Systems of Linear Equations and their Graphical Solution

Jaime Cerda and Alberto Avalos

Abstract

The solution to a large kind of problems can be addressed by solving a system of linear equations (SLE). In particular, the main tool used to solve non-linear optmisation problems are based on Newton's method. This method solves iteratively an SLE until convergence is reached (if the solution exists). Therefore if the goal is to derive efficient algorithms to solve non-linear optimisation problems then a deeper insight into how an SLE is solved has to be done.

***Index Terms*—Systems of Linear Equations, Matrices, Graphs, Gaussian Elimination, Trees, Sparsity.**

I. Introduction

THE solution to a large kind of problems can be addressed by solving a system of linear equations (SLE). In particular, iterative methods such as Newton's method used to solve non-linear optmisation problems. This method solves iteratively an SLE until convergence is reached (if the solution exists). Therefore if the goal is to derive efficient algorithms to solve non-linear optimisation problems then a deeper insight into how an SLE is solved has to be done. In this dicument the graphical solution to an SLE is described and various special cases are analysed. To this end the organization of this document is as follows: Section II presents a graphical representation for an SLE. Then, section III moves on the focus to symmetrical SLEs (SSLE) and their graphical representation. Next section IV presents a closer look to special SSLEs whose structure are represented by a tree (TSSSLE). After becoming familiar with the different classes of SLE and their graph representation, the solution to those graphs is addressed. Section V presents the graphical interpretation for the Gaussian elimination. Next, section VI compares the different strategies to solve the TSSSLE graph using the transformations introduced in section V.

Finally section VII provides some concluding remarks.

Manuscript received July 26, 2014; revised August 5, 2014. This work was supported by the Scientific Research Coordination, Universidad Michoacana de San Nicolas de Hidalgo. J. Cerda (jcerda@umich.mx) and J. Avalos (javalos@umich.mx) are with the Electrical Engineering School, Universidad Michoacana de San Nicolas de Hidalgo, Morelia, Michoacan, Mexico

II. Graphical Representation for Systems of Linear Equations

An n-order system of linear equations represent systems where the number of equations is equal to the number of variables have the form of equation 1 i.e. n.

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^n$. The non-zero elements in $\boldsymbol{A}$, excluding those in the diagonal, define the topology of the graph which represents the SLE. If $\boldsymbol{A}$ is very sparse then the graph will have very few interconnections. Sparse systems can be solved using special algorithms which may need special data structures. Therefore a measure of sparsity is needed in order to evaluate how sparse a matrix is. Let us define n_z as the number of zeroed off-diagonal elements in $\boldsymbol{A}$, where $0 \leq n_z \leq n(n-1)$. Equation 2 defines the sparsity degree for matrix $\boldsymbol{A}$.

$$s = \frac{n_z}{n(n-1)} \tag{2}$$

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The possible values for s lie in the interval [0,1]. As the ratio between the number of zeroed elements and the number of elements if it were full grows, the sparsity degree will grow. On the other hand, $s = 0$ represents a fully connected system (i.e. every node is connected to every other node). Between these two extremes (i.e. $0 < s < 1$), $\boldsymbol{A}$ is neither full nor decoupled and its corresponding graph will not be fully connected (i.e. not all the nodes are connected among them). It turns out that many physical systems solved using linear systems are very sparse i.e $s \to 1$. In particular, in electrical power systems, the matrices which represent transmission networks are very sparse. This is the main reason why sparsity techniques have been improved by research whose goal is to solve efficiently the actual state of the power system; such as the best elimination ordering [5], [8]. Sparsity techniques have been around since at least 1970 using a technique known as bifactorization [9]. The main principles used in bifactorization are strongly directed toward exploiting the underlying matrix graph.

Therefore in order to approach the solution using its graph representation, first an appropriate model has to be derived. This model has to be able to represent the complete SLE elements (i.e. $\boldsymbol{A}, \boldsymbol{x}$, and $\boldsymbol{b}$). The model proposed in this document is based on a per equation basis. This representation is given in figure 1.

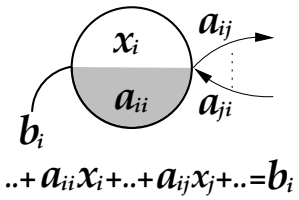


Fig. 1. Conversion from a linear system of equations to its graph model

In this model an equation is represented by two components: a node and a set of links. The node is a well defined component which consists of two subcomponents: a circle consisting of two half parts and an arc. The upper part of the circle represents the variable related to this equation which has to be solved by the system (i.e. x_i) and the lower part represents the coefficient related to this variable in equation i (i.e. a_{ii}). The arc represents the $i-th$ component in $\boldsymbol{b}$ (i.e. b_i). The second part depends on the SLE topology and is represented by links which connect the nodes. These links will be denoted as (i, j) where i and j represent the row and the column number respectively. There can be zero or more links which connect the node with some other nodes in the graph. Each link has an associated value for the coefficient located in the row i column j (i.e. a_{ij}). Perhaps it is a little absurd to consider the case where there are no external links. However, as will be shown later, this is the basic configuration which will always be pursued in order to solve the SLE.

In order to illustrate these concepts let us instantiate equation 1 to equation 3

$$\begin{pmatrix} 0.5 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 1 & -1 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -5 \\ 0 \end{pmatrix} \quad (3)$$

Applying the model defined in figure 1 to each equation, the graph shown in figure 2 is obtained.

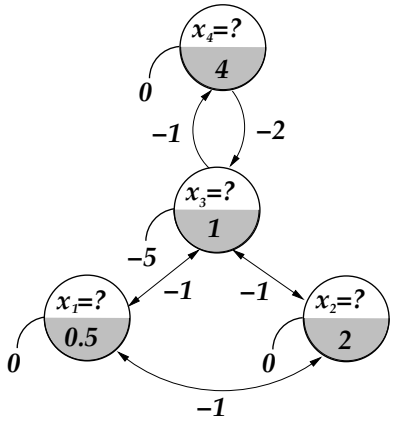


Fig. 2. Graph corresponding to the system defined by equation 3

Unidirectional links ($\rightarrow$ and $\leftarrow$) have to be used as in general $a_{ij} \neq a_{ji}$, as shown by links $(3, 4)$ and $(4, 3)$, representing elements a_{34} and a_{43}. If $a_{ij} = a_{ji}$ then these elements are represented with a bidirectional link ($\leftrightarrow$) as shown by the link $(1, 3)$, representing elements $a_{1,3}$ and $a_{3,1}$. This graph represents an asymmetric SLE (ASLE). An ASLE is a SLE where there exists at least one pair of links $(i, j), (j, i)$ where $i \neq j$, such that $a_{ij} \neq a_{ji}$ holds.

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

In this document the main aim is to express Newton's method to solve non-linear optimisation problems using a graph approach. The kind of matrices involved in such problems are symmetric, therefore the main focus will be on this subset of SLE.

III. Symmetric Systems of Linear Equations and Its Graphical Representation.

Symmetric systems of linear equations (SSLE) are SLEs where $a_{ij} = a_{ji}$ holds for all i, j. These are very well behaved matrices with some special properties; such as real eigenvalues, orthogonal eigenvalues; as analysed in [7]. This document will not deal with the analysis of the properties these systems hold. The main interest here is how to represent such systems using graphs and how to exploit them in order to solve the SLE. SSLEs are very common in physical systems, in particular, a great range of problems in electrical power systems can be addressed with SSLEs. To represent SSLEs into its graphical form, the graph representation proposed in figure 1 is modified as shown in figure 3.

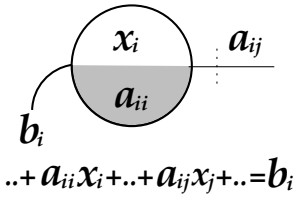


Fig. 3. Conversion from a symmetric linear system of equations to its graph model

The only modification in this variant is with regard to the unidirectional links. The graphs representing SSLEs must contain only bidirectional links. The link representation has been modified and the arrows are no longer used as they do not give any extra information. In order to illustrate these concepts let us instantiate equation 1 with equation 4 which is basically equation 3 where the element a_{43} has been set to -1 in order to be equal with element a_{34}.

$$\begin{pmatrix} 0.5 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 1 & -1 \\ 0 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -5 \\ 0 \end{pmatrix} \quad (4)$$

Applying the model defined in figure 3, yields the graph shown in figure 4

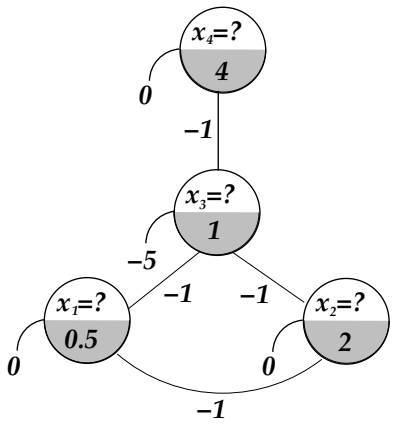


Fig. 4. Graph corresponding to the system defined by equaion 4

SSLEs can be described as perfect SLEs; they are well behaved and their properties have been known for a long time. However, there exists a subset of SSLE which besides all those properties possessed by them, have another property. They can be represented with a graph known as a tree and as a consequence all the well known algorithms regarding trees can be applied to them. For reasons which will be explained in the following chapters these will be the SLEs this document will be dealing with. Therefore, the attention will be focused on this kind of systems.

IV. Tree Structured Symmetric Systems of Linear Equations and Its Graphical Representation.

Tree structured symmetrical systems of linear equations (TSSSLE) are SSLE where the graph representing the SSLE is a tree. Based on this structure, efficient algorithms can be derived in order to solve this kind of systems. These algorithms emerge naturally, just by exploiting the properties of trees. This kind of graphs have been applied to solve electrical distribution networks whose main characteristic is its radial shape (i.e. no loops exists in the network). Therefore a tree structure can be derived for the SLE representing these systems. Algorithms to solve different problems with different degree of complexity have been proposed for distribution networks based

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

on this structure in [4], [3], [2], [6]. A deeper analysis of this kind of systems will be done in another contribution.

A tree-shaped graph has to be free of loops. This work does not deal with how to identify and remove loops from graphs. Therefore, graph 4 will be converted into a graph representing a TSSLE by removing the link $(1, 2)$. This implies removing elements a_{12} and a_{21} from matrix $\boldsymbol{A}$. Let us instantiate equation 1 with equation 5 which is basically equation 4 where elements a_{12} and a_{21} have been set to 0

$$\begin{pmatrix} 0.5 & 0 & -1 & 0 \\ 0 & 2 & -1 & 0 \\ -1 & -1 & 1 & -1 \\ 0 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -5 \\ 0 \end{pmatrix} \quad (5)$$

Applying the model defined in figure 3, leads to the graph shown in figure 5 which is graph 4 where link $(1, 2)$ has been removed.

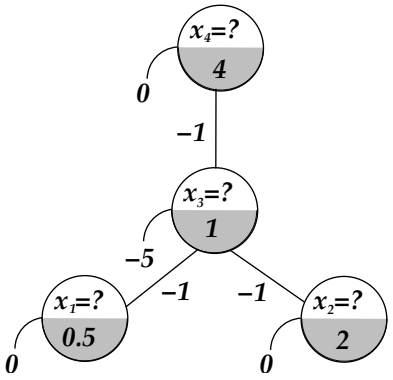


Fig. 5. Graph corresponding to the system defined by equation 5

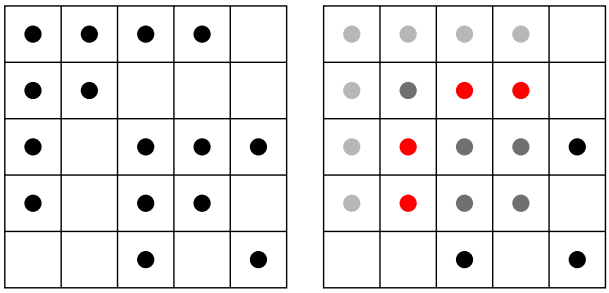
This example will be used throughout this chapter. The system will be solved using different strategies which have to lead to the same solution. To this end Gaussian elimination will be used as the main tool to solve the system.

V. Gaussian Elimination and Its graphical Interpretation

Gaussian elimination is a general method to solve a SLE. It consists of the iterative application of elementary row operations which lead the system to an echelon form. This is achieved by modifying each of the elements which do not belong to the column and row to the equation under reduction. These elements are modified using expression 6.

$$a'_{ij} = a_{ij} - \frac{a_{ik}a_{kj}}{a_{kk}} \quad (6)$$

Gaussian elimination can be regarded as a matrix transformation from $\mathbb{R}^{n\times n} \rightarrow \mathbb{R}^{n-1\times n-1}$. The resulting system has all the information needed to solve the subsystem resulting from the transformation. When dealing with sparse systems several observations have to be done. Figure 6(a) represents an sparse matrix and figure 6(b) shows the transformation it undertakes when Gaussian elimination is applied to x_1. Dark gray entries represent elements which di not change. Elements in light gray represent those whose value changes and the entries in red denote elements whose values were zero before the transformation, i.e. they were created.



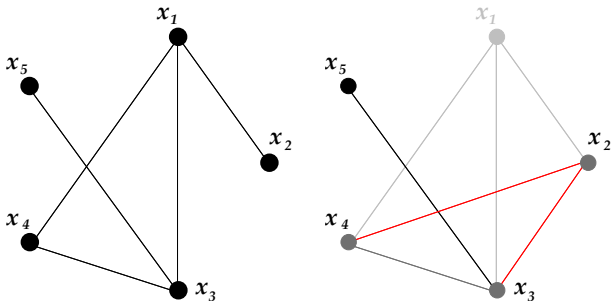
(a) Before the elimination (b) After the elimination

Fig. 6. Gaussian elimination and its matrix interpretation for a sparse matrix

Now, let us analyse the transformation in its graphical representation. To this end, let us define Γ_k as the set of nodes connected to node k. In this case let us instantiate $k = 1$ as the node to be eliminated; consequently, $\Gamma_1 = \{2, 3, 4\}$. A graph interpretation for this transformation is shown in figure 7. Here figure 7(a) represents the state of the graph before the transformation is applied and figure 7(b) represents the state of the graph after the transformation has been applied.

This shows that when node k is eliminated then the nodes which are connected to it, Γ_k, will form a complete graph among them as a result of this transformation. This is reflected by equation 7

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



(a) Before the elimination (b) After the elimination

Fig. 7. Gaussian elimination and its graph interpretation for a sparse matrix

$$\Gamma'_j \leftarrow (\Gamma_j \cup \Gamma_k) \setminus \{j,k\} \qquad \forall j \in \Gamma_k \qquad (7)$$

Where Γ_j and Γ'_j denote the neighbour nodes of node j before and after the transformation, respectively. If nodes i and j were connected before the transformation then the value for the link (i,j) which was connecting them (shown in dark gray) will be updated by equation 6. On the other hand, if they were not connected (i.e. $a_{ij} = 0$) then these interconnections would have to be created (shown in red). If no pair of nodes i,j, where $i,j \in \Gamma_k$, were connected before the transformation then a complete subgraph would be created among them. The number of links needed to build this subgraph, N_k, is given by equation 8.

$$N_k = \frac{|\Gamma_k|\,(|\Gamma_k| - 1)}{2} \qquad (8)$$

Therefore, Gaussian elimination has two costs: one which has to be applied every time is updating, and the second one is the creation of new links, known in the literature as *fill-ins*. Furthermore, from figure 7(b) link $(3,5)$ and node 5 were not used at all in the transformation. Here is where the power of sparse methods appears in systems whose components are lossely coupled as they do not deal with elements not involved in the transformation. Obviously, the burden set by sparse methods have to be avoided if the systems under study are known to be very full matrices which derive almost complete graphs tranformations. [1] gives a deeper description about the modifications of the graph as the reduction process is applied.

VI. Graph-based Solution for Symmetric Systems of Linear Equations

Solving a SLE, when translated to its graph counterpart, means to assign some value to the question marks shown in figure 5 such that they fulfills all the equations. It is desirable to end up with the same values as the initial configuration but as it will be seen this is not possible as the successive application of the Gaussian elimination will modify these values. Furthermore, the final configuration will depend on the order in which the Gaussian elimination was applied. How were these values obtained? There are several methods to solve SLE which are based on Gaussian Elimination. Here the graph is reduced by applying Gaussian elimination, one node at a time iteratively, until the graph is reduced to just one node. This method is known as forward elimination. At this point the system can be solved as its configuration is

$$a'_{ii} x_i = b'_i$$

from this x_i is solved with a value of

$$x_i = \frac{b'_i}{a'_{ii}}$$

Then a process called backward substitution can be applied by solving the previous node and so on. It is important to keep the tree structure in the elimination process as it will perform the fastest and cheapest solution for the SLE. Let us apply the Gaussian elimination to the example graph given in figure 5. The first question is which node has to be applied the elimination on? A more advanced question is which elimination order has to be applied?. This is an open question and has been addressed in different scenarios. There are several elimination orders which will take us to the solution of the system. In fact, the total number of elimination orders, N_e, for a system with n variables and n equations is n^n. However, some of them can not be applied as they would lead to an inconsistent system. An inconsistent configuration appears when the node where Gaussian elimination is to be applied is zero. This would lead equation 6 to an undefined value and would stop the reduction process. If we derive a configuration where all the nodes are zero then the system is said to be singular. Therefore, in its matrix version, this situation is avoided by

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

interchanging (or renumbering) rows and columns, provided the system is not singular. In the graph representation just an inspection have to be done at the actual node where the Gaussian elimination is to be applied. If its value is zero then its reduction is delayed to a later moment.

In this section, two different elimination orders will be applied to the graph system shown in figure 5 in order to obtain some insight about the Gaussian elimination process (applying all the possible orders implies $4! = 24$ elimination orders). The first elimination order shown in 8 is $1 \rightarrow 2 \rightarrow 4 \rightarrow 3$. First, node one, using the reduction shown in figure 5, is reduced into node 3 as shown in figure 8(a). In the same way, node 2 is reduced into node 3 as shown in figure 8(b). Finally, as for the reduction process, node 4 is reduced into node 3 as shown in the upper part of figure 8(c). Now x_3 can be solved, as shown in bottom part of figure 8(c). Once x_3 is solved, the substitution process can be applied as node $1, 2$, and 4 were connected to node 3 only. This process leads to the configuration shown in figure 8(d). As it can be appreciated, no new links are created. Furthermore, node 3 is the only node whose initial configuration is modified.

The second elimination order, shown in figure 9, is $3 \rightarrow 4 \rightarrow 2 \rightarrow 1$. Here, three new links are created when node 3 is eliminated as $|\Gamma_3| = 3$. The initial configuration for all nodes has been modified. Furthermore, in order to solve x_3 the rest of the variables have to be solved. To solve x_4, first x_1 and x_2 have to be solved. Finally, to solve x_2, first x_1 must be solved.

A. About the Importance of the Initial Configuration

In the previous examples, the modification to the initial configuration was mentioned. This is important to preserve or at least try to preserve it as much as possible as there are iterative algorithms which will be using this configuration in order to reach the solution. If the algorithm which solves the graph modifies this configuration at every iteration, then this will have to be reinstantiated in each of them.

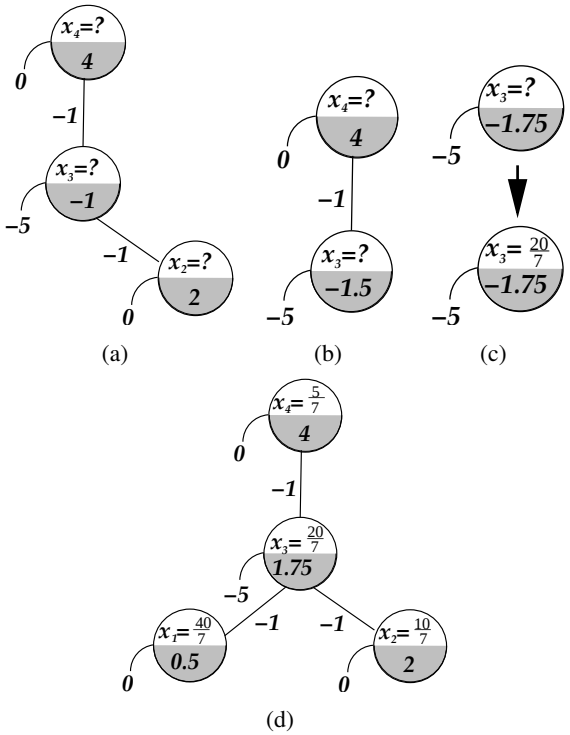


Fig. 8. The graph solution with elimination order $1 \rightarrow 2 \rightarrow 4 \rightarrow 3$

VII. Concluding Remarks

In this work the graph representations for SLEs, SSLE and TSSSLE have been presented. Here we have learnt the differences among them and the fact that $TSSSLE \subset SSLE \subset SLE$. Then Gaussian elimination and its graphical interpretation has been presented. Even that different elimination orders have the same solution when Gaussian elimination is applied, some will require less operations to reach the solution. Also depending on this elimination order a variable number of new links will be created or not. Furthermore, some elimination orderings are not allowed as they will derive graphs whose pivot where gaussian elimination is to be applied is zero.

When solving a SLE, the objective is to find the values for variables represented by vector $\boldsymbol{x}$. The basic tool to find those values will be based on the Gaussian elimination. The processing task for this graph will address the previous features so no links are created at all. A TSSSLE is a SLE which can be represented with a tree. Finally,the solution to TSSSLE does not require the creation of links,

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

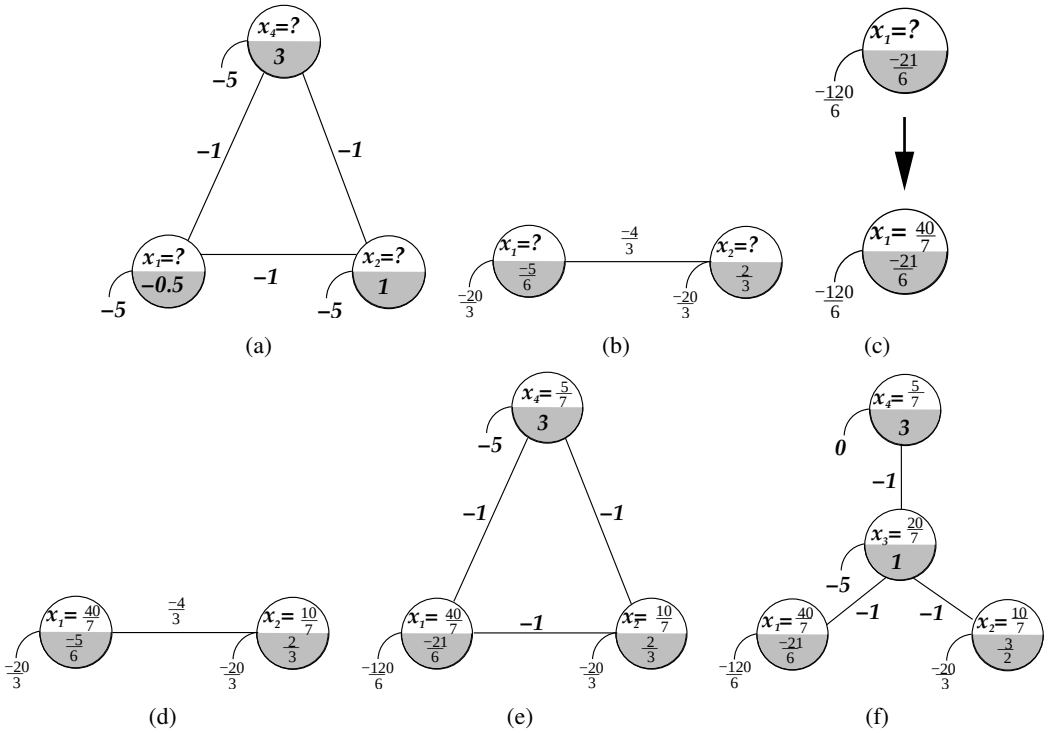


Fig. 9. The graph solution with elimination order $3 \rightarrow 4 \rightarrow 2 \rightarrow 1$

therefore is a very efficient structure which every solution algorithm must try to derive.

References

[1] Anne Berry and Pinar Heggernes. The minimum degree heuristic and the minimal triangulation process. In *Proceedings of WG 2003*, pages 58–70. Springer Verlag, 2003.

[2] G.J. Chen, K.K. Li, T.S. Chung, and G.Q. Tang. An efficient two-stage load flow method for meshed distribution networks. In *Proceedings of the 5th Conference on Advances in Power System, Control, Operation and Management, APSCOM 2000*, pages 537–542, October 2000.

[3] D. Das, H. S. Nagi, and D. P. Kothari. Novel method for solving radial distribution networks. *IEE Proceedings on Generation, Transmission and Distribution*, 141(4):291–298, July 1994.

[4] S. K. Goswami and S.K. Basu. Direct solution of distribution systems. *IEE Proceedings on Generation, Transmission and Distribution*, 138:78–85, 1991.

[5] Harry M. Markowitz. The elimination form of the inverse and its application to linear programming. *Management Science*, 3(3):255–269, April 1957.

[6] S.F. Mekhamer, S.A. Soliman, M.A. Moustafa, and M.E. El-Hawary. Load flow solution of radial distribution feeders: A new contribution. *Electrical power and Energy Systems*, 24:70–707, 2002.

[7] Gilbert Strang. *Linear Algebra and Its Applications*. Brooks Cole, 4 edition, July 2005.

[8] W. F. Tinney and J. W. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proceedings of the IEEE*, 55(11):1801–1809, 1967.

[9] K. Zollenkopf. Bifactorization: Basic computational algorithm and programming techniques. In Oxford, editor, *Conference on Large Sets of Sparse Linear Equations*, pages 76–96, 1970.

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)