follows:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

This matrix need (5*5=25) iteration or (5 buses*8lines=40) iteration while at executed calculation will be only 16 to construct Jacobian Matrix as been tested in Matlab program. By RLE algorithm, number of iteration will be 16 as required and will represented as follows:



Then it will be re-constructed as vector including all data after compression as follows:



Total iteration needed equal summation of tree paths for each node (except slack bus) plus number of load buses

which mean in this case (4+3+4+2+3=16). From this example, it was proved that the less number of iteration has validated by RLE algorithm while other results for rest of systems show bellow:

Table .2 Comparison results of iterations for calculation between normal and RLE method

NO. of Buses	No. of iteration for accuracy			
	1	2	5	10
5(Normal)	40	80	200	400
5 (RLE)	16	32	80	160
14	280	560	1400	2800
14 (RLE)	40	80	200	400
30	1230	2460	6150	12300
30 (RLE)	82	164	410	820

The column diagram for iterations for accuracy and iterations for calculations show bellow:

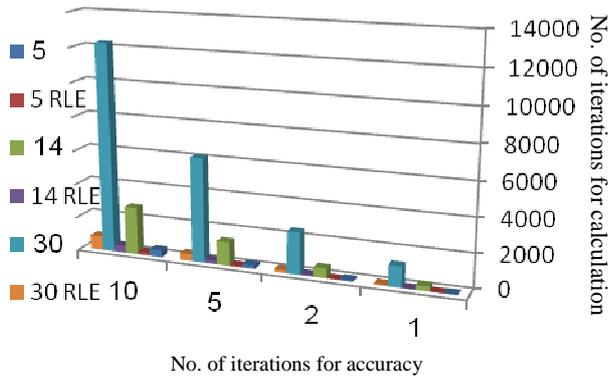


Fig.6 Result comparison by RLE and normal method

While the differences between normal method and RLE method in iterations of calculations with exponential change can be explain by linear curve bellow:

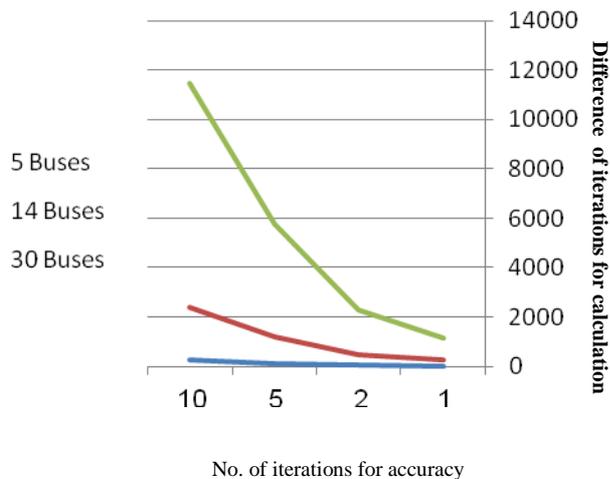


Fig.7 Differences of iterations comparison by RLE with normal method

6. CONCLUSION

An efficient data compression (RLE) method has been simulated as new algorithm to reduce the time computation spent by load flow calculations. This algorithm has tested three different types of IEEE standard buses as (5, 14 and 30 Buses). Results shows the efficiency of algorithm to reduce iterations numbers to calculated active and reactive power flow in these systems to the minimum number without losses of time. It noted that time required to implement these calculation increased exponentially with increase of number of buses and more accuracy required. Algorithm optimized CPU execution time and number of calculation iterations with less memory.

References:

- [1] Tripathy, S.C., Rao, N.D., Kumar, Arun, " Real-time monitoring of power systems using fast-decoupled load flow", Electrical Engineers, Proceedings of the Institution of (Volume:124 , Issue: 7), July 1977.
- [2] B. Scott, "Review of load flow calculation methods," Proc. IEEE, Vol.62, No.7, pp. 916- 929, July 1974.
- [3] S.K. Goswami, S. K. Basu, "Direct solution of distribution system," IEE Proc. C, Vol.188, No.1, pp.78-88, 1991.
- [4] D. Das, H. Nagi, D. Kothari, "Novel method for solving radial distribution networks," IEE Proc C, Vol.141, No.4, pp. 291-298, 1994.
- [5] K. Prasad, N. C. Sahoo, A. Chaturvedi and R. Ranjan "A simple approach in branch current computation in load flow analysis of radial distribution systems", IJEEE - International Journal for Electrical Engineering Education, Vol.44/1, PP: 49-63, January 2007.
- [6] D. P. Sharma, A. Chaturvedi, G.Purohit , R.Shivarudraswamy, " Distributed Load Flow Analysis using Graph Theory", World Academy of Science, Engineering and Technology 50 2011.
- [7] T. Ramana, V. Ganesh, and S. Sivanagaraju, " Simple and Fast Load Flow Solution for Electrical Power Distribution Systems", International Journal on Electrical Engineering and Informatics, Volume 5, Number 3, September 2013.
- [8] W.F. Tinney and C.E. Hart, "Power flow solution by Newton's method," IEEE Trans. Power App. Syst., vol. PAS-86, pp. 1449-1456, Nov. 1967.
- [9] B. Stott and O. Alsac, "Fast decoupled load flow,"IEEE Trans. Power Appl Syst., vol. PAS-93, pp. 859-869, May/June 1974.
- [10] S. Iwamoto, Y. Tamura," A Load Flow Calculation Method For Ill-Conditioned Power Systems", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, No. 4, April 1981.
- [11] Adam Semlyen, and Francisco de León," Quasi-Newton Power Flow Using Partial Jacobian Updates", IEEE Trans. Power App. Syst., vol. 16, No. 3, Aug. 2001.
- [12] Hadi Saadat, Power System Analysis, McGraw-Hill International Editions, 1999.
- [13] D. Salomon "Data Compression", 3rd Edition, Springer, 2004.
- [14] M. B. Akhtar, Adil M. Qureshi and Q. Islam, "Optimized Run Length Coding for JPEG Compression used in Space Research Program of IST", International Conference on Computer Networks and information Technology (ICCNIT), Pakistan 2011.
- [15] E. Capo-Chichi, H. Guyennet and Jean-Michel Friedt, "K-RLE : A new Data Compression Algorithm for Wireless Sensor Network", Third International Conference on Sensor Technologies and Applications, Athens, Glyfada 2009. SENSORCOMM '09.
- [16] P. Srikanth, O. Rajendra, A. Yesuraj, M. Tilak K.Raja, " Load Flow Analysis Of Ieee14 Bus System Using MATLAB", International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 5, May – 2013.